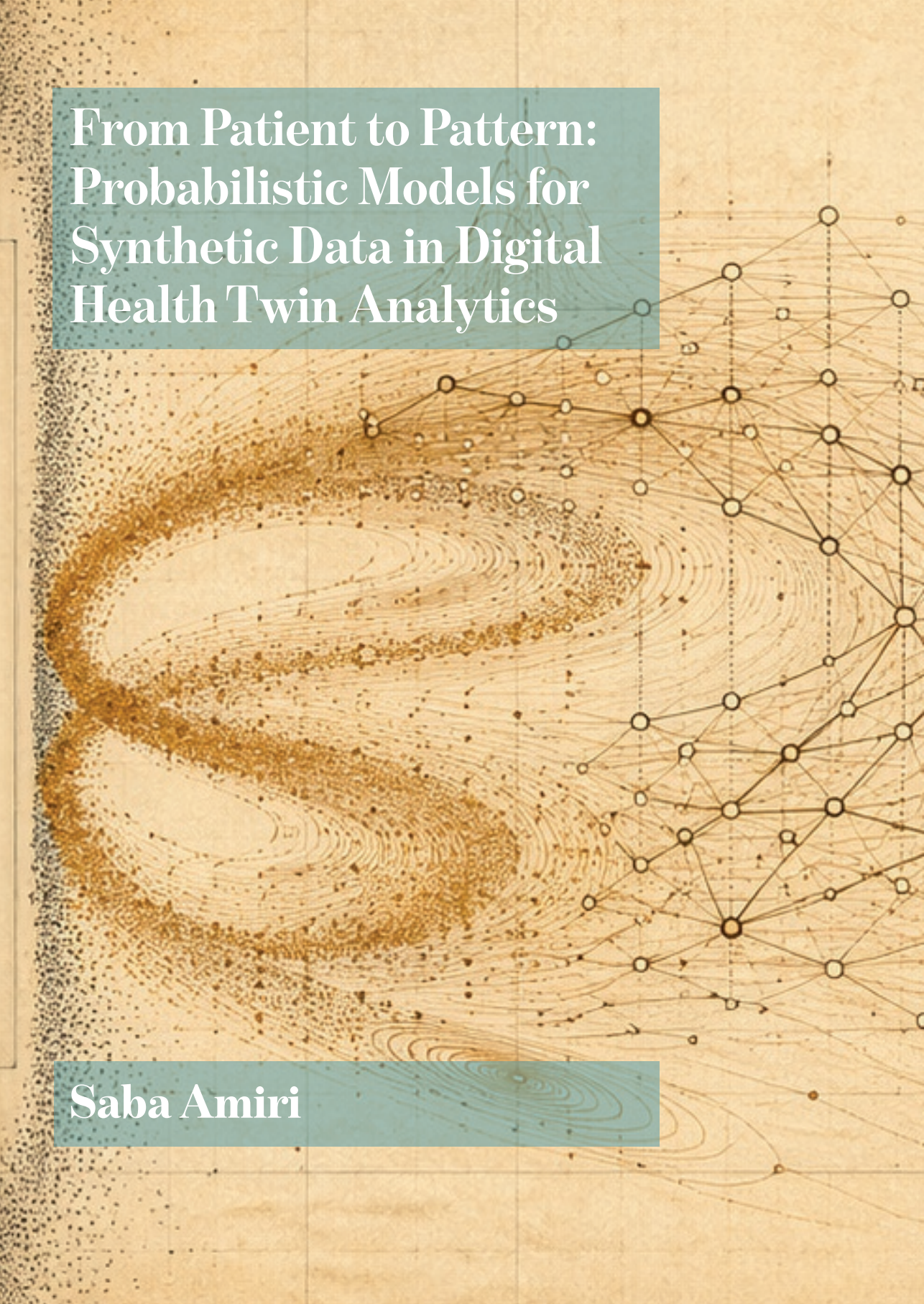
From Patient to Pattern: Probabilistic Models for Synthetic Data in Digital Health Twin Analytics



Saba Amiri

From Patient to Pattern: Probabilistic Models for Synthetic Data in Digital Health Twin Analytics

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op donderdag 25 juni 2026, te 16.00 uur

door Saba Amiri
geboren te Teheran

Promotiecommissie

Promotores:	prof. dr. S. Klous	Universiteit van Amsterdam
	dr. ing. L.H.M. Gommans	Universiteit van Amsterdam
Copromotores:	dr. A.S.Z. Belloum	Universiteit van Amsterdam
	dr. E.T. Nalisnick	Johns Hopkins University
Overige leden:	prof. dr. ir. C.T.A.M. de Laat	Universiteit van Amsterdam
	prof. dr. P. Grosso	Universiteit van Amsterdam
	prof. dr. S. Ghebreab	Universiteit van Amsterdam
	prof. dr. ir. A. van Halteren	Vrije Universiteit Amsterdam
	prof. dr. F.E. Scheepers	Universiteit Utrecht
	dr. L.M. Dijksman	St Antonius Ziekenhuis
	dr. A.M. Oprescu	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

From Patient to Pattern: Probabilistic Models for Synthetic Data in Digital Health Twin Analytics

Saba Amiri

Informatics Institute

University of Amsterdam



This research has been performed as part of the *Enabling Personalized Intervention* (EPI) project. The EPI project is funded by the Dutch Science Foundation in the Commit2Data program, grant number 628.011.028.

Summary

This thesis addresses fundamental challenges in the development and implementation of Digital Health Twin (DHT) frameworks for distributed medical data analytics. The increasing volume and complexity of healthcare data, combined with stringent privacy regulations and data governance policies, necessitate innovative approaches to data access and utilization. We propose and evaluate methodologies for generating high-quality synthetic data using advanced generative models as an alternative to direct access of sensitive patient information.

The Digital Health Twin concept offers a promising framework for personalized medicine, enabling virtual representations of individual health status that can be continuously updated with diverse data streams. However, realizing this potential is hampered by practical difficulties in accessing and processing real-world medical data across different administrative domains. While federated learning approaches partially address these challenges by bringing computation to the data rather than centralizing sensitive information, they still face significant hurdles related to permission acquisition, coordination of training iterations, and continuous data availability.

To overcome these limitations, we investigate the use of generative models to create synthetic data that statistically mirrors the properties of real-world patient data without containing actual information about individuals. This approach creates a privacy-preserving abstraction layer between sensitive data and the model training process. Unlike federated learning, which requires repeated access to real data throughout the iterative model development process, synthetic data generation requires only a single authorization phase, after which model development can proceed independently without further access to sensitive information. Additionally, synthetic data enables asynchronous model development across distributed nodes, eliminating the coordination challenges inherent in federated approaches where all participants must be simultaneously available. This fundamental shift in data access patterns offers significant practical advantages in healthcare settings where obtaining and maintaining data permissions is particularly complex and time-consuming.

Given the potential advantages of synthetic data for DHT implementation, our research focuses on enhancing the utility and privacy characteristics of generative models specifically for healthcare applications. Healthcare data presents unique challenges that standard generative approaches struggle to address adequately, potentially limiting the practical value of synthetic data in clinical contexts. Our work addresses four critical challenges to increase the suitability of synthetic data for DHT frameworks: (1) accurately modeling mixed-tail behavior in healthcare data, where variables within the same dataset exhibit fundamentally different distributional characteristics; (2) balancing fidelity and diversity in synthetic data generation to ensure both accurate representation of the original data and sufficient variation; (3) preserving privacy while maintaining data utility; and (4) handling data heterogeneity in geographically distributed environments.

For the mixed-tail modeling challenge, we develop tail-adaptive normalizing flows that can simultaneously accommodate both heavy-tailed and light-tailed distributions within a unified framework. This approach aims to overcome the optimization instability, flexibility limitations, and likelihood balancing issues common in standard generative models when applied to healthcare data with diverse distributional characteristics.

To address the fidelity-diversity trade-off, we propose a model-agnostic framework for unsupervised learning of semantic constraints in target datasets. This includes a validator component that guides generative models toward respecting both explicit and implicit boundaries in the real data, enhancing the quality of synthetic outputs while maintaining appropriate variation.

For privacy preservation, we introduce a noise-free mechanism for generating differentially private tabular synthetic data using normalizing flows. This method improves upon current noise-adding approaches, potentially offering stronger privacy guarantees while preserving important statistical properties of the original data.

Finally, to handle data heterogeneity in distributed environments, we first demonstrate the sensitivity of privacy-preserving federated learning setups to data distribution shifts and their adverse impacts on utility and fairness metrics. Building on these insights, we develop and evaluate a proof-of-concept implementation using real-world private

healthcare data that demonstrates how synthetic data can effectively replace real data in distributed analytics pipelines. We empirically evaluate this approach, comparing performance metrics between synthetic and real data scenarios to quantify the practical viability of synthetic data solutions in addressing challenges of data heterogeneity while maintaining both utility and fairness.

The thesis demonstrates the practical application of these methodologies through a proof-of-concept DHT framework optimized for clinical pathways. Our findings suggest that appropriately designed generative models can produce synthetic data with sufficient quality to support various downstream tasks while addressing privacy concerns and regulatory requirements. The proposed approaches may help unlock the potential of distributed healthcare data for personalized medicine and preventive care while respecting the fundamental principles of data privacy and sovereignty.

This research contributes to the broader fields of distributed data analytics, privacy-preserving machine learning, and healthcare informatics by offering novel technical solutions to long-standing challenges in data access and utilization. The methodologies developed may be applicable beyond healthcare to other domains where sensitive data and distributed processing present similar challenges.

Samenvatting

Dit proefschrift behandelt fundamentele uitdagingen bij de ontwikkeling en implementatie van Digital Health Twin (DHT)-raamwerken voor de gedistribueerde analyse van medische data. Het toenemende volume en de groeiende complexiteit van gezondheidszorgdata, in combinatie met strikte privacyregelgeving en beleid rond datagovernance, maken innovatieve benaderingen van datatoegang en datagebruik noodzakelijk. Wij presenteren en evalueren methodologieën voor het genereren van hoogwaardige synthetische data met behulp van geavanceerde generatieve modellen, als alternatief voor directe toegang tot gevoelige patiëntinformatie.

Het concept van de Digital Health Twin biedt een veelbelovend raamwerk voor gepersonaliseerde geneeskunde, doordat het virtuele representaties van de individuele gezondheidstoestand mogelijk maakt die continu kunnen worden bijgewerkt met uiteenlopende datastromen. De verwezenlijking van dit potentieel wordt echter bemoeilijkt door praktische problemen bij de toegang tot en verwerking van medische data uit de praktijk in verschillende administratieve domeinen. Hoewel federated learning-benaderingen deze uitdagingen gedeeltelijk ondervangen door de berekeningen naar de data te brengen in plaats van gevoelige informatie te centraliseren, kampen zij nog steeds met aanzienlijke obstakels op het gebied van het verkrijgen van toestemmingen, de coördinatie van trainingsiteraties en de continue beschikbaarheid van data.

Om deze beperkingen te ondervangen, onderzoeken wij het gebruik van generatieve modellen om synthetische data te creëren die de eigenschappen van patiëntdata uit de praktijk statistisch weerspiegelen, zonder feitelijke informatie over individuen te bevatten. Deze benadering creëert een privacybeschermende abstractielaag tussen gevoelige data en het modeltrainingsproces. In tegenstelling tot federated learning, dat gedurende het iteratieve modelontwikkelingsproces herhaaldelijk toegang tot echte data vereist, vergt het genereren van synthetische data slechts één enkele autorisatiefase. Daarna kan de modelontwikkeling zelfstandig doorgaan zonder verdere toegang tot gevoelige informatie. Daarnaast maakt synthetische data asynchrone modelontwikkeling tussen gedistribueerde knooppunten mogelijk, waardoor de coördinatieproblemen verdwijnen

die inherent zijn aan federated benaderingen waarbij alle deelnemers gelijktijdig beschikbaar moeten zijn. Deze fundamentele verschuiving in patronen van datatoegang biedt belangrijke praktische voordelen in zorgomgevingen waar het verkrijgen en behouden van datatoestemmingen bijzonder complex en tijdrovend is.

Gezien de potentiële voordelen van synthetische data voor de implementatie van DHT richt ons onderzoek zich op het verbeteren van de bruikbaarheids- en privacy-eigenschappen van generatieve modellen, specifiek voor toepassingen in de gezondheidszorg. Gezondheidszorgdata brengen unieke uitdagingen met zich mee die met standaard generatieve benaderingen moeilijk adequaat kunnen worden aangepakt, wat de praktische waarde van synthetische data in klinische contexten mogelijk beperkt. Ons werk behandelt vier kritieke uitdagingen om de geschiktheid van synthetische data voor DHT-raamwerken te vergroten: (1) het nauwkeurig modelleren van mixed-tail-gedrag in gezondheidszorgdata, waarbij variabelen binnen dezelfde dataset fundamenteel verschillende distributiekenmerken vertonen; (2) het in evenwicht brengen van fidelity en diversity bij het genereren van synthetische data, om zowel een accurate representatie van de oorspronkelijke data als voldoende variatie te waarborgen; (3) het waarborgen van privacy met behoud van databruikbaarheid; en (4) het omgaan met dataheterogeniteit in geografisch gedistribueerde omgevingen.

Voor de uitdaging van mixed-tail-modellering ontwikkelen wij tail-adaptive normalizing flows die zowel heavy-tailed als light-tailed verdelingen gelijktijdig binnen één uniform raamwerk kunnen accommoderen. Deze benadering is erop gericht de instabiliteit in de optimalisatie, beperkingen in flexibiliteit en problemen met likelihood-balancering te overwinnen die vaak voorkomen bij standaard generatieve modellen wanneer deze worden toegepast op gezondheidszorgdata met uiteenlopende distributiekenmerken.

Om de balans tussen fidelity en diversity te optimaliseren, presenteren wij een modelagnostisch raamwerk voor het ongesuperviseerd leren van semantische beperkingen in de betreffende datasets. Dit omvat een validatorcomponent die generatieve modellen zo stuurt dat zij zowel expliciete als impliciete grenzen in de echte data respecteren, waardoor de kwaliteit van synthetische uitkomsten wordt verbeterd met behoud van passende variatie.

Voor privacybescherming introduceren wij een ruisvrij mechanisme voor het genereren van differentially private tabulaire synthetische data met behulp van normalizing flows. Deze methode vormt een verbetering ten opzichte van huidige benaderingen waarin ruis wordt toegevoegd en biedt mogelijk sterkere privacygaranties, terwijl belangrijke statistische eigenschappen van de oorspronkelijke data behouden blijven.

Ten slotte tonen wij, om dataheterogeniteit in gedistribueerde omgevingen te kunnen hanteren, eerst de gevoeligheid aan van privacybeschermende federated learning-opstellingen voor verschuivingen in datadistributies en hun nadelige effecten op bruikbaarheids- en fairness-metrieken. Voortbouwend op deze inzichten ontwikkelen en evalueren wij een proof-of-concept-implementatie met private gezondheidszorgdata uit de praktijk, die laat zien hoe synthetische data echte data effectief kunnen vervangen in gedistribueerde analysepijplijnen. Wij evalueren deze benadering empirisch door prestatiemetrieken tussen scenarios met synthetische en echte data te vergelijken, om zo de praktische haalbaarheid van oplossingen op basis van synthetische data te kwantificeren bij het aanpakken van uitdagingen rond dataheterogeniteit, met behoud van zowel bruikbaarheid als fairness.

Het proefschrift demonstreert de praktische toepassing van deze methodologieën aan de hand van een proof-of-concept DHT-raamwerk dat is geoptimaliseerd voor klinische zorgpaden. Onze bevindingen suggereren dat zorgvuldig ontworpen generatieve modellen synthetische data van voldoende kwaliteit kunnen produceren om diverse downstreamtaken te ondersteunen, waarbij tegelijkertijd tegemoet wordt gekomen aan privacyzorgen en regelgevende vereisten. De voorgestelde benaderingen kunnen helpen het potentieel van gedistribueerde gezondheidszorgdata voor gepersonaliseerde geneeskunde en preventieve zorg te ontsluiten, met respect voor de fundamentele principes van dataprivacy en datasoevereiniteit.

Dit onderzoek levert een bijdrage aan de bredere vakgebieden van gedistribueerde dataanalyse, privacybehoudend machine learning en gezondheidsinformatica door nieuwe technische oplossingen te bieden voor al lang bestaande uitdagingen op het gebied van datatoegang en datagebruik. De ontwikkelde methodologieën kunnen mogelijk ook worden toegepast buiten de gezondheidszorg, in andere domeinen waar gevoelige data

en gedistribueerde verwerking vergelijkbare uitdagingen met zich meebrengen.

Acknowledgments

A PhD is a deeply personal journey. You give years of your life to a piece of work that, for all its public formality, has a very private meaning. It asks for discipline, stubbornness, and more faith than one usually admits. Just as important, it asks for people. My PhD began right around the time the pandemic arrived and took the world apart. For most of the time that followed, home was the office, the library, the cafe, and the corridor conversation, all of it collapsed into a single set of rooms. That could have been isolating in ways that broke the work and the person. Instead, the people in my life found ways to be present in the hours when I needed them. I would like to acknowledge them here.

To Adam: in many ways this started before the PhD, when I was doing my Master's thesis with you. Your guidance continued from there, and so did your friendship. You were calm when I was not, generous when I doubted myself, and sharp when the work needed it. Your questions found exactly the place where my argument was still soft. You had my back throughout. I, in turn, frustrated you by trying to do many things at once while being obsessive about every single one of them as one should frustrate one's supervisor. We still work together, still on multiple things, still all at once, and I still look forward to every conversation.

To Eric: you came into my supervision somewhere in the middle, and the impact was profound. You gave structure to how I approach being a scientist and sharpened my skills. I would arrive in our weekly meetings with something that felt promising but wasn't quite legible yet, and by the end of the conversation it had a shape I could work with. And above all, knowing you as a friend as well as a supervisor made those years richer. To this day, when I run into a problem that genuinely resists me, one of my favorite mental exercises is to channel you—wearing a robe in the swamps of Dagobah, mostly because the weather resembles Amsterdam weather with suspicious accuracy—and ask myself: how would Eric approach this? The answer always points somewhere good.

To Sander: you were, far beyond being my promptor, a mentor to me. You pushed

me to think about impact and implications of my work, and about the world that the research was supposed to connect to. You had a way of holding the small picture and the large one at the same time, and making sure I could see both. The fact that, after the PhD, you have remained a mentor to me says something important about the seriousness and generosity with which you invest in people.

To Leon: during the course of my PhD, I came to trust your guidance completely. You were present, engaged, and unfailingly helpful. Every meeting left me with something of value to think about—a reframing, a question, a small shift in how I was seeing the problem. That kind of effortlessness is rarer than it appears, and I thank you for it.

To my partner, Jeyran: you listened, for years, to monologues about differential privacy and flow-based models that were long, not always coherent, and — I suspect — not always welcome. Through no fault of your own, your chemistry degree now carries an uninvited minor in machine learning. You knew, with a clarity I rarely had myself, when to pull me away from the desk and back into the world. You planned our life during the years when I could not see past the next deadline to plan anything at all, and you made the uncertain stretches feel livable, which is harder than it sounds and more important than I knew how to say. Your belief in me never wavered, even when mine did. This thesis is yours too[1].

To Ana, Cees, Chrysa, Florian, Giovanni, Paola, Thomas , Tom, Yuri, Zhiming, and the rest of the greater MNS, CCI, and PCS members: thank you for building and maintaining an environment where good research felt possible and where the people doing the research felt seen. The institutional conditions for thinking well are made, carefully, by people who care. You are those people, and it mattered.

To Corinne, Jamila, Milen, Rosanne, and Tim: working with you as part of the EPI project was one of the more quietly enjoyable parts of these years. Collaborating with people who are genuinely good at what they do is a real pleasure. Corinne, writing a review paper is the kind of task that sounds manageable until you are inside it, at which point it reveals itself to be a long and frequently humbling exercise in sustained

[1]This thesis also goes to Carmen the cat, who is an arson and a criminal, and whose commitment to waking me at 6:15am sharp arguably contributed more to this thesis than any single methodological choice. May she sleep on it well and dream of electric sheep.

effort. Doing it with you made it not only bearable but something I look back on with real satisfaction.

To Carlijn, Christopher, Cyril, Dolly, Floris-Jan, Giulio, Hongyun, Joseph, Julius, Leonardo, Lu-Chi, Marco, Marius, Misha, Na, Peter, Pooya, Ralph, Reggie, Ruye, Sara, Siamak, Spiros, Xin, Yuandou, Zeshun, and the rest of my friends and colleagues at IvI: our paths crossed more in passing than in close collaboration, and the pandemic made that more limited than it should have been. But I was still glad to be part of the same place and among the same people.

To Anahita, Amin, Dani, Emile, HamidReza, Hemen, Marzieh, Marzieh, Masoud, Mozhdeh, Nazi, Saeede, Saeid, Sahar, Shahriar, Vahideh, and the rest of my friends: being far from home is easier when you find people with whom very little needs to be explained. Your friendship and the ordinary fact of having you in my life made these years much better than they otherwise would have been.

To my family, Behnaz, Farhad, and Sara, and my extended family—Azar, Behrooz, Mina, and Majid: I saved you for last simply because thanking you properly is hard to put into words. Each of you supported me in your own specific way over these years. Thank you for always being there, and for making this possible. I am deeply grateful.

To my wonderful paranymphs, Daphnee and Andrea: thank you for agreeing to stand beside me at the defense, which is either an act of friendship or of poor judgment and academic fatigue, and I prefer not to investigate which. I'm glad it's you two.

I sincerely thank the members of my committee for the generosity of their time in reading this thesis and considering its arguments.

The Enabling Personalized Interventions consortium provided the broader setting in which this thesis developed. I benefited greatly from being part of that collective effort, and I appreciate all who contributed to it.

CONTENTS

1 Introduction **1**

1.1 Digital Health Twin Framework for Distributed Medical Data Analytics 4

1.2 The Challenge of Data Access in DHT 7

1.3 DHT Data Access Abstraction Using Synthetic Data 10

1.4 Research Questions 15

1.5 Thesis overview 20

1.6 Contributions 21

2 Capturing Tail Behavior in Synthetic Data Generation **25**

2.1 MOTIVATION 25

2.2 Formalizing Tail Behavior Characteristics 29

2.3 LIMITATIONS OF EXISTING APPROACHES FOR MIXED-TAILED MODELLING 31

2.4 METHOD 33

2.5 EXPERIMENTS 39

2.6 SELECTIVE GENERATION BY EXPLORING LATENT SPACE . . 49

2.7 Summary 52

3 Generating Synthetic Tabular Data with Improved Semantic Integrity 53

3.1 Motivation 53

3.2 The Challenge of Semantic Integrity in Generative Models 56

3.3 Latent Semantic Regularization Framework 60

3.4 Previous Works 67

3.5 Experiments 69

3.6 Summary 82

4 Constrained-Sensitivity Differentially Private Normalizing Flows for Tabular Data Synthesis 83

4.1 Method 86

4.2 Experiments 97

4.3 Summary 101

5 Decentralized Synthetic Data Generation for Digital Health Twin 103

5.1 Motivation 103

5.2 Conceptual Workflow to Optimize Clinical Pathways 106

5.3 Use-Case Description 108

5.4 PoC Framework, Workflows and Policies 109

5.5 Federated Analysis and Learning 110

5.6 Why federated ML is not always the right answer? 112

5.7 Federated Synthetic Data Generation 125

5.8 Summary 129

6 Discussion and Conclusion **131**

6.1 **RQ1**: How can generative models estimate densities exhibiting heterogeneous tail behaviors? 132

6.2 **RQ2**: How can generative models synthesize data with high fidelity and diversity while respecting the implicit semantic boundaries of the real data? 134

6.3 The Big Picture 143

Appendix A **147**

A.1 Introduction 147

A.2 survey protocol 149

A.3 PSDDL Taxonomy 151

A.4 Collaborative Learning 158

A.5 Distributed Learning 170

A.6 Discussion 185

A.7 Conclusion 193

Appendix B **199**

B.1 Basic Concepts 199

B.2 Generative Models 205

LIST OF FIGURES

2-1 The impact of mis-specification of base density on the capabilities of normalizing flows to capture the tail behavior of the target density. We train an RNVP on three short, normal and heavy-tailed *Generalized Gaussian (GGD)* targets (columns 2-4) with short, normal and heavy-tailed GGD base densities (rows 1-3). We plot the probability density of the plot. Mis-specified parameterization of base leads to flow being only able to estimate the target with the same tail behavior as its base density accurately 28

2-2 Mis-estimation of tail indices by bootstrapped Hill method for datasets sampled from a Pareto distribution. Tail indices vary from heavy (α=0.5) to short (α=10.5) and sample sizes from 64 to 2048. Each experiment is conducted 9 times and the mean - solid lines - and standard error - shades - of the difference between the true tail index and the estimated one are reported. 33

2-3 Probability density functions (left) and quantile functions (right) of generalized Gaussian distributions with shape parameter $\beta = 1.0, 2.0$, and 10.0. $\beta = 2.0$ yields a Gaussian. As β drops below 2.0, the PDF becomes heavy-tailed and the quantile function transitions from linear to concave down and vice versa. 35

2-4 P-values of the two-sample Kolmogorov-Smirnov test on mixtures of generalized Gaussians with different number of components trained to estimate target distributions samples from a Student's t with different degrees of freedom. 36

2-5 The impact of heavy-tailed target and base on the maximum likelihood estimation. This graph shows the mean and geometric median of loss and gradient norms a Real NVP flow trained with Student's t base density on a Student's t target, with and without mis-specification of the base with respect to the target. 38

2-6 Performance of RNVP flow with different base distributions for estimation of a mixed-tail target. Our modified Neal's funnel with $\lambda = 6.0$ has a heavier-tailed and a lighter-tailed variable. 43

2-8 Ablation study framework. The two graphs on the left depict the probability density functions (pdf) of our dual random variables at the extreme points of γ. The pair of graphs on the right illustrate how the shape parameters ν for the Student's t-distribution and β for the GGD vary as functions of γ. 44

2-9 Ablation study results. Two different base distributions are plotted on different rows. Variables $\mathbb{X}_1$ and $\mathbb{X}_2$ are plotted on different columns. The light gray lines show the results of models without robust gradient estimation and the dark gray lines show the results with robust gradient estimation. 45

2-10 Selective generation from an RNVP model with a 20 component GGD mixture base for a 2d Student's t target. 51

3-1 **The Explore/Exploit Dilemma in Semantically Constrained Synthesis.** ***Left***: *Green*: High exploitation (strong semantic integrity) concentrates mass, missing tails and diversity. *Blue*: High exploration (diversity) may venture into semantically invalid regions (hallucinations). ***Right***: Example empirical posterior density of generated samples mapped into the validator's latent space, showing how LSR might structure this space. 58

3-2 Comparative visualization of the validator's latent space for semantically incorrect samples (red dots) from the 2D experiment. **Left:** Validator with a standard Gaussian prior shows poor separation. **Right:** Validator with a Tail-Adaptive NF prior effectively clusters incorrect samples in low-density regions, enabling stronger semantic regularization. 77

3-3 Distribution of synthetic variables (left column: $V1$-$V2$; right column: $V3$-$V4$) within subgroups for the augmented *Child* dataset. Black bars represent the ground-truth frequencies. LSR (S+V) generally shows good fidelity, while some baselines (e.g., MedGAN on $V2$) exhibit significant distortions 79

5-1 Conceptual workflow of an optimized clinical pathway 106

5-2 Distribution of target feature and the protected class among 15 FL participants. Left column shows the IID data with $\alpha = 100$, middle column shows mild non-IID data with $\alpha = 100$ and right column shows extreme non-IID data with $\alpha = 100$ 115

5-3 Trade-offs between privacy, distribution, and utility. The upper subfigure shows the privacy-utility trade-off, and the lower subfigure shows the privacy-distribution trade-off. 118

5-4 F1-score of FL models with different privacy levels and distributions for 100 epochs. 119

5-5 Privacy trade-off against fairness 120

5-6 Distribution trade-off against fairness 120

5-7 Distribution trade-off against fairness. EOR is measured for each underprivileged group against the designated privileged group and for the desirable outcome of *Annual Salary > 50k* 123

5-8 Privacy trade-off against fairness. EOR is measured for each underprivileged group against the designated privileged group and for the desirable outcome of *Annual Salary > 50k* 124

5-9 Optimized pathways for DHT through synthetic data generation. We first train a differentially private generative model in a federated fashion. Then, instead of performing the analysis on the ECT datasets directly in a federated manner, we generate a synthetic dataset and perform our analysis on the synthetic data. We also perform our analysis directly on the real data and compare the results. 125

5-10 Comparative Performance Metrics of Classifiers Trained on Real and Synthetic Datasets. This bar chart illustrates the performance of classifiers trained on synthetic data and real data (comprising joint datasets from St. Antonius and UMCU). Performance is measured via two metrics: accuracy and F1 score. The classifiers trained on synthetic data demonstrate superior performance. 129

A-1 Two major architectures for distributed learning. In FL participants are actively involved in training the model while in collaborative learning participants contribute by sharing their data or results for DL training and inference on their data in a private and secure manner by an orchestrator 152

A-2 Left: CBL workflow. It shows at which point secure inference, secure training, and private server training are applied. Right: Distribution of papers among the different types of protective mechanisms for the different points in the workflow. 159

A-3 A FL workflow demonstrating where privacy-preserving measures protect model inference and secure communication safeguards the training process (left). The distribution of research papers investigating these protective mechanisms is shown on the right. 172

LIST OF TABLES

2.1 General performance metrics for the Neal's funnel for the average NLL as well as AULLP difference and tVaR difference for each variable separately. 43

2.2 Comparison of NLL, AULLP difference and tVaR difference for our proposed method applied to a RNVP in comparison with the same model with different base distributions and CTGAN. *The tVaR tends to grow exponentially when the tail is heavily underestimated 45

2.3 Comparison with [1]. We run the same experiment as [1]'s Table 1 experiment for $d_h = 4$. We repeat our experiment 25 times and report the average results and standard error. 46

2.4 Comparison of NLL, AULLP difference and tVaR difference for our proposed method applied to a MAF in comparison with the same model with different base distributions. 48

3.1 Percentage of incorrectly labelled samples on the 2D Gaussian mixture dataset. Lower values indicate better adherence to the implicit semantic rule defined by cluster assignments. 76

3.2 Statistical Fidelity (Numerical Features): Average KS test statistic across all numerical features. Lower is better. 78

3.3 Statistical Fidelity (Categorical Features): Average Log Chi-Squared test statistic across all categorical features. Lower is better. 78

3.4 Downstream Task Efficacy (F1-Score): Performance of a classifier trained on synthetic data vs. real data. Higher is better. 79

3.5 Exploration Factor (S_d): Cumulative adjusted Hamming distance to nearest neighbors in the real dataset. Higher values indicate greater generative diversity. 80

3.6 Semantic Integrity Violations on Augmented *Child*: Percentage of generated samples violating zero-probability constraints. Absolute percentages are reported against ground truth. According to the qualitative analysis, CTGAN is the only contender with our method since other methods have lower violation at the expense of suppressing minority subgroups. 81

4.1 Average test log-likelihood. Higher is better. Our DP release uses ε=4.0 and δ=0. §: values reported in Waites and Cummings [2]. †: our re-run of their code. 98

5.1 Distribution of protected feature *Race* categories over the target class *Annual Salary* 115

A.1 Comparison of surveys within the field of PSDDL 151

A.2 Overview of Secure Inference methods. The papers are sorted by year of publication, and by alphabetical order of the author. Collusion: whether participants can collude with the server. Verification: whether the paper addresses verifiability of the data Abbreviations: AHE= additional homomorphic encryption, BFV=Brakerski-Fan-Vercauteren scheme, CKKS= Cheon-Kim-Kim-Song scheme, CNN= convolutional neural network, DNN= deep neural network FHE= Full Homomorphic Encryption, GC= Garbled Circuits, GMW= Goldreich-Micali-Wigderson scheme, HBC = honest but curious, LSTM= Long short-term memory, LHE= Linearly Homomorphic Encryption, MA= malicious adversary, MLP= multi-layer perceptron, NC= No collusion, ND= not determined, OT= Oblivious transfer, SS= secret sharing 195

A.3 Summary of secure training papers. The papers are sorted by year of publication, and by alphabetical order of the author. Efficiency: Speed up compared to benchmark, complexity, or seconds, in most ideal scenario. Collusion and Drop Out: max. number of participants. Fairness: whether the framework offers at least fairness. Abbreviations: FSS = Function secret sharing GMW= Goldreich-Micali-Wigderson, LHE = Linear Homomorphic Encryption, LSTM= Long short-term memory, MA= malicious adversary, ND= not determined 196

A.4 Overview of private distributed learning methods. The papers are sorted by year of publication, and by alphabetical order of the author. Privacy level: reviewed works rarely share the same settings for different variables in these types of methods, e.g. model architecture, number of clients, rounds of training, etc. The reported values here are the most significant and/or relevant results from the most common dataset used, usually MNIST if available. The privacy levels should not be a basis of comparison between methods and each method should be evaluated individually Abbreviations: CNN= Convolutional neural network, MA= Moments accountant, FL= Federated learning, ... 197

A.5 Encryption-based protective measures for FL, sorted by year and author. Efficiency in speedup compared to benchmark, complexity or seconds. Collusion and Drop Out quantified in max. number of participants Abbreviations: DM = double masking, ND= not determined, SSP = Shakespeare, PT= plaintext * SMA = semi malicious ** with irregular users *** in (t-1), t refers to t-out-of-n SS **** papers show improvement compared to a similar Paillier based protocol, but do not compare to a benchmark 198

CHAPTER
1

INTRODUCTION

In the age of geographically separated and private big data, a key approach to utilizing the trove of data at our disposal and unlock its true potential is distributed data analytics and machine learning. This distributed framework can be realized in different conceptual contexts. The Digital Health Twin (DHT) is such a concept for processing of medical data. In this chapter we define the DHT framework, introduce the main challenge to its realization, namely data access and usage, and position generative model-based data synthesizers as a solution to this problem. We define our research questions based on the main challenges in replacing real data with synthetic data in a digital health twin framework.

The 21st century is characterized by an explosion of "big data" vast, complex datasets that permeate every aspect of our digital lives. This phenomenon is fundamentally reshaping computer science, driving a paradigm shift in data capture, storage, management, and analysis. The significance of this revolution lies not only in data volume, but also in its velocity and variety [3].

In light of these developments, the necessity for distributed data management has become more pronounced. Distributed systems enable the segmentation of data across

different administrative domains, each with their own heterogeneous resources, facilitating parallel processing that enhances efficiency in handling voluminous data [4, 5]. The rationale against the accumulation and centralization of the data extends beyond mere technical necessities such as substantial communication time and costs associated with the transfer of large datasets, stringent privacy regulations and corporate and public policy considerations that often prohibit the centralization of sensitive information: In the age of big data, domain specific data is organically generated in a distributed manner; Since the production of the data is distributed and although belonging to the same domain, different datasets fall under different jurisdictions with heterogeneous infrastructure, administration and regulation, it is against its nature to be centralized. The same logic applies for the consumption of the data products, e.g. data analysis: The developments and advancements in methods and systems for processing, presentation and visualization of the data as well as the democratization of compute access in forms of cheap and powerful commodity hardware and cloud computing has spurred on local consumption of the data - as opposed to being reliant on centralized authority to perform the consumption and provide the data products to its subsidiaries. Therefore, the inherent structure of big data generation and consumption naturally lends itself to a decentralized paradigm.

The large size of data sets imposes significant communication overhead when moved across networks, leading to latency and bottleneck issues [6]. This is problem is increasingly evident in various mission critical sectors, including finance, healthcare, and security [7]. Distributed data storage and processing mitigate these issues by allowing data to remain in proximity to its source, reducing the need for extensive data transfer and thereby enhancing system responsiveness [8].

Furthermore, privacy concerns and policy regulations, such as the General Data Protection Regulation (GDPR) in the European Union, impose legal constraints on data management practices [9]. These regulations often mandate that personal data be stored and processed within specific geographic boundaries, necessitating a distributed approach to data management. Such constraints also encourage the development of data processing techniques that can efficiently operate within the confines of privacy-preserving frameworks [10].

To address these challenges of distributed analytics, the principle of bringing computation to the data, rather than data to the computation, has gained prominence. This is particularly pertinent given the aforementioned communication costs and latency issues. By employing distributed computing frameworks, such as Federated Learning [11], organizations can analyze large volumes of data in situ, effectively reducing the need to move data over the network and thereby mitigating associated delays and costs [12]. Distributed computing frameworks, such as federated learning, also give organizations flexibility to implement additional policies on their data processing pipeline, one of the most pertinent ones being privacy [13].

Distributed machine learning approaches allow model training over decentralized datasets. Techniques such as federated learning go a step further by enabling the training of models across multiple devices or servers holding local data samples, without exchanging them [14]. Federated learning not only maintains privacy to some extent but also allows for the incorporation of a wide array of data characteristics and sources, potentially leading to more comprehensive and generalizable models [15, 16, 17].

The convergence of distributed data analytics and machine learning fosters collaborative and transparent data ecosystems. Distributing these processes allows stakeholders from different domains to contribute to a collective intelligence. This approach not only ensures regulatory compliance and data sovereignty but also enables fine-grained cross-domain management and policy enforcement, allowing stakeholders to derive greater value from the shared data.

In this work we focus on the healthcare domain. The necessity for a sophisticated framework for distributed healthcare data analytics and machine learning is underscored by the burgeoning volume and complexity of health-related data coupled with the necessity for tailored medical interventions [18]. The integration of vast arrays of decentralized data sourcesincluding electronic health records, omics data, and real-time biometric monitoringrequires a robust architecture that can effectively manage and analyze this information while adhering to stringent privacy and security regulations [19, 20]. Such a framework is essential to unlock the potential of machine learning algorithms, which thrive on large, diverse datasets to identify patterns and make accurate

predictions within a reasonable timeframe. It enables the synthesis of multi-dimensional data streams, providing a comprehensive picture of individual health states and informing precision medicine approaches. This is critical for advancing patient outcomes, streamlining healthcare delivery, and catalyzing the shift from reactive to proactive and preventive care, thereby underscoring the foundational role of an advanced data analytics framework in modern healthcare innovation. The Digital Health Twin framework, which we will define in the next section, provides an ideal setting to address these challenges and realize this potential.

1.1 Digital Health Twin Framework for Distributed Medical Data Analytics

A Digital Health Twin (DHT) is a virtual representation of an individual's health status, which is dynamically updated with data to reflect any changes in that individual's physical well-being [21]. The foundation of a DHT lies in the Digital Twin (DT) concept [22]. A digital twin is a digital replica of a physical entity, process, or system that can be used for various purposes such as simulation, analysis, and control. In the healthcare context, this entity is the human body, or more specifically, the health status of an individual.

Essentially, DHT acts as a personalized, predictive model of an individual's health, continuously updated with diverse data streams. To establish a rigorous foundation for our research, we now formally define the DHT network as a set of n participating nodes, denoted by $\mathcal{N} = \{N_1, N_2, ..., N_n\}$. Each node N_i possesses a local dataset D_i. We assume that each D_i contains realizations of a random vector X representing the features of a data point, drawn from a local data distribution $P_i(X)$. In the context of healthcare, X could encapsulate a patient's demographics, medical history, vital signs, lab results, genomic data, etc. These local distributions $P_i(X)$ may differ, reflecting the heterogeneity of data sources: $P_i(X) \neq P_j(X)$ for some $i \neq j$. This constitutes the non-IID (Independent and Identically Distributed) characteristic often encountered in real-world DHT scenarios. It is important to highlight that although a centralized

system is theoretically possible, it is practically infeasible due to the aforementioned challenges relating to privacy, data security, and regulatory compliance.

A digital health twin integrates various data streams, including genomics data, biometric data, medical records, etc. Using computational models, the digital health twin can simulate various physiological and pathological processes of the body. These models are based on medical knowledge and can be personalized with the individuals data and **enable personalized intervention**. This allows for the assessment of risk for certain diseases, prediction of health trajectories, and the evaluation of potential interventions.

The applications of digital health twins are numerous, e.g. personalized medicine, predictive analytics, telemedicine, clinical trials and personal health management [23, 24]. For example, in clinical trials, DHTs can be used to simulate patient responses to different treatments, potentially reducing the time and cost associated with traditional trials. In telemedicine, DHTs can enable remote monitoring and personalized interventions, improving access to care for patients in remote or underserved areas. Furthermore, in personal health management, DHTs can empower individuals to take a more active role in managing their health by providing them with personalized insights and recommendations. For the purpose of this thesis, to better define our approach and without loss of generality, we focus on the task of training a machine learning model on these disparate datasets.

Formally, within this DHT framework, a common downstream task is to learn a function $f : \mathcal{X} \to \mathcal{Y}$. This function maps input features x from the feature space $\mathcal{X}$ (e.g., $\mathcal{X}$ could be a high-dimensional space combining clinical measurements, imaging features, and -omics data) to target variables y in the target space $\mathcal{Y}$ (e.g., $\mathcal{Y}$ could represent diagnostic classifications, future risk scores, or predicted response to a specific treatment). The function f is typically parameterized by θ (e.g., the weights of a neural network) and is learned by minimizing a suitable loss function $\mathcal{L}(f_\theta(x), y)$ over the conceptually aggregated dataset $\mathcal{D} = \cup_{i \in \mathcal{N}} D_i$. This optimization is posed as:

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \sum_{i \in \mathcal{N}} \sum_{(x,y) \in D_i} \mathcal{L}(f_\theta(x), y)$$

where θ^* represents the optimal parameters of the model. This optimization is typically performed using iterative algorithms like stochastic gradient descent, adapted to the distributed setting.

Digital health twins offer a scalable approach to data analytics and machine learning. As they are virtual representations, they can be easily replicated across different healthcare settings and scaled up or down according to the computational resources available, ensuring efficient use of resources. The DHT framework is designed to integrate data from multiple sources seamlessly. It can handle diverse data types, including electronic health records (EHR), genomics data, and data from wearable devices, ensuring high-quality, comprehensive health profiles. The DHT framework allows for complex machine learning models to be used and integrated within the health twin, leveraging the distributed nature of the data. These models can be updated continuously as new data becomes available, providing up-to-date insights into patient health [25].

By employing such a distributed data processing and analytics approach, the digital health twin framework ensures:

- Privacy preservation: Sensitive health data remains on-premises, mitigating privacy concerns.
- Regulatory compliance: The model adheres to jurisdiction-specific data governance standards.
- Scalability: New health twins (scaling up) and more complicated health twins (scaling up) is possible without significant research and deployment effort.

The construction of digital health twins for data analytics generally is done through data-driven statistical models [25]. These models are primarily predicated on the analysis of extensive datasets. They utilize statistical methods to discern patterns and relationships within the data, which can then be used to make predictions or decisions without necessarily understanding the underlying mechanisms. In the context of Digital Health Twins, these models underpin the computational framework that enables the prediction of health outcomes based on historical and real-time health data.

1.2 The Challenge of Data Access in DHT

While the digital health twin is a robust framework and offers a dynamic, virtual representation of an individual's health status, informed by a confluence of real-time data and historical health records. Despite its potential to personalize and enhance healthcare delivery, the implementation of the digital health twin framework faces considerable challenges and difficulties.

One of the primary challenges in realization of DHT lies in difficulties related to real world data access. The ethical implications of using patient data for machine learning necessitate rigorous scrutiny by ethics committees, ensuring that the rights and privacy of individuals are safeguarded. Moreover, the digital health twin framework must operate within a mosaic of policies and regulations that vary across different jurisdictions. Adhering to these diverse policies while accessing and processing data requires a nuanced understanding of legal frameworks such as the General Data Protection Regulation (GDPR) [9] and the Health Insurance Portability and Accountability Act (HIPAA) [26].

Furthermore, the development and tuning of privacy-preserving machine learning models, such as those implementing differential privacy, introduce a set of technical challenges. Differential privacy provides formal guarantees about the maximum amount of information that can be learned about an individual from the output of a computation. These models must be carefully calibrated to maintain the utility of the data while sufficiently obfuscating individual contributions to protect privacy. The trade-off between privacy and model performance is a non-trivial challenge that requires innovative solutions to ensure that the benefits of the DHT framework can be fully realized without compromising individual privacy.

Distributed and Federated Learning for Heterogeneous, Decentralized Data Processing in DHT An established and widely-used approach towards solving the problems associated with data access and accumulation in DHTs involves decentralized processing of the disparate private patient data through distributed computing. Fed-

erated learning (FL) typically entails training machine learning models across multiple decentralized nodes (e.g., hospitals, research centers) that hold local data, without requiring the exchange of this data. This preserves privacy and complies with data governance policies. In the context of DHTs, each node could represent a hospital, clinic, or research institution, each holding valuable patient data. FL enables collaborative model training without compromising the privacy of sensitive patient information at each node. The adaptation of generative models to federate settings, particularly using synthetic data, can further enhance privacy protections and data utility across distributed environments.

The objective of distributed data processing in DHTs is to perform computations (such as training a model or performing statistical analysis) on the aggregated data $\mathcal{D} = \cup_{i \in \mathcal{N}} D_i$, subject to limitations such as respecting data governance policies and privacy constraints.

Federated learning (FL) addresses these limitations by bringing the computation to the data. In FL, each node N_i trains a local machine learning model, represented by parameters θ_i, on its local dataset D_i. These local model parameters are then aggregated (potentially after multiple local training steps) to update a global model, represented by parameters θ_{global}. This process typically iterates and can be formalized as follows (using M to denote the model state, primarily its parameters θ):

1. **Local Training:** Each participating node N_i updates its local model parameters θ_i by minimizing a local loss function $L_i(f_{\theta_i}, D_i)$ on its data D_i. Let M_i represent the updated local model state (e.g., the parameters θ_i or the parameter updates $\Delta\theta_i$).

2. **Aggregation:** The global model parameters θ_{global} (represented by state M_{global}) are updated using an aggregation function A, combining the states from the participating local models: $M_{\text{global}} \leftarrow A(\{M_i\})$. A common example is Federated Averaging (FedAvg) [27], where the global parameters are updated by averaging the parameters of the local models: $\theta_{\text{global}} \leftarrow \sum_i \frac{|D_i|}{|\mathcal{D}|}\theta_i$.

This iterative process aims to converge to a global model M_{global} (with parameters

θ_{global}) that performs well across all data.

The Challenges Federated Learning Does Not Directly Address While federated learning presents a significant advancement in addressing data access challenges by obviating the need for central data aggregation, several practical and logistical hurdles remain. Securing the necessary permissions from all participating entities to utilize their private data for distributed training can be a protracted and complex process, often requiring extensive legal and ethical reviews. Furthermore, the requirement for simultaneous data availability from all parties for each training iteration poses significant coordination challenges. The iterative nature of model development, which frequently necessitates multiple training cycles with intervening adjustments based on performance evaluation, further exacerbates these issues. Each iteration may require renewed ethical approvals and data access permissions, adding to the administrative burden. Moreover, ensuring continuous data availability throughout the potentially extended duration of the model development process presents a significant logistical obstacle.

Subsequent model refinement and evaluation often necessitate expert review of the training results, which may involve access to aggregated model outputs or performance metrics derived from the distributed training process. This introduces further privacy considerations, potentially requiring additional ethical oversight and consent protocols. The dynamic nature of health data also necessitates model adaptation to evolving data distributions and the incorporation of new information. Retraining or updating the model in a federated setting requires renewed access to the underlying data, potentially triggering further rounds of permissions and approvals. Therefore, while FL offers a valuable approach to distributed data processing, it does not entirely circumvent the fundamental challenges associated with data access in the context of DHT development and deployment. It shifts the complexities of data access from the initial aggregation stage to the distributed training and iterative refinement phases, necessitating robust governance and coordination mechanisms. This motivates the exploration of alternative approaches, such as the use of synthetic data, to realize the full potential of DHTs while respecting privacy and data governance requirements.

1.3 DHT Data Access Abstraction Using Synthetic Data

The realization of the Digital Health Twin (DHT) concept, as introduced in Section 1.1, hinges on the ability to access and analyze diverse, representative health-related data. Ideally, this would involve pooling data from various sources to create a comprehensive, multi-faceted view of an individual's health. However, the path to acquiring such data is fraught with obstacles. Significant practical and ethical limitations exist, including stringent privacy regulations, the logistical complexities of data sharing agreements between institutions, and the fundamental difficulty of obtaining access to large-scale, high-quality health datasets that are often siloed within different organizations. To overcome these limitations and unlock the full potential of DHTs, we propose a novel approach: leveraging generative models to create *synthetic data*. This synthetic data will serve as a privacy-preserving and readily available proxy for real-world health data, enabling the construction, validation, and iterative refinement of DHTs without the need for direct access to sensitive patient information.

Synthetic data, in this context, refers to artificially generated data that statistically mirrors the properties of real-world patient data without containing any actual information about real individuals. Let $\mathcal{D} = \cup_{i \in \mathcal{N}} D_i$ denote the aggregate real-world dataset, composed of local datasets D_i potentially drawn from heterogeneous local distributions $P_i(X)$ as defined previously. We can conceptualize the collective characteristics of this aggregate data $\mathcal{D}$ as being represented by a *true underlying global distribution*, denoted by $P(X)$. This $P(X)$ represents the complete data generating process we ideally wish to capture, encompassing the variations across all nodes. Synthetic data, denoted $\tilde{\mathcal{D}}$, is produced by a class of machine learning algorithms known as generative models. These models aim to learn an approximate distribution $Q(X)$ such that samples drawn from $Q(X)$ closely resemble samples drawn from the target distribution $P(X)$.

Let us revisit the formalization presented earlier. We have a set of n participating nodes $\mathcal{N} = \{N_1, N_2, ..., N_n\}$, where each node N_i has a local dataset D_i drawn from a local data distribution $P_i(X)$. These distributions can be heterogeneous, meaning

$P_i(X) \neq P_j(X)$ for some $i \neq j$. Our overarching goal is to learn a global function $f : \mathcal{X} \to \mathcal{Y}$ that maps input features $x \in \mathcal{X}$ (patient data) to target variables $y \in \mathcal{Y}$ (health outcomes).

In the context of DHTs, the challenge is to learn this function f without direct access to the aggregated real dataset $\mathcal{D} = \cup_{i \in \mathcal{N}} D_i$ due to privacy and logistical constraints. This is where generative models offer a powerful solution. From a statistical perspective, we can frame the problem as *density estimation*. Each local distribution $P_i(X)$ represents the probability density function of the data at node N_i. A generative model, trained on D_i, aims to learn an approximation of this density, which we denote as $Q_i(X)$. Ideally, $Q_i(X)$ should be a close approximation of $P_i(X)$, such that samples drawn from $Q_i(X)$ are statistically indistinguishable from those from $P_i(X)$ but do not reveal any sensitive information about individual patients.

Formally, we can represent the process of generating synthetic data at each node N_i as follows:

1. **Learning the Local Density:** A generative model G_i is trained on the local dataset D_i to learn the underlying probability density function $P_i(X)$. This results in an estimated density $Q_i(X)$.
2. **Sampling Synthetic Data:** New data points x_{synth} are sampled from the learned density $Q_i(X)$. These synthetic data points constitute the local synthetic dataset $\tilde{D}_i$ at node N_i.

The collection of these local synthetic datasets, $\{\tilde{D}_1, \tilde{D}_2, ..., \tilde{D}_n\}$, can then be used for various downstream tasks, such as training the global function f in a distributed or federated manner, without compromising the privacy of the original data. Instead of sharing the real data, the nodes can share their synthetic data, effectively abstracting away the need for direct access to real data.

This approach significantly mitigates the challenges associated with data access in federated learning, as was previously highlighted in Section 1.2. While FL allows for decentralized model training without central data aggregation, it still requires iterative access to the real data at each node. This iterative access can be logistically complex,

requiring repeated ethical approvals and data access permissions for each round of model training and refinement. Moreover, ensuring simultaneous data availability from all participating entities throughout the model development process poses significant coordination challenges.

By employing privacy preserving generative models, we essentially create a privacy-preserving abstraction layer between the sensitive real data and the model training process. Each node independently generates its local synthetic dataset $\tilde{D}_i$ based on its local data D_i. This synthetic data, by design, does not contain identifiable information... The federated learning process then operates on these synthetic datasets, eliminating the need for direct access to the original data during the iterative training and update phases.

Furthermore, the use of synthetic data addresses the issue of data availability and coordination. Since each node can generate its synthetic data independently and asynchronously, there is no need for simultaneous access to the real data across all nodes. This asynchronous generation allows for a more flexible and efficient training process, as nodes can proceed at their own pace without being bottlenecked by the availability of others. Model refinement and evaluation can also be performed using the synthetic data, reducing the need for expert review of real data outputs and further simplifying the ethical and logistical aspects of the process.

Generative models, such as Generative Adversarial Networks (GANs) [28], Variational Autoencoders (VAEs) [29], and other probabilistic generative approaches, have demonstrated their capacity to learn complex data distributions and generate new instances that retain the statistical characteristics of the original datasets. By training these models on available health data, they can produce synthetic datasets that are not directly traceable to real individuals, thereby mitigating privacy concerns while providing a fertile ground for the development and testing of DHTs and training and deploying the underlying machine learning models enabling sophisticated data analytics.

It is important to note that the success of this approach hinges on the ability of the generative models to accurately capture the statistical properties of the real data ($\mathcal{D}$). This means that the synthetic data must preserve the essential characteristics of the real

data, including the distribution of individual variables, correlations between variables, and potentially the tail behavior of the distributions, while not memorizing or revealing specific information about any individual in the original dataset $\mathcal{D}$. For instance, ensuring the synthetic data accurately captures rare events or complex correlations present in the real data is a significant challenge. Furthermore, the computational cost of training complex generative models, especially in a distributed setting, needs to be considered. This leads us to identify several largely unaddressed challenges which must be addressed to ensure the effective replacement of real data with synthetic data. These challenges range from quality and fidelity of the generated data, privacy preservation, computational overhead, data drift and distribution shift, model and data validation and evaluation, longitudinal data complexity, explainability and regulatory compliance. We will define our research questions based on these challenged in Section 1.4.

However, before we continue with our research question, we will make a quick visit back to federated learning. We believe that to fully realize the potential of this synthetic data paradigm, we must carefully consider how it integrates with the federated learning framework, particularly in terms of model aggregation and optimization across distributed synthetic datasets. This necessitates a detailed exploration of how to reconcile the principles of federated learning with the use of locally generated synthetic data to achieve globally optimal DHT models.

Reconciling Federated Learning and Synthetic Data for DHT Data Access and Processing To further illustrate the integration of federated learning with synthetic data, let us consider a federated network consisting of n nodes, where each node i possesses a local dataset D_i drawn from the overall data distribution $P_i(X)$. Each local dataset D_i represents a subset of the complete real dataset $\mathcal{D}$ and reflects the local characteristics and variability of the data at node i. In a conventional federated learning setup, a model M_{global} is trained across these nodes to learn a global representation by aggregating locally computed updates, typically gradients or model parameters, to achieve a consensus model M_{global}.

In the proposed approach, each node i first employs a local generative model G_i to learn the distribution $P_i(X)$ of its dataset D_i. The objective of each G_i is to generate a

local synthetic dataset $\tilde{D}_i$ that mimics D_i in terms of statistical properties but does not contain any identifiable or direct replicas of the original data instances. Mathematically, the local generative model G_i aims to minimize the divergence between $P_i(X)$ and the generated distribution $Q_i(X)$, ideally achieving $Q_i(X) \sim P_i(X)$.

Once each node has trained its generative model G_i and produced $\tilde{D}_i$, the federated learning process begins, but with a significant modification: instead of using the actual local datasets D_i, each node uses its synthetic counterpart $\tilde{D}_i$ for local training. This approach ensures that the sensitive real data remains confined to its original node, enhancing privacy. Each node trains a local model M_i on $\tilde{D}_i$ and computes updates based on this training.

The federated learning framework then aggregates these local updates to form a global model M_{global}. This aggregation can be performed using techniques such as Federated Averaging (FedAvg), where the updates (e.g., weights from neural networks) from each local model M_i are averaged to update M_{global}.

This process iterates over several rounds, with each round consisting of local training on synthetic data and aggregation of the updates to refine M_{global}. The global model M_{global} thus trained is expected to perform well across the federation, representing the collective knowledge and data characteristics of all nodes, albeit indirectly through their synthetic datasets ($\tilde{D}_i$).

The efficacy of this approach hinges on the quality and representativeness of the synthetic data generated by the local generative models. Since each $\tilde{D}_i$ is intended to be a statistically faithful reproduction of D_i, the global model M_{global} trained on these datasets should ideally approximate the performance of a model trained directly on the aggregated real dataset $\mathcal{D} = \cup_{i \in \mathcal{N}} D_i$ across the federation, whilst significantly mitigating risks related to data privacy and security breaches. This leads us to identify several largely unaddressed challenges which must be addressed to ensure the effective replacement of real data with synthetic data. These challenges range from quality and fidelity of the generated data, privacy preservation, computational overhead, data drift and distribution shift, model and data validation and evaluation, longitudinal data complexity, explainability and regulatory compliance.

In Appendix A, we provide a comprehensive review of the state-of-the-art in privacy-preserving federated learning, analyzing the strengths and limitations of existing approaches and highlighting the open research questions in this rapidly evolving field. However, for the remainder of this thesis, our focus will shift towards the development and application of generative models for synthetic data generation, exploring how these models can be leveraged to unlock the full potential of Digital Health Twins while upholding the highest standards of data privacy.

1.4 Research Questions

The overarching aim of this thesis is to operationalize Digital Health Twins (DHTs) by enabling private distributed data analytics through the replacement of real data with private synthetic data. This is achieved through the innovative application of generative models, abstracting data sources from the direct utilization of real-world data in DHT-related tasks. To refine the scope of this research, consultations with principal stakeholders in the healthcare and data privacy domains were conducted. These consultations, along with a thorough literature review, resulted in the identification of four focal challenges: *the fidelity/diversity trade-off*, *the handling of mixed-tail data*, *privacy*, and the *decentralized nature of the data*. We define our research questions based on these main challenges, which represent critical barriers to the successful deployment of DHTs.

Mixed Tail Behavior Real-world health data often exhibit *mixed tail behavior*, where variables within the same dataset possess fundamentally different distributional tailssome decaying slowly (heavy-tailed) while others decay rapidly (light-tailed) [30, 31, 32, 33]. This reflects the multifaceted nature of health data, containing both rare, extreme events (e.g., disease incidence) and common, less variable measurements (e.g., heart rate).

The challenge lies not just in correctly specifying the tail for each variable individually, but in fitting a single generative model that can simultaneously accommodate these disparate characteristics. Standard generative models often implicitly assume a uniform

tail behavior (e.g., Gaussian priors/latents imply balanced and/or light tails) or lack the flexibility to represent both heavy and light tails accurately across different dimensions. This mismatch poses several difficulties during model training and generation:

- **Optimization Instability:** Heavy-tailed variables generate outliers that can dominate gradients, destabilizing the training process (e.g., MLE or adversarial training) and hindering convergence for the lighter-tailed components.
- **Model Flexibility:** Architectures may struggle to capture both the rapid decay of light tails and the power-law-like decay of heavy tails within a unified framework.
- **Likelihood Balancing:** Likelihood-based models must balance fitting high-density regions (often associated with light tails) against capturing low-density but potentially high-impact tail events (heavy tails).
- **Joint Distribution Modeling:** Capturing the complex dependencies *between* variables with different tail types is crucial but particularly difficult.

Formally, let $X = \{X_1, X_2, \ldots, X_d\}$ be a set of variables in the real dataset $\mathcal{D}$, drawn from a true distribution $P(X)$. The challenge is to learn a generative model $Q(X)$ such that the tail behavior $T_Q(X_i)$ of each generated variable accurately reflects the true tail behavior $T_P(X_i)$, despite the inherent heterogeneity in tail types across the set $\{X_i\}$. Failure manifests as a significant discrepancy, $\text{Divr}(T_P(X_i), T_Q(X_i)) > \tau$, for one or more variables, often because the model compromises on fitting some tails to accommodate others, or because the training process is unduly influenced by one type of tail behavior. (Here, Divr is a suitable divergence metric such as KL-divergence and τ an acceptable tolerance).

Effectively modeling this mixed-tail reality is essential for generating realistic synthetic health data for DHTs, particularly for risk assessment and outcome prediction. Standard generative approaches often fall short, motivating the need for specialized techniques. This leads to our first research question:

> **RQ1**: How can generative models estimate densities exhibiting heterogeneous tail behaviors?

Fidelity-Diversity Trade-Off The utility of synthetic data is predicated on its ability to reflect the real-world data's complexity. Generative models face the challenge of balancing fidelity (the accuracy with which the synthetic data represents the original data) with diversity (the variety within the synthetic data). In this context, the exploration/exploitation trade-off is evident as increasing the emphasis on fidelity (exploitation of known data patterns) inherently reduces the model's incentive to generate diverse outputs (exploration of new data variations), potentially limiting the synthetic dataset's comprehensive representation of real-world complexity. Given a real-world dataset $\mathcal{D} \sim P(X)$, the objective is to train a generative model that can produce a synthetic dataset $\tilde{\mathcal{D}}$ from a learned distribution $Q(X)$. The generative model's performance in the context of exploration/exploitation trade-off can be characterized by two criteria:

1. **Fidelity**: The synthetic data should accurately reflect the real dataset. This can be quantified by a similarity metric S between distributions, such that the fidelity is high if $S(Q(X), P(X))$ is maximized, indicating that $Q(X)$ closely approximates $P(X)$.
2. **Diversity**: The synthetic data should exhibit variability that is representative of the real dataset. This can be quantified by a diversity metric Divr within $\tilde{\mathcal{D}}$, such that diversity is high if $\text{Divr}(\tilde{\mathcal{D}})$ is maximized, subject to the constraint that $Q(X)$ maintains an acceptable level of fidelity to $P(X)$

The trade-off between fidelity and diversity can be represented as an optimization problem:

$$\min_{Q} \left\{ -S(Q(X), P(X)) - \lambda \cdot \text{Divr}(\tilde{\mathcal{D}}) \right\}$$

where λ is a regularization parameter that balances the two objectives. The generative model must be optimized in such a way that $\tilde{\mathcal{D}}$ not only captures the complex statistical properties of $\mathcal{D}$ but also introduces a breadth of novel instances that avoid mere replication of the existing data points, thus ensuring a robust and comprehensive dataset for developing Digital Health Technologies (DHTs). Overemphasizing fidelity

can lead to overfitting, where the generative model simply replicates the training data, while prioritizing diversity too highly might result in synthetic data that fails to capture crucial statistical properties of the original dataset. To address this challenge, we pose the following research question:

> **RQ2**: How can generative models synthesize data with high fidelity and diversity while respecting the implicit semantic boundaries of the real data?

