FACULDADE DE CIÊNCIAS DA UNIVERSIDADE DO PORTO
FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Privacy-Preserving Synthetic Tabular Data: A Systematic Comparison of Model-Agnostic Perturbation Strategies

André Filipe Santos Sousa





Mestrado em Inteligência Artificial

Supervisor: Prof. Moisés Santos

Second Supervisor: Aníbal Silva

September 25th, 2026

© André Sousa, 2026

Privacy-Preserving Synthetic Tabular Data: A Systematic Comparison of Model-Agnostic Perturbation Strategies

André Filipe Santos Sousa

Mestrado em Inteligência Artificial

Approved in oral examination by the committee:

President: Prof. Henrique Lopes Cardoso
Examiner: Prof. Nuno Moniz
Supervisor: Prof. Moisés Santos

September 25th, 2026

Resumo

Os modelos generativos têm vindo a afirmar-se como uma solução promissora para a criação de dados sintéticos em saúde, permitindo a partilha e análise de informação sensível sem recorrer a dados pessoais e confidenciais. No entanto, a capacidade destes modelos para reproduzirem fielmente a distribuição dos dados de treino levanta preocupações de privacidade, uma vez que pode resultar na memorização e divulgação de informação relativa a exemplos específicos. Assim, a utilização de dados sintéticos não garante, por si só, a proteção da privacidade, tornando necessária a avaliação de mecanismos que avaliem e limitem este risco. Este trabalho analisa cinco mecanismos de perturbação independentes da arquitetura do modelo generativo utilizado (regularização, dropout, differential privacy, corrupção dos dados e contrastive learning baseado em corrupção) aplicadas sistematicamente sobre um único Variational Autoencoder (VAE) de referência, deliberadamente simples. Esta abordagem permite comparar diretamente o compromisso entre preservação da privacidade, qualidade dos dados sintéticos e utilidade em tarefas posteriores. A avaliação foi realizada em oito conjuntos de dados tabulares reais da área da saúde, com diferentes características. Cada estratégia foi testada sob várias configurações de hiperparâmetros e avaliada através de um protocolo comum baseado em três dimensões complementares: qualidade da distribuição, privacidade e utilidade. Os resultados mostram um padrão consistente: o reforço progressivo das estratégias de perturbação conduz a uma degradação da qualidade distribucional e da utilidade dos dados sintéticos. Entre os métodos avaliados, dropout, a corrupção dos dados e contrastive learning oferecem, de forma consistente, os melhores compromissos entre privacidade e desempenho, enquanto a regularização e differential privacy implicam perdas significativamente superiores. A análise conjunta das métricas indica ainda que os ganhos de privacidade observados estão fortemente associados a uma perda global de fidelidade distribucional, e não a uma redução específica da memorização dos dados de treino. Este padrão já está presente no modelo de base e torna-se mais evidente com a aplicação das diferentes estratégias. No seu conjunto, os resultados evidenciam os compromissos práticos entre as abordagens estudadas e reforçam a dificuldade de distinguir entre uma proteção efetiva da privacidade e um simples desvio da distribuição original dos dados na avaliação de dados tabulares sintéticos.

Palavras-chave: Geração de Dados Tabulares • Aprendizagem Computacional Preservadora da Privacidade • Estratégias para Preservação da Privacidade • Compromisso entre Privacidade e Utilidade • Variational Autoencoders

Abstract

Synthetic data generation is increasingly proposed as a way to share and study sensitive health data without exposing the individuals it describes. However, the same flexibility that lets a generative model closely fit its training data distribution can also cause it to memorize and leak information about specific training examples, undermining the privacy protection synthetic data is meant to provide. This project surveys five conceptually different, model-agnostic perturbation strategies (regularization, dropout, differential privacy, corruption, and contrastive learning via corruption) applied consistently on top of a single, deliberately simple baseline Variational Autoencoder (VAE), to compare how they trade off privacy against generative quality and downstream utility. We implemented each strategy in several hyperparameter configurations and evaluated them across eight real world, health related tabular datasets of varying size, dimensionality, and class balance, using a shared evaluation framework covering three complementary axes: distributional quality, privacy, and downstream utility. Results show a clear, steady cost to distributional quality and utility as each perturbation strategy is applied more aggressively, with dropout, corruption, and contrastive learning consistently offering the best trade-offs, while regularization and differential privacy come with much bigger costs. Notably, differential privacy, despite offering a formal mathematical privacy guarantee, does not empirically outperform the heuristic methods studied. Looking at privacy and distributional metrics together further suggests that the observed privacy gains are closely tied to a general loss of distributional fidelity rather than a targeted reduction in training set memorization, a pattern already present in the unperturbed baseline and backed up by the joint analysis of the used metrics. These findings highlight both the practical trade-offs between the perturbation strategies evaluated and the broader difficulty of clearly separating genuine privacy protection from distributional drift when evaluating synthetic tabular data.

Keywords: Tabular Data Synthesis • Privacy-Preserving Machine Learning • Privacy-Preserving Strategies • Privacy-Utility Trade-off • Variational Autoencoders

The United Nations Sustainable Development Goals

The United Nations provide a global framework for achieving a better and more sustainable future for all. It includes 17 goals (https://sdgs.un.org/goals) to address the world's most pressing challenges, including poverty, inequality, climate change, environmental degradation, peace, and justice. This thesis contributes to the advancement of privacy-preserving Artificial Intelligence in sensitive healthcare contexts, with a direct focus on synthetic data generation, patient data protection, and the responsible sharing of health data for research.

The specific Sustainable Development Goals addressed have the following names:

SDG 3 Ensure healthy lives and promote well-being for all at all ages

SDG 9 Build resilient infrastructure, promote inclusive and sustainable industrialization and foster innovation

SDG 16 Promote peaceful and inclusive societies for sustainable development, provide access to justice for all and build effective, accountable and inclusive institutions at all levels

SDG	Target	Contribution	Performance Indicators and Metrics
3	3.8	Privacy-preserving synthetic data generation can widen access to health datasets for research purposes without exposing the patients they describe, supporting the health research that underpins broader, more equitable access to quality health services.	Number of health-related datasets that can be safely shared for research under the evaluated privacy-preserving strategies.
9	9.5	This project advances scientific understanding of how different privacy-preserving mechanisms affect the quality and utility of synthetic tabular data, contributing methodological knowledge and an empirical comparison framework that can inform future research and technological development in this area.	Number of perturbation strategies systematically compared under a shared evaluation framework.
16	16.10	By evaluating mechanisms that reduce the risk of re-identification and memorization in synthetic health data, this project supports the protection of individuals' fundamental right to privacy when their data is used for research or shared more broadly.	Reduction in privacy-risk metrics achieved by the evaluated perturbation strategies relative to the unperturbed baseline.
	16.b	This project's findings show that the evaluated generative models, including the baseline, consistently underrepresent minority classes and other low-density regions of the data distribution. Surfacing this limitation is a first step towards non-discriminatory synthetic data generation practices, guarding against downstream applications that could otherwise disadvantage underrepresented groups.	Gap in downstream predictive performance attributable to underrepresented classes or low-density regions of the data.

Acknowledgements

The past year was a challenging one for me, both personally and academically. There were moments when the way forward was far from clear and when simply finding the strength to keep going felt like an accomplishment in itself. There were times when I doubted whether I would be able to reach this point.

That I did is thanks in large part to the people who stood by my side throughout, often carrying me when I was struggling to carry myself. Their unwavering support helped me through a period that pushed me beyond what I had thought I was capable of enduring. In facing those challenges, I came to realize that I possessed a resilience and strength far greater than I had ever imagined. From this experience, I take away more than the technical expertise acquired through this dissertation, but also a deeper understanding of myself, a greater appreciation for the people around me, and a renewed perspective about what truly matters.

I cannot properly express how grateful I am to my supervisors, Moisés Santos, Aníbal Silva and Carlos Soares, without whom this thesis simply would not exist. From the very beginning, they offered not only their knowledge and guidance, but also a level of genuine care and understanding that went far beyond what was expected of them. They were there whenever I needed help, and on more than one occasion, they became people I could simply talk to about life outside research. When I struggled, they never made me feel as though I had become a burden or a disappointment. Instead, they met me with patience, understanding, and a constant willingness to help me find my way forward. What I am most grateful for is that they remained present for me even during the times when I was not fully present for my own work. Their continued belief in me, even when my own belief in myself faltered, gave me the space and confidence to keep going. More than that, I truly cannot imagine having reached this point with anyone else beside me. Their advice, compassion, support, and reassurance did not just make the journey easier, they made it possible. I will always carry with me the gratitude of knowing that, at the times I could not find a way forward, they were there to help me find one.

To my mother, thank you for being a constant source of love, patience, and encouragement, and for the silly, ridiculous and wonderfully unserious conversations I can only have with you. Your strength is something I deeply admire, but also something I feel lucky to have witnessed so closely. It has carried us through difficult moments and shown me a kind of perseverance I hope to carry with me. “All That I Am or Hope to Be, I Owe to My Angel mother.” - Abraham Lincoln

To my brother, thank you for the journey we have shared, the late-night conversations about life, and all the laughter and disagreements along the way. And for the way you can be both unbelievably frustrating and deeply lovable, often at the very same time. Thank you for the constant company trough this expedition of life.

To my father, thank you for your advice, the opportunities you gave me and for always pushing me towards a better future, sometimes even harder than I was willing to push myself. I do not always listen and I certainly do not always appreciate it, but with time I have come to appreciate

that your advice comes from love, from wanting the best for me, and from believing I am capable of more.

And to my friends and colleagues, thank you for putting up with me, laughing alongside me, and somehow accepting all the odd little ways that make me who I am. Thank you for staying close through the moments when I was not quite myself, for making the difficult days a little easier, and for sharing the good ones in a way that made them all the more meaningful.

After everything this past year has asked of me, I am still here and this dissertation has been submitted. There is something deeply peaceful about looking back at a period I once feared I would not make it through and knowing that, despite everything, I did. I am not proud of every part of this year, or of everything I managed (and did not manage) to do during it. But I am proud to be here now. Not loudly, and not without reservations, but in a quiet way that feels perhaps more honest, especially knowing how many times I once thought I would not be.

To anyone who finds themselves in the middle of a difficult chapter, do not mistake a difficult journey for a failed one. You are allowed not to have all the answers. You are allowed to have days when you do not feel capable, and days that simply getting through feels like enough. Sometimes all you can do is take the next step, then the one after that, and trust that eventually those small steps will lead you somewhere. Sometimes you will need to slow down, change direction, step aside, or even go backwards for a while. What matters is finding a way to keep moving, even when the path looks different from the one you once imagined. Keep going, even when you cannot yet see the end. One day, you may eventually find yourself looking back at a time you once feared would define you, and accept that it was a necessary part of your story. And when that day finally arrives, I hope you allow yourself a moment to recognize not only where you arrived, but everything it took to get there. And then, surrounded by the people who stood beside you along the way, smile and proudly proclaim: "We did it!".

This work is a result of Agenda “Center for Responsible AI”, nr. C645008882-00000055, investment project nr. 62, financed by the Recovery and Resilience Plan (PRR) and by European Union - NextGeneration EU. Funded by the European Union – NextGenerationEU. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Commission. Neither the European Union nor the European Commission can be held responsible for them.

André Sousa

Declaration of honour

I declare that this dissertation is my own work and has not previously been submitted or assessed at this or any other institution. References to other authors (statements, ideas, thoughts), as well as the use of published works, strictly comply with the rules of attribution and are duly identified in the text and in the bibliographic references, in accordance with the applicable referencing standards. I am aware that the practice of plagiarism or self-plagiarism constitutes an academic offence, as established in the U.Porto Code of Ethics.

I also declare that I have used Artificial Intelligence tools to complete part(s) of this dissertation, and that this use and its results are duly noted and have been carefully checked for errors, inaccuracies, unfounded attributions or other forms of bias in content and arguments, as well as determining authorship.

André Filipe Santos Sousa

"Não estando disposto a esperar que a humanidade venha alguma vez a ser melhor
Jeremias escolheu o seu lugar do lado de fora"

Jorge Palma, *Jeremias, o Fora da Lei*

"In the depths of winter, I finally learned that within me there lay an invincible summer."

Albert Camus, *Return to Tipasa*

"The struggle itself toward the heights is enough to fill a man's heart.
One must imagine Sisyphus happy."

Albert Camus, *The Myth of Sisyphus*

"I am the master of my fate,
I am the captain of my soul."

William Ernest Henley, *Invictus*

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Problem Statement . . . 2
- 1.3 Research Objectives and Questions . . . 3
  - 1.3.1 General Objective . . . 3
  - 1.3.2 Specific Objectives . . . 3
  - 1.3.3 Research Questions . . . 3
- 1.4 Thesis Structure . . . 4

2 Related Work **5**
- 2.1 Generative Models for Tabular Data . . . 5
- 2.2 The Variational Autoencoder as a Generative Backbone . . . 6
- 2.3 Privacy Formalization and Empirical Privacy Metrics . . . 7
  - 2.3.1 Differential Privacy . . . 7
  - 2.3.2 Empirical Privacy Metrics . . . 7
- 2.4 The Privacy-Utility Trade-off: Empirical Evidence . . . 9
- 2.5 Evaluation Frameworks for Synthetic Tabular Data . . . 9
- 2.6 Synthetic Data in Clinical Contexts . . . 11
- 2.7 Positioning of the Present Work . . . 12

3 Methodology **14**
- 3.1 Baseline Variational Autoencoder . . . 14
  - 3.1.1 Core Idea . . . 14
  - 3.1.2 The Variational Objective . . . 15
  - 3.1.3 The Reparameterization Trick . . . 17
  - 3.1.4 Training and Inference . . . 17
  - 3.1.5 Architecture . . . 17
- 3.2 Perturbation Strategies . . . 18
  - 3.2.1 Regularization . . . 18
  - 3.2.2 Dropout . . . 20
  - 3.2.3 Differential Privacy . . . 21
  - 3.2.4 Corruption . . . 25
  - 3.2.5 Contrastive Learning . . . 26

4 Experimental Setup **30**
- 4.1 Datasets . . . 30
- 4.2 Pipeline . . . 32
  - 4.2.1 Preprocessing . . . 32

4.2.2 Training, Generation, and Post-processing . . . 33
4.2.3 Cross Validation . . . 34
4.3 Metrics . . . 34
4.3.1 Distribution Based Metrics . . . 34
4.3.2 Privacy . . . 36
4.3.3 Utility . . . 38
4.4 Hyperparameters . . . 40
4.4.1 Regularization . . . 40
4.4.2 Dropout . . . 40
4.4.3 Differential Privacy . . . 40
4.4.4 Corruption . . . 41
4.4.5 Contrastive Learning . . . 41

5 Results and Discussion **43**
5.1 Individual Metric Results . . . 43
5.1.1 Distribution based Metrics . . . 43
5.1.2 Privacy Metrics . . . 47
5.1.3 Utility Metrics . . . 50
5.1.4 Train and Generation Times . . . 51
5.2 Dual-Metric Results . . . 52
5.2.1 Trade-off between α-Precision and β-Recall . . . 52
5.2.2 Trade-off between α-Precision and Authenticity . . . 53
5.2.3 Trade-off between Machine Learning (ML) Utility and Authenticity . . . 54
5.3 Metafeature Correlation Results . . . 55
5.3.1 Distribution . . . 56
5.3.2 Privacy . . . 57
5.3.3 Utility . . . 58
5.4 Discussion . . . 59

6 Conclusion **63**
6.1 Main Findings . . . 63
6.2 Limitations and Future Work . . . 64
6.2.1 Limitations . . . 64
6.2.2 Future Work . . . 65
6.3 Final Remarks . . . 66

References **68**

A Additional Results **74**
A.1 Single Metric Results . . . 74
A.1.1 Alpha Precision . . . 76
A.1.2 Beta Recall . . . 77
A.1.3 Detection . . . 78
A.1.4 Authenticity . . . 79
A.1.5 Train Adversarial Accuracy . . . 80
A.1.6 Test Adversarial Accuracy . . . 81
A.1.7 Privacy Loss . . . 82
A.1.8 Balanced Accuracy Gap . . . 83
A.1.9 Alignment . . . 84