Privacy Generative models can mitigate disclosure by producing de-identified outputs, yet the risk of membership or attribute inference remains non-negligible when models capture fine-grained structure in the source data. Our aim is to formalize privacy for *released generative mechanisms*. We require that an adversary cannot reliably infer the presence or absence of any individual record from access to the mechanisms output. We adopt differential privacy (DP) under the *add/remove* (unbounded) adjacency: datasets $\mathcal{D}$ and $\mathcal{D}'$ are neighbors if one can be obtained from the other by adding or removing a single record. A randomized mechanism M is (ε, δ)-DP if, for all measurable sets $\mathcal{S}$,

$$\Pr\Big(M(\mathcal{D}) \in \mathcal{S}\Big) \;\leq\; e^{\varepsilon}\, \Pr\Big(M(\mathcal{D}') \in \mathcal{S}\Big) \;+\; \delta,$$

so the output distributions induced by $\mathcal{D}$ and $\mathcal{D}'$ are nearly indistinguishable in the strongest single-record sense [34].

Differential privacy can be combined with generative modeling to yield formal protection, but naive applications often degrade fidelity in regions that matter for downstream analysis, including heterogeneous tails and clinically meaningful semantic constraints. For deployment in digital health technologies (DHT), the problem is to obtain quantifiable privacy assurances that preserve density fidelity under add/remove adjacency and admit stable accounting without relying on brittle noise-injected optimization. To address this problem, we pose the following research question:

RQ3: How can Normalizing Flows be released under (ε, δ)-differential privacy while preserving density fidelity without resorting to private stochastic optimization?

Data Disparity and Geographic Distribution Health data used in DHTs often originate from disparate sources, each with its own data collection protocols, formats, and levels of granularity. Moreover, these data entities may be stored in geographically diverse locations subject to different regulatory frameworks. This fragmentation complicates the creation of a coherent, unified dataset which generative models require for training, and poses challenges for centralized approaches [35]. Formally, let's consider a subset of k participating nodes $\{N_i\}_{i=1}^{k}$ from the total network $\mathcal{N}$ (which has size $|\mathcal{N}| = n$). Each node N_i in this subset (where $k \leq n$) possesses a local dataset D_i. Due to various factors such as demographic differences, distinct medical practices, and varying data collection protocols, the data is often non-IID (Independent and Identically Distributed), implying:

$$P_i(X) \neq P_j(X)), \quad \text{for some } i, j$$

The challenge in this context is to train a global generative model G, using the potentially heterogeneous data distributed across the participating nodes, such that it learns an approximate distribution $Q(X)$ which accurately reflects the true, underlying global data distribution $P(X)$ (i.e., aiming for $Q(X) \approx P(X)$). Despite the heterogeneity of the local data sources, the resulting model G should generalize well across all nodes and produce synthetic data $\tilde{\mathcal{D}}$ that is representative of the collective real-world health data distribution $P(X)$. This ensures that the Distributed Health Technologies (DHTs) can be constructed, validated, and refined effectively, accounting for the full spectrum of health data variability. Therefore, we define our fourth research question as:

RQ4: How to perform efficient, fair data analysis in a distributed environment by utilizing synthetic data?

The prioritization of these challenges is based on their recognized significance in the context of DHT utility and acceptability. The privacy issue is central due to the confi-

dential nature of health data and the legal and ethical requirements to protect it. The fidelity/diversity trade-off is critical in maintaining the quality and robustness of DHT simulations. For instance, a model that perfectly memorizes the training data would have high fidelity but no diversity, while a model that generates completely random data would have high diversity but no fidelity. Finally, the accurate handling of mixed-tail data is essential to ensure realistic modeling of health outcomes. Addressing these challenges is vital to advance the use of DHTs in a manner that is both secure and effective.

1.5 Thesis overview

This thesis is structured as follows. Chapters 2-5 address research questions 1-4 respectively, proposing novel methods applicable to both centralized and distributed settings. Chapter 2 introduces a method to make normalizing flows tail-adaptive, enabling accurate estimation of target density and tail properties. Chapter 3 presents a model-agnostic framework for unsupervised learning of semantic constraints in the target dataset, incorporating a validator to steer the generative model towards adhering to these constraints. This framework utilizes the tail-adaptive normalizing flow from Chapter 2 as a prior for the latent space validator component. Chapter 4 proposes a mechanism for generating differentially private tabular synthetic data using normalizing flows without stochastic private optimization, improving upon current noise-adding approaches. Chapter 5 integrates our tail-adaptive, semantically-aware flow model into a proof-of-concept DHT framework optimized for clinical pathways, demonstrating the practical application of our research. Chapter 6 concludes by providing a critical perspective on the presented ideas and outlining potential future research directions. Appendix Appendix A reprises our survey of privacy-preserving federated and secure learning, providing the context for distributed training and inference against which the thesis positions its synthetic-data alternative, and fixing the terminology and baselines used in later chapters (including Chapter 5), and concise preliminary knowledge section is included in the (Appendix B) to ensure this document is self-contained (Having provided this preliminary knowledge section, in the method chapters we refrain from including

these definitions unless absolutely necessary and when they are being directly referred to and built upon in that specific chapter).

1.6 Contributions

This thesis presents several original contributions to the fields of latent modelling, semantic analysis, differential privacy, and federated learning. Each chapter builds upon the author's prior work, as detailed below. Additional publications by the author, which align with the research methodologies and thematic directions of this dissertation but are not elaborated within its specific narrative scope, are also listed to provide a comprehensive record of the author's research contributions in these areas.

Chapter 2 The developments in Chapter 2 include and extend the findings reported in:

- Saba Amiri, Eric Nalisnick, Adam Belloum, Sander Klous, and Leon Gommans. Generating heavy-tailed synthetic data with normalizing flows. In *The 5th Workshop on Tractable Probabilistic Modeling*, 2022,
- Saba Amiri, Eric Nalisnick, Adam Belloum, Sander Klous, and Leon Gommans. Practical synthesis of mixed-tailed data with normalizing flows. *Transactions on Machine Learning Research*, 2024.

Chapter 3 This chapter incorporates and builds upon:

- Saba Amiri, Carlijn Nijhuis, Eric Nalisnick, Adam Belloum, Sander Klous, and Leon Gommans. Latent Semantic Regularization: Enhancing semantic integrity in tabular data synthesis. In *Proceedings of the Symposium on Probabilistic Machine Learning*, Seoul, South Korea, July 2026. URL `https://probml.cc/`. ProbML 2026, co-located with ICML 2026.

Chapter 4: The contributions of Chapter 4 are based on:

- Saba Amiri, Eric Nalisnick, Adam Belloum, Sander Klous, and Leon Gommans. Constrained-expressivity differentially private normalizing flows for tabular data synthesis. Working Paper: Manuscript in preparation, 2025.

Additional related work (not detailed herein):

- Saba Amiri, Adam Belloum, Sander Klous, and Leon Gommans. Compressive differentially private federated learning through universal vector quantization. In *AAAI Workshop on Privacy-Preserving Artificial Intelligence*, pages 2–9, 2021
- Saba Amiri, Eric Nalisnick, Adam Belloum, Sander Klous, and Leon Gommans. Differential privacy vs detecting copyright infringement: A case study with normalizing flows. In *1st Workshop on Generative AI and Law, International Conference on Machine Learning (ICML)*, 2023

Chapter 5 The material in Chapter 5 extends the works:

- Saba Amiri, Adam Belloum, Eric Nalisnick, Sander Klous, and Leon Gommans. On the impact of non-iid data on the performance and fairness of differentially private federated learning. In *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 52–58. IEEE, 2022,
- Saba Amiri, Tim Muller, Rosanne J Turner, Karin Hoogoort, Aki Härmä, Johanna Maria Hegeman, Jamila Alsayed Kassem, Milen Kebede, Cees De Laat, Paul Van Der Nat, Auke Pals, Floortje Scheepers, Sander Klous, et al. Optimizing clinical pathways with federated data. *Jusletter-IT*, (20-Februar-2025), 2025[1].

Additional related work (not detailed herein):

- Tsvetan R Yordanov, Anita CJ Ravelli, Saba Amiri, Marije Vis, Saskia Houterman, Sebastian R Van der Voort, and Ameen Abu-Hanna. Performance of fed-

[1]Saba Amiri, Tim Muller, and Rosanne Turner were shared first author of this paper. They co-designed and co-executed this research equally. Amiri contributed to the structure and design of this study alongside his co-authors. Amiri performed research and wrote the corresponding section related to evaluation of synthetic data on downstream machine learning tasks.

erated learning-based models in the dutch tavi population was comparable to central strategies and outperformed local strategies. *Frontiers in Cardiovascular Medicine*, 11:1399138, 2024[2]

Appendix A Appendix A includes material from:

- Corinne Allaart, Saba Amiri, Henri Bal, Adam Belloum, Leon Gommans, Aart Van Halteren, and Sander Klous. Private and secure distributed deep learning: A survey. *ACM Computing Surveys*, 57(4):1–43, 2024[3].

[2]Amiri contributed to the research and experimental design of the Federated Learning aspects of this study.

[3]Amiri and Allaart were shared first author of this paper. They co-designed and co-executed the literature study equally. Amiri and Allaart wrote the privacy and security sections of the manuscript respectively. All other chapters were co-written and edited jointly by Amiri and Allaart. Bal, Belloum, Gommans, van Halteren and Klous provided supervision during the design and execution of the project, and contributed to the final manuscript.

CHAPTER
2

CAPTURING TAIL BEHAVIOR IN SYNTHETIC DATA GENERATION

RQ1: How can generative models estimate densities exhibiting heterogeneous tail behaviors?

2.1 MOTIVATION

As established in Chapter 1, synthetic data has proven invaluable across various domains, serving as a cornerstone for training machine learning models, enhancing data privacy, and augmenting insufficient real-world datasets. The versatility of synthetic data is underpinned by its ability to replicate complex patterns inherent in real-world data, thereby facilitating robust model training and validation. There is a rich body of literature on techniques that leverage generative models, such as Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), and Normalizing Flows (NFs), to produce multi-modal, high-fidelity, tabular synthetic data. These methodologies have been increasingly effective in capturing the intricate distributions and modalities present in diverse data types.

Despite the advancements in synthetic data generation, however, a significant gap persists in the literature regarding the generation of data that accurately reflects heavy-

and mixed-tailed distributions. Real-world datasets often deviate from the idealized assumption of exponentially decaying tails, presenting tail behaviors that can range from sub-exponential to super-exponential. In empirical investigations of real-world data, the manifestation of either short or heavy tails is a recurrent characteristic that can be observed across a multitude of domains. For example, in financial econometrics, the distribution of stock returns has been frequently observed to exhibit heavy tails, deviating from the Gaussian assumption and indicating a higher tendency for extreme variations. Such distributions can often be better characterized by power laws or Pareto distributions. Similarly, in the realm of telecommunications, data traffic can exhibit heavy-tailed behavior, with a small number of users accounting for a disproportionate amount of data usage, reflecting principles akin to Zipf's law. In contrast, the physical dimensions of manufactured components, as a consequence of precise engineering and quality control processes, typically follow a short-tailed distribution, where deviations from the mean are modest and extreme values are exceedingly rare. This dichotomy in tail behavior exemplifies the diversity of distributional forms in real-world data, underscoring the need for tailored generative models to accurately capture the underlying stochastic phenomena.

The multimodal nature of datasets further complicates this issue, as different modes may exhibit distinct tail behaviors. A single dataset may encompass subpopulations that follow subexponential tails, alongside others that exhibit exponential or superexponential decay. This mixed-tail behavior is a facet of the underlying data complexity that generative models must accurately capture to ensure the fidelity of the synthetic data.

The accurate capture of tail behavior is not merely a statistical concern, but has profound implications for the application of synthetic data. Failing to model the tails of a distribution correctly can lead to synthetic data that either overestimates or underestimates the likelihood of extreme events, known as tail risks. Overestimation of tail heaviness can result in an overly cautious stance, where models anticipate extreme events more frequently than they occur. This can lead to inefficient resource allocation, such as over-provisioning in supply chain management or excessive capital reserves in finance.

In contrast, underestimating tail behavior can be equally detrimental. Models trained on synthetic data with insufficiently heavy tails may exhibit overconfidence in their predictions, neglecting the possibility of rare but impactful events. In domains such as financial risk management, this can lead to undervaluing the risk of market crashes, while in healthcare, it might translate into underestimating the potential for rare adverse drug reactions.

Therefore, there is a pressing need for generative models that can adapt to the heavy- and mixed-tailed characteristics of data. Such models would not only improve the quality of synthetic data, but would also enhance the reliability of predictive models that are trained using these data. The development of these generative techniques should focus on incorporating mechanisms that allow them to learn and replicate the statistical properties of tail behavior, acknowledging the variance in tail heaviness across and within datasets.

Normalizing flows [46] present an attractive model for mixed-tailed data synthesis. This is due to the fact that the tail behavior of the base distribution directly and transparently affects the tail of the generated data [47]. Other classes of generative models do not provide such a mechanism to control their tails. Generative adversarial networks have been a popular choice for data synthesis [48], but the effect of the base distribution on the resulting data model is unknown. Figure 2-1 demonstrates the impact of mis-specification of parameterization of base density on the capabilities of normalizing flows on capturing the tail behavior of targets with different tail behaviors.

The task of synthesizing mixed-tail data using normalizing flows is an intriguing research domain within generative modeling. The previous methods adapt the base distribution to match the target tails. One potential concern with applying these methods to mixed-tail targets is that the base distribution might be overly conditioned on the estimated or perceived tail of the target. This makes the flow model brittle, as will be discussed in Section 2.3. We also posit that mixed-tail data can negatively impact model gradients during training regardless of the choice the family and tail behavior of the base density (See Section 2.4.2). With heavy tails, a few outlier samples can contribute huge gradients that overwhelm the signal from the majority of data. This makes opti-

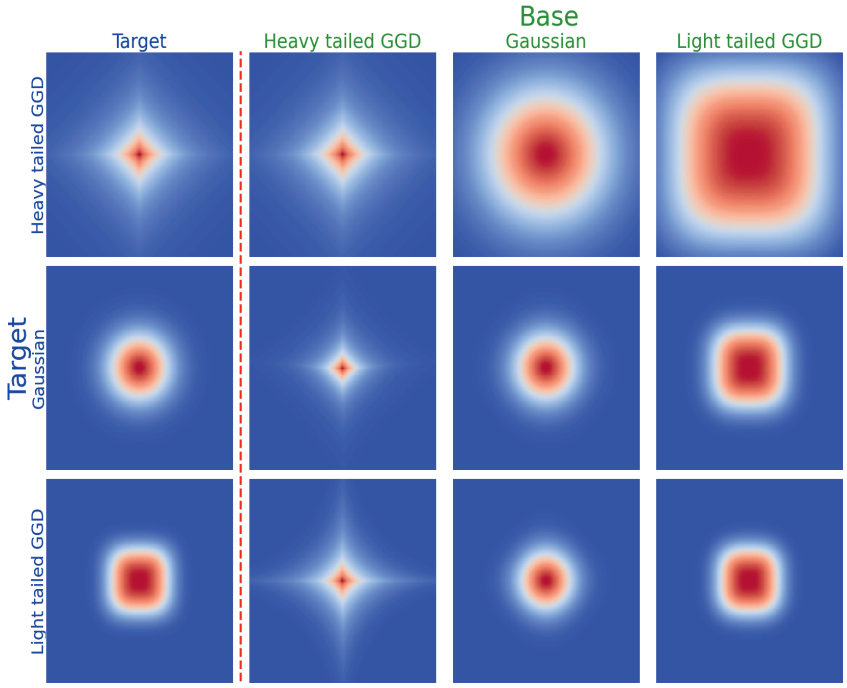


Figure 2-1: The impact of mis-specification of base density on the capabilities of normalizing flows to capture the tail behavior of the target density. We train an RNVP on three short, normal and heavy-tailed *Generalized Gaussian (GGD)* targets (columns 2-4) with short, normal and heavy-tailed GGD base densities (rows 1-3). We plot the probability density of the plot. Mis-specified parameterization of base leads to flow being only able to estimate the target with the same tail behavior as its base density accurately

mization unstable and sensitive to initial conditions. Also for mixed-tailed minibatches, the Monte Carlo estimates of the log-likelihood may not concentrate around the true log-likelihood.

In this chapter, we focus on improving the flow model for mixed-tail targets. We make the base density of the flow flexible enough to capture the tailedness of the target through a flexible mixture of *Generalized Gaussians* without binding the base to the tail properties of the target distribution. To make the flow model itself capable of capturing the tail of the target distribution, we make the training process robust to heavy-tailed gradients. By making the Monte Carlo estimator robust, we stabilize the training process and enable model to capture the tail behavior of the target distribution

and admit gradient-based adaptation of the tail behavior. We validate our method on a range of data synthesis tasks and empirically show the advantage of our method for mixed-tail targets, since our focus on the flow model rather than just the base distribution provides a more adaptable way to synthesize mixed-tail data.

2.2 Formalizing Tail Behavior Characteristics

A random variable X is said to follow a *heavy-tailed* distribution if its tail is not exponentially bounded. This can be formally defined using the moment generating function (MGF): $M_X(t) = E[e^{tX}]$. A distribution is heavy-tailed if there exists some $t > 0$ such that $M_X(t)$ is infinite or does not exist. An example of this category is the Pareto distribution with scale parameter $\gamma > 0$, for which the MGF is $M_X(t) = \frac{\gamma}{\gamma - t}$, which is undefined for $t \geq \gamma$ [1]

The tails of a *light-tailed* random variable X decay sub-exponentially. The Gumbel distribution, with location parameter μ and scale parameter $\beta > 0$, is an example of a light-tailed distribution as its survival function $S_X(x) = e^{-e^{-(x-\mu)/\beta}}$ indicates a sub-exponential decay for large x.

Normal-tailed distributions refers to the special case of Gaussian distribution. The moment generating function of Gaussian is $M_X(t) = e^{\mu t + \frac{1}{2}\sigma^2 t^2}$, where μ is the mean and σ is the standard deviation. This function exists for all t, as long as σ^2 is finite, defining the normally-tailed behavior.

In many real-world scenarios, we encounter data with mixed tail properties. We define such data as having *mixed tails* if some of its marginal distributions exhibit heavy tails and others light tails.

[1]The MGF of the Pareto distribution is defined as $M_X(t) = \int_{x_m}^{\infty} e^{tx} f_X(x)\, dx$ where x_m is the minimum possible value of X and $f_X(x)$ is the Pareto PDF. When $t \geq \gamma$, the MGF denominator $\gamma - t$ becomes zero or negative, leading to a division by zero or a negative number in the denominator, causing the integral to diverge to infinity or not converge to a finite value.

2.2.1 Tail Index Estimation

The tail behavior of a probability distribution is primarily characterized by their *tail index*. The tail index, which we denote as α, is defined as the exponent in the power-law tail of the distribution:

$$F(x) = P(X \leq x) \sim 1 - x^{-\alpha}$$

where $P(X \geq x)$ is the probability of the random variable X being greater than or equal to a certain value x, and $F(x)$ is the cumulative distribution function of X. As α approaches zero, the distribution follows a power law to a greater degree [2]. *Hill's estimator* [49] is a widely-used method for estimating the tail index. It is defined as follows:

$$\hat{\xi}H = \left[\frac{1}{k}\sum i = 1^{k} \left(\log(X_{n-i+1}) - \log(X_{n-k})\right)\right]^{-1}$$

Here, n represents the sample size, k is the number of upper order statistics, and X_{n-i+1} denotes the ith largest observation in the sample. The parameter k is of crucial importance: a large k inflates the variance of the estimator, while a small k increases the bias [50].

An alternative nonparametric class of estimators for α are *kernel estimators*, first proposed by Csorgo et al. [51]. The main idea is to use a kernel function K(x) to smooth the empirical distribution and estimate the tail function as:

$\hat{F}(x) = (1/nh)\sum_{i=1}^{n} K((x - X_i)/h)$

where h is a bandwidth parameter. With second order kernels, these estimators converge at rate $n^{-2/5}$, slower than the parametric rate [3].

Dekkers et al. [52] proposed a class of *moment estimators* that leverage the relationship for large k [4]:

[2] A power law is a functional relationship between two quantities, where one quantity varies as a power of another. As α approaches 0, the expression $x^{-\alpha}$ approaches 1 for any fixed x. However, this does not mean that $1 - x^{-\alpha}$ goes to 0 independently of x, because α is not approaching 0 independently of x. If α were to approach 0, the distribution would become heavier-tailed, meaning that the probability of observing very large values of X does not fall off as quickly as x increases.

[3] This convergence rate is slower than parametric estimators, typically $n^{-1/2}$, meaning more data is required for similar accuracy, affecting the choice between more flexible nonparametric methods and more efficient parametric ones, particularly in scenarios with limited data in distribution tails.

[4] Using a large k in tail index estimation increases the number of extreme values included in the

$E(X_{(n-k)}) = Cn/k^{\alpha}$

where C is a constant. This motivates the moment estimator:

$\hat{\xi}m = k/n \sum i = 1^{k} \ln(X_{(n-i+1)}/X_{(n-k)})$

2.3 LIMITATIONS OF EXISTING APPROACHES FOR MIXED-TAILED MODELLING

Reviewing literature indicates that to adapt a flow model's base density to the target tail, existing methods either incorporate assumptions about tail behavior into the model through a specific base density selection, as in [47], or directly estimate the target's tail behavior, as in the marginal adaptive base method by Laszkiewicz et al. [1]. We explore the limitations of these approaches in modeling mixed-tailed data.

2.3.1 Pre-Hoc Heavy-Tailed Base Density Selection Challenges for Mixed-Tail Targets

Real-world data may have tail behaviors that deviate from any specific choice of base distribution family such as having a multi-modal mixed-tail distribution. Therefore, choosing a specific heavy-tailed parametric distribution family as the flow base as suggested by Jaini et al. [47] a priori will inherently restrict the range of possible tail shapes and asymptotic decay rates that can be represented. The other issue with this method for mixed-tail targets is based on the fact that normalizing flow transforms on top of the base distribution and can either alter its tail behavior significantly or not converge due to exploding/vanishing gradients. Thus to make the flow robust, the tail modeling capabilities must come more from the flows rather than the base distribution alone.

calculation, which can provide a more accurate representation of the tail behavior. However, this comes at the cost of higher variance in the estimator. Leveraging this relationship, Dekkers et al. [52] intentionally choose a larger k to capture more information about the tail at the expense of estimator stability, which can be beneficial when the tail properties are of primary interest and the increased sample variability can be tolerated.

2.3.2 Pre-Hoc Target Tail-Estimation Challenges for Mixed-Tail Targets

Adapting the tail behavior of the flow's base density directly on the target distribution is another way to make the base distribution tail adaptive, with a prominent example being Marginal Tail Adaptive Flows (mTAF) by Laszkiewicz et al. [1] . mTAFs calculate the tail index of each marginal of x. Since tail estimation is notoriously difficult, Laszkiewicz et al. [1] combine the results from three different tail index estimators. The base density for light-tail marginals is set to a Gaussian and a univariate Student's t distribution with a degree of freedom equal to the estimated tail index for heavy tail marginals.

The limitation of these methods for modelling complicated mixed-tail targets is their reliance on estimated tail index of the target distribution. Tail index estimators generally suffer from high variance and sensitivity to tuning parameters, especially for small sample sizes or multidimensional settings. For example, a major drawback of Hill's estimator is its sensitivity to the choice of k. Mason [53] demonstrated that $\hat{\xi}_H$ is inconsistent if k stays fixed as n approaches infinity. In moment-based tail estimators, the estimator $\hat{\xi}_m$ needs the second order regular variation condition to hold, which may not always be the case in practice [54]. Furthermore, $\hat{\xi}_m$ can exhibit large variance and deviation when the underlying assumptions are violated. In the case of the non-parametric kernel tail estimators, the choice of bandwidth h is highly sensitive, and optimizing over the class of kernel functions can be challenging [55].

Figure 2-2 shows the accuracy and consistency of the bootstrapped Hill's estimator $\hat{\xi}_H$ for samples with varying sizes from a Pareto distribution with different tail indices. We report the difference between true and estimated tail indices. We observe that $\hat{\xi}_H$ overestimates for heavy-tailed samples and underestimates for short-tailed ones. We also observe increased variance for smaller batch sizes – closer to the usual minibatch sizes – from extremely heavy-tailed targets..

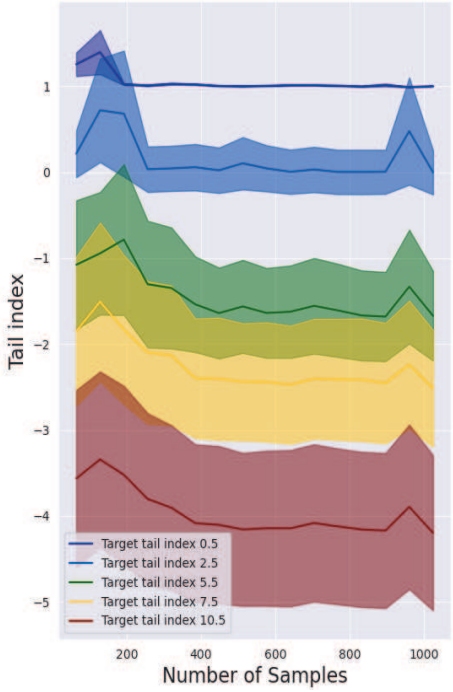


Figure 2-2: Mis-estimation of tail indices by bootstrapped Hill method for datasets sampled from a Pareto distribution. Tail indices vary from heavy (α=0.5) to short (α=10.5) and sample sizes from 64 to 2048. Each experiment is conducted 9 times and the mean - solid lines - and standard error - shades - of the difference between the true tail index and the estimated one are reported.

2.4 METHOD

Our approach to tail adaptiveness in any normalizing flow hinges on: a) ensuring the base distribution's flexibility to align with the target's tail behavior, and b) enhancing the flow's training process robustness to adequately capture the tail dynamics of the target distribution.

2.4.1 Tail-Adaptive Base Density for Mixed-Tail Targets

We propose using a mixture of generalized Gaussian distributions as the base density of the flow. The Generalized Gaussian Distribution (GGD) represents a family of continuous probability distributions that extend the concept of the Gaussian distribution by incorporating a shape parameter, denoted as $\beta > 0$. Formally, the probability density function (PDF) of a GGD with mean μ, scale parameter $\alpha > 0$, and shape parameter

$\beta > 0$ is defined as:

$$f(x; \mu, \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp\left(-\left(\frac{|x-\mu|}{\alpha}\right)^{\beta}\right)$$

where $\Gamma(\cdot)$ is the gamma function. This distribution encompasses a wide range of density shapes including but not limited to the Gaussian distribution ($\beta = 2$) and Laplacian distribution ($\beta = 1$). Higher values of β give rise to light-tailed platykurtic distributions, whereas lower β values lead to heavy-tailed leptokurtic distributions. Figure 2-3 shows the flexibility of the tail behavior of GGD with different shape parameters.

A trainable mixture model as the base distribution is defined as:

$$p_u(u; \pi_1, \ldots, \pi_K, \psi_1, \ldots, \psi_K) = \sum_{k=1}^{K} \pi_k \; p(u; \psi_k)$$

where k is the component index, $\pi_k \in [0, 1]$ is the weight of the kth component such that $\sum_k \pi_k = 1$, and ψ_k are the parameters of the kth GGD component parameterized by (μ, σ, β).

Our choice for using a GGD mixture for better modelling mixed tail behavior is substantiated by previous work that successfully used mixtures to model tailed distributions [56, 57, 58]. Hagemann and Neumayer [59] show that normalizing flow training can be stabilized in cases (e.g. disjoint support of base and target) by using a mixture for $p_u(\boldsymbol{u})$. We posit that training a flow model on mixed-tail targets will introduce optimization difficulties that may only be exacerbated by having a base density with wide-ranging tail behavior.

Adherence to [47] Our choice is motivated by the the ability of GGD to control its tail behavior through its shape parameter. Formally, consider a mixture model p_u where component densities $p(u; \psi_k)$ decay super-exponentially, i.e., $p(u; \psi_k) \lesssim e^{-c|x|^{\alpha}}, \alpha \in (0, 2)$, for some $c > 0$. The slowest decaying component ensures the mixture's heavy tail, as $p_u(u; \pi_1, \ldots, \pi_K, \psi_1, \ldots, \psi_K) \sim \pi_k \; p(u; \psi_k)$ as $x \to \infty$.

Considering a mixture of generalized Gaussians, $p(u; \psi_k)$ with $\psi = (\mu_k, \sigma_k, \beta_k)$, a heavy tail is possible if there exists component j which satisfies (i) $\beta_j < \beta_i \; \forall i \neq j$ and

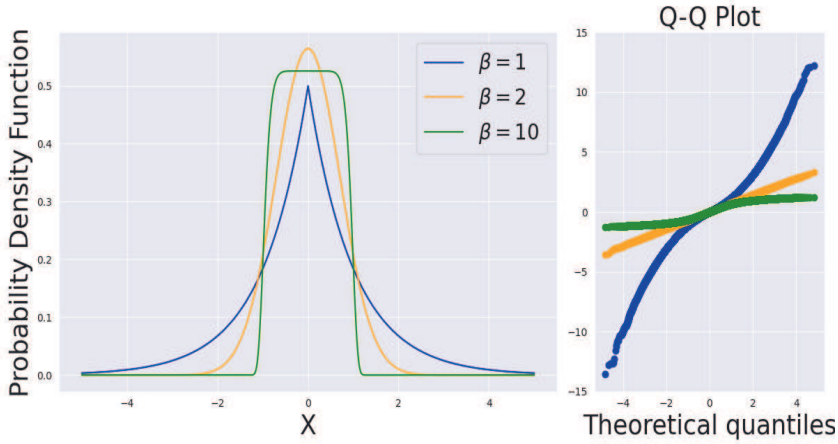


Figure 2-3: Probability density functions (left) and quantile functions (right) of generalized Gaussian distributions with shape parameter $\beta = 1.0, 2.0$, and 10.0. $\beta = 2.0$ yields a Gaussian. As β drops below 2.0, the PDF becomes heavy-tailed and the quantile function transitions from linear to concave down and vice versa.

(ii) $\beta_j \leq 2$. This indicates the j-th component's tail is heaviest and decays super-exponentially. By making β_j sufficiently small, we ensure a heavy tail, leading to $p_u(u) \sim \pi_j p(u; \psi_j)$ for large u. A single heavy-tailed component is enough to induce heavy tails in the mixture, while the remaining components can vary to adjust the overall density shape. This logic extends to creating arbitrarily light tails.

Advantage of our method in presence of a variety of tail behavior We empirically show the effectiveness of our proposed method in Figure 2-4, where we fit mixtures of generalized Gaussians with different number of components to data sampled from a Student's t-distribution with different degrees of freedom. As can be seen in the case of the heaviest-tailed sample with $\nu = 1$, e.g., a Cauchy distribution, increasing the number of components to 200 will result in the model being able to estimate the target distribution well. As the target distribution gets less heavy or short-tailed, we need less number of components in our mixture to estimate it accurately.

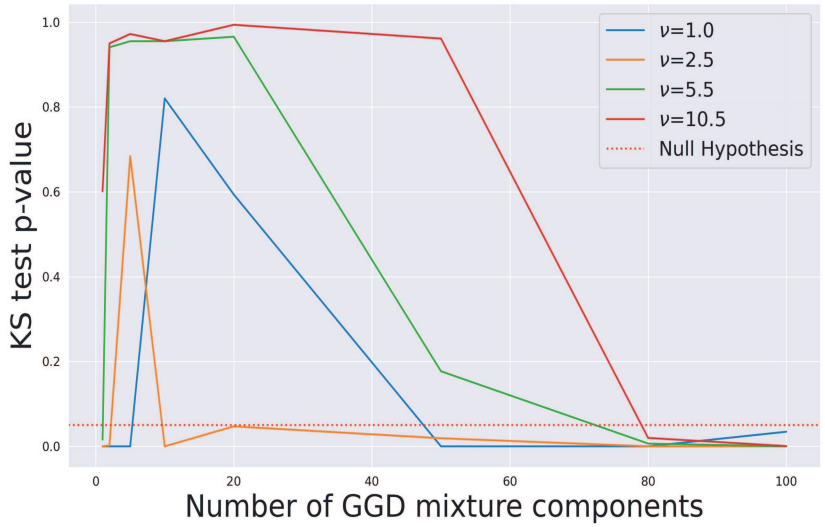


Figure 2-4: P-values of the two-sample Kolmogorov-Smirnov test on mixtures of generalized Gaussians with different number of components trained to estimate target distributions samples from a Student's t with different degrees of freedom.

2.4.2 Training Flows On Mixed-Tail Targets

In the previous section we proposed using a mixture of generalized Gaussians base density for modelling heavy and mixed-tail targets. We also argued that the flexibility to model the tail behavior should come from the flow itself and not be put entirely upon the base density. In this section we motivate and propose our method for making the training process of the flow model tail adaptive.

Challenges in Maximum Likelihood Training of Flows for Mixed-Tailed Targets Regardless of the choice for the base distribution and its tail properties, we argue that it is not enough to enable the flow to be trained properly and capture the tail behavior of the target distribution. The reason is training instability due to increased variance of the gradients when the target is heavy-tailed.

Consider the likelihood term in the maximum likelihood training objective (Equation (1)). Let $\boldsymbol{u} = T^{-1}(\mathbf{x}; \boldsymbol{\phi})$ be the inverse transformation of the observation $\mathbf{x}$. The term $\log p_{\mathrm{u}}(\boldsymbol{u})$ in the likelihood involves the logarithm of the base density evaluated at

$\boldsymbol{u}$. In the presence of heavy tails, the density can have slower-than-exponential decay, resulting in higher likelihood contributions from extreme values of $\boldsymbol{u}$ (and consequently, $\mathbf{x}$). Similarly, $\log|\det J_{T^{-1}}(\boldsymbol{u};\boldsymbol{\phi})|$ can become large in magnitude when $\boldsymbol{u}$ falls in the tails of the distribution since the Jacobian determinant accounts for the local expansion or contraction of the transformation and heavy tails can amplify this effect, leading to large values of the determinant.

We demonstrate this phenomenon in Figure 2-5, which depicts the instability of the standard maximum likelihood estimation in presence of heavy tails. We plot the gradients and the loss value of the same Real NVP normalizing flow [60] with Student's t base and target densities (following Jaini et al. [47]). We provide the results for three experiments: one experiment with base and target densities having the same tail behavior (both densities having degree of freedom of 10, labelled "10.0->10.0"), and two experiments with base density mis-specification, with base density being heavier tailed than the target ("2.0->10.0") and target density being heavier tailed than the base density ("10.0->2.0"). For each case, we plot the mean and the geometric median of the loss and its gradients. According to [47], as long as the base density is at least as heavy-tailed as the target density, the flow should theoretically be able to capture its tail behavior. However, as can be seen in the plot, regardless of whether the model is mis-specified or not, the loss function and its gradients are impacted by the heavy-tailed minibatches, resulting in a biased estimation of expectation. This bias is observed through the difference between the mean and geometric median values..

Our Proposed Robust Monte Carlo Maximum Likelihood-Based Training

To mitigate the problem described in previous section, we propose using robust estimation methods during the maximum likelihood estimation. Specifically, we propose employing the *Geometric Median (GM)*:

$$\bar{\mathbf{g}} = \arg\min_{\mathbf{g}} \sum_{b=1}^{B} \|\mathbf{g} - \nabla_\theta L(\theta; \mathbf{x}_b)\|_2 , \quad \mathbf{g} \in \mathbb{R}^d \tag{2.1}$$

during the Monte Carlo estimation of the likelihood while training the flow to mitigate the problem of high variance gradients when the target is heavy or mixed-tailed.

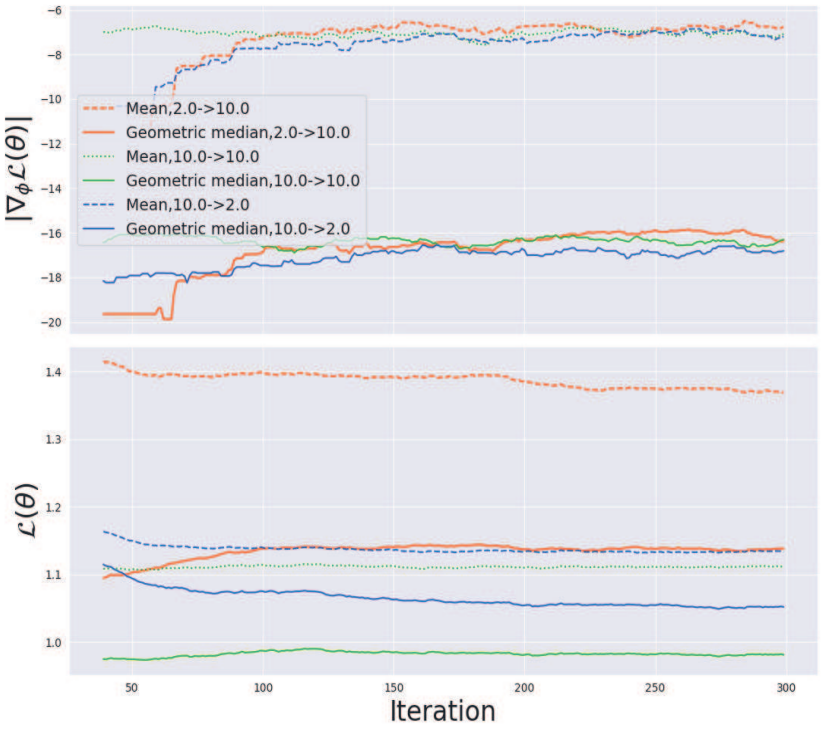


Figure 2-5: The impact of heavy-tailed target and base on the maximum likelihood estimation. This graph shows the mean and geometric median of loss and gradient norms a Real NVP flow trained with Student's t base density on a Student's t target, with and without mis-specification of the base with respect to the target.

To motivate this, we pay attention to mean estimation in presence of heavy-tailed data. Let $X_1, X_2, \ldots, X_n$ be i.i.d. random variables with a heavy-tailed distribution. Let $\bar{X}_n = \frac{1}{n}\sum_{i=1}^{n} X_i$ be the sample mean and X_{med} be the sample geometric median. Then $\text{Var}(\bar{X}_n) \gg \text{Var}(X_{med})$. For heavy-tailed distributions, the variance can be arbitrarily large. In fact, by definition of heavy tails, the tails of the distribution decay slower than an exponential distribution. Therefore, for any finite V[5], there exists some x_0 such that:

$$\int_{|x|>x_0} f(x)dx > \frac{V}{n}$$

[5] V represents a threshold value used to demonstrate the tailedness of a heavy-tailed distributions in terms of its influence on the tail probabilities. No matter how large V is chosen to be, it is always possible to find a point x_0 beyond which the integral of the probability density function $f(x)$ over the region where $|x| > x_0$ exceeds $\frac{V}{n}$. This essentially illustrates the concept that heavy-tailed distributions can have significant probability mass far into their tails.

Where $f(x)$ is the probability density function. This means there is non-negligible probability mass in the heavy tails that can lead to extremely large values of X_i. These extreme values disproportionately increase the variance of the sample mean $\bar{X}_n$. On the other hand, the geometric median only depends on the relative ordering of the X_i values, not their magnitudes. Therefore, it has robust variance in the presence of heavy tails.

Based on this observation we posit that using the geometric median estimator for gradients instead of the sample mean when training normalizing flows on mixed-tailed data reduces the variance and leads to more stable optimization. From the training objective, the gradient is:

$$\nabla_\theta \mathcal{L}(\theta) = \mathbb{E}_{p_\mathbf{z}}[\nabla_\theta \log p_\mathbf{x}(f_\theta(\mathbf{z}))]$$

This expectation is typically approximated with Monte Carlo sampling:

$$\nabla_\theta \mathcal{L}(\theta) \approx \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta \log p_\mathbf{x}(f_\theta(\mathbf{z}_i))$$

We already established that when $p_\mathbf{x}$ is heavy-tailed, the gradients $\nabla_\theta \log p_\mathbf{x}(f_\theta(\mathbf{z}))$ have very high variance. Therefore, using the geometric median instead of the sample mean to aggregate the gradients reduces the variance while still being a consistent estimator. This results in more stable optimization. Robust estimators like the geometric median can mitigate the training instability induced by mixed-tailed target distributions for normalizing flows.

Robust gradient estimators have been similarly employed by Hsu and Sabato [61] for example. We employ the Weiszfeld method [62] to estimate the geometric median.

2.5 EXPERIMENTS

We experiment using a range of simulated and tabular data. We train Real NVP (RNVP [60] flows with GGD mixture base density (100 components, learnable parameters) and robust MLE. The choice of RNVP is motivated by it being a Lipschitz-continuous function, which results in the density function has the same tail properties as its base.

The shift and scale operations of RNVPs are modelled by an feed forward network with one hidden layer the width of 1024. Each flow model is 12 steps deep. Minibatch size is 1024 and we perform Adam optimization ($lr = 10^{-5}$) All trainings are over 1000 iterations. We perform experiments in this section 5 times - unless explicitly mentioned - and report the mean and standard error.

2.5.1 Data Quality metrics

To measure the performance of our proposed method we perform experiments on synthetic and real data and report the negative log likelihood. Also following [1] we report the tail-specific performance metrics of Tail Value at Risk (tVaR) and Area Under Log-Log Plot (AULLP)[6].

Average Negative Log Likelihood The Negative Log Likelihood (NLL)[7] loss measures the model's ability to approximate the true data distribution. The negative log-likelihood of the model given the dataset $\mathcal{D}$ is defined as:

$$\mathcal{L}(\theta; \mathcal{D}) = -\sum_{i=1}^{N} \log p_\theta(\mathbf{x}^{(i)})$$

Tail Value at Risk Tail Value at Risk (tVaR) is a measure used to estimate expected losses beyond a certain threshold, known as the Value at Risk (VaR). It calculates the average loss above the VaR level based on a given quantile. The tVaR is obtained by taking the conditional expectation of the loss variable, given that the loss exceeds the VaR level. The VaR itself is determined as the minimum value of the loss variable for which the cumulative distribution function exceeds the specified quantile level.

[6]It is important to note that there are alternative approaches, such as conditional tail expectation (CTE) and extreme value theory (EVT), that can also be used to evaluate tail behavior. However, our choice of these metrics was motivated by their ability to provide a thorough indication of how well our approach captures the tail behavior of the target and to make our results comparable with the other prominent tail-adaptive normalizing flow method of Laszkiewicz et al. [1].

[7]The NLL is commonly used as a performance metric to train normalizing flows because it directly aligns with the objective of modeling the underlying data distribution. The NLL quantifies the discrepancy between the true data distribution and the distribution estimated by the normalizing flow model. It measures the probability assigned by the model to each observed data point.

Formally, the tVaR at a quantile level α is an expected loss measure for losses exceeding the Value at Risk (VaR) at the same level, defined as:

$$\text{tVaR}_\alpha(X) = \mathbb{E}[X \mid X > \text{VaR}_\alpha(X)]$$

with X being the loss variable and $\text{VaR}_\alpha(X)$:

$$\text{VaR}_\alpha(X) = \inf\{x \in \mathbb{R} : F(x) > \alpha\}$$

where $F(x)$ is X's cumulative distribution function.

The tVaR-difference metric, the absolute difference between empirical (tVaR_α^{emp}) and model-generated ($\text{tVaR}_\alpha^{model}$) tVaRs:

$$\text{tVaR-diff}_\alpha = \left|\text{tVaR}_\alpha^{emp} - \text{tVaR}_\alpha^{model}\right|$$

evaluates a generative model's tail capture capability. Smaller tVaR-difference implies better model performance in mimicking the distribution's extreme quantiles. We report the tVaR difference between test dataset and synthetic dataset.

Area Under Log-Log Plot The Area Under Log-Log Plot (AULLP) involves integrating the logarithm of the survival function, which represents the probability of a random variable exceeding a given value, in log-log space beyond a high quantile threshold. We report the The AULLP difference between test dataset and synthetic dataset.

The Area under the log-log plot (AULLP) measures a model's ability to capture the far distribution tail, emphasizing extreme values more than the Tail Value at Risk (tVaR). It involves integrating the survival function ($S(x) = 1 - F(x)$), which represents the probability of a random variable X exceeding a value x, in log-log space beyond a high quantile threshold x_α (e.g., 95th or 99th percentile):

$$\mathrm{AULLP}(X) = \int_{\log x_\alpha}^{\infty} \log S(x)\, d(\log x)$$

To gauge generative models' tail capture accuracy, the AULLP difference (AULLP-diff) between the empirical training data distribution and the model-generated distribution is calculated:

$$\text{AULLP-diff} = \left|\mathrm{AULLP}^{\mathrm{emp}}(X) - \mathrm{AULLP}^{\mathrm{model}}(X)\right|$$

$\mathrm{AULLP}^{\mathrm{emp}}(X)$ is the empirical distribution's AULLP, and $\mathrm{AULLP}^{\mathrm{model}}(X)$ is for the model-generated distribution.

2.5.2 2D Dataset

Data We use a synthetic datasets to demonstrate the performance of our proposed method. For our mixed-tail dataset, following [47] we choose our target distribution as a bi-variate Neal's funnel:

$$\mathbf{x}_i = \begin{cases} x_{i,1} \sim \mathrm{N}(\gamma, 1), \\ \\ x_{i,2} \sim \mathrm{N}(0,\ \exp\{x_{i,1}/2 + \lambda\}). \end{cases}$$

To better control the tail behavior of the distribution, we add an offset to the variance of the second variable via parameter $\lambda \in \mathbb{R}^{\geq 0}$.

Results The density plots illustrated in Figure 2-6 provide a visual comparison between the performance of our proposed method employing a Real NVP flow and several baseline methods utilizing different base densities: multivariate Gaussian, Student's t-distribution, and generalized Gaussian distributions and their mixtures. Our method demonstrates superior performance in terms of fitting the data more accurately compared to the baseline methods. This is particularly evident where the alternative models with Gaussian, Student's t, or generalized Gaussian base densities appear to be sub-optimal, either failing to adequately model one or both of the variables under con-

sideration.

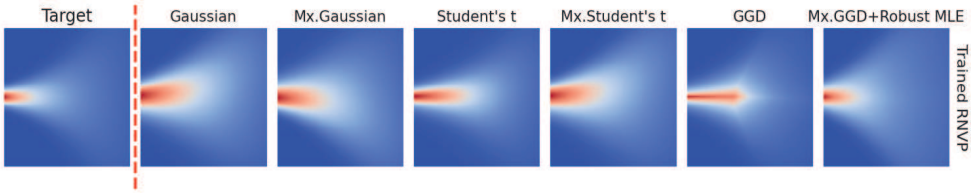


Figure 2-6: Performance of RNVP flow with different base distributions for estimation of a mixed-tail target. Our modified Neal's funnel with $\lambda = 6.0$ has a heavier-tailed and a lighter-tailed variable.

Table 2.1 shows performance of an RNVP with difference base densities. Results reported here show the advantage of our proposed method. We observe a meaningful advantage for our method, especially for the heavier-tailed variable (v2). Of note is the high variance of the flow with Student's t base (following [47]) which as discussed in section 2.3 is due to attenuated tail misspecification in the presence of mixed-tailed targets. The mixture of Gaussian base reports better results for the lighter tailed variable (4.58 vs 6.1 for AULLP, 54.5 vs 73.72 for tVaR) which is to be expected since v_1 is a standard Gaussian and has the same normal tail behavior as the flow base. As can be observed in the next section, for targets with more complicated tail behavior, the mixture of Gaussian base loses its advantage.

Table 2.1: General performance metrics for the Neal's funnel for the average NLL as well as AULLP difference and tVaR difference for each variable separately.

Base Density	NLL ↓	AULLP$_{v1}$↓	AULLP$_{v2}$↓	tVaR$_{v1}$↓	tVaR$_{v2}$↓
Gaussian	3.93 ± 0.22	8.15 ± 0.7	18.09 ± 1.14	$2.4e4 \pm 2e2$	$1.1e5 \pm 7.6e3$
Mx.Gaussian	3.86 ± 0.2	$\mathbf{4.58 \pm 0.19}$	11.15 ± 0.56	$\mathbf{54.5 \pm 2.6}$	36.45 ± 1.72
Student's t	3.86 ± 0.39	89.9 ± 10.9	93.6 ± 11.6	$2e30 \pm 4e27$	$2e31 \pm 9e28$
Mx.Student's t	3.88 ± 0.2	5.73 ± 0.53	14.15 ± 0.89	$8.8e3 \pm 1e1$	$8.8e5 \pm 9.2e3$
GGD	3.83 ± 0.31	7.09 ± 0.45	13.93 ± 1.19	81.1 ± 7.61	161.17 ± 8.32
Our Method	$\mathbf{3.77 \pm 0.31}$	6.1 ± 0.35	$\mathbf{5.81 \pm 0.49}$	73.72 ± 3.73	$\mathbf{33.0 \pm 2.62}$

2.5.3 Ablation Study

Data To measure the impact of each aspect of our method, we perform an ablation study. We use two base distributions, a GGD and a mixture of GGDs. We train an RNVP model on both bases with and without robust MLE. For each base, we train

flows on a two-dimensional distribution $\mathbb{X} = (\mathbb{X}_1, \mathbb{X}_2)$ such that $\mathbb{X}_1$ follows a generalized Gaussian distribution, and $\mathbb{X}_2$ adheres to a Student's t-distribution. $\mathbb{X}$ is parameterized by γ. The shape parameter of $\mathbb{X}_1$ is transformed by γ via a skew-adjusted Gamma cumulative distribution function, while $\mathbb{X}_2$'s degrees of freedom will be retrieved by $1+\gamma$. The parameter γ is explored within the interval [0, 1000] using a non-exhaustive, selective sampling strategy. Our setup enables us to test our method for a range of distributions with extreme tail behaviors, on one end (γ=0) having a heavy-tailed Cauchy and a close to uniform short-tailed GGD and at the other end of the spectrum ending up with a shorter-tailed Student's t and a Laplace distribution. We train our ablation models for each chosen γ, and subsequently report the AULLP-diff of two samples from the target and the trained model, each with size 50000.

Ablation Framework Figure 2-8 shows the β and ν parameters of the random variable we use for our ablation study as a function of γ (as shown in the two plots on the right). Furthermore, we provide a visual representation of the probability density functions (pdf) for $\mathbb{X}_1$ and $\mathbb{X}_2$ at the extremities of the γ spectrum, namely when $\gamma = 0$ and $\gamma = 1000$ (illustrated in the two plots on the left).

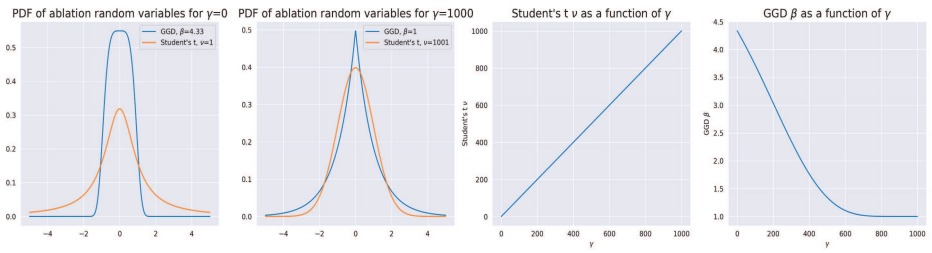


Figure 2-8: Ablation study framework. The two graphs on the left depict the probability density functions (pdf) of our dual random variables at the extreme points of γ. The pair of graphs on the right illustrate how the shape parameters ν for the Student's t-distribution and β for the GGD vary as functions of γ.

Results The results of our ablation study depicted in Figure 2-9 show that the flow with mixture of GGD base outperform single GGD in terms of lower AULLP difference across diverse γ settings, underscoring the versatility of the mixture in capturing tail behavior of our mixed tail target. Further, robust gradient estimation generally improves model fidelity, although its efficacy varies with the tail behavior of the distributions due

to the stochasticity of the models - since we do not optimize hyperparameters for different cases individually. These insights reveal that especially for heavier tailed target densities (variable 2) both the GGD mixture base and the robust gradient estimation are needed for the flow to be able to model the tail behavior of the target density accurately.

Table 2.2: Comparison of NLL, AULLP difference and tVaR difference for our proposed method applied to a RNVP in comparison with the same model with different base distributions and CTGAN.
*The tVaR tends to grow exponentially when the tail is heavily underestimated

	GAS			HEPMASS			POWER			MINIBOONE		
RNVP +	NLL↓	AULLP↓	tVaR↓	NLL↓	AULLP↓	tVaR↓	NLL↓	AULLP↓	tVaR↓	NLL↓	AULLP↓	tVaR↓
Gaussian base	8.86 ± 0.56	16.57 ± 1.32	0.74 ± 0.05	10.41 ± 0.64	3.38 ± 0.22	$1.5e2 \pm 9.93$	7.8 ± 0.47	9.28 ± 0.48	2.02 ± 0.16	56.29 ± 4.75	1.34 ± 0.10	67.29 ± 4.07
Mx.Gaussian base	8.01 ± 0.36	16.51 ± 1.06	0.72 ± 0.04	10.33 ± 0.59	3.21 ± 0.20	1.21 ± 0.08	7.78 ± 0.52	9.12 ± 0.64	1.79 ± 0.10	52.97 ± 3.58	$\mathbf{1.01 \pm 0.05}$	65.91 ± 2.72
Student's t base	9.13 ± 1.53	24.6 ± 2.74	100.5 ± 10.14	10.19 ± 1.38	10.07 ± 1.19	$2.1e9 \pm 3e8^*$	10.33 ± 1.07	21.45 ± 2.28	$2.7e6 \pm 2.4e5$	61.63 ± 6.09	2.05 ± 0.28	$1.6e3 \pm 2.5e2$
Mx.Student's t base	7.9 ± 0.53	16.6 ± 1.01	0.72 ± 0.06	10.2 ± 0.83	3.34 ± 0.27	3.38 ± 0.29	7.82 ± 0.50	9.21 ± 0.75	1.73 ± 0.17	54.75 ± 5.02	1.25 ± 0.09	66.32 ± 5.36
GGD base	9.02 ± 0.82	17.5 ± 1.96	0.69 ± 0.08	10.41 ± 0.84	3.38 ± 0.39	$1.5e2 \pm 11.76$	8.45 ± 0.89	9.64 ± 0.56	2.1 ± 0.13	65.39 ± 6.67	4.37 ± 0.27	72.17 ± 6.45
Mx.GGD + Robust MLE	$\mathbf{7.22 \pm 0.54}$	$\mathbf{15.4 \pm 1.67}$	$\mathbf{0.44 \pm 0.03}$	$\mathbf{8.17 \pm 0.70}$	$\mathbf{3.18 \pm 0.30}$	$\mathbf{1.12 \pm 0.09}$	$\mathbf{6.02 \pm 0.68}$	$\mathbf{8.64 \pm 0.76}$	$\mathbf{1.59 \pm 0.15}$	$\mathbf{49.68 \pm 3.74}$	1.19 ± 0.10	$\mathbf{63.39 \pm 6.77}$
CTGAN	-	17.02 ± 1.35	0.49 ± 0.04	-	3.17 ± 0.14	$\mathbf{1.08 \pm 0.08}$	-	8.8 ± 0.57	1.73 ± 0.10	-	2.47 ± 0.18	69.74 ± 2.98

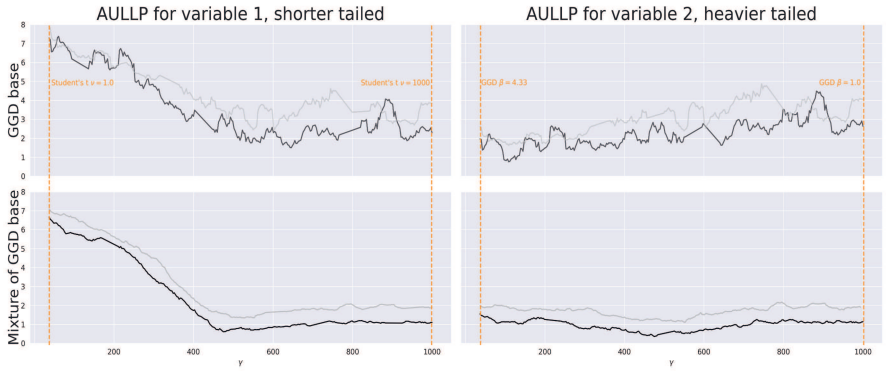


Figure 2-9: Ablation study results. Two different base distributions are plotted on different rows. Variables $\mathbb{X}_1$ and $\mathbb{X}_2$ are plotted on different columns. The light gray lines show the results of models without robust gradient estimation and the dark gray lines show the results with robust gradient estimation.

2.5.4 Performance Comparison with Pre-Hoc Tail Estimation Methods

In this section we report the performance of our method on a synthetic experiment from [1] to compare the performance of our method to the state of the art in pre-hoc tail estimation-dependent models (Section 2.3.2).

Data The target dataset is generated by a 8-dimensional Gaussian copula. The marginals of the copula consist of two Gaussians, followed by a 2-mixture and a 3-mixture of Gaussians. The last four marginals are a mixture of two t-distributions with $\nu = 2$. Mixtures have randomized means and variances and equal weights and the correlation matrix is randomized. More details can be found in Appendix C.2 of [1].

Results Table 2.3 compares the performance of our method with that of [1]. We report the results of training our model on an RNVP - with the same architecture used in previous experiments - with the same setup as per Section 4.1 of [1]. Results of [1] are copied from the $d_h = 4$ section of their Table 1. The metrics are reported for heavy tailed and light tailed components separately (denoted by $_h$ and $_l$ subscripts). We observe that our method performs better in case of heavy-tailed components while Laszkiewicz et al. [1]'s mTAF reports better results for light-tailed components. We attribute our method's lower performance in case of light tails to mis-specification of tails since Laszkiewicz et al. [1] use Student's t base marginals with the same degree of freedom as the synthetic example - $\nu = 2$ - for heavy-tailed target components. This pattern is consistent with results from Section 2.5.2.

Table 2.3: Comparison with [1]. We run the same experiment as [1]'s Table 1 experiment for $d_h = 4$. We repeat our experiment 25 times and report the average results and standard error.

Method	NLL ↑	AULLP-diff$_l$↓	AULLP-diff$_h$↓	tVaR-diff$_l$↓	tVaR-diff$_h$↓
TAF	-8.69	0.42	4.05	0.89	4.36
mTAF	-8.55	**0.25**	2.6	**0.57**	6.74
gTAF	-8.57	0.5	3.38	0.98	5.55
Ours	$\mathbf{-9.03 \pm 0.08}$	0.26 ± 0.03	$\mathbf{1.53 \pm 0.21}$	0.61 ± 0.07	$\mathbf{3.8 \pm 0.49}$

2.5.5 Tabular Datasets

Data To test our method in a more realistic setting, we train our model on four real world tabular datasets from the UCI repository: .POWER [63], GAS [64], HEPMASS [65] and MINIBOONE [66]. These datasets are commonly used in machine learning research, particularly in the context of generative modeling. Each of these datasets has unique characteristics that make them suitable for testing the capabilities of generative

models, especially in capturing complex and rare event distributions.

For example, MINIBOONE contains a mixture of background (noise) events and signal (neutrino events), where the signal events are relatively rare compared to the background. This characteristic makes it an excellent testbed for models that need to identify rare events within a predominantly noise-like dataset. Another example is HEPMASS, which includes a large amount of simulated data representing both background noise and signals of hypothetical particles, similar to Miniboone in terms of having a clear distinction between the 'normal' and 'anomalous' events.

In this context, the chosen datasets closely resemble health-related datasets where rare diseases are clearly distinguishable from normal conditions or fraud detection in finance or cybersecurity. This similarity allows us to demonstrate the effectiveness of our model in identifying rare events without the ethical and privacy concerns often associated with medical data. This capability extends the applicability of our findings beyond the experimental datasets to real-world scenarios where detecting rare occurrences is crucial.

We follow the same preparation process as Papamakarios et al. [67] for each dataset (More details in the Appendix Section D of [67]). We train a Real NVP flow with different base densities as well as our own method. We report negative log likelihood and mean AULLP difference and tVaR difference over all columns.

Results In Table 2.2, we present a comprehensive evaluation of our proposed approach for an RNVP normalizing flow. The results are contrasted against various base distributions with trainable parameters. We report average tVaR and AULLP over all variables. Our method's superiority is evidenced by the lowest NLL and tVar for all datasets. We report lowest AULLPs for all datasets except MINIBOONE (1.19 vs 1.01). Even in this case the difference is relatively small. Notably, the stark tVaR discrepancy, especially in HEPMASS with Student's t base (2.1e9 vs 1.12), underscores the challenge of tail estimation which our method effectively addresses. We also report results for CTGAN, which is one of the highest performing tabular data synthesizers to add context to the numbers we are reporting. Our method reports superior results across

the board except for tVar in HEPMASS dataset (1.08 vs 1.12), again with a very small margin. This comparison showcases the robustness of our methodology for capturing the tail behavior of the mixed-tail targets. Also of note is the inferior performance of the Student's t and GGD bases. Considering these results alongside results from section 2.5.2 we conclude that while the Gaussian base maintains a fixed tail behavior and produces average results dependant on the tail behavior of the target distribution and its pathologies, since both Student's t and GGD's tail behavior is parameterized and trainable, the flow either pushes them to underestimate the tail - violating the requirements set by Jaini et al. [47] - or overestimate the tail, leading to generation of samples with extreme values which will results in subpar performance in tail-specific metrics.

Table 2.4: Comparison of NLL, AULLP difference and tVaR difference for our proposed method applied to a MAF in comparison with the same model with different base distributions.

	GAS			HEPMASS			POWER			MINIBOONE		
MAF +	NLL↓	AULLP↓	tVaR↓	NLL↓	AULLP↓	tVaR↓	NLL↓	AULLP↓	tVaR↓	NLL↓	AULLP↓	tVaR↓
Gaussian base	-3.21 ± -0.23	18.07 ± 1.38	7.8 ± 0.70	-54.25 ± -2.99	13.42 ± 1.15	17.27 ± 1.42	2.37 ± 0.14	9.45 ± 0.72	3.39 ± 0.25	-97.12 ± -6.97	3.86 ± 0.26	7.84 ± 0.55
Mx.Gaussian base	-3.23 ± -0.19	18.05 ± 1.04	7.77 ± 0.47	-54.25 ± -2.39	13.34 ± 0.77	17.1 ± 1.12	2.37 ± 0.15	9.38 ± 0.61	3.32 ± 0.21	-97.12 ± -5.07	3.87 ± 0.19	$\mathbf{7.81 \pm 0.45}$
Student's t base	-1.72 ± -0.17	24.35 ± 2.78	17.03 ± 1.56	-27.55 ± -4.33	16.95 ± 2.84	36.23 ± 3.57	1.29 ± 0.12	15.8 ± 1.57	14.92 ± 2.06	-36.12 ± -3.83	12.9 ± 2.15	13.06 ± 1.93
Mx.Student's t base	-3.18 ± -0.29	18.17 ± 1.56	7.94 ± 0.76	-53.48 ± -4.36	13.42 ± 1.15	17.27 ± 1.22	3.81 ± 0.37	9.7 ± 0.64	3.11 ± 0.32	-95.57 ± -10.20	3.89 ± 0.25	7.11 ± 0.70
GGD base	-1.36 ± -0.16	27.79 ± 2.04	26.45 ± 2.04	-51.13 ± -5.89	13.68 ± 0.95	19.08 ± 1.91	$\mathbf{-1.1 \pm -0.09}$	13.47 ± 1.44	11.57 ± 1.25	-81.53 ± -10.37	17.37 ± 1.69	18.52 ± 2.40
Mx.GGD + Robust MLE	$\mathbf{-5.95 \pm -0.58}$	$\mathbf{15.19 \pm 1.44}$	$\mathbf{5.36 \pm 0.63}$	$\mathbf{-102.4 \pm -7.21}$	$\mathbf{9.94 \pm 1.08}$	$\mathbf{8.65 \pm 0.63}$	1.57 ± 0.14	$\mathbf{6.96 \pm 0.63}$	$\mathbf{1.47 \pm 0.10}$	$\mathbf{-96.13 \pm -6.77}$	$\mathbf{3.81 \pm 0.43}$	7.93 ± 0.95

We also investigate the efficacy of our method on a Masked Autoregressive Flow (MAF)[67] model. Similar to experiments in Section 2.5.5 we compare our approach against a variety of base distributions within a MAF model across four distinct datasets: GAS, HEPMASS, POWER, and MINIBOONE. The MAF has 12 autoregressive layers which use a stacked transformation with 1024 hidden units and 2 blocks to model the conditional dependencies of the variables. Our findings, summarized in Table 2.4, strongly suggest that integrating the GGD mixture base with robust MLE significantly improves the model's general performance - in terms of Negative Log Likelihood (NLL) - as well as its capabilities to capture tail behavior - as evident from metrics AULLP and tVaR, for which we report the difference.

Our proposed method achieves the lowest NLL for all datasets except POWER, and in this exception the margin of difference is quite small. This improvement is most pronounced in the HEPMASS dataset, where the NLL reached a remarkable -102.4. Additionally, our approach also yields the smallest AULLP and tVaR values in most datasets, highlighting its ability to more accurately capture tail distributions and overall

density estimation.

2.6 SELECTIVE GENERATION BY EXPLORING LATENT SPACE

Given our use of a tail-adaptive mixture to model the latent space of our flow model, and considering that each component of the mixture can exhibit a spectrum of tail behaviors, we posit that different individual components of the mixture – or subsets thereof – will, upon model convergence, be responsible for generating different parts of the target marginals relative to their tail behavior. In other words, some components will be responsible for generating the tail of the target distribution while alternate components are assigned to represent the regions of high probability density, typically centered around the distribution's mean. To capture extreme tails, we find that it is helpful for the components of the mixture base distribution to be spread-out. We can encourage this behavior by adding an extra term to the maximum likelihood objective:

$$\mathrm{R}(\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_K) = -\frac{1}{K} \sum_{k=1}^{K} ||\boldsymbol{\mu}_g - \boldsymbol{\mu}_k||_2 \tag{2.2}$$

where $\boldsymbol{\mu}_g$ denotes the geometric median of the component locations ($\boldsymbol{\mu}_k$). Since the means could, due to initialization or the stochastic nature of the optimization, have long tails, we calculate the geometric median of the means, μ_g and then calculate the sum of the l_2 norms of the means and the calculated medians as the total spread of the means. We add the R term to the model's loss function. The model is thereby rewarded during optimization by increasing the distance across the locations.

Our training algorithm including the regularization term will be:

The regularization term in our loss function incentivizes the model to push the means of the mixture modes apart, covering a larger part of the latest space - including the tail area - instead of being centered around certain areas. This by design leads to each sample mapped to the latent space having a small subset of mixture modes as its nearest neighbors. We leverage this aspect of our model for *Selective Generation.*

Algorithm 1 Training Algorithm for tail-adaptive normalizing flows with enhanced selective generation

Input: θ: flow, $\psi\{\mu, \Sigma, \beta, \pi\}$: base **for** *minibatch* $\mathbf{x}i$ **do**

$R(\boldsymbol{\mu}1, \ldots, \boldsymbol{\mu}K) = -\frac{1}{K}\sum k = 1^K |\boldsymbol{\mu}g - \boldsymbol{\mu}k|2$

$\mathcal{L} = \mathbb{E}\left[p^*(x_i)|p\mathrm{x}(\mathbf{x_i}; \boldsymbol{\theta})\right] + R(\boldsymbol{\mu}1, \ldots, \boldsymbol{\mu}K)$

$\mathbf{g}\mu = \arg\min \mathbf{g} \sum_{k=1}^{n} \|\mathbf{g} - \nabla_\theta L(\theta; \mu k)\|_2$

$\Delta\boldsymbol{\theta} \propto -\nabla\theta\mathcal{L}$

$\Delta\boldsymbol{\psi} \propto -\nabla_\psi\mathcal{L}$

end

To generate samples that are on the tail area of the distribution – or near its mean, or in case of a complex real world dataset in a specific area of the joint density of a subset of its variables – we can map one or few samples from that area to the latent space, find k nearest mixture modes to these samples – k being a predetermined number or the radius of a hypersphere around the geometric median of our subsample – and then generate our desired number of samples from these modes by sampling from them indiscriminately or by normalizing their weights to form a discrete mixing distribution and forming a separate mixture model.

The advantage of this method is especially evident for areas such as fairness where minority groups are defined in the joint tail section of several features of the dataset. We can upsample any subgroup within our population which has been shown to help mitigate some of the problems with fairness and more generally with performance of ML models by reducing class imbalanc. The advantage here is that we are not blindly exploring the latent space. Rather, we are systematically choosing the components in our base distribution with highest probability of being responsible for generating samples similar to our target sample or subsample and sample directly from those components to be pushed forward through the flow model and be transformed into our target.

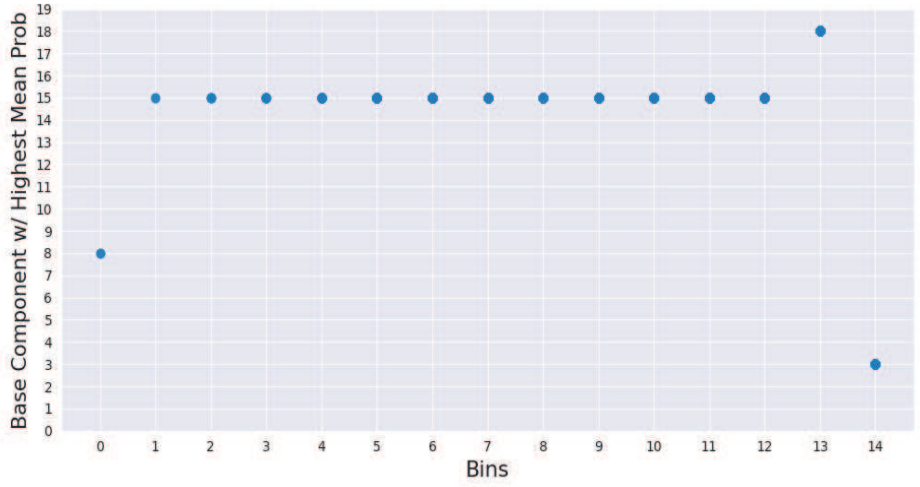


Figure 2-10: Selective generation from an RNVP model with a 20 component GGD mixture base for a 2d Student's t target.

2.6.1 Selective Generation in Practice

Figure 2-10 illustrates an example of our method's capabilities for selective generation. We train a real NVP using Algorithm 1 with a 20-component GGD mixture base[8] and robust MLE on a target dataset sampled from a two dimensional Student's t with ν=2. We then sample a dataset with the size of 100000 from the Student's t target and evaluate the probabilities of each sample on our target distribution. We bin the probabilities into 15 equal sized bins[9], with samples belonging to bins with lower probabilities assumed to be further down either tails of the distribution and vice versa. We then evaluated the samples of each bin on each of the 20 components of the training base mixture. The component that exhibits the highest average probability value for each bin is assigned as the primary generator for that bin. The samples have the highest average probability value when evaluated on that specific component. Therefore, if we sample from that component in the latent space and push the samples through the

[8]The number of component in the GGD mixture base is a hyperparameter which could be tuned based on the specifics of the problem. Here, we use 20 as our baseline. Reducing the number of components could result in not having a nuanced enough partitioning of the latent space. A larger number might result on mixture components being placed too close to each other in the latent space with several components being responsible for generating the samples from the same soft partition of the latent space while unnecessarily increasing the complexity of the model.

[9]A smaller number of bins would have resulted in a less nuanced examination of the different parts of the distribution with regards to its tail. A larger number would have resulted in visual clutter.

flow, we will predominantly yield instances that are belong to that specific bin. In Figure 2-10, for each bin the component with the highest mean probability values is marked. We note that component 0 is primarily responsible for generating instances for the bin with the lowest average probabilities (i.e., instances on the tail as evaluated on the target distribution itself), while the majority of the distribution mass is generated by components 3 and 18. Component 15 generates the remaining instances, notably those that are not extreme on the distribution's tail but are situated further from the distribution's mean. It should be noted that our target is a two dimensional target. However, to refrain from overcomplicating this analysis, we do not discriminate between the dimensions when binning the samples. If each marginal is considered separately, it will offer more nuanced control over the sample generation process.

In a scenario with a real world target dataset, we will not bin the samples, but rather select a representative sample of the type we want to generate from the target dataset. We then map these representative samples to the latent space. Once we have the mappings in the latent space, we could, for example, choose the top k components with the highest probability values for our selective generation and use a simple weighting scheme (e.g., using the average likelihood values or distance to the component mean), creating a sub-mixture for that particular subset of samples. This sub-mixture will then enable us to upsample our dataset through generation of samples similar to our representative samples.

2.7 Summary

In this chapter, we focused on enhancing the capability of normalizing flows to adaptively model the tail behavior of data distributions, which is crucial for generating synthetic data that accurately reflects real-world phenomena. We proposed a method that integrates generalized Gaussian mixtures as the base density with a robust maximum likelihood training algorithm, specifically designed to handle mixed-tailed data scenarios. This approach allows us to capture the diverse tail characteristics effectively, addressing the limitations of traditional methods that often fail to model tail behavior accurately.

CHAPTER
3

GENERATING SYNTHETIC TABULAR DATA WITH IMPROVED SEMANTIC INTEGRITY

RQ2: How can generative models synthesize data with high fidelity and diversity while respecting the implicit semantic boundaries of the real data?

3.1 Motivation

The utility of machine learning models often scales with the volume of training data available (e.g., see [68]). However, acquiring and utilizing such extensive datasets presents considerable challenges. Firstly, data availability can be severely restricted. Physical constraints, such as the geographical dispersion of data sources or their existence within isolated organizational silos, often impede the creation of large, centralized repositories. The logistical effort and cost associated with aggregating data from diverse, distributed systems can be prohibitive. Secondly, data privacy is a paramount concern. Data holders are frequently unable or unwilling to share their private datasets due to stringent regulatory frameworks (like GDPR or HIPAA), ethical considerations regarding sensitive information, or concerns about maintaining competitive advantages.

Synthetic data generation, as introduced by Rubin [69], offers a compelling strategy

to mitigate these interconnected problems. The core idea is to create a statistically representative proxy dataset that can be used for downstream analytical tasks without exposing the original, sensitive information. In modern approaches, this is typically achieved by training a generative model on the real data. This model learns the underlying probability distributions, correlations, and structural patterns present in the original dataset, effectively forming a compact, learned representation of it. Once trained, this generative model can produce an arbitrary number of new, artificial data samples. This methodology addresses the data access problem because the synthetic data, which does not contain direct records of real individuals, can often be shared and utilized with fewer restrictions than the original data. Consequently, privacy is preserved to a significant extent because the direct link to individual, real-world records is severed, and analysis is performed on the synthetic proxy rather than the original, sensitive data.

Within this context, our work focuses on the task of synthesizing high-fidelity tabular data. While achieving statistical similarity to the original data is a fundamental requirement, it is often insufficient. A critical challenge, particularly in complex domains, is ensuring that the generated synthetic data is also semantically correct. Semantic violations occur when synthetic records contain combinations of features that contradict established domain knowledge or plausible real-world constraints. These constraints can be explicit (e.g., age cannot be negative) or implicit (e.g., complex relationships between medical conditions and prescribed treatments that a domain expert would recognize as logical or illogical). Such violations, which we designate as hallucinations, represent scenarios that are not merely rare but are fundamentally implausible or logically inconsistent within the context of the domain. Preserving semantic integrity is therefore of utmost importance, especially in high-stakes applications like healthcare. For example, generating a synthetic medical record for a male patient who is documented as being pregnant is clearly unacceptable, as it represents an impossible scenario that would undermine the credibility and utility of any model trained on such data, potentially leading to erroneous conclusions or unsafe predictions.

To address the aforementioned challenge of ensuring semantic correctness in synthetic data, in this chapter we propose a novel framework termed **Latent Semantic Regularization (LSR)**. This framework conceptualizes the enhancement of semantic in-

tegrity as an unsupervised, regularized optimization problem. A core principle of LSR is its ability to unsupervisedly learn implicit semantic rules directly from the provided training data. These learned rules, which capture the inherent logical and contextual relationships within the data, are subsequently leveraged to guide the data synthesis process. The LSR architecture is characterized by two principal, yet distinct, components: a probabilistic latent-space *validator* and a separate *synthesizer*. The validator's role is to model the underlying semantic structure of the real data, effectively learning a representation of what constitutes a semantically plausible data instance. The synthesizer, in contrast, is primarily responsible for generating a diverse set of new data samples. The key mechanism of LSR involves the validator imposing learned, soft semantic bounds upon the latent space from which the synthesizer generates its samples. This interaction actively steers the synthesizer towards producing samples that exhibit improved semantic consistency, thereby reducing the occurrence of hallucinations.

In our work, we proceed to formalize the LSR framework and subsequently instantiate its components using well-established generative models. Specifically, we employ a Variational Autoencoder (VAE) to serve as the validator and a Generative Adversarial Network (GAN) to function as the synthesizer. This particular architectural choice is made to facilitate a clear investigation into the interplay between the validator's learned semantic representation and the synthesizer's sample generation process. This investigation is grounded within a well-understood probabilistic setting, allowing for a principled analysis of their interaction. The inherent objective of the GAN naturally encourages exploration of the data space, aiming for sample diversity. Concurrently, the VAE-based validator is designed to enforce the learned semantic constraints. This dual action provides a unified objective that inherently balances the critical trade-off between exploration (generating diverse and novel samples) and exploitation (adhering to learned semantic rules). It is important to emphasize that the LSR framework is, by design, architecturally agnostic. The instantiation with a VAE and GAN serves as a compelling proof-of-concept, demonstrating the framework's viability. The loosely-coupled design of LSR ensures that researchers can readily integrate various other types of latent variable models for either the validator or the synthesizer component, or both. This inherent flexibility offers researchers the adaptability required to tailor the LSR

framework to the specific needs and characteristics of diverse data synthesis tasks.

3.2 The Challenge of Semantic Integrity in Generative Models

Generative models are tasked with learning a probability distribution, denoted as $p_{\text{model}}(\mathbf{x})$, that serves as an approximation of the true underlying data distribution, $p_{\text{data}}(\mathbf{x})$. This learning process is typically facilitated by a generator function, $G_\theta(\mathbf{z})$, which is parameterized by θ. This generator maps latent variables $\mathbf{z}$, drawn from a prior distribution $p(\mathbf{z})$, to observable samples $\mathbf{x}'$. The primary goal of this mapping is to achieve statistical fidelity, ensuring that the learned distribution p_{model} closely mirrors the original data distribution p_{data}.

However, for synthesized data to be practically useful, particularly in critical applications, statistical fidelity alone is often insufficient. The generated data must also exhibit *semantic integrity*. This concept refers to the consistency of the synthesized samples with established domain-specific knowledge and inherent real-world constraints. Such constraints encompass a wide range, from explicitly defined physical laws or logical rules to more nuanced, implicitly understood contextual relationships. Semantic integrity, therefore, demands a level of contextual coherence that extends beyond mere statistical plausibility.

Standard training objectives for generative models, which predominantly focus on minimizing statistical divergence, may fail to adequately capture these semantic constraints. This is especially true for implicit constraints that are not directly or easily encoded within the observed data distribution $p_{\text{data}}(\mathbf{x})$. This failure can lead to the generation of samples that, while statistically plausible, are semantically nonsensical or violate fundamental domain principles; such instances are termed *hallucination*. This deficiency in capturing semantic integrity can arise from several factors. These include inherent limitations in the capacity of the generative model to accurately represent complex, high-dimensional data manifolds, biases introduced by the model architecture or training process, and the intrinsic incompleteness or noise present in the original training

data $p_{\text{data}}(\mathbf{x})$.

We approach the challenge of instilling semantic integrity by framing it as a constrained optimization problem. Let $C(\mathbf{x})$ represent the set of unknown or implicitly defined semantic constraints that a valid data sample $\mathbf{x}$ must satisfy. The objective, then, is to learn the parameters θ of the generator $G_\theta(\mathbf{z})$ such that these parameters maximize a measure of data fidelity, $\mathcal{L}_{\text{fidelity}}(\theta; p_{\text{data}})$. This maximization is performed subject to the condition that all generated samples $\mathbf{x}' = G_\theta(\mathbf{z})$ adhere to the specified semantic constraints $C(\mathbf{x}')$. This formulation is formally expressed as:

$$\begin{aligned} &\underset{\theta}{\text{maximize}} \quad \mathcal{L}_{\text{fidelity}}(\theta; p_{\text{data}}) \\ &\text{subject to} \quad C(G_\theta(\mathbf{z})) \text{ is satisfied, } \forall \mathbf{z} \sim p(\mathbf{z}). \end{aligned} \tag{3.1}$$

The constrained optimization problem presented in Equation 3.1 serves to underscore a fundamental tension inherent in the task of semantically sound synthetic data generation. This tension arises from the competing objectives of optimizing for statistical fidelity, which often encourages the model to closely replicate the observed data patterns, and simultaneously enforcing semantic constraint satisfaction, which may require the model to deviate from purely data-driven patterns to adhere to external or implicit rules. This situation necessitates a careful balancing act between two fundamental modes of operation for the generative model: exploration and exploitation. Exploration refers to the model's capacity to generate a diverse range of samples $\mathbf{x}'$, potentially uncovering novel or underrepresented configurations within the data space. Exploitation, in this context, pertains to the model's ability to adhere to the learned or imposed semantic constraints, thereby ensuring the semantic integrity of the generated samples $\mathbf{x}'$.

Figure 3-1 provides a visual illustration of this exploration-exploitation dilemma. The left panel of Figure 3-1 depicts two conceptual scenarios. The green contours represent a scenario of high exploitation, where a strong emphasis on semantic integrity leads the generative model to concentrate its probability mass within a narrow, semantically valid region. While this ensures that generated samples are largely consistent with domain knowledge, it may come at the cost of reduced diversity and potentially missing the tails or less common but still valid regions of the true data distribution. Conversely, the blue

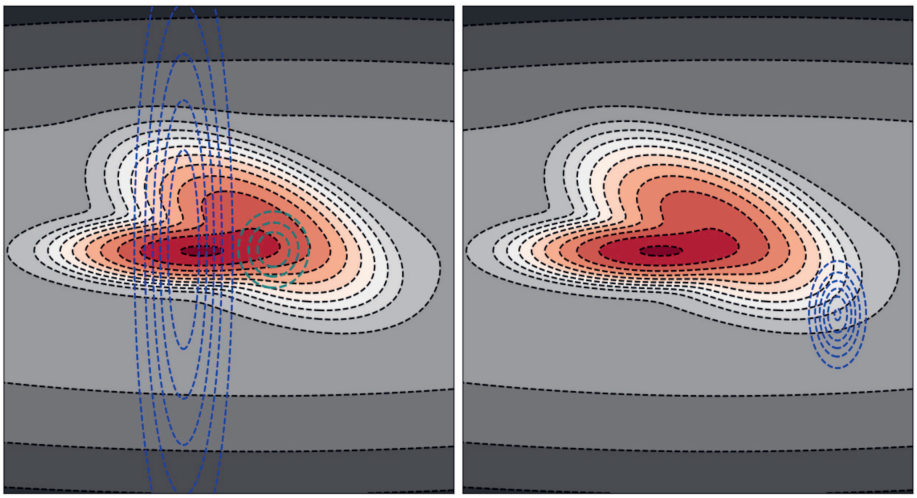


Figure 3-1: **The Explore/Exploit Dilemma in Semantically Constrained Synthesis.** ***Left***: *Green*: High exploitation (strong semantic integrity) concentrates mass, missing tails and diversity. *Blue*: High exploration (diversity) may venture into semantically invalid regions (hallucinations). ***Right***: Example empirical posterior density of generated samples mapped into the validator's latent space, showing how LSR might structure this space.

contours illustrate a scenario of high exploration. Here, the model prioritizes generating diverse samples, covering a broader area of the potential data space. However, this expansive exploration may lead the model to venture into regions that are semantically invalid, resulting in the generation of hallucinations, or samples that violate established constraints.

The right panel of Figure 3-1 offers a conceptualization of how Latent Semantic Regularization (LSR), the framework we propose, might address this dilemma. It shows an example of an empirical posterior density of generated samples that have been mapped into the validator's latent space. The structure of this density, potentially characterized by distinct modes and regions of varying probability, suggests how LSR could organize the latent space. By learning and imposing soft semantic bounds, LSR aims to guide the synthesizer towards regions of high semantic validity while still permitting sufficient exploration to capture the diversity of the underlying data. This structured latent space, shaped by the validator, is central to LSR's approach to balancing the competing demands of exploration for diversity and exploitation for semantic integrity.

In scenarios where a closed-form, explicit definition of semantic constraints $C(\mathbf{x})$ is

unavailable, the existing body of literature, as reviewed in Section 3.4, proposes various methodologies to address the problem of generating semantically valid synthetic data. A foundational, albeit simplistic, approach involves the assumption of access to an external constraint verification mechanism. This mechanism often takes the form of a probabilistic oracle, such as the one described in [70]. Such an oracle is designed to provide noisy, imperfect observations regarding the feasibility or semantic validity of given data samples. These observations are then used to inform and update a posterior distribution over what is considered the feasible or semantically valid region of the data space. This iterative process allows for a progressive refinement of a variational approximation towards the true, underlying posterior distribution of valid data.

An alternative strategy, pursued by other methods (e.g., [71]), relies on the availability of an existing mechanism capable of generating samples that are known to be semantically non-conforming or invalid. This collection of non-conforming samples then serves as a negative example set. This set provides a basis for developing a generative model that learns to adhere to unknown, predefined semantic constraints. This is typically achieved by refining the model's posterior distribution over the latent variables that govern the sample generation process, thereby improving its ability to generate valid designs while avoiding the characteristics of the non-conforming examples.

Our work, however, focuses on a more challenging, fully unsupervised setting. In this setting, we assume that the semantic constraints $C(\mathbf{x})$ are entirely implicit and that external tools such as explicit constraint definitions, pre-defined feasibility oracles, or curated collections of negative (invalid) samples are not available. This lack of explicit guidance makes the unsupervised learning and subsequent enforcement of semantic integrity a particularly difficult task, presenting several significant hurdles. Firstly, the implicit nature of $C(\mathbf{x})$ means that these constraints may resist direct quantification or formalization. Secondly, the manifold of valid data itself can possess intricate and complex boundaries or exhibit complex topological features. For instance, rare but semantically valid examples might correspond to the heavy tails of the data distribution, making them difficult to model and distinguish from outliers or true semantic violations. Lastly, the challenges associated with learning and enforcing these implicit constraints are often exacerbated by the curse of dimensionality, particularly when dealing with

high-dimensional tabular data.

To the best of our knowledge, much of the existing literature on semantic integrity in generative modeling tends to focus on specific application domains, such as image generation (e.g., [72]), or addresses semantic integrity implicitly as a secondary concern rather than formulating it as the primary research objective. In contrast, our work explicitly targets the unsupervised learning and enforcement of these implicit semantic constraints. We aim to achieve this within a generalizable framework, designed to robustly balance the dual objectives of generating diverse synthetic data and ensuring its semantic validity across a range of tabular data synthesis tasks.

3.3 Latent Semantic Regularization Framework

We introduce a general framework, termed **Latent Semantic Regularization (LSR)**, specifically designed to explicitly enhance the semantic integrity of synthetic data produced by latent variable models. The LSR framework directly addresses the challenge of generating semantically correct data, as outlined in Section 3.2, through a process of unsupervisedly learning implicit semantic constraints from the real training data. Once these underlying rules and relationships are learned, they are utilized to impose soft, probabilistic bounds upon the data generation mechanism. This regularization serves to steer the synthesis process away from generating semantically implausible samples, or hallucinations, while concurrently striving to preserve the generative diversity and statistical fidelity of the resulting synthetic dataset.

The architecture of the LSR framework is characterized by its modularity and a loosely-coupled interaction between its two principal components. These components are:

The Synthesizer (G_θ) This component is a generative model, parameterized by a set of parameters θ. The primary objective of the Synthesizer is twofold: first, to learn an approximation of the target data distribution $p_{\text{data}}(\mathbf{x})$, and second, to generate novel data samples $\mathbf{x}'$. These samples are produced by transforming latent variables $\mathbf{z}$, which are typically drawn from a simple, predefined prior distribution $p(\mathbf{z})$, through

the generator function, i.e., $\mathbf{x}' = G_\theta(\mathbf{z})$. The training of the Synthesizer is generally aimed at maximizing both the statistical fidelity of the generated samples (how well they resemble real data) and their diversity (the variety of samples it can produce). This aim is often formalized through an objective function, denoted as $\mathcal{L}_{\text{syn}}(\theta)$.

The Validator (V_ϕ) This component is also a latent variable model, parameterized by a distinct set of parameters ϕ. The Validator is trained exclusively on real data with the purpose of learning a structured latent representation, z', residing in a semantic latent space Z'. This latent representation is designed to capture the underlying semantic regularities and consistencies present in the input data x. In a typical instantiation, the Validator involves an encoder function, $E_\phi : X \to Z'$, which maps observed data points x from the original data space X to this semantic latent space Z'. Furthermore, the Validator defines a model of semantic validity within this latent space. This model is often grounded in, or represented by, a prior distribution $p_\eta(z')$ over the semantic latent variables z'. The parameters η that govern this prior distribution $p_\eta(z')$ may either be fixed beforehand or learned as part of the Validator's training process.

The core operational principle of the Latent Semantic Regularization (LSR) framework revolves around the interaction between its two primary components, the validator V_ϕ and the synthesizer G_θ. Specifically, the validator V_ϕ is tasked with providing a *semantic regularization signal* to the synthesizer G_θ. This signal serves as a quantitative measure, indicating the degree to which a synthesized sample x', generated by G_θ, conforms to the semantic structure that V_ϕ has learned from the real data.

This semantic regularization signal is then incorporated directly into the training objective of the synthesizer G_θ. By integrating this signal, the LSR framework effectively penalizes the generation of samples that are deemed semantically implausible by the validator V_ϕ. This penalty mechanism, in turn, guides the synthesizer G_θ during its learning process. It encourages G_θ to explore and favor regions of the data manifold that are not only statistically representative of the real data but also adhere to the learned semantic constraints. Consequently, the synthesizer is steered towards producing samples that simultaneously satisfy two criteria: firstly, statistical fidelity, which is typically measured by the synthesizer's own objective function $\mathcal{L}_{\text{syn}}$, and secondly,

semantic consistency, the adherence to which is assessed and quantified by the validator V_ϕ. This dual optimization aims to produce synthetic data that is both a good statistical match to the original data and semantically sound within the given domain.

3.3.1 Formalization of the LSR Framework

Let the synthesizer's objective be maximizing $\mathcal{L}_{\text{syn}}(\theta)$. Let the validator V_ϕ trained on $p_{\text{data}}(\mathbf{x})$ define a semantic consistency score for sample x. We define the *semantic regularization penalty*, $R_{\text{sem}}(x'; \phi, \eta)$, as a function that increases as the semantic consistency decreases. A natural choice, related to the negative log-likelihood under the prior, is:

$$R_{\text{sem}}(x'; \phi, \eta) = -\log p_\eta(E_\phi(x')) + K \tag{3.2}$$

where K is an arbitrary constant (often ignored) R_{sem} assigns higher penalties to samples x' whose latent representations $E_\phi(x')$ are atypical according to the learned semantic model $p_\eta(z')$. The overall LSR objective for training the synthesizer G_θ then becomes:

$$\underset{\theta}{\text{maximize}} \quad \mathcal{L}_{\text{syn}}(\theta) - \lambda \mathbb{E}_{\mathbf{z} \sim p(\mathbf{z})}[R_{\text{sem}}(G_\theta(z); \phi, \eta)] \tag{3.3}$$

where the hyperparameter $\lambda \geq 0$ controls the strength of the semantic regularization. This objective balances the synthesizer's goal $\mathcal{L}_{\text{syn}}$ with the imperative of semantic validity by enforcing the penalty term $\lambda \mathbb{E}[R_{\text{sem}}]$. This formulation transforms the constrained optimization problem of Equation 3.1 into an unconstrained, regularized optimization problem. The hard constraint $C(\mathbf{x}')$ is replaced by a soft, probabilistic penalty R_{sem} derived from the validator V_ϕ. This approach is particularly powerful when explicit semantic rules are unknown or hard to formalize, as it relies on the validator's ability to implicitly capture these regularities in its learned latent space Z' and prior $p_\eta(z')$.

The effectiveness of LSR hinges on the validator's capacity to learn a meaningful semantic representation and prior. A well-trained validator should assign high probabilityto latent codes z' corresponding to semantically plausible data configurations and low density to regions corresponding to implausible configurations as the regularization term penalizes hallucinations. Furthermore, modeling the tail behavior of $p_\eta(z')$ accurately

becomes crucial, especially for tabular data with complex dependencies and potentially rare but valid feature combinations, preventing the over-penalization of tail-bound samples.

Advantages of the Decoupled Mechanism of LSR The deliberate separation of the synthesizer (G_θ) and validator (V_ϕ) offers several advantages, drawing inspiration from principles in robust machine learning and ensemble methods [73].

A distinct validator acts as a more objective assessor of semantic integrity, as its parameters (ϕ, η) are optimized based on real data independently of the synthesizer's optimization process (θ). This also helps prevent a form of overfitting where the combined system might find degenerate solutions that satisfy the joint objective trivially without capturing the true data distribution or semantic rules.

The separation also allows each component to specialize through specific architectures for G_θ and V_ϕ. The synthesizer can focus on achieving high sample fidelity and diversity (often the strength of models like GANs), while the validator can focus on accurately modeling the data's density and semantic structure in its latent space (often a strength of models like VAEs or flows). This modularity also facilitates extensions, such as using multiple or conditional validators to enforce different types of semantic constraints on specific subsets of features or data instances, offering nuanced control over the generation process.

In the subsequent sections, we will instantiate this general LSR framework using specific, well-established generative models: a Variational Autoencoder (VAE) as the validator V_ϕ and a Generative Adversarial Network (GAN) as the synthesizer G_θ. This concrete instantiation will allow us to empirically demonstrate the effectiveness of LSR in improving semantic integrity for data synthesis.

3.3.2 Validator Instantiation

We instantiate the validator V_ϕ within the LSR framework via a Conditional Variational Autoencoder (CVAE) [74]. The principled probabilistic framework of VAEs makes them

particularly well-suited as the validator. The encoder approximates the true posterior $p(\mathbf{z}'|\mathbf{x}, \mathbf{y})$, and the model allows for approximate likelihood estimation of data points, modeling the data density in the latent space through the prior $p_\eta(\mathbf{z}'|\mathbf{y})$. This capability is crucial for assessing the semantic typicality of a given sample.

The VAE Objective and its Role in LSR The CVAE aims to maximize the conditional log-likelihood $\log p(\mathbf{x}|\mathbf{y})$ by maximizing the Evidence Lower Bound (ELBO) :

$$\begin{aligned}\mathcal{L}_{\text{ELBO}}(\phi, \psi, \eta; \mathbf{x}, \mathbf{y}) =& \mathbb{E}_{\mathbf{z}' \sim q_\phi(\mathbf{z}'|\mathbf{x}, \mathbf{y})}[\log p_\psi(\mathbf{x}|\mathbf{z}', \mathbf{y})] \\ &- KL\left[q_\phi(\mathbf{z}'|\mathbf{x}, \mathbf{y}) \| p_\eta(\mathbf{z}'|\mathbf{y})\right] \\ &\leq \log p(\mathbf{x}|\mathbf{y})\end{aligned} \tag{3.4}$$

where $q_\phi(\mathbf{z}'|\mathbf{x}, \mathbf{y})$ is the probabilistic encoder , $p_\psi(\mathbf{x}|\mathbf{z}', \mathbf{y})$ is the probabilistic decoder , and $p_\eta(\mathbf{z}'|\mathbf{y})$ is the prior distribution over the latent semantic space Z'. The KL term penalizes deviations between the approximate posterior $q_\phi(\mathbf{z}'|\mathbf{x}, \mathbf{y})$ for a given real sample $\mathbf{x}$ and the prior $p_\eta(\mathbf{z}'|\mathbf{y})$. By maximizing the ELBO over the training data, the VAE learns an encoder and prior such that the latent codes conform to the prior distribution. This prior $p_\eta(\mathbf{z}'|\mathbf{y})$ thus becomes an operational model of semantic validity within the latent space Z'.

Deriving the Semantic Regularization Signal from the VAE Recall the general LSR objective (Equation 3.3) requires a semantic regularization penalty $R_{\text{sem}}(x'; \phi, \eta)$. Using the VAE validator, we can directly derive this penalty from the learned semantic prior $p_\eta(\mathbf{z}'|\mathbf{y})$ and the encoder q_ϕ. For a generated sample x', we first encode it into the semantic latent space: $\mathbf{z}' \sim q_\phi(\mathbf{z}'|\mathbf{x}', \mathbf{y}')$ (where $\mathbf{y}'$ is conditioning info for x', if applicable). The measure of semantic inconsistency is how unlikely this latent code $\mathbf{z}'$ is under the learned prior $p_\eta(\mathbf{z}'|\mathbf{y}')$. Therefore, we define the LSR penalty as the expected negative log-likelihood under the validator prior:

$$R_{\text{sem}}^{\text{VAE}}(x'; \phi, \eta) = \mathbb{E}_{\mathbf{z}' \sim q_\phi(\mathbf{z}'|\mathbf{x}', \mathbf{y}')}[-\log p_\eta(\mathbf{z}'|\mathbf{y}')] \tag{3.5}$$

This term quantifies the KL divergence between the encoder output for the synthetic

sample x' and the learned semantic prior, up to an entropy term for $q_\phi(\mathbf{z}'|\mathbf{x}', \mathbf{y}')$ which is independent of the prior itself. We decompose the KL term from the ELBO (Equation 3.4):

$$\begin{aligned} & KL\left[q_\phi(\mathbf{z}'|\mathbf{x}', \mathbf{y}') \| p_\eta(\mathbf{z}'|\mathbf{y}')\right] \\ & = \mathbb{E}_{q_\phi(\mathbf{z}'|\mathbf{x}', \mathbf{y}')}[-\log p_\eta(\mathbf{z}'|\mathbf{y}')] - \mathbb{H}[q_\phi(\mathbf{z}'|\mathbf{x}', \mathbf{y}')] \\ & = R_{\text{sem}}^{\text{VAE}}(x'; \phi, \eta) - \mathbb{H}[q_\phi(\mathbf{z}'|\mathbf{x}', \mathbf{y}')] \end{aligned} \tag{3.6}$$

where $\mathbb{H}[\cdot]$ denotes the differential entropy. Minimizing this KL divergence for real data during VAE training encourages the alignment of $q_\phi(\mathbf{z}'|\mathbf{x}, \mathbf{y})$ with $p_\eta(\mathbf{z}'|\mathbf{y})$. Consequently, when the synthesizer generates an x' whose encoded distribution $q_\phi(\mathbf{z}'|\mathbf{x}', \mathbf{y}')$ deviates significantly from the learned prior $p_\eta(\mathbf{z}'|\mathbf{y}')$, the term $R_{\text{sem}}^{\text{VAE}}$ will be large, imposing a penalty via the LSR objective (Equation 3.3), imposing soft semantic bounds as deviations from the learned semantic structure incur a differentiable penalty.

Expressive Learnable Priors via Tail-Adaptive Normalizing Flows. Standard (C)VAE priors, typically fixed isotropic Gaussians, lack the capacity to model the potentially complex and heavy-tailed distributions $p_\eta(\mathbf{z}'|\mathbf{y})$ required to capture nuanced semantic structures in latent spaces Z'. To address this, we employ a learnable prior parameterized by a Normalizing Flow (NF) [60]. An NF transforms samples $\mathbf{z}'_0$ from a simple base distribution $p_0(\mathbf{z}'_0|\mathbf{y})$ via a sequence of invertible, differentiable functions $\mathbf{z}' = f_\eta(\mathbf{z}'_0)$, allowing the prior $p_\eta(\mathbf{z}'|\mathbf{y})$ to adapt during training.

Substituting this learnable prior into the VAE objective (Eq. 3.4) modifies the ELBO, requiring joint optimization of the flow parameters η alongside ϕ, ψ. The resulting objective, sampling $\mathbf{z}'_0 \sim p_0$, is:

$$\begin{aligned} \mathcal{L}_{\text{ELBO}}^{\text{NF}} = & \mathbb{E}_{\mathbf{z}'_0, \mathbf{z}'=f_\eta(\mathbf{z}'_0)}[\log p_\psi(\mathbf{x}|\mathbf{z}', \mathbf{y})] \\ & - KL\left[q_\phi(f_\eta(\mathbf{z}'_0)|\mathbf{x}, \mathbf{y}) \| p_0(\mathbf{z}'_0|\mathbf{y})\right] \\ & + \mathbb{E}_{\mathbf{z}'_0}[\log \left| \det \left(\frac{\partial f_\eta}{\partial \mathbf{z}'_0} \right) \right|]. \end{aligned} \tag{3.7}$$

Modeling the *tail behavior* of $p_\eta(\mathbf{z}'|\mathbf{y})$ is crucial for distinguishing rare but valid data configurations from semantically invalid ones (hallucinations). Standard NFs may struggle

here. We therefore specifically utilize a *tail-adaptive normalizing flow* [37] to instantiate f_η. This class of flows is explicitly designed to capture varying tail behaviors, enabling a more precise representation of semantic boundaries within Z'. Consequently, the derived semantic penalty $R_{\text{sem}}^{\text{VAE}}$ becomes more nuanced, appropriately penalizing semantically implausible samples without penalizing valid configurations in the learned tails, thus better balancing semantic integrity and diversity.

3.3.3 Synthesizer Instantiation

For the synthesizer component G_θ in our demonstration of the LSR framework, we employ a conditional Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP) [75], augmented with established techniques: an Auxiliary Classifier (ACGAN) objective [76, 77] and an Information Loss term [48], aimed at enhancing sample quality and conditional consistency.

Let the generator be $G_\theta : Z \times Y \to X$, mapping latent noise $\mathbf{z} \sim p(\mathbf{z})$ and conditioning information $\mathbf{y} \sim p(\mathbf{y})$ to the data space X. Let the critic be $D_\omega : X \times Y \to \mathbb{R}$. The critic D_ω is trained to estimate the Wasserstein distance, while the generator G_θ is trained to minimize this estimated distance and satisfy additional criteria detailed below.

Base Generator Objective (without LSR) The generator G_θ's original objective function designed to encourage fidelity, diversity, and conditional correctness is defined as:

$$\mathcal{L}_{G,\text{base}}(\theta) = \mathcal{L}_{G,\text{WGAN}}(\theta) + \lambda_{ac,g}\mathcal{L}_{G,\text{AC}}(\theta) + \lambda_{info}\mathcal{L}_i(\theta) \tag{3.8}$$

where $\mathcal{L}_{G,\text{WGAN}}(\theta)$ is the standard WGAN generator loss, $\mathcal{L}_{G,\text{AC}}(\theta)$ is an auxiliary classification loss computed on generated samples $\tilde{\mathbf{x}} = G_\theta(\mathbf{z}, \mathbf{y})$, penalizing inconsistencies between $\tilde{\mathbf{x}}$ and its conditioning $\mathbf{y}$ as classified by an auxiliary output of the critic D_ω, $\mathcal{L}_i(\theta)$ is an information loss term matching statistics of feature representations extracted from an intermediate layer of the critic D_ω for real and generated data batches, promoting feature-level similarity, and $\lambda_{ac,g} \geq 0$ and $\lambda_{info} \geq 0$ are hyperparameters weighting the auxiliary classifier and information loss contributions, respectively.

Integrating LSR: The Regularized Synthesizer Objective We now incorporate the semantic regularization signal derived from the pre-trained VAE validator by adding the semantic penalty $R_{\text{sem}}^{\text{VAE}}$ (Equation 3.5) to the generator's base objective. This penalty quantifies the inconsistency of a generated sample $\tilde{\mathbf{x}} = G_\theta(\mathbf{z}, \mathbf{y})$ with the validator's learned semantic prior $p_\eta(\mathbf{z}'|\mathbf{y})$, assessed via the validator's encoder $q_\phi(\mathbf{z}'|\tilde{\mathbf{x}}, \mathbf{y})$.

The final objective function for the synthesizer generator G_θ under the LSR framework becomes:

$$\mathcal{L}_G(\theta) = \mathcal{L}_{G,\text{base}}(\theta) + \lambda_{sem}\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z}),\mathbf{y}\sim p(\mathbf{y})}[R_{\text{sem}}^{\text{VAE}}(G_\theta(\mathbf{z}, \mathbf{y}), \mathbf{y}; \phi, \eta)] \tag{3.9}$$

where $\lambda_{sem} \geq 0$ controls the strength of the semantic regularization. The generator G_θ is trained to minimize $\mathcal{L}_G(\theta)$, while the critic D_ω minimizes its corresponding objective $\mathcal{L}_D(\omega)$. The validator parameters (ϕ, η) remain fixed during this phase.

This objective $\mathcal{L}_G(\theta)$ balances the generation of high-fidelitysamples (via $\mathcal{L}_{G,\text{base}}$) with the enforcement of semantic validity learned unsupervisedly by the validator (via $R_{\text{sem}}^{\text{VAE}}$), stabilized and enhanced by auxiliary terms ($\mathcal{L}_{G,\text{AC}}, \mathcal{L}_i$).

3.4 Previous Works

Implicit Constraints In the literature the integration of generative models to address specific limitations has been extensively explored. For example, VAEs have been shown to suffer from inability to learn marginal distributions [78]. Thus they have been paired with other generative models to overcome these issues, a popular choice among them being GANs. Dumoulin et al. [79] and Donahue et al. [80] combine GANs and VAEs to improve performance on discriminative tasks. Makhzani et al. [81] and Tolstikhin et al. [82] transform the KL term into a synthetic likelihood representation that can be imposed by a discriminator and Bhattacharyya et al. [83] propose a "Best-of-Many-Samples" reconstruction loss term to further improve on fixing the problem of mode collapse. Bao et al. [84] propose combining a GAN and a CVAE to have more fine-grained control over the sample generation process. However, none of these methods are aimed at explicitly defining and improving the semantic integrity of the

generated data. Furthermore, our dichotomous formulation of learning and imposition of semantic boundaries leads to a lot of flexibility in choice of the model for each component and the impact of the validator on the synthesizer, differentiating it from the existing methods.

Implicit Constraints for Improved Semantic Integrity While integration of explicit constraints has been previously studied [85], we consider the task of implicit latent constraints, where the constraints are unknown and need to be both learned and applied to the generative model. Park et al. [48] propose a tabular data generation method based on DCGAN add a classifier to the model, with its loss added to the generator objective function - alongside an information loss term - to increase semantic integrity of the generated records. However, they don't provide an examination of the semantic integrity gain of their method. In this work, we provide an examination of the semantic integrity of this method along a select group of widely cited and used methods for comparison. Both Zhao et al. [86] and Engelmann and Lessmann [87] also use auxiliary classifier to train a GAN for tabular data generation. However, none of these methods explicitly mention using the auxiliary classifier to elevate the semantic integrity of the generated samples.

Flow-Based Hybrid Models with Implicit Constraints Previous research has combined normalizing flows with VAEs for addressing discrete sequence modeling [88] and structured sequence prediction [89]. Our approach distinctively employs tail-adaptive flow to capture joint tail behavior of the validator prior, significantly impacting the synthesizer's semantic integrity.

Advances in Conditional GANs and Diversity Recent GAN advancements focus on diversity and conditional fidelity, primarily for images. DSCGAN [90] and MSGAN [91] encourage mode coverage and diversity. RCGAN [92] enhances robustness against noisy conditioning labels. HM-GAN [93] balances adversarial loss with maximum likelihood for multimodal conditional tasks. These differ fundamentally from our LSR approach, which specifically addresses *implicit semantic integrity* through a distinct,

probabilistically-grounded validator learning semantic constraints unsupervisedly and imposing them as soft regularizing bounds (Eq. 3.3) on a separate synthesizer.

3.5 Experiments

To evaluate the efficacy of the Latent Semantic Regularization (LSR) framework, we conduct a series of experiments. These experiments are designed with two primary objectives: first, to assess LSR's capability to enhance the semantic integrity of generated synthetic data and, concurrently, to reduce the occurrence of hallucinations; second, to quantify the impact of this semantic regularization on the overall utility and statistical fidelity of the synthesized data. For the purpose of these experiments, we instantiate the LSR framework utilizing specific, well-established generative models. The validator component is realized using a Conditional Variational Autoencoder (CVAE) that incorporates a Tail-Adaptive Normalizing Flow as its prior distribution. The synthesizer component is implemented as a conditional Wasserstein Generative Adversarial Network with Gradient Penalty (WGAN-GP). A detailed description of this particular instantiation is provided in Section 3.3. Henceforth, we denote this complete instantiation of our proposed framework as **LSR (S+V)**.

To specifically isolate and understand the contribution of the semantic regularization mechanism, which is the core novelty of LSR, we also include an ablation study in our experimental design. In this ablation, only the synthesizer component of our framework is utilized and trained. This synthesizer-only configuration, denoted as **LSR (S)**, is trained using solely its base objective function, as defined in Equation 3.8, without the semantic regularization signal from the validator. This allows for a direct comparison against the full LSR (S+V) model, thereby highlighting the impact of the validator's contribution.

For the purpose of comparative analysis, we select a set of established Generative Adversarial Network (GAN)-based models that are commonly used for tabular data synthesis. These include MedGAN [94], CTGAN [95], and TableGAN [48]. The selection of these particular baselines is a methodologically deliberate choice. Given that LSR is for-

mulated as a regularization framework and our specific instantiation utilizes a GAN as the synthesizer, these GAN-based baselines provide a suitable and controlled basis for evaluation. Comparing LSR against generative models from fundamentally different paradigms, such as diffusion models [96] or transformer-based architectures [97], would introduce confounding variables. Such a comparison would make it difficult to disentangle the assessment of the LSR mechanism itself from the effects of the distinct inductive biases and inherent generation processes characteristic of those alternative model classes.

To ensure a fair comparison, all models, including our LSR variants and the selected baselines, are trained on identical, stratified 80/20 data splits. We utilize the standard, publicly available implementations of the baseline models and adhere to their recommended hyperparameter settings. To account for stochasticity in the training process, all experiments are repeated five times. We report the average results across these repetitions, along with the corresponding standard errors, to provide a robust and statistically sound evaluation. The evaluation proceeds by first examining a low-dimensional illustrative example to visualize LSR's effect. Subsequently, we evaluate performance on real-world benchmark datasets using quantitative metrics.

The principal hyperparameter introduced by LSR is the semantic regularization weight, λ_{sem} (3.9), governing the trade-off between the synthesizer and the validator. For the presented experiments, λ_{sem} was fixed to 1.0 across all datasets, demonstrating the framework's utility *without* dataset-specific tuning of this parameter. Hyperparameters defining the aforementioned validator and synthesizer (network architectures, latent dimensions, optimizer settings, auxiliary loss weights) were adopted following standard practices and configurations used in the baseline implementations where applicable. Further investigation into λ_{sem} sensitivity remains future work.

3.5.1 Metrics

To facilitate a comprehensive assessment of the performance of the generative models under investigation, including our proposed Latent Semantic Regularization (LSR) framework, we employ a suite of established quantitative metrics. These metrics are

selected to cover two principal aspects of model performance: firstly, the statistical fidelity of the generated synthetic data in comparison to the original real data, and secondly, the utility of the synthetic data when used for downstream machine learning tasks. In addition to these established metrics, we also address the specific aspects of generative diversity (which relates to the model's capacity for exploration of the data space) and adherence to semantic constraints (which relates to the model's capacity for exploitation of learned semantic rules). As, to the best of our knowledge, there is no universally established assessment framework for these latter two aspects, particularly in the context of tabular data, we contribute two novel assessment methods specifically designed for a detailed evaluation of semantic quality.

Statistical Fidelity To quantify the statistical similarity between the marginal distributions of individual features in the real dataset and those in the generated synthetic dataset, we utilize distinct metrics appropriate for numerical and categorical features. For numerical features, we compute the Kolmogorov-Smirnov (KS) test statistic for each feature independently. The KS statistic measures the maximum distance between the empirical cumulative distribution functions (CDFs) of the real and synthetic versions of a feature. The average KS statistic across all numerical features is then reported. Lower values of the average KS statistic indicate a higher degree of similarity between the distributions of numerical features in the real and synthetic datasets. For categorical features, we employ the Log Chi-Squared test statistic. This test assesses the independence of the observed frequencies of categories in the synthetic data from those in the real data. We compute this statistic for each categorical feature and report the average value. In this case, lower values of the average Log Chi-Squared statistic are preferred, signifying a closer resemblance between the distributions of categorical features in the real and synthetic datasets.

Downstream Task Efficacy To evaluate the practical utility of the synthesized data, we assess its efficacy when used as training data for a standard downstream machine learning (ML) task. Following common practice in the evaluation of generative models for tabular data, as described in [95, 48], we select a classification task relevant to the

datasets being used. We train a standard classifier (e.g., a logistic regression model or a simple neural network) twice: once using only the real training data, and once using only the generated synthetic data. Both of these trained classifiers are then evaluated on a common, held-out test set, which consists of real data not seen during the training of either the generative model or the classifiers. The performance of each classifier on this test set is measured using the F1-score. The F1-score is the harmonic mean of precision and recall and provides a balanced measure of a classifier's performance, particularly in cases of class imbalance. Higher F1-scores for the classifier trained on synthetic data are indicative of greater utility. Ideally, the F1-score achieved by the model trained on synthetic data should approach the score achieved by the model trained on real data, demonstrating that the synthetic data can effectively substitute for real data in this downstream application.

Exploration Factor A critical characteristic of a proficient generative model is its capacity for generative diversity. This ensures that the synthesizer component adequately explores the underlying data manifold, thereby capturing the variety inherent in the true data distribution, rather than collapsing to a few dominant modes or merely replicating training instances. To quantify this aspect of exploration, we define a metric termed the Exploration Factor, denoted as S_d. The calculation of S_d proceeds as follows: for each sample r generated by the model, thus $r \in \mathcal{D}_g$ where $\mathcal{D}_g$ is the set of generated samples, we identify its nearest neighbor, $\hat{r}$, within the original, real dataset $\mathcal{D}_o$. The distance used for this nearest neighbor search is the Hamming distance. The Exploration Factor S_d is then defined as the sum of these minimum Hamming distances, aggregated over all generated samples $r \in \mathcal{D}_g$. Formally, if $d_H(r, \hat{r})$ represents the Hamming distance between a generated sample r and its nearest neighbor $\hat{r}$ in $\mathcal{D}_o$, then $S_d = \sum_{r \in \mathcal{D}_g} d_H(r, \text{argmin}_{\hat{r}' \in \mathcal{D}_o} d_H(r, \hat{r}'))$.

To ensure that this metric focuses on meaningful diversity rather than being unduly influenced by random noise or minor, inconsequential variations between samples, the Hamming distance calculation is adjusted. This adjustment involves a selective consideration of features. Specifically, the distance calculation is restricted to only those combinations of categorical attributes that exhibit a high degree of correlation in the

original dataset. These highly correlated attributes are identified by examining the correlation matrix ρ of the categorical features in $\mathcal{D}_o$. A pair of attributes (i, j) is considered highly correlated if their correlation coefficient ρ_{ij} exceeds a specified threshold. This threshold is dynamically set as $\bar{\rho} \cdot \alpha$, where $\bar{\rho}$ represents the mean correlation across all pairs of categorical attributes, and α is a scaling factor, which in our experiments is set to 100. By focusing the Hamming distance on these pre-identified, strongly related attribute combinations, we aim to measure diversity in terms of structurally significant variations rather than isolated, potentially noisy, feature differences. A higher value of S_d suggests that the generative model produces samples that are, on average, more distinct from the instances present in the original training data. This indicates a greater capacity for exploration of the data space beyond the immediate vicinity of the training examples.

Semantic Integrity Quantification The direct measurement of a generative model's adherence to unknown or implicitly defined semantic constraints presents a significant methodological challenge. To address this, we adopt a controlled evaluation strategy. This strategy involves the use of synthetically engineered features, for which specific semantic constraints are explicitly defined by design. These engineered features are then injected into an existing real-world dataset. For our experiments, we utilize the *Child* dataset as the base for this augmentation. This dataset is selected primarily due to its predominantly categorical nature, which simplifies the process of defining and embedding clear, rule-based semantic constraints.

The core of this evaluation strategy is the introduction of a set of binary synthetic variables, denoted as V_i. The generation process for each such variable V_i is designed to be conditionally dependent on the subgroup membership k of instances within certain existing base features X_j from the *Child* dataset. This conditional dependency is modeled using a Binomial distribution, such that $V_i | X_{j,k} \sim \text{Binomial}(n_{j,k}, p_{i|j,k})$. Here, $X_{j,k}$ indicates that the base feature X_j belongs to subgroup k, and $p_{i|j,k}$ is the probability parameter of the Binomial distribution for V_i given this specific subgroup membership. The parameter $n_{j,k}$ (number of trials) is typically set to 1 for binary variables.

To establish ground-truth semantic rules, we deliberately set the probability $p_{i|j,k} = 0$

for certain pre-selected combinations of synthetic variables V_i and subgroups k of base features X_j. This effectively defines a hard constraint: for example, if feature X_1 belongs to a specific subgroup k^*, then the synthetic feature V_1 *must* take the value 0. Any generated sample where X_1 is in subgroup k^* but V_1 is not 0 constitutes a clear violation of this engineered semantic rule. The generative models under evaluation are then trained on this augmented dataset, which now includes these engineered features and their inherent constraints. After training, we generate a set of synthetic samples from each model and measure the frequency with which these engineered semantic rules are violated. This violation frequency serves as a direct, quantitative proxy for assessing the model's ability to learn and enforce the embedded semantic rules.

Furthermore, this methodology allows for a more nuanced analysis beyond simple rule violation. By setting other $p_{i|j,k}$ values to levels that represent either under-represented or over-represented subgroups (i.e., $p_{i|j,k}$ values that are small but non-zero, or large, respectively), we can also evaluate the model's capacity for preserving distributional characteristics within these defined semantic contexts. This comprehensive approach, based on engineered features with known ground-truth constraints, enables an objective comparison of the capabilities of different generative models in adhering to semantic constraints, even within an otherwise unsupervised learning paradigm where the original, underlying semantic rules of the base dataset remain unknown.

3.5.2 General Experiment Setup

The experimental configuration for both the validator and synthesizer components of our Latent Semantic Regularization (LSR) framework, as well as for the baseline models, is detailed as follows. The encoder and decoder networks within the validator component are implemented as Multi-Layer Perceptrons (MLPs). Each of these MLPs consists of three hidden layers, with each hidden layer containing 128 neurons. The dimensionality of the semantic latent space Z', denoted as $dim(Z')$, is set to 200. The prior distribution over this latent space, $p_\eta(z')$, is modeled using a tail-adaptive normalizing flow. This normalizing flow is specifically a Real NVP (Non-Volume Preserving) architecture, as described in [60], with a transformation depth of 4. The base density

for this Real NVP is a 20-component mixture of Generalized Gaussian Distributions (GGDs). The coupling layers within the Real NVP transformations utilize MLPs, each constructed with a single hidden layer containing 128 neurons.

For the synthesizer component, both the generator and the critic networks are also implemented as MLPs. These MLPs are designed with two hidden layers, and each hidden layer has a width of 256 neurons. For the baseline models, namely MedGAN, CTGAN, and TableGAN, we utilize their standard, publicly available implementations, adhering to their established architectures and default configurations where applicable.

All models subjected to evaluation, including our proposed LSR variants (LSR (S+V) and LSR (S)) and the selected baseline synthesizers, are trained for a fixed duration of 300 epochs. The optimization of model parameters during training is performed using the ADAM (Adaptive Moment Estimation) optimization algorithm. A consistent learning rate of 2×10^{-4} is employed across all training procedures to ensure a standardized optimization process.

3.5.3 Motivating Experiment

To provide intuition for the role of LSR's validator in identifying and regularizing semantic violations, we first consider a controlled experiment on a low-dimensional synthetic dataset.

Dataset and Task Setup We construct a 2D dataset by sampling from a mixture of three isotropic Gaussian distributions. We apply K-Means clustering ($K = 3$) to the sampled points to obtain cluster assignments, which serve as ground-truth labels, and add them as a feature to the dataset. The task for the generative models is twofold: capture the underlying 2D Gaussian mixture distribution; and generate the correct label associated with each generated point. Generating samples with incorrect labels constitutes a violation of the implicit semantic rule. This setup serves as a tractable proxy for more complex semantic constraints of real-world data.

Table 3.1: Percentage of incorrectly labelled samples on the 2D Gaussian mixture dataset. Lower values indicate better adherence to the implicit semantic rule defined by cluster assignments.

Model	Incorrect Labels ↓
MEDGAN	$51.4\% \pm 11.9\%$
TABLEGAN	$44.2\% \pm 3.5\%$
CTGAN	$10.9\% \pm 0.5\%$
LSR (S+V)	$6.4\% \pm 0.1\%$
LSR (S)	$7.9\% \pm 0.2\%$

Results We evaluate the semantic integrity of the generated data by measuring the percentage of samples with incorrect cluster labels. Table 3.1 presents these results. We observe that baseline methods including our ablated synthesizer exhibit significant difficulty in adhering to the semantic constraint. In contrast, the LSR (S+V) model achieves the lowest error rate (6.4%), demonstrating clear improvement attributable to the validator's regularization signal.

Visualizing the Validator's Role: Impact of Prior Flexibility To understand how the validator enables this improvement, we investigate the structure of its learned latent semantic space under different prior choices, using the same 2D dataset. We compare the validator component trained with a standard Gaussian prior versus one trained with our flexible Tail-Adaptive NF prior. We then visualize the latent representations $z' = E_\phi(x')$ for generated samples.