A.1.10 Train and Generation Times 85
A.2 Dataset Feature Relevance Results 87
A.2.1 Alpha Precision 87
A.2.2 Authenticity 88
A.2.3 Balanced Accuracy Gap 89

List of Figures

3.1 Baseline VAE architecture. 18
3.2 Baseline VAE Model with Regularization 20
3.3 Baseline VAE Model with Dropout 21
3.4 Baseline VAE Model with Differential Privacy 25
3.5 Baseline VAE Model with Contrastive Learning 29

5.1 α-Precision per experiment across datasets. 44
5.2 β-Recall per experiment across datasets. 45
5.3 Detection per experiment across datasets. 46
5.4 Authenticity per experiment across datasets. 47
5.5 Train Adversarial Accuracy per experiment across datasets. 48
5.6 Test Adversarial Accuracy per experiment across datasets. 48
5.7 Privacy Loss per experiment across datasets. 49
5.8 Balanced Accuracy Gap per experiment across datasets. 50
5.9 Alignment per experiment across datasets. 51
5.10 α-Precision and β-Recall per Experiment type across datasets 53
5.11 α-Precision and Authenticity per Experiment type across datasets 54
5.12 Balanced Accuracy Gap and Authenticity per Experiment type across datasets . . 55
5.13 5 most relevant dataset Metafeatures to α-Precision. 56
5.14 5 most relevant dataset Metafeatures to Authenticity. 57
5.15 5 most relevant dataset Metafeatures to Balanced Accuracy Gap. 58

List of Tables

4.1 Dataset general characteristics. 31
4.2 Hyperparameter configurations for the Regularization variants. 40
4.3 Hyperparameter configurations for the Dropout variants. 40
4.4 Hyperparameter configurations for the Differential Privacy variants. 41
4.5 Hyperparameter configurations for the Corruption variants. 41
4.6 Hyperparameter configurations for the Contrastive Learning variants. 42

A.1 Description of each experiment configuration. 75
A.2 Alpha Precision Results of each method on each dataset. 76
A.3 Beta Recall Results of each method on each dataset. 77
A.4 Detection Results of each method on each dataset. 78
A.5 Authenticity Results of each method on each dataset. 79
A.6 Train Adversarial Accuracy Results of each method on each dataset. 80
A.7 Test Adversarial Accuracy Results of each method on each dataset. 81
A.8 Privacy Loss Results of each method on each dataset. 82
A.9 Balanced Accuracy Gap Results of each method on each dataset. 83
A.10 Alignment Results of each method on each dataset. 84
A.11 Train Time of each method on each dataset (in seconds). 85
A.12 Generation Time of each method on each dataset (in seconds). 86
A.13 Dataset Meta-feature Relevance for Performance in Alpha Precision. 87
A.14 Dataset Meta-feature Relevance for Performance in Authenticity. 88
A.15 Dataset Meta-feature Relevance for Performance in Balanced Accuracy Gap. . . 89

List of Acronyms

CDF	Cumulative Distribution Function
DCR	Distance to Closest Record
DP	Differential Privacy
DP-SGD	Differentially Private Stochastic Gradient Descent
ELBO	Evidence Lower Bound
GAN	Generative Adversarial Network
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
KL	Kullback-Leibler
MIA	Membership Inference Attack
ML	Machine Learning
MLP	Multilayer Perceptron
MSE	Mean Squared Error
RQ	Research Question
VAE	Variational Autoencoder

Chapter 1

Introduction

1.1 Motivation

Data driven research has become central to progress in healthcare, from improving diagnostic tools to shaping public health policy. Much of this progress depends on access to large, detailed datasets that describe real patients. At the same time, medical data is one of the most sensitive forms of personal information. It can reveal a person's medical history, genetic predispositions, and other deeply personal details. Misusing or exposing it can cause lasting harm. Legal and ethical frameworks such as General Data Protection Regulation (GDPR) [1] and Health Insurance Portability and Accountability Act (HIPAA) [2] reflect this tension directly, placing strict limits on how personal health data can be collected, stored, and shared, which often restricts how much valuable data can be released for research, benchmarking, or collaboration beyond the institution that originally collected it.

Synthetic data generation has emerged as a promising way to help address this issue [3]. Instead of sharing real patient records directly, we can train a generative model to learn the underlying statistical patterns of a dataset, and use it to produce entirely new, artificial records that resemble the original data's distribution, without corresponding to any real individual. In principle, this lets us share and analyze the useful statistical structure of a sensitive dataset more freely while avoiding the exposure of private information from the specific people whose data was collected and used.

However, this idea is not automatically guaranteed. We typically train generative models to fit the distribution of their training data as closely as possible, and, especially with limited data or insufficiently constrained models, this can lead to models that memorize specific training examples instead of learning only the broader patterns behind them. A generative model that has memorized parts of its training data may produce synthetic records that closely resemble or even reproduce real individuals records, undermining the very privacy protection that motivated the use of synthetic data in the first place [4]. This creates a fundamental tension at the heart of privacy-preserving synthetic data generation. The same flexibility that enables a model to produce high

quality, realistic synthetic data is also what allows it to memorize and leak information about the individuals in its training set.

Researchers have proposed many approaches to reduce this risk, ranging from architectural or training changes meant to discourage memorization, to techniques with formal, mathematical privacy guarantees. However, these approaches are rarely compared under a common baseline model and a shared, thorough evaluation framework covering both privacy and generative quality [5]. This project is motivated by this gap in the literature. It implements and evaluates several conceptually different perturbation strategies on top of a common baseline generative model. These strategies are assessed across a range of real world healthcare related tabular datasets, providing an empirically grounded comparison of the trade-offs between privacy, generative quality, and downstream utility.

1.2 Problem Statement

As explained above, generative models trained on sensitive data face an inherent tension, as the same capacity that lets a model closely fit the distribution of its training data also lets it memorize and leak information about individual training examples. Existing work has explored a wide range of techniques to reduce this risk, spanning fundamentally different mechanisms: constraining model complexity, limiting the information directly seen during training, and formal guarantees of the influence of single examples. However, the literature evaluating these techniques is fragmented. Individual works typically propose and test a single technique, often against a different baseline model, on a different dataset, and using a different, narrow set of evaluation metrics that focus on either privacy or utility, but rarely both together. As a result, it is still hard to answer a seemingly simple and practical question:

Given a sensitive tabular dataset and a standard generative model, which of these mitigation strategies gives the best trade-off between privacy and generative quality?

This thesis addresses this gap by evaluating several conceptually different perturbation strategies, applied consistently on top of a single, common baseline generative model, and assessed under a shared evaluation framework that jointly captures distributional quality, privacy, and downstream utility. Rather than treating each perturbation strategy as an isolated solution, evaluated on its own terms, this project puts them side by side, deliberately controlling for the model architecture, training procedure, and evaluation protocol used. Consequently, any differences we see in performance can be attributed to the methodologies themselves, rather than to inconsistencies in the experimental setup. Furthermore, since we cannot assume privacy-utility trade-offs behave the same way across all datasets, this project evaluates each perturbation strategy across a range of datasets with different characteristics, so we can look not only at which strategies perform best on average, but also in which data conditions may make each strategy more or less effective.

In summary, the core problem this project addresses is the lack of a systematic and controlled comparison between conceptually different privacy-preserving perturbation strategies for the generative modeling of sensitive tabular data, jointly evaluated across the three axes of distributional

quality, privacy, and downstream utility, and checked for consistency across datasets with varying characteristics.

1.3 Research Objectives and Questions

Building on the problem described above, this section turns that problem into a concrete set of objectives for this project, and the specific Research Questions (RQs) these objectives are intended to answer.

1.3.1 General Objective

The general objective of this project is to empirically survey and compare a set of established and conceptually different perturbation strategies, applied on top of a common baseline generative model, in terms of how well they improve the privacy of generated synthetic tabular data while keeping its distributional quality and downstream utility.

1.3.2 Specific Objectives

We pursue this general objective through the following specific objectives:

1. Implement a common baseline Variational Autoencoder and five existing perturbation strategies on top of it (regularization, dropout, differential privacy, corruption, and contrastive learning via corruption) under a shared architecture and training pipeline, so any differences we see in performance can be attributed to the perturbation strategy itself.

2. Build a comprehensive evaluation framework that can jointly assess the distributional quality, privacy, and downstream utility of synthetic data generated by each method.

3. Apply this framework across a range of real world, healthcare related tabular datasets, varying in size, dimensionality, and class balance, to evaluate how consistent each method's behavior is across different data conditions.

4. Characterize the trade-offs introduced by each perturbation strategy between privacy and generative quality/utility, both on its own and compared to the other strategies studied.

5. Conduct an exploratory correlation based analysis of the metafeatures of the synthetic datasets generated by each method, looking within each perturbation strategy at which structural or statistical properties of the generated data are most strongly linked to higher or lower values of the evaluation metrics.

1.3.3 Research Questions

These objectives are guided by the following research questions:

RQ1 What impact does each method have on the distributional quality and downstream utility of the generated synthetic data compared to the unperturbed baseline Variational Autoencoder (VAE)?

RQ2 To what extent does each perturbation strategy improve privacy preservation of the generated synthetic data compared to the unperturbed baseline VAE?

RQ3 What trade-offs between privacy and generative quality/utility emerge across the different methods studied, and which of them offer the best (or worst) such trade-off?

RQ4 How do these methods affect the underlying data generation process, and what patterns across the evaluation metrics reveal about how these approaches work?

RQ5 Within the synthetic data produced by each perturbation strategy, are structural properties consistently associated with better or worse performance on a given evaluation metric, helping explain why certain configurations outperform others?

1.4 Thesis Structure

The rest of this document is organized into five further chapters:

- Chapter 2 - Literature Review places this project within the shared areas, practices, and goals of existing work on privacy-preserving generative modeling and synthetic tabular data, and highlights the contribution of this work to the scientific community.
- Chapter 3 - Methodology presents, from a theoretical standpoint, the baseline generative model used throughout this project and each of the perturbation strategies added on top of it.
- Chapter 4 - Experimental Setup describes how we applied the methodology from Chapter 3 in practice: the datasets used, the full experimental pipeline from raw data to generation and evaluation, the metrics used to compare methods, and the implementation details and hyperparameter configurations for every method variant tested.
- Chapter 5 - Results reports and analyzes the experiment outcomes from three complementary angles: individual metrics, cross-metric trade-offs, and a metafeature analysis.
- Chapter 6 - Conclusion summarizes the main findings of this project, answers the research questions, discusses the limitations of this work, and outlines directions for future work.

Chapter 2

Related Work

The problem of generating synthetic tabular data that is simultaneously useful for machine learning and safe from privacy breaches sits at the intersection of three research streams: the development of generative models for tabular data, the formalization of privacy guarantees and their empirical measurement, and the construction of evaluation frameworks that assess both dimensions jointly. This section reviews each stream critically, identifies the methodological gaps that motivate our work and positions our framework within the existing literature.

2.1 Generative Models for Tabular Data

The generation of tabular data presents distinct challenges compared to image or text generation, given the heterogeneous feature types, non-Gaussian and often multimodal joint distributions, and complex inter-variable dependencies that characterize tabular datasets [6]. The generative paradigms proposed for this task have shifted considerably over time, and tracing this progression briefly clarifies why the field arrived at the architectures reviewed below.

Before deep learning was widely applied to this task, synthetic tabular data was generated mainly through parametric statistical models. Gaussian copulas [7] and Bayesian networks [8] remain relevant for their interpretability and reliability on small samples, but their fixed parametric structure limits their ability to represent complex, high dimensional joint distributions [3]. This limitation motivated a shift to deep generative models. Generative Adversarial Networks (GANs) [9] offered a way to model such distributions implicitly, and medGAN [10] was among the first to apply this idea to healthcare records, combining an autoencoder with adversarial training. Directly applying GANs to tabular data tends to induce mode collapse on imbalanced categorical variables. CTGAN [11] addressed this with mode specific normalization and conditional training, becoming one of the standard baselines in comparative studies [12–14].

Variational Autoencoder based alternatives such as TVAE [11] developed in parallel, trading weaker categorical enforcement for smoother latent interpolation.

The instability of adversarial training also motivated alternative paradigms: Adversarial Random Forests [15] combine tree based density estimation with adversarial training at lower computational cost, GReaT [16] serializes tabular rows as text to leverage pre-trained language models, and normalizing flows [17] trade scalability for exact likelihood computation. Most recently, denoising diffusion models, following their success in image generation, have been adapted to tabular data through models such as TabDDPM [18], and are increasingly explored for privacy oriented data sharing in specific.

Despite this progression, a recurring finding across benchmarks is that no single generator dominates across datasets and metrics [12, 13]. Höllig and Geierhos [12] evaluated twelve synthesizers across seventeen health datasets and found that SMOTE, a considerably simpler neighborhood interpolation technique originally designed for class imbalance [19], achieved the best average privacy-utility trade-off, a counterintuitive result reflecting the sensitivity of utility rankings to evaluation protocol choices rather than any fundamental superiority of the method.

2.2 The Variational Autoencoder as a Generative Backbone

Among the generative paradigms reviewed above, the Variational Autoencoder (VAE) [20] occupies a unique position: rather than competing with more recent GAN, diffusion, or transformer based approaches in terms of generative fidelity, it remains one of the simplest and most thoroughly studied generative architectures, with a well understood probabilistic formulation and training objective. Tabular data specific extensions such as TVAE [11] adapt the VAE with mode specific normalization and additional architectural machinery to better enforce categorical structure.

The VAE's specific vulnerability to memorization and membership inference has also been studied directly. Hilprecht et al. [21] proposed a reconstruction based membership inference attack that specifically targets VAEs, in addition to a model-agnostic attack applicable to generative models more broadly, demonstrating that VAEs can leak detectable information about individual training records through their reconstruction behavior. Chen et al. [22] proposed differentially private VAE based generative models and evaluated their robustness against membership inference and model inversion attacks, while Bernau et al. [23] specifically assessed the privacy-accuracy trade-off of differentially private VAEs under membership inference, finding that favorable trade-offs are rarely observed for this architecture and that the outcome depends strongly on the dataset and mechanism used.

This body of work motivates the choice, in the present project, to evaluate differential privacy and several other perturbation mechanisms on a VAE backbone, precisely because its simplicity and maturity make it a suitable, low complexity baseline on top of which the effect of separate perturbation mechanisms can be isolated, rather than a competitive candidate for state-of-the-art generative fidelity in its own right.

2.3 Privacy Formalization and Empirical Privacy Metrics

2.3.1 Differential Privacy

Differential Privacy (DP) [24] provides the dominant formal framework for privacy-preserving data publication. A randomized mechanism $\mathscr{M}$ satisfies ε-DP if, for any two neighboring datasets D and D' that differ in a single record, and for any set of possible outputs S, the probability that $\mathscr{M}$ produces an output in S differs by at most a factor of e^{ε} between D and D'. The parameter ε quantifies the privacy budget: lower values imply stronger guarantees but require larger perturbation noise, typically degrading data utility [25]. The application of DP to deep generative models, commonly achieved through Differentially Private Stochastic Gradient Descent (DP-SGD) [26], enables the training of neural network synthesizers with provable privacy bounds. This mechanism underpins DPGAN [27] and DTGAN [28], which add calibrated noise to gradient updates during adversarial training.

An alternative approach is the Private Aggregation of Teacher Ensembles (PATE) framework [29], implemented for tabular data as PATE-GAN. Rather than perturbing gradients, PATE partitions the training data among an ensemble of teacher discriminators whose aggregated votes, themselves privatized via the Laplace mechanism, supervise a student generator. This architectural separation reduces the cumulative privacy cost across training steps, enabling better utility at equivalent privacy budgets compared to DP-SGD based approaches [14]. Differentially private Bayesian networks, implemented in PrivBayes [8] and MC-GEN [30], add noise directly to marginal distributions, a strategy that can preserve structural properties of the data more faithfully than end to end gradient perturbation at small sample sizes [31].

A fundamental tension inherent to DP based synthesis is that the noise required to mask individual contributions inevitably distorts the marginal and joint distributions that downstream models depend on. Empirical studies consistently confirm that differentially private synthesizers suffer from substantially lower utility than their non private counterparts, particularly for datasets with complex multimodal distributions [14, 32]. Rosenblatt et al. [33] introduced the concept of *epistemic parity* - the degree to which analytical conclusions replicate across real and synthetic datasets - as a utility metric aligned with the scientific use case of synthetic clinical data, and demonstrated that even modest privacy budgets ($\varepsilon = 1$) can substantially compromise epistemic parity in health research contexts.

2.3.2 Empirical Privacy Metrics

The gap between formal DP guarantees and the empirical privacy of released datasets has motivated a parallel literature on empirical privacy assessment. The core insight is that DP bounds worst-case information leakage under adversarial composition theorems, but it does not directly characterize the probability that a particular released record can be linked to a specific real world

individual. Two families of empirical metrics have emerged: distance based metrics, which quantify how geometrically close synthetic records lie to training records, and attack based metrics, which simulate an adversary directly.

Distance based metrics quantify the geometric proximity of synthetic records to training records in feature space. The Distance to Closest Record (DCR) measures, for each synthetic sample, its Euclidean distance to the nearest real training point, with higher DCR indicating greater empirical separation [12]. The Nearest Neighbor Distance Ratio (NNDR) complements DCR by computing the ratio of the first to second nearest neighbor distances, penalizing synthetic records that are substantially closer to one real point than to any other, a signature of partial memorization [34]. These metrics are computationally tractable and domain-agnostic, but two limitations restrict their use as standalone privacy signals. First, their reliance on Euclidean distance in the original feature space renders them insensitive to semantic similarity in categorical variables and to complex non linear dependencies [3]. Second, and more fundamentally, raw geometric separation conflates privacy with poor fidelity. A generator that fails to reproduce the real distribution trivially achieves high DCR simply by being far from everything, appearing private while offering little utility.

This tension motivated reformulations of nearest neighbor distance that target memorization directly rather than raw separation. Yale et al. [35] proposed a nearest neighbor adversarial accuracy framework and defined Privacy Loss as the gap between adversarial accuracy computed against a held-out test set and against the training set, isolating memorization from resemblance to the underlying distribution. Alaa et al. [36] similarly observed that fidelity metrics can be trivially maximized by near copies of training records, and introduced generalization as an independent evaluation axis via the Authenticity score, a hypothesis test flagging synthetic samples that lie closer to a training record than any other real record does.

A second family of metrics, Membership Inference Attacks (MIAs), models a more realistic adversarial scenario directly: given a synthetic dataset and partial knowledge of the real distribution, can an attacker determine whether a specific individual's record contributed to training? Shokri et al. [37] formalized this attack for discriminative models, with its extension to generative models systematized by Chen et al. [38]. In the tabular synthetic data literature, MIA is typically implemented as a distance based classifier, with success measured by AUC-ROC [14, 31]. Zhang et al. [39] showed that MIA vulnerability in synthetic health data is highly sensitive to sample size and feature dimensionality, with small clinical datasets exhibiting critical leakage risk even from non private generators.

More broadly, Stadler et al. [40] showed that synthetic data offers substantially weaker privacy protection than commonly assumed, with privacy gains under realistic adversarial scenarios often indistinguishable from simple anonymization baselines. Rocher et al. [41] provided complementary grounding by demonstrating that re-identification risk in incomplete datasets can be estimated from generative models, challenging the assumption that incompleteness alone offers meaningful protection. These findings motivate rigorous, multi-metric privacy evaluation as a non negotiable component of any benchmark for synthetic health related tabular data.

2.4 The Privacy-Utility Trade-off: Empirical Evidence

The privacy-utility trade-off in synthetic tabular data has been studied empirically across several dimensions. Tao et al. [32] conducted the first large scale benchmark of differentially private tabular synthesizers, evaluating PrivBayes, DP-CTGAN, and related methods across multiple datasets and privacy budgets, establishing that the trade-off is systematic and that no single method dominates across all utility metrics and privacy levels. Dankar et al. [13] extended this to non private generators, showing that utility rankings vary substantially depending on whether broad distributional metrics (propensity score, KL divergence) or narrow task specific metrics (ML classification performance) are used. A finding that underscores the insufficiency of single-metric evaluation.

Höllig and Geierhos [12] formalized the trade-off through a privacy-utility score (UPS) combining a machine learning utility score with a membership inference protection score, providing the most comprehensive comparative evaluation to date for tabular health data. Their central finding, a Pearson correlation of $r = -0.53$ ($p < 10^{-15}$) between utility and privacy scores across all synthesizer-dataset combinations, quantifies the trade-off as moderate to strong and statistically robust. However, their privacy metric relies exclusively on cosine similarity based disclosure risk, which, as discussed above, may not capture MIA-type vulnerabilities.

Souza et al. [31] provided a complementary multidimensional framework integrating distributional fidelity metrics (KL divergence, Jensen-Shannon divergence, Hellinger distance, Wasserstein distance), multivariate structure metrics (MMD, pairwise correlation difference, energy distance), statistical hypothesis tests (Kolmogorov-Smirnov, Mann-Whitney), and dual privacy metrics (MIA and NNDR), applied to a small clinical dataset ($N = 48$). Their central finding, that no universally superior generator exists and that the trilemma of fidelity, structural preservation, and privacy is dataset dependent, is consistent with the broader literature and reinforces the necessity of comprehensive evaluation frameworks. Bayesian network based models emerged as offering the most stable trade-off profile in small sample clinical settings, a result attributed to their structural inductive bias and lower propensity for memorization compared to neural approaches [42].

Park and Ko [43] investigated the dependence of the trade-off on dataset properties, sample size, feature dimensionality, class cardinality, and data type composition, finding that feature and class size have the most substantial impact on synthesizer performance. This dependence on dataset characteristics implies that benchmark conclusions are not transferable across domains without re-evaluation, a point particularly relevant for clinical data, where sample sizes are often constrained by ethical and logistical factors [44, 45].

2.5 Evaluation Frameworks for Synthetic Tabular Data

Evaluation of synthetic tabular data is commonly organized around two complementary dimensions: distribution fidelity and utility. This perspective closely aligns with the distinction introduced by Snoke et al. [46] between *general utility*, which measures the statistical resemblance

between real and synthetic data, and *specific utility*, which evaluates whether analyses performed on synthetic data yield conclusions comparable to those obtained from the original dataset. This distinction is practically important, as a synthesizer may accurately reproduce marginal distributions while failing to preserve multivariate relationships required for downstream machine learning tasks [47].

More recent work has attempted to combine multiple evaluation dimensions into unified metrics. Chundawat et al. [48] proposed TabSynDex, which aggregates statistical similarity, preservation of feature relationships, machine learning efficacy, and distinguishability into a single score. Although composite metrics simplify model comparison, they may obscure specific failure modes by masking poor performance in individual dimensions [3]. Consequently, recent surveys advocate reporting complementary evaluation metrics rather than relying solely on aggregate scores [49, 50]. The framework adopted in this work therefore evaluates synthetic data across separate fidelity and utility dimensions, enabling a more detailed assessment of synthesizer performance.

Distribution Fidelity

Distribution fidelity evaluates how closely the synthetic data reproduces the underlying probability distribution of the original dataset. Traditional measures include comparisons of marginal distributions using the Kolmogorov-Smirnov statistic, Wasserstein distance, and correlation preservation. More recently, sample level fidelity metrics have been introduced to provide a more comprehensive assessment.

Alaa et al. [36] proposed the α-Precision and β-Recall framework for evaluating generative models. α-Precision quantifies the proportion of synthetic samples that fall within the high density regions of the real data distribution, measuring the realism of generated samples. Conversely, β-Recall measures the extent to which the support of the real data distribution is covered by the synthetic data, thereby assessing distributional coverage. Together, these metrics distinguish generators that produce realistic samples from those that adequately capture the diversity of the original data.

Utility

While fidelity measures statistical similarity, utility evaluates whether synthetic data can effectively replace real data in downstream tasks.

One widely used measure is the difference in predictive performance between a model trained on real data and an equivalent model trained on synthetic data. Although accuracy is commonly reported, any appropriate predictive metric, such as F1-score, AUROC, or mean squared error, may be used depending on the task [51].

Beyond predictive performance, recent work has proposed measuring alignment (also referred to as the agreement rate) between models trained on real and synthetic data. Rather than evaluating correctness, alignment measures the proportion of instances for which both models produce the same prediction, thereby quantifying the similarity of the learned decision boundaries [52]. High

alignment suggests that the synthetic data induces comparable predictive behavior even when absolute predictive performance differs.

Together, distribution fidelity and utility provide complementary perspectives on synthetic data quality. High fidelity does not necessarily imply high utility, nor does strong downstream performance guarantee accurate representation of the underlying data distribution. Consequently, evaluating both dimensions independently provides a more comprehensive assessment of synthetic tabular data than any single metric or composite score alone [3, 48].

2.6 Synthetic Data in Clinical Contexts

The clinical domain presents specific challenges for synthetic data generation that compound the general privacy-utility trade-off. Small sample sizes are endemic in clinical research [44], reducing the generative model's ability to learn the underlying distribution and increasing the risk of memorization. High feature dimensionality relative to sample size, a common characteristic of electronic health records, exacerbates this risk for neural generators, which require substantial data to achieve stable convergence [31]. Class imbalance in clinical outcomes (*e.g.*, rare adverse events, specific diagnoses) further complicates generation, requiring conditional sampling strategies to avoid systematic under representation of minority classes [11]. The coexistence of continuous physiological measurements, binary indicators, ordinal severity scales, and nominal categorical codes within a single record demands generators capable of handling mixed type joint distributions [6].

Several domain specific evaluations have been conducted. Goncalves et al. [53] evaluated multiple synthesizers on synthetic patient datasets, finding that the statistical properties of synthetic clinical records vary substantially depending on both the generator and the downstream analytical task. Kaur et al. [42] demonstrated that Bayesian networks generate synthetic health data with controlled privacy risk and adequate structural fidelity, attributing their favorable performance in clinical contexts to their capacity to encode domain knowledge through directed graph structures. Murtaza et al. [54] reviewed the state of the art specifically in healthcare, identifying the privacy-utility trade-off as the central unresolved challenge and noting that formal DP guarantees, while theoretically rigorous, are routinely insufficient to prevent empirical re-identification attacks in practice [40, 41]. Rujas et al. [55] extended this review to a broader scoping of synthetic data generation motivations and applications in healthcare, confirming that evaluation methodology remains heterogeneous across studies and that integrated privacy-utility benchmarks are absent from the clinical synthetic data literature.

Legal and ethical frameworks reinforce the urgency of rigorous evaluation. The GDPR and the HIPAA impose strict constraints on the sharing of identifiable health information, while simultaneously recognizing data synthesis as a permissible mechanism for data sharing under appropriate safeguards [56]. However, the legal status of synthetic data as non personal data remains contested: if a synthetic record can be linked to a real individual with non trivial probability, a risk quantified but not eliminated by DP, it may retain personal data status under GDPR Article 4. This

legal ambiguity further motivates the development of principled evaluation frameworks that go beyond formal DP guarantees to assess empirical re-identification risk under realistic adversarial models.

2.7 Positioning of the Present Work

The literature reviewed above reveals several recurring limitations.

First, generative models for tabular data are diverse and domain sensitive, and comparative studies consistently find that no single architecture dominates across datasets and metrics [3, 12, 13].

Second, privacy evaluation is routinely fragmented: studies use either distance based metrics or MIA based metrics, but rarely both, despite evidence that these capture distinct and non redundant aspects of privacy risk [31, 40].

Third, the privacy-utility trade-off is empirically confirmed as a systematic, strong negative correlation, but its characterization relies heavily on the choice of metrics and datasets, making results from individual studies hard to generalize. [12, 32].

Fourth, the clinical domain compounds these challenges through small sample sizes, heterogeneous feature types, class imbalance, and stringent regulatory requirements.

A more specific gap follows from the first point above because comparative studies in this literature typically vary the generator architecture itself. When comparing different generative paradigms or different formal privacy mechanisms implemented on top of specific architectures, the effect of a given privacy-preserving mechanism is rarely isolated from the architecture on which it happens to be evaluated. This makes it difficult to attribute observed privacy-utility outcomes specifically to the mechanism under study, since the choice of architecture is itself a major factor that varies alongside it.

The present work addresses this narrower gap. Rather than comparing generator architectures, we hold a single, deliberately simple generative backbone fixed (a plain Variational Autoencoder) and systematically compare five conceptually distinct, model-agnostic perturbation strategies, regularization, dropout, differential privacy, data corruption, and contrastive learning, applied consistently on top of it. Our contributions are threefold:

1. We isolate the effect of each perturbation strategy from the choice of generative architecture, by evaluating all of them under a common, fixed baseline model and training pipeline.

2. We integrate distributional fidelity, privacy (via both distance based and adversarial accuracy based metrics), and downstream machine learning utility into a single, shared evaluation protocol, applied consistently across all methods and datasets.

3. We characterize the resulting privacy-utility trade-offs across the evaluated methods and further examine which structural properties of the generated data are associated with stronger or weaker performance on each metric.

Because the perturbation strategies studied are, in principle, model-agnostic, this comparative framework is not limited to the specific VAE backbone adopted here and can readily be applied to other generative models.

Chapter 3

Methodology

This chapter describes the modeling approaches investigated in this project. The main goal of this project is to explore if perturbations applied to a standard generative model can improve privacy, while having little negative impact on generative quality and data utility. To do this, we use the Variational Autoencoder (VAE) as our baseline model. We chose the VAE because it is a simple and recognized architecture in the generative modeling literature, and its behavior is well understood and widely documented. This choice is deliberate. The focus of this thesis is not to advance model architecture or focus on model complexity, but to isolate and evaluate the effects of the perturbations themselves. Using a simple, well established model as the baseline ensures that the perturbation strategy remains the primary variable under investigation.

On top of this baseline, we study five perturbation strategies, namely regularization, dropout, differential privacy, input corruption, and contrastive learning via corruption. We chose these because they represent clearly different mechanisms for changing the training process, some act directly on the loss function, some act on the network architecture, some act on the training data itself, and some add formal guarantees about what the model can leak about individual training examples. Despite their differences, all five share a common motivation to this project as each is a candidate mechanism for improving the privacy properties of the trained model by trying to limit model access and therefore memorization, and each is evaluated on how well it does this while keeping the quality and utility of the generated data.

The remainder of this chapter first introduces the baseline VAE architecture and objective and presents each of the perturbation strategies, describing their underlying principle, how they change or extend the baseline objective, and the intuition behind their inclusion in this project.

Implementation details and hyperparameters are described in detail in the next chapter.

3.1 Baseline Variational Autoencoder

3.1.1 Core Idea

The Variational Autoencoder (VAE), introduced by Kingma and Welling [20], is a latent-variable generative model that learns a probabilistic representation of data. Unlike a standard

autoencoder, which learns a deterministic mapping from an input to a latent representation and back, a VAE models the latent representation as a probability distribution. This probabilistic formulation enables the generation of new data by sampling latent variables from a prior distribution and decoding them into the data space.

Formally, the VAE assumes that each observation x is generated from an unobserved latent variable z. The generative process is defined as

$$z \sim p(z) = \mathcal{N}(0, I), \tag{3.1}$$
$$x \sim p_\theta(x \mid z), \tag{3.2}$$

where $p(z)$ is the prior distribution over the latent variables and $p_\theta(x \mid z)$ is the decoder, parameterized by the neural network weights θ. Equivalently, the joint distribution factorizes as

$$p_\theta(x, z) = p(z) p_\theta(x \mid z). \tag{3.3}$$

The objective is to learn the decoder parameters θ such that this probabilistic model approximates the true data distribution as closely as possible.

3.1.2 The Variational Objective

Ideally, we would learn the model by maximizing the log-likelihood of the observed data,

$$\log p_\theta(x) = \log \int p_\theta(x, z)\, dz. \tag{3.4}$$

However, evaluating this integral requires marginalizing over the latent variables which is generally intractable.