Figure 3-2 presents this comparison. When using a standard Gaussian prior, the latent codes of incorrectly labeled samples are scattered throughout the latent space mixed with that of correctly labeled samples. This lack of separation indicates that the validator, constrained by the simplistic prior, fails to learn a latent structure that effectively distinguishes semantically valid from invalid samples. In contrast, when using the Tail-Adaptive NF prior, the validator learns a more structured latent space. The incorrectly labeled samples are now predominantly clustered in lower-density regions of the learned prior, distinctly separated from the high-density regions corresponding to semantically correct samples. This improved separation implies that the flexible prior allows the validator to better capture the subtle distributional shifts associated with semantic violations, highlighting the importance of using an expressive, learnable prior within the

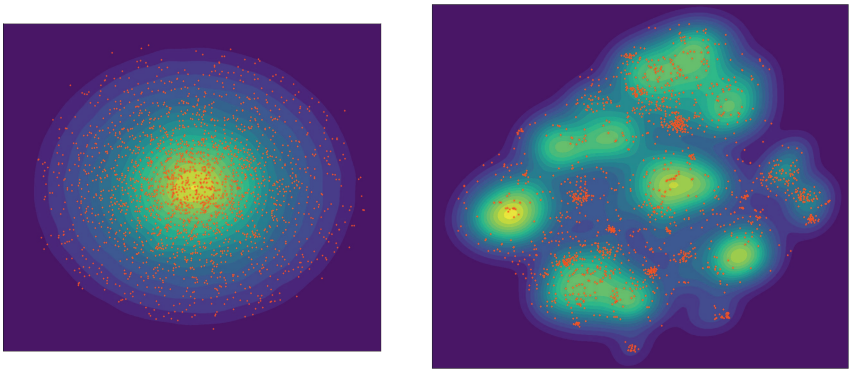


Figure 3-2: Comparative visualization of the validator's latent space for semantically incorrect samples (red dots) from the 2D experiment. **Left:** Validator with a standard Gaussian prior shows poor separation. **Right:** Validator with a Tail-Adaptive NF prior effectively clusters incorrect samples in low-density regions, enabling stronger semantic regularization.

validator component to accurately model semantic boundaries.

3.5.4 Evaluation on Benchmark Datasets

We extend our evaluation to standard real-world benchmark datasets commonly used in tabular data synthesis literature. This allows for a comprehensive assessment of LSR's performance across diverse data characteristics and complexities.

Datasets We utilize five benchmark datasets: *Adult Income*, *Covertype*, *Census*, *Intrusion*, obtained from the UCI Machine Learning Repository [98], and the *Child* dataset [99]. These datasets represent a variety of domains, feature types, and class balances, providing an extensive testbed for evaluating generative models.

Data Utility Evaluation We first assess the general utility of the models by measuring both statistical fidelity and downstream task performance.

Statistical Fidelity: Table 3.2 presents the average KS statistic for numerical features, while Table 3.3 shows the average Log Chi-Squared statistic for categorical features. For numerical features (Table 3.2), LSR (S+V) achieves the best fidelity on the Adult, Intrusion, and Census datasets. While TableGAN performs best on Covertype, LSR

Table 3.2: Statistical Fidelity (Numerical Features): Average KS test statistic across all numerical features. Lower is better.

Model	Adult ↓	CovType ↓	Intrusion↓	Census ↓	Child ↓
MEDGAN	0.92 ± 0.003	$0.42 \pm 0.$	0.18 ± 0.002	0.82 ± 0.005	N/A
TABLEGAN	0.40 ± 0.02	$\mathbf{0.11 \pm 0.01}$	0.03 ± 0.003	0.51 ± 0.007	N/A
CTGAN	0.2 ± 0.02	0.14 ± 0.02	0.033 ± 0.006	0.22 ± 0.02	N/A
LSR (S+V)	$\mathbf{0.17 \pm 0.001}$	0.12 ± 0.02	$\mathbf{0.027 \pm 0.007}$	$\mathbf{0.15 \pm 0.02}$	N/A
LSR (S)	0.18 ± 0.01	0.13 ± 0.02	0.03 ± 0.007	0.19 ± 0.03	N/A

Table 3.3: Statistical Fidelity (Categorical Features): Average Log Chi-Squared test statistic across all categorical features. Lower is better.

Model	Adult ↓	CovType ↓	Intrusion↓	Census ↓	Child ↓
MEDGAN	9.4 ± 8.1	10.8 ± 10.8	8.6 ± 7.3	9.6 ± 8.1	8.4 ± 7.1
TABLEGAN	8.7 ± 7.4	$\mathbf{5.4 \pm 4.4}$	$\mathbf{3.1 \pm 2.2}$	$9.1 \pm 8.$	6.0 ± 5.6
CTGAN	8.1 ± 6.2	9.9 ± 7.1	$9. \pm 6.6$	8.45 ± 6.9	6.4 ± 5.6
LSR (S+V)	$\mathbf{7.8 \pm 5.}$	$14.5 \pm 13.$	8.9 ± 5.2	$\mathbf{8.38 \pm 6.8}$	$\mathbf{4.49 \pm 3.1}$
LSR (S)	8.0 ± 4.9	10.1 ± 7.5	9 ± 5.5	10.5 ± 10.4	4.6 ± 3.12

(S+V) remains highly competitive. For categorical features (Table 3.3), LSR (S+V) yields the lowest Log Chi-Squared error on Census and Child datasets. Notably, the ablated LSR (S) consistently underperforms the full model across most datasets and metrics. This suggests that the semantic regularization provided by the validator not only enhances semantic integrity (as shown later) but also contributes positively to preserving the statistical properties of the original data, possibly by preventing the synthesizer from collapsing modes or deviating excessively into statistically unsupported regions.

Downstream Task Efficacy: Table 3.4 assesses the utility of synthetic data for training a classification model, reporting the F1-score relative to training on real data. LSR (S+V) demonstrates strong performance, achieving the highest F1-scores on Adult, Covertype, Child, and Intrusion datasets, exceeding even real data performance on Intrusion and Child (which indicates the LSR implicitly smoothing noisy decision boundaries present in the original data, leading to better generalization on the held-out test set). LSR (S+V) slightly underperforms CTGAN on the Census dataset in this specific task. However, CTGAN's comparatively weaker statistical fidelity scores on Census indicate overfitting to the specific classification task or dataset characteristics, potentially at the

Table 3.4: Downstream Task Efficacy (F1-Score): Performance of a classifier trained on synthetic data vs. real data. Higher is better.

Model	Adult ↑	Covtype↑	Intrusion↑	Census ↑	Child ↑
Real Data	0.67 ± 0.0	0.82 ± 0.02	0.91 ± 0.07	0.51 ± 0.01	0.49 ± 0.0
MEDGAN	0.48 ± 0.3	0.11 ± 0.02	0.19 ± 0.08	0.03 ± 0.03	0.31 ± 0.08
TABLEGAN	0.24 ± 0.09	0.49 ± 0.02	0.79 ± 0.11	0.04 ± 0.01	0.46 ± 0.02
CTGAN	0.61 ± 0.02	0.46 ± 0.84	0.10 ± 0	$\mathbf{0.47 \pm 0.0}$	0.49 ± 0.01
LSR (S+V)	$\mathbf{0.64 \pm 0.0}$	$\mathbf{0.59 \pm 0.01}$	$\mathbf{0.97 \pm 0.0}$	0.27 ± 0	$\mathbf{0.51 \pm 0.0}$
LSR (S)	0.64 ± 0.0	0.57 ± 0.02	0.96 ± 0.0	0.27 ± 0.0	0.49 ± 0.0

cost of broader accuracy.

Having established the general data utility of LSR, we now turn to the core claim: its ability to balance exploration (sample diversity) and exploitation (adhering to semantic constraints).

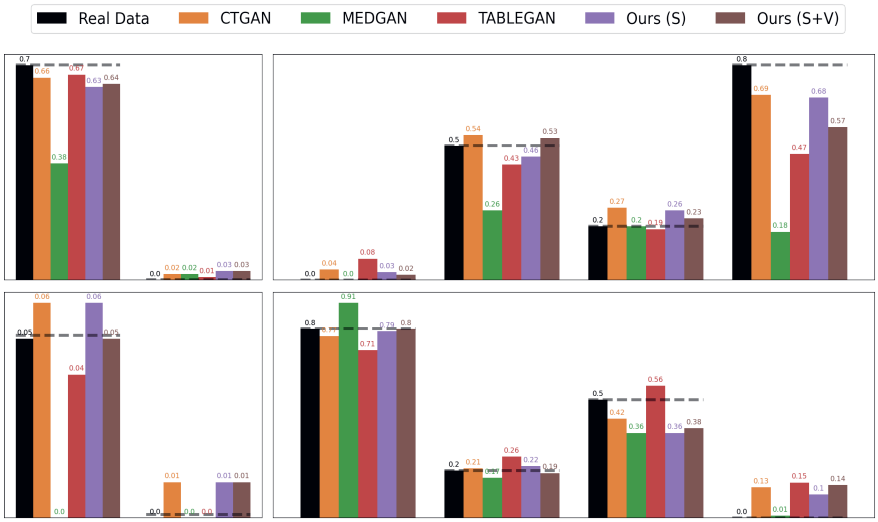


Figure 3-3: Distribution of synthetic variables (left column: $V1$-$V2$; right column: $V3$-$V4$) within subgroups for the augmented *Child* dataset. Black bars represent the ground-truth frequencies. LSR (S+V) generally shows good fidelity, while some baselines (e.g., MedGAN on $V2$) exhibit significant distortions

Generative Diversity (Exploration) Table 3.5 reports the Exploration Factor (S_d) for each model, with higher S_d values indicate greater average dissimilarity between generated samples and training data, signifying broader exploration of the potential data space).

The results show that LSR (S+V) achieves the highest exploration factor on the Adult, Intrusion, Census, and Child, indicating that semantic validator, contrary to stifling diversity through constraint enforcement, enables the synthesizer to explore more extensively by helping prevent mode collapse. The ablated model while showing strong exploration falls short of LSR (S+V). This further supports the notion that the validator component positively contributes to generative diversity in our framework. CTGAN achieves the highest score only on the Covtype dataset. These findings suggest that LSR encourages the generation of novel and diverse samples.

Table 3.5: Exploration Factor (S_d): Cumulative adjusted Hamming distance to nearest neighbors in the real dataset. Higher values indicate greater generative diversity.

Model	Adult ↑	Covtype↑	Intrusion↑	Census ↑	Child ↑
MEDGAN	$2240 \pm 43\%$	$118 \pm 100\%$	$2282 \pm 71\%$	$132774 \pm 9\%$	$1058 \pm 42\%$
TABLEGAN	$6163 \pm 20\%$	$81 \pm 43\%$	$1090 \pm 27\%$	$240295 \pm 11\%$	$2381 \pm 33\%$
CTGAN	$4296 \pm 13\%$	$\mathbf{721 \pm 29\%}$	$7849 \pm 27\%$	$117534 \pm 6\%$	$1392 \pm 13\%$
LSR (S+V)	$\mathbf{14571 \pm 5\%}$	$259 \pm 11\%$	$\mathbf{17078 \pm 2\%}$	$\mathbf{315871 \pm 4\%}$	$\mathbf{2694 \pm 4\%}$
LSR (S)	$10432 \pm 2\%$	$141 \pm 18\%$	$15922 \pm 4\%$	$249433 \pm 1\%$	$1374 \pm 2\%$

However, diversity alone is insufficient as unconstrained exploration can lead to hallucination. Therefore, we must evaluate exploration in tandem with exploitation. Having demonstrated LSR's capacity for exploration, next we evaluate its primary objective: enhancing semantic integrity by exploiting learned constraints.

Controlled Semantic Constraint Setup We utilize the controlled experimental setup described in Section 3.5.1 for the *Child* dataset. We introduce (V_1, V_2) associated with a base feature X_1, and (V_3, V_4) associated with a base feature X_2. Their values are determined by conditional binomial distributions $V_i|X_{j,k} \sim \text{Binomial}(n_{j,k}, p_{i|j,k})$, setting $p_{i|j,k} = 0$ for at least one specific subgroup k of the corresponding base feature X_j. Violating these zero-probability rules constitutes a semantic error. Other $p_{i|j,k}$ parameters are set to induce under- and over-representation, allowing assessment of distributional fidelity within these semantic contexts.

Quantitative Analysis of Semantic Violations Table 3.6 reports the percentage of generated samples that violate the ground-truth zero-probability constraints. Lower

percentages indicate better adherence to the engineered semantic rules. We report the *absolute percentage* of violations for all models. The results demonstrate the effectiveness of LSR (S+V) in enforcing semantic constraints. It consistently achieves the lowest violation rates across V_1, V_2, and V_3. For V_4, CTGAN exhibits a marginally lower violation rate, but as discussed below, this comes at a potential cost. Importantly, the ablated model shows significantly higher violation rates than the full model across all variables, confirming that the semantic regularization signal provided by the validator is crucial for guiding the synthesizer away from producing semantically invalid combinations.

Examining the baselines, MedGAN and TableGAN, while exhibiting relatively low violation rates for some variables, achieve this alongside poor overall data utility and diversity (Tables 3.2-3.5), indicating a failure to capture the broader data distribution effectively. CTGAN presents a stronger baseline, showing good utility and moderate exploration. However, LSR (S+V) generally outperforms CTGAN in constraint adherence (V_1, V_2, V_3) while simultaneously demonstrating superior exploration and comparable or better data utility. The slight advantage of CTGAN on V_4 violation rate must be weighed against LSR (S+V)'s overall stronger performance profile.

Table 3.6: Semantic Integrity Violations on Augmented *Child*: Percentage of generated samples violating zero-probability constraints. Absolute percentages are reported against ground truth. According to the qualitative analysis, CTGAN is the only contender with our method since other methods have lower violation at the expense of suppressing minority subgroups.

Model	V_1 Violations (%) ↓	V_2 Violations (%) ↓	V_3 Violations (%) ↓	V_4 Violations (%) ↓
MEDGAN	0.13 ± 0.04*	0.04 ± 0.01*	0.11 ± 0.01*	0.07 ± 0.01*
TABLEGAN	0.02 ± 0.01*	0.37 ± 0.11*	2.59 ± 0.91*	0.36 ± 0.08*
CTGAN	1.53 ± 0.70*	0.61 ± 0.22*	1.00 ± 0.28*	$\mathbf{0.42 \pm 0.03}$*
LSR (S+V)	$\mathbf{0.90 \pm 0.10}$*	$\mathbf{0.60 \pm 0.10}$*	$\mathbf{0.50 \pm 0.06}$*	0.50 ± 0.06*
LSR (S)	1.10 ± 0.11*	0.86 ± 0.16*	0.77 ± 0.08*	0.61 ± 0.07*

Qualitative Analysis of Subgroup Distributions Beyond constraint violations, we examine the model's ability to capture the distribution of synthetic variables within under- and over-represented subgroups. This provides insight into whether semantic integrity is achieved at the cost of distorting subgroup distributions. Figure 3-3 reveals that models like MedGAN, despite low violation counts in Table 3.6, significantly distort

the underlying distributions (e.g., suppressing subgroup 1 for $V2$). This confirms that low violation rates alone are insufficient; they might result from poor exploration or mode collapse. LSR (S+V), in contrast, generally maintains better fidelity to the target subgroup distributions while simultaneously achieving low violation rates and high exploration. This holistic view reinforces the conclusion that LSR successfully balances exploration and exploitation, enhancing semantic integrity without sacrificing distributional accuracy or diversity. The underperformance of the ablated LSR (S) across both violation rates and distributional fidelity further emphasizes the crucial role of the validator-driven semantic regularization.

3.6 Summary

In this chapter, we have introduced a model-agnostic framework designed for the unsupervised learning and enforcement of semantic constraints inherent within a target dataset. The primary objective of this framework is to improve the semantic integrity of synthetic data generated by probabilistic models. Our proposed approach achieves this by actively steering a generative model, the synthesizer, to produce samples that adhere to these learned semantic constraints. This guidance is facilitated through the incorporation of a distinct component, termed the validator.

A key aspect of our framework is the specific design of the validator's latent space prior. The validator employs the tail-adaptive normalizing flow, the development of which was detailed in Chapter 2, to model this prior distribution. The utilization of such a flexible and adaptive prior for the validator's latent space is intended to ensure that the generative model, when regularized by the validator, remains faithful to the intrinsic statistical properties and complex tail behaviors of the original data. Simultaneously, this mechanism aims to compel the generative model to respect the learned semantic boundaries of the data, thereby reducing the generation of implausible or inconsistent samples. This comprehensive framework is engineered to enhance both the statistical fidelity and the practical applicability of generative models across a diverse range of unsupervised learning scenarios, particularly where the semantic correctness of synthetic data is of critical importance.

CHAPTER
4

CONSTRAINED-SENSITIVITY DIFFERENTIALLY PRIVATE NORMALIZING FLOWS FOR TABULAR DATA SYNTHESIS

RQ3: How can Normalizing Flows be released under (ε, δ)-differential privacy while preserving density fidelity without resorting to private stochastic optimization?

4.0.1 Motivation

The proliferation of large-scale datasets, often acquired through extensive internet crawling and aggregation, has significantly advanced the capabilities of generative models (GMs). However, this trend towards training on increasingly vast and diverse data sources also elevates the likelihood that these models will be exposed to, and consequently learn from, sensitive information. Such sensitive data can encompass a wide spectrum, including personally identifiable information (PII), private communications, or media subject to copyright restrictions. The potential for GMs to inadvertently memorize and subsequently reveal such sensitive information during generation poses substantial privacy risks. In response to these concerns, Differential Privacy (DP) has emerged as the de facto standard and a formally rigorous framework for privacy-

preserving machine learning, as discussed in [100].

The majority of established DP methods achieve their privacy guarantees by introducing a controlled amount of stochasticity into the model training or data processing workflow. This is commonly realized by adding carefully calibrated noise, often during the training phase, for instance, by perturbing the gradients computed during model optimization, a technique exemplified by DP-SGD (Differentially Private Stochastic Gradient Descent) [101]. The magnitude of the injected noise is a critical parameter, as it directly governs the trade-off between the utility of the trained model (i.e., its performance on its intended task) and the strength of the privacy guarantee it offers, typically quantified by a privacy budget ε [102, 33]. By providing measurable and mathematically rigorous privacy guarantees, differential privacy ensures that the output of an algorithm is statistically almost indistinguishable whether or not any particular individual's information was included in the input dataset. This property effectively safeguards sensitive information by limiting what can be inferred about any single data contributor.

In this chapter, we introduce a novel approach to applying differential privacy specifically to normalizing flows (NFs), a contemporary and powerful subclass of generative models detailed in [103], by taking advantage of their architectural characteristics. Our method diverges from conventional noise-addition techniques. Instead of introducing stochasticity to the model parameters or their gradients during training, we leverage the NFs exact and efficiently computable log-density to define a *randomized release* distribution in output space. Concretely, after deterministic training, we sample synthetic data from a tempered distribution proportional to $p_\theta(x)^\lambda$ (an exponential-mechanism sampler calibrated by our sensitivity bounds); model parameters are not released.

The core premise of our approach is that the level of differential privacy can be directly related to, and controlled by, a deliberate limitation of the NF's expressivity. By carefully constraining the capacity of the transformations within the normalizing flow, we obtain uniform bounds on per-example log-likelihoods, which yield a finite global sensitivity for the release score and thus calibrate the randomized sampler. The randomness enters only at release time; training remains noise-free.

As an initial proof of concept and to clearly illustrate the foundational principles of this approach, we begin with a specific sub-family of normalizing flows: volume-preserving flows. These flows, by their nature, have a simpler Jacobian determinant structure, which facilitates the analysis and implementation of expressivity constraints. We then extend the same principles to affine and residual flows by constraining the Jacobian term via scaling and spectral normalization. Our empirical evaluations demonstrate that this training-noise-free, release-private method can achieve competitive performance, particularly in settings that demand strong privacy guarantees (i.e., small privacy budgets). This suggests a promising alternative to noise-based DP, potentially offering a better balance between privacy and utility by avoiding direct degradation during optimization.

Normalizing Flows

A *normalizing flow* (NF) transforms a simple distribution into a complex one through a series of invertible transformations. Let $z \in \mathbb{R}^D$ be a latent variable that follows a known, (usually) fixed distribution $p_z(z)$. Training a NF amounts to fitting a parameterized, invertible map $f_\theta : \mathbb{R}^D \to \mathbb{R}^D$ such that $z = f(x;\theta)$. A NF's density function can then be computed exactly using a change of variables:

$$\log p(x;\theta) = \log p_z\left(f(x;\theta)\right) + \log \left| \det \frac{\partial f}{\partial x} \right|, \tag{4.1}$$

where $\det \frac{\partial f}{\partial x}$ is the determinant of the transformation's Jacobian. The Jacobian determinant of flow models captures how the volume changes locally as data is transformed from one distribution to another. It ensures the probability density is correctly adjusted under the change of measure. Due to NFs allowing for exact density calculations, they can be trained via (exact) maximum likelihood estimation. Given a dataset $\mathcal{D} = \{x_1, x_2, \ldots, x_N\}$, the log-likelihood is: $\mathcal{L}(\theta;\mathcal{D}) = \sum_{i=1}^{N} \log p(x_i;\theta)$. Once the NF is trained, we can draw samples via the inverse transformation: $\hat{z} \sim p(z), \hat{x} = f^{-1}(\hat{z};\theta)$.

Differential Privacy

Differential Privacy (DP) safeguards the privacy of the data that is the subject to some analysis, guaranteeing that the presence or absence of a particular individual has a minimal effect on the outcome of the analysis. Let $\mathcal{D}$ and $\mathcal{D}'$ be two 'neighboring' datasets differing by one element. To satisfy pure ε-DP, a randomized mechanism M must, for any subset $\mathcal{S}$ of possible analysis outcomes, adhere to the condition:

$$\Pr(M(D) = O) \leq e^{\varepsilon} \cdot \Pr(M(D') = O). \tag{4.2}$$

The privacy budget ε quantifies the level of privacy. The most common approach to DP in machine learning is to add noise—usually sampled from a Gaussian or Laplacian distribution—to the model gradients during optimization [101]. However, introducing this perturbation creates a utility/privacy trade-off and can exacerbate issues such as model unfairness and the handling of non-IID data [102, 104, 42].

4.0.2 Previous Work on Privacy for Normalizing Flows

Normalizing flows have been made differentially private through noise adding mechanism and gradient clipping [2, 105, 106]. [107] privatize the computation of posterior in Bayesian inference for exponential families by truncating the likelihood term. They assume the model to be univariate and truncate its support to bound the sensitivity of unbounded sufficient statistics which are then released in a noisy manner through Laplace mechanism to the modeller to compute the posterior.

4.1 Method

We now turn to our novel approach. Consider training a differentially private NF on neighboring datasets $\mathcal{D}$ and $\mathcal{D}'$, which differ only in one member: $x_N \in \mathcal{D}$ vs $x'_N \in \mathcal{D}'$. We will leverage the NF's closed-form density, given in Equation 4.1, to directly derive a DP mechanism. We begin by writing the definition of $\varepsilon - DP$ in its likelihood ratio

form for the NF:

$$\frac{p(\mathcal{D})}{p(\mathcal{D}')} = \frac{\prod_{n=1}^{N} p_z(f(x_n)) \left|\frac{\partial f}{\partial x_n}\right|}{\prod_{n=1}^{N} p_z(f(x'_n)) \left|\frac{\partial f}{\partial x'_n}\right|} = \frac{p_z(f(x_N)) \left|\frac{\partial f}{\partial x_N}\right|}{p_z(f(x'_N)) \left|\frac{\partial f}{\partial x'_N}\right|} \leq e^{\varepsilon} \tag{4.3}$$

where $x'_n = x_n$ for $i = 1, \ldots, N-1$, and $x'_N \neq x_N$. Cancelling the common data points allows the product to be reduced to one term. Our approach is to restrict the expressivity of the NF, ensuring that the likelihood ratio is upper bounded by e^{ε}. Below we describe separate strategies for adapting the transformation, base density, and volume element such that the above ratio is properly bounded to ensure $\varepsilon - DP$.

4.1.1 Constant-Volume Flows

We first consider a less-expressive subclass of NFs: volume preserving and constant-volume flows. As mentioned in Appendix A.1, these NFs are characterized by having a volume element that is not a function of x. In turn, Equation 4.3 simplifies to:

$$\frac{p_z(f(x_N)) \cdot C}{p_z(f(x'_N)) \cdot C} = \frac{p_z(f(x_N))}{p_z(f(x'_N))} \leq e^{\varepsilon}. \tag{4.4}$$

Thus we turn our focus to the ratio $p_z(f(x))/p_z(f(x'))$, dropping the data index to simplify the notation. Below we propose two methods to contain this ratio. First, we propose a simple method through clipping the transformation, demonstrating the idea for a standard Gaussian base density. Then we propose a more general method of constraining the base density ratio through truncating the density function itself. This method does not make any assumptions about the base density family and can be applied to any base density, including mixture densities.

Clipping the Transformation

First consider a simple volume preserving flow (such as NICE [108]) with a standard Normal base density. Substituting the Gaussian density function for p_z and canceling

the constants, we obtain:

$$\frac{p_z(f(x))}{p_z(f(x'))} = \exp\left\{\frac{1}{2}\left(\|f(x')\|^2 - \|f(x)\|^2\right)\right\}.$$

Now we modify the NF's transformation so that it is clipped above and below by a constant:

$$\bar{f}(x) = \begin{cases} \min(f(x), \beta) & \text{if } f(x) > 0, \\ \max(f(x), -\beta) & \text{otherwise,} \end{cases}$$

for $\beta \in \mathbb{R}^+$. With this modification in place, the upper bound for the likelihood ratio is then

$$\frac{p_z\left(\bar{f}(x)\right)}{p_z\left(\bar{f}(x')\right)} \leq \exp\left\{\frac{1}{2}\beta^2\right\}. \tag{4.5}$$

Thus for the desired level of privacy ε, the clipping threshold should be set to $\beta = \sqrt{2\varepsilon}$. It is easy to extend the above derivation to any member of the exponential family, building upon previous work that ensures DP for the exponential family by bounding the sufficient statistics [107].

Truncating the Density Function

Of course the transformation is not the only means by which we can control the density ratio, as it is also a function of the base density. Thus, we present the more general case of truncating p_z, placing a lower bound γ on p_z by truncating it to the interval $[a, b]$. In turn, the denominator of $p_z(f(x))/p_z(f(x'))$ is lower-bounded, controlling the growth of the ratio. Again, similar approaches have been previously used to apply DP to the exponential family [107].

Again we present the approach using a Gaussian density, a standard base density in flow models, and determine cutoff points via its cumulative distribution function (CDF). Specifically, by evaluating the CDF at $-a$ and a, we capture the probability of values outside the interval $[-a, a]$: $P(Z \leq a) - P(Z \leq -a) = 1 - P(-a < Z < a) \leq \gamma$, where Z is a random variable following the standard Gaussian distribution. Due to the symmetry of the Gaussian CDF, we derive cutoffs where $P(Z \leq a) = 1 - \frac{\gamma}{2}$ and $P(Z \leq -a) = \frac{\gamma}{2}$.

In the worst-case scenario after truncation, we have $\frac{\max(P_z \text{ clipped})}{\min(P_z \text{ clipped})} \leq e^{\varepsilon}$, which simplifies the calculation of ε to:

$$\varepsilon = \log(\max(p_z)) - \log(\min(p_z)).$$

Assuming a multivariate Gaussian with mean 0 and unit variance in all dimensions, described by:

$$p_z(f(x)) = \frac{1}{\sqrt{(2\pi)^D}} \exp\left(-\frac{1}{2}\|\mathbf{f}(\mathbf{x})\|^2\right)$$

the ratio of the pdf values for any two points $f(x)$ and $f(x')$ in the support of the distribution is:

$$\frac{p_z(f(x))}{p_z(f(x'))} = \frac{\exp\left(-\frac{1}{2}\|f(x)\|^2\right)}{\exp\left(-\frac{1}{2}\|f(x')\|^2\right)} = \exp\left(-\frac{1}{2}\|f(x)\|^2 + \frac{1}{2}\|f(x')\|^2\right).$$

This ratio will be maximized when $\|f(x)\|$ is minimized and $\|f(x')\|$ is maximized. Since the mean is 0 and each dimension is truncated to $[-a, a]$, the minimum norm occurs at the mean $\mathbf{0}$, and the maximum norm occurs at any corner of the hypercube defined by the truncation limits, which would be a point where each component is either a or $-a$. Thus, the maximum value of $\|f(x)\|^2$ in the truncated distribution would be Da^2, and the minimum would be 0. Hence, the maximum value of the ratio of pdf values is:

$$\max\left(\frac{p_z(f(x))}{p_z(f(x'))}\right) = \exp\left(\frac{1}{2}Da^2\right) \tag{4.6}$$

This is the maximum ratio of the density at the mean to the density at the most extreme point in the support of the truncated D-dimensional Gaussian distribution.

4.1.2 Non-Volume Preserving Flows

We now turn to more general types of NFs: the expressive class of non-volume preserving flows such as RNVP, Glow, and autoregressive flows. We must now work with all terms

of Equation 4.3: the ratio of the base densities as well as the ratio of the volume elements (i.e. Jacobian determinants (JDs)). For the base density ratio, we have already shown in Section 4.1.1 how truncating the base density can constrain this ratio. Turning our attention to the JD term, we will constrain it by putting upper and lower bounds on the Lipschitz constant of the flow. Intuitively, by constraining the Lipschitz norm of the transformations in a normalizing flow, we are effectively controlling the rate at which the transformations stretch or compress the local volume. Consequently, the JD, which quantifies these changes since it is comprised of first-order derivatives, is also bounded. Constraining the JD of a normalizing flow's transformations through Lipschitz continuity exhibits a direct parallel to the fundamental principle of differential privacy by inherently bounding the model's sensitivity to individual data points.

Constraining the Jacobian Determinants Ratio of Affine Flows

We focus on constraining the JD ratio of affine flows. The scale transformation in the affine coupling layers of the normalizing flow, $s(z;\theta)$, is a function of some part of the input z and parameters θ. To bound the JD, we apply a transformation $t : \mathbb{R}^+ \to [\zeta, \eta]$, where ζ,η are user-defined minimum and maximum values for JD term respectively, which can be calculated based on our desired ε.

For the sake of simplicity, we set $\eta = 1$, since it allows us to control the sensitivity by only considering the minimum value of JD, as then in the worst-case scenario, numerator (1) does not influence the growth when exponentiated.

To define t rigorously, we want t to be a continuous, differentiable function that maps the unrestricted scale function s to the restricted interval $[\zeta, 1]$. One candidate for such a function is the hyperbolic tangent, which can be rescaled and shifted appropriately.

Given a scale s, define t as follows:

$$t(s) = (1 - \zeta) \cdot \frac{\tanh(s) + 1}{2} + \zeta.$$

This transformation ensures that $t(s) \in [\zeta, 1]$ for any $s \in \mathbb{R}$.

The log-determinant of the Jacobian for each transformation, considering the limitation

of s, is given by:

$$\log \left| \det \left(\frac{\partial f_{1:K}}{\partial z_0} \right) \right| = \sum_{k=1}^{K} \sum_{i=d+1}^{D} t(s_i(z_{k-1,1:d}; \theta_k)).$$

Here, $t(s_i)$ is the limited transformation of the scale s_i.

The ratio of the Jacobian determinants for the neighboring datasets is given by:

$$\frac{\left| \det \left(\frac{\partial f_{1:K}}{\partial x_N} \right) \right|}{\left| \det \left(\frac{\partial f_{1:K}}{\partial x'_N} \right) \right|} = \frac{\exp \left(\sum_{k=1}^{K} \sum_{i=d+1}^{D} t(s_i(x_{N,k-1,1:d}; \theta_k)) \right)}{\exp \left(\sum_{k=1}^{K} \sum_{i=d+1}^{D} t(s_i(x'_{N,k-1,1:d}; \theta_k)) \right)}.$$

By selecting ζ according to the privacy budget ε, we can ensure that the above condition is met. The exact choice of ζ would need to be determined based on the desired level of privacy and the specifics of the NF architecture and dataset. This approach ensures that the model's expressivity is limited enough to satisfy differential privacy constraints while still allowing for meaningful data transformations.

To calculate the privacy bounds, we consider the worst case scenario for the ratio of Jacobian determinants, where the numerator is fixed at its maximum value of 1, and the denominator is at its minimum value, ζ. We solve the inequality for the maximum ratio of Jacobian determinants in a normalizing flow with K steps, $\left(\frac{1}{\zeta}\right)^{K \cdot D} \leq e^{\varepsilon}$, for ε:

$$K \cdot D \cdot \log \left(\frac{1}{\zeta} \right) \leq \varepsilon. \tag{4.7}$$

The choice of scaling function in differentially private normalizing flows presents an opportunity to optimize the trade-off between privacy and utility. While *tanh* offers a convenient range, tailoring the scaling function to the input data's characteristics allows for more granular control over expressivity. This data-driven approach can preserve critical information while adhering to privacy constraints, potentially mitigating performance degradation observed in methods like DP-SGD, particularly with heavy-tailed distributions. This strategic limitation of expressivity, guided by the data's inherent structure, enables the development of more effective differentially private normalizing flows.

Spectral Normalization of the Affine Transformation

While the method presented in the previous section allows us to apply our noise-free differential privacy technique to a wide range of common affine flows, it limits our ability to generalize this approach to more complex flow architectures. To extend our noise-free DP method to more general flow families, such as residual flows, we need a mechanism to control the expressivity of the transformations without relying on the affine assumption. Spectral normalization offers a solution by constraining the Lipschitz constant of the transformation, which directly bounds the sensitivity of the flow's density function to changes in the input data. By employing spectral normalization, we can relax the affine requirement and potentially apply our DP methodology to a broader class of normalizing flows, including those with non-constant JDs.

Consider the L-Lipschitz function $f : \mathbb{R}^n \to \mathbb{R}^n$:

$$\forall x, y \in \mathbb{R}^n, \quad \|f(x) - f(y)\| \leq L\|x - y\|$$

Let $J(x)$ denote the Jacobian of f at x. By the Mean Value Theorem:

$$f(x) - f(y) = J(z)(x - y)$$

where z lies on the line segment connecting x and y. Taking norms:

$$\|f(x) - f(y)\| = \|J(z)(x - y)\| \leq \|J(z)\|\|x - y\|$$

Since $\|A\| \leq \sigma_{\max}(A)$ for any matrix A:

$$\|f(x) - f(y)\| \leq \sigma_{\max}(J(z))\|x - y\|$$

Comparing with the Lipschitz condition:

$$L \leq \sigma_{\max}(J(z))$$

Thus, $\sigma_{\max}(J(z))$ upper bounds the Lipschitz constant L for f. Based on this we define the spectral norm of matrix A as

$$\hat{A} = \frac{A}{\sigma_{max}(A)}.$$

This ensures that the Lipschitz constant L of the transformation is upper bounded by 1.

Now consider a flow with N steps, each step parameterized by a neural network of L layers g_l and its corresponding 1-Lipschitz activation function defined as a_l. At each step n, we have an affine transformation defined by the neural network parameters and the activation function:

$$x_{n+1} = a(g_L(a(g_{L-1}(\ldots a(g_1(x_n))\ldots)))).$$

To ensure that the entire flow has an upper bound of 1 for its Jacobian determinant term, we will show that the spectral norm of the Jacobian matrix of the entire transformation at each step is bounded by 1.

Using the previously established inequality, we can write the Lipschitz constant $L_{n,l}$ of each layer l of step n as:

$$L_{n,l} \leq \sigma_{\max}(J_{n,l}),$$

where $J_{n,l}$ is the Jacobian matrix of the layer g_l at each step n.

Since the activation functions are assumed to be 1-Lipschitz, each activation function a_l does not increase the Lipschitz constant. Therefore, for the composition of activation functions and transformations within each step, the Lipschitz constant of the entire step n will be:

$$|J_n| = \prod_{n=1}^{N} |\hat{J}_{n,l}| \leq 1$$

where $\hat{J}_{n,l}$ is the specrally normalized jacobian matrix of $g_{n,l}$.

The Jacobian determinant of the entire flow after N steps is the product of the determinants of the individual steps:

$$|J_{\text{flow}}| = \prod_{n=1}^{N} |J_n|.$$

Since each $|J_n|$ is bounded by 1 due to the spectral normalization and the activation functions' Lipschitz constant, we can conclude that the entire flow's Jacobian determinant is also bounded by 1:

$$|J_{\text{flow}}| \leq 1.$$

This implies that the entire flow, with each layer's parameters spectrally normalized and activation functions being 1-Lipschitz, has an upper bound of 1 for its Jacobian determinant term.

Scaled Spectral Normalization: Instead of directly dividing by the largest singular value, we scale the weight matrix W by $1/c$ before computing the spectral norm:

$$\hat{W} = \frac{c \cdot W}{\sigma_{\max}(W)}.$$

This modified normalization ensures that the Lipscthiz norm of the scaled weight matrix $\hat{W}$ has a an upper bound of of c.

For an affine flow with N steps each step parameterized by neural networks of l layers, the overall Lipschitz constant of the entire flow is upper bounded by:

$$L_{\text{flow}} \leq \prod_{l=1}^{L} \prod_{n=1}^{N} c. \tag{4.8}$$

To establish a lower bound on the Jacobian determinant , we can employ a similar scaling technique with spectral normalization. Instead of scaling down by the largest singular value, we scale *up* the weight matrix by a factor $c' > 1$ before applying spectral

normalization and then reconstruct the weight matrix. This results in a modified weight matrix $\hat{W} = c' \cdot \frac{W}{\sigma_{\max}(W)}$, where $\sigma_{\max}(W)$ represents the largest singular value of W. This ensures the smallest singular value of $\hat{W}$ is at least 1, effectively lower bounding the JD by a factor related to c and the dimensionality of the transformation. This approach, analogous to the upper bounding technique, constrains the minimum scaling effect of the transformation, preventing excessive compression of the probability density.

Differential Privacy Bounds Calculation: Combining the worst-case scenarios for both the p_z ratio and the Jacobian determinant ratio, we derive the differential privacy bounds as follows:

$$\varepsilon \leq \log \left(\frac{1}{2} Da^2 \cdot \frac{1}{\zeta} \right) \tag{4.9}$$

This equation provides a concise formulation for the upper bound on ε, ensuring that the training of the normalizing flow on dataset $\mathcal{D}$ satisfies ε-DP, as per the constraints set forth in Equation 4.3.

This construction not only adheres to the theoretical underpinnings of differential privacy but also leverages the inherent properties of normalizing flows to achieve privacy without the addition of noise, thus preserving the utility of the model.

Scope and release. After constraining the sensitivity of the deterministic map $\mathcal{D} \mapsto \theta(\mathcal{D})$, the differentially private object is the *release mechanism* that uses the trained flow. Let

$$U(\mathcal{D}, x) \;=\; \log p_{\theta(\mathcal{D})}(x).$$

Under the architectural constraints above (base truncation and Jacobian bounds), there exist finite constants $L_{\min} < L_{\max}$ such that for all θ in the constrained family and all x,

$$L_{\min} \;\leq\; \log p_\theta(x) \;\leq\; L_{\max},$$

hence the global sensitivity of U under add/remove adjacency satisfies

$$\Delta \;\coloneqq\; \sup_{x,\, \mathcal{D} \sim \mathcal{D}'} \left| U(\mathcal{D}, x) - U(\mathcal{D}', x) \right| \;\leq\; 2B, \qquad B \;\coloneqq\; \max\{|L_{\min}|, |L_{\max}|\}.$$

Private release via exponential mechanism in output space. Define, for a target privacy level $\varepsilon > 0$, the density

$$\pi_\varepsilon(x \mid \mathcal{D}) \;\propto\; \exp\!\Big(\tfrac{\varepsilon}{2\Delta}\, U(\mathcal{D}, x)\Big) \;=\; p_{\theta(\mathcal{D})}(x)^{\lambda}, \qquad \lambda \;\coloneqq\; \tfrac{\varepsilon}{2\Delta} \in (0, 1].$$

The mechanism $\mathcal{M}_\varepsilon$ that releases m i.i.d. draws $X_{1:m} \sim \pi_\varepsilon(\cdot \mid \mathcal{D})$ is $(m\varepsilon)$-DP by basic composition. In all experiments we only privatize the *synthetic data release*; model parameters $\theta(\mathcal{D})$ are not released.

Practical sampler. To sample from $\pi_\varepsilon(x \mid \mathcal{D}) \propto p_\theta(x)^\lambda$ without altering training, draw M proposals $\tilde{x}_j \sim p_\theta$ and perform importance resampling with weights

$$w_j \;\propto\; \Big[p_\theta(\tilde{x}_j)\Big]^{\lambda - 1},$$

emitting m resampled points. For a Gaussian base, $\lambda < 1$ corresponds to a simple temperature on the base; the rest of the flow is unchanged.

4.1.3 Privacy Accounting

Adjacency and sensitivity. We use add/remove adjacency. Let $U(\mathcal{D}, x) = \log p_{\theta(\mathcal{D})}(x)$ be the score in the exponential mechanism and let

$$\Delta \;\coloneqq\; \sup_{\mathcal{D} \sim \mathcal{D}',\, x} \Big|U(\mathcal{D}, x) - U(\mathcal{D}', x)\Big|.$$

Under the architectural constraints (base truncation and Jacobian bounds), there exist finite $L_{\min} < L_{\max}$ such that $L_{\min} \le \log p_\theta(x) \le L_{\max}$ for all x and all θ in the constrained family, hence

$$\Delta \;\le\; L_{\max} - L_{\min} \;=:\; \Delta_\star.$$

We set $\lambda = \varepsilon/(2\Delta_\star)$ and release from $\pi_\varepsilon(x \mid \mathcal{D}) \propto p_{\theta(\mathcal{D})}(x)^\lambda$.

Single draw. The exponential mechanism with score U and sensitivity $\Delta_\star$ is *pure* ε-DP. Thus, releasing one synthetic record $X \sim \pi_\varepsilon(\cdot \mid \mathcal{D})$ is ε-DP.

Multiple draws and batches. If we release m records $X_{1:m}$ (possibly adaptively), basic composition yields $(m\varepsilon)$-DP. If we perform R separate releases, where the r-th release emits m_r records at privacy parameter ε_r (and may use different architectural bounds $\Delta_\star^{(r)}$), the total guarantee is

$$\Big(\sum_{r=1}^{R} m_r\,\varepsilon_r\Big)\text{-DP}.$$

Group privacy. For datasets differing in up to g records, the guarantee scales to $\Big(g\sum_{r=1}^{R} m_r\,\varepsilon_r\Big)$-DP.

Post-processing and non-release of parameters. Any (randomized) transformation of the released samples is free by post-processing. Model parameters $\theta(\mathcal{D})$ are not released and only serve to define π_ε.

Exact sampling. To keep $\delta = 0$, we sample *exactly* from π_ε: for exponential-family bases this is a temperature on the base; otherwise we use an acceptreject or MCMC kernel targeting π_ε until convergence diagnostics stabilize, and then draw independent samples. We do not use subsampling amplification or moments/rényi accountants, since optimization is deterministic and privacy arises solely at release time.

4.2 Experiments

We evaluate the proposed noise-free release on five tabular benchmarks with increasing structural complexity: *Power*, *Gas*, *HEPMASS*, *MINIBOONE*, and *Lifesci*. Three flow families are considered: NICE (volume-preserving), affine RealNVP, and Residual flows. All training is deterministic. Privacy enters only at release via the tempered sampler $\pi_\varepsilon(x \mid \mathcal{D}) \propto p_{\theta(\mathcal{D})}(x)^\lambda$ with $\lambda = \varepsilon/(2\Delta_\star)$ (Section 4.1.3). Unless stated otherwise, $\varepsilon = 4.0$.

Complexity profile. We use *complexity* to denote the joint difficulty induced by dimensionality, multimodality, tail heaviness, and local stiffness of the density. Power is

low-dimensional and smooth with skewed marginals. Gas exhibits moderate dimension with structured, locally stiff correlations. HEPMASS is higher dimensional with strong linear structure and class-conditional heterogeneity. MINIBOONE is moderately high dimensional with marked multimodality and non-Gaussian marginals. Lifesci is smaller in sample size with heavy tails and heteroscedasticity. These properties are standard in flow evaluations and manifest in the learnt log-densities.

Setup. Each model uses a truncated Gaussian base; truncation level a is shared across folds and chosen once per dataset to satisfy the sensitivity bound. For RNVP we constrain the JD via scale bounding; for Residual flows we use spectral normalization with target Lipschitz constants as in Section 4.1. Architectures are held fixed across DP and non-DP runs. We report mean test log-likelihoods over 10-fold cross-validation. DP-NF (GDP) numbers are from Waites and Cummings [2] (paper table) and from re-running their public code when available.

4.2.1 Results

Table 4.1: Average test log-likelihood. Higher is better. Our DP release uses ε=4.0 and δ=0. §: values reported in Waites and Cummings [2]. †: our re-run of their code.

Dataset	Lifesci ↑	Gas ↑	HEPMASS ↑	MINIBOONE ↑	Power ↑
OursNICE	4.14 ± 0.10	31.66 ± 1.00	73.64 ± 3.20	24.37 ± 1.10	19.21 ± 0.60
Non-DP NICE	3.26 ± 0.10	40.62 ± 1.10	86.14 ± 3.90	23.68 ± 0.40	16.47 ± 0.50
OursRNVP	12.40 ± 0.50	86.28 ± 4.00	161.03 ± 10.20	26.17 ± 1.60	26.57 ± 0.90
Non-DP RNVP	15.03 ± 1.00	100.52 ± 4.80	222.30 ± 9.20	34.66 ± 0.80	38.11 ± 1.40
OursResidual	$\mathbf{14.98 \pm 0.70}$	91.09 ± 4.20	$\mathbf{235.60 \pm 9.80}$	$\mathbf{41.81 \pm 1.00}$	$\mathbf{61.46 \pm 2.50}$
Non-DP Residual	16.42 ± 0.90	112.83 ± 5.10	301.76 ± 12.10	43.46 ± 1.00	71.02 ± 3.00
DP-NF (GDP)§	10.77 ± 0.30	N/A	N/A	N/A	N/A
DP-NF (GDP)†	9.31 ± 0.20	$\mathbf{102.60 \pm 7.70}$	149.40 ± 6.70	33.41 ± 0.90	54.89 ± 2.50

Aggregate view. Across datasets, Residual flows with spectral constraints provide the best DP utility and the smallest gap to their non-DP counterparts. RNVP trails Residual but remains competitive. NICE underperforms on complex densities, as expected, yet occasionally benefits from the release-time tempering. Relative to DP-SGD training (DP-NF), our release-only DP is consistently stronger on Lifesci, HEPMASS, MINIBOONE, and Power; Gas is the only dataset where the DP-SGD baseline is higher,

which we attribute to differences in preprocessing and to the fact that Gas benefits from mild optimizer noise as an implicit regularizer.

4.2.2 Per-dataset analysis

Power (low dimension, smooth skew). The small dimensionality and smooth conditionals reduce the burden on the JD constraint. Base truncation removes extreme latents that otherwise dominate tails. Our DP Residual obtains 61.46 and exceeds DP-NF (54.89). The DP gap to Non-DP Residual is modest (≈ 9.6), indicating that release-time tempering absorbs much of the expressivity loss without training noise. Interestingly, NICE improves under DP (19.21 vs. 16.47) due to the implicit regularization of the latent temperature.

Gas (moderate dimension, locally stiff correlations). Gas exhibits sharp ridges; JD constraints curb local expansions that non-DP models exploit. Our DP Residual reaches 91.09 versus Non-DP 112.83. DP-NF (102.6) is higher. We observed that for Gas, gradient noise plus clipping can regularize narrow ridges without hampering likelihood as much as a global Lipschitz cap. Increasing ε or relaxing the lower JD bound ζ narrows this gap but weakens privacy. Within our fixed ε=4, RNVP and Residual remain competitive and stable across folds.

HEPMASS (higher dimension, correlated structure). Sensitivity constraints primarily act through the JD. Spectral normalization stabilizes optimization and prevents over-compression. Our DP Residual obtains 235.60, substantially above DP-NF (149.40), and the DP gap to Non-DP is expectedly larger (≈ 66) due to the higher ambient dimension and stronger coupling between layers. RNVP sees a larger drop ($222.30 \rightarrow 161.03$) because its affine scales absorb much of the JD budget; spectral constraints retain more usable capacity.

MINIBOONE (multimodal, non-Gaussian marginals). This dataset stresses latent geometry. Temperature at release reduces overconfident modes and mitigates

mode-collapse artifacts. DP Residual reaches 41.81 versus Non-DP 43.46 (small gap $\approx$ 1.65), outperforming DP-NF (33.41). The combination of base truncation and spectral control appears to preserve inter-mode mass while damping spurious high-density spikes. NICE again benefits marginally from DP (24.37 vs. 23.68), consistent with tempering acting as calibrated smoothing.

Lifesci (small-n, heavy tails). Small sample size amplifies overfitting under non-DP training. The release distribution with $\lambda < 1$ behaves as an explicit temperature, which, together with bounded JD, regularizes tail likelihoods. Our DP Residual achieves 14.98 and exceeds both DP-NF (9.31) and our Non-DP NICE baseline. RNVP remains competitive but shows a larger variance across folds, reflecting sensitivity of affine scales to heavy-tailed marginals.

4.2.3 Discussion

Two effects recur. First, *release-time tempering* offsets expressivity limits by reallocating mass away from narrow spikes; this helps on low-dimensional or small-n regimes (Power, Lifesci) and on multimodal densities (MINIBOONE). Second, *spectral control* is crucial as dimension and stiffness grow (HEPMASS, MINIBOONE): it preserves stable gradients while enforcing a usable JD envelope. Where DP-SGD performs better (Gas), the advantage appears when optimization noise regularizes sharp structures without a global Lipschitz penalty; in our setting, matching those gains would require either a larger ε or architecture-specific JD allocation, which we avoid to keep the mechanism simple and auditable.

4.2.4 Ablations summary

We varied (a, ζ) within the admissible region implied by $\Delta_\star$. Larger a helps on Gas and HEPMASS until JD constraints saturate; smaller a improves Power by damping tails. Tightening ζ uniformly hurts RNVP more than Residual. Full curves are provided in the appendix; the qualitative ranking and the DP vs. Non-DP gaps remain unchanged.

4.2.5 Reproducibility

All runs use identical seeds across DP and Non-DP variants per fold. Preprocessing follows standard flow practice (feature-wise standardization; no task-specific augmentation). Architectural and optimization hyperparameters are shared across privacy settings. Only the release temperature λ depends on ε via Section 4.1.3. We release only synthetic data. Parameters are not released.

4.3 Summary

In this work we proposed a noise-free mechanism for making normalizing flow models differentially private. We present our method for the simple case of volume preserving flows, the popular family of affine flows and the more powerful residual flows. We show that such DP flows work well in high privacy regimes while retaining superior performance. Since we are not adding noise to the data, there is less chance of performance and utility degradation in the model happening because of the introduced perturbation. This advantage, we argue, is more prominent in specific cases such as estimating distributions with long tails.

An orthogonal direction is to train a small family of expressivityconstrained flows deterministically and apply a suitably calibrated privatization mechanism at the model level rather than only at the point of data release. This will be considered as an extension of this work.

CHAPTER
5

DECENTRALIZED SYNTHETIC DATA GENERATION FOR DIGITAL HEALTH TWIN

RQ4: How to perform performant, fair data analysis in a distributed environment by utilizing synthetic data?

5.1 Motivation

As discussed in 1.1, in contemporary healthcare the emergence of the digital health twin signifies a paradigm shift towards a more personalized and precise delivery of medical services. The DHT serves as an intricate virtual model of an individual patient's health status, constructed from a diverse array of data, encompassing both physiological metrics and behavioral patterns. This digital construct is poised to enable the tailoring of clinical pathways to the individual needs of patients with a level of precision previously unattainable.

The realization of this vision necessitates the performance of sophisticated data analysis and the application of federated machine learning techniques on distributed, disparate, and heterogeneous private patient data. These datasets are typically partitioned across various healthcare institutions and are subject to stringent privacy regulations, which present significant challenges in terms of data accessibility and usability.

Federated machine learning emerges as a compelling solution to these challenges, as it allows for the collaborative training of models on decentralized data. This approach ensures that sensitive patient information remains within the organizational boundaries of its source, addressing privacy concerns and regulatory constraints, while still contributing to the collective intelligence of the model.

We also discussed the problem of data access as the main challenge in realization of DHT. Access to real world data can be problematic due to various reasons such as policy and regulations, privacy and communication costs. We positioned synthetic data created by generative models as a solution to this problem. In previous chapters we addressed various issues with synthetic data in medical domain and provided potential solutions in forms of various probabilistic generative models. Federated training of these models will bring us full circle for replacing real datasets with synthetic ones in DHTs.

In fact, the integration of DHTs with federated training of generative models fosters a symbiotic relationship wherein DHTs provide a robust infrastructure for data handling, federated learning offers a (semi-)private methodology for differentially private generative models to synthesize private synthetic data which will then enable safe and private data analysis and model development.

This synergy is crucial for real world applications of DHTs, for example refining *clinical pathways* and advancing the application of digital health twins in the clinical setting. In the domain of healthcare, clinical pathways serve as critical instruments designed to orchestrate patient care with efficiency and coherence, underpinned by the guiding principles of evidence-based medicine. These pathways are essentially algorithmic decision trees that, when optimally tailored, can substantially enhance patient outcomes by ensuring care consistency and reducing variability in clinical practice. However, the endeavor to refine these clinical pathways to cater to the nuanced needs of individual patients encounters a significant impediment in the form of data accessibility (as discussed in 1.2).

Existing initiatives that strive to address these issues are often tailored to address specific instances of data sharing and integration. For example, during the exigencies of the COVID-19 pandemic, there were concerted efforts to facilitate the flow of critical infor-

mation between emergency departments. Nonetheless, these solutions are not without their deficiencies. The setup of such collaborative frameworks is both time-consuming and coordination-intensive, rendering them suboptimal for rapid deployment. The scalability of these solutions is consequently limited by the administrative overhead they incur. The integration of digital health twins into the healthcare ecosystem offers a promising avenue to circumvent the data accessibility and integration challenges that currently hamper the optimization of clinical pathways. By leveraging a digital counterpart that mirrors the patient's unique health landscape, healthcare providers can unlock a dynamic and comprehensive source of patient data. This approach not only streamlines the customization of clinical treatments but also ensures a higher fidelity of care delivery. The digital health twin concept, therefore, represents a critical leap forward in realizing the full potential of evidence-based, patient-centric care. As we continue to refine these virtual models, it is incumbent upon us to also evolve our healthcare frameworks, policies, and technologies to fully harness the benefits of this groundbreaking innovation.

In light of these constraints, in this chapter we present a federated approach to synthetic data generation and integration in DHTs. Our strategy circumvents the direct sharing of patient data by instead creating and employing synthetic datasets. This study is intended to be the culmination of the methods we have introduced in previous chapter, enabling real-world data analysis and model training in a critical domain such as healthcare without compromising individual privacy.

This chapter will first introduce a Proof of Concept (PoC) conducted to demonstrate the viability of our approach. This PoC is predicated upon a simplified scenario, abstracted from a genuine clinical case in psychiatric treatment. By documenting the process and findings of this PoC, we aim to illustrate the practicality of our federated model and its potential for broader application in the optimization of clinical pathways. This demonstration will delve into the specifics of synthesizing data that is both representative and compliant with prevailing policy mandates, thereby offering a glimpse into the future of data-driven healthcare that is both private and personalized. After defining the PoC, we first perform our analysis in a federated setup without relying on synthetic data. We then discuss some of the inherent shortcomings of performing fed-

erated analysis on distributed data related to non-IID distribution of the data between participants. We then use synthetic data synthesized by a federated-trained generative model for a distributed analysis and report the results.

Note: The experiments in this chapter were intended to be carried out on Electroconvulsive therapy (ECT) data from hospitals involved in the EPI project. However, due to extreme difficulties of accessing the data and getting permission to use it for different use-cases as well as the nature of the dataset and homogeneity of its features, the study in Section 3.6 was carried out using public datasets. Also, the scope of the experiments in Section 3.7 is narrow due to the mentioned limitations.

5.2 Conceptual Workflow to Optimize Clinical Pathways

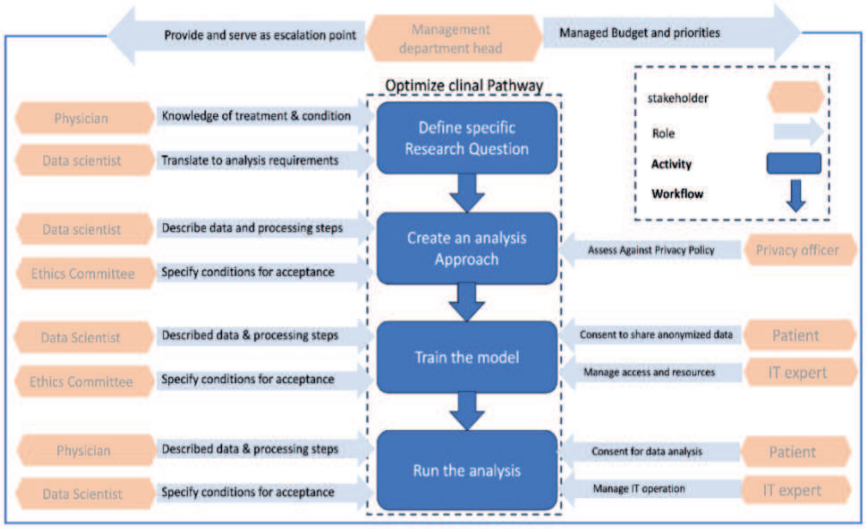


Figure 5-1: Conceptual workflow of an optimized clinical pathway

Figure 5-1 illustrates a theoretical framework intended to enhance the efficiency of a clinical pathway. This figure delineates the sequential stages involved in the workflow, the key participants, and their respective interactions. This chapter will expand upon the context surrounding this workflow.

The drive for refining a clinical pathway originates from a physician who identifies the necessity to tailor treatment processes for a distinct patient cohort. The process begins with formulating a research inquiry that necessitates validation through data analysis. Collaboration with a data scientist is crucial, since they will be able to translate medical requirements into stringent data analysis specifications and policies. Concurrently, a data privacy officer is tasked with establishing and enforcing privacy-related policies and guidelines.

Prior to initiating the analysis, it is important to obtain consent from an ethics committee. This body evaluates the ethical implications of the proposed analysis and may enforce conditions to ensure compliance with ethical standards.

With the requisite approvals in place, the data scientist is then responsible for creating, refining, and implementing a computational model capable of conducting the analysis. Access to an IT infrastructure that aligns with the previously determined requirements and policies is mandatory. An IT specialist facilitates this access, ensuring the infrastructure is equipped to adhere to and enforce the established policies. The data scientist must then train the model using extensive datasets, which may necessitate either anonymization or explicit patient consent for data processing.

Upon completion of the model, individualized analyses can be performed for patients. It is the duty of the healthcare provider to educate the patients regarding the data they will supply and the manner in which it will be utilized, aligning with their personal data access policies. Informed consent is a prerequisite. Subsequently, the data scientist, potentially in collaboration with the physician and patient, may run or scrutinize the analysis. They are also tasked with discussion of the findings and addressing any inquiries.

Throughout the entire optimization process, the department heads play a pivotal role in orchestrating and prioritizing each phase. They ensure adherence to all policies, manage the financial aspects, and provide context and support for the overarching project.

5.3 Use-Case Description

Electro-Convulsive Therapy (ECT), while often reserved as a treatment of last resort, has been shown to be highly effective for individuals suffering from severe depressive disorders. Despite its efficacy, the ability to predict which patients will benefit from ECT, and who might experience deleterious cognitive side effects, remains elusive at the individual level [109].

Research has identified potential indicators of positive ECT outcomes, such as the absence of a personality disorder and the presence of psychotic or catatonic symptoms. The existence of a personality disorder as a comorbidity is linked to a reduced likelihood of a favorable response to ECT, with an odds ratio of 0.3, suggesting a strong argument against the initiation of ECT in these cases [110]. Conversely, severe depression characterized by catatonic or psychotic features has been associated with an increased likelihood of a positive response, as indicated by an odds ratio of 1.7 [111]. Additionally, the interplay between these traits and personality disorders could influence the response to ECT, potentially exemplifying Simpsons paradox.

These findings present a complex decision-making scenario for psychiatrists and patients contemplating ECT. Access to current, individualized predictions regarding ECT response, which consider personality disorders and catatonic or psychotic traits, would be a significant asset in this decision-making process. However, since ECT is infrequently used, amassing sufficient data to train predictive models poses a substantial challenge for data scientists and researchers in the field of ECT. To achieve accurate predictive outcomes, data amalgamation from multiple healthcare institutions is necessary.

In our proof of concept, we have amalgamated patient data from UMC Utrecht and St. Antonius Hospital to train algorithms designed to assist in addressing this issue. Traditional algorithm training methods, prior to the enhancements described in this paper, necessitate that data be collected and algorithmic models be developed by data scientists within each institution's computational framework, as established by IT professionals. Additionally, such collaborative efforts require patient consent for data shar-

ing, along with the establishment of data transfer agreements and ethical approvals, managed by principal investigators, privacy officers, and ethics committees at each participating institution. This process is renowned for being labor-intensive and protracted, often extending over several months or even years [112, 113].

5.4 PoC Framework, Workflows and Policies

In this study, we conducted a Proof-of-Concept (PoC) using a federated computational infrastructure that involved SURF, a network for education and research, Utrechts research medical center, and St. Antonius hospital. Each of these entities contributed to the establishment of a collaborative platform tailored for the analysis of ECT data.

The research centers, Utrecht and St. Antonius, each maintained a portion of the ECT dataset on their premises, ensuring the ability to independently process data within their domain. SURF operated as a neutral third party, providing computational resources without direct access to the data. This infrastructure allowed the two hospitals to safely combine their analytic outputs into a comprehensive result, securely stored within the SURF domain.