To address this, the VAE introduces an approximate posterior distribution

$$q_\phi(z \mid x) = \mathcal{N}\left(\mu_\phi(x), \sigma^2_\phi(x)\right), \tag{3.5}$$

parameterized by a second neural network, called the encoder, with parameters ϕ. Rather than maximizing the intractable log-likelihood directly, training maximizes the Evidence Lower Bound (Evidence Lower Bound (ELBO)):

$$\mathcal{L}_{\text{ELBO}}(\theta, \phi; x) = \mathbb{E}_{q_\phi(z|x)}\left[\log p_\theta(x \mid z)\right] - D_{\text{KL}}\left(q_\phi(z \mid x) \,\|\, p(z)\right). \tag{3.6}$$

The ELBO naturally balances two competing objectives:

- **Reconstruction term** $\mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x \mid z)]$, which encourages the decoder to reconstruct the input accurately from its latent representation.

- **Kullback-Leibler (KL) divergence term** $D_{\mathrm{KL}}(q_\phi(z \mid x) \,\|\, p(z))$, which regularizes the approximate posterior towards the prior distribution, encouraging a smooth latent space that can be meaningfully sampled during generation.

The encoder therefore outputs the parameters μ and σ defining the approximate posterior distribution, rather than a single deterministic latent vector.

Reconstruction Loss

In theory, the reconstruction term corresponds to the conditional log-likelihood $\log p_\theta(x \mid z)$. In practice, the datasets used in this thesis contain both continuous and categorical features, requiring different likelihood models for each feature type.

Continuous features are reconstructed using Mean Squared Error (MSE), corresponding to a Gaussian reconstruction model with fixed variance, while one-hot encoded categorical features are reconstructed using cross-entropy, corresponding to a categorical likelihood.

The reconstruction loss used during training combines MSE for numerical features and cross-entropy for categorical features:

$$\mathscr{L}_{\mathrm{recon}}(x,\hat{x}) = \frac{\sum_{j \in \mathrm{numeric}} \mathrm{MSE}(x_j,\hat{x}_j) + \sum_{k \in \mathrm{categorical}} \mathrm{CE}(x_k,\hat{x}_k)}{n_{\mathrm{cols}}}, \tag{3.7}$$

where n_{cols} denotes the number of features in the original input space.

KL Divergence

Since the approximate posterior is modeled as a diagonal Gaussian and the prior is a standard Gaussian,

$$q_\phi(z \mid x) = \mathscr{N}(\mu, \mathrm{diag}(\sigma^2)), \qquad p(z) = \mathscr{N}(0, I),$$

the KL divergence admits the following closed form expression:

$$D_{\mathrm{KL}}\left(q_\phi(z \mid x) \,\|\, p(z)\right) = -\frac{1}{2d_z} \sum_{i=1}^{d_z} \left[1 + \log \sigma_i^2 - \mu_i^2 - \sigma_i^2\right], \tag{3.8}$$

where d_z denotes the dimensionality of the latent space.

Training Objective

Rather than maximizing the ELBO directly, the implementation minimizes its negative. The baseline loss function used throughout this thesis is therefore

$$\mathscr{L}_{\mathrm{VAE}} = \mathscr{L}_{\mathrm{recon}} + D_{\mathrm{KL}}\left(q_\phi(z \mid x) \,\|\, p(z)\right), \tag{3.9}$$

which corresponds to the negative ELBO and is minimized using gradient descent. This objective serves as the reference point for all perturbation strategies introduced later in this chapter. Each perturbation modifies either this objective, the network architecture, the optimization procedure, or the training data, while preserving the underlying VAE framework.

3.1.3 The Reparameterization Trick

Because the latent variable z is sampled from a distribution whose parameters (μ and σ) are themselves outputs of the encoder, directly sampling from $q_\phi(z \mid x)$ would prevent gradients from propagating through the sampling operation. The VAE overcomes this using the reparameterization trick, which expresses sampling as

$$z = \mu + \sigma \odot \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, I), \tag{3.10}$$

where the randomness is isolated in the auxiliary variable ε. This reformulation makes z a differentiable function of the encoder outputs, allowing the model to be trained end-to-end using standard gradient based optimization.

3.1.4 Training and Inference

During training, an input x is passed through the encoder to obtain the parameters μ and σ of the approximate posterior. The latent representation z is then sampled using the reparameterization trick and decoded into a reconstruction $\hat{x}$. The encoder and decoder parameters are jointly optimized by minimizing the baseline loss in Equation 3.9.

At inference time, the encoder is no longer required. Instead, a latent vector is sampled directly from the prior distribution:

$$z \sim \mathcal{N}(0, I),$$

and passed through the trained decoder to generate a new synthetic sample. Since training encourages the approximate posterior to match this prior, latent vectors sampled in this way lie in regions of the latent space that the decoder has learned to model, allowing it to generate coherent synthetic data.

3.1.5 Architecture

Figure 3.1 illustrates the baseline VAE architecture used throughout this project. Both the encoder and decoder are implemented as simple Multilayer Perceptrons (MLPs) consisting of two fully connected layers with 128 neurons each and ReLU activations.

The encoder maps the input space of dimensionality d_{in} to the parameters of a 32-dimensional latent distribution,

$$d_{\text{in}} \to 128 \to 128 \to (\mu, \log \sigma^2),$$

while the decoder mirrors this structure,

$$32 \to 128 \to 128 \to d_{\text{in}},$$

producing the reconstructed sample $\hat{x}$.

We deliberately chose this relatively shallow and intentionally simple architecture so that any observed changes in generative quality, utility, or privacy can be primarily attributed to the perturbation strategies studied, rather than to architectural complexity.

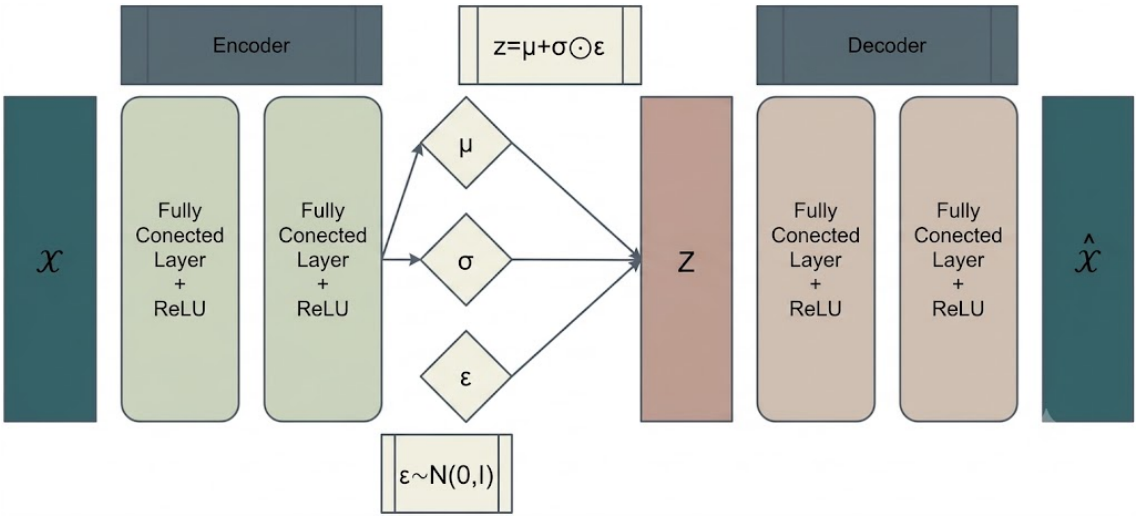


Figure 3.1: Baseline VAE architecture.

3.2 Perturbation Strategies

3.2.1 Regularization

Motivation and Core Idea

Regularization is a family of techniques used in neural network training to limit model complexity and stop the network from fitting the training data too closely [57]. Without these constraints, a sufficiently expressive network can start to memorize the specific details of individual training examples instead of learning patterns that generalize to unseen data. This is called overfitting, and it is undesirable for two reasons. It hinders performance on new, previously unseen data, and is closely linked to memorization. A model that fits training examples too precisely is more likely to encode identifiable information about them, which has direct consequences for privacy.

A common way to regularize a model is to add a penalty term to the training objective that discourages large or overly complex model parameters. Instead of optimizing the original loss function alone, the network is trained to minimize a modified objective:

$$\mathscr{L}_{\text{reg}}(\theta,\phi) = \mathscr{L}_{\text{VAE}}(\theta,\phi) + \lambda\, R(\theta,\phi) \tag{3.11}$$

where $\mathscr{L}_{\text{VAE}}(\theta,\phi)$ is the original task loss, $R(\theta,\phi)$ is a regularization term computed over the model's parameters θ and ϕ, and λ is a hyperparameter controlling how strong the penalty is. By penalizing large parameter values, the network is pushed to find simpler solutions that do not rely as heavily on any single weight. This tends to produce smoother, more generalizable functions and reduces the model's ability to memorize specific training examples.

L1 and L2 Regularization

The two most commonly used forms of regularization are L1 and L2 regularization, which differ in how the penalty term $R(\theta,\phi)$ is defined [58].

L1 regularization penalizes the sum of the absolute values of the weights:

$$R_{L1}(\theta,\phi) = \sum_i |\theta_i| + \sum_i |\phi_i| \tag{3.12}$$

Since this penalty grows linearly instead of quadratically, it applies constant pressure toward zero no matter how large a weight currently is. This tends to push some weights to exactly zero during training, resulting in sparse models where only some parameters stay active. In effect, L1 regularization can be seen as a form of implicit feature or connection selection within the network.

Often referred to as weight decay, L2 regularization penalizes the sum of the squared values of the model's weights:

$$R_{L2}(\theta,\phi) = \sum_i \theta_i^2 + \sum_i \phi_i^2 \tag{3.13}$$

This penalty grows quadratically with the size of each weight, so it pushes harder against large weights while having a much smaller effect on weights already close to zero. In practice, L2 regularization tends to shrink weights smoothly toward zero without necessarily making them exactly zero, resulting in models with generally small, spread-out weight values rather than sparse ones.

Both forms of regularization work on the same basic idea, limiting the size of the model's parameters, but they favor different kinds of outcomes. L2 encourages small, evenly spread-out weights, while L1 encourages sparsity.

Application to the Baseline VAE

In this project (as seen in Figure 3.2), we apply L1 and L2 regularization directly on top of the baseline VAE's training objective. We add the corresponding penalty term to the baseline loss, computed over the parameters of the encoder and decoder networks, and scaled by a regularization strength λ.

We make no architectural changes to the baseline model. Regularization is added purely as a modification to the training objective. The reasoning behind including it in this study is simple: by discouraging the network from relying too heavily on any single weight, regularization is expected

to reduce how much the model memorizes specific training examples, potentially improving privacy with (hopefully) reduced impact on generative quality and utility.

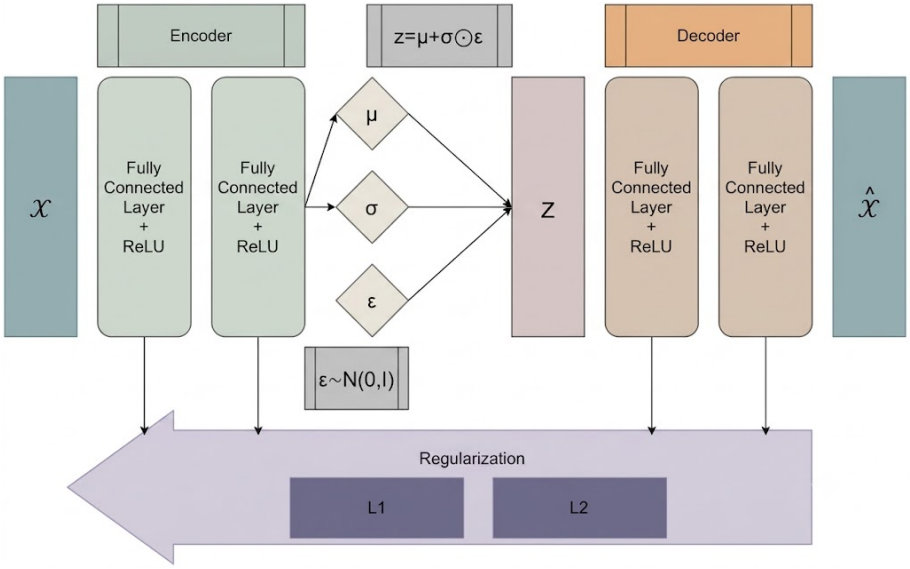


Figure 3.2: Baseline VAE Model with Regularization

3.2.2 Dropout

Motivation and Core Idea

Dropout, introduced by Srivastava et al. [59], is a technique that works directly on a neural network's architecture during training, rather than on its loss function. The core idea is to randomly and independently switch off a subset of neurons in a given layer at each training step, forcing the network to learn representations that do not depend on any specific set of units being present.

Concretely, during training, each neuron in a dropout-enabled layer is kept with probability p and dropped (set to zero) with probability $1-p$, independently for every training example and every forward pass. We can express this as multiplying the layer's output by a binary mask m sampled from a Bernoulli distribution:

$$h' = h \odot m, \qquad m_i \sim \text{Bernoulli}(p) \tag{3.14}$$

where h is the layer's original output and h' is the resulting output actually passed forward through the network. Because a different random mask is sampled at every step, the network cannot rely on any fixed combination of neurons to solve the task, and is instead pushed to learn redundant, distributed representations where multiple units carry overlapping, complementary information [60].

This can be naturally understood as an implicit form of model ensembling since a different subset of neurons is active on every forward pass, training with dropout can be seen as training many thinned sub-networks at the same time, all sharing parameters, with the final trained model

approximating an average over this ensemble. This tends to make the network less dependent on the fine grained co-adaptation of specific neurons to specific training examples, which in turn discourages overfitting and reduces the likelihood of a model memorizing individual training examples.

During training, all neurons are scaled by $\frac{1}{p}$. At inference time, dropout is turned off: all neurons are used without the scaling term to keep the expected size of activations consistent between training and inference.

Application to the Baseline VAE

In this project, we apply dropout architecturally, by inserting a dropout layer between the two fully connected layers of both the encoder and the decoder of the baseline VAE. We make no changes to the training objective itself. Instead, the perturbation acts on how information flows through the network. During training, activations of the first fully connected layer in the encoder are randomly masked before being passed to the second, and the same happens in the decoder, as depicted in Figure 3.3.

The reasoning for including dropout in this study is closely related to that of regularization as both aim to stop the network from overfitting to and memorizing specific training examples, but they do this through different mechanisms. Regularization directly constrains the size of the model's weights, while dropout prevents the network from depending too heavily on specific neurons or their co-adaptation during training. Including both lets us directly compare the effect of these two different mechanisms on privacy and generative quality.

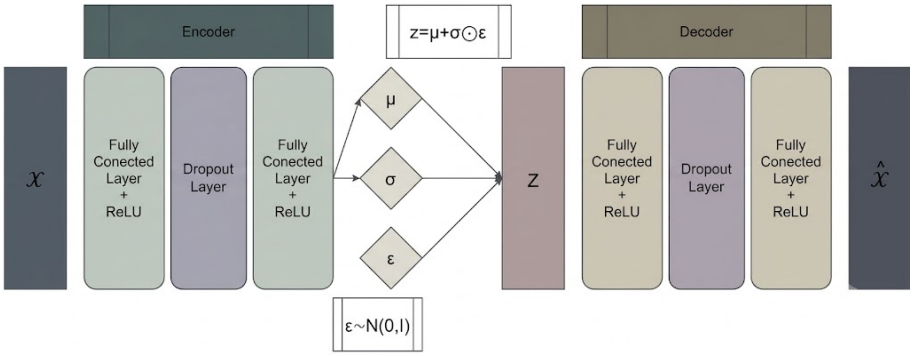


Figure 3.3: Baseline VAE Model with Dropout

3.2.3 Differential Privacy

Motivation and Core Idea

Of the perturbation strategies used in this project, differential privacy (DP) is the only one that gives a formal, mathematical guarantee of privacy, rather than an empirical or heuristic expectation that memorization is reduced. Regularization and dropout are based on the intuition that limiting model complexity should make it less likely for individual training examples to be memorized.

Differential privacy, on the other hand, precisely defines what it means for an algorithm to protect the privacy of the individuals in its training data [61], and gives mechanisms that meet this definition. This makes it the natural reference point for comparing the other, more heuristic approaches in this study.

Informally, an algorithm is differentially private if its output does not depend too heavily on any single training example: someone looking at the algorithm's output should not be able to confidently tell whether a particular example was in the training set at all. This is a strong, deliberately worst-case notion of privacy. It is not about what a specific attacker might be able to figure out in practice, but about a guarantee that holds no matter what background knowledge or computing power the attacker has.

Formal Definition

Formally, differential privacy is defined over pairs of neighboring datasets, datasets D and D' that differ in exactly one record. A randomized algorithm $\mathscr{M}$ satisfies (ε, δ)-differential privacy if, for all such neighboring datasets D and D', and for every possible set of outcomes S in the output space of $\mathscr{M}$:

$$\Pr[\mathscr{M}(D) \in S] \leq e^{\varepsilon} \Pr[\mathscr{M}(D') \in S] + \delta \qquad [24, 61]$$

This states that the presence or absence of any single record in the dataset can change the probability of any particular outcome by at most a bounded factor. Concretely, if we compare the algorithm's behavior on D and on the neighboring dataset D', the probability of seeing any given outcome (or set of outcomes) S under D is at most e^{ε} times the corresponding probability under D', plus a small additive term δ.

The two parameters in this definition control how strong the privacy guarantee is:

- ε (the privacy budget) limits how much the probability of any outcome can differ, whether or not a given individual's data was included. Smaller values of ε mean stronger privacy guarantees, since they force the algorithm's behavior to change less depending on whether one record is present or not. Larger values allow more dependence on individual records, and so give weaker privacy.

- δ is the probability that the pure multiplicative (ε-only) bound is allowed to fail entirely. With probability at most δ, the algorithm's output can fall outside the e^{ε} guarantee altogether, so δ should be interpreted as a small budget for privacy failure rather than as ordinary, graceful leakage. In practice δ is set very small precisely because pure ε-differential privacy ($\delta = 0$) is difficult to achieve for many practical mechanisms.

Because this guarantee must hold for *any* pair of neighboring datasets, it protects every individual in the dataset at the same time and equally, no matter how unusual or distinctive their data point is. This is what makes DP much stronger than heuristic notions of privacy. It is a property of the algorithm itself, not of any specific dataset or attack.

Achieving Differential Privacy: Sensitivity and Noise

We typically achieve differential privacy by adding carefully calibrated random noise to the output of a computation, where the amount of noise needed depends on the sensitivity of that computation, that is how much its output can change because of a single record differing between D and D'. Formally, the sensitivity of a function f is defined as:

$$\Delta f = \max_{D,D'} \|f(D) - f(D')\| \tag{3.15}$$

Intuitively, a function with low sensitivity already changes very little when one record is added or removed, so it needs relatively little noise to hide that record's influence. On the other hand, a function with high sensitivity needs more noise to reach the same guarantee. A common mechanism for achieving (ε, δ)-differential privacy is the Gaussian mechanism [24], which outputs $f(D) + \mathcal{N}(0, \sigma^2 I)$ instead of $f(D)$, where the noise scale σ is set based on the sensitivity Δf and the desired (ε, δ) pair.

Another important property of differential privacy is composition. If an algorithm has multiple steps, each individually satisfying some $(\varepsilon_i, \delta_i)$-DP, the overall algorithm satisfies DP with a privacy cost that adds up across steps [61]. This matters a lot for iterative training procedures like neural network training, where we need to add privacy-preserving noise repeatedly, once per training step, and track its combined effect over the whole training process.

Differentially Private Stochastic Gradient Descent (DP-SGD)

To train a neural network under differential privacy, it is not enough to just add noise to the model's final output. The model's parameters come from many iterative gradient updates during training, and each update is computed directly from the training data, so each one could individually leak information about specific examples. DP-SGD [26] deals with this by making the training process itself private, applying the sensitivity and noise idea described above to every gradient update.

At each training step, DP-SGD changes the standard stochastic gradient descent procedure in two key ways:

1. **Per example gradient clipping.** For each individual training example x_i in a mini-batch, we compute its gradient $g_i = \nabla_\theta \mathcal{L}(\theta; x_i)$ separately and clip it so that its L_2 norm does not go above a fixed clipping threshold C:

$$\bar{g}_i = g_i \,/\, \max\left(1, \frac{\|g_i\|_2}{C}\right) \tag{3.16}$$

   This step limits the sensitivity of the gradient computation, since no single example's gradient can contribute more than C to the update. By capping the norm this way, we explicitly limit how much influence any one training example can have on the model, no matter how unusual or extreme that example is.

2. **Noise addition.** We average the clipped per example gradients, and add calibrated Gaussian noise to the resulting sum before using it to update the model parameters:

$$\tilde{g} = \frac{1}{N}\left(\sum_{i=1}^{N} \bar{g}_i + \mathcal{N}(0, \sigma^2 C^2 I)\right) \tag{3.17}$$

where N is the mini-batch size and σ is a noise multiplier that controls the trade-off between privacy and the accuracy of the gradient estimate. A larger σ gives stronger privacy guarantees but adds more noise to training, which comes at some cost to model performance.

We then update the model using $\tilde{g}$ instead of the ordinary mini-batch gradient, and repeat this clip and noise procedure at every training step throughout training. Since each step uses up some of the overall privacy budget, and training involves many such steps, we track the total privacy cost of training.

Given the clipping threshold C, noise multiplier σ, batch size B, dataset size N, and number of steps T, we compute a tight overall (ε, δ) guarantee using the moments accountant [26], which composes the per step privacy cost across all T training iterations.

Application to the Baseline VAE

In this project, we train the baseline VAE entirely using the DP-SGD framework, pictured in Figure 3.4. We fully replace the standard training procedure with the clip and noise procedure described above, applied to the gradients of the ELBO objective (Equation 3.6) with respect to both the encoder parameters ϕ and decoder parameters θ. We make no other changes to the model architecture or objective. Here, the perturbation acts purely on how gradients are computed and applied during training, rather than on the objective itself or the network's structure. In practice, we use the `opacus` library's `PrivacyEngine` [62] to track this automatically and report the final (ε, δ) at the end of training.

As noted above, the reasoning for including differential privacy in this study is different from that of the other methods, as instead of relying on the hope or intuition that constraining the model will happen to reduce memorization, DP-SGD gives an explicit, measurable (ε, δ) privacy guarantee for the trained model, obtained by directly limiting and hiding the influence of every individual training example during training. This allows us compare, both in practice and in terms of the strength of the guarantee, how well this formal approach trades off against the other proposed methods.

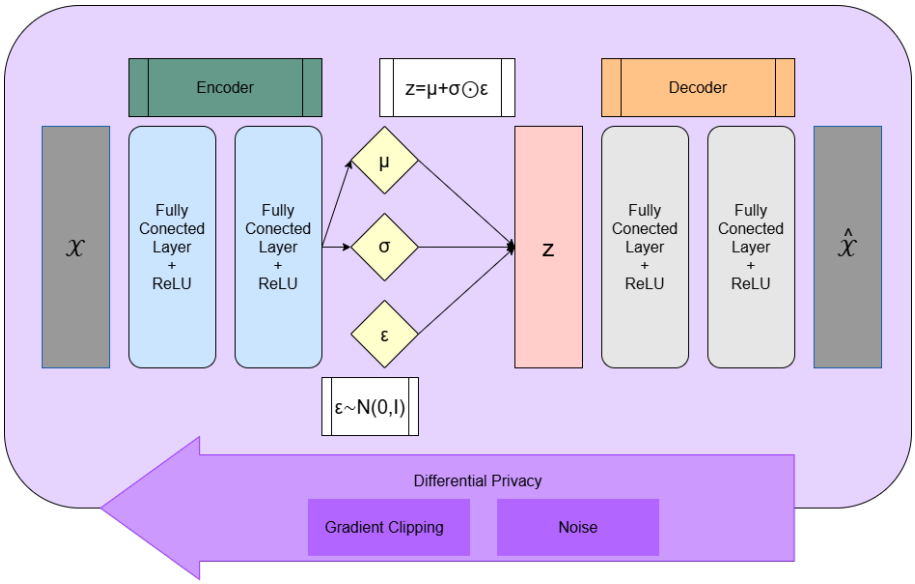


Figure 3.4: Baseline VAE Model with Differential Privacy

3.2.4 Corruption

Motivation and Core Idea

Unlike the previous perturbation strategies, which change either the training objective, the flow of information through the network's architecture, or the optimization procedure itself, corruption acts directly on the input data. The idea behind it is simple, a model cannot memorize what it never fully sees. By deliberately altering part of each training example before showing it to the network, corruption limits how much true, exact information about any single record is available to the model at any given training step, reducing the chance that record gets memorized.

This puts corruption in a similar family to denoising approaches in representation learning, where a model is trained to reconstruct or work from a deliberately degraded version of its input [63]. Here, though, the goal is aimed specifically at privacy rather than representation quality. We do not explicitly want the model to learn to "undo" the corruption, just to make sure no individual training example is ever shown to the model in a fully intact, exact form, limiting how much its specific values can be memorized.

Corruption Mechanism

We apply corruption independently to each training example right before it is passed to the model, based on a corruption ratio $r \in [0,1]$, which sets the proportion of values in the example that get corrupted. Given an input vector $x = (x_1, x_2, \ldots, x_d)$ with d features, we select a random subset of indices $I \subset \{1, \ldots, d\}$, with $|I| = \lfloor r \cdot d \rfloor$, and build the corrupted input $\tilde{x}$ as:

$$\tilde{x}_j = \begin{cases} u_j \sim \mathcal{U}(\min_j, \max_j) & \text{if } j \in I \\ x_j & \text{otherwise} \end{cases} \tag{3.18}$$

That is, for each feature chosen to be corrupted, we throw away its original value entirely and replace it with a sample drawn independently from a uniform distribution over that feature's value range, $\mathcal{U}(\min_j, \max_j)$, where $\min_j$ and $\max_j$ are the minimum and maximum values observed for feature j. This is a fairly aggressive form of corruption, instead of adding noise around the original value, we replace the true value with one that carries no information about it beyond the feature's overall range. So, for any corrupted feature, the model gets no signal linking $\tilde{x}_j$ back to the true x_j.

We resample the set of corrupted indices I independently for every training example. The corruption ratio r is the main control over how strong this perturbation is with higher values of r corrupt a bigger share of each example's features, further reducing the true information available to the model at each step, though this can make training harder, since the model sees less reliable information about the true data distribution.

Application to the Baseline VAE

In this thesis, we apply corruption as a preprocessing step on each training example, before it goes through the encoder of the baseline VAE described previously. Importantly, corruption is applied only to the input the encoder sees. The reconstruction target used in the loss function stays the original, uncorrupted example x. This means the model is trained to reconstruct the true data from a deliberately incomplete, partly randomized view of it. We make no changes to the model's architecture or to the ELBO objective (Equation 3.6).

The reasoning for including corruption in this thesis follows directly from how we implement it. Since a fraction r of each example's values are, at any given training step, replaced with values carrying no information about the true data, we directly limit the model's ability to precisely fit (and therefore memorize) any single training example. This gives us a way to improve privacy that is conceptually different from the previous three.

3.2.5 Contrastive Learning

Motivation and Core Idea

The last perturbation strategy in this project builds directly on the corruption method described above, extending it with an added contrastive learning component. Contrastive learning is a self supervised representation learning approach where a model learns to organize its embedding space based on similarity. Representations of inputs considered related to each other (a positive pair) are pulled closer together in the embedding space, while representations of unrelated inputs (negative pairs) are pushed further apart. Importantly, this notion of "relatedness" does not need explicit labels. Instead, positive pairs are usually built from two different views of the same underlying example, for instance by applying two different augmentations or perturbations to it, on the idea that both views should still be recognized by the model as representing the same underlying data point.

In this project, we use corruption exactly as such a perturbation, the corrupted version of an example and its original, clean counterpart form a natural positive pair, since both come from the same underlying data point but differ in how much true information is directly visible. The idea behind adding a contrastive objective on top of corruption is that, beyond simply limiting what the model can observe at each step as before, we can give the model an explicit incentive to recognize the corrupted and clean versions of an example as the same underlying data [64]. This encourages it to learn representations that can, to some extent, undo the effect of corruption. In effect, the model is nudged to recover some of the information lost to corruption purely from context, rather than relying only on the reconstruction objective to do this implicitly.

The Contrastive Objective

Contrastive objectives are generally built to increase the similarity between the representations of a positive pair relative to their similarity to a set of negative examples. A widely used version is the InfoNCE loss [65], which, for a given positive pair of embeddings (u_i, u_i^+), treats all other examples in the same training batch as negatives, and takes the form:

$$\mathscr{L}_{\text{contrastive}} = -\log \frac{\exp\left(\text{sim}(u_i, u_i^+)/\tau\right)}{\sum_{k=1}^{N} \exp\left(\text{sim}(u_i, u_k)/\tau\right)} \tag{3.19}$$

where $\text{sim}(\cdot,\cdot)$ is a similarity function between two embedding vectors (commonly cosine similarity), τ is a temperature hyperparameter that controls how sharply the loss tells apart similar and dissimilar pairs, N is the number of examples in the batch, and the sum in the denominator covers the positive pair itself along with all negative pairs formed with the other $N-1$ examples in the batch. Minimizing this loss pushes the embedding of u_i to be much more similar to its positive counterpart u_i^+ than to the embeddings of unrelated examples in the batch, shaping the embedding space so that related inputs cluster together.

Architecture: The Projector Head

To apply a contrastive objective to the baseline VAE, we add a separate projector head, shown in Figure 3.5. This projector is a single fully connected layer with a ReLU activation, and it maps the encoder's output into its own embedding space, where we compute the contrastive loss. Instead of taking the sampled latent code z as input, the projector is fed directly from μ, the mean output of the encoder, before the reparameterization trick. We do this to keep the contrastive signal separate from the randomness introduced by sampling ε, so the two embeddings being compared differ only because of the difference between their corrupted and clean inputs, not also because of independent random noise drawn for each forward pass.

Given a training example x and its corrupted counterpart $\tilde{x}$ (obtained via the corruption process described in the previous section), only $\tilde{x}$ is passed through the encoder and projector as part of the online training loop, producing the live embedding $\tilde{u}$. The embedding u of the clean example x is

instead precomputed through the same encoder-projector pipeline without gradient tracking, since the clean data is never passed through the encoder during training itself. This pair $(\tilde{u}, u)$ forms the positive pair used in the contrastive loss (Equation 3.19), with the cached offline embeddings of the other examples in the batch serving as negatives. As a result, the contrastive loss only backpropagates through the corrupted branch: it shapes the encoder's representation of corrupted inputs to move toward the fixed representation of the corresponding clean input, rather than pulling both representations toward each other symmetrically.

Combined Objective

The full training objective combines the standard reconstruction objective with the contrastive term introduced above. As in the corruption method, the encoder receives the corrupted input $\tilde{x}$, while the decoder is trained to reconstruct the original, clean example x, following the standard ELBO formulation (Equation 3.6). We add the contrastive loss (Equation 3.19), computed between the live embedding of the corrupted input and the offline embedding of its clean counterpart, to this objective as an extra term, weighted by a hyperparameter β:

$$\mathscr{L}_{\text{total}}(\theta, \phi; \tilde{x}, x) = \mathscr{L}_{\text{VAE}}(\theta, \phi; \tilde{x}, x) + \beta \, \mathscr{L}_{\text{contrastive}}(\tilde{u}, u) \tag{3.20}$$

where $\mathscr{L}_{VAE}(\theta, \phi; \tilde{x}, x)$ is the baseline VAE loss computed with the corrupted input $\tilde{x}$ passed through the encoder and the clean x used as the reconstruction target. The two terms of this objective play different roles, while the reconstruction term guides the model to produce outputs consistent with the true, uncorrupted data, the contrastive term gives a smaller, additional push for the encoder to represent corrupted examples similarly to their clean counterparts, encouraging it to learn representations that can at least partly reverse the corruption process.