This system was designed to adhere to stringent policies specific to the project while maintaining the capacity to accommodate a broad spectrum of data analysis regulations. These ranged from collaborative data processing across domains with unique data-access limitations to the enforcement of data access rules, such as those based on patient consent, and preserving local autonomy where participating organizations retain control over their data and its applications.

The analytical process was divided into discrete tasks organized within a computational workflow. Tasks were allocated to either the local hospital domains or the central SURF domain, depending on the requirements of the task and the policies in place. The framework meticulously ensured compliance with all necessary protocols before initiating the data processing pipeline. Additionally, ongoing checks were performed to align with hospital data-access policies during runtime.

Policy enforcement was facilitated through the deployment of a policy reasoner, a com-

ponent capable of interpreting and applying the policies to govern task execution and data access within its domain. This reasoner, which could be complex in nature, had the potential to incorporate normative concepts and reason about them. Not only did it uphold institutional policies regarding task execution and data access, but it also supported low-level networking policies that safeguard security and privacy.

Virtualized Network Functions (VNF) can be applied within the system, which allows for the creation of security measures such as firewalls, encryption, and Intrusion Detection Systems (IDS) on an as-needed basis. This innovation considerably alleviates the IT officers' burden, as it circumvents the need for intricate negotiations associated with establishing Virtual Private Networks (VPNs).

The policies within the framework varied in complexity, from straightforward allow/-deny rules to more elaborate stipulations governing federated machine learning processes. During these processes, the central node (SURF) distributed a global model to the hospitals, which then executed training iterations on their local data. Subsequently, they returned their local models to SURF for the integration into an updated global model.

Policies were crucial throughout this federated learning process, ensuring that hospital nodes only performed tasks related to their local analysis, only shared model parameters rather than sensitive datasets, and that SURF allowed access to the aggregated global model exclusively to approved entities, thus maintaining the integrity and privacy of the data involved.

5.5 Federated Analysis and Learning

In the initial use case, a data scientist collaborates with psychiatrists and engages with various hospitals to access necessary datasets for a specific analysis. The hospitals may agree to allow the data scientist to conduct certain analyses, on the condition that the patient data remains within the hospital's control, thereby ensuring that direct access to the data is not given to external researchers and only the agreed upon analysis is carried out. The data scientist is then provided access to the outcomes of a trained

model, rather than the raw data itself, thereby preserving the privacy of the individual patients involved. The system is designed to support privacy-preserving federated learning techniques to further bolster the protection of patient privacy. *Federated Learning (FL)* [27] is a decentralized scheme to train *Machine Learning (ML)* models on several nodes, each being able to contribute compute resources and/or their own private dataset. In most common FL scenarios, local models are trained at participant sites and model parameters are sent to a trusted orchestrator to be fused into one set of parameters and sent back to participants. Among other advantages, FL contributes to preserving privacy of participants by negating the need to share their private data with each other or the orchestrator.

This approach is tailored to provide a clear and precise method to assess the impact of personality disorders on the likelihood of a positive response to ECT, while adjusting for the presence of catatonic and psychotic features. The method utilizes anytime-valid confidence intervals as described by [112]. These intervals can be understood as similar to traditional confidence intervals, but they are especially suited for federated environments and allow for the ongoing observation of the research ("optional stopping") without compromising the validity of the confidence intervals.

To compute anytime-valid confidence intervals, a "safe test value" is calculated for each patient, or for a very small cluster of patients within one hospital. This value is then aggregated across all the patients within a hospital to generate a local result and further combined across all participating hospitals to produce a global result. The multiplication of these safe test values allows for the continuous updating of confidence intervals, providing a robust statistical tool that adapts to the iterative nature of federated learning, and maintains the integrity of the study even when data analysis is performed at multiple points throughout the study duration.

5.6 Why federated ML is not always the right answer?

Despite the prevalence and effectiveness of FL and its numerous advantages, it has been shown that FL by itself does not guarantee privacy and has been known to be vulnerable to different adversarial attacks such as *Membership Inference Attacks* [114, 115], *Reconstruction Attacks* [116, 117] and *Inversion Attacks* [118, 119].

Among methods proposed to provide measurable guarantees of privacy, *Differential Privacy (DP)* is capable of providing algorithmic guarantees of privacy against linkage attacks [120, 34]. In the context of FL, one can employ Global DP - when the participants trust the orchestrator and privacy is provided on participant level - [121] or Local DP - when there is no trust between the orchestrator and participants and privacy guarantees are provided on a record level [122].

Although adopting FL and DP can lead to both flexibility in terms of distributed training of ML models and privacy guarantees, they also incur costs such as computational overhead [123], reduction of performance [121] and negative impact on fairness [104]. The latter two problem are especially prominent in a federated learning scenario where distribution among nodes is Non-IID, i.e., the distribution of the data on local FL nodes is different than the distribution of the global dataset. Formally, consider the federated learning setup with n participants for a discriminative ML use-case. Training dataset X consists of samples $(x, y) \sim P(x, y)$ with x being the input feature vector and y being the target feature we are aiming to predict. Non-IID data for participant i includes target feature non-IID ($P_i(y) \neq P(y)$) and non-target feature non-IID for feature x_j ($P_i(x_j) \neq P(x_j)$).

In this section, we investigate the effect of non-IID data on differentially private federated learning of a discriminative model. We focus on utility - as defined by common metrics such as F1-score, accuracy, etc. - and fairness of the model. We will establish baselines for centralized DP and non-DP FL models using metrics for utility and fairness of the model output in presence of the skewness in distribution of target and non-target features. Then we extend our experiments to the case of FL and measure

the impact of non-IID data among different nodes on utility and performance of the model.

5.6.1 Related Work

Impact of DP on model performance and fairness. Bagdasaryan et al. [102] empirically show that in both centralized and federated settings, adding DP to the deep model will exacerbate any potential "unfair" traits of the model towards underrepresented groups. They show this disparate impact on vision and language models. Gu et al. [124] research the impact of DP on model fairness in FL and conclude that although both the noise adding mechanism and the gradient clipping step have regularization impact on the outcome of the deep model, they also have a negative impact on the fairness of the model for underrepresented classes. [33] extensively research the impact of DP on machine learning in healthcare and empirically analyze the trade-offs between privacy, utility and fairness. They show that applying the DP-SGD mechanism [27] to machine learning models in healthcare has a negative impact on the robustness and fairness of the model in presence of classes and labels with long-tailed distributions. They also show that DP disparately impacts group fairness by looking at loss of influence for majority groups.

Impact of data distribution on utility and fairness. Zhao et al. [125] show steep degradation of machine learning utility in federated setting in presence of non-IID and highly skewed data. Farrand et al. [104] empirically show that adding DP to a centralized model will have disparate impact on both utility and fairness of the model outcome, even in presence of slight class imbalance. They further show that the disparate impact of adding DP is not limited to the high-privacy regime and also can be observed in low-privacy settings. Finally they conclude that increasing the privacy level will result in loss of utility across all classes, which makes the model less efficient but more fair. Ozdayi and Kantarcioglu [126] research the impact of data distribution on fairness and robustness of FL and report the negative impact of non-IID data on model performance and fairness. They also conclude that the impact of non-IID data on fairness is far greater than its impact on model utility.

Our work focuses on the impact of non-IID data on a federated, differentially private model. To the best of our knowledge this is the first work to address the interplay between federated learning, differential privacy and non-IID data.

5.6.2 Methodology

In this work we measure the impact of non-IID data on the performance of a differentially private model trained in a federated fashion.

Dataset

To understand the implications of having non-IID data in a DP FL setup, we chose the *Census-Income (KDD)* dataset from UCI Machine Learning Repository [127], which has been extensively used in research literature on fairness [128, 129, 130]. It contains demographic census data of U.S population. The target feature *Salary* indicates whether the person has an annual income of over or under 50k dollars. Protected feature is *Race*.

We perform extensive preprocessing on features such as education, marital status and employment status and group disparate values into high level categories to increase the baseline classification efficiency. The preprocessed dataset contains 30 predictive features - including one protected feature -, one target feature and 248466 samples. Table 5.1 shows the distribution of our protected feature *Race* against the target feature *Annual Salary*.

To measure the impact of DP and non-IID distribution on fairness, we define the group *White* in *Race* feature as the *Privileged Group* and *Annual Salary>50k* as the desired outcome. For the relevant fairness metrics, the rest of the categories in *Race* will be conditioned against the privileged group and the (un)fairness will be measured against the desired outcome.

Salary \Race	White	Black	Asian	Other
≤50K	193,844	23,364	6,641	6,021
>50K	16,398	801	600	197

Table 5.1: Distribution of protected feature *Race* categories over the target class *Annual Salary*

Non-IID Data

To simulate non-IID data in an FL setting, we follow [126] and distribute the samples among different FL participants based on their target feature using Dirichlet distribution. We control the amount of deviation from IID using the concentration parameter of the Dirichlet distribution. Assuming prior binomial distribution q over target feature and a categorical distribution p over target feature from which samples are drawn independently on participant level, we sample p from $Dir(\alpha q)$. Noting that $\alpha \to \infty$ emulates fully-IID data while $\alpha \to 0$ results in fully-non IID data, we choose three parameters for α: 0.1, 1 and 100. These parameters based on our experiments will emulate extreme non-IID, medium non-IID and almost IID data respectively. Figure 5-2 illustrates the distribution of target class and the protected attribute (Race) among 15 participants of our FL scenario with different concentration parameters.



Figure 5-2: Distribution of target feature and the protected class among 15 FL participants. Left column shows the IID data with $\alpha = 100$, middle column shows mild non-IID data with $\alpha = 100$ and right column shows extreme non-IID data with $\alpha = 100$

Metrics

To measure the utility of our experiments, we will report *F1-score*, *Precision* and *Recall*. In research area of fairness in ML, different metrics are defined and used for different usecases. For our binary discriminative model, we have chosen *Equalized Odds Rate* (EOR), *Generalized Entropy Index* (GEI) and *Differential Fairness Bias Amplification* (DFBA).

Differential Fairness Let $P \subset \mathbb{R}^k \times \{0,1\}$ be the input space of a binary classifier model. Consider dataset $\mathcal{X}$ with feature set $x : \{x_1, x_2, ..., x_n\}$ and protected features set $\mathcal{A} \subset x$ and $\mathbf{s}_i, \mathbf{s}_j \in A$ tuples of protected feature values. Randomized mechanism $\mathrm{M} : \mathcal{X} \to \mathcal{Y}$ is ϵ-Differentially Fair (DF) with respect to (A, Θ) if for all $(\mathbf{s}_i, \mathbf{s}_j) \in A \times A$ and $\mathbf{x} \sim \theta$:

$$e^{-\epsilon} \leq \frac{P_{M,\theta}\left(M(\mathbf{x}) = y \mid \mathbf{s}_i, \theta\right)}{P_{M,\theta}\left(M(\mathbf{x}) = y \mid \mathbf{s}_j, \theta\right)} \leq e^{\epsilon},$$

for $\theta \in \Theta$ and $y \in \mathrm{Range}(M)$ where $P\left(\mathbf{s}_i \mid \theta\right) > 0, P\left(\mathbf{s}_j \mid \theta\right) > 0$ [131].

Intuitively ϵ-DF bounds the difference in log-likelihood of the probabilities of the outcomes of the randomized mechanism for any combination of protected feature values by ϵ.

To measure the impact of applying ϵ_1-DF mechanism M on ϵ_2-DF dataset $\mathcal{X}$ on the dataset's fairness, we can measure the Differential Fairness Bias Amplification (DFBA) which is the difference between ϵ values: $\epsilon_2 - \epsilon_1$ [131].

Generalized Entropy Index Generalized Entropy Index (GEI) is a measure of inequality, originally defined in the context of economics to analyze the distribution of income and economic (in)equality [132].

For $\alpha \notin \{0,1\}$ and $b_i = \left(y_{predict_i} - y_{label_i} + 1\right)$, with N being the number of individual samples in dataset $\mathcal{X}$ the Generalized Entropy Index with mean $\mu = \frac{1}{N}\sum_N^1 b_i$ is defined as:

$$\frac{1}{N\alpha(\alpha-1)} \sum_{i=1}^{N} \left[\left(\frac{b_i}{\mu} \right)^{\alpha} - 1 \right]$$

GEI is an individual and group level fairness metric and can be thought as the measure

of redundancy in the data against the desired outcome from an information theoretic perspective.

Equal Odds Rate Equal Odds Rate (EOR) [133] measures how close sensitivity of the outcome of the mechanism for privileged groups is to that of unprivileged groups in the protected feature with regard to the desired outcome. Equalized odds is achieved when there is absolute conditional independence between unprivileged groups in the protected feature and model outcome with regards to the desirable outcome.

Formally, mechanism M exhibits absolute equal odds -i.e., is fair - for privileged group G and unprivileged group G' and desired outcome $O \in \{0, 1\}$ if $\mathbb{E}_{(\mathbf{x},y)\sim G}\left[\mathrm{M}(\mathbf{x}) \mid y = O\right] = \mathbb{E}_{(\mathbf{x},y)\sim G'}\left[\mathrm{M}(\mathbf{x}) \mid y = O\right]$

Implementation Details

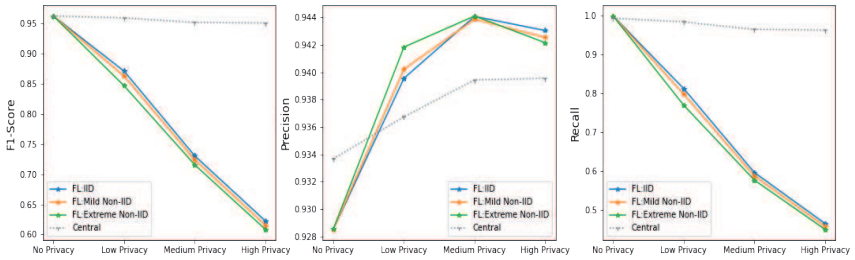
We use a shallow feed-forward network with three fully connected layers. For federated setup, we simulate 15 clients and perform cross-silo FL. All our experiments are implemented using PyTorch[1] v1.10 and Opacus[2] v1.0.0 libraries. Experiments are run on a machine with a Tesla-P100 GPU and 32GB of RAM.

In this work we use *Rènyi*-DP [134] to achieve tighter privacy bounds. We assume no trust between participants and the orchestrator and opt for local DP by performing DP-SGD on each participant. We also assume a lack of availability of any public datasets to pre-train the models and/or to share between participants to mitigate the impact of non-IID data as has been suggested in some recent literature.

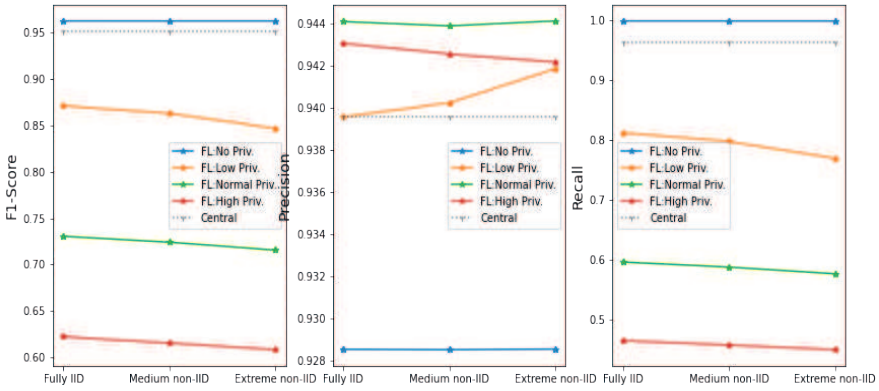
Impact on Utility

Figure 5-3 visualizes the privacy and distribution-utility trade-offs of our FL system. On privacy-utility front, considering the scale of the graphs makes it clear that recall is far more impacted than precision when the level of privacy increases. Looking at the F1-scores it can be deduced that the performance does decrease with increase in privacy

[1] https://pytorch.org/
[2] https://opacus.ai/



(a) Privacy-utility trade-off



(b) Privacy-distribution trade-off

Figure 5-3: Trade-offs between privacy, distribution, and utility. The upper subfigure shows the privacy-utility trade-off, and the lower subfigure shows the privacy-distribution trade-off.

level, with no significant difference between different distribution levels. However, looking at precision and recall graphs, we observe that while recall also drops with increase in privacy, precision shows an improvement which keeps its trend except for the high privacy regime. This could be contributed to the regularization effect of noise-adding mechanisms of DP. Although there is negligible drop of performance for the central model as we increase the privacy level, the positive impact of DP on precision can still be observed.

We have visualized the distribution-utility trade-off on the bottom row of Figure 5-3. We observe that going from fully-IID to extreme non-IID data, there is a drop in F1-

Score and recall for privacy preserving regimes. However, while we see the same trend in precision for non-DP case and the high privacy regime, especially in case of low privacy regime we can see an increase in precision going from fully-IID to extreme non-IID. We can also contribute this to the regularization effect of DP, which decreases the number of samples identified positively by the algorithm in this binary classification experiment while increasing the number of samples incorrectly identified as negative.

Impact on Convergence Rate

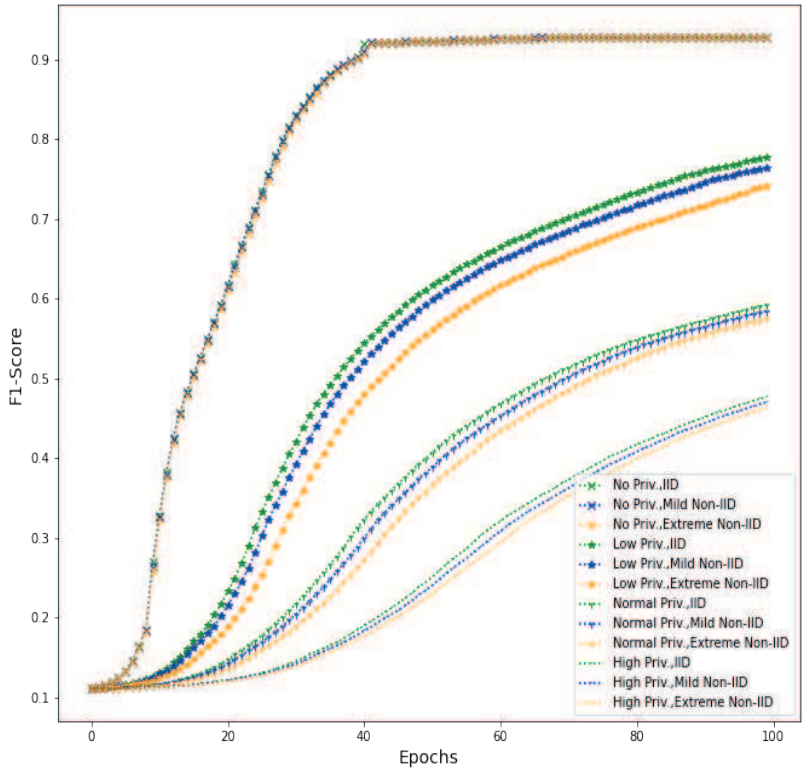


Figure 5-4: F1-score of FL models with different privacy levels and distributions for 100 epochs.

Figure 5-4 depicts the impact of different privacy levels and distributions on the perfor-

mance of our federated learning system. As can be observed, with non-DP setup the data distribution has no discernible impact on the performance and the convergence rate of the final model. However, while in private setups the performance is negatively impacted, the data distribution has no significant impact on the convergence rate of the model.

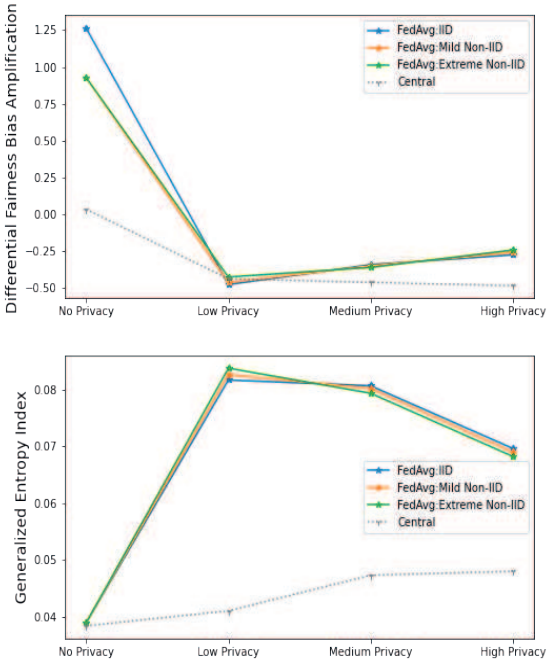


Figure 5-5: Privacy trade-off against fairness

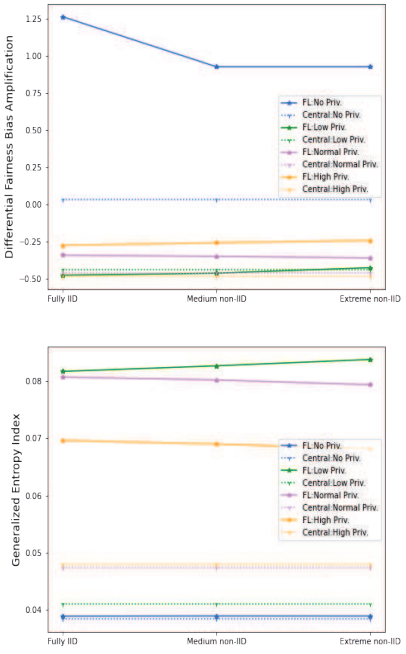


Figure 5-6: Distribution trade-off against fairness

Figures 5-5 and 5-6 visualizes the privacy and distribution-fairness trade-off of our FL system for dataset level fairness metrics of differential fairness bias amplification and generalized entropy index. We observe that for centrally trained model, increasing the privacy level has a negative impact on both DFBA and GEI. However for the FL scenario the behavior of the system with regards to these two metrics is different. GEI is increased for all distribution scenarios when we go from no privacy to low privacy regime. The observed increase in the GEI in the Federated scenario when transitioning

from no privacy to a low privacy regime reflects an intrinsic interplay between privacy mechanisms and the inherent data diversity across multiple decentralized nodes. In our FL scenario we have assumed the data to be non-IID. When no privacy is applied, each node's model training may closely fit its local data, possibly leading to overfitting or a lack of generalizability. Upon introducing privacy enhancements, such as noise addition or parameter averaging, there is a regularization effect which can help to smooth out model predictions across the diverse data sets. This regularization often prevents extreme model decisions that heavily favor one group over another, leading to a more uniform distribution of errors across groups, thereby increasing the GEI, which measures inequality in outcome distributions among different demographic groups. This increase indicates a shift towards more equal treatment of groups within the dataset, despite potential trade-offs in individual model accuracy or performance. Thus, the transition to a low privacy regime introduces a balancing act between maintaining individual privacy and achieving a model that is fairer across different groups.

Increasing the privacy results in slight GEI drop, meaning that although the model is performing worse overall in terms of utility, increasing the DP reduces its negative impact on fairness by around 0.01. As for DFBA, we observe that in no privacy regime the classifier is causing bias to increase, while adding any measure of DP to the model will reduce the classifier bias to less than dataset bias. Increasing the DP level decreases this reduction in bias, but even in high privacy regime DP reduces the classifier bias to a level less than the dataset bias.

As for fairness-distribution trade-off, DFBA decreases as we go from fully IID to extreme non-IID data. In other privacy regimes, the DFBA stays the same for different data distributions. GEI stays the same for no privacy regime and increases slightly for low privacy regime, but shows a moderate decrease in medium and high privacy regimes which could be attributed to the regularization effect of DP on model fairness. Consistent with our previous observations, the behaviors reported in GEI and DFBA across different privacy regimes can be attributed to the interaction between differential privacy levels and data heterogeneity. In cases of non-IID data, the discrepancies in data distribution among various nodes lead to diverse local models, which, when aggregated, can result in a model that does not necessarily perform uniformly

across all data segments. Initially, in no privacy or low privacy regimes, the slight increase in GEI indicates that introducing mild noise through DP can help to mitigate overfitting to specific local data distributions, thus promoting a more balanced error distribution among different groups and slightly enhancing fairness. As privacy constraints are tightened (medium to high privacy regimes), the stronger DP mechanisms impose more substantial noise addition or more aggressive parameter averaging. This increased regularization can overshadow the model's ability to capture and reflect nuanced data characteristics, leading to a deterioration in model performance and a moderate decrease in GEI, as specific group characteristics are overly smoothed out. Thus, the interaction between the level of DP applied and the degree of data heterogeneity inherent in non-IID FL setups crucially influences the trade-offs between model utility and fairness. This perfectly encapsulates the different trade-offs discussed in this section.

Impact on Group-Level Fairness

Figures 5-7 and 5-8 show the impact of privacy and data distribution on EOR for underprivileged groups. We observe that adding privacy to the model has a small negative impact on the fairness of the model according to EOR. It is less prominent for the *Asian* group as their class imbalance is less severe than the other underprivileged groups. However, in the federated setup, regardless of data distribution, the negative impact of privacy is more prominent as we increase the privacy level. It is much more severe for the *Other* ethnicity group which is both under-represented in number of samples and in percentage of samples with desirable outcome.

In terms of distribution-fairness trade-off, we can observe that with no privacy the impact of data distribution on EOR is negligible. In private regimes the level of non-IID data has a negative impact on EOR fairness. The regularization effect of high levels of DP can still be observed, as the negative impact on EOR is dampened going from low privacy to high privacy level.

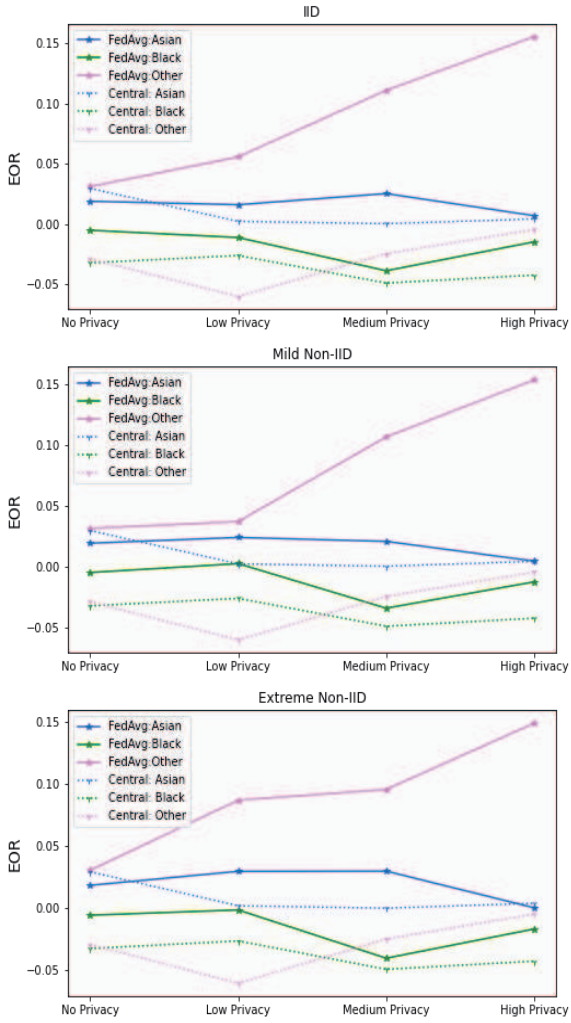


Figure 5-7: Distribution trade-off against fairness. EOR is measured for each under-privileged group against the designated privileged group and for the desirable outcome of *Annual Salary* > *50k*

5.6.3 Mitigation Solutions and Strategies

In this section, we performed an extensive empirical analysis of the impact of non-IID data on the utility and fairness of differentially private federated deep learning models. Employing a real world dataset ubiquitous in ML fairness research with all three types of imbalance - class, label and node, the latter added during FL simulations - we explored the trade-offs between utility, privacy and data distribution. We showed that DP has

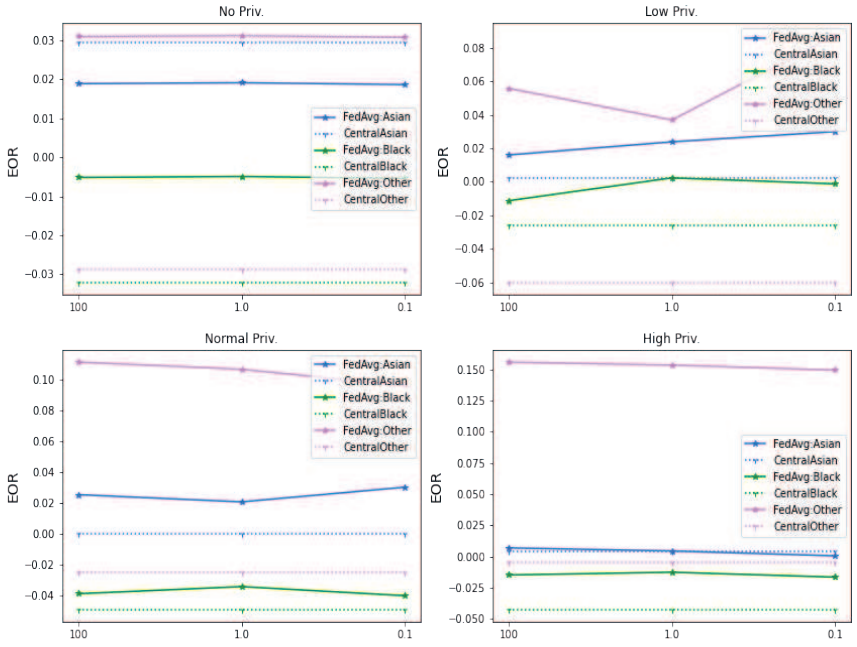


Figure 5-8: Privacy trade-off against fairness. EOR is measured for each underprivileged group against the designated privileged group and for the desirable outcome of *Annual Salary* $> 50k$

a generally negative, although disparate, impact on both utility and performance of discriminative deep models for underprivileged groups. We also showed that non-IID data deepens the utility and fairness gap between minority and majority groups. We chose our utility and fairness metrics to enable us to explore the different aspects of non-IID data impact on our privacy preserving federated experimental setup to the fullest. Specifically differential fairness by definition has an organic connection to differential privacy, which encourages us to further explore how we can utilize this connection, e.g., providing lower bounds for differential fairness degradation by considering the privacy budget and non-IID aspects of the data.

Although consistent with our general intuitions about the impact of DP and non-IID data, this work is a limited exploration of the interplay between fairness and utility and their trade-off with privacy and data distribution. We are currently working on

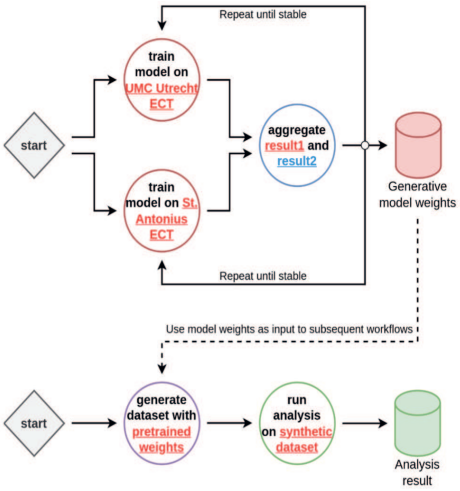


Figure 5-9: Optimized pathways for DHT through synthetic data generation. We first train a differentially private generative model in a federated fashion. Then, instead of performing the analysis on the ECT datasets directly in a federated manner, we generate a synthetic dataset and perform our analysis on the synthetic data. We also perform our analysis directly on the real data and compare the results.

extending our experiments with a wide variety of datasets and more complicated models, e.g., large scale vision models. It would allow us to better understand the impact of non-IID data in privacy preserving FL.

While employing more complicated FL algorithms other than the vanilla *FedAvg* -e.g. ,*FedProx* [135], FairFed [136] - might have a positive impact on the utility and fairness degradation in non-IID FL setups and need to be assessed carefully, most still do not directly deal with the underlying problem of non-IID distribution of data in federated settings. As we have shown in this section, in presence of non-IID data, the utility and fairness of the privacy preserving discriminative models are negatively impacted which leads us to the conclusion that breaking the IID assumption hinders FL, and this must be carefully monitored when FL is deployed.

5.7 Federated Synthetic Data Generation

In the domain of data analysis within machine learning in DHTs, the federated approach explained in 5.5 enables analyses that were previously impractical. However,

this method involves complex negotiations regarding access conditions and the establishment of necessary safeguards, which may prove to be laborious. Furthermore, the application of federated machine learning architectures may not be universally appropriate due to challenges such as limited data availability or non-IID data as discussed earlier in this chapter. To address these limitations, an alternative approach is proposed.

The secondary approach introduces an extra phase in the process, which involves the generation of synthetic data derived from the original datasets required for the analysis, as outlined in the workflow depicted in Figure 5-9. The introduction of this additional step, although technically intricate, yields governance advantages. The use of synthetic data ensures the privacy of patient information, providing an additional safeguard against the risk of data exposure. This approach allows for the standardization of synthetic data generation policies, which significantly streamlines the negotiation process and facilitates a high degree of automated data processing. Such an advancement has the potential to transform the practices of medical professionals and data scientists. Traditionally, models are trained and evaluated only once to preserve the validity of patient consent and adherence to policy regulations. In contrast, the proposed secondary scenario enables a more flexible and iterative approach to model development, aligning with the preferences of health researchers for incremental analysis and training (Menger et al., 2016).

It is important to note that it is not viable to produce a universal synthetic dataset that would be suitable for all conceivable analyses simultaneously. Rather, a synthetic dataset should be generated specifically for the analysis in question, taking into account the identified data sources and the pertinent correlations necessary for model training. This tailored generation of synthetic data necessitates that each time there is a modification to the model or the datasets involved, the synthetic data must be reconstituted and undergo validation to ensure its continued relevance and accuracy.

In the realm of machine learning and data analysis, the first approach facilitates analyses that were once not feasible, but it is accompanied by intricate negotiations concerning data access and the establishment of necessary protocols, which can be a tedious process.

Additionally, employing federated learning structures may not be universally applicable due to limitations such as restricted data availability or the challenge of handling non-independent and identically distributed (non-iid) data ([36]). To address these concerns, we propose an alternative methodology.

This alternative methodology incorporates an incremental step of synthesizing data from the original data intended for analysis, as depicted in Figure 5-9. This additional procedural layer, though increasing technical complexity, offers substantial benefits at the governance level. Synthetic data ensures the confidentiality of patient information, even in scenarios where direct control over the data might be compromised, thereby providing an augmented safeguard. Consequently, the processes for creating synthetic data can be standardized, streamlining negotiations and enabling highly automated data processing. This development could significantly alter the workflows of both physicians and data scientists. Presently, models are generally trained and evaluated a single time due to the potential invalidation of patient consent and compliance with policies if changes are made. The approach we introduce permits agile model development, allowing for iterative analysis and training in a stepwise manner, which aligns with the preferences of health researchers (Menger et al., 2016). It is imperative to acknowledge that it is not practical to produce a universal synthetic dataset that is appropriate for all conceivable analyses in one instance. Synthetic datasets must be specifically generated for the analysis at hand, considering the selected data sources and essential correlations for model training. This means that synthetic data creation and validation are necessary whenever there are updates to the model or changes in the required datasets.

In this second scenario, a privacy-preserving generative model is trained using federated learning techniques on the ECT datasets across multiple hospitals. The model is designed to learn the statistical attributes of the datasets without retaining or revealing any identifiable patient information. The application of differential privacy ([34]) to the model provides strong mathematical assurances that the synthetic data produced cannot be used to deduce or reveal information about the original data (Dyda, Amalie, et al., 2023; El Ouadrhiri, 2022). The data scientist architects and trains the model in a federated fashion. Post-training, there is a quantitative assessment to verify that the statistical properties of the synthetic data correspond to those of the actual data, while

also ensuring that the privacy-preserving capabilities of the model are intact.

An optional step involves domain experts conducting a qualitative review to ascertain the quality of the synthetic data. Upon validation and confirmation that the synthetic model meets the necessary criteria, it can be employed to generate synthetic data for research purposes. In our study, to validate the efficacy of the trained models, we employ multiple classifiers trained on different datasets: one using the synthetic data and another using the real ECT data through a federated approach (the direct method). The performance of these classifiers is evaluated using two metrics: accuracy, which measures the proportion of correct predictions made by a model, and the F1 score, which is a harmonic mean of precision and recall, both of which are essential measures in the field of machine learning.

5.7.1 Results

In our research, we conducted a comparative study to assess the performance of classifiers using both real and synthetically generated data, and the findings offer valuable insights as indicated in Figure 5-10. The classifiers leveraging synthetic data outperformed those trained with real data, achieving an accuracy of 0.333 and an F1 score of 0.417. In contrast, the classifiers using real data exhibited lower performance metrics, with an accuracy of 0.238 and an F1 score of 0.385.

The enhanced performance observed with synthetic data in this specific classification task suggests that the data synthesis process effectively addresses several prevalent issues encountered in real-world medical datasets. Such issues include class imbalances, missing values, and various types of data noise that can significantly impact model performance. The generative model, when meticulously designed, can approximate the underlying probability distribution of the original data. This allows the model to either implicitly or explicitly (as seen with conditional generative models) refine the data quality, thereby rectifying the aforementioned complications.

In the context of this study, the improvement in data balance was instrumental in training a classifier that was more proficient. It is important to note, however, that the degree of enhancement achieved through synthetic data is contingent upon several

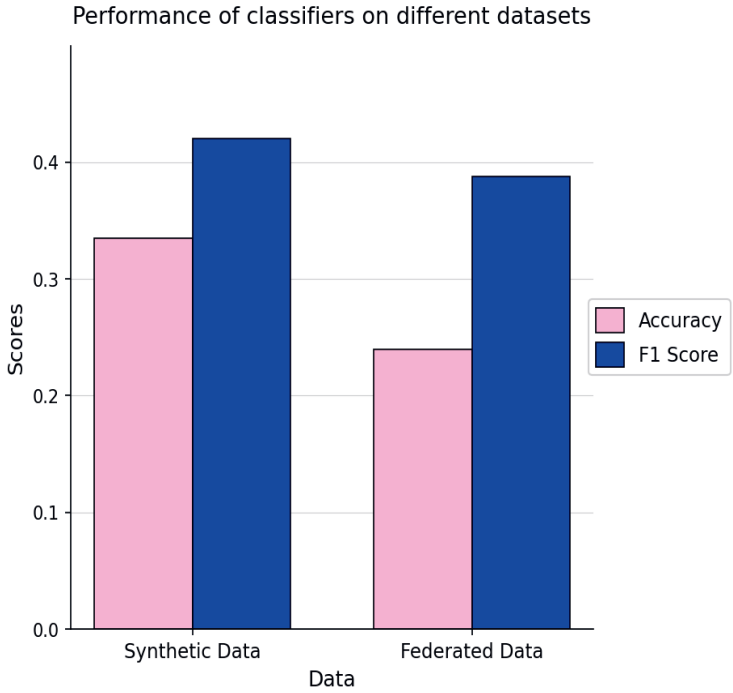


Figure 5-10: Comparative Performance Metrics of Classifiers Trained on Real and Synthetic Datasets. This bar chart illustrates the performance of classifiers trained on synthetic data and real data (comprising joint datasets from St. Antonius and UMCU). Performance is measured via two metrics: accuracy and F1 score. The classifiers trained on synthetic data demonstrate superior performance.

factors related to the specific use case. These factors include the inherent quality of the data, the selection of the generative model, and the particular characteristics of the medical conditions represented in the data, which might include outliers or inherent biases.

5.8 Summary

In this chapter, we documented our efforts to integrate the methodologies developed in the preceding chapters into a cohesive proof of concept for a Digital Health Twin (DHT) framework, specifically optimized for clinical pathways. This chapter serves as a culmination of our research, where we attempt to apply the tail-adaptive normalizing flows, the model-agnostic framework for learning semantic constraints, and the noise-

less differential privacy approach in a unified, optimized framework. The goal is to demonstrate the practical viability and effectiveness of our proposed methods in a real-world setting, thereby advancing the application of DHTs in the healthcare industry.

CHAPTER
6

DISCUSSION AND CONCLUSION

This thesis addresses the critical challenges in realizing Digital Health Twins (DHTs) within distributed data environments. Such environments are inherently constrained by both infrastructural limitations, including communication latency and bandwidth restrictions, and regulatory mandates, such as data access consents and stringent privacy policies. These constraints fundamentally impede the direct application of conventional data analysis and machine learning methodologies to sensitive, geographically dispersed health data.

To overcome these obstacles, this work investigates the potential of generative models for data synthesis as a transformative paradigm. Specifically, we posit that replacing direct reliance on real-world patient data with meticulously synthesized, statistically representative surrogates offers a viable pathway to enable robust data analysis and machine learning within the DHT framework, while inherently respecting privacy and access limitations.

Recognizing the multifaceted demands of secure, private, and distributed DHT implementations, and building upon the state-of-the-art in generative data synthesis, we identified a set of key open research questions. These questions are centered on the core challenge of creating synthetic data that is not only statistically faithful but also practically useful within the demanding context of digital health. This chapter will articulate

concrete answers to these research questions, contextualize and critically evaluate the proposed solutions, and delineate promising avenues for future investigation, thereby charting a course for impactful advancements in this domain.

6.1 RQ1: How can generative models estimate densities exhibiting heterogeneous tail behaviors?

Addressing the challenge of accurately modeling tail behavior is crucial for generating useful synthetic data, particularly in domains like healthcare where extreme events can be highly significant. In Chapter 2, we tackled this research question by introducing a specific methodology for training normalizing flows designed for data exhibiting mixed-tail characteristics (i.e., distributions with tails that decay at different rates).

Our approach diverges from methods that either assume a specific tail type or attempt to explicitly estimate it beforehand. Instead, we focused on the normalizing flow's base density. We proposed using a base density formed by a mixture of Generalized Gaussian Distributions (GGDs). This choice provides inherent flexibility, as GGD mixtures can approximate a wide variety of tail behaviors, from lighter-than-Gaussian (sub-exponential) to heavier-than-Gaussian (super-exponential), without needing prior specification. We combined this flexible base density with a standard Maximum Likelihood Estimation (MLE) training procedure.

A key aspect of this design is that it allows the base density's tail properties to adapt during training without being rigidly forced to match the target distribution's tails exactly. This decoupling grants the model greater capacity to learn complex, potentially mixed-tail distributions often found in real-world data, overcoming a common limitation of standard normalizing flows that typically rely on Gaussian or other fixed-tail base distributions.

We evaluated this method systematically. Initial experiments used a 2D synthetic dataset specifically constructed with mixed tails. These experiments demonstrated that our approach captured the tail behavior significantly better than baseline normalizing flows using conventional base densities and training. We then extended this evaluation

to real-world datasets known to possess mixed-tail properties, comparing our method against other contemporary data synthesis techniques.

Across these experiments, our proposed method consistently generated synthetic data that not only matched the overall distribution well but also more accurately reflected the specific tail characteristics of the target data. Based on these results, we conclude that the proposed normalizing flow architecture, incorporating a GGD mixture base density and trained via MLE, provides an effective and scientifically grounded solution for the research question: it enables the synthesis of data that faithfully reproduces the mixed-tail behavior present in complex, real-world datasets.

Discussion While this tail-adaptive normalizing flow method shows considerable promise, several practical considerations and potential limitations must be acknowledged. As with any probabilistic generative model, its performance depends heavily on the amount and quality of the available training data.

One significant concern, especially when tail events carry high importance, is the potential for tail underestimation. If the training data is sparse in the extreme tails, or if the model's capacity is insufficient, the generated synthetic data might underestimate the probability of rare but consequential events. This could lead to overly optimistic conclusions in downstream applications, such as risk modeling or anomaly detection. Although our GGD mixture base is designed to be flexible, careful model selection, considering the trade-off between model complexity (expressivity) and computational cost, remains necessary to mitigate this risk.

Conversely, there is also a risk of overfitting specifically to the tail behavior observed in the training sample. A model that captures the training data's tails too closely might not generalize well to new data with similar, but slightly different, tail characteristics. Standard practices like careful cross-validation, appropriate regularization, and systematic hyperparameter tuning are essential to balance tail fidelity with generalization capability.

Furthermore, the inherent nature of tail events poses a statistical challenge: they are, by definition, rare. This sparsity of tail data means that estimating tail properties from

empirical data is intrinsically subject to higher variance than estimating properties of the distribution's main body. This can affect the reliability of any method, including ours, that learns from these sparse observations. While our experiments suggest the proposed model handles mixed tails reasonably well even with limited tail data, the outputs, especially concerning rare events, should ideally be validated by domain experts before deployment in critical applications.

Future Directions The performance of our method, particularly the observation that increasing the number of GGD mixture components improves results up to a point, suggests avenues for future work. One natural extension is to explore the use of infinite mixture models for the base density. Such models could, in principle, adapt their complexity automatically based on the data, potentially removing the need to manually select the number of mixture components and offering a more principled way to handle varying levels of distributional complexity.

Additionally, to better capture potentially intricate real-world distributions that might exhibit multimodality or asymmetry even within the tails, investigating a mixture of mixtures architecture for the base density could be fruitful. This hierarchical structure might offer even greater flexibility in modeling complex tail dependencies, potentially improving synthetic data fidelity further and broadening the applicability of this approach to even more challenging data synthesis problems.

6.2 RQ2: How can generative models synthesize data with high fidelity and diversity while respecting the implicit semantic boundaries of the real data?

The generation of synthetic data often prioritizes statistical similarity to the original data. However, for many applications, particularly those involving sensitive information or critical decision-making, this is insufficient. Semantic coherenceadherence to both

explicitly stated rules (e.g., variable ranges) and implicitly learned constraints (e.g., plausible relationships between features)is essential for the synthetic data to be truly useful. Chapter 3 addressed this research question by introducing a framework designed to produce tabular data that is both diverse and semantically sound.

We approached this challenge by formulating the problem as an unsupervised constrained optimization task. This formulation acknowledges that semantic boundaries in real-world data are often not perfectly defined and must be learned. The proposed solution is a modular architecture consisting of two main components operating in a loosely-coupled manner: a generic probabilistic synthesizer and a latent variable validator. This separation allows for distinct handling of the exploration-exploitation aspects of generative modeling.

The synthesizer's objective is to generate diverse samples, effectively exploring the potential data space. Simultaneously, the validator's role is to learn the implicit semantic boundaries present in the training data distribution. A key feature of the framework is that the semantic constraints learned by the validator are applied as soft regularization during the synthesizer's training. This feedback loop guides the synthesizer's exploration, ensuring that the generated samples tend towards semantic validity as defined by the validator, thereby balancing diversity with coherence. We emphasize that this architectural approach is general; the specific synthesizer and validator models used in our work can be substituted with other suitable latent variable models, offering flexibility for various data synthesis tasks.

To assess the effectiveness of this framework, we developed specific metrics aimed at quantifying the trade-off between exploration (diversity) and exploitation (semantic adherence). We benchmarked our approach against established techniques for tabular data synthesis. The empirical results showed that our framework achieved measurable improvements not only in standard statistical fidelity metrics but also, importantly, in the semantic integrity of the generated data. These findings suggest that the proposed framework offers a principled method for generating synthetic tabular data that is both statistically representative and semantically valid.

Discussion The need for semantic integrity, particularly in fields like medicine, goes beyond simple statistical matching. It requires preserving explicit rules (e.g., physiological constraints, valid demographic combinations) and, more subtly, the implicit semantic relationships within the data. These implicit relationships might include correlations between patient characteristics and disease likelihood, dependencies between treatments and outcomes, or complex interactions among medications. Achieving semantic integrity thus requires generative models to learn and reproduce this network of relationships that defines the data's meaning within its domain.

Our proposed method attempts to address this by learning implicit boundaries from unlabeled data and enforcing them during synthesis. However, the probabilistic nature of both the validator and the synthesizer introduces nuances. Careful validation is needed to ensure the learned semantic constraints are appropriatenot so strict that they stifle diversity and exclude valid but uncommon data patterns, nor so loose that they permit the generation of semantically flawed instances.

While explicit rules are relatively straightforward to check, validating implicit semantic correctness remains challenging. Many important constraints, especially those involving complex interactions (e.g., drug interactions, gene-environment effects), may not be formally documented or fully understood even by domain experts. To address this, improving the transparency and explainability of the validator component is important. An interpretable validator would allow domain experts to examine the learned semantic boundaries, compare them against their knowledge, identify potential inaccuracies or omissions, and potentially help refine the validator by incorporating previously unstated rules. This kind of expert-in-the-loop validation is likely crucial for building trust and ensuring the practical utility of synthetically generated data, especially in high-stakes applications.

Furthermore, the inherent trade-off between exploration (diversity) and exploitation (semantic constraints) requires careful consideration during model configuration. The adjustable feedback strength in our framework allows users to tune the validator's influence, balancing these competing goals according to specific application needs. Incorporating domain knowledge more directly, perhaps by designing specialized validators

focused on particular subsets of features or known semantic relationships (inductive bias), could also improve the precision of semantic boundary enforcement, tailoring the synthesis process more closely to the requirements of complex datasets.

Future Directions A key area for future research is enhancing the explainability of the validator component. Although the current framework improves semantic integrity, the validator largely functions as a black box, which can limit user understanding and trust. Moving towards validator models that are inherently more interpretable would be a valuable step.

Future work could explore using rule-based systems, decision tree ensembles, or neural networks with attention mechanisms as validators. Such models might represent the learned semantic boundaries in forms more accessible to humans, such as explicit rules, feature importance scores, or visualizations highlighting feature combinations deemed semantically valid or invalid. Developing tailored visualization techniques to help users understand the high-dimensional semantic landscape learned by the validator is also important.

Such tools would enable domain experts to more effectively inspect, validate, and potentially refine the learned semantic rules by integrating their own knowledge. Improving validator explainability is not just an academic exercise; it is essential for fostering confidence in synthetic data generation methods, promoting their adoption in critical applications, and enabling more transparent and accountable data-driven practices in complex fields like healthcare.

RQ3: How can Normalizing Flows be released under (ε, δ)-differential privacy while preserving density fidelity without resorting to private stochastic optimization?

Ensuring data privacy while preserving the utility and specific characteristics (like tail behavior and semantic structure) of the original data is a central challenge in synthetic data generation. Chapter 4 addressed this by proposing an alternative approach to achieving differential privacy (DP) in the context of normalizing flows, moving away from common techniques like adding noise during training (e.g., DP-SGD).

Our method leverages the structure of normalizing flows to *calibrate a randomized release* via their exact log-densities while keeping training deterministic. The core idea is that DP guarantees for the *synthetic data release* can be met by carefully limiting the capacity of the transformations (yielding uniform bounds on per-example log-likelihoods) and then sampling from a tempered distribution proportional to $p_\theta(x)^\lambda$ (an exponential-mechanism sampler calibrated by the resulting sensitivity). This achieves privacy at release time without injecting external noise into optimization; model parameters are used to define the sampler but are not released.

This training-noise-free, release-private mechanism offers a distinct alternative for generating private data using normalizing flows. A primary benefit is the potential for improved model interpretability; because privacy is effected by architectural constraints (e.g., base truncation and Jacobian control via scaling or spectral normalization) rather than stochastic perturbations during training, the model's behavior remains analyzable with standard tools. Our empirical results indicated a favorable trade-off between the level of privacy achieved (measured by the privacy budget ε) and the statistical fidelity of the generated data.

We validated this approach through experiments using various normalizing flow architectures, including volume-preserving, affine coupling, and residual flows. These experiments, conducted under different privacy settings, demonstrated the method's ability

to generate tabular data that satisfied differential privacy requirements for the released samples while maintaining a good level of quality compared to the original data. These findings support the potential of using expressivity-constrained normalizing flows as a viable method for privacy-preserving data synthesis.

Discussion The proposed framework relies on specific properties of normalizing flows. Because flows are constructed as a sequence of invertible transformations, one can control how much each step alters the distribution. This fine-grained control enables limiting expressivity (via base truncation and Lipschitz/Jacobian constraints) to obtain a uniform sensitivity bound for the release score, which in turn calibrates a DP sampler. This contrasts with noise-based DP methods, where added randomness during training can obscure the learned function. The choice of normalizing flows is therefore integral to the feasibility of our training-noise-free, release-private mechanism.

A critical factor influencing the practical application of this method is the relationship between the required expressivity constraints and the dimensionality of the input data. The constraints needed to ensure a given level of differential privacy tend to become tighter as dimensionality increases, potentially scaling unfavorably. Our experiments on tabular datasets, which typically have moderate dimensionality, suggest this scaling is manageable in such scenarios.

However, for high-dimensional data types, such as images or complex time series, the limits required for strong privacy guarantees might become so severe that they impair the model's ability to capture the true data distribution, leading to a poor privacyutility trade-off. Addressing this dimensionality dependence is crucial for extending applicability. Potential strategies include applying dimensionality reduction before modeling with the private flow, or developing new flow architectures that are more parameter-efficient or possess structural properties (e.g., sparsity, hierarchy) better suited to high-dimensional spaces under DP-calibrated release.

Future Directions Building on this work, several avenues appear promising. While we focused on standard flow types, the core principleexpressivity control to obtain sensitivity bounds and a calibrated releasemay extend to broader flow families. One direction

is to investigate continuous-time normalizing flows (CNFs), adapting the constraints to their dynamics while retaining the same release mechanism. Separately, training a small family of expressivity-constrained flows deterministically and applying a suitably calibrated privatization mechanism at the *model level* (rather than only at the point of data release) is a natural extension.

RQ4: How to perform performant, fair data analysis in a distributed environment by utilizing synthetic data?

The challenges of analyzing data distributed across multiple locations, as envisioned in Digital Health Twin (DHT) frameworks, are significant. Issues like data heterogeneity, privacy restrictions, and fairness concerns complicate direct analysis using techniques like federated learning. Chapter 5 explored whether using synthetic data, generated using the methods developed earlier in this thesis, could serve as a viable alternative for performing effective and equitable downstream analysis in such distributed settings. This investigation was structured as a proof-of-concept study.

We first established a baseline by conducting experiments using standard federated learning. In this setup, machine learning models were trained directly on simulated non-identically distributed (non-IID) datasets held by different participating nodes, without any data sharing. These baseline experiments confirmed known difficulties: the heterogeneity of data across nodes led to measurable decreases in the overall performance (utility) of the globally aggregated model and also exacerbated fairness issues, particularly for subgroups potentially underrepresented in some local datasets.

Having quantified these baseline challenges, we then evaluated the use of synthetic data as an intermediary. We focused on a specific medical use case involving the analysis of clinical pathways. Using a federated setup, we applied the tail-adaptive normalizing flow method (from Chapter 2) to generate a synthetic dataset intended to represent the combined data distribution while preserving privacy characteristics implicitly through

the generation process in this specific study setup (though explicit DP methods from Chapter 4 could be integrated). We then trained a discriminative machine learning model for the clinical pathway task solely on this generated synthetic data.

The results of this experiment were notable. The model trained on the synthetic data achieved performance levels that were comparable to, and on some metrics slightly better than, the models trained directly on the distributed real data via federated learning. A possible explanation for this outcome is that the generative modeling process itself may act as a form of regularization or data balancing. By learning the overall data distribution and generating samples from it, the synthesizer might implicitly smooth over irregularities like label imbalances or certain biases present in the original fragmented datasets, leading to a synthetic dataset that facilitates better generalization for the downstream task.

Discussion This proof-of-concept study provides initial evidence supporting the use of synthetic data for analysis in distributed DHT environments. The methods for tail-adaptive generation, semantic integrity (Chapter 3), and differential privacy (Chapter 4) were developed and validated individually; however, integrating them fully into a complex, real-world DHT scenario faced practical limitations related to data access approvals, ethical considerations, and the inherent messiness of clinical data. Consequently, the scope of this initial validation was necessarily constrained.

Despite these limitations, the findings are encouraging. The observation that models trained on synthetic data can perform on par with, or even slightly outperform, those trained via federated learning on real data lends practical support to the arguments made throughout this thesis. It suggests that synthetic data may offer a pathway not only to overcome privacy and access barriers but potentially also to improve the robustness and fairness of analyses conducted in distributed settings, particularly when the underlying real data suffers from issues like non-IID distributions or imbalances. While these preliminary results are promising, they must be interpreted with caution. More extensive validation across diverse datasets, clinical applications, and using a broader suite of performance and fairness metrics is required to solidify these conclusions. Nonetheless, this study provides a foundation and motivation for further exploration of synthetic

data as a key enabling technology for distributed health data analysis.

Future Directions To build upon this initial proof-of-concept and fully assess the utility of synthetic data in distributed DHTs, several research directions should be pursued. First, empirical evaluations need to be broadened significantly. This includes testing on more complex and diverse clinical datasets encompassing various diseases, patient populations, and data types (e.g., longitudinal records, imaging, genomics).

Second, rigorous comparative studies are essential. These studies should directly compare the downstream task performance of models trained using different data sources: (1) private synthetic data generated using formal DP mechanisms, (2) non-private synthetic data (to isolate the utility cost of privacy), and (3) models trained via standard federated learning on the original distributed data. Such benchmarks are crucial for understanding the practical privacy-utility trade-offs and identifying scenarios where synthetic data offers the most significant advantages.

Third, the architecture of the federated synthetic data generation process itself warrants further investigation. Beyond the approach used here (federated training to produce synthetic data for a central task), alternative strategies should be explored and compared. These could include:

- *Local Synthesis with Central Aggregation*: Each node trains its own generative model locally on its private data, generates synthetic samples, and these samples (rather than models or gradients) are aggregated centrally.
- *Federated Training of the Generator*: Applying federated learning algorithms directly to train a single, shared generative model across all nodes without sharing raw data.
- *Hierarchical Federated Synthesis*: More complex, multi-level approaches. For instance, local nodes could train simpler private generators whose outputs or parameters inform the training of a more powerful global generator. Nodes might dynamically switch roles between generating and validating synthetic data. Alternatively, a global synthesizer could be trained collaboratively, then fine-tuned locally using private data with DP constraints for personalization.

Each architectural variant presents different trade-offs regarding communication costs, computational load, privacy guarantees, and the statistical quality of the final synthetic dataset. The optimal choice is likely application-dependent. Systematic evaluation of these alternatives is needed to guide the design of effective and efficient synthetic data pipelines for diverse DHT applications.

6.3 The Big Picture

The prospect of Digital Health Twins (DHTs), integrating insights from distributed data sources through collaborative machine learning while ensuring privacy and security, holds significant promise for advancing healthcare. However, translating this vision into practice encounters substantial hurdles inherent in the nature of real-world health data and distributed systems. This thesis has contended that directly applying advanced distributed learning and privacy techniques to raw, dispersed health data often falls short due to the compounding effects of underlying data challenges.

Real-world health datasets, particularly when aggregated across diverse institutions or settings, frequently exhibit characteristics that complicate analysis. These include inherent heterogeneity in data distributions arising from varied collection practices and patient populations; the pervasive issues of missing values and data sparsity that can undermine model training and inference; potential biases stemming from sampling methods, measurement inaccuracies, or historical inequities reflected in the data; and the non-identically distributed (non-IID) nature of data both within individual sources and across the distributed network. Attempting complex analyses or model training directly upon such data, especially under the additional constraints imposed by privacy-preserving mechanisms like federated learning or differential privacy applied post-hoc, can lead to results that are suboptimal in performance, potentially unreliable for clinical decision support, and may even inadvertently amplify existing fairness disparities.

As a principled alternative to circumvent these direct analysis challenges, this thesis has investigated and advocated for the strategic use of generative models to create high-fidelity synthetic data. The central proposition is that by first learning the com-

plex, high-dimensional probability distributions underlying the real health data, we can subsequently generate artificial datasets. These synthetic datasets can serve as effective, privacy-preserving surrogates for the original data in downstream analytical tasks, effectively decoupling the analysis from the sensitive, raw information.

This synthetic data paradigm offers several fundamental advantages within the DHT context. Crucially, it provides a mechanism to embed privacy guarantees, such as differential privacy, directly into the data generation process itself. By ensuring the generative model adheres to privacy constraints, the resulting synthetic data inherits these protections, facilitating data sharing and collaborative analysis in compliance with stringent regulatory requirements and ethical considerations, thereby mitigating risks associated with direct data access. Furthermore, synthetic data generation addresses practical limitations like data scarcity or imbalance. It allows for the creation of large, tailored datasets on demand, potentially enriched or balanced in ways that support the training of more robust and equitable machine learning models than might be achievable with the original, limited data.

Probabilistic generative models, including the Variational Autoencoders, Generative Adversarial Networks, and particularly the normalizing flows explored in depth within this work, provide the mathematical and computational frameworks for this undertaking. These models are designed to capture the intricate statistical dependencies and latent structures within complex data. By training them on real health datasets, we can create models that encapsulate the essential characteristics of the data distribution without retaining specific, identifiable patient details. The trained models then act as simulators, capable of producing new data points that are statistically indistinguishable from real data according to relevant metrics, encompassing multivariate correlations, marginal distributions, and potentially complex temporal patterns or subgroup characteristics. This capability enables diverse downstream applications, from developing predictive algorithms for personalized risk assessment to conducting large-scale exploratory analyses for public health insights, all operating on the synthetic representation.

When deployed within a distributed DHT architecture, this synthetic data approach

can actively mitigate some of the core pathologies of distributed real data. By learning a unified representation of the data distribution (potentially through federated training of the generator itself, or by harmonizing outputs from local generators), the resulting synthetic data can exhibit greater statistical homogeneity compared to the fragmented, non-IID real data sources. This harmonization can lead to improved performance and stability for models trained subsequently in the distributed setting. Generative models also offer inherent capabilities for handling missing data; they can learn to impute missing values consistent with the learned data manifold, yielding more complete datasets for analysis.