Inference and Role of the Projector

We use the projector head only during training to compute the contrastive term of the loss function. At inference time, generation works exactly as in the baseline VAE. A latent z is sampled from the prior $p(z) = \mathscr{N}(0, I)$ and passed through the decoder, without any involvement from the encoder or the projector head. So the projector has no role once training is done. Its only purpose is to shape the encoder's representations during training through the contrastive objective, after which we discard it [66].

Application to the Baseline VAE

The reasoning for including this method in this study follows directly from that of corruption alone, while adding a mechanism that is "missing" from the simpler approach. Where corruption alone relies only on the reconstruction objective to implicitly make up for missing information, adding the contrastive term gives the model an explicit, direct signal to link corrupted and clean views of the same example, without ever showing the clean example itself to the encoder (when

gradients are being tracked). This enables us to isolate and compare the effect of this extra contrastive incentive against corruption alone, as well as against the other proposed approaches, in terms of both generative quality and the resulting privacy properties of the model.

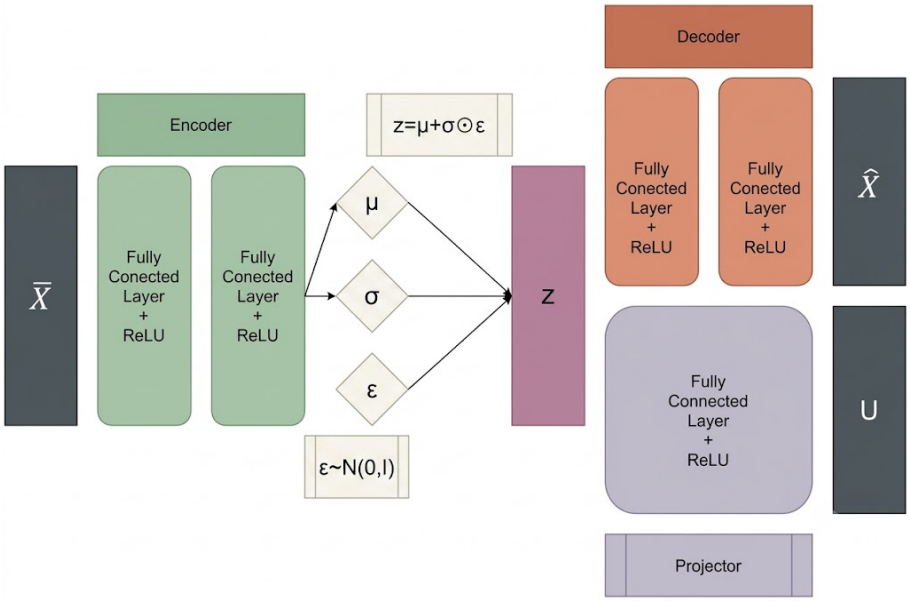


Figure 3.5: Baseline VAE Model with Contrastive Learning

Chapter 4

Experimental Setup

4.1 Datasets

To evaluate the perturbation strategies described in the previous chapter across a range of data characteristics, eight publicly available tabular datasets were used, all drawn from the healthcare and biomedical domain. These datasets were selected to provide variety in terms of size, dimensionality, and prediction task, while remaining representative of the kind of sensitive, privacy relevant data for which privacy-preserving generative modeling is most directly motivated.

Before model training, each dataset underwent a brief preprocessing stage to ensure a consistent representation across all experiments. This included removing records containing missing values, encoding categorical features into numerical representations while preserving their categorical nature, standardizing the naming of the target variable across datasets, and performing minor formatting adjustments where necessary. These preprocessing steps were deliberately kept minimal so as not to alter the underlying statistical properties of the data.

Table 4.1 summarizes the general characteristics of each dataset including the number of rows and columns, the number of numeric and categorical features, and the number of classes in the target variable. A brief description of each dataset and its associated prediction task is given below.

- **Parkinsons (Oxford Parkinson's Voice Measurements)** A dataset of biomedical voice recordings collected to distinguish healthy individuals from those with Parkinson's disease based on a range of vocal characteristics extracted from sustained phonations [67].
- **Breast Cancer** A classical diagnostic dataset containing features derived from breast mass biopsies, used to classify tumors as benign or malignant [68].
- **HCV Data (Hepatitis C Virus)** A dataset of laboratory blood test results and demographic information collected to characterize the progression of Hepatitis C virus infection, distinguishing blood donors from patients at different stages of the disease, including advanced fibrosis and cirrhosis [69].

- **Diabetes** A dataset of diagnostic clinical measurements, such as glucose concentration, blood pressure, and body mass index, used to predict the presence of diabetes in patients [70].
- **Heart Disease** A widely used clinical dataset combining several clinical attributes collected to predict the presence or absence of heart disease in patients [71].
- **Fetal Health Classification (CTG)** A dataset of cardiotocogram (CTG) exam features, extracted from fetal heart rate and uterine contraction monitoring, used to classify fetal health status as normal, suspect, or pathological [72].
- **Stroke Prediction Dataset** A dataset of patient demographic information and clinical risk factors, used to predict whether a patient is likely to experience a stroke [73].
- **Cardiovascular Disease** A large scale dataset of objective patient measurements, examination results, and laboratory features, used to predict the presence of cardiovascular disease [74].

Table 4.1: Dataset general characteristics.

	Dataset Name	Rows	Columns	Numeric Cols	Categorical Cols	Classes
0	parkinsons	195	23	22	0	2
1	breast-cancer	277	10	1	8	2
2	hepatitis	589	13	11	1	5
3	diabetes	768	9	8	0	2
4	heart	1025	14	5	8	2
5	fetal_health	2126	22	17	4	3
6	stroke	4909	11	3	7	2
7	cardio	70000	12	5	6	2

Collectively, all selected datasets are tabular, healthcare related classification tasks that mix continuous and categorical features, and vary in size from a few hundred rows to tens of thousands of rows. This selection was made deliberately, as these properties make the datasets particularly well suited to studying the trade-off between privacy and utility that is central to this project:

1. **Sensitivity** - Medical attributes are inherently sensitive, personally identifying, and realistic targets for privacy attacks, making them a meaningful setting in which to evaluate whether a given perturbation genuinely limits information leakage, rather than doing so only in an artificial or low stakes setting.
2. **Heterogeneity** - The datasets exhibit different signal-to-noise regimes, feature types, and degrees of class imbalance, allowing the proposed methods to be tested across heterogeneous tabular modalities, rather than validated on a single, narrow data distribution.

3. **Scale** - The range of dataset sizes, from a few hundred to tens of thousands of records, allows the behavior of each privacy intervention to be examined both in regimes of data abundance and in regimes of sample scarcity, where models are typically more prone to overfitting and memorization, and where privacy-utility trade-offs would be expected to be most pronounced.

Taken together, these properties make this selection of datasets well suited to evaluate how these strategies preserve the core predictive structure of the data, as measured by the generative quality and utility metrics, while limiting the leakage of information about individual training records.

4.2 Pipeline

This section describes the full experimental pipeline used in this project, from raw data to the evaluation of generated samples. The pipeline consists of four stages, applied consistently across all datasets and all method variants: data preprocessing, model training, synthetic data generation, and evaluation.

4.2.1 Preprocessing

Before being passed to any of the models, each dataset is preprocessed to place numeric and categorical features on a common, model-friendly scale.

For each numeric feature, the values are transformed to approximately follow a standard normal distribution, while preserving their original ordering. This is done in two steps. First, the feature's Cumulative Distribution Function (CDF) is estimated non-parametrically using a fixed number of quantile landmarks, placed adaptively according to the dataset size. For a given input value, this yields an estimated probability $u \in (0,1)$ representing its position within the distribution. Second, each probability u is mapped to a real number using the *probit function*, the inverse CDF of the standard normal distribution:

$$\tilde{x} = \Phi^{-1}(u) \tag{4.1}$$

Since this mapping is monotonic, the relative ordering of values within a feature is preserved, that is, larger original values remain larger after transformation. To ensure smoothness and robustness, linear interpolation is used between quantile landmarks rather than relying on the full empirical CDF. The number of landmarks K used to estimate each feature's distribution is chosen adaptively, as a function of the dataset size N:

$$K = \max\left(\min\left(\left\lfloor \frac{N}{30} \right\rfloor, 1000\right), 10\right) \tag{4.2}$$

This rule allows K to grow with the amount of available data, yielding a finer grained estimate of the distribution when more samples are available, while bounding it between a minimum of 10

and a maximum of 1000 landmarks, preventing both an overly coarse estimate on small datasets and overfitting to small fluctuations in the tails of the distribution on large ones.

The effect of this transformation is intuitive. Densely populated regions of a feature's original distribution are stretched out, while rare, extreme values in the tails are compressed. The resulting marginal distributions are approximately standard normal, which improves the numerical conditioning of the models and places all numeric features on a comparable scale, regardless of their original units or ranges.

Categorical features are converted to one-hot vectors, so that the model observes an explicit indicator for each category rather than an arbitrary numeric encoding. All transformed numeric features and one-hot encoded categorical features are then concatenated into a single input vector x, which forms the input to the models. This preprocessing pipeline is fitted exclusively on each training split and subsequently applied, without refitting, to the corresponding held-out test split, to avoid information from the test data leaking into the preprocessing step itself.

4.2.2 Training, Generation, and Post-processing

Once preprocessed, each dataset is used to train the baseline VAE and each of its perturbed variants. All models are implemented in PyTorch and trained on CPU, using the Adam optimizer [75] with a fixed learning rate of 1×10^{-3}. Every model is trained for exactly 500 epochs, with no early stopping or checkpoint selection. The model obtained at the final epoch is the one used for generation. The batch size is scaled according to the size of the dataset being used, so that the number of gradient updates per epoch remains comparable across datasets of different sizes, following:

$$B = \max\left(1, 2^{\lfloor \log_2\left(\frac{0.05N}{5}\right)\rfloor}\right) \tag{4.3}$$

where N is the number of samples in the (full) dataset, and the division by 5 accounts for the 5-fold cross validation scheme described below, so that B is computed relative to the size of each training split rather than the full dataset. Rounding to the nearest lower power of two keeps batch sizes within a small, consistent set of values across datasets. This heuristic was chosen empirically to maintain a roughly comparable number of gradient updates while keeping batch sizes computationally efficient.

After training, synthetic data is generated by sampling latent vectors from the prior $p(z) = \mathcal{N}(0, I)$ and passing them through the trained decoder, as described in Section 3.1. For each train-test split, the number of synthetic samples generated is always equal to the number of samples in the original (real) training data, so that generated and real data are directly comparable in size. Since the generated output lies in the same preprocessed space used during training (standardized numeric features and one-hot encoded categorical ones), the inverse of the preprocessing transformation described above is applied to the generated samples before any evaluation takes place. Numeric features are mapped back through the empirical quantile function corresponding to their

forward probit transformation, and one-hot categorical outputs are converted back to their original categorical labels. This step is necessary to enable the synthetic data to be compared directly against the original data, in its original feature space.

4.2.3 Cross Validation

To ensure that results are not dependent on a single, arbitrary split of the data, all experiments are conducted using a 5-fold cross validation scheme. Each dataset is partitioned into five disjoint folds. For each of the five runs, one fold is held out as the test set, while the remaining four are used as the training set on which the model is fitted and from which synthetic data is generated. Over the course of the five runs, every fold is used exactly once as the held-out test set. Every method and dataset combination is therefore evaluated across five independent train-test splits, and reported results reflect an aggregate over these five runs, rather than a single train-test split, reducing the extent to which results are affected by the particular partition of the data used.

4.3 Metrics

To evaluate the synthetic data, three complementary families of metrics are used:

Distribution based metrics, which assess how closely the statistical distribution of the generated data matches that of the real data.

Privacy based metrics, which assess the extent to which the generated data leaks information about the specific real records used during training.

Utility based metrics, which assess whether the generated data remains useful for downstream predictive tasks in place of the real data.

Together, these three families allow each method to be evaluated along the three axes of central interest to this project.

Each metric is computed by comparing, for a given train-test split, the synthetic data generated by a trained model against the corresponding real data. Where relevant, metrics are computed separately against the real training data and the held-out real test data, in order to disentangle genuine distributional fidelity from potential overfitting to the specific training examples.

4.3.1 Distribution Based Metrics

Distribution based metrics measure how closely the distribution of the generated data resembles the real distribution, without reference to any specific downstream task. In this thesis, we use three such metrics: α-Precision and β-Recall, introduced in [36] as robust, quantile based generalizations of the classical precision and recall metrics used to evaluate generative models, and detection, a complementary metric that measures overall distinguishability between real and synthetic data using a machine learning classifier.

In their standard form, precision and recall for generative models measure two complementary notions of distributional similarity. Precision measures the fidelity of the generated data, that is,

what fraction of synthetic samples are plausible, realistic-looking instances that could have come from the real data distribution. And recall measures the diversity or coverage of the generated data, in other words, what fraction of the variation in the real data distribution shows up somewhere among the synthetic samples. A model can reach high precision by generating only "safe," typical samples that closely resemble the real data, while still getting low recall if it fails to capture the full diversity of the real distribution.

However, naive precision and recall have a limitation. They are sensitive to outliers. A small number of atypical real or synthetic samples can disproportionately affect the estimated support of each distribution, and therefore the resulting scores. α-Precision and β-Recall fix this by parameterizing precision and recall over a quantile level, α and β respectively, each ranging over $[0,1]$. Instead of measuring support over the entire distribution, these metrics focus only on the α-quantile (respectively β-quantile) most typical region of the distribution, dropping the most extreme, atypical samples before computing the score.

α-Precision

α-Precision measures the fraction of synthetic samples that fall within the α-support of the real data distribution, that is the region containing the most typical α proportion of real samples. For a given α, a high α-Precision score means most generated samples are plausible, realistic instances consistent with the corresponding typical region of the real data. On the other hand, a low score means a large fraction of synthetic samples are unrealistic, falling outside the region where real data is typically found.

β-Recall

β-Recall measures the complementary notion of coverage: the fraction of real samples that fall within the β-support of the synthetic data distribution, in other words, the region containing the most typical β proportion of synthetic samples in the embedding space. For a given β, a high β-Recall score means the synthetic data captures most of the diversity and variation in the real data. A low score, on the other hand, means the generative model failed to reproduce large parts of the real data distribution, for instance by collapsing onto only some of the modes present in the real data.

Reporting both metrics together matters precisely because they capture different, complementary failure modes. We computed both these metrics using the `synthcity` implementation [76].

Detection

Unlike α-Precision and β-Recall, which each isolate one specific notion of distributional similarity (fidelity or diversity), detection instead measures how easy it is to tell real and synthetic data apart in a single and direct test. We do this by framing the comparison as a binary classification problem. Real and synthetic samples are pooled into one dataset, each labeled by origin, and a simple machine learning classifier is trained to predict, from the features of a given row

alone, whether it is real or synthetic. The idea is straightforward, if the synthetic data is truly indistinguishable from the real data, no classifier should perform this task any better than random guessing. On the other hand, if the synthetic data is systematically different from real data in some way, a classifier should be able to exploit this and reach high accuracy.

In this project, we compute detection using the implementation from `sdmetrics` [77]. Real and synthetic data are combined into a single labeled table, and a classifier is trained and validated across cross validation splits to predict each row's origin. The classifier's discriminative performance is measured using the area under the ROC curve (AUC), averaged across splits, and converted into a detection score as:

$$\text{Detection} = 1 - (\max(\text{AUC}, 0.5) \times 2 - 1) \tag{4.4}$$

An AUC of 0.5 (random guessing classification) maps to a detection score of 1, while an AUC of 1.0 (perfect discrimination) maps to a score of 0. A high detection score therefore means that real and synthetic data are hard to tell apart, while a low score means the classifier easily finds systematic differences between them.

An unusually high detection score is not necessarily a good sign. If synthetic samples are indistinguishable from real data because the model has closely copied training records, this would also yield a high score, despite reflecting a privacy failure rather than genuine generative quality. For this reason, detection should not be interpreted in isolation.

Taken together, α-Precision and β-Recall give a fine-grained breakdown of distributional similarity into fidelity and diversity, while detection gives a single overall measure of how distinguishable the two distributions are. Reporting all three together gives a fuller picture of generative quality.

4.3.2 Privacy

Privacy based metrics measure how much the generated data reveals about the specific real records used to train the generative model. Privacy based metrics check whether individual synthetic samples are, in effect, disguised copies of real training examples, that is, a form of memorization, which would defeat the purpose of using synthetic data as a privacy-preserving stand-in for the original data in the first place. In this thesis, we use two privacy based metrics.

Authenticity

Authenticity, introduced by Alaa et al. [36], measures the proportion of synthetic samples that are genuinely new, rather than near duplicates of real training examples. For each synthetic sample, we find its nearest neighbor in the real training data and the nearest training data point to that. A synthetic sample is considered authentic if the nearest neighbor of its closest real data point is itself a real data point. On the other hand, if the closest overall neighbor to that closest real data point is the generated sample at hand, this suggests the model has generated a sample that is, a memorized or near copied version of that real record. The authenticity score for a given synthetic dataset

is the fraction of synthetic samples meeting this condition. A high authenticity score means the generative model produces genuinely new samples that are not just disguised copies of the training data, while a low score means a large portion of the generated data is suspiciously close to specific real training records, raising concerns about memorization and, consequently, privacy.

Privacy Loss

The second privacy based metric in this project is privacy loss, based on nearest neighbor adversarial accuracy (*AA*), proposed by Yale et al. [35]. This metric builds on the idea of the previous one, but is more complex , so it is best to first look at the underlying adversarial accuracy measure it is built on, before explaining how it is used to define privacy loss.

Consider two datasets, drawn from a target domain T and a source domain S (for instance, real and synthetic data), embedded in a shared, standardized feature space. For each point i in T, let $d_{TS}(i)$ be the distance from that point to its nearest neighbor in S, and let $d_{TT}(i)$ be its "leave-one-out" nearest neighbor distance to the closest *other* point in T itself (that is, excluding i). We define symmetric quantities $d_{ST}(i)$ and $d_{SS}(i)$ the same way for points in S. The nearest neighbor adversarial accuracy between T and S is then:

$$AA_{TS} = \frac{1}{2}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\left(d_{TS}(i) > d_{TT}(i)\right) + \frac{1}{n}\sum_{i=1}^{n}\mathbf{1}\left(d_{ST}(i) > d_{SS}(i)\right)\right) \tag{4.5}$$

Intuitively, we can think of AA_{TS} as the balanced accuracy of an adversarial classifier trying to tell points from T and S apart purely from nearest neighbor distances. A point in T counts as "correctly identified" if it lies farther from its nearest neighbor in S than from its nearest neighbor within T itself, and the same logic applies for points in S. If the two datasets are statistically indistinguishable, AA_{TS} should be close to 0.5, matching what we would expect from random chance.

To apply this measure when evaluating a trained generative model G, we consider four datasets: the real data used to train G (R_{tr}), a held-out real test set drawn independently from the same distribution (R_{te}), and two independently generated synthetic datasets, A_1 and A_2, both produced by G. Using these, we define three values:

$$\text{TrResemblLoss} = \mathbb{E}\left[AA_{R_{tr}A_1}\right] \tag{4.6}$$

$$\text{TeResemblLoss} = \mathbb{E}\left[AA_{R_{te}A_2}\right] \tag{4.7}$$

$$\text{PrivacyLoss} = \mathbb{E}\left[AA_{R_{te}A_2} - \mathscr{A}\mathscr{A}_{R_{tr}A_1}\right] \tag{4.8}$$

TrResemblLoss measures how distinguishable the synthetic data is from the real data the model was trained on, while TeResemblLoss measures how distinguishable it is from unseen real data drawn from the same distribution. PrivacyLoss is simply the difference between the two.

The value of looking at this difference, rather than either resemblance score on its own, becomes clear when we think about how a generative model can fail. If G generalizes well, without

overfitting to any specific training example, both the training and test data should be similarly hard to distinguish from synthetic data, so TrResemblLoss and TeResemblLoss should both be close to 0.5, and PrivacyLoss should be close to 0. If G underfits, producing synthetic data that is unrealistic overall, an adversarial classifier will easily tell real from synthetic data whether the real data comes from the training or test set and both TrResemblLoss and TeResemblLoss should then be high ($>> 0.5$) and similar to each other, and PrivacyLoss should again be close to 0, but in this case, privacy is only preserved because the synthetic data is of low quality, not because the model generalizes well. If, however, G overfits the training data, effectively memorizing individual training examples, synthetic samples will tend to sit suspiciously close to the specific real records used in training. TrResemblLoss will then be low (closer to 0), while TeResemblLoss, that was computed against real data the model never saw, will stay around 0.5. The resulting gap between the two gives a high PrivacyLoss, near 0.5, correctly flagging this as a privacy risk caused by memorization of the training set.

Taken together, PrivacyLoss lets us isolate memorization specifically, separating it from the broader question of whether the synthetic data resembles real data at all. In this thesis, we report TrResemblLoss and TeResemblLoss alongside PrivacyLoss itself, since, as shown above, the same PrivacyLoss value could, in principle, come from different underlying combinations of resemblance to training and test data. Reporting all three together allows us to not only spot possible privacy concerns, but also understand how the generator is performing in a broader sense.

4.3.3 Utility

Utility based metrics check whether synthetic data still works as a substitute for real data in a downstream, practical task. This is an important addition to the distribution based and privacy based metrics as a synthetic dataset could, in principle, score reasonably on distributional similarity while still providing limited practical use, or could get strong privacy scores only because it has become uninformative. Utility based metrics test this directly, by checking how well a model trained on synthetic data performs on the classification tasks, compared to a model trained on real data.

To do this, we train two classifiers, one on the real training data, and one on the corresponding synthetic data generated by the model being evaluated. In both cases, we use an XGBoost classifier [78], because of its strong general purpose performance on tabular data and its wide use as a standard benchmark model in this setting. We then evaluate both classifiers on the same held-out real test set, letting us compare their predictions directly. From these predictions, we compute two utility metrics.

Balanced Accuracy Gap

The balanced accuracy gap measures the difference in downstream predictive performance between the two classifiers, using balanced accuracy (the average of the recall for each class) instead of raw accuracy, so that performance on minority classes is not overshadowed by performance

on the majority class. This matters especially since several of the classification tasks tied to the datasets in this thesis and in the medical domain are class imbalanced. We define the balanced accuracy gap as:

$$\text{BalAccGap} = \text{BalAcc}(\text{Real-trained model}) - \text{BalAcc}(\text{Synthetic-trained model}) \tag{4.9}$$

where we compute both balanced accuracies on the same held-out real test set. A value close to 0 means a model trained purely on synthetic data performs about as well as one trained on real data, including on minority classes, so it can be considered a viable substitute for the downstream task. A high, positive value means the synthetic-data-trained model performs worse than the real-data-trained model, and since we are using balanced rather than raw accuracy, this may specifically mean the synthetic data fails to properly represent underrepresented classes or regions of the feature space, even if it looks reasonable overall.

Alignment

While the balanced accuracy gap compares each model's performance against the test set's ground truth labels, alignment instead measures how much the two models agree with each other, regardless of whether either prediction is correct. Concretely, we compute alignment as the proportion of test samples where the model trained on real data and the model trained on generated data give the same predicted label:

$$\text{Alignment} = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} \mathbf{1}\left(\hat{y}_i^{\text{real}} = \hat{y}_i^{\text{synthetic}}\right) \tag{4.10}$$

A high alignment score means the two models tend to make the same decisions across the test set, suggesting the synthetic data leads to a similar decision boundary as the real data.

Looking at these two metrics together gives a fuller picture of downstream utility than either one alone. Since it is measured against ground truth labels, the balanced accuracy gap tells us a lot about how a model trained on synthetic data performs in low-populated regions of the feature space, like minority classes, where drops in performance can be hidden by raw accuracy but show up in the balanced accuracy version. Alignment, on the other hand, shows whether the two models behave consistently with each other across the whole test set.

For example, a high alignment score alongside a large balanced accuracy gap would suggest that both models behave similarly in the majority of the data (therefore the high alignment), but the model trained on synthetic data fails to correctly predict minority class examples, which causes a big accuracy gap without much interference on the alignment metric.

4.4 Hyperparameters

This section explains the specific hyperparameter configurations used for each of the five proposed experiment types. For each method, we vary one or more hyperparameters to create a set of distinct experimental variants, while keeping any remaining hyperparameters fixed across all variants of that method. Tables 4.2-4.6 summarize these configurations. All variants of all methods share the same training details described before.

4.4.1 Regularization

For regularization, we vary both the L1 and L2 penalty weights, each taking a value from $\{0, 10^{-3}, 10^{-4}\}$. Since the configuration where both weights are 0 is just the unregularized baseline VAE, we exclude it from this set of experiments. This leaves $3 \times 3 - 1 = 8$ combinations of L1 and L2 weights, shown in Table 4.2, each corresponding to one experimental variant.

Table 4.2: Hyperparameter configurations for the Regularization variants.

	Reg-1	Reg-2	Reg-3	Reg-4	Reg-5	Reg-6	Reg-7	Reg-8
L1	0	0	10^{-4}	10^{-4}	10^{-4}	10^{-3}	10^{-3}	10^{-3}
L2	10^{-4}	10^{-3}	0	10^{-4}	10^{-3}	0	10^{-4}	10^{-3}

4.4.2 Dropout

For dropout, we vary a single hyperparameter, the dropout rate p, applied the same way between the two fully connected layers of both the encoder and the decoder, as described in Section 3.2.2. We test four values of p, shown in Table 4.3, covering a fairly wide range from a mild perturbation ($p = 0.1$) to a very aggressive one ($p = 0.75$), letting us look at how dropout intensity affects both generative quality and privacy.

Table 4.3: Hyperparameter configurations for the Dropout variants.

	Drop-1	Drop-2	Drop-3	Drop-4
Dropout rate p	0.1	0.25	0.5	0.75

4.4.3 Differential Privacy

For differential privacy, we vary the privacy budget ε across four values, shown in Table 4.4, ranging from the most permissive setting ($\varepsilon = 2$) to the strictest ($\varepsilon = 0.1$), where lower values mean stronger privacy guarantees. We keep the failure probability δ fixed across all four variants, set as $\delta = \frac{1}{100N}$, where N is the number of samples in the dataset being used. Since δ is defined relative to dataset size rather than as one fixed global constant, its numeric value differs between datasets, but we apply the same formula consistently across all DP variants and all datasets. We

also keep the gradient clipping threshold C, which bounds the per example gradient norm used during DP-SGD training (Equation 3.16), fixed at $C = 1.5$ across all variants.

Table 4.4: Hyperparameter configurations for the Differential Privacy variants.

	DP-1	DP-2	DP-3	DP-4
Privacy budget ε	2	1	0.5	0.1
Failure probability δ		$1/(100N)$		
Clipping norm C		1.5		

4.4.4 Corruption

For corruption, we vary the corruption ratio r across four values, shown in Table 4.5, following the corruption mechanism described in Subsection 3.2.4. Instead of corrupting a fixed set of columns, we select the specific columns to corrupt independently for every row. For each row, we compute the number of columns to corrupt as $\lfloor r \cdot n_{\text{cols}} \rfloor$, where n_{cols} is the total number of columns in the dataset, and then choose exactly that many columns at random (independently per row) to corrupt, as described in Equation 3.18. So, while the overall corruption ratio r stays fixed for a given variant, the specific columns affected change from row to row, rather than corrupting a fixed subset of columns for the whole dataset.

Table 4.5: Hyperparameter configurations for the Corruption variants.

	Corr-1	Corr-2	Corr-3	Corr-4
Corruption ratio r	0.1	0.25	0.5	0.75

4.4.5 Contrastive Learning

For contrastive learning, we vary the corruption ratio r across the same four values used for the standalone corruption method, shown in Table 4.6, applied the same way to get the corrupted view of each training example passed to the encoder. We keep two additional hyperparameters of the contrastive objective fixed across all four variants. We fix the temperature τ used in the InfoNCE contrastive loss (Equation 3.19) at $\tau = 0.2$. The weight β applied to the contrastive loss term in the combined objective (Equation 3.20) is not held at a single constant value. Instead, it follows a fixed linear schedule during training, decreasing from $\beta = 0.1$ at the start of training to $\beta = 0.01$ by the end. We apply this schedule the same way across all four variants. It is designed to make the contrastive term matter more early in training, when it may help the encoder build representations that already link corrupted and clean views of the same example, while letting the reconstruction term take over as training progresses.

Table 4.6: Hyperparameter configurations for the Contrastive Learning variants.

	Cont-1	Cont-2	Cont-3	Cont-4
Corruption ratio r	0.1	0.25	0.5	0.75
Temperature τ	0.2			
Contrastive weight β	Linear: $0.1 \rightarrow 0.01$			

Chapter 5

Results and Discussion

This chapter presents and analyzes the experimental results for the evaluated methods. The analysis is organized into three complementary parts, each providing a different perspective on the behavior of the evaluated methods.

The first part, *Individual Metric Analysis*, looks at each evaluation metric on its own, comparing the performance of all experimental methods and their hyperparameter configurations against the baseline VAE. This gives a detailed understanding of how each modification affects specific aspects of data quality, privacy, and downstream utility, letting us assess the impact of individual design choices on their own.

The second part, *Dual-Metric Analysis*, looks at pairs of complementary metrics together to give a broader view of model behavior. Instead of considering each metric separately, this analysis highlights the relationships between different performance dimensions. It shows how the evaluated methods position themselves in terms of overall data quality, utility, and privacy, and illustrates the trade-offs that come with increasingly aggressive modifications to the baseline model.

Finally, the *Metafeature Correlation Analysis* looks at the relationship between the structural characteristics of the generated synthetic datasets and their performance across the evaluated metrics. By correlating general and statistical metafeatures from the synthetic data with the corresponding evaluation scores, this analysis shifts the focus from comparing methods to understanding what properties of the generated datasets consistently go along with higher or lower performance. Although this part is exploratory, it gives useful insights into the characteristics of synthetic data that may help explain why some methods get better results than others.

5.1 Individual Metric Results

5.1.1 Distribution based Metrics

In this section, we will go over each metric, one by one, and analyze the performance of each proposed method and the baseline approach across the multiple datasets for that specific metric. For further detailed analysis, the original data from the experiments is available in Tables A.2

through A.12. To measure the uncertainty of the estimated means, we calculated confidence intervals for each metric within each method. For every experiment, we calculated the mean, standard deviation, and sample size and computed the standard error of the mean by dividing the standard deviation by the square root of the sample size. We then calculated the lower and upper confidence bounds by adding the corresponding quantiles of the sampling distribution, scaled by the standard error, to the group mean. In particular, we used the 10th and 90th percentiles, yielding a central 80% confidence interval.

α-Precision

Figure 5.1 summarizes the average α-Precision obtained by each experimental configuration across all datasets.

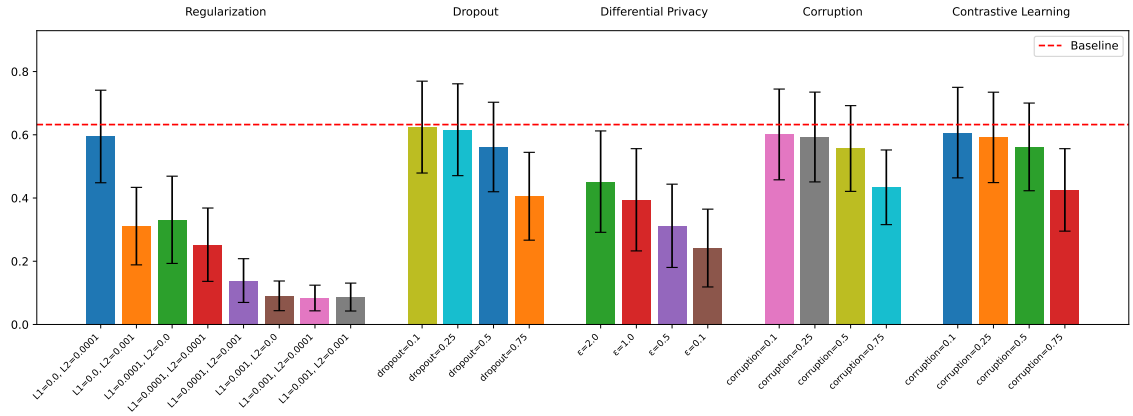


Figure 5.1: α-Precision per experiment across datasets.