However, it is essential to acknowledge that the efficacy of this synthetic data approach is not automatic. Its success hinges critically on the ability of the chosen generative models to accurately capture the nuances of the target data distribution, especially in high dimensions and concerning features crucial for the downstream task (such as the tail behaviors and semantic constraints investigated herein). This necessitates the development and application of sophisticated modeling techniques, like those presented in this thesis, accompanied by rigorous validation protocols to assess the fidelity and utility of the generated data. While synthetic data enhances privacy, potential residual risks must still be carefully managed through appropriate privacy budget allocation (in DP contexts) and thorough privacy audits. Moreover, the computational resources required for training advanced generative models, particularly on large-scale distributed datasets, remain a practical consideration demanding ongoing research into algorithmic efficiency and scalable implementations.

The primary contribution of this thesis, therefore, lies not simply in promoting synthetic data for DHTs, but in addressing specific, critical technical obstacles to generating synthetic health data that is simultaneously high-fidelity, privacy-preserving, and semantically meaningful. The development of tail-adaptive normalizing flows, methods for enforcing semantic integrity during synthesis, and a performant approach to differentially private flow-based generation represent concrete steps towards overcoming these obstacles. By tackling the challenges of accurately modeling extreme events, preserving domain-specific constraints, and achieving strong privacy guarantees without excessive utility loss, this work aims to establish a more robust foundation for the use of synthetic

data in healthcare.

This thesis posits that synthetic data generation, when underpinned by the advanced and rigorously validated probabilistic modeling techniques developed herein, offers a compelling and practical pathway toward realizing the vision of performant, private, and equitable data analysis within distributed Digital Health Twin ecosystems. By providing high-utility, privacy-enhanced data representations, this approach helps unlock the collective power of distributed health data, ultimately contributing to advancements in personalized medicine, clinical understanding, and population health outcomes.

APPENDIX A

Private and Secure Distributed Deep Learning: A Survey

A.1 Introduction

The ubiquitous popularity of Deep Learning (DL) in diverse applications in different domains, and the fact that these methods generally benefit from large amounts of data as training input, has led to an increased need to access big data reservoirs. This need continues beyond the availability of central databases, as the quantity of data necessary to train deep models in certain use cases can be higher than many centralized datasets alone can provide. Moreover, the age of big data leads to data generation being a normal function of entities in different scales, from individuals to large organizations. Creating extensive centralized databases can be costly, problematic in terms of privacy, or simply infeasible due to processing and communication restrictions. This has led to an interest in Distributed Deep Learning (DDL), where multiple parties cooperate to utilize distributed data for a combined DL effort.

Distributed learning covers a wide set of paradigms, such as Machine Learning as a Service (MLaaS) - which makes models available for inference on new distributed samples - and federated learning (FL) - which allows for the collaborative training of a single neural network on distributed local datasets. In distributed deep learning, the

problem of decentralized data access is resolved by moving computing to where the data resides. However, additional problems arise in the DDL context. One of the most prominent challenges is retaining the privacy and security of data and models in distributed settings. With the cooperation of multiple different parties, there are often incentives to keep local data and/or models private, be it for legal regulations such as GDPR [137], an unwillingness to share to keep a competitive advantage towards other parties, or ethical concerns. This has amplified interest in the application of privacy- and security-preserving mechanisms during model development, testing, implementation, and inference.

We define the privacy-preserving paradigm of distributed deep learning in two different contexts: **Privacy** and **Security**. Security is defined as the safeguarding of data and models. Achieving secure DDL means a malicious party having access to the DDL process would not be able to gain unauthorized access to training data, model parameters, communications, etc. Privacy, on the other hand, is defined as stopping the models, either partially trained local models or fully trained centralized ones, from leaking information about the training data and the model itself. In this paper, we refer to security as the classification of the methods that protect against unauthorized access to data, mainly through cryptographic means. Common types of cryptographic methods are **Secure Multiparty Computation (MPC)**, which provides secure messaging protocols between different parties, and **Homomorphic Encryption (HE)**, a type of encryption that allows accurate calculations to be performed on encrypted data. The privacy front in this work aims for methods that provide formal guarantees of privacy with quantifiable privacy levels and therefore focuses on methods that provide **Differential Privacy (DP)** guarantees.

Secure and private methods of DDL have been well-researched. Several surveys have previously attempted to structure and summarize studies performed in this research area [138][139][140][141]. Most of these reviews only focus on one class of protective measures, such as differential privacy[142] or multiparty computation [139], or on a subset of the available types of distributed learning, such as MLaaS [138], or Federated Learning [140][141]. In this chapter, we aim to provide a larger and more comprehensive overview of the field of private and secure distributed deep learning (PSDDL), covering

both the security and privacy aspects of these methods. In this way, this survey serves as a starting point for comparisons between the different privacy and security methods and evaluations of applicable protective measures in different use cases. Depending on the use case requirements, including the distribution of data and models, the level of security of privacy required, and the possibility of a central server, the requirements of the security- and privacy-preserving methods change. Therefore this survey aims to provide a high-level overview of the field of private and secure distributed deep learning, and to discuss the possibilities of privacy and security mechanisms per type of distributed deep learning workflow.

The structure of this chapter is thus as follows: section A.2 shows our survey protocol, where we define the scope of the survey. In section A.3 we describe our taxonomy along which we organized the selected papers. We divided the type of distribution workflow is two main categories. sections A.4 and A.5 discuss the papers applicable to the two main DDL categories, respectively collaborative learning (CBL), and distributed learning. Both the categories will focus on centralized workflows, but the discussion (section A.6) will also touch upon decentralized frameworks.

A.2 survey protocol

A.2.1 Scope of Survey

The scope of this survey is *private and secure distributed deep learning*. We explain the choice of this scope by explaining its following parts: Private and Secure, Distributed, and Deep Learning.

Private and Secure: In this survey, we make a distinction between security and privacy of distributed DL models. The security aspect of these methods is concerned with stopping adversarial behavior, both by members of the distributed pipeline as well as outside entities, from having unauthorized access to the training or model data and being able to potentially tamper with them. For privacy, we look at how to reduce and/or mitigate the risk of information leakage on the inference surface of the model

and protect it against adversarial attacks. Privacy is mainly achieved by differential privacy methods, while cryptographic methods are the main mechanisms for security. While their methods and definitions are different, their goals are complementary, and both need to be considered for properly safe architecture.

Distributed: We aim for this survey to serve as a starting point for recommendations of security and privacy depending on the use case requirements, including the distribution of data and models, the level of security of privacy required, and the possibility of a central server, the requirements of the security- and privacy-preserving methods change. Therefore, it is essential to discuss different types of data distribution. Our comprehensive view explains the strengths and vulnerabilities of different types of distributed learning, and how different protective measures help to reach the necessary privacy and security depending on the use case.

Deep Learning: We chose to limit the survey to Deep Learning instead of all Machine Learning (ML). DL offers state-of-the-art performance for most tasks, especially on tasks with high-volume data, which is often the subject of distributed learning. Moreover, the security and privacy challenges can be different from other ML models, as the complexity of the models increases the chances of private information being extracted from DL models. For brevity and clarity, we therefore only focus on DL.

The combination of these three aspects of this survey offers us the opportunity to provide a high-level and more comprehensive overview of the field of privacy and secure distributed deep learning, by being able to compare not only within one sub-specialt, but among different types of security and privacy and distributed learning methodologies. To underline this, we compare the scope of our survey to other recent surveys in Table A.1. We highlight that we did not find any surveys that focus on both collaborative learning and federated learning, as well as other distributed learning methods, such as split learning [143]. However, these methods are often used for similar purposes and therefore warrant comparison for a full overview of the field of PSDDL.

Reference	ML or DL	Collaborative Learning		Federated Learning		Other DDL methods	Attacks
		Cryptographic methods	Differential Privacy	Cryptographic methods	Differential Privacy		
[138]	DL	✓	✓				✓
[139]	DL	✓					
[140]	ML			✓	✓		
[141]	DL			✓	✓		✓
Our	DL	✓	✓	✓	✓	✓	

Table A.1: Comparison of surveys within the field of PSDDL

A.3 PSDDL Taxonomy

To effectively compare the selected papers, in this work we present a taxonomy based on the use case of the methods described in the papers. There are different configurations of Distributed Deep Learning in which security and privacy preserving mechanisms can be applied, which leads us to base our taxonomy on the workflow of secure and private DDL. As such, this paper can serve as a guideline, showing which types of preservation mechanisms can be applied in different domains and for different use cases, depending on the intended DDL architecture.

A.3.1 The Distributed Deep Learning Framework

While many different types of DDL workflows have been developed, the principal areas of research in this context can be classified into two main categories: *Collaborative Learning* and *Federated Learning*.

Collaborative Learning (CBL) In this setup, the central server trains one central model, while the data is distributed over different clients. The local data and its derivatives are securely communicated to the server. A distinction here can be made between collaborative learning, where the distributed data is used for model training, and *Secure Inference (SI)*, where inference by a previously trained model is performed on a local sample. The latter usually is the basis of Deep Learning as a Service (DLaaS).

Distributed Learning Here, clients do not share their data, but cooperatively train a model by sharing their private models' details, such as gradients and weights. The most common architecture is federated learning (FL). In FL's basic form, clients train a local copy of the central model on their private datasets and share raw model parameters or their derivatives with an orchestrator. The orchestrator uses an aggregation mechanism

to combine the received parameters and communicates that back to the participants. This process is iterated until convergence. We will also discuss other distributed learning methods, such as split learning [143] and ensemble methods. It is important to mention we use a slightly narrower definition of distributed learning, to refer to methods that involve model sharing.

These categories are illustrated in Figure A-1, with FL as an example of distributed learning. Within each main category, there are different points in the architecture where privacy or security mechanisms can be applied. section A.4 will discuss the developed methods for CBL, section A.5 will cover DDL. Some workflows cannot be classified into these two categories, most notably decentralized distributed learning. We choose not to include these in the body of our survey, as the protective measures on these types of workflows are still in the emerging stage. In section A.6 we will highlight how they compare to the protective measures in our two main categories.

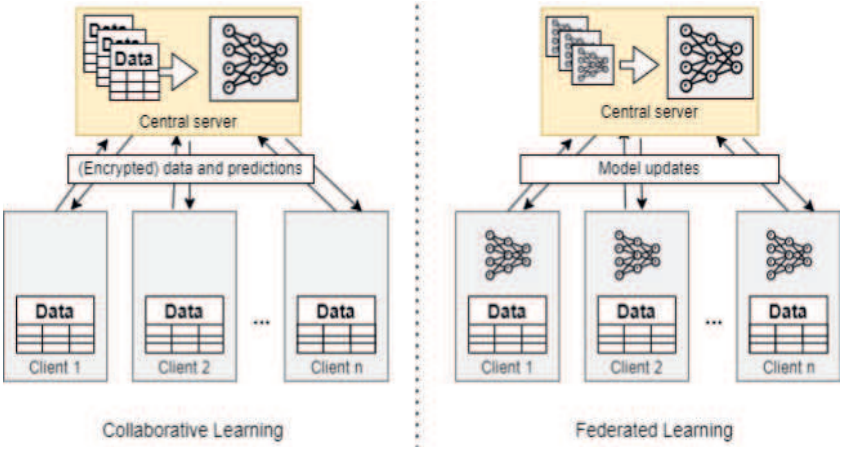


Figure A-1: Two major architectures for distributed learning. In FL participants are actively involved in training the model while in collaborative learning participants contribute by sharing their data or results for DL training and inference on their data in a private and secure manner by an orchestrator

Data Partitioning In Distributed Learning There are different ways a dataset can be divided over different clients. The most commonly occurring distribution is horizontal partitioning, where every client has a partial dataset that contains a partition of the samples. In vertical partitioning, they contain a part of the features. Most research focuses on horizontally partitioned data. Therefore, we will specifically mention when

a framework offers privacy protection for vertically partitioned distributed learning.

A.3.2 Privacy Vulnerabilities of Deep Learning Models Against Privacy Attacks

Distributed and federated learning aim to decentralize model training, mitigating risks associated with centralized data storage. Both paradigms leverage multiple devices or servers holding local data samples, thereby enhancing privacy by minimizing raw data sharing. Distributed learning performs parallel computations on data distributed across various locations. Federated learning, a specialized form of distributed learning, employs orchestrators to facilitate training while data resides on user devices. Despite these decentralized approaches, both distributed and federated learning remain susceptible to privacy vulnerabilities during training and inference [144].

While essential for model convergence, the exchange of model updates in distributed and federated learning creates privacy vulnerabilities. Analyzing update patterns over multiple iterations can reveal sensitive information about local datasets [119]. Even during inference, model responses can leak information, potentially allowing adversaries to reconstruct a surrogate model and infer characteristics of the training data [145].

Privacy Attacks for Exploiting Machine Learning Models

Privacy attacks against DDL are methods by which malicious entities attempt to exploit vulnerabilities in these systems to gain unauthorized access to sensitive data. Privacy attacks generally exploit the data processing capabilities of distributed ML models and architectures to infer protected or sensitive information. These attacks can occur at various stages of a DDL model's lifecycle, including during collaborative training, inference, or data transmission phases. Below, we outline some general mechanisms often employed in privacy attacks on ML systems [146]. Based on their goals and their exploitation method, these adversarial attacks are categorized as *Membership Inference*, *Reconstruction*, *Inversion*, and *Property Inference* attacks.

Membership Inference Attacks. Membership inference attacks [147] are sophisticated

privacy breaches aimed at deducing the presence or absence of specific data instances in the training set of an ML model. These attacks exploit differential behaviors exhibited by the model processing input data that it has encountered during training versus unseen data. The core hypothesis driving these attacks is that ML models tend to predict with higher confidence and lower entropy when presented with familiar data. Attackers leverage this by analyzing the confidence levels, decision boundaries, and output probabilities of the model for various inputs, and can infer with some probability if a given data point was part of the training dataset. This type of attack poses significant risks, particularly in scenarios where the inclusion of an individual's data in a model's training set could reveal sensitive personal information.,

Reconstruction Attacks. Reconstruction attacks [148] focus on the capability to approximate or fully reconstruct the input data used to train a machine-learning model, based solely on observing the model's outputs. These attacks meticulously probe the ML model with a series of strategically crafted inputs and analyze the corresponding outputs to deduce the characteristics or even exact entries of the training data. Attackers may employ techniques such as gradient-based optimization to modify their input iteratively, moving closer to the original training data with each step by minimizing the difference between their input's predicted outputs and the outputs observed when the actual training data is used. This attack vector is particularly alarming for models trained on confidential or proprietary data, as it can lead to significant privacy breaches or intellectual property theft.

Inversion Attacks. Inversion attacks [149] are a nuanced subset of reconstruction attacks where the objective is to invert the model's predictive function to reveal details about the training data or reconstruct the input leading to specific outputs. These attacks utilize the predictable nature of the model and the information encoded within its parameters to reverse-engineer the inputs. By understanding how model outputs change in response to variations in input and by exploiting the model's gradients, attackers can craft inputs that, when processed by the model, reveal substantial information about the data on which the model was trained. The challenge and effectiveness of inversion attacks depend significantly on the model's complexity and the amount of information implicitly stored in its parameters.

Property Inference Attacks. Property inference attacks [150] are designed to uncover global statistical properties of the dataset upon which a machine-learning model was trained, rather than focusing on individual data entries. These attacks exploit the model's learned characteristics to infer broader patterns, trends, or biases in the training data. For instance, an attacker might analyze the model's outputs across a range of inputs to determine if the training data over-represents a particular demographic or if certain predictive features are correlated with sensitive attributes. Property inference attacks typically involve complex statistical methods and hypothesis testing against the model's predictions to derive insights that might reveal sensitive or discriminatory practices in data handling and model training.

A.3.3 Protective Mechanisms Against Adversarial Attacks

In this work we make a distinction between security mechanisms and privacy-preserving methods. Within these two categories, many different methods are applied by the reviewed papers. To make this work approachable to a larger audience, we will give a short overview and explanation of these mechanisms in this section. Readers interested in detailed explanations and formal definitions of these methods are encouraged to consult the appendices.

Security Mechanisms

The type of security mechanisms reviewed in this work is based on the type of encryption protocols. In PPDDL, two main types of advanced cryptography are used: **Secure Multiparty Computation (SMPC)** and **Homomorphic Encryption (HE)**. Secure Multiparty Computation (SPMC) Multiple parties collaborate to perform a calculation while keeping their input in the calculation secret. We make a distinction between multiparty computation (MPC), with 3 or more clients, and two-party computation (2PC) with two clients. While various protocols have been developed to achieve SPMC, the most common method used is *garbled circuits* - which is based on oblivious transfer - for 2PC and *secret sharing* for MPC. These methods are further explained and discussed in the appendix. Homomorphic encryption allows for calculations to be

performed on encrypted data. When the results of these calculations are decrypted, it is as if the calculations were performed on the unencrypted data. In *full homomorphic encryption (FHE)* all types of calculations are possible. Using HE introduces noise, and FHE is only possible by often re-encrypting the data to limit the noise. Doing these calculations can become very computationally expensive. There are several ways this cost can be reduced. One possibility is to limit the type of allowed computations, such as with additional homomorphic encryption (AHE), which only allows for addition-based computations. Another possibility is to retain the option for all calculations but maintain a limited depth of the circuit, as with *Leveled FHE*. While LFHE allows for both multiplication and addition (essentially the basis of all calculations) there is a limited depth, which might not allow for complicated functions such as the activation function in a neural network. It is important to note that HE offers honest-but-curious privacy protection.

Security Levels

When promoting the privacy preservation of a protocol or workflow, the level of adversary it has been proven to withstand needs to be addressed. This includes both their access level and their behavior. These classifications, security levels, determine in what type of security attacks a network would remain secure. The two main classifications that are covered are semi-honest or *honest-but-curious (HBC)* and *Malicious Adversary (MA)*. In both situations, adversaries are trying to collect secure information. In the HBC scenario, the adversaries will try to get as much information as possible, but will not break the rules of the protocol. In the MA scenario, the adversary will also break protocol wherever possible. Another classification of security levels is the collaboration or collusion of adversaries, as a setting where multiple adversaries are cooperating to gain private information needs a more secure protocol. In the MA scenario, there is also robustness and fairness to consider: when a malicious party is discovered, the standard MA protocol will simply be interrupted. In a robust or fair scenario, either all or no parties receive the outcome respectively.

Privacy Mechanisms

We define the privacy of ML models as preventing them from leaking information about their training data and their internal information and state. In a centralized setting, this only concerns the inference API of the model. In a distributed and/or federated setup, we also need to consider the privacy of the output of the participants and the orchestrator as well as the privacy of the final model(s) once their inference API is released. Different techniques have been proposed to preserve the privacy of randomized mechanisms such as ML models, including different levels of anonymization. However, it has been shown that these methods are susceptible to adversarial attacks such as linkage attacks to certain degrees [151].

Differential Privacy (DP) is a standard for preserving the privacy of randomized mechanisms that provides a rigorous mathematical proof of privacy. In its most extreme case, it provides plausible deniability to any individual member of the training set if our ML model is trained on two datasets that differ in only that member and the adversary knows all about the difference in the datasets and can observe the distribution of the output of our model. DP will limit the difference in the log-likelihood of the same model trained on these two *adjacent* datasets by a privacy budget ϵ. There are different definitions for DP, but ϵ-DP, originally introduced by Dwork et al. [34] only considers the privacy budget. The lower the privacy budget is, the higher the level of privacy of the model would be. This comes at a cost of performance and potential fairness for the model and a tradeoff needs to be considered between the privacy and utility of the model. Differential privacy is typically achieved by introducing some type of perturbation at training and/or inference time, usually through adding carefully calibrated noise. The aim of the noise-adding mechanism is to add enough perturbation to the system so that the model learns as little as possible about individual members of the training set while learning as much as possible about the population. More details on the definition of DP and its noise-adding mechanisms can be found in the appendix.

Privacy Levels

In a typical centralized distributed learning scenario such as federated learning depending on the threat model, different DP levels can be achieved. Generally, we can add DP on *local* and *global* levels. Local DP adds noise-adding mechanisms - and thus ensures DP guarantees - on participant levels while global DP uses DP mechanisms to ensure privacy on the parameterset level on the orchestrator. Global DP is desirable when the orchestrator is trusted but there are no trust assumptions about the participants. Local DP is used when the orchestrator is not trusted but no expectation of adversarial behavior is included in the threat model. Combining local and global DP ensures privacy in a most pessimistic threat model with both the orchestrator and the participants being expected to behave in an adversarial manner. Although, reiterating our earlier point, adding additional levels of privacy usually comes at the price of loss of utility - although this assumption is not unequivocally true as DP noise-adding mechanisms have been shown to act as regularization mechanisms which would benefit certain aspects of the model utility; details of this process are out of the scope of this review.

A.4 Collaborative Learning

In collaborative learning (CBL), the distributed data of the clients is sent to a central server, where the data is used for machine learning. Thus, the local clients do not build a local model themselves. The workflow of CBL is shown in Figure A-2(left). As shown, there are three points where protective measures can be taken:

1. during the exchange of the training set data
2. during the training of the model on the server
3. during the exchange of the sample and subsequent label for inference

If there is a previously trained model, the clients often may only look to perform the last step of the workflow: inference. The trained model is stored in a server and clients have unlabeled samples that they want to classify. An example would be Deep learning as a service (DLaaS), where the service provider aims to keep their model

private, and the client aims to keep their data private, yet both parties want the model to perform inference on the data. These situations are referred to as secure inference (SI), as will be discussed in section 4.2. In section 4.3, the papers are included that focus on secure training, the training phase of CBL. In those situations, next to the security-preserving mechanisms applied to the exchanging of partial datasets or a single samples for inference, the server might also apply differential privacy when training the model on the server which is referred to as step 2 in the workflow. We exclude privacy-preserving methods through differential privacy in this section. As the parties do not perform their own training or analysis, there is no special setting for DP. We therefore refer to other surveys on local DP, such as [152] Figure A-2(right) gives an overview and categorization of the discussed papers in this chapter.

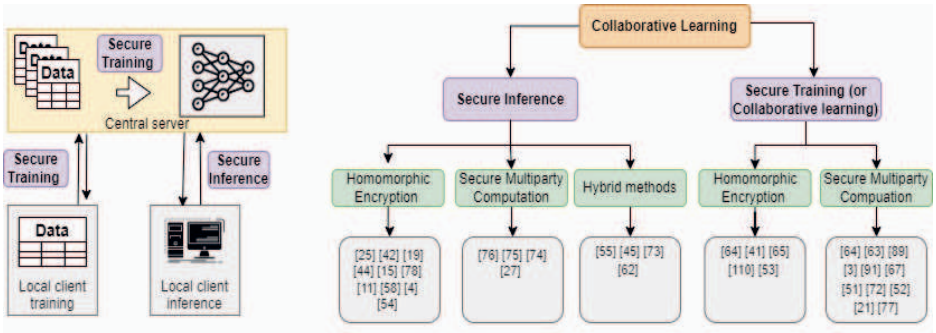


Figure A-2: Left: CBL workflow. It shows at which point secure inference, secure training, and private server training are applied. Right: Distribution of papers among the different types of protective mechanisms for the different points in the workflow.

A.4.1 Secure Inference

This section will discuss the selected frameworks that focus on facilitating secure inference. In these cases, there is an already trained network that the server provides for inference of the client's unseen samples and the data of the client should remain private. The server would also like to keep its neural network secure, because of issues such as data leakage, or, in a situation such as DLaaS where the model in the product they are providing, to keep a competitive advantage. All of the discussed frameworks that facilitate secure inference used encryption-based security methods, although a few combined them with differential privacy. The different frameworks used homomorphic encryption

(HE), secure multiparty computation (MPC), or a combination of the two. In the following sections, the papers will be discussed organized by encryption method. section 4.2.1 focuses on HE, section 4.2.2. on MPC, and section 4.2.3. on hybrid methods. The SI frameworks are summarized in Table A.2.

Secure Inference With Homomorphic Encryption

Several papers had an idea of using encrypted data for machine learning purposes[174][175], but the first to develop a system that allowed for deep learning inference of homomorphically encrypted data was [154] with Cryptonets. They applied leveled HE using the YASHE scheme to neural networks. This YASHE scheme has since been shown to be vulnerable [176] [177] and later papers will use different schemes. Nevertheless, they laid the foundation for secure inference.

One of the main issues with inference over homomorphically encrypted data is the activation function of neural networks. It decides whether a certain neuron should be activated by calculating the weighted sum of the inputs to the neuron and adding bias to it. The most common activation functions are ReLU, Sigmoid, and Tanh. The whole purpose of the activation functions is to introduce non-linearity into the outputs of the neurons, for them to be able to perform more complex tasks. Therefore the activation functions are non-linear and complex functions. Unfortunately, those are the types of functions that the commonly applied leveled FHE schemes have trouble with, as they support addition and multiplication, but only up until a set calculation depth. The standard activation function would be too 'deep' in that regard. As the approximation of the activation function is a main bottleneck in the implementation of HE-based secure inference, it is a main topic in the field. One way to circumvent this complexity problem is to provide a different activation function that is an approximation of the original one, but less complex, or more suitable for homomorphic encryption. As this is an approximation, it can lead to a loss of predictive performance. This is what Cryptonets introduced and several other papers in this section thereafter aimed to improve. Another option is to use a cryptographic protocol to replace to activation function, these papers will be discussed in the hybrid section.

With CryptoNets, they approximated the sigmoid activation by replaceing the max pooling step i with a scaled mean-pool square function. While HE-suitable, it resulted in the computing costs increasing with the number of NN layers, and for bigger networks eventually becoming prohibitive. CryptoDL by [156] builds on Cryptonets by adding approximations of several new activation functions. It implements low-degree polynomial approximations of several activation functions (i.e. ReLU, Sigmoid, and Tanh). The use of their approximates of activation functions leads to a closer approximation of a normal NN, which is visible in their performance.

Several other papers improved on other aspects of the CryptoNets framework. Faster-Cryptonets [160] aims to provide a speed up in efficiency on CryptoNets. It not only increases efficiency, thus speeding up the inference, but it also provides an option for differential privacy for a small cost in predictive performance. [178] focuses on the addition of a batch normalization to the scheme. Batch normalization is a technique that converts interlayer outputs into a standard format, which allows the layers to learn independently of the others. The purpose of the paper is to create a normalization that is reparametrized to support the conditions of the leveled FHE scheme.

[155] and [163] take a different approach to improve the efficiency of the secure inference framework. As mentioned in section 3, there are different HE schemes, with different properties. One main difference is the type of data input they can handle. While the previous frameworks used schemes that supported long values, [155] and [163] used TFHE, a scheme that only allows for binary operations. This makes the values used in the networks less precise, as they need to be binarized, but this HE scheme is more efficient than the other schemes. [155] developed FHE-DiNN, a discretized network whose complexity is strictly linear to the depth of the network. [163] also implemented TFHE in TAPAS, thereby allowing only binary weights and activations. The framework is implemented as a Binary Neural Network (BNN), by allowing for XOR, AND, and XNOR operations through a function called reduce tree. While efficient and secure, the binary NN does lead to an, albeit small, decrease in predictive performance, compared to other HE frameworks. More recent papers mainly focus on improving the speedup of the networks, to allow for larger networks to be run as secure inference.

The [154][156] frameworks have shown that the predictive performance of SI is comparable to unencrypted inference. The following papers show speedup through a compiler [165], GPUs [170] and different approximation functions [173]. [165] introduces their framework based on their previously developed NGraphHE compiler [179]. They leverage the CKKS FHE scheme, which offers support for real numbers. They introduce a client-aided model, where the two parties can cooperatively compute activation functions. Thereby, they implemented the largest network at the time, pre-trained MobileNetV2 and ran inference on the ImageNet dataset. This is extended in [171], where hybrid memory is used to implement even larger models, such as ResNet-50. Al Badawi et al. [170] improves efficiency by presenting the first GPU-accelerated FHE, creating a speed up of the classification process. This allows for the classification of large batches of images. [173] propose a new method of approximation of the activation function. It replaces the ReLU and max-pooling functions of other activation functions, with a composition of min-max approximate polynomials of small degrees. The small degrees allow for more efficiency while it allows for common and well-developed deep learning models, as shown for ResNet and VGGNet, to be implemented without further modification and reusing of the pre-trained parameters.

Secure Inference with MPC

While the first frameworks of secure deep learning inference were developed with HE, several papers focused on implementing a system with MPC methods. MPC approaches are generally faster, but they do usually rely on all parties being involved in the whole computation. This can be problematic, but not necessarily in a two-party inference set-up, like is common in secure inference. Often a combination of different MPCs was used for different steps in the neural network, for the different protocols have different characteristics suitable for different types of calculations. DeepSecure by [161], chose to diverge from homomorphic encryption, by using Garbled Circuits (GC) to create a provably secure inference framework. It achieves this by transforming the input data into a 'lower-dimensional subspace ensemble'. Moreover, it converts and approximates the neural network into XOR, AND, XNOR gates that allow for the garbled circuit protocol. It is provably secure and offers a speed up compared to Cryptonets.

[159] uses a hybrid system of MPC to create a more efficient system for secure inference. It combines both 2PC and MPC techniques. In particular, it performs linear operations using additively secret shared values and nonlinear operations using Garbled Circuits or the Goldreich-Micali-Wigderson protocol[180]. Moreover, Chameleon's framework moves away from the common assumption of secret sharing models that three or more parties need to communicate in the online phase, and reduces communication overhead by precomputing heavy cryptographic operations in the offline phase before the parties communicate. It thereby provides a speed up of 4.5x compared to DeepSecure.

The XONN framework [166] creates secure inference through a novel protocol for conditional addition based on Oblivious Transfer (OT). For OT, XNOR operations are of low computational cost[181]. Therefore, in order to create a more efficient setup, XONN replaces costly multiplication steps in the neural network with XNOR operations. Next to this, it also implements a trimming technique for neurons that contribute little to the inference. This adds up to an increase in efficiency compared to previously developed secure inference methods, (2.7x speed up on Cifar-10 over Gazelle[162] mentioned in 4.3). XONN is particularly efficient for larger CNNs, compared to previous studies.

Another advantage of using MPC methods, compared to homomorphic encryption, is that they can more easily also incorporate data verifiability, instead of only data confidentiality. An example of this is PVDLI by [172], which applies secret sharing. Verifiability of the inference result of the cloud was previously the focus of [182]. They did not use SPMC methods but instead made the server provide a simple short mathematical proof of the correctness of inference tasks using a sum-check protocol.

[172] takes a different approach: they inject labeled verifiable data into the model input, which offers verification of the correctness of the inference results with a high probability. Moreover, the secret sharing allows the injected verification data to be indistinguishable from the normal ones, so the malicious party cannot determine which are the verification data.

Secure Inference with Hybrid Methods

Several papers aimed to combine the best features of both SPMC and HE. As previously mentioned, one way to achieve this is to implement a HE method but replace the activation function with an MPC protocol instead of a mathematical approximation. This way, the most costly calculation of HE can be replaced with a more efficient, and possibly more accurate protocol. [158] was the first to attempt a hybrid setup, by introducing oblivious neural networks (ONNs) with the *MiniONN* framework. ONNs are a combination of leveled FHE (using the now outdated YASHE framework) and oblivious transfer (OT). It uses the oblivious transfer to replace the non-linear functions (such as ReLu) with oblivious transfer protocols. It makes the basis of the non-linear functions into dot triplet products. These are similar to the multiplication triplets in oblivious transfer, and therefore OT can be applied. This manages to highly outperform CryptoNets in terms of efficiency, especially for small numbers of requests that require quick responses.

Gazelle [162] continues the evolvement of facilitating secure inference using leveled FHE encryption but different from miniONN, it handles the non-linear functions through the application of garbled circuits (GC). The goal of the paper is to use techniques that involve division-free arithmetic, as when they are converted to homomorphic operations, they are significantly slower compared to the operations on plaintext. Moreover, it uses the BFV scheme, instead of the YASHE in *MiniONN*. This method was further optimized by *Cheetah* [168], which focused exclusively on optimizing the process as proposed by Gazelle. It did not necessarily achieve speed up through a different cryptographic protocol, but instead focused on speed up by optimizing the server side of secure inference through tuning and scheduling. This leads to a speed up of up to 79x compared to *Gazelle*.

Delphi[167] uses a (lightweight) cryptographic protocol, in combination with a planner, to speed up the architecture as developed by Gazelle. It uses a truncation trick to avoid overflow when multiplying secret-shared fixed-point numbers. Truncation is limiting the number of digits behind the decimal point. However, ABY3, discussed in section 4.3, demonstrated that this trick can introduce large errors with non-negligible probability.

This could result in erroneous classifications, so to verify that DELPHI avoids such errors, it was compared to miniONN, with a 99.2% overlap in prediction outcome, showing largely avoidance of these errors.

A.4.2 Secure Training

The previous section focused on secure inference, a common goal for distributed networks. However, there is also interest in training the network over distributed data, instead of using an already trained model. This is referred to as secure training (ST). Usually, due to the nature of neural networks, if training is possible, so is inference. Training implies forward and backward Many methods are adapted from the secure inference, and therefore also use HE, SMPC, or a combination of both. We will discuss these papers along the same axis, dividing between homomorphic methods and secure multiparty computation methods. The papers discussed are summarized in Table A.3.

SecureML [183] provides the very first method that focuses on training as well as inference. They use the ABY framework [199]. *SecureML* proposes a new approximate fixed point multiplication protocol that avoids binary circuits and truncates decimal numbers and uses this to train neural network models. This technique is limited to 2 parties. *SecureML* provides both options for MPC and HE using respectively OT and LHE. However, the accuracy with LHE is very low as higher degree polynomials could not be used, as they introduced too much overhead. For the OT solution, even by focusing on simple MLPs, the training time was in the thousands of seconds per iteration, and therefore too long and not practical. The works in the following sections try to improve on this work, either through the use of homomorphic encryption or secure multiparty computation.

Secure Training with Homomorphic Encryption

Following *SecureML, CryptoDL2* [185] was the first to focus on homomorphic encryption for training and inference. It implements FHE using the HElib library, and approximations of activation functions were based on Chebyshev polynomials [200] In comparison to *secureML*, it offers both a significant speed up and an increase in predic-

tive performance. However, the approximated polynomial activation functions caused training to be unstable. To manage the computational complexity and noise, by avoiding the costly bootstrapping of homomorphic encryption, their approach allowed some communications between client and server when the noise reaches a threshold. While this increases communication overhead, it is still lower compared to SPMC protocols, where several communications are necessary for each operation.

Nandakumar et al. [190] focuses on training a network with FHE fully in the fixed-point domain and achieves convergence and reasonable accuracy. They argue that for other papers that only employ floating point operations for some intermediate computations and truncate them, errors can quickly accumulate and prevent convergence, especially in a non-interactive protocol such as FHE. [190] achieves FHE in the fixed-point domain with bootstrapping by pre-computing in a table and performing homomorphic table lookup. This optimization leads to a 50x speed up compared to their naive setup, with 96 % accuracy.

GELUnet[186] solves the issue of high computational complexity in approximations of activation functions differently. They use partial homomorphic encryption with the Paillier system, but leave the local calculations unencrypted. In this setup, the input for the activation (f.e. the encrypted intermediate weighted sum) is sent back to the client. The client has the key and thus decrypts the input executes the activation, re-encrypts the result, and sends it to the server for the next layer. While this limits computational cost, it leads to higher communication and higher necessity of compute power of the client side. As a main reason for ST can be to avoid this, [186] might not be suitable in many situations.

MScryptonet [189] aims to provide an architecture for the secure training of the model, through the application of a multi-key fully homomorphic encryption architecture. Previous frameworks, such as *CryptoDL2* [185], raise the option of HE model training, but with the encryption of only one key, thereby safely allowing only one client at a time. Thereby encryption is outsourced, but not necessarily collaborative among multiple parties. Multikey FHE can be a solution to these issues. A multi-key fully homomorphic encryption (MK-FHE) scheme allows a server to evaluate circuits over

ciphertexts encrypted under different keys. To achieve ST, *MScryptonet* uses approximations of activation functions with low-degree polynomials. Although training time was not reported, there was a 2.5x speed up compared to *CryptoNets* for inference. While multikey encryption solves the problem of collaboration among clients, another approach is applying SPMC, as discussed in the next section.

Secure Training with Secure Multiparty Computation (SPMC)

Most methods for secure training use MPC techniques. A first example of this is *ABY3* ([184]). Their approach is based on the previously developed ABY framework [199] similar to *secureML*, but extended it for 3 parties with support security against malicious adversaries. In this mixed method framework, they introduce new approximate fixed-point multiplication protocols for shared decimal numbers. This provides a more workable framework: next to better security, they report a speed up of 55000x in the most extreme scenario compared to *SecureML*.

Concurrently, Wagh et al. [201] provides a secure training setup for CNNs. It provides three- and four-party secure computation protocols, with various neural network building blocks such as matrix multiplication, convolutions, ReLu, Maxpool, and normalization. Its predictive performance is good, providing a 99% accuracy with a speed up towards both secure inference and ST setups. The protocol is established in such a way that it also protects against attacks from a malicious adversary, where no single party learns any information about the data. By creating more efficient communication, and eliminating oblivious transfer protocols, it creates up to a 407x increase over *SecureML*.

Another approach is taken by Agrawal et al. [188], which focuses on neural networks with ternarized network weights, similar to FHE-DiNN [155] and *TAPAS* [163]. Therefore, ternary matrix-vector multiplication is a crucial primitive for training. they implemented a protocol for ternary matrix-vector multiplication which is based on Correlated Oblivious Transfer [202] that combines Boolean-sharing and additive-sharing for efficiency. Compared to *secureML,* they obtain an improvement of 50X in WAN time and 6% in absolute accuracy while working in the honest but curious setting.

Falcon [195] provides a 3PC setup that gains efficiency by a factor of 2 by rebuilding a derivative of ReLu. Moreover, to create a better accuracy, they also enable batch normalization. It offers MA security: Falcon provides for secure training in an honest-majority adversarial setting. It is 4.4 times faster than *ABY3* and 6Œ faster than *SecureNN*.

Blaze [193] takes an approach with three servers, tolerating up to one malicious corruption, using new secret-sharing semantics. They also introduce a dot product protocol, where communication is independent of vector size, and a less expensive truncation method than *ABY3*, leading to an increase of up to 278x for neural networks. *Swift* [192] extends Blaze to offer maliciously secure, three-party computation (3PC). It is robust in the honest-majority setting, and also offers maliciously security for 4PC. Moreover, it is as fast as Blaze.

Trident [196] and *Tetrad* [198] build on the methods from *Blaze* [193] and *SWIFT*[192]. *Trident* [196] uses 4PC and has an additional honest party in the protocol. This leads to a speed up of up to 45x compared to *ABY3*, for CNNs. While *Trident* offers fairness, *Tetrad*s [198] protocol also comes with robustness. They provide this robust option by figuring out the trusted party in the protocol and delegating the calculation to them. While this is secure in a cryptographic sense, it is not the privacy-preserving solution that would be preferable in distributed learning.

Fan4 [191] focuses on providing a robust 4-party system. They mention that previous frameworks provide their robustness by delegating the calculation to a trusted party. While this is secure in a formal sense, in a real-world scenario most participants will not accept this as a suitable solution. *Fan4* [191] introduces robustness, as well as private robustness, where the protocol can continue privately. They achieve this in an actively secure 3PC protocol, for one corruption. *AriaNN* [197] uses their previously developed method of function secret sharing [203] This is different from classic secret sharing, where a shared input is applied to a public function. Function secret sharing applies a public input on a private shared function. While offering a relatively minor speed up to previous methods, their framework is implantable on a GPU.

A.4.3 Summary of Collaborative Learning

We can see both for secure training and secure inference, the increase in accuracy, efficiency, and security as methods have improved over time. Importantly, we also start to see protocols, especially for ST, that also take into account malicious adversaries. The two situations, SI and, ST, are highly related to each other, however, the ideal setup can differ between the two purposes.

For secure inference, the largest group of papers consists of FHE-based security. Those methods assume Honest-but-Curious clients and servers, though some, such as *XONN*[166], do remark on the possibility of adapting the framework to withstand a malicious adversary. For secure inference, a key question to raise is the extent to which a higher level of protection would be necessary. It is important to remember that most of the private training set-ups can also be used for private inference, so more secure and robust frameworks are available for secure inference.

With regards to secure training, implementing a situation protected against malicious adversaries seems more important, with a preference for MPC methods for private training. However, few papers have included differential privacy in their methods. Most papers only look at the safekeeping of the data and do not seek to combine this with verifiability. Especially secure training could be sensitive to a poisoning attack, which involves the manipulation of training data to inject false information into the model which can hinder the training process. Incorporating mechanisms for verification and accountability would be a critical step toward enhancing the security of CBL systems. Moreover, it is worth noting that there are still issues regarding excessive computational times. While this is an active problem and efficiency is continuously improving, an alternative approach may involve distributed learning, as this can delegate computation to the participants thereby potentially reducing computational time. This topic will be expanded on and discussed in the next section.

A.5 Distributed Learning

One of the most prominent methods within distributed learning is *Federated Learning (FL)* [27]. FL is a specialized form of distributed learning that enables multiple participants, each with their local data, to collaboratively train a shared model while keeping the data localized. Most of this chapter will focus on FL, which we will describe in the next section. However, It is important to note that several configurations of distributed learning do not follow this architecture, most notably *Split Learning* [143], where one model is split up among different parties. We will describe these methods in section A.5.4.

A.5.1 Description of a Federated Learning Workflow

FL Overview

FL involves iterative model training across decentralized clients (For thorough discussion, See [11]). Clients train local models on their private datasets, transmitting updated parameters to a central server. The server aggregates these updates (e.g., via federated averaging) to produce a global model, subsequently redistributed to clients for further training. This process continues until convergence or a predefined performance criterion is met. This approach enables robust deep learning while preserving data privacy and mitigating centralization risks.

FL, while promoting distributed training without direct data sharing, does not inherently guarantee privacy. This misunderstanding stems from conflating data isolation with data privacy. Although raw data remains local, shared model updates (gradients, weights) implicitly reveal information about the training data, vulnerable to inference attacks. Sophisticated adversaries can exploit these shared updates to infer properties about the original data, potentially leading to privacy breaches. The central aggregation process lacks intrinsic privacy mechanisms, creating further vulnerabilities to malicious actors or data interception. Repeated communication rounds exacerbate this risk.

It is crucial for our main discussion to distinguish between *cross-silo* and *cross-device* FL, as these variations impact the scale and nature of participation in the distributed learning process. In cross-silo FL, the setup typically involves fewer clients; however, these clients are usually entities that possess substantial datasets and computational resources. This contrasts sharply with cross-device FL, where a larger number of less reliable clients participate, but each typically has access to smaller datasets and has less computational powers. Cross-device FL scenarios often require tailored privacy-preserving algorithms to address the unique challenges presented by the scale and reliability of the devices involved, and the risk of data exposure or loss may be heightened due to the increased number of less secure endpoints.

Federated Learning Security and Privacy Workflow

Similar to the previous chapter, our focus remains on FL algorithms that are specifically designed for privacy preservation. We visualize the security and privacy mechanisms in the FL workflow in Figure A-3. To address the inherent privacy vulnerabilities in FL, the integration of differential privacy (DP) provides a robust solution. The DP noise-adding mechanisms can be employed in different parts of the workflow. This will have implications for the performance of the pipeline, the privacy level, and the privacy surface of the model. To better understand this surface, we divide the differentially private federated deep learning methods into two broad domains: *local DP* and *global DP*.

Global differentially private federated learning In a Global Differential Privacy (GDP) scheme, a global, trusted curator receives non-perturbed parameters from involved parties in response to a query and applies random noise to the output of any queries being performed against the system. The randomized mechanism M is (ϵ, δ)-*differentially private* in the global domain if it satisfies Equation (B.5).

Local differentially private federated learning In Local Differential Privacy (LDP), each node applies its own set of noise to the network parameters, and what the curator receives is the perturbed data[14]. This ensures that if the orchestrator behavior is adversarial, the amount of information leakage will be bounded by the DP guarantees

of the system. The randomized mechanism M is (ϵ, δ)-*differentially private* in both local and global domains if it satisfies Equation (B.5).

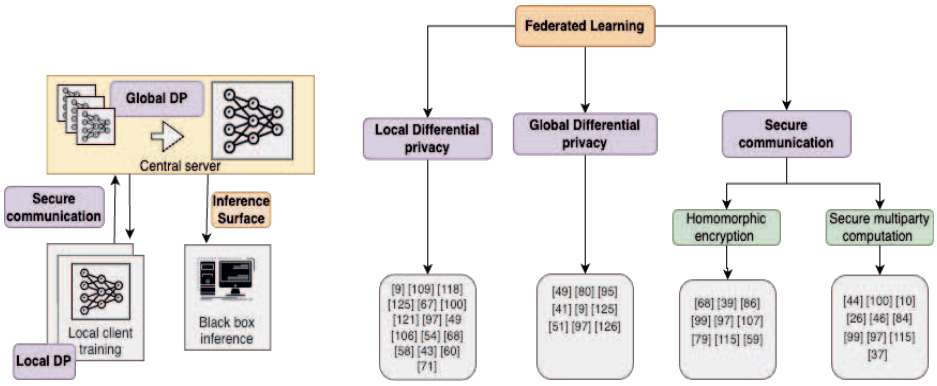


Figure A-3: A FL workflow demonstrating where privacy-preserving measures protect model inference and secure communication safeguards the training process (left). The distribution of research papers investigating these protective mechanisms is shown on the right.

Local and global privacy are added at different points in the FL workflow as can be seen in figure A-3, but they are both based on similar noise-adding techniques. These techniques and their main challenges, such as the utility/privacy trade-off, in FL are discussed in section A.5.2.

Beyond privacy, FL requires robust security measures. Cryptographic methods are employed to secure communication between the orchestrator and participants, protecting the integrity and confidentiality of both model updates and the aggregated model. section A.5.3 explores these security challenges, including scalability and verifiability. section A.5.2 covers approaches that combine noise-addition and cryptography to simultaneously address both privacy and security concerns.

A.5.2 Differential Privacy in Federated Learning

Noise-adding is the main method to achieve differential privacy in FL. The noise is usually added during the training to the internal state of the model, e.g., the model gradients. Considering the general workflow of FL with participants as the data holders and top and mid-level actors serving as orchestrators, the noise-adding mechanism can be employed in different parts of the workflow. The process of adding noise is carefully

calibrated to ensure that while the individual data points are obscured, the overall utility of the model is not significantly compromised. This balance is critical in maintaining both the performance of the model and the trust of the participants. A considerable number of proposed methods in the area of DPFL are application-oriented, as they trade the regular optimization method of their model with a DP version, usually DP-SGD, for local DP - e.g. [211, 222, 223, 224] - or global DP - e.g. [204, 212, 225]. The methods we cover in more detail in this section provide additional privacy and/or performance-related novelties. We summarize the selected differentially private FL workflow papers in Table A.4.

Private Training and Inference in Federated Learning

Geyer et al. [205] propose a global-DP-based FL mechanism based on adding noise to the communicated gradients of each client by the orchestrator. They use a random subsampling mechanism to pick K out of N participants for the parameter aggregation phase. Then the updated model parameters are only sent back to this subset of participants.

Beaulieu-Jones et al. [211] use a cyclic approach, where each Learning Party will train their model until they spend their privacy budget, then pass on the computed differentially private set of gradients to the next Learning Party.

Zhao et al. [224] communicate the network weights from Learning Parties to the parameter server and average them to obtain and send back the optimum global model parameters for the next training iteration. They use a publicly available dataset to validate the performance of each learning party's local model and only choose the best-performing ones to average over to mitigate the problem of unreliable partners with noisy and low-quality datasets. To make sure the learning parties cannot infer which are dropped and which are selected, the parameter server uses the Exponential Mechanism to protect the selection mechanism. The total privacy budget will be the largest of the two privacy budget values - on the learning party to perturb the objective function and on the Parameter Server to obfuscate the selection process.

Hartmann and West [209] follow the local DP FL scheme by adding uniform noise to the local gradients before submitting them to the parameters server. However, on the orchestrator level, the collective anonymized noise values will be subtracted from the cumulative gradients at each step to achieve higher utility.

Wang et al. [217] propose a federated DP Latent Dirichlet Analysis (LDA) mechanism for privacy-preserving distributed text mining. The orchestrator is assumed to be untrustworthy. They propose a modified version of Warner's randomized response method called *random response with priori (RRP)* to make their federated LDA mechanism DP on local nodes. The orchestrator will attempt to infer the topic-word distribution from DP local updates received from each participant. The RRP method non-uniformly samples the noisy terms a priori drawn from the local model.

Seif et al. [210] consider a FL system connected over a wireless channel and propose leveraging the superposition characteristics of a wireless channel to preserve the privacy of the communicated gradients. They make use of one flat-fading Gaussian Multiple Access Channel (MAC) bound by Local DP constraints. The channel coefficient and additive Gaussian noise determine the amount of perturbation applied to the gradient sets. They provide tighter privacy bounds dependent on the number of clients on the wireless channel through their proposed transmission scheme.

Huang et al. [215] use the Alternating Direction Method of Multipliers (ADMM) to perform their distributed learning task and augment it with differential privacy. The ADMM method optimizes a Lagrangian function associated with the global loss function by alternately minimizing it with regard to local models and the global models. Based on the intuition that it is not necessary to solve the local optimization problems to a very high precision to guarantee global convergence, the Lagrangian function is replaced by a first-order approximation of itself. The authors use the Gaussian noise-adding mechanism to achieve DP and vary the variance of the noise with time, decreasing its magnitude as the number of iterations increases.

Kairouz et al. [219] address the fact that most DP-SGD-based methods rely on privacy amplification through subsampling by carefully choosing mini-batches, which can be challenging in a distributed setting. They propose the follow-the-regularized-leader

(FTRL), a DP-distributed training algorithm that does not rely on subsampling. FTRL adds correlated noise to the gradients as opposed to the DP-SGD algorithm in which independent noise is added to the gradient vector.

Although most differentially private distributed deep learning methods are in the context of generative and discriminative models, other types of deep learning methods can also benefit from this paradigm. An example is the work by Gohari et al. [218] on differentially private reinforcement learning. This work considers a deep reinforcement learning setting in which agents would ask for a demonstration from other agents to improve their performance following a teacher-student scheme. In the case where the private dataset of the agent contains sensitive information, the demonstration opens up the teacher to privacy attacks such as membership inference. They limit the context of demonstration to the teacher's policy and use the Dirichlet mechanism to privatize the discrete-space policy of the teacher.

Reducing the Impact of Privacy/Utility Trade-off

The balance between privacy protection and data utility is a fundamental challenge in the design of differential privacy mechanisms. While stringent privacy guarantees often necessitate the introduction of noise or other obfuscating alterations to the data, these modifications can detrimentally affect the utility or accuracy of the resulting models. In distributed learning systems, this trade-off becomes even more critical as the system must maintain high data utility to ensure effective learning and decision-making, while also upholding rigorous privacy standards. Addressing this challenge requires innovative approaches that optimize noise addition and data processing techniques, thereby minimizing the loss of utility without compromising the privacy of individuals. This section explores various strategies and methodologies developed to improve this delicate balance, aiming to achieve optimal outcomes in both privacy preservation and the practical usefulness of data.

Shokri and Shmatikov [204] in their seminal paper use the Laplacian mechanism of differential privacy to protect the communicated gradients. But first, they select only a subset of the most prominent gradient values for communication and the Laplacian noise

(generated according to a fixed budget) will only be added to this subset of gradients before communicating them to the parameter server.

Liu et al. [226] propose a DP FL scheme with a dynamic noise-adding mechanism for improved utility. They employ an extended version of the *layer-wise relevance propagation* algorithm and add noise calibrated to the calculated relevance of each attribute class. Furthermore, they propose a randomized privacy adjustment mechanism in which a threshold for the efficacy of each feature is defined by the user. The threshold is then used in a method similar to Warner's randomized response to decide whether and with what probability to add noise to a feature or just pass the data unperturbed.

Liu et al. [216] adapt the amount of noise they add to the data by using a modified relevance propagation algorithm to measure the contribution of each neuron of the input layer and scale the noise based on its impact on the model output. They further threshold the contribution of each feature with a parameter f. Features with contributions above the threshold will have noise injected into them while for the nodes with contributions lower than the threshold, a randomized response mechanism is used to decide whether noise is added to the data or the original data passes through. Noise sampled from a Laplace distribution is added to the loss function.

Zhang et al. [214] design a three-tier FL mechanism to perform deep learning on resource-constrained edge devices. They pre-train the model on public data and then partition the model on the last layer among edge nodes to reduce their computational overhead. They apply local DP to the model on the client side.

Kim et al. [220] use an enhanced privacy budget accountant in a federated DP-SGD scheme and provide a variable sensitivity parameter - as opposed to a fixed bound on gradients - leading to tighter privacy bounds. Furthermore, they provide an analysis of the communication rate, privacy level, and model utility which leads to a new way to calculate the DP noise variance, resulting in a lower transmission rate and higher utility while maintaining the same level of privacy.

Reinforcing Privacy while Maintaining Security

We earlier discussed the privacy and security risks differ by nature and require different treatments. DP by nature can protect the training and inference surfaces of a distributed learning system against specific privacy attacks. However, to mitigate security risks, existing research often combines private and secure training and inference methods. This could potentially complicate the dynamics of both the training phase and inference phase with regards to compute/time costs and the trade-off between utility and security/privacy and needs careful consideration and development. One example is the work by Hao et al. [212], which combines local DP with homomorphic encryption to prevent malicious parties from modifying the communicated gradient values. They encrypt the DP gradients with PPDM additive homomorphic encryption method [227] to provide a more lightweight option. This allows for an encrypted addition of the model updates at the server level. They propose a scheme that is robust against participants dropping out, and that preserves privacy against malicious Learning Parties. Phuong et al. [225] also use homomorphic encryption in addition to DP for private, secure training of a deep model.

Hartmann and West [209] imagine a situation where the parameter server is malicious and could collude with several Learning Parties. To mitigate privacy concerns, they add uniform noise to the local gradients and send them to the parameter server. Simultaneously they send the actual noise vector through an anonymizing service like Tor so that the server cannot recover the exact gradient values of any single Learning Party, but would be able to subtract the sum of noise values from the sum of gradient values and compute the gradient values for the next step of the training process with high utility. Truex et al. [213] combine local DP with secure multiparty computation between participants to provide security during training. Using an adjustable security parameter they are able to reduce the amount of calibrated noise added to the local model parameters during the training phase.

Hao et al. [212] apply both DP and homomorphic encryption to the FL protocol, by using additional HE to provide a more lightweight option. It does this by applying ϵ-DP by the Laplace mechanism to the model updates, and thereafter encrypting them in an

AHE applicable manner. This allows for an encrypted addition of the model updates at the server level. They propose a scheme that is robust against participants dropping out. This is expanded and improved on by [228], who propose a multi-key HE scheme, this method is secure when the server colludes with one of the participants.

Sun et al. [206] augment the local DP FL pipeline by masking the identity of each participant from the orchestrator through the utilization of techniques such as IP masking or secure multiparty computation. The weights of local models are split and shuffled before being received by the orchestrator. Each client samples a random time for sending its parameters to the server. Each set of weights is split and then the weights of different local sets are shuffled before being sent to the orchestrator. On the local level, the authors add noise to the gradients of the deep net using a randomized noise-adding mechanism to make sure the overall noise has a zero mean.

Zheng et al. [208] introduce a DP FL mechanism utilizing the notion of Gaussian f-DP [229]. They address two threat models, one assuming one malicious learning party and the other assuming several malicious parties, and introduce *weak* and *strong federated f-differential privacy* schemes to deal with these threat models respectively. The general framework involves training customized models on each client - as opposed to federated averaging in which at each round, clients adopt the global model - with the help of a noisy global helper model. At each round of training, the selected participants pull the helper model and utilize it to increase the efficiency of their private local model. The local DP model is then synchronized to the server to update the helper model.

Malekzadeh et al. [221] propose *Dopamine*, a federated differentially private deep learning mechanism for medical data processing. They employ DP-SGD for local training. To provide hospital-level privacy guarantees, they use a secure aggregation method based on homomorphic encryption which is performed on collected local noisy DP models.

The work of Hao et al. [212] is expanded and improved on by [228], who propose a multi-key HE scheme in a similar way. This way, the method is secure when the server colludes with one of the participants.

Ma et al. [230] also use multikey HE, specifically a multi-key protocol based on CKKS. There, the model updates are encrypted via an aggregated public key before sharing

with a server for aggregation. Collaboration between all participating devices is required for decryption. It is secure for collusion until n-1 participants.

A.5.3 Security Methods for FL

Not only differential privacy mechanisms are important for safe distributed learning, but security measures can also be incorporated to improve data and model privacy. The papers discussed in this section are summarized in Table A.5.

Security of the Updates

Phong et al. [231] showed that in the DP setup of [204], the shared gradients would still cause severe information leakage of the local data. To prevent this, they proposed AHE with the Paillier system and learning with errors (LWE) [232] to encrypt the shared gradients. They specifically consider the concept of a dishonest server. It leads to higher security, but it does not consider the dropout of participants.

HybridAlpha [233] takes a different approach by using functional encryption as a security measure. For functional encryption systems, the decryption key does not allow the user to learn the encrypted data, but instead only a function of the encrypted data. *HybridAlpha* uses this to provide security with a dishonest client and an HBC server. Moreover, *HybridAlpha* combines this with ϵ-DP, added by the participating parties. *BatchCrypt* [234] propose an AHE scheme for encoding of batch gradients, as batched training might lead to better predictive performance. It does not increase the communication costs, but does lead to an increased computational cost for the clients. *SSDDL* [235] uses secret sharing, but instead of sharing gradients, the clients split their local gradients into shares and distribute them to each other. When they receive the shares the participant calculates aggregation results and uploads that to the server which updates the global network parameters. It is secure for an HBC server colluding with HBC participants. *DAEQ-FL* [236] uses Paillie, with efficient threshold encryption. This leads to good predictive performance, and lower computation and communication costs. Moreover, it works with no trusted third party and can handle non i.i.d. data.

Security of the Aggregated Results

While most papers focus on the clients gradients , some also argue for the security of the final model weights. *PPFDL*[237] created a protocol based on the Paillier Scheme and Garbled Circuits. They incorporate a method to reduce the impact of irregular clients on the accuracy. While increasing the server's computational cost, the clients' computational and communication costs are significantly lower than [238]. The scheme also protects the aggregated result. *Poseidon*[239] uses multiparty homomorphic encryption (CKKS scheme) with distributed bootstrapping. It focuses on protecting both the model updates as well as the final model weights. It functions under an HBC model with n-1 parties colluding. Moreover, it argues to be the first framework that allows for quantum-resistant distributed learning.

Security of Cross-Device FL

Most of the papers in this chapter focus on cross-silo FL, where a few clients with larger amounts of data participate in FL. The alternative is cross-device FL: many clients with smaller amounts of data, f.e. mobile devices. For cross-device FL, it is essential to handle large groups of participants and be robust to clients dropping out. Secure Aggregation [238] proposed the first approach to FL that is both privacy-preserving and robust to users dropping out. It privately shares the gradients between the clients and the server using additive secret sharing: each client double masks the gradient and uploads that to the server. Sharmirs t-out-of-n secret sharing scheme is adopted to handle the dropout of participants. When receiving enough masked gradients, the server aggregates them and the pairwise masks cancel out. They prove that even when the server and t-1 clients are colluding, no client information is leaked. However, while the pairwise structure is secure and robust to dropout, the overhead grows quadratically with the number of clients, which limits its application to large-scale scenarios. Several frameworks have aimed to improve the overhead and speed up aggregation.

Bell et al. [241] continues on secure aggregation by proposing a protocol that allows for (poly)logarithmic overhead. They argue that every client double masking with all the other clients is not necessary, and they propose a more efficient communication graph.

Their HBC secure aggregation claims to handle a billion clients, and their more secure semi-malicious setup can handle thousands of clients. Their scheme can handle up to 5% of corrupt clients and 5% of dropouts, and allows for differential privacy through secure shuffling [242] FastSecAgg [243] focuses on speed up of the aggregation by developing a novel multi-secret sharing scheme, FastShare, which is based on the Fast Fourier Transform (FFT). It leads to computational speedup with similar communication and protects against adaptive adversaries. FastSecAgg does struggle with handling dropout of clients (10%). TurboAgg [244] limits the overhead to O($NlogN$) from O(N^2) and is robust to a 50% dropout. It limits overhead by performing the SS in groups. The users are divided into several groups, and at each aggregation stage, the users in one group pass the aggregated models of all the users in the previous groups and the current group to users in the next group. The running time is almost linear with the number of clients.

Verifiability

VerifyNet [250] broadens the scope of privacy protection by focusing on verifiability next to confidentiality. This is achieved by exploiting a homomorphic hash function into the underlying structure of VerifyNet with a double-masking protocol to guarantee the confidentiality of users local gradients during the FL. It does not only consider the confidentiality of the model updates to the server, but also the integrity of the returned average. As such, it offers protection against a malicious server. However, the double masking protocol is computationally expensive, and a trusted third party is still needed to establish keys. [248] is an extension of Verifynet: They also use a double-masking protocol, but propose a secure scheme where the key generation center is HBC.

A.5.4 Other Distributed Learning Architectures

Split Learning

Split learning (SL)[143] is another method that falls under distributed learning. With SL, a deep neural network is split into multiple sections of several layers, each of which

is trained by different parties. Generally, this process works by transferring the weights of the last layer of each part to the next in forward propagation, without sharing any raw data only the weights of the last layer (the cut layer) are sent to the next client. Afterwards in the backpropagation, the gradients will be sent back: from the first layer of the second party to the last layer of the first party. SL allows for different configurations, including vertical SL, where the first layers are vertically split for vertically partitioned data, or the last layers are trained by the party with the labels. It also allows for a combined federated and split network, where the earlier layers are trained in a federated manner [251].

While SL offers promising new methods of handling distributed data and models, it comes with several security and privacy issues, due to the sharing of the cut layer during forward and backward propagation. The different possibilities in distribution can also lead to different privacy issues. [252] highlights the necessity for extra security or privacy on SL by showing its susceptibility to inference attacks both from malicious servers and malicious clients. They also showed that previously proposed distance correlations [253], which showed more privacy to reverse engineering attacks than DP-SGD, are not secure. Moreover, [254] highlights how even an HBC server can infer the data from participants, especially if the NN does not have many layers in depth. Recently, there have been several studies that offer potential solutions to these insecurities in SL. [255] highlights another issue: label leakage. In a setup of one party having the features and the other party the data, the second party can recover the labels during the training process. This paper protects the confidentiality of the label by adding noise perturbation that is optimized to prevent label leakage. [256] offers a solution based on HE with a CKKS scheme for a single client and server scenario. It encrypts the clients cut layer gradients that serve as input for the server layers.

Ensemble Methods

Ensemble methods are a fundamental technique in machine learning that involves combining the predictions from multiple models to improve the robustness, accuracy, and generalizability of predictions compared to any single model alone. These methods are

based on the principle that a group of "weak learners" can come together to form a "strong learner." Ensemble methods and FL both involve training multiple models on distributed datasets, but they serve different primary purposes. FL focuses on training models collaboratively across multiple decentralized devices or servers (clients) holding local data, without sharing that data centrally, thereby preserving privacy and reducing data centralization. This framework can naturally incorporate ensemble methods by using the local models trained on each client as members of an ensemble, where their predictions can be aggregated to improve overall model accuracy, robustness, and generalization. In the context of distributed differentially private deep learning, ensemble methods can be particularly beneficial as they can enhance model performance without compromising the stringent privacy requirements. For example, each client can add noise to their local model's outputs (or during the training process) to meet differential privacy criteria, and the central server can then aggregate these differentially private models, potentially using techniques like weighted voting or averaging. This aggregation can help in mitigating the noise added for privacy, leading to both a robust ensemble and adherence to privacy guarantees, effectively balancing utility and privacy in distributed environments.

In the context of DP ensemble methods, one of the most prominent examples is the Private Aggregate of Ensemble Teachers (PATE) framework. Papernot et al. [257] proposed this framework as a DP learning method where multiple teacher models, each trained on disjoint subsets of sensitive data, contribute to the training of a student model without directly exposing the sensitive data. The PATE approach aggregates the outputs of the teacher models such that the privacy of the underlying data is preserved, typically using mechanisms like noisy voting. Several variants enhance the robustness and privacy guarantees of the PATE framework. For instance, Papernot et al. [258] introduce a scalable version of PATE with granular control over the noise added to the teacher's outputs. The aggregation of teacher responses is followed by the addition of noise calibrated to the sensitivity of the counting queries, which are indicative of the number of teachers agreeing on a particular label. This noise addition is governed by the Gaussian or Laplacian mechanisms, aligning with DP standards. It also addresses the challenge of scaling PATE to a larger number of teacher models and

a broader set of data classes. Structuring the teacher ensemble into several smaller groups and performing noisy aggregation within each group before a final aggregation step reduces the overall noise required for a desired privacy guarantee, which could improve the utility of the student model. PATE framework has been since adopted for many different use cases, e.g. G-PATE by Long et al. [259] which adapts the core principles of PATE for generative adversarial networks (GANs), enabling the training of DP data generators that can produce high-quality synthetic data while preserving the privacy of original dataset with private aggregation happening among different discriminators.

A.5.5 Summary of Secure and Private Federated Learning

Due to vulnerabilities of federated and other distributed learning methods to information and model leaks, privacy-preserving methods to mitigate these problems are being increasingly developed and advanced. The majority of methods for privacy preservation in distributed learning in literature are differentially private methods. Following the proposal of the DP-SGD method [101], many deep learning methods have adopted this DP optimization method to perform FL in a privacy-preserving manner. The selection of the DP regime is usually problem dependent and impacts both the utility of the model and the "measured" privacy budget spent. The threat model also has an important role in major decisions like whether to implement global or local DP. However, these choices in the majority of the existing works in literature do not impact the privacy method itself - DP-SGD. The methods we focused on in this section push the private training and inference of distributed DL model further. The main motivation for the reviewed DP methods can be categorized into *improving the privacy/utility trade-off* and *improving the security as well as the privacy aspects*. Since adding privacy to DDL models generally comes at the cost of utility, the methods focused on this aspect try to improve the utility of the model while maintaining an acceptable level of privacy. On the hybrid approaches, the researchers recognize that in some use cases the privacy itself is not enough and based on the assumptions about the actors - or lack thereof - and the setup of the training and inference pipeline, we need to combine security and

privacy methods such that the method would provide acceptable privacy and security, high utility and a seamless combination of these two aspects.

With regard to secure communication in FL, we notice a difference between cross-silo and cross-device FL. Most DP methods focus on cross-silo FL. HE seems to have the preference in cross-silo situations, while SPMC, and especially secure aggregation based on SS are often applied to cross-device settings. This might be related to the scalability of the different methods. Different from the CBL frameworks in section 4, we see fewer papers that are resistant to a malicious server, only [241] considers a semi-malicious scenario. In section A.5.2, we discuss the methods that combine security and privacy. It is an essential avenue of research that aims to offers solutions for both security and privacy threats. Many of these methods either offer improvement on the privacy or the security side of the mechanisms. It is important that these papers properly investigate the issues that come both with differential privacy and cryptographic methods.

We also regarded two other distributed learning methods; split learning (SL) and ensemble methods. SL offers an interesting new avenue in distributed methodologies. Not only could off-load model computation to different parties, but it also shows opportunities for handling vertically distributed data, which is often more complicated and understudied. However, SL still has significant security and privacy issues that need addressing. Ensemble methods are an interesting concept that could potentially help reduce the impact of noise-adding for differentially private FL. Due to the configurations of the framework, it is also scalable and offers opportunities for GANs. However, an evaluation of robustness in security would be beneficial.

A.6 Discussion

A.6.1 Design trade-offs for PSDDL

When designing a PSDDL system several trade-offs can be made, as illustrated throughout this paper. There is not a single best solution that addresses the privacy and security needs for each possible type of distributed workflow. sections 4.4 and 5.4 provide a sum-

mary of how workflow impacts the protective mechanism choice. In this section, we highlight several design choices that need to be considered when deciding on PSDDL methods. We highlight trade-offs of security and privacy methods, and differences between DDL and CBL. Other factors that play a role are how the DP is accounted, and the threat level and model.

Security/Privacy and Accuracy/Efficiency Trade-Offs

Several papers report an inverse relationship between enhanced privacy measures and operational efficiency (or accuracy) of the system. This inverse relationship necessitates a detailed articulation of system requirements, prioritizing either privacy or efficiency based on the specific needs of the application. The implementation of DP methods further demonstrates the trade-off between privacy and accuracy. DP is a technique designed to ensure that the output of a database query is minimally affected by any single individual's data, thereby offering strong privacy guarantees. However, as privacy constraints are tightened (by reducing the probability of an individual's data influencing the output), the accuracy of the data output can diminish. This reduction in accuracy occurs because more noise is introduced to mask individual contributions, which can obscure meaningful insights.

Moreover, the deployment of advanced security methods often results in decreased system efficiency. Even with modern advancements in cryptographic techniques and secure computation protocols described in section 4 and 5, we still see a significant computational overhead introduced by these security measures compared to unencrypted computations. While the direction of research should, and will, focus on more efficient protocols, it is important to highlight this pay-off. Deciding on the level of security requires a careful look at what the application needs and what risks it faces. For example, while the strong security frameworks from secure training can be useful in secure inference, the high computational cost is an important consideration. Evaluating the necessity of advanced security measures can help find a balance between maintaining security and ensuring the system works efficiently. Similarly, using both DP and ENC can greatly improve accuracy and security. However, it's important to think about whether

such high levels of security are always necessary. For example, if the task involves only inference and not training, simpler security measures might be sufficient. Decisions on using these advanced technologies should consider their impact on computational resources and whether they are essential for the specific application.

These points of pay-off between security and privacy versus accuracy and efficiency lead us to an important issue in the field of PSDDL. Currently, there is a notable absence of a comprehensive framework to quantify the trade-offs between privacy, accuracy, and efficiency. Such a framework would facilitate the evaluation of different security and privacy methods under varying operational constraints, workflows, and preferences.

Collaborative Learning vs Federated Learning

One major challenge with CBL is speed. Although research efforts are continuously making these methods faster, the high computational demands can still be a significant issue. An alternative approach in that situation may better be found in a FL set-up as this can delegate computation to the participants thereby potentially reducing computational time, especially if scaled in a cross-device scenario. Therefore, where computational power is available should be taken into account, combined with the loss in efficiency for all set-ups. Another issue that can affect the choice between CBL and FL is the choice of data and model ownership. In a collaborative situation, while secure, the data is still sent to the central server. Thus, the parties are not fully in control of what happens with their data. Similarly, in a situation where the central party wants to have control over or ownership of the model, it might not want to share it with FL clients. Whether due to risk minimization, business incentives, legal or ethical constraints, this should be considered in the multifaceted decision of selecting a PSDDL set-up for a particular use case.

Differential Privacy Accounting

Privacy costs of noise-adding are incurred by the iterative running of the DL algorithm on the training dataset. Calculating the privacy loss of the DP algorithm during the training phase, however, is not a trivial task. Different "accounting" methods exist

for the composition of DP guarantees of DP mechanisms. Usually, the accounting mechanism works under a specific DP regime, e.g. f-DP, and supports the application of certain DP mechanisms, e.g. adding calibrated noise from certain distributions. These accounting methods range from the application of composition theorems and advanced composition theorems of DP [34] to widely adopted moments accountant [101] to more recent methods such as Edgeworth accountant [260]. The problem definition limits the choice of the regime, mechanism, and accounting method. Nevertheless, care needs to be applied in the selection of each aspect of the privacy-preserving mechanism to increase the chance of success in training a model with high levels of privacy.

Threat Models and Security LevelS

As discussed in section A.6.1 the problem definition limits the range of available choices while designing a DP algorithm. One of the problem aspects that can inform the problem definition is the security and threat parameters. Putting aside the black box access level of the external party, assumptions about the participant and orchestrator - e.g., trusted, honest but curious - could impact the choice of privacy preservation mechanism and also the performance and privacy level of the outcome of the training as well as the training process itself. One of the main points to mention here is the issue with homomorphic encryption. Methods with HE, especially secure inference by HE, do often not contain proper security analyses. HE can be considered Honest-but-Curious by itself, but in real-life scenarios, the security requirement is often stricter. Moreover, there are also still the issues with regars to GDPR, as there is still discussion about whether protection by HE would be GDPR compliant [261].

Finally, as pointed out in [191], there can be a difference between a formal definition of privacy, and how this would be regarded in practice. Privacy as defined by stopping information leakage by models can happen during two stages: training and inference. For inference privacy, since usually a central model trained in a distributed manner on several disparate datasets will be exposed to real-world use cases, taking measures such as making the model differentially-private should protect the model. However, during the training phase, the malicious actors could be either the participants themselves or

the orchestrator. Considering the flow of information in a typical distributed ML setup, i.e. partial updates shared by participants with the orchestrator and central updates shared by the orchestrator with the participants, to ensure privacy in a no-trust setup we need to make the output of both the orchestrator and the participants private. In DP, this will be done through global and local DP methods respectively.

A.6.2 Current Research Gaps of PSDDL

While the field of PSDDL has grown substantially in the last few years, there are still some issues that remain understudied. We have highlighted some of the current issues in PSDDL, that are important for the safe, accurate, and fair application of PSDDL in practical scenarios. We will highlight the issues that we believe deserve more detailed studying,

Data Distribution and the Problem of Non-i.i.d Data

Since the focus of this paper is on DDL methods, it would be beneficial to look at the assumptions the reviewed methods make about data distribution. Most assume independent and identically distributed data among different nodes. Furthermore, these methods usually assume i.i.d classes for classification tasks. While this would help in developing research, real-world use cases often don't benefit from such an advantage. On one hand, in many cases there exists some form of skewness and class imbalance in the data. It can be easily observed in the standard vision datasets such as CIFAR and MNIST which are used by almost all of the reviewed methods and do not have any imbalance in their classes - digits 0-9 for MNIST, object classes for CIFAR. The situation gets more complicated with tabular data, where the dataset could easily have multiple categorical or mixed features. Even considering a "class" feature for the dataset, it still could be the case that there are under-represented classes in other categorical features, and while the method would potentially show high utility overall for predicting the "class", it would perform poorly for some of the under-represented classes.

The distributed nature of the methods we have reviewed adds another complication which is the distribution of the data among participants. Regardless of the distribution

of the data among classes in the original dataset, many real-world factors such as geographical location and temporal aspects of data gathering could impact the distribution of the data among nodes and lead to datasets of participants having distributions differing from the original distribution of the whole dataset. It is important to note that with collaborative learning, due to the set-up of the server building a singular model, the effect would be less severe than with distributed learning methods such as FL.

Measuring and Validating Privacy Level

In this survey, we have reviewed a wide variety of protective mechanisms for distributed deep learning. Most of these methods employ DP and cryptographic methods, either separately or in combination. Each method provides its own guarantee of privacy, but measuring and validating the actual privacy level of a machine learning pipeline or a trained machine learning model is not a trivial task. DP, for example, provides privacy guarantees in terms of ϵ and δ in the case of ϵ, δ)-DP and similar parameters for other DP regimes. While DP parameters give us the theoretical bounds of the DP of the final model, giving the end users a more tangible indication of "how much" their data and/or model is private if they spent a certain amount of privacy budget is still a challenge. One can showcase the privacy level of DP and non-DP mechanisms by showing their resilience to different types of adversarial attacks, but finding the suitable forms of attack given the problem at hand, implementing and performing them against trained models, and keeping our evaluation up-to-date has its own set of challenges.

Combining Privacy and Security in Collaborative Learning

Privacy and security take a different approach to protecting information. We see that both have their uses and while we see papers combining both in distributed learning, we unfortunately see few papers in CBL that combine cryptographic methods with privacy. Combining both security and privacy can be seen as creating a 'double line of defense' that provided protection against overlapping but different threats. The two types of protection would be able to complement and strengthen each other. While we see a few papers that combine security and privacy for CBL [160], the majority do

not. This is quite different from FL, where a majority of papers consider DP. It is important to consider that, due to the nature of CBL sharing data, vulnerabilities can be different. However, it is still vulnerable to attacks based on model outputs, such as reconstruction attacks, that DP offers effective protection against. Naturally, there is a pay-off in accuracy and efficiency that needs to be considered, but the exact pay-off in CBL should be more extensively studied.

Integrity of Data

While most research regarding protective measures of DDL has focused on providing data (and model) confidentiality, other issues distributed learning can bring forth should not be ignored, an important one being the integrity of data that the cooperating parties provide. We have covered several papers that combine confidentiality with integrity through verifiability [172][167][250], but we believe that it is essential when untrusted parties create a distributed model to have a form of verification of their data. Especially methods that include training could be sensitive to a poisoning attack, which involves the manipulation of training data to inject false information into the model which can hinder the training process. Incorporating mechanisms for verification and accountability would be a critical step toward enhancing the security of collaborative learning systems. If we cannot guarantee the reliability of the data, we can also not guarantee the reliability of the outcomes of the model. While we see some methods in CBL to include integrity, we believe this should be more widespread among all distributed learning methodologies.

Vertically Partitioned Data

We have encountered only a few papers in this survey that are able to perform distributed learning with vertically partitioned data. While vertically partitioned data is not as common of a data distribution as horizontally partitioned data, it can be common within in fields of medicine and finance. As these situations often deal with highly sensitive and possibly easily identifiable data, there must be proper security and privacy mechanisms in place. Most notably, within split learning, there has been some research

into vertical split learning [262]. Unfortunately, many of these methods are shown to be vulnerable to attacks. More research into these vertically distributed data frameworks, also outside of split learning, as well as their protective measures, is recommended.

A.6.3 Emerging Areas of PSDDL

This section of the discussion focuses on privacy and security methods, and frameworks, that were outside of the scope of sections 4 and 5 of this survey, but we believe deserve attention, and they involve distributed frameworks, new security methods, and other aspects of distributed learning, that will promote new areas of research in PSDDL.

Divergent Distributed Frameworks

Although DDL and CBL have been the dominant paradigm in distributed machine learning and despite the fact that they contain a wide variety of methods, there exist other distributed privacy-preserving deep learning methods in literature that can not be classified under these umbrella terms. In this section, we review the prominent methods in this area. Cheng et al. [263] propose a decentralized deep network training scheme by proposing a differentially private iteration of the Leader-Follower Elastic Averaging Stochastic Gradient Descent (LEASGD) algorithm [264]. They apply calibrated Gaussian noise to the update rule of LEASGD, making each step of the algorithm (ϵ, δ)-differentially private. Yan et al. [265] designed a top-down hierarchical approach for deep learning on edge devices. The model is trained in a differentially-private fashion using a Gaussian mechanism. Then, to tailor the trained model for usage on resource-constrained edge devices, the authors weight-prune the network and retrain it, again with DP using a Gaussian mechanism with a separate privacy budget, and publish the compressive models to the edge devices. In this setting, the data is centralized but we have a distributed setting in which using DP helps preserve the privacy of the data on Learning Parties. Yuan et al. [266] propose a distributed learning method, combining local DP and SMC to train a discriminative model. They adopt the definition of label DP, i.e. only label information is considered sensitive. They propose two mechanisms, one based on the randomized response and one adapting the standard

DP-SGD algorithm for label-DP.

Emerging Security Methods

While we mainly focused on the cryptographic methods of HE and SMC, more methods can achieve security. Two prominent examples of new methods are game theory [267] and blockchain [268]. Domingo-Ferrer et al. [267] set up a reputation management system, with parties participating in FL with co-utility. This gives the parties have an incentive to stick with the protocol. In [268] they implement a verifiable and auditable FL framework based on the blockchain system. While not necessarily a new method, one of the main topics is quantum security. With the rise of quantum computers, many 'standard' forms of encryption could be broken. While there has been a rise in research of quantum secure encryption, as of yet there has been little applied to distributed deep learning. Zuo et al. [269] focus on adapting secure aggregation to withstand quantum attacks.

Model interpretations of DDL

Next to model training and inference there are other aspects of deep learning that could be applied in a distributed manner, such as feature selection, explainability, and interpretability. Examples of these techniques in a distributed framework can be found in [270][271]. These methods would aid to create a more complete distributed learning setup. However, if these methods were utilized in DDL, there could be more risk of data and model leakage. Therefore, it would be essential to adapt them to be private and secure.

A.7 Conclusion

In this work, we provided a high-level overview of secure and private distributed deep learning. We categorized different methods in this research area based on their distribution methodologies, namely collaborative learning and distributed learning, provided a comparative summary of the different workflows, and highlighted promising methods

and current research gaps. We provided background knowledge in privacy and security for this work to be self-contained while aiming toward a more general audience.

Framework	Name	Encryption Mechanism	Threat Level		Type of NN	Activation function	Evaluation Datasets	Results on MNIST [153]		Verif.	Novelty
			Level	Col				Accuracy (%)	efficiency		
[154]	CryptoNets	FHE (YASHE)	HBC	ND	CNN	polynomial	MNIST	98.95	297.65 s	-	First paper to use HE for SI
[155]	DiNN	TFHE	HBC	ND	MLP	polynomial	MNIST	96	1.7 s	-	Discretized neural networks
[156]	CryptoDL	FHE (BGV)	HBC	ND	CNN	polynomial	MNIST CIFAR10[157]	99.5	2.1x to [154]	-	Low degree polynomial approximations of activation functions
[158]	MiniONN	GC SS AHE	HBC	ND	CNN LSTM	exact	MNIST CIFAR10 a.o.	99	1.3s	-	Introduces oblivious neural networks
[159]	Chameleon	SS GMW GC	HBC	NC	MLP CNN	exact	MNIST	99	2.7s	-	Combined MPC methods for efficiency, using a trusted third party
[160]	Faster CryptoNets	FHE (FV-RNS)	HBC	ND	CNN	polynomial	MNIST CIFAR10	99	46s	-	Speeds up CryptoNets while adding DP
[161]	DeepSecure	GC OT	HBC	ND	CNN DNN	exact	MNIST a.o.	99	438 to [154]	-	Efficient SI based on GC
[162]	Gazelle	GC SS LHE (BFV)	HBC	ND	CNN	exact	MNIST CIFAR10	ND	0.03s	-	Uses garbled circuits to linearize inference for LHE
[163]	Tapas	TFHE	HBC	ND	CNN	binary	MNIST diabetes[164] a.o.	99	147s	-	Uses TFHE to implement binary NNs
[165]	NGraphHe2	FHE (CKKS)	HBC	ND	CNN	polynomial	MNIST	99	2s	-	Client-aided activation function computation
[166]	XONN	GC OT	HBC	NC	CNN	exact	MNIST CIFAR10 a.o.	99	1.28s	-	Replaces multiplication with XNOR for low-cost OT
[167]	Delphi	GC LHE (BFV)	HBC	ND	CNN	polynomial	CIFAR10 CIFAR100[157]	ND	ND	-	Combines GC and LHE with planner for efficiency
[168]	Cheetah	LHE (BFV)	HBC	ND	NN CNN	polynomial	Imagenet[169] MNIST	ND	79x to [162]	-	Tuning and operator schedule optimisations for speedup, with GPUs
[170]	HCNN	FHE (BFV)	HBC	ND	CNN	polynomial	MNIST CIFAR10	99	5.2s	-	GPU-acclerated FHE
[171]	-	FHE (CKKS, SEAL)	HBC	ND	DNN CNN	polynomial	Imagenet	ND	ND	-	Uses hybrid memory systems to run large neural networks
[172]	PVDLI	SS	HBC	NC	CNN	polynomial	MNIST CIFAR10	98.1	2.5s	Y	Verifiability of the inference using SS
[173]	-	FHE (RNS CKSS)	HBC	ND	CNN	polynomial	CIFAR10	-	-	-	Polynomial activation functions for deep models

Table A.2: Overview of Secure Inference methods. The papers are sorted by year of publication, and by alphabetical order of the author.
Collusion: whether participants can collude with the server. Verification: whether the paper addresses verifiability of the data
Abbreviations: AHE= additional homomorphic encryption, BFV=Brakerski-Fan-Vercauteren scheme, CKKS= Cheon-Kim-Kim-Song scheme, CNN= convolutional neural network, DNN= deep neural network FHE= Full Homomorphic Encryption, GC= Garbled Circuits, GMW= Goldreich-Micali-Wigderson scheme, HBC = honest but curious, LSTM= Long short-term memory, LHE= Linearly Homomorphic Encryption, MA= malicious adversary, MLP= multi-layer perceptron, NC= No collusion, ND= not determined, OT= Oblivious transfer, SS= secret sharing

Ref.	Framework	Enc. Mechanism	Threat Level	No. servers	Collusion	Type of NN	Evaluation Datasets	Acc. on MNIST(%)	Efficiency	Fairness	Novelty
[183]	SecureML	LHE or OT	HBC	2	1 server + subset of clients	MLP	MNIST	93.1	High	-	introduces secure training with high computational cost
[184]	ABY3	OT GC aSS	MA	3	1 server + subset of clients	MLP	MNIST	94%	55000Œ to [183]	-	adaptation of ABY to three-server setup
[185]	CryptoDL	LHE	HBC	1	-	MLP CNN	MNIST a.o.	95	improves [183]	-	LHE for secure training, without re-encryption due to high cost
[186]	GeluNet	AHE (Paillier)	HBC	2	ND	MLP CNN	MNIST a.o.	96.9	0.281 s	-	AHE with the Paillier, leaving local calculations unencrypted
[187]	SecureNN	OT GC aSS	HBC MA	3	1 server + subset of clients	CNN DNN	MNIST	99	upto 93x to[183]	-	3-party set-up for secure training, that can also protect against a malicious adversary
[188]	Quotient	OT	HBC	2	ND	MLP CNN	MNIST and others	99.4	50x [183]	-	ST with ternarized neural networks
[189]	MSCryptonets	Multi-scheme FHE	HBC	1	-	CNN	MNIST Diabetes a.o.	99.6	*	-	Enables training phase using a multi-scheme FHE set-up
[190]	-	FHE (HElib)	HBC	1	-	CNN	MNIST	96	40 min per batch	-	Homomorphic encryption for training
[191]	Fan4	SS	MA	4	Honest Majority	CNN (ResNet a.o.)	MNIST ImageNet	98.6	20x to [192]	Y	Introduces robustness while retaining privacy.
[193]	Blaze	GC SS	MA	3	1 server + subset of clients	DNN	Parkinson[194]		278x to [184]	Y	Introduce a new secret-sharing semantics for 3PC tolerating up to one malicious corruption
[195]	FALCON	SS	MA	3	Honest Majority	MLP CNN	MNIST CIFAR10 T-ImageNet	98.6	6x to [187]	-	Batch normalization in a 3PC MA setting
[196]	Trident	GC SS ABY	MA	4	ND	MLP CNN	MNIST a.o.	98.3	187Œ to ABY3	Y	Fairness for 4PC MA scenario
[192]	SWIFT	ABY	MA	4	Honest Majority	MLP CNN	MNIST CIFAR10	*	*	Y	Robustness, guaranteed output delivery, for HBC
[197]	ARIANN	FSS	HBC	2	1 server + subset of clients	MLP CNN	MNIST CIFAR10 T-Imagenet	99.2	42 h	-	Offers ST suitable for GPU
[198]	Tetrad	ABY	MA	4	Honest Majority	MLP CNN	MNIST CIFAR10	ND	4x to [196]	Y	Robust ST for MA scenario

Table A.3: Summary of secure training papers. The papers are sorted by year of publication, and by alphabetical order of the author.
Efficiency: Speed up compared to benchmark, complexity, or seconds, in most ideal scenario. Collusion and Drop Out: max. number of participants. Fairness: whether the framework offers at least fairness.
Abbreviations: FSS = Function secret sharing GMW= Goldreich-Micali-Wigderson, LHE = Linear Homomorphic Encryption, LSTM= Long short-term memory, MA= malicious adversary, ND= not determined

Name	Privacy Level	PP Regime	PP Mechanism	Distributed Architecture	Type of NN	Evaluation Datasets	Highest Privacy Level
[204]	Global	(ϵ, δ)-DP	Laplace, Sparse Vector Technique	FL	MLP,CNN	MNIST, SVHN	$\epsilon = 1$ per parameter
[205]	Global	(ϵ, δ)-DP + MA	Gaussian	FL	N/D	MNIST	$\epsilon = 8$
[206]	Global, Local	(ϵ, δ)-DP	LDP [207]	FL	CNN, VGG	MNIST, FMNIST,CIFAR-10	$\epsilon = 5$
[208]	Global	f-DP	Gaussian	Private client models, shared noisy global model	CNN	MNIST, CIFAR-10	N/D
[209]	Global	(ϵ, δ)-DP + noisy gradients	Gaussian	FL	N/D	N/D	N/D
[210]	Local	(ϵ, δ)-DP	Gaussian	FL over wireless	N/D	N/D	N/D
[211]	Local	(ϵ, δ)-DP	Gaussian	FL cyclic	CNN	eICU, TCGA	$\epsilon = 3.84$
[212]	Local	(ϵ, δ)-DP	Laplace	FL	N/D	MNIST	$\epsilon = 1$
[213]	Local	(ϵ, δ)-DP	Gaussian	FL	SVM, DT, CNN	MNIST, UCI Nursery	$\epsilon = 0.5$
[214]	Local	(ϵ)-DP	Laplace	FL 3-tier	CNN	MNIST	N/D
[215]	Local	(ϵ, δ)-DP + MA	Gaussian with adaptive variance	FL + ADMM	Logistic Regression	Adult	$\epsilon = 1.01$
[216]	Local	(ϵ)-DP	Laplace, Randomized response	FL	N/D	MNIST	$\epsilon = 8$
[217]	Local	(ϵ, δ)-DP	Randomized response prior	FL	LDA	Amazon reviews, spam emails, sentiment	$\epsilon = 7.5$
[218]	Local	Renyi-DP + MA	Dirichlet	Ensemble	Teacher-student reinforcement learning	AirRaid-ram-v0 simulation from OpenAI Gym	$\epsilon = 4.46$
[219]	Local	(ϵ, δ)-DP	Gaussian	FL	CNN, LSTM	MNIST, EMNIST, CIFAR-19, Stackoverflow	$\epsilon = 19$
[220]	Local	(ϵ, δ)-DP + proposed accountant	Gaussian	FL	N/D	N/D	N/D
[221]	Local	(ϵ, δ)-DP + MA	Gaussian	FL	SqueezeNet	Diabetic retinopathy	$\epsilon = 11$

Table A.4: Overview of private distributed learning methods. The papers are sorted by year of publication, and by alphabetical order of the author.
Privacy level: reviewed works rarely share the same settings for different variables in these types of methods, e.g. model architecture, number of clients, rounds of training, etc. The reported values here are the most significant and/or relevant results from the most common dataset used, usually MNIST if available. The privacy levels should not be a basis of comparison between methods and each method should be evaluated individually
Abbreviations: CNN= Convolutional neural network, MA= Moments accountant, FL= Federated learning,

Ref.	Framework Name	Encryption method	Threat level	Collusion	Type of NN	Evaluation Datasets	Acc. on MNIST	Efficiency	Drop Out	DP	Novelty
[238]	SecAgg	SS	HBC	$n/3$-1 participants	-	-	-	$O(N^2)$	$n/3$-1	Gaussian	Introduces secure aggregation of models in FL
[231]	PPDL	AHE (LWE, Paillier)	HBC	No collusion	MLP CNN	MNIST SVHN [240] a.o.	99	2.40x loss to PT	None	Laplace	AHE-based scheme for cross silo FL
[212]	-	AHE (PPDM)	HBC	Multiple participants	CNN	MNIST	98.1	100x to [231]	Arbitrary subset	Laplace	Combines AHE and DP for efficient FL
[228]	-	AHE Re-encryption	HBC	Any participant	ASGD NN	MNIST	97	2.64x loss to [231]	None	-	Multikey set-up using re-encryption and ASGD
[233]	Hybrid-Alpha	FE	HBC	Maj. honest	CNN	MNIST	-	68% to [213]	Dep. on collusion	Gaussian	Functional encryption scheme resistant to dropout and addition of participants
[241]	-	SS	HBC SMA*	5%	-	-	-	-	5%	Shuffle [242]	Allows for secure aggregation with DP shuffling
[235]	SSDDL	SS	HBC	Some participants	CNN MLP	MNIST SVHN	99	$O(nN)$	ND	-	Effective secret sharing that can handle collusion
[243]	FastSecAgg	SS: Fast-Share	HBC	10%	CNN	-	-	-	10%	-	Secret sharing scheme based on Fast Fourier Transform
[244]	TurboAgg	SS	HBC	Maj. honest	CNN	CIFAR10	-	$O(NlogN)$	50%	-	Multi-group circular strategy for efficient model aggregation
[237]	PPFDL	GC Paillier	HBC	No collusion	CNN	MNIST	95**	Improves [238]	Arbitrary subset	-	Solution to reduce the negative impact of irregular users on accuracy
[245]	VerifyNet	SS HE hash	HBC	(t - 1) participants***	CNN	MNIST	99	High overhead	30% tested	-	Double-masking protocol to encrypt users local gradients and provide verification
[234]	BatchCrypt	AHE (Paillier)	HBC	No collusion	MLP CNN LTSM	FMNIST-SSP [246] CIFAR10	-	93x to FATE [247]	None	-	Uses batch-encrypting for more efficient cross silo FL
[239]	Poseidon	FHE (CKKS)	HBC	(n - 1) participants	MLP CNN	MNIST CIFAR100 o.a.	89.9	5283.1s	None	-	Multiparty HE for unlimited amount of participants
[236]	DAEQ-FL	ElGamal SS	HBC	Maj.honest	CNN LSTM	MNIST CIFAR10 SSP	99	****	t***	-	FL with no trusted third party and non-i.i.d. data
[248]	-	SS	HBC	(t - 1) participants	-	-	-	-	ND	-	Extension of [245] for non fully trusted centers
[230]	xMK-CKKS	MK-CKKS AHE (Paillier)	HBC	(n - 1) participants	CNN	UP-FALL [249]	-	****	ND	-	Multi-key set up efficient, requiring all parties for decryption

Table A.5: Encryption-based protective measures for FL, sorted by year and author.
Efficiency in speedup compared to benchmark, complexity or seconds. Collusion and Drop Out quantified in max. number of participants
Abbreviations: DM = double masking, ND= not determined, SSP = Shakespeare, PT= plaintext
* SMA = semi malicious ** with irregular users *** in (t-1), t refers to t-out-of-n SS **** papers show improvement compared to a similar Paillier based protocol, but do not compare to a benchmark

APPENDIX B

B.1 Basic Concepts

B.1.1 Probability Distributions and Their Properties

A random variable X is characterized by its probability distribution. For a continuous random variable, this is often described by a probability density function (PDF), denoted $_X(x)$, such$_t hat(\text{a} \leq X \leq b) = \int_a^b {}_X(x)dx$. The corresponding cumulative distribution function (CDF) is $_X(x) = (X \leq x) = \int_{-\infty}^x {}_X(t)dt$.

The behavior of a distribution in its extremities, known as its *tail behavior*, is of particular interest. We distinguish several types of tail behavior: A distribution is considered *heavy-tailed* if its tails decay slower than any exponential function. More formally, for a random variable X with PDF $_X(x)$, if its moment generating function (MGF), $_X(t) = [e^{tX}]$, is infinite for all $t > 0$, or for some $t_0 > 0$ it is infinite for $t \geq t_0$, the distribution is heavy-tailed. Examples include the Pareto and Student's t-distributions. Such distributions assign higher probabilities to extreme events compared to light-tailed distributions. Conversely, a distribution is *light-tailed* if its tails decay at least as fast as an exponential function. The MGF for a light-tailed distribution exists in a neighborhood around zero. A special case is *normal-tailed* behavior, characteristic of the Gaussian distribution, where tails decay very rapidly, proportional to e^{-x^2}.

The heaviness of a distribution's tail can be quantified by its *tail index*, often denoted α. For distributions exhibiting power-law decay in their tails, the survival function

$\bar{}_X(x) = 1 - {}_X(x) = (X > x)$ behaves as $x^{-\alpha}$ for large x. A smaller α indicates a heavier tail.

In many real-world scenarios, datasets comprise variables with differing tail characteristics. We refer to such data as exhibiting *mixed-tail behavior*, where some marginal distributions might be heavy-tailed, while others are light-tailed or normal-tailed.

Generalized Gaussian Distribution (GGD)

The Generalized Gaussian Distribution (GGD) is a family of continuous probability distributions that extends the Gaussian distribution by introducing a shape parameter, allowing for a wider range of tail behaviors. The probability density function (PDF) of a GGD with mean $\mu \in \mathbb{R}$, scale parameter $\alpha > 0$, and shape parameter $\beta > 0$ is given by:

$$_{\text{GGD}}(x; \mu, \alpha, \beta) = \frac{\beta}{2\alpha\Gamma(1/\beta)} \exp\left(-\left(\frac{|x-\mu|}{\alpha}\right)^{\beta}\right), \tag{B.1}$$

where $\Gamma(\cdot)$ is the gamma function. The shape parameter β dictates the distribution's kurtosis and tail characteristics:

- For $\beta = 2$, the GGD reduces to a Gaussian (normal) distribution with variance $\sigma^2 = \alpha^2$.
- For $\beta = 1$, it becomes a Laplace distribution, which has heavier tails than a Gaussian.
- For $0 < \beta < 2$, the distribution exhibits heavier tails than the Gaussian (leptokurtic). As $\beta \to 0^+$, the distribution becomes increasingly concentrated around the mean with extremely heavy tails.
- For $\beta > 2$, the distribution has lighter tails than the Gaussian (platykurtic) and becomes flatter as β increases.

This flexibility makes the GGD a suitable candidate for modeling data features with diverse distributional shapes, especially when used as a component in mixture models or as a base distribution in Normalizing Flows.

Mixture Models

A mixture model is a probabilistic model that represents a distribution as a convex combination (a weighted sum) of several simpler component probability distributions. If $p_k(\mathbf{x}; \boldsymbol{\phi}_k)$ denotes the PDF of the k-th component distribution with parameters $\boldsymbol{\phi}_k$, and π_k is the mixing proportion (weight) of the k-th component, then the PDF of a K-component mixture model is:

$$p_{\text{mix}}(\mathbf{x}; \{\pi_k, \boldsymbol{\phi}_k\}_{k=1}^K) = \sum_{k=1}^{K} \pi_k p_k(\mathbf{x}; \boldsymbol{\phi}_k), \tag{B.2}$$

where $\pi_k \geq 0$ for all k, and $\sum_{k=1}^K \pi_k = 1$. Mixture models are highly flexible and can approximate a wide variety of complex, multimodal, or non-standard distributions that cannot be adequately captured by a single parametric distribution.

Mixture of Generalized Gaussian Distributions (MoGGD): A particularly relevant instance for modeling varied tail behaviors is a Mixture of GGDs. In this case, each component $p_k(\mathbf{x}; \boldsymbol{\phi}_k)$ in Equation B.2 is a GGD as defined in Equation B.1, with its own set of parameters $\boldsymbol{\phi}_k = \{\mu_k, \alpha_k, \beta_k\}$. Thus, an MoGGD can be written as:

$$p_{\text{MoGGD}}(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \text{GGD}(x; \mu_k, \alpha_k, \beta_k). \tag{B.3}$$

By allowing each component GGD to have a different shape parameter β_k, an MoGGD can effectively model distributions with complex and varying tail characteristics across different regions of the data space. The parameters $\{\pi_k, \mu_k, \alpha_k, \beta_k\}_{k=1}^K$ are typically learned during model training.

B.1.2 Geometric Median

In multivariate statistics, the concept of a central tendency extends beyond the arithmetic mean. The Geometric Median (GM) of a set of points $\{x_1, x_2, \ldots, x_N\}$ in ^{d} is defined as the point μ_{GM} that minimizes the sum of Euclidean distances to these points:

$$\mu_{\text{GM}} = \arg\min_{\mu \in ^d} \sum_{i=1}^{N} \|x_i - \mu\|_2. \tag{B.4}$$

Unlike the arithmetic mean, the geometric median is robust to outliers, as its location is less influenced by extreme values in the dataset. This robustness is particularly valuable when dealing with heavy-tailed or contaminated data distributions.

B.1.3 Differential Privacy (DP)

Differential Privacy (DP) is a formal, mathematical framework for quantifying and limiting the privacy leakage of algorithms operating on sensitive datasets. It provides a strong guarantee that the output of a randomized mechanism does not significantly change if any single individual's data is altered, added, or removed from the input dataset, thereby offering plausible deniability to individuals contributing their data.

Core Definition: ϵ-Differential Privacy

Let $\mathcal{D}$ and $\mathcal{D}'$ be two datasets, referred to as *neighboring datasets*, if they differ in at most one individual's record (e.g., $\mathcal{D}'$ can be obtained from $\mathcal{D}$ by adding or removing one record, or by changing one record). A randomized mechanism (algorithm) $\mathcal{M}$ mapping datasets from a domain $\mathbb{D}$ to a range $\mathcal{R}$ satisfies ϵ-*Differential Privacy* (or pure DP) if for all neighboring datasets $\mathcal{D}, \mathcal{D}' \in \mathbb{D}$ and for all subsets of possible outputs $S \subseteq \mathcal{R}$:

$$\Pr(\mathcal{M}(\mathcal{D}) \in S) \leq e^{\epsilon} \cdot \Pr(\mathcal{M}(\mathcal{D}') \in S). \tag{B.5}$$

The parameter $\epsilon \geq 0$ is the *privacy budget* (or privacy loss parameter). A smaller ϵ implies a stronger privacy guarantee, as it bounds more tightly the ratio of probabilities of observing any output S whether or not a particular individual's data is included. When $\epsilon = 0$, the output of the mechanism is entirely independent of any single individual's data.

Likelihood Ratio Formulation for Generative Models

In the context of training a generative model $p_{\text{model}}(\cdot; \boldsymbol{\theta})$ whose parameters $\boldsymbol{\theta}$ are learned from a dataset $\mathcal{D}$, DP can be related to the likelihood of the learned parameters. If the

training process itself is considered a randomized mechanism $\mathcal{M}$ that outputs parameters $\boldsymbol{\theta}$, then ϵ-DP implies that for any neighboring datasets $\mathcal{D}, \mathcal{D}'$:

$$\frac{p(\boldsymbol{\theta}|\mathcal{D})}{p(\boldsymbol{\theta}|\mathcal{D}')} \leq e^{\epsilon}, \tag{B.6}$$

for all possible parameter sets $\boldsymbol{\theta}$. For Normalizing Flows (NFs, Section B.2.2) trained via MLE, where the model directly defines a density $p_{\text{model}}(\mathbf{x}; \boldsymbol{\theta})$, Chapter 4 explores achieving DP by constraining the model's expressivity. The core idea relies on bounding the likelihood ratio of the *data itself* under the model, particularly for the single differing record between $\mathcal{D}$ and $\mathcal{D}'$. If $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ and $\mathcal{D}' = \{\mathbf{x}_1, \ldots, \mathbf{x}_{N-1}, \mathbf{x}'_N\}$, then the ratio of product likelihoods for the datasets under a fixed model $\boldsymbol{\theta}$ becomes:

$$\frac{\prod_{i=1}^{N} p_{\text{model}}(\mathbf{x}_i; \boldsymbol{\theta})}{\prod_{i=1}^{N-1} p_{\text{model}}(\mathbf{x}_i; \boldsymbol{\theta}) \cdot p_{\text{model}}(\mathbf{x}'_N; \boldsymbol{\theta})} = \frac{p_{\text{model}}(\mathbf{x}_N; \boldsymbol{\theta})}{p_{\text{model}}(\mathbf{x}'_N; \boldsymbol{\theta})}. \tag{B.7}$$

Thus, a mechanism that ensures $\frac{p_{\text{model}}(\mathbf{x}_N; \boldsymbol{\theta})}{p_{\text{model}}(\mathbf{x}'_N; \boldsymbol{\theta})} \leq e^{\epsilon}$ for any two possible records $\mathbf{x}_N, \mathbf{x}'_N$ (representing the change between neighboring datasets) for a chosen $\boldsymbol{\theta}$ can satisfy DP. Chapter 4 details how properties of p_{model} (e.g., base density, Jacobian determinant of transformations) are constrained to achieve this bound.

Privacy Accounting and Composition

When an algorithm involves multiple accesses to sensitive data or combines several differentially private mechanisms, it is crucial to account for the cumulative privacy loss. This is managed through composition theorems and privacy amplification techniques.

Sequential (Basic) Composition. If a sequence of k mechanisms, $\mathcal{M}_1, \ldots, \mathcal{M}_k$, are applied, where each mechanism $\mathcal{M}_i$ is ϵ_i-differentially private (and independently accesses the data or operates on the output of the previous stage in a way that maintains DP for its step), then the overall sequence of mechanisms $\mathcal{M}_{\text{seq}} = (\mathcal{M}_1, \ldots, \mathcal{M}_k)$ is $(\sum_{i=1}^{k} \epsilon_i)$-differentially private. Formally, if $\mathcal{M}_i : \mathbb{D} \to \mathcal{R}_i$ is ϵ_i-DP, then the mechanism $\mathcal{M}_{\text{seq}}(\mathcal{D}) = (\mathcal{M}_1(\mathcal{D}), \ldots, \mathcal{M}_k(\mathcal{D}))$ is $\left(\sum_{i=1}^{k} \epsilon_i\right)$-DP. This provides a simple but often loose bound on the total privacy loss.

Advanced Composition Theorem. For iterative algorithms common in machine learning, more sophisticated composition theorems provide tighter bounds. The advanced composition theorem states that for any $\delta' > 0$, the k-fold adaptive composition of (ϵ, δ)-differentially private mechanisms (where each mechanism can depend on prior outputs) is $(\epsilon', k\delta + \delta')$-differentially private, where

$$\epsilon' = \sqrt{2k \log(1/\delta')}\epsilon + k\epsilon(e^{\epsilon} - 1). \tag{B.8}$$

A common simplification for small ϵ yields $\epsilon' \approx \sqrt{2k \log(1/\delta')}\epsilon + k\epsilon^2$. This theorem is particularly useful when each step of an iterative algorithm (e.g., each gradient update in DP-SGD) is made (ϵ, δ)-DP. For pure ϵ-DP mechanisms (i.e., $\delta = 0$), this theorem can still be applied, yielding an overall (ϵ', δ')-DP guarantee where ϵ' is calculated as above.

Privacy Amplification by Subsampling. Differential privacy guarantees can be amplified (i.e., the effective privacy loss can be reduced) if each mechanism operates on a random subsample of the original dataset rather than the entire dataset.

Let $\mathcal{M}$ be an ϵ-differentially private mechanism.

- **Subsampling without replacement:** If $\mathcal{M}$ is applied to a random subsample of size m drawn without replacement from a dataset $\mathcal{D}$ of size N (where $m < N$), and the sampling ratio is $q = m/N$, then the resulting mechanism $\mathcal{M}_{\text{sub}}$ is approximately $(q\epsilon, q\delta)$-DP if $\mathcal{M}$ was (ϵ, δ)-DP, for small $q\epsilon$. More precisely, for pure ϵ-DP mechanisms, subsampling without replacement from a dataset of size N to a sample of size m transforms an ϵ-DP mechanism into an (ϵ', δ')-DP mechanism where $\epsilon' = \log(1 + q(e^{\epsilon} - 1))$.
- **Poisson Subsampling (or sampling with replacement):** If each record from a dataset $\mathcal{D}$ of size N is included in a subsample $\mathcal{D}_{\text{sub}}$ independently with probability q (thus, the expected size of $\mathcal{D}_{\text{sub}}$ is qN), and an ϵ-differentially private mechanism $\mathcal{M}$ is applied to $\mathcal{D}_{\text{sub}}$, the resulting mechanism $\mathcal{M}_{\text{Pois}}$ is $(\log(1+q(e^{\epsilon}-1)), q\delta)$-differentially private if $\mathcal{M}$ was (ϵ, δ)-DP. For pure ϵ-DP mechanisms ($\delta = 0$), this becomes $(\log(1 + q(e^{\epsilon} - 1)))$-DP. For small q and ϵ, this is often approximated as

$(q\epsilon)$-DP.

This amplification is critical for achieving strong privacy guarantees in iterative training processes over large datasets, where each iteration (e.g., a mini-batch update) uses a small subsample of the data. The effective per-iteration ϵ is reduced by the sampling ratio q, which then leads to a much smaller total ϵ when composed over many iterations using the Advanced Composition Theorem.

B.2 Generative Models

Generative models aim to learn the underlying probability distribution, $p_{\text{data}}(\mathbf{x})$, of a given dataset $\mathcal{D} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}\}$, where each $\mathbf{x}^{(i)} \in \mathcal{X}$ is a data sample from some (often high-dimensional) space $\mathcal{X}$. The core objective is to construct a model, $p_{\text{model}}(\mathbf{x}; \boldsymbol{\theta})$ parameterized by $\boldsymbol{\theta}$, that approximates $p_{\text{data}}(\mathbf{x})$.

Once trained, such a model can be used for several purposes. Primarily, it allows for the generation of new, *synthetic samples* $\tilde{\mathbf{x}}$ such that $\tilde{\mathbf{x}} \sim p_{\text{model}}(\mathbf{x}; \boldsymbol{\theta})$. Ideally, these synthetic samples should be indistinguishable from real samples drawn from $p_{\text{data}}(\mathbf{x})$ according to relevant statistical metrics. This capability is valuable for tasks including data augmentation (to enlarge training sets), imputation of missing data, and understanding the underlying structure of the data.

Furthermore, in contexts requiring privacy, generative models can be trained under privacy-preserving constraints (e.g., Differential Privacy) to produce synthetic datasets that retain statistical utility while protecting sensitive information about individual records in $\mathcal{D}$. Various classes of generative models exist, including Variational Autoencoders (VAEs, Section B.2.3), Generative Adversarial Networks (GANs, Section B.2.4), and Normalizing Flows (NFs, Section B.2.2), each with distinct architectural and training paradigms. The choice of model often depends on the nature of the data and the specific application requirements, such as desired sample quality, diversity, and computational tractability.

The evaluation of generative models typically involves assessing both the *fidelity* (how well p_{model} matches p_{data}) and *diversity* (the variety of samples p_{model} can produce) of

the generated samples.

B.2.1 Maximum Likelihood Estimation

A cornerstone of parametric statistical inference is Maximum Likelihood Estimation (MLE). Given a dataset $= \{x_1, x_2, \ldots, x_N\}$ of N independent and identically distributed (i.i.d.) samples drawn from a distribution $_X(x;\theta)$ parameterized by a vector θ, the likelihood function is defined as $(\theta;) = \prod_{i=1}^{N} {}_X(x_i;\theta)$. The MLE estimate of θ, denoted $\hat{\theta}_{\text{MLE}}$, is the value that maximizes this likelihood function:

$$\hat{\theta}_{\text{MLE}} = \arg\max_{\theta}(\theta;) = \arg\max_{\theta} \sum_{i=1}^{N} \log {}_X(x_i;\theta). \tag{B.9}$$

MLE provides a principled approach for fitting models to data and possesses desirable asymptotic properties under certain regularity conditions.

B.2.2 Normalizing Flows (NFs)

Normalizing Flows (NFs) represent a class of generative models that explicitly learn the data distribution $p_{\text{model}}(\mathbf{x};\boldsymbol{\theta})$ by transforming a simple base (or latent) probability distribution $p_Z(\mathbf{z})$ into a more complex target distribution that approximates $p_{\text{data}}(\mathbf{x})$. Here, $\mathbf{z} \in \mathcal{Z}$ is a latent variable in a latent space $\mathcal{Z}$ (often $\mathcal{Z} = \mathcal{X} = \mathbb{R}^D$), and $\mathbf{x} \in \mathcal{X}$ is a data variable.

The core of an NF is an invertible and differentiable transformation (a diffeomorphism) $f_{\boldsymbol{\theta}} : \mathcal{Z} \to \mathcal{X}$, such that $\mathbf{x} = f_{\boldsymbol{\theta}}(\mathbf{z})$. The inverse transformation is $f_{\boldsymbol{\theta}}^{-1} : \mathcal{X} \to \mathcal{Z}$, so $\mathbf{z} = f_{\boldsymbol{\theta}}^{-1}(\mathbf{x})$. The probability density of $\mathbf{x}$ can be computed exactly using the change of variables formula:

$$p_{\text{model}}(\mathbf{x};\boldsymbol{\theta}) = p_Z(f_{\boldsymbol{\theta}}^{-1}(\mathbf{x})) \left| \det \left(\frac{\partial f_{\boldsymbol{\theta}}^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right) \right|, \tag{B.10}$$

or equivalently, in log-space:

$$\log p_{\text{model}}(\mathbf{x};\boldsymbol{\theta}) = \log p_Z(f_{\boldsymbol{\theta}}^{-1}(\mathbf{x})) + \log \left| \det \left(\frac{\partial f_{\boldsymbol{\theta}}^{-1}(\mathbf{x})}{\partial \mathbf{x}} \right) \right|. \tag{B.11}$$

The term $\frac{\partial f_{\boldsymbol{\theta}}^{-1}(\mathbf{x})}{\partial \mathbf{x}}$ is the Jacobian matrix of the inverse transformation $f_{\boldsymbol{\theta}}^{-1}$ with respect to $\mathbf{x}$, and its determinant quantifies the local change in volume induced by the transformation. The base distribution $p_Z(\mathbf{z})$ is typically chosen to be simple, such as an isotropic Gaussian $\mathcal{N}(\mathbf{z}|\mathbf{0}, \mathbf{I})$, for which $\log p_Z(\mathbf{z})$ is easy to compute.

Training NFs is commonly performed via Maximum Likelihood Estimation by maximizing the sum of log-likelihoods over the training dataset $\mathcal{D}$:

$$\boldsymbol{\theta}^* = \arg\max_{\boldsymbol{\theta}} \sum_{\mathbf{x}^{(i)} \in \mathcal{D}} \log p_{\text{model}}(\mathbf{x}^{(i)}; \boldsymbol{\theta}).$$

Sampling from a trained NF involves drawing a sample $\mathbf{z} \sim p_Z(\mathbf{z})$ and then computing $\tilde{\mathbf{x}} = f_{\boldsymbol{\theta}}(\mathbf{z})$.

A key characteristic of NFs is that the tail behavior of the generated data $p_{\text{model}}(\mathbf{x}; \boldsymbol{\theta})$ is directly influenced by both the tail behavior of the chosen base distribution $p_Z(\mathbf{z})$ and the properties of the transformation $f_{\boldsymbol{\theta}}$. NFs can be categorized based on the properties of their Jacobian determinant. *Volume preserving flows* have a constant Jacobian determinant (often unity), simplifying Equation B.10. More expressive *non-volume preserving flows* allow the Jacobian determinant to depend on the input, offering greater modeling flexibility. The Lipschitz continuity of $f_{\boldsymbol{\theta}}$ and its inverse $f_{\boldsymbol{\theta}}^{-1}$ is also a critical property, particularly for ensuring stable training and for applications in differentially private settings.

B.2.3 Variational Autoencoders (VAEs)

Variational Autoencoders (VAEs) are a class of generative models that learn a probabilistic mapping from a high-dimensional data space $\mathcal{X}$ to a lower-dimensional continuous latent space $\mathcal{Z}'$, and vice versa. VAEs are trained by optimizing a lower bound on the data log-likelihood.

A VAE consists of two main neural network components:

1. **Encoder (or Recognition Model):** Parameterized by $\boldsymbol{\phi}$, the encoder $q_{\boldsymbol{\phi}}(\mathbf{z}'|\mathbf{x})$ approximates the true but intractable posterior distribution $p(\mathbf{z}'|\mathbf{x})$ of the latent

variable $\mathbf{z}'$ given a data sample $\mathbf{x}$. Typically, $q_{\boldsymbol{\phi}}(\mathbf{z}'|\mathbf{x})$ is chosen to be a simple parametric distribution, such as a multivariate Gaussian with a diagonal covariance matrix: $q_{\boldsymbol{\phi}}(\mathbf{z}'|\mathbf{x}) = \mathcal{N}(\mathbf{z}'|\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x}), \text{diag}(\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x})))$, where $\boldsymbol{\mu}_{\boldsymbol{\phi}}(\mathbf{x})$ and $\boldsymbol{\sigma}^2_{\boldsymbol{\phi}}(\mathbf{x})$ are outputs of the encoder network.

2. **Decoder (or Generative Model/Likelihood Model):** Parameterized by $\boldsymbol{\psi}$, the decoder $p_{\boldsymbol{\psi}}(\mathbf{x}|\mathbf{z}')$ defines a distribution over the data space conditioned on the latent variable $\mathbf{z}'$. For continuous data, $p_{\boldsymbol{\psi}}(\mathbf{x}|\mathbf{z}')$ might be a Gaussian $\mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_{\boldsymbol{\psi}}(\mathbf{z}'), \text{diag}(\boldsymbol{\sigma}^2_{\boldsymbol{\psi}}(\mathbf{z}')))$, and for binary data, a Bernoulli distribution.

A prior distribution $p_{\boldsymbol{\eta}}(\mathbf{z}')$ is imposed over the latent variables, typically a standard isotropic Gaussian $\mathcal{N}(\mathbf{z}'|\mathbf{0}, \mathbf{I})$. The parameters $\boldsymbol{\eta}$ of this prior might be fixed or learned.

The VAE is trained by maximizing the Evidence Lower Bound (ELBO), $\mathcal{L}_{\text{ELBO}}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\eta}; \mathbf{x})$, for each data point $\mathbf{x}$:

$$\log p(\mathbf{x}) \geq \mathcal{L}_{\text{ELBO}}(\boldsymbol{\phi}, \boldsymbol{\psi}, \boldsymbol{\eta}; \mathbf{x}) = \mathbb{E}_{\mathbf{z}' \sim q_{\boldsymbol{\phi}}(\mathbf{z}'|\mathbf{x})} [\log p_{\boldsymbol{\psi}}(\mathbf{x}|\mathbf{z}')] - D_{\text{KL}}(q_{\boldsymbol{\phi}}(\mathbf{z}'|\mathbf{x}) || p_{\boldsymbol{\eta}}(\mathbf{z}')). \tag{B.12}$$

The first term is the expected negative reconstruction error, encouraging the decoder to accurately reconstruct $\mathbf{x}$ from latent codes $\mathbf{z}'$ sampled from the approximate posterior $q_{\boldsymbol{\phi}}(\mathbf{z}'|\mathbf{x})$. The second term is the Kullback-Leibler (KL) divergence between the approximate posterior and the latent prior. This term acts as a regularizer, encouraging the learned latent representations to conform to the prior distribution $p_{\boldsymbol{\eta}}(\mathbf{z}')$. The expectation is typically approximated using Monte Carlo sampling and the reparameterization trick for differentiability.

To generate new samples from a trained VAE, one samples $\mathbf{z}' \sim p_{\boldsymbol{\eta}}(\mathbf{z}')$ and then samples $\tilde{\mathbf{x}} \sim p_{\boldsymbol{\psi}}(\mathbf{x}|\mathbf{z}')$.

In the context of Chapter 3, VAEs (or Conditional VAEs, CVAEs, where $q_{\boldsymbol{\phi}}(\mathbf{z}'|\mathbf{x}, \mathbf{y})$, $p_{\boldsymbol{\psi}}(\mathbf{x}|\mathbf{z}', \mathbf{y})$, and $p_{\boldsymbol{\eta}}(\mathbf{z}'|\mathbf{y})$ are conditioned on auxiliary information $\mathbf{y}$) are employed as the *Validator* component. The learned prior $p_{\boldsymbol{\eta}}(\mathbf{z}'|\mathbf{y})$ (potentially an expressive, tail-adaptive Normalizing Flow prior) models the semantic structure of the data in the latent space, and the KL divergence term (or a related penalty derived from $p_{\boldsymbol{\eta}}(\mathbf{z}'|\mathbf{y})$) provides the semantic regularization signal.

B.2.4 Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) constitute a framework for training generative models through an adversarial process involving two neural networks: a *Generator* (G) and a *Discriminator* (D). The Generator $G(\mathbf{z}; \boldsymbol{\theta}_G)$, parameterized by $\boldsymbol{\theta}_G$, maps samples $\mathbf{z}$ from a simple prior distribution $p_Z(\mathbf{z})$ (e.g., Gaussian noise) to the data space $\mathcal{X}$, attempting to produce samples $\tilde{\mathbf{x}} = G(\mathbf{z}; \boldsymbol{\theta}_G)$ that resemble real data $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$. The Discriminator $D(\mathbf{x}; \boldsymbol{\theta}_D)$, parameterized by $\boldsymbol{\theta}_D$, is a binary classifier trained to distinguish between real samples from $p_{\text{data}}(\mathbf{x})$ and synthetic samples $\tilde{\mathbf{x}}$ produced by G.

The training process involves a minimax game where G and D are trained iteratively. The objective for the original GAN formulation is:

$$\min_{\boldsymbol{\theta}_G} \max_{\boldsymbol{\theta}_D} V(D, G) = \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[\log D(\mathbf{x}; \boldsymbol{\theta}_D)] + \mathbb{E}_{\mathbf{z} \sim p_Z(\mathbf{z})}[\log(1 - D(G(\mathbf{z}; \boldsymbol{\theta}_G); \boldsymbol{\theta}_D))]. \tag{B.13}$$

Here, D aims to maximize the objective (correctly classifying real and fake samples), while G aims to minimize it by generating samples that D misclassifies as real (i.e., maximizing $D(G(\mathbf{z}))$).

Wasserstein GAN with Gradient Penalty (WGAN-GP)

Training standard GANs can be unstable. Wasserstein GANs (WGANs) address this by using the Wasserstein-1 distance (Earth Mover's distance) to measure the discrepancy between p_{data} and the distribution p_G implicitly defined by G. The discriminator (often called a "critic" in WGANs) $D(\mathbf{x}; \boldsymbol{\theta}_D)$ is trained to approximate the Wasserstein distance, and its output is a scalar score rather than a probability. The WGAN objective for the critic is:

$$\max_{\boldsymbol{\theta}_D} \left(\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[D(\mathbf{x}; \boldsymbol{\theta}_D)] - \mathbb{E}_{\tilde{\mathbf{x}} \sim p_G(\tilde{\mathbf{x}})}[D(\tilde{\mathbf{x}}; \boldsymbol{\theta}_D)] \right),$$

subject to D being 1-Lipschitz. The generator's objective is:

$$\min_{\boldsymbol{\theta}_G} \left(-\mathbb{E}_{\tilde{\mathbf{x}} \sim p_G(\tilde{\mathbf{x}})}[D(\tilde{\mathbf{x}}; \boldsymbol{\theta}_D)] \right).$$

WGAN with Gradient Penalty (WGAN-GP) enforces the Lipschitz constraint by adding a penalty term to the critic's loss:

$$\mathcal{L}_{\text{critic}} = \mathbb{E}_{\tilde{\mathbf{x}}\sim p_G(\tilde{\mathbf{x}})}[D(\tilde{\mathbf{x}})] - \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}[D(\mathbf{x})] + \lambda_{\text{GP}}\mathbb{E}_{\hat{\mathbf{x}}\sim p_{\hat{\mathbf{x}}}}[(\|\nabla_{\hat{\mathbf{x}}}D(\hat{\mathbf{x}})\|_2 - 1)^2], \quad \text{(B.14)}$$

where $\hat{\mathbf{x}}$ are samples along straight lines between pairs of points sampled from p_{data} and p_G, and λ_{GP} is a penalty coefficient.

Conditional GANs and Augmentations

GANs can be conditioned on auxiliary information $\mathbf{y}$ (e.g., class labels), denoted $G(\mathbf{z}, \mathbf{y}; \boldsymbol{\theta}_G)$ and $D(\mathbf{x}, \mathbf{y}; \boldsymbol{\theta}_D)$. In Chapter 3, GANs (specifically WGAN-GP) are employed as the *Synthesizer* component. The objective is often augmented, for instance, with an *Auxiliary Classifier* (AC) loss $\mathcal{L}_{G,\text{AC}}(\boldsymbol{\theta}_G)$ (from AC-GANs) to ensure generated samples $\tilde{\mathbf{x}}$ are consistent with their conditioning $\mathbf{y}$, and an *Information Loss* term $\mathcal{L}_I(\boldsymbol{\theta}_G)$ to promote feature-level similarity between real and generated data batches. The base generator objective for the synthesizer $G_{\boldsymbol{\theta}}$ (using G as shorthand for $G_{\boldsymbol{\theta}_G}$) in such a setup might be:

$$\mathcal{L}_{G,\text{base}}(\boldsymbol{\theta}_G) = \mathcal{L}_{G,\text{WGAN}}(\boldsymbol{\theta}_G) + \lambda_{\text{AC}}\mathcal{L}_{G,\text{AC}}(\boldsymbol{\theta}_G) + \lambda_I\mathcal{L}_I(\boldsymbol{\theta}_G), \quad \text{(B.15)}$$

where $\mathcal{L}_{G,\text{WGAN}}$ is the WGAN generator loss, and λ_{AC}, λ_I are hyperparameters. This base objective is then further regularized by the semantic penalty from the Validator in the LSR framework.

BIBLIOGRAPHY

[1] Mike Laszkiewicz, Johannes Lederer, and Asja Fischer. Marginal tail-adaptive normalizing flows. In *International Conference on Machine Learning*, pages 12020–12048. PMLR, 2022.

[2] Chris Waites and Rachel Cummings. Differentially private normalizing flows for privacy-preserving density estimation. In *Proceedings of the 2021 AAAI/ACM Conference on AI, Ethics, and Society*, pages 1000–1009, 2021.

[3] Ibrar Yaqoob, Ibrahim Abaker Targio Hashem, Abdullah Gani, Salimah Mokhtar, Ejaz Ahmed, Nor Badrul Anuar, and Athanasios V Vasilakos. Big data: From beginning to future. *International Journal of Information Management*, 36(6): 1231–1247, 2016.

[4] Eric P Xing, Qirong Ho, Pengtao Xie, and Dai Wei. Strategies and principles of distributed machine learning on big data. *Engineering*, 2(2):179–195, 2016.

[5] Shlomi Dolev, Patricia Florissi, Ehud Gudes, Shantanu Sharma, and Ido Singer. A survey on geographically distributed big-data processing using mapreduce. *IEEE Transactions on Big Data*, 5(1):60–80, 2017.

[6] Russell Clapp, Martin Dimitrov, Karthik Kumar, Vish Viswanathan, and Thomas Willhalm. Quantifying the performance impact of memory latency and bandwidth for big data workloads. In *2015 IEEE International Symposium on Workload Characterization*, pages 213–224. IEEE, 2015.

[7] Nader Mohamed and Jameela Al-Jaroodi. Real-time big data analytics: Applications and challenges. In *2014 international conference on high performance computing & simulation (HPCS)*, pages 305–310. IEEE, 2014.

[8] Yaser Mansouri, Adel Nadjaran Toosi, and Rajkumar Buyya. Data storage management in cloud environments: Taxonomy, survey, and future directions. *ACM Computing Surveys (CSUR)*, 50(6):1–51, 2017.

[9] Paul Voigt and Axel Von dem Bussche. The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10 (3152676):10–5555, 2017.

[10] Nils Gruschka, Vasileios Mavroeidis, Kamer Vishi, and Meiko Jensen. Privacy issues and data protection in big data: a case study analysis under gdpr. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 5027–5033. IEEE, 2018.

[11] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

[12] Preeti Gupta, Arun Sharma, and Rajni Jindal. Scalable machine-learning algorithms for big data analytics: a comprehensive review. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 6(6):194–214, 2016.

[13] Xuefei Yin, Yanming Zhu, and Jiankun Hu. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Computing Surveys (CSUR)*, 54(6):1–36, 2021.

[14] Ji Liu, Jizhou Huang, Yang Zhou, Xuhong Li, Shilei Ji, Haoyi Xiong, and Dejing Dou. From distributed machine learning to federated learning: A survey. *Knowledge and Information Systems*, 64(4):885–917, 2022.

[15] Avishek Ghosh, Justin Hong, Dong Yin, and Kannan Ramchandran. Robust federated learning in a heterogeneous environment. *arXiv preprint arXiv:1906.06629*, 2019.

[16] Fuxun Yu, Weishan Zhang, Zhuwei Qin, Zirui Xu, Di Wang, Chenchen Liu, Zhi Tian, and Xiang Chen. Heterogeneous federated learning. *arXiv preprint arXiv:2008.06767*, 2020.

[17] Zhuangdi Zhu, Junyuan Hong, and Jiayu Zhou. Data-free knowledge distillation for heterogeneous federated learning. In *International conference on machine learning*, pages 12878–12889. PMLR, 2021.

[18] Mu-Hsing Kuo, Tony Sahama, Andre W Kushniruk, Elizabeth M Borycki, and Daniel K Grunwell. Health big data analytics: current perspectives, challenges and potential solutions. *International Journal of Big Data Intelligence*, 1(1-2): 114–126, 2014.

[19] Choong Ho Lee and Hyung-Jin Yoon. Medical big data: promise and challenges. *Kidney research and clinical practice*, 36(1):3, 2017.

[20] Blagoj Ristevski and Ming Chen. Big data analytics in medicine and healthcare. *Journal of integrative bioinformatics*, 15(3):20170030, 2018.

[21] Tianze Sun, Xiwang He, and Zhonghai Li. Digital twin in healthcare: Recent updates and challenges. *Digital Health*, 9:20552076221149651, 2023.

[22] Maulshree Singh, Evert Fuenmayor, Eoin P Hinchy, Yuansong Qiao, Niall Murray, and Declan Devine. Digital twin: Origin to future. *Applied System Innovation*, 4 (2):36, 2021.

[23] Tolga Erol, Arif Furkan Mendi, and Dilara Doğan. The digital twin revolution in healthcare. In *2020 4th international symposium on multidisciplinary studies and innovative technologies (ISMSIT)*, pages 1–7. IEEE, 2020.

[24] Alexandre Vallée. Digital twin for healthcare systems. *Frontiers in Digital Health*, 5:1253050, 2023.

[25] Frank Emmert-Streib and Olli Yli-Harja. What is a digital twin? experimental design for a data-centric machine learning perspective in health. *International Journal of Molecular Sciences*, 23(21):13149, 2022.

[26] HIPAA Compliance Assistance. Summary of the hipaa privacy rule. *Office for Civil Rights*, 2003.

[27] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[28] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Neural Information Processing Systems (NIPS)*, 2014.

[29] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[30] Tanujit Chakraborty, Swarup Chattopadhyay, Suchismita Das, Uttam Kumar, and J Senthilnath. Searching for heavy-tailed probability distributions for modeling real-world complex networks. *IEEE Access*, 10:115092–115107, 2022.

[31] Mark E Crovella, Murad S Taqqu, and Azer Bestavros. Heavy-tailed probability distributions in the world wide web. *A practical guide to heavy tails*, 1:3–26, 1998.

[32] Joseph C Gardiner, Zhehui Luo, Xiaoqin Tang, and RV Ramamoorthi. Fitting heavy-tailed distributions to health care data by parametric and bayesian methods. *Journal of Statistical Theory and Practice*, 8:619–652, 2014.

[33] Vinith M Suriyakumar, Nicolas Papernot, Anna Goldenberg, and Marzyeh Ghassemi. Chasing your long tails: Differentially private prediction in health care settings. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, pages 723–734, 2021.

[34] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407, 2014.

[35] Hangyu Zhu, Jinjin Xu, Shiqing Liu, and Yaochu Jin. Federated learning on non-iid data: A survey. *Neurocomputing*, 465:371–390, 2021.

[36] Saba Amiri, Eric Nalisnick, Adam Belloum, Sander Klous, and Leon Gommans. Generating heavy-tailed synthetic data with normalizing flows. In *The 5th Workshop on Tractable Probabilistic Modeling*, 2022.

[37] Saba Amiri, Eric Nalisnick, Adam Belloum, Sander Klous, and Leon Gommans. Practical synthesis of mixed-tailed data with normalizing flows. *Transactions on Machine Learning Research*, 2024.

[38] Saba Amiri, Carlijn Nijhuis, Eric Nalisnick, Adam Belloum, Sander Klous, and Leon Gommans. Latent Semantic Regularization: Enhancing semantic integrity in tabular data synthesis. In *Proceedings of the Symposium on Probabilistic Machine Learning*, Seoul, South Korea, July 2026. URL `https://probml.cc/`. ProbML 2026, co-located with ICML 2026.

[39] Saba Amiri, Eric Nalisnick, Adam Belloum, Sander Klous, and Leon Gommans. Constrained-expressivity differentially private normalizing flows for tabular data synthesis. Working Paper: Manuscript in preparation, 2025.

[40] Saba Amiri, Adam Belloum, Sander Klous, and Leon Gommans. Compressive differentially private federated learning through universal vector quantization. In *AAAI Workshop on Privacy-Preserving Artificial Intelligence*, pages 2–9, 2021.

[41] Saba Amiri, Eric Nalisnick, Adam Belloum, Sander Klous, and Leon Gommans. Differential privacy vs detecting copyright infringement: A case study with normalizing flows. In *1st Workshop on Generative AI and Law, International Conference on Machine Learning (ICML)*, 2023.

[42] Saba Amiri, Adam Belloum, Eric Nalisnick, Sander Klous, and Leon Gommans. On the impact of non-iid data on the performance and fairness of differentially private federated learning. In *2022 52nd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, pages 52–58. IEEE, 2022.

[43] Saba Amiri, Tim Muller, Rosanne J Turner, Karin Hoogoort, Aki Härmä, Johanna Maria Hegeman, Jamila Alsayed Kassem, Milen Kebede, Cees De Laat, Paul Van Der Nat, Auke Pals, Floortje Scheepers, Sander Klous, et al. Optimizing clinical pathways with federated data. *Jusletter-IT*, (20-Februar-2025), 2025.

[44] Tsvetan R Yordanov, Anita CJ Ravelli, Saba Amiri, Marije Vis, Saskia Houterman, Sebastian R Van der Voort, and Ameen Abu-Hanna. Performance of federated learning-based models in the dutch tavi population was comparable to central strategies and outperformed local strategies. *Frontiers in Cardiovascular Medicine*, 11:1399138, 2024.

[45] Corinne Allaart, Saba Amiri, Henri Bal, Adam Belloum, Leon Gommans, Aart Van Halteren, and Sander Klous. Private and secure distributed deep learning: A survey. *ACM Computing Surveys*, 57(4):1–43, 2024.

[46] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *Journal of Machine Learning Research*, 22(57):1–64, 2021. URL `http://jmlr.org/papers/v22/19-1028.html`.

[47] Priyank Jaini, Ivan Kobyzev, Yaoliang Yu, and Marcus Brubaker. Tails of lipschitz triangular flows. In *International Conference on Machine Learning*, pages 4673–4681. PMLR, 2020.

[48] Noseong Park, Mahmoud Mohammadi, Kshitij Gorde, Sushil Jajodia, Hongkyu Park, and Youngmin Kim. Data synthesis based on generative adversarial networks. *Proceedings of the VLDB Endowment*, 11(10):1071–1083, 2018.

[49] Bruce M Hill. A simple general approach to inference about the tail of a distribution. *The annals of statistics*, pages 1163–1174, 1975.

[50] Igor Fedotenkov. A review of more than one hundred pareto-tail index estimators. *Statistica*, 80(3):245–299, 2020.

[51] Sandor Csorgo, Paul Deheuvels, and David Mason. Kernel estimates of the tail index of a distribution. *The Annals of Statistics*, pages 1050–1077, 1985.

[52] Arnold LM Dekkers, John HJ Einmahl, and Laurens De Haan. A moment estimator for the index of an extreme-value distribution. *The Annals of Statistics*, pages 1833–1855, 1989.

[53] David M Mason. Laws of large numbers for sums of extreme values. *The Annals of Probability*, pages 754–764, 1982.

[54] Tucker McElroy and Dimitris N Politis. Moment-based tail index estimation. *Journal of statistical planning and inference*, 137(4):1389–1406, 2007.

[55] Alex YiHou Huang. A value-at-risk approach with kernel estimator. *Applied Financial Economics*, 19(5):379–395, 2009.

[56] Anja Feldmann and Ward Whitt. Fitting mixtures of exponentials to long-tail distributions to analyze network performance models. *Performance evaluation*, 31(3-4):245–279, 1998.

[57] Makoto Okada, Kenji Yamanishi, and Naoki Masuda. Long-tailed distributions of inter-event times as mixtures of exponential distributions. *Royal Society open science*, 7(2):191643, 2020.

[58] Sergio Venturini, Francesca Dominici, and Giovanni Parmigiani. Gamma shape mixtures for heavy-tailed distributions. *The Annals of Applied Statistics*, 2(2): 756–776, 2008.

[59] Paul Hagemann and Sebastian Neumayer. Stabilizing invertible neural networks using mixture models. *Inverse Problems*, 37(8):085002, 2021.

[60] Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *International Conference on Learning Representations*, 2017.

[61] Daniel Hsu and Sivan Sabato. Loss minimization and parameter estimation with heavy tails. *The Journal of Machine Learning Research*, 17(1):543–582, 2016.

[62] Emre Eftelioglu. Geometric median., 2017.

[63] Georges Hebrail and Alice Berard. Individual household electric power consumption. UCI Machine Learning Repository, 2012. DOI: https://doi.org/10.24432/C58K54.

[64] Jordi Fonollosa, Sadique Sheik, Ramón Huerta, and Santiago Marco. Reservoir computing compensates slow response of chemosensor arrays exposed to fast varying gas concentrations in continuous monitoring. *Sensors and Actuators B: Chemical*, 215:618–629, 2015.

[65] Pierre Baldi, Kyle Cranmer, Taylor Faucett, Peter Sadowski, and Daniel Whiteson. Parameterized machine learning for high-energy physics. *arXiv preprint arXiv:1601.07913*, 2016.

[66] Byron P Roe, Hai-Jun Yang, Ji Zhu, Yong Liu, Ion Stancu, and Gordon McGregor. Boosted decision trees as an alternative to artificial neural networks for particle identification. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 543(2-3):577–584, 2005.

[67] George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. *Advances in neural information processing systems*, 30, 2017.

[68] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[69] Donald B Rubin. Statistical disclosure limitation. *Journal of official Statistics*, 9 (2):461–468, 1993.

[70] Saeid Naderiparizi, Xiaoxuan Liang, Berend Zwartsenberg, and Frank Wood. Don't be so negative! score-based generative modeling with oracle-assisted guidance. *arXiv preprint arXiv:2307.16463*, 2023.

[71] Giorgio Giannone, Lyle Regenwetter, Akash Srivastava, Dan Gutfreund, and Faez Ahmed. Learning from invalid data: On constraint satisfaction in generative models. *arXiv preprint arXiv:2306.15166*, 2023.

[72] Jingyi Xu, Hieu Le, and Dimitris Samaras. Assessing sample quality via the latent space of generative models. *arXiv preprint arXiv:2407.15171*, 2024.

[73] Hyunsun Choi, Eric Jang, and Alexander A Alemi. Waic, but why? generative ensembles for robust anomaly detection. *arXiv preprint arXiv:1810.01392*, 2018.

[74] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in neural information processing systems*, 28, 2015.

[75] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

[76] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *International conference on machine learning*, pages 2642–2651. PMLR, 2017.

[77] Mingming Gong, Yanwu Xu, Chunyuan Li, Kun Zhang, and Kayhan Batmanghelich. Twin auxilary classifiers gan. *Advances in neural information processing systems*, 32, 2019.

[78] Mihaela Rosca, Balaji Lakshminarayanan, and Shakir Mohamed. Distribution matching in variational inference. *arXiv preprint arXiv:1802.06847*, 2018.

[79] Vincent Dumoulin, Ishmael Belghazi, Ben Poole, Olivier Mastropietro, Alex Lamb, Martin Arjovsky, and Aaron Courville. Adversarially learned inference. *arXiv preprint arXiv:1606.00704*, 2016.

[80] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *arXiv preprint arXiv:1605.09782*, 2016.

[81] Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*, 2015.

[82] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*, 2017.

[83] Apratim Bhattacharyya, Mario Fritz, and Bernt Schiele. " best-of-many-samples" distribution matching. *arXiv preprint arXiv:1909.12598*, 2019.

[84] Jianmin Bao, Dong Chen, Fang Wen, Houqiang Li, and Gang Hua. Cvae-gan: fine-grained image generation through asymmetric training. In *Proceedings of the IEEE international conference on computer vision*, pages 2745–2754, 2017.

[85] Mihaela Cătălina Stoian, Salijona Dyrmishi, Maxime Cordy, Thomas Lukasiewicz, and Eleonora Giunchiglia. How realistic is your synthetic data? constraining deep generative models for tabular data. *arXiv preprint arXiv:2402.04823*, 2024.

[86] Zilong Zhao, Aditya Kunar, Robert Birke, and Lydia Y Chen. Ctab-gan: Effective table data synthesizing. In *Asian Conference on Machine Learning*, pages 97–112. PMLR, 2021.

[87] Justin Engelmann and Stefan Lessmann. Conditional wasserstein gan-based oversampling of tabular data for imbalanced learning. *Expert Systems with Applications*, 174:114582, 2021.

[88] Zachary Ziegler and Alexander Rush. Latent normalizing flows for discrete sequences. In *International Conference on Machine Learning*, pages 7673–7682. PMLR, 2019.

[89] Apratim Bhattacharyya, Michael Hanselmann, Mario Fritz, Bernt Schiele, and Christoph-Nikolas Straehle. Conditional flow variational autoencoders for structured sequence prediction. *arXiv preprint arXiv:1908.09008*, 2019.

[90] Dingdong Yang, Seunghoon Hong, Yunseok Jang, Tianchen Zhao, and Honglak Lee. Diversity-sensitive conditional generative adversarial networks. *arXiv preprint arXiv:1901.09024*, 2019.

[91] Qi Mao, Hsin-Ying Lee, Hung-Yu Tseng, Siwei Ma, and Ming-Hsuan Yang. Mode seeking generative adversarial networks for diverse image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1429–1437, 2019.

[92] Grigorios G Chrysos, Jean Kossaifi, and Stefanos Zafeiriou. Robust conditional generative adversarial networks. *arXiv preprint arXiv:1805.08657*, 2018.

[93] Soochan Lee, Junsoo Ha, and Gunhee Kim. Harmonizing maximum likelihood with gans for multimodal conditional generation. *arXiv preprint arXiv:1902.09225*, 2019.

[94] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In *Machine learning for healthcare conference*, pages 286–305. PMLR, 2017.

[95] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. *Advances in Neural Information Processing Systems*, 32, 2019.

[96] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. Tabddpm: Modelling tabular data with diffusion models. In *International Conference on Machine Learning*, pages 17564–17579. PMLR, 2023.

[97] Andreas Voskou, Charalambos Christoforou, and Sotirios Chatzis. Transformers with stochastic competition for tabular data modelling. *arXiv preprint arXiv:2407.13238*, 2024.

[98] Dheeru Dua, Casey Graff, et al. Uci machine learning repository. 2017.

[99] David J Spiegelhalter. Learning in probabilistic expert systems. *Bayesian statistics*, 4:447–465, 1992.

[100] Ahmed El Ouadrhiri and Ahmed Abdelhadi. Differential privacy for deep and federated learning: A survey. *IEEE Access*, 10:22359–22380, 2022.

[101] Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC conference on computer and communications security*, pages 308–318, 2016.

[102] Eugene Bagdasaryan, Omid Poursaeed, and Vitaly Shmatikov. Differential privacy has disparate impact on model accuracy. *Advances in Neural Information Processing Systems*, 32:15479–15488, 2019.

[103] George Papamakarios, Eric Nalisnick, Danilo Jimenez Rezende, Shakir Mohamed, and Balaji Lakshminarayanan. Normalizing flows for probabilistic modeling and inference. *The Journal of Machine Learning Research*, 22(1):2617–2680, 2021.

[104] Tom Farrand, Fatemehsadat Mireshghallah, Sahib Singh, and Andrew Trask. Neither private nor fair: Impact of data imbalance on utility and fairness in differential privacy. In *Proceedings of the 2020 Workshop on Privacy-Preserving Machine Learning in Practice*, pages 15–19, 2020.

[105] Jaewoo Lee, Minjung Kim, Yonghyun Jeong, and Youngmin Ro. Differentially private normalizing flows for synthetic tabular data generation. 2022.

[106] Bingyue Su, Yu Wang, Daniele E Schiavazzi, and Fang Liu. Differentially private normalizing flows for density estimation, data synthesis, and variational inference with application to electronic health records. *arXiv preprint arXiv:2302.05787*, 2023.

[107] Garrett Bernstein and Daniel R Sheldon. Differentially private bayesian inference for exponential families. *Advances in Neural Information Processing Systems*, 31, 2018.

[108] Laurent Dinh, David Krueger, and Yoshua Bengio. Nice: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

[109] Yuri van der Does, Rosanne J Turner, Miel JH Bartels, Karin Hagoort, Aäron Metselaar, Floortje Scheepers, Peter D Grünwald, Metten Somers, and Edwin van Dellen. Outcome prediction of electroconvulsive therapy for depression. *Psychiatry Research*, 326:115328, 2023.

[110] Joan Prudic, Mark Olfson, Steven C Marcus, Rice B Fuller, and Harold A Sackeim. Effectiveness of electroconvulsive therapy in community settings. *Biological psychiatry*, 55(3):301–312, 2004.

[111] Linda Van Diermen, Seline Van Den Ameele, Astrid M Kamperman, Bernard CG Sabbe, Tom Vermeulen, Didier Schrijvers, and Tom K Birkenhäger. Prediction of electroconvulsive therapy response and remission in major depression: meta-analysis. *The British journal of psychiatry*, 212(2):71–80, 2018.

[112] Rosanne Turner and Peter Grunwald. Safe sequential testing and effect estimation in stratified count data. In *International Conference on Artificial Intelligence and Statistics*, pages 4880–4893. PMLR, 2023.

[113] Corinne G Allaart, Björn Keyser, Henri Bal, and Aart Van Halteren. Vertical split learning-an exploration of predictive performance in medical and other use cases. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2022.

[114] Milad Nasr, Reza Shokri, and Amir Houmansadr. Comprehensive privacy analysis of deep learning: Stand-alone and federated learning under passive and active white-box inference attacks. 2018.

[115] Jingwen Zhang, Jiale Zhang, Junjun Chen, and Shui Yu. Gan enhanced membership inference: A passive local attack in federated learning. In *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2020.

[116] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pages 2512–2520. IEEE, 2019.

[117] Mengkai Song, Zhibo Wang, Zhifei Zhang, Yang Song, Qian Wang, Ju Ren, and Hairong Qi. Analyzing user-level privacy attack against federated learning. *IEEE Journal on Selected Areas in Communications*, 38(10):2430–2444, 2020.

[118] Yangsibo Huang, Samyak Gupta, Zhao Song, Kai Li, and Sanjeev Arora. Evaluating gradient inversion attacks and defenses in federated learning. *Advances in Neural Information Processing Systems*, 34, 2021.

[119] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. Inverting gradients–how easy is it to break privacy in federated learning? *arXiv preprint arXiv:2003.14053*, 2020.

[120] Cynthia Dwork, Adam Smith, Thomas Steinke, and Jonathan Ullman. Exposed! a survey of attacks on private data. *Annual Review of Statistics and Its Application*, 4:61–84, 2017.

[121] Kang Wei, Jun Li, Ming Ding, Chuan Ma, Howard H Yang, Farhad Farokhi, Shi Jin, Tony QS Quek, and H Vincent Poor. Federated learning with differential privacy: Algorithms and performance analysis. *IEEE Transactions on Information Forensics and Security*, 15:3454–3469, 2020.

[122] Stacey Truex, Ling Liu, Ka-Ho Chow, Mehmet Emre Gursoy, and Wenqi Wei. Ldp-fed: Federated learning with local differential privacy. In *Proceedings of the Third ACM International Workshop on Edge Systems, Analytics and Networking*, pages 61–66, 2020.

[123] T Tony Cai, Yichen Wang, and Linjun Zhang. The cost of privacy: Optimal rates of convergence for parameter estimation with differential privacy. *arXiv preprint arXiv:1902.04495*, 2019.

[124] Xiuting Gu, Tianqing Zhu, Jie Li, Tao Zhang, and Wei Ren. The impact of differential privacy on model fairness in federated learning. In *International Conference on Network and System Security*, pages 419–430. Springer, 2020.

[125] Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

[126] Mustafa Safa Ozdayi and Murat Kantarcioglu. The impact of data distribution on fairness and robustness in federated learning. *arXiv preprint arXiv:2112.01274*, 2021.

[127] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017. URL `http://archive.ics.uci.edu/ml`.

[128] Vasileios Iosifidis and Eirini Ntoutsi. Adafair: Cumulative fairness adaptive boosting. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 781–790, 2019.

[129] Tai Le Quy, Arjun Roy, Vasileios Iosifidis, and Eirini Ntoutsi. A survey on datasets for fairness-aware machine learning. *arXiv preprint arXiv:2110.00530*, 2021.

[130] Vasileios Iosifidis, Arjun Roy, and Eirini Ntoutsi. Parity-based cumulative fairness-aware boosting. *arXiv preprint arXiv:2201.01148*, 2022.

[131] James R Foulds, Rashidul Islam, Kamrun Naher Keya, and Shimei Pan. An intersectional definition of fairness. In *2020 IEEE 36th International Conference on Data Engineering (ICDE)*, pages 1918–1921. IEEE, 2020.

[132] Anthony F Shorrocks. The class of additively decomposable inequality measures. *Econometrica: Journal of the Econometric Society*, pages 613–625, 1980.

[133] Moritz Hardt, Eric Price, and Nati Srebro. Equality of opportunity in supervised learning. *Advances in neural information processing systems*, 29:3315–3323, 2016.

[134] Ilya Mironov. Rényi differential privacy. In *2017 IEEE 30th Computer Security Foundations Symposium (CSF)*, pages 263–275. IEEE, 2017.

[135] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450, 2020.

[136] Yahya H Ezzeldin, Shen Yan, Chaoyang He, Emilio Ferrara, and Salman Avestimehr. Fairfed: Enabling group fairness in federated learning. *arXiv preprint arXiv:2110.00857*, 2021.

[137] General Data Protection Regulation (GDPR). General data protection regulation (gdpr) final text neatly arranged. 2018.

[138] Harry Chandra Tanuwidjaja, Rakyong Choi, Seunggeun Baek, and Kwangjo Kim. Privacy-preserving deep learning on machine learning as a servicea comprehensive survey. *IEEE Access*, 8:167425–167447, 2020. doi: 10.1109/ACCESS.2020.3023084.

[139] Qiao Zhang, Chunsheng Xin, and Hongyi Wu. Privacy-preserving deep learning based on multiparty secure computation: A survey. *IEEE Internet of Things Journal*, 8(13):10412–10429, 2021. doi: 10.1109/JIOT.2021.3058638.

[140] Xuefei Yin, Yanming Zhu, and Jiankun Hu. A comprehensive survey of privacy-preserving federated learning: A taxonomy, review, and future directions. *ACM Comput. Surv.*, 54(6), jul 2021. ISSN 0360-0300. doi: 10.1145/3460427. URL https://doi.org/10.1145/3460427.

[141] David Enthoven and Zaid Al-Ars. *An Overview of Federated Deep Learning Privacy Attacks and Defensive Strategies*, pages 173–196. Springer International Publishing, Cham, 2021. ISBN 978-3-030-70604-3. doi: 10.1007/978-3-030-70604-3_8. URL https://doi.org/10.1007/978-3-030-70604-3_8.

[142] Emmanuel Antwi-Boasiako, Shijie Zhou, Yongjian Liao, Qihe Liu, Yuyu Wang, and Kwabena Owusu-Agyemang. Privacy preservation in distributed deep learning: A survey on distributed deep learning, privacy preservation techniques used and interesting research directions. *Journal of Information Security and Applications*, 61:102949, 2021. ISSN 2214-2126. doi: https://doi.org/10.1016/j.jisa.2021.102949. URL https://www.sciencedirect.com/science/article/pii/S2214212621001630.

[143] Praneeth Vepakomma, Otkrist Gupta, Tristan Swedish, and Ramesh Raskar. Split learning for health: Distributed deep learning without sharing raw patient data. *arXiv preprint arXiv:1812.00564*, 2018.

[144] Lingjuan Lyu, Han Yu, Xingjun Ma, Chen Chen, Lichao Sun, Jun Zhao, Qiang Yang, and S Yu Philip. Privacy and robustness in federated learning: Attacks and defenses. *IEEE transactions on neural networks and learning systems*, 2022.

[145] Maria Rigaki and Sebastian Garcia. A survey of privacy attacks in machine learning. *ACM Computing Surveys*, 56(4):1–34, 2023.

[146] Bo Liu, Ming Ding, Sina Shaham, Wenny Rahayu, Farhad Farokhi, and Zihuai Lin. When machine learning meets privacy: A survey and outlook. *ACM Computing Surveys (CSUR)*, 54(2):1–36, 2021.

[147] Hongsheng Hu, Zoran Salcic, Lichao Sun, Gillian Dobbie, Philip S Yu, and Xuyun Zhang. Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)*, 54(11s):1–37, 2022.

[148] Seung Ho Na, Hyeong Gwon Hong, Junmo Kim, and Seungwon Shin. Closing the loophole: rethinking reconstruction attacks in federated learning from a privacy standpoint. In *Proceedings of the 38th Annual Computer Security Applications Conference*, pages 332–345, 2022.

[149] Zecheng He, Tianwei Zhang, and Ruby B Lee. Model inversion attacks against collaborative inference. In *Proceedings of the 35th Annual Computer Security Applications Conference*, pages 148–162, 2019.

[150] Junhao Zhou, Yufei Chen, Chao Shen, and Yang Zhang. Property inference attacks against gans. *arXiv preprint arXiv:2111.07608*, 2021.

[151] Youyang Qu, Mohammad Reza Nosouhi, Lei Cui, and Shui Yu. Leading attacks in privacy protection domain. In *Personalized Privacy Protection in Big Data*, pages 15–21. Springer, 2021.

[152] Trung Ha, Tran Khanh Dang, Tran Tri Dang, Tuan Anh Truong, and Manh Tuan Nguyen. Differential privacy in deep learning: an overview. In *2019 International Conference on Advanced Computing and Applications (ACOMP)*, pages 97–102. IEEE, 2019.

[153] Li Deng. The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, 29(6):141–142, 2012.

[154] Nathan Dowlin, Ran Gilad-Bachrach, Kim Laine, Kristin Lauter, Michael Naehrig, John Wernsing, and Microsoft Research. CryptoNets: Applying Neural Networks to Encrypted Data with High Throughput and Accuracy. Technical report, 2016. URL `http://sealcrypto.codeplex.com`.

[155] Florian Bourse, Michele Minelli, Matthias Minihold, and Pascal Paillier. Fast homomorphic evaluation of deep discretized neural networks. In Hovav Shacham and Alexandra Boldyreva, editors, *Advances in Cryptology – CRYPTO 2018*, pages 483–512, Cham, 2018. Springer International Publishing. ISBN 978-3-319-96878-0.

[156] Ehsan Hesamifard, Hassan Takabi, and Mehdi Ghasemi. Cryptodl: Deep neural networks over encrypted data. *CoRR*, abs/1711.05189, 2017. URL `http://arxiv.org/abs/1711.05189`.

[157] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[158] Jian Liu, Mika Juuti, Yao Lu, and N. Asokan. Oblivious neural network predictions via minionn transformations. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 619631, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349468. doi: 10.1145/3133956.3134056. URL `https://doi.org/10.1145/3133956.3134056`.

[159] Sadegh Riazi, Christian Weinert, Oleksandr Tkachenko, Ebrahim M. Songhori, Thomas Schneider, and Farinaz Koushanfar. Chameleon: A hybrid secure computation framework for machine learning applications. 01 2018.

[160] Edward Chou, Josh Beal, Daniel Levy, Serena Yeung, Albert Haque, and Li Fei-Fei. Faster cryptonets: Leveraging sparsity for real-world encrypted inference, 2018. URL `https://arxiv.org/abs/1811.09953`.

[161] Bita Darvish Rouhani, M. Sadegh Riazi, and Farinaz Koushanfar. DeepSecure: Scalable Provably-Secure Deep Learning Bita. In *Proceedings - Design Automation Conference*, volume Part F1377, pages 1–6, 2018. ISBN 9781450357005.

[162] C. Juvekar, V. Vaikuntanathan, and A. Chandrakasan. Gazelle: A low latency framework for secure neural network inference. In *IACR Cryptol. ePrint Arch.*, 2018.

[163] Amartya Sanyal, Matt J. Kusner, Adrià Gascón, and Varun Kanade. TAPAS: tricks to accelerate (encrypted) prediction as a service. *CoRR*, abs/1806.03461, 2018. URL http://arxiv.org/abs/1806.03461.

[164] Beata Strack, Jonathan P DeShazo, Chris Gennings, Juan L Olmo, Sebastian Ventura, Krzysztof J Cios, and John N Clore. Impact of hba1c measurement on hospital readmission rates: analysis of 70,000 clinical database patient records. *BioMed research international*, 2014, 2014.

[165] Fabian Boemer, Anamaria Costache, Rosario Cammarota, and Casimir Wierzynski. ngraph-he2: A high-throughput framework for neural network inference on encrypted data. *CoRR*, abs/1908.04172, 2019. URL http://arxiv.org/abs/1908.04172.

[166] M. Riazi, Mohammad Samragh, Hao Chen, K. Laine, K. Lauter, and F. Koushanfar. Xonn: Xnor-based oblivious deep neural network inference. In *IACR Cryptol. ePrint Arch.*, 2019.

[167] Pratyush Mishra, Ryan Lehmkuhl, Akshayaram Srinivasan, Wenting Zheng, and Raluca Ada Popa. Delphi: A cryptographic inference service for neural networks. Cryptology ePrint Archive, Paper 2020/050, 2020. URL https://eprint.iacr.org/2020/050. https://eprint.iacr.org/2020/050.

[168] Brandon Reagen, Wooseok Choi, Yeongil Ko, Vincent Lee, Gu-Yeon Wei, Hsien-Hsin S. Lee, and David Brooks. Cheetah: Optimizing and accelerating homomorphic encryption for private inference, 2020. URL https://arxiv.org/abs/2006.00505.

[169] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[170] Ahmad Al Badawi, Chao Jin, Jie Lin, Chan Fook Mun, Sim Jun Jie, Benjamin Hong Meng Tan, Xiao Nan, Khin Mi Mi Aung, and Vijay Ramaseshan Chandrasekhar. Towards the alexnet moment for homomorphic encryption: Hcnn, the first homomorphic cnn on encrypted data with gpus. *IEEE Transactions on Emerging Topics in Computing*, 9(3):1330–1343, 2021. doi: 10.1109/TETC.2020.3014636.

[171] Guillermo Lloret-Talavera, Marc Jorda, Harald Servat, Fabian Boemer, Chetan Chauhan, Shigeki Tomishima, Nilesh N. Shah, and Antonio J. Pena. Enabling homomorphically encrypted inference for large dnn models. *IEEE Transactions on Computers*, 71(5):1145–1155, 2022. doi: 10.1109/TC.2021.3076123.

[172] Jia Duan, Jiantao Zhou, Li Yuanman, and Caishi Huang. Privacy-preserving and verifiable deep learning inference based on secret sharing. *Neurocomputing*, 483, 01 2022. doi: 10.1016/j.neucom.2022.01.061.

[173] Joon-Woo Lee, Hyungchul Kang, Yongwoo Lee, Woosuk Choi, Jieun Eom, Maxim Deryabin, Eunsang Lee, Junghyun Lee, Donghoon Yoo, Young-Sik Kim, and Jong-Seon No. Privacy-preserving machine learning with fully homomorphic encryption for deep neural network. *IEEE Access*, 10:30039–30054, 2022. doi: 10.1109/ACCESS.2022.3159694.

[174] Louis J. M. Aslett, Pedro M. Esperança, and Chris C. Holmes. Encrypted statistical machine learning: new privacy preserving methods, 2015. URL `https://arxiv.org/abs/1508.06845`.

[175] Thore Graepel, Kristin Lauter, and Michael Naehrig. Ml confidential: Machine learning on encrypted data. Cryptology ePrint Archive, Paper 2012/323, 2012. URL `https://eprint.iacr.org/2012/323`. `https://eprint.iacr.org/2012/323`.

[176] Paul Kirchner and Pierre-Alain Fouque. Revisiting lattice attacks on overstretched ntru parameters. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 3–26, Cham, 2017. Springer International Publishing. ISBN 978-3-319-56620-7.

[177] Martin Albrecht, Shi Bai, and Léo Ducas. A subfield lattice attack on overstretched ntru assumptions: Cryptanalysis of some fhe and graded encoding schemes. Cryptology ePrint Archive, Paper 2016/127, 2016. URL `https://eprint.iacr.org/2016/127`. `https://eprint.iacr.org/2016/127`.

[178] Alberto Ibarrondo and Melek Önen. Fhe-compatible batch normalization for privacy preserving deep learning. In Joaquin Garcia-Alfaro, Jordi Herrera-Joancomartí, Giovanni Livraga, and Ruben Rios, editors, *Data Privacy Management, Cryptocurrencies and Blockchain Technology*, pages 389–404, Cham, 2018. Springer International Publishing. ISBN 978-3-030-00305-0.

[179] Fabian Boemer, Yixing Lao, Rosario Cammarota, and Casimir Wierzynski. Ngraph-he: A graph compiler for deep learning on homomorphically encrypted data. In *Proceedings of the 16th ACM International Conference on Computing Frontiers*, CF '19, page 313, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366854. doi: 10.1145/3310273.3323047. URL `https://doi.org/10.1145/3310273.3323047`.

[180] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game, or a completeness theorem for protocols with honest majority. In *Providing Sound Foundations for Cryptography*, 2019.

[181] Michael Rabin. How to exchange secrets with oblivious transfer. *IACR Cryptology ePrint Archive*, 2005:187, 01 2005.

[182] Zahra Ghodsi, Tianyu Gu, and Siddharth Garg. Safetynets: Verifiable execution of deep neural networks on an untrusted cloud. *CoRR*, abs/1706.10268, 2017. URL `http://arxiv.org/abs/1706.10268`.

[183] Payman Mohassel and Yupeng Zhang. SecureML: A System for Scalable Privacy-Preserving Machine Learning. In *Proceedings - IEEE Symposium on Security and Privacy*, pages 19–38. Institute of Electrical and Electronics Engineers Inc., jun 2017. ISBN 9781509055326. doi: 10.1109/SP.2017.12.

[184] Payman Mohassel and Peter Rindal. Aby3: A mixed protocol framework for machine learning. Cryptology ePrint Archive, Paper 2018/403, 2018. URL https://eprint.iacr.org/2018/403. https://eprint.iacr.org/2018/403.

[185] Ehsan Hesamifard, Daniel Takabi, Mehdi Ghasemi, and Rebecca Wright. Privacy-preserving machine learning as a service. *Proceedings on Privacy Enhancing Technologies*, 2018:123–142, 06 2018. doi: 10.1515/popets-2018-0024.

[186] Qiao Zhang, Cong Wang, Hongyi Wu, Chunsheng Xin, and T.V.X. Phuong. Gelu-net: A globally encrypted, locally unencrypted deep neural network for privacy-preserved learning. pages 3933–3939, 07 2018. doi: 10.24963/ijcai.2018/547.

[187] Sameer Wagh, Divya Gupta, and Nishanth Chandran. Securenn: Efficient and private neural network training. In *Privacy Enhancing Technologies Symposium*. (PETS 2019), February 2019. URL https://www.microsoft.com/en-us/research/publication/securenn-efficient-and-private-neural-network-training/.

[188] Nitin Agrawal, Ali Shahin Shamsabadi, Matt J. Kusner, and Adrià Gascón. QUOTIENT: two-party secure neural network training and prediction. *CoRR*, abs/1907.03372, 2019. URL http://arxiv.org/abs/1907.03372.

[189] Owusu Agyemang Kwabena, Zhiguang Zhen Qin, Zhiguang Zhen Qin, and Tianming Zhuang. Mscryptonet multi-scheme privacy-preserving deep learning in cloud computing, 2019. ISSN 21693536. URL https://ieeexplore.ieee.org/document/8651459.

[190] Karthik Nandakumar, Nalini Ratha, Sharath Pankanti, and Shai Halevi. Towards deep neural network training on encrypted data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 40–48, 2019. doi: 10.1109/CVPRW.2019.00011.

[191] Anders Dalskov, Daniel E. Escudero, and Marcel Keller. Fantastic four: Honest-majority four-party secure computation with malicious security. *IACR Cryptol. ePrint Arch.*, 2020:1330, 2020.

[192] Nishat Koti, Mahak Pancholi, Arpita Patra, and Ajith Suresh. SWIFT: super-fast and robust privacy-preserving machine learning. *CoRR*, abs/2005.10296, 2020. URL https://arxiv.org/abs/2005.10296.

[193] Arpita Patra and Ajith Suresh. BLAZE: blazing fast privacy-preserving machine learning. *CoRR*, abs/2005.09042, 2020. URL https://arxiv.org/abs/2005.09042.

[194] C Okan Sakar, Gorkem Serbes, Aysegul Gunduz, Hunkar C Tunc, Hatice Nizam, Betul Erdogdu Sakar, Melih Tutuncu, Tarkan Aydin, M Erdem Isenkul, and Hulya Apaydin. A comparative analysis of speech signal processing algorithms for parkinsons disease classification and the use of the tunable q-factor wavelet transform. *Applied Soft Computing*, 74:255–263, 2019.

[195] Sameer Wagh, Shruti Tople, Fabrice Benhamouda, Eyal Kushilevitz, Prateek Mittal, and Tal Rabin. FALCON: honest-majority maliciously secure framework for private deep learning. *CoRR*, abs/2004.02229, 2020. URL `https://arxiv.org/abs/2004.02229`.

[196] Rahul Rachuri and Ajith Suresh. Trident: Efficient 4pc framework for privacy preserving machine learning. *CoRR*, abs/1912.02631, 2019. URL `http://arxiv.org/abs/1912.02631`.

[197] Theo Ryffel, Pierre Tholoniat, David Pointcheval, and Francis R. Bach. Ariann: Low-interaction privacy-preserving deep learning via function secret sharing. *Proceedings on Privacy Enhancing Technologies*, 2022:291 – 316, 2022.

[198] Nishat Koti, Arpita Patra, Rahul Rachuri, and Ajith Suresh. Tetrad: Actively secure 4pc for secure training and inference. *CoRR*, abs/2106.02850, 2021. URL `https://arxiv.org/abs/2106.02850`.

[199] Demmler, Schneider, and Zohner. Aby - a framework for efficient mixed-protocol secure two-party computation. Internet Society, 2015. doi: 10.14722/ndss.2015.23113.

[200] Yuan Xu. On discrete orthogonal polynomials of several variables. *Advances in Applied Mathematics*, 33(3):615–632, 2004. ISSN 0196-8858. doi: https://doi.org/10.1016/j.aam.2004.03.002. URL `https://www.sciencedirect.com/science/article/pii/S0196885804000405`.

[201] Sameer Wagh, Divya Gupta, and Nishanth Chandran. Securenn: Efficient and private neural network training. *IACR Cryptol. ePrint Arch.*, 2018:442, 2018.

[202] Gilad Asharov, Yehuda Lindell, Thomas Schneider, and Michael Zohner. More efficient oblivious transfer and extensions for faster secure computation. Cryptology ePrint Archive, Paper 2013/552, 2013. URL `https://eprint.iacr.org/2013/552`. `https://eprint.iacr.org/2013/552`.

[203] Elette Boyle, Niv Gilboa, and Yuval Ishai. Function secret sharing. In *Advances in Cryptology-EUROCRYPT 2015: 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Sofia, Bulgaria, April 26-30, 2015, Proceedings, Part II*, pages 337–367. Springer, 2015.

[204] Reza Shokri and Vitaly Shmatikov. Privacy-preserving deep learning. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1310–1321, 2015.

[205] Robin C Geyer, Tassilo Klein, and Moin Nabi. Differentially private federated learning: A client level perspective. *arXiv preprint arXiv:1712.07557*, 2017.

[206] Lichao Sun, Jianwei Qian, Xun Chen, and Philip S Yu. LDP-FL: Practical Private Aggregation in Federated Learning with Local Differential Privacy. Technical report, 2020.

[207] John C Duchi, Michael I Jordan, and Martin J Wainwright. Minimax optimal procedures for locally private estimation. *Journal of the American Statistical Association*, 113(521):182–201, 2018.

[208] Qinqing Zheng, Shuxiao Chen, Qi Long, and Weijie J Su. Federated f-Differential Privacy. Technical report, 2021.

[209] Valentin Hartmann and Robert West. Privacy-preserving distributed learning with secret gradient descent. *arXiv preprint arXiv:1906.11993*, 2019.

[210] Mohamed Seif, Ravi Tandon, and Ming Li. Wireless federated learning with local differential privacy. *arXiv preprint arXiv:2002.05151*, 2020.

[211] Brett K Beaulieu-Jones, William Yuan, Samuel G Finlayson, and Zhiwei Steven Wu. Privacy-preserving distributed deep learning for clinical data. *arXiv preprint arXiv:1812.01484*, 2018.

[212] Meng Hao, Hongwei Li, Guowen Xu, Sen Liu, and Haomiao Yang. Towards efficient and privacy-preserving federated deep learning. In *ICC 2019-2019 IEEE International Conference on Communications (ICC)*, pages 1–6. IEEE, 2019.

[213] Stacey Truex, Nathalie Baracaldo, Ali Anwar, Thomas Steinke, Heiko Ludwig, Rui Zhang, and Yi Zhou. A hybrid approach to privacy-preserving federated learning. In *Proceedings of the 12th ACM workshop on artificial intelligence and security*, pages 1–11, 2019.

[214] Jiale Zhang, Junyu Wang, Yanchao Zhao, and Bing Chen. An efficient federated learning scheme with differential privacy in mobile edge computing. In *International Conference on Machine Learning and Intelligent Communications*, pages 538–550. Springer, 2019.

[215] Zonghao Huang, Rui Hu, Yuanxiong Guo, Eric Chan-Tin, and Yanmin Gong. DP-ADMM: ADMM-Based Distributed Learning with Differential Privacy. *IEEE Transactions on Information Forensics and Security*, 15:1002–1012, 2020. ISSN 15566021. doi: 10.1109/TIFS.2019.2931068.

[216] Xiaoyuan Liu, Hongwei Li, Guowen Xu, Rongxing Lu, and Miao He. Adaptive privacy-preserving federated learning. *Peer-to-Peer Networking and Applications*, 13(6):2356–2366, 2020.

[217] Yansheng Wang, Yongxin Tong, and Dingyuan Shi. Federated Latent Dirichlet Allocation: A Local Differential Privacy Based Framework. Technical report, 2020. URL `www.aaai.org`.

[218] Parham Gohari, Bo Chen, Bo Wu, Matthew Hale, and Ufuk Topcu. Privacy-Preserving Teacher-Student Deep Reinforcement Learning. 2021. URL `http://arxiv.org/abs/2102.09599`.

[219] Peter Kairouz, Brendan Mcmahan, Shuang Song, Om Thakkar, Abhradeep Thakurta, and Zheng Xu. Practical and Private (Deep) Learning without Sampling or Shuffling. Technical report, 2021.

[220] Muah Kim, Onur Günlü, and Rafael F Schaefer. FEDERATED LEARNING WITH LOCAL DIFFERENTIAL PRIVACY: TRADE-OFFS BETWEEN PRIVACY, UTILITY, AND COMMUNICATION. Technical report, 2021.

[221] Mohammad Malekzadeh, Burak Hasircioglu, Nitish Mital, Kunal Katarya, Emre Ozfatura, and Deniz G Í UndüzundĹundüz. Dopamine: Differentially Private Federated Learning on Medical Data. Technical report, 2021. URL https://www.kaggle.com/c/aptos2019-blindness-detection/notebooks.

[222] Zhengxin Yu, Jia Hu, Geyong Min, Zi Wang, Wang Miao, and Shancang Li. Privacy-preserving federated deep learning for cooperative hierarchical caching in fog computing. *IEEE Internet of Things Journal*, 2021.

[223] Wenqi Wei, Ling Liu, Yanzhao Wut, Gong Su, and Arun Iyengar. Gradient-leakage resilient federated learning. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*, pages 797–807. IEEE, 2021.

[224] Lingchen Zhao, Qian Wang, Qin Zou, Yan Zhang, and Yanjiao Chen. Privacy-preserving collaborative deep learning with unreliable participants. *IEEE Transactions on Information Forensics and Security*, 15:1486–1500, 2019.

[225] Tran Thi Phuong et al. Privacy-preserving deep learning via weight transmission. *IEEE Transactions on Information Forensics and Security*, 14(11):3003–3015, 2019.

[226] Xiaoyuan Liu, Hongwei Li, Guowen Xu, Rongxing Lu, and Miao He. Adaptive privacy-preserving federated learning. *Peer-to-Peer Networking and Applications*, 13(6):2356–2366, nov 2020. ISSN 19366450. doi: 10.1007/s12083-019-00869-2. URL https://doi.org/10.1007/s12083-019-00869-2.

[227] Jun Zhou, Zhenfu Cao, Xiaolei Dong, and Xiaodong Lin. Ppdm: A privacy-preserving protocol for cloud-assisted e-healthcare systems. *IEEE Journal of Selected Topics in Signal Processing*, 9(7):1332–1344, 2015.

[228] Fengyi Tang, Wei Wu, Jian Liu, Huimei Wang, and Ming Xian. Privacy-preserving distributed deep learning via homomorphic re-encryption. *Electronics*, 8(4), 2019. ISSN 2079-9292. doi: 10.3390/electronics8040411. URL https://www.mdpi.com/2079-9292/8/4/411.

[229] Jinshuo Dong, Aaron Roth, and Weijie J Su. Gaussian differential privacy. *arXiv preprint arXiv:1905.02383*, 2019.

[230] Jing Ma, Si-Ahmed Naas, Stephan Sigg, and Xixiang Lyu. Privacy-preserving federated learning based on multi-key homomorphic encryption. *International Journal of Intelligent Systems*, 37(9):5880–5901, 2022. doi: https://doi.org/10.1002/int.22818. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/int.22818.

[231] Le Trieu Phong, Yoshinori Aono, Takuya Hayashi, Lihua Wang, and Shiho Moriai. Privacy-preserving deep learning via additively homomorphic encryption. *IEEE Transactions on Information Forensics and Security*, 13(5):1333–1345, 2018. doi: 10.1109/TIFS.2017.2787987.

[232] Richard Lindner and Chris Peikert. Better key sizes (and attacks) for lwe-based encryption. In *Topics in Cryptology–CT-RSA 2011: The Cryptographers Track at the RSA Conference 2011, San Francisco, CA, USA, February 14-18, 2011. Proceedings*, pages 319–339. Springer, 2011.

[233] Runhua Xu, Nathalie Baracaldo, Yi Zhou, Ali Anwar, and Heiko Ludwig. Hybridalpha: An efficient approach for privacy-preserving federated learning. *CoRR*, abs/1912.05897, 2019. URL `http://arxiv.org/abs/1912.05897`.

[234] Chengliang Zhang, Suyi Li, Junzhe Xia, Wei Wang, Feng Yan, and Yang Liu. Batchcrypt: Efficient homomorphic encryption for cross-silo federated learning. In *Proceedings of the 2020 USENIX Annual Technical Conference (USENIX ATC 2020)*, 2020.

[235] Jia Duan, Jiantao Zhou, and Yuanman Li. Privacy-preserving distributed deep learning based on secret sharing. *Information Sciences*, 527:108–127, 2020. ISSN 0020-0255. doi: https://doi.org/10.1016/j.ins.2020.03.074. URL `https://www.sciencedirect.com/science/article/pii/S0020025520302553`.

[236] Hangyu Zhu, Rui Wang, Yaochu Jin, Kaitai Liang, and Jianting Ning. Distributed additive encryption and quantization for privacy preserving federated deep learning. *Neurocomputing*, 463:309–327, 2021. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2021.08.062. URL `https://www.sciencedirect.com/science/article/pii/S0925231221012522`.

[237] Guowen Xu, Hongwei Li, Yun Zhang, Shengmin Xu, Jianting Ning, and Robert H. Deng. Privacy-preserving federated deep learning with irregular users. *IEEE Transactions on Dependable and Secure Computing*, 19(2):1364–1381, 2022. doi: 10.1109/TDSC.2020.3005909.

[238] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H. Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, CCS '17, page 11751191, New York, NY, USA, 2017. Association for Computing Machinery. ISBN 9781450349468. doi: 10.1145/3133956.3133982. URL `https://doi.org/10.1145/3133956.3133982`.

[239] Sinem Sav, Apostolos Pyrgelis, Juan Ramón Troncoso-Pastoriza, David Froelicher, Jean-Philippe Bossuat, Joao Sa Sousa, and Jean-Pierre Hubaux. POSEIDON: privacy-preserving federated neural network learning. *CoRR*, abs/2009.00349, 2020. URL `https://arxiv.org/abs/2009.00349`.

[240] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[241] James Bell, K. A. Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. Secure single-server aggregation with (poly)logarithmic overhead. Cryptology ePrint Archive, Paper 2020/704, 2020. URL https://eprint.iacr.org/2020/704. https://eprint.iacr.org/2020/704.

[242] Borja Balle, James Bell, Adrià Gascón, and Kobbi Nissim. The privacy blanket of the shuffle model. In *Advances in Cryptology–CRYPTO 2019: 39th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 18–22, 2019, Proceedings, Part II 39*, pages 638–667. Springer, 2019.

[243] Swanand Kadhe, Nived Rajaraman, Onur Ozan Koyluoglu, and Kannan Ramchandran. Fastsecagg: Scalable secure aggregation for privacy-preserving federated learning. *CoRR*, abs/2009.11248, 2020. URL https://arxiv.org/abs/2009.11248.

[244] Jinhyun So, Baak Güler, and A. Salman Avestimehr. Turbo-aggregate: Breaking the quadratic aggregation barrier in secure federated learning. *IEEE Journal on Selected Areas in Information Theory*, 2(1):479–489, 2021. doi: 10.1109/JSAIT.2021.3054610.

[245] Guowen Xu, Hongwei Li, Sen Liu, Kan Yang, and Xiaodong Lin. Verifynet: Secure and verifiable federated learning. *IEEE Transactions on Information Forensics and Security*, 15:911–926, 2019.

[246] Tensorflow. Text generation with an rnn., 2019. URL https://www.tensorflow.org/tutorials/text/text_generation.

[247] Yang Liu, Tao Fan, Tianjian Chen, Qian Xu, and Qiang Yang. Fate: An industrial grade platform for collaborative learning with data protection. *The Journal of Machine Learning Research*, 22(1):10320–10325, 2021.

[248] Gang Han, Tiantian Zhang, Yinghui Zhang, Guowen Xu, Jianfei Sun, and Jin Cao. Verifiable and privacy preserving federated learning without fully trusted centers. *Journal of Ambient Intelligence and Humanized Computing*, 13:1–11, 03 2022. doi: 10.1007/s12652-020-02664-x.

[249] Lourdes Martínez-Villaseñor, Hiram Ponce, Jorge Brieva, Ernesto Moya-Albor, José Núñez-Martínez, and Carlos Peñafort-Asturiano. Up-fall detection dataset: A multimodal approach. *Sensors*, 19(9):1988, 2019.

[250] Guowen Xu, Hongwei Li, Sen Liu, Kan Yang, and Xiaodong Lin. Verifynet: Secure and verifiable federated learning. *Trans. Info. For. Sec.*, 15:911926, jan 2020. ISSN 1556-6013. doi: 10.1109/TIFS.2019.2929409. URL https://doi.org/10.1109/TIFS.2019.2929409.

[251] Chandra Thapa, Pathum Chamikara Mahawaga Arachchige, Seyit Camtepe, and Lichao Sun. Splitfed: When federated learning meets split learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 8485–8493, 2022.

[252] Dario Pasquini, Giuseppe Ateniese, and Massimo Bernaschi. Unleashing the tiger: Inference attacks on split learning. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, pages 2113–2129, 2021.

[253] Valeria Turina, Zongshun Zhang, Flavio Esposito, and Ibrahim Matta. Federated or split? a performance and privacy analysis of hybrid split and federated learning architectures. In *2021 IEEE 14th International Conference on Cloud Computing (CLOUD)*, pages 250–260. IEEE, 2021.

[254] Ege Erdoğan, Alptekin Küpçü, and A Ercüment Çiçek. Unsplit: Data-oblivious model inversion, model stealing, and label inference attacks against split learning. In *Proceedings of the 21st Workshop on Privacy in the Electronic Society*, pages 115–124, 2022.

[255] Oscar Li, Jiankai Sun, Xin Yang, Weihao Gao, Hongyi Zhang, Junyuan Xie, Virginia Smith, and Chong Wang. Label leakage and protection in two-party split learning. *arXiv preprint arXiv:2102.08504*, 2021.

[256] George-Liviu Pereteanu, Amir Alansary, and Jonathan Passerat-Palmbach. Split he: Fast secure inference combining split learning and homomorphic encryption. *arXiv preprint arXiv:2202.13351*, 2022.

[257] Nicolas Papernot, Martín Abadi, Ulfar Erlingsson, Ian Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. *arXiv preprint arXiv:1610.05755*, 2016.

[258] Nicolas Papernot, Shuang Song, Ilya Mironov, Ananth Raghunathan, Kunal Talwar, and Úlfar Erlingsson. Scalable private learning with pate. *arXiv preprint arXiv:1802.08908*, 2018.

[259] Yunhui Long, Suxin Lin, Zhuolin Yang, Carl A Gunter, Han Liu, and Bo Li. Scalable differentially private data generation via private aggregation of teacher ensembles. 2019.

[260] Hua Wang, Sheng Gao, Huanyu Zhang, Milan Shen, and Weijie J Su. Analytical composition of differential privacy via the edgeworth accountant. *arXiv preprint arXiv:2206.04236*, 2022.

[261] SEAL. Microsoft SEAL (release 3.0). `http://sealcrypto.org`, October 2018. Microsoft Research, Redmond, WA.

[262] Iker Ceballos, Vivek Sharma, Eduardo Mugica, Abhishek Singh, Alberto Roman, Praneeth Vepakomma, and Ramesh Raskar. Splitnn-driven vertical partitioning. *arXiv preprint arXiv:2008.04137*, 2020.

[263] Hsin-Pai Cheng, Patrick Yu, Haojing Hu, Syed Zawad, Feng Yan, Shiyu Li, Hai Li, and Yiran Chen. Towards decentralized deep learning with differential privacy. In *International Conference on Cloud Computing*, pages 130–145. Springer, 2019.

[264] Hsin-Pai Cheng, Patrick Yu, Haojing Hu, Feng Yan, Shiyu Li, Hai Li, and Yiran Chen. Leasgd: an efficient and privacy-preserving decentralized algorithm for distributed learning. *arXiv preprint arXiv:1811.11124*, 2018.

[265] Yushuang Yan, Qingqi Pei, and Hongning Li. Privacy-preserving compressive model for enhanced deep-learning-based service provision system in edge computing. *IEEE Access*, 7:92921–92937, 2019.

[266] Sen Yuan, Milan Shen, Ilya Mironov, and Anderson CA Nascimento. Practical, label private deep learning training based on secure multiparty computation and differential privacy. *Cryptology ePrint Archive*, 2021.

[267] Josep Domingo-Ferrer, Alberto Blanco-Justicia, Jesús Manjón, and David Sánchez. Secure and privacy-preserving federated learning via co-utility, 2021. URL `https://arxiv.org/abs/2108.01913`.

[268] Zhe Peng, Jianliang Xu, Xiaowen Chu, Shang Gao, Yuan Yao, Rong Gu, and Yuzhe Tang. Vfchain: Enabling verifiable and auditable federated learning via blockchain systems. *IEEE Transactions on Network Science and Engineering*, 9 (1):173–186, 2022. doi: 10.1109/TNSE.2021.3050781.

[269] Ruozhou Zuo, Haibo Tian, Zhiyuan An, and Fangguo Zhang. Post-quantum privacy-preserving aggregation in federated learning based on lattice. In *Cyberspace Safety and Security: 14th International Symposium, CSS 2022, Xi'an, China, October 16–18, 2022, Proceedings*, pages 314–326. Springer, 2022.

[270] Alessandro Renda, Pietro Ducange, Francesco Marcelloni, Dario Sabella, Miltiadis C Filippou, Giovanni Nardini, Giovanni Stea, Antonio Virdis, Davide Micheli, Damiano Rapone, et al. Federated learning of explainable ai models in 6g systems: Towards secure and automated vehicle networking. *Information*, 13(8):395, 2022.

[271] Anran Li, Rui Liu, Ming Hu, Luu Anh Tuan, and Han Yu. Towards interpretable federated learning. *arXiv preprint arXiv:2302.13473*, 2023.