The baseline VAE gets the highest average α-Precision scores. Only a few other setups reach similar results. Specifically, low dropout rates (0.1 and 0.25), low corruption levels (0.1 and 0.25), and contrastive learning variants up to a corruption level of 0.5 perform similarly to the baseline. Their confidence intervals overlap with the baseline's. This suggests that these methods match the model's ability to generate samples within the real data distribution, without a substantial loss in fidelity.

We see a common pattern across all methods. α-Precision drops more as the modification gets stronger. Higher regularization, higher dropout rates, less privacy budget and stronger input corruption cause a slow decline in performance. Adding perturbation mechanisms to generative models makes the synthetic data less faithful to the real data, which is consistent with expectations.

Overall, α-Precision values remain relatively modest across all evaluated methods, including the baseline, with none of the approaches exceeding approximately 0.63. Since α-Precision reflects the extent to which generated samples lie within the support of the real data distribution, these results suggest that the synthetic data distribution only partially overlaps with the real one, irrespective of the training strategy employed. While some methods preserve α-Precision better

than others, none of the evaluated methods, including the baseline, fully reproduce the original data distribution.

β-Recall

Figure 5.2 presents the average β-Recall for each experimental configuration across all datasets, together with the corresponding confidence intervals.

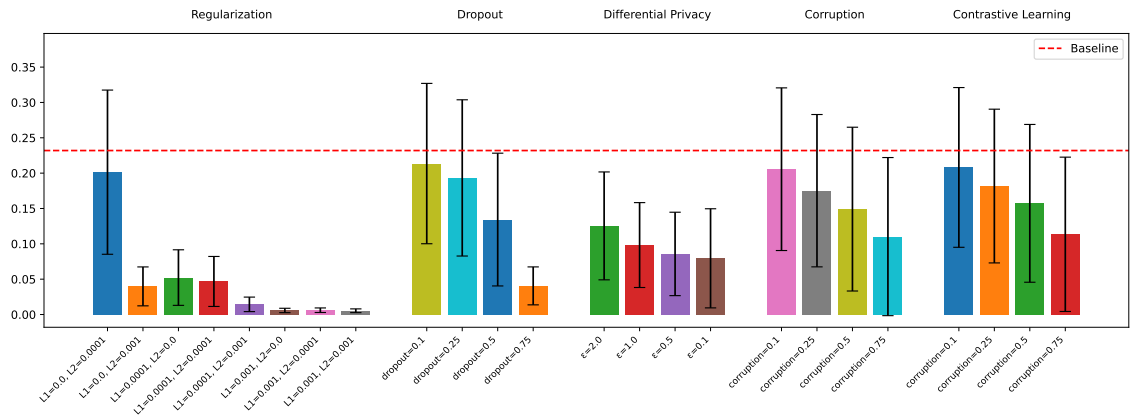


Figure 5.2: β-Recall per experiment across datasets.

As the plot shows, the baseline VAE gets the highest mean β-Recall, although several other setups reach similar values within the confidence intervals. Specifically, regularization (L1 = 0 and L2 = 0.0001), low dropout rates (0.1 and 0.25), a low corruption level (0.1), and contrastive learning variants with corruption rates of 0.1 and 0.25 stay fairly close to the baseline. This suggests that these methods mostly preserve the diversity of the generated samples, while stronger modifications tend to reduce the model's ability to capture the full variability of the real data distribution.

We see a steady drop in β-Recall as the modifications get stronger. Stronger regularization causes a sharp drop in recall. Except for the weakest setting, every regularization configuration gives values close to zero. Similarly, higher dropout or corruption rates gradually lower β-Recall, and differential privacy shows the same steady decline as the privacy budget gets smaller. These trends show that stronger constraints on training limit the model's ability to reproduce the diversity of the original data.

Overall, β-Recall values stay fairly low across all methods, including the baseline, which gets a mean value of only about 0.23. Since β-Recall measures how well the generated samples cover the real data distribution, these results suggest that none of the evaluated models fully captures the diversity of the original data. Some approaches keep recall better than others, but in general, the synthetic distributions only represent part of the variability found in the real datasets.

Detection

Figure 5.3 compares the average Detection score achieved by each method across the evaluated datasets.

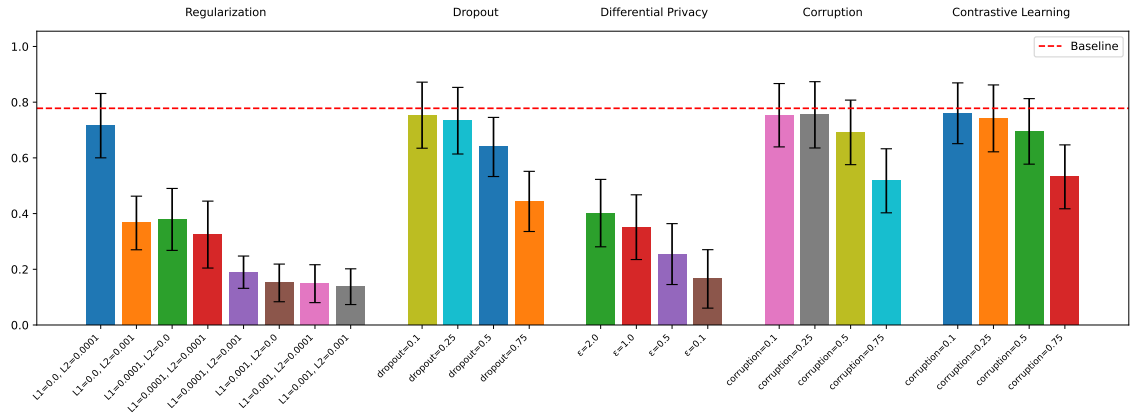


Figure 5.3: Detection per experiment across datasets.

The baseline VAE gets the highest Detection score, though several other setups reach similar results within the confidence intervals. Specifically, the least aggressive approaches stay close to the baseline. Since a higher Detection score means that a classifier has greater difficulty distinguishing real and synthetic samples, these results suggest that these methods keep the generated data realistic and hard to tell apart from the original distribution.

As with the other metrics, stronger modifications generally lead to lower Detection scores. Strong regularization reduces performance substantially, and only the weakest regularization setting stays close to the baseline. In the same way, higher dropout rates and stronger corruption gradually lower the score. This shows that more constrained training produces synthetic data that a detection classifier can more easily tell apart from real data.

Overall, the Detection scores suggest that the baseline and the milder variants generate synthetic data that is statistically similar to the original data, making it difficult for the classifier to distinguish between the two. However, this metric should not be interpreted in isolation. A high Detection score does not necessarily imply better synthetic data quality, as it may also arise when a model closely reproduces the training data rather than learning its underlying distribution. Likewise, a lower Detection score indicates that the classifier can identify differences between the real and synthetic data, but does not reveal whether these differences stem from poor generative quality, increased privacy preservation, or other reasons. Consequently, Detection should be viewed as an indicator of statistical similarity that complements, rather than replaces, the quality and privacy metrics presented in the preceding and following subsections.

5.1.2 Privacy Metrics

Authenticity

Figure 5.4 shows the average Authenticity score for each experimental configuration across all datasets, with the corresponding 80% confidence intervals.

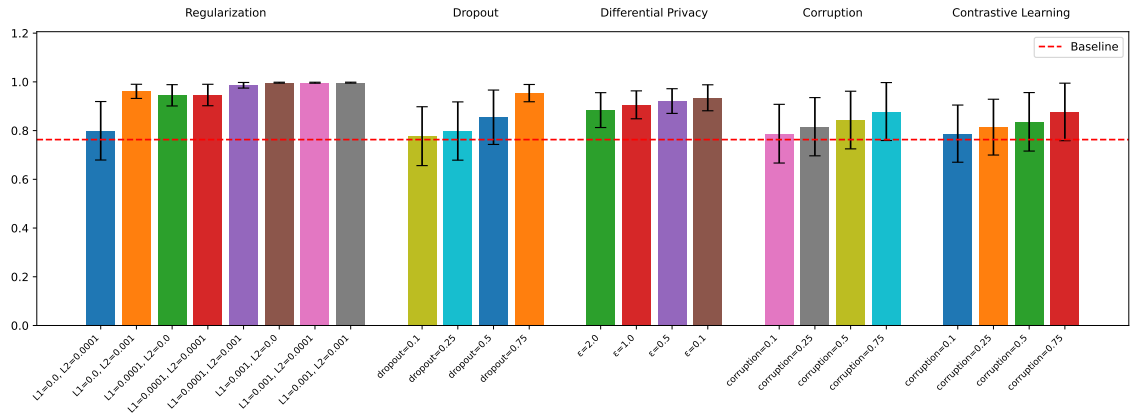


Figure 5.4: Authenticity per experiment across datasets.

The baseline VAE gets the lowest Authenticity score, while almost all modified models achieve higher scores than it. The biggest improvements come from regularization based approaches, where several get close to the maximum score. Differential privacy also consistently gets high Authenticity, while dropout, corruption, and contrastive learning show smaller improvements over the baseline. These results suggest that adding constraints during training generally reduces the generator's tendency to produce samples that closely resemble individual training examples.

For most methods, Authenticity increases as the modification gets stronger. This overall pattern stays consistent across methods, which suggests that mechanisms which limit model capacity or add noise tend to make the generator produce samples that are more different from the training data.

A higher Authenticity score means that generated samples are less likely to closely resemble individual training examples, suggesting that the model relies less on memorization. This provides evidence that the risk of disclosing information about specific training samples is reduced. However, a high Authenticity score alone is not sufficient to conclude that privacy has improved. The increase may also result from the generator producing samples that have simply shifted away from the true data distribution, rather than generating novel yet representative samples. Therefore, Authenticity should be interpreted as evidence of reduced memorization, but in combination with other metrics to distinguish genuine privacy improvements from a loss of fidelity to the underlying data distribution.

Privacy Loss

Figures 5.5, 5.6, and 5.7 present the privacy evaluation. The first two show the Train and Test Nearest Neighbor Adversarial Accuracy, while the third reports the resulting Privacy Loss, computed as the difference between the two metrics.

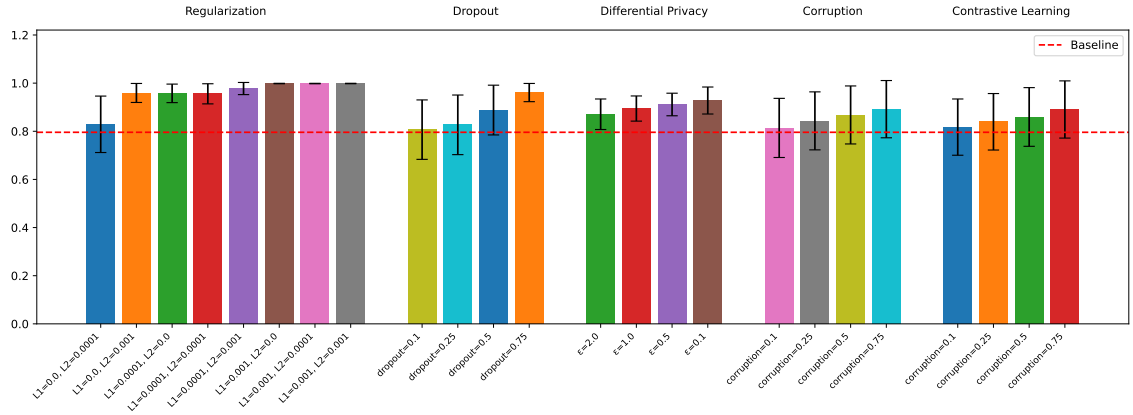


Figure 5.5: Train Adversarial Accuracy per experiment across datasets.

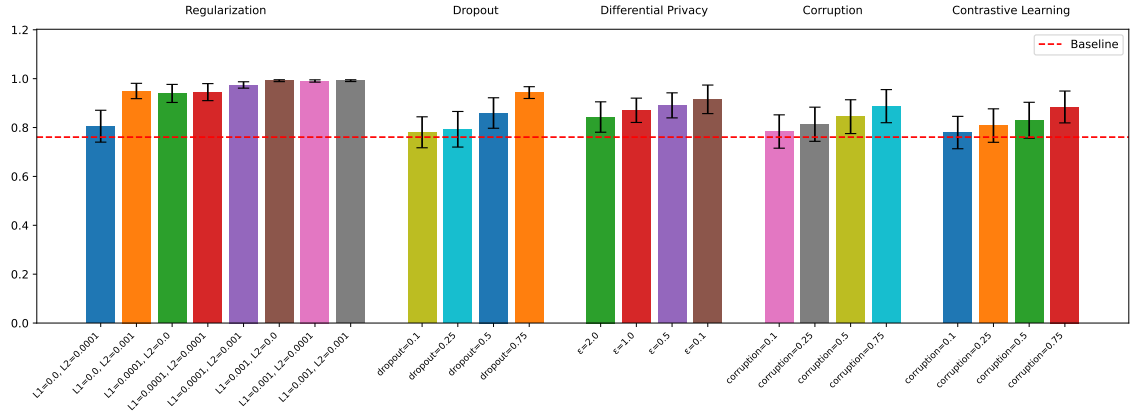


Figure 5.6: Test Adversarial Accuracy per experiment across datasets.

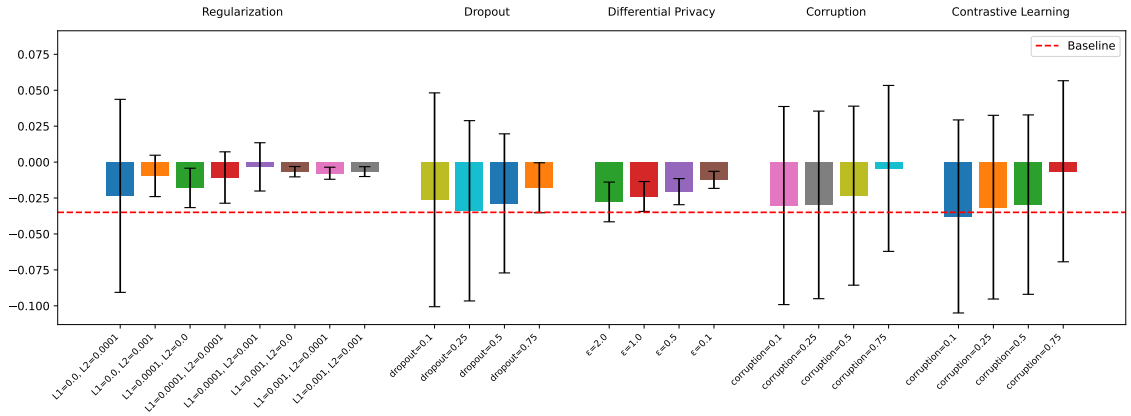


Figure 5.7: Privacy Loss per experiment across datasets.

Across all methods, both metrics stay well above the ideal value of 0.5, with most configurations ranging between about 0.8 and 1.0. Rather than suggesting high levels of privacy, these high adversarial accuracies suggest that the synthetic datasets are easy to tell apart from both the training and holdout sets. Notably, this already happens with the baseline VAE, which shows that the generated samples are noticeably different from real data, regardless of any added modifications.

The proposed methods mostly keep this same behavior. Stronger variation of the experiment types generally raise both sub-metrics. However, these increases happen almost equally for the training and holdout sets, so the two sub-metrics follow very similar trends. This suggests that the evaluated modifications mainly affect the overall resemblance between synthetic and real data, rather than specifically changing the relationship between the synthetic data and the training samples.

This shows up in the Privacy Loss results. Since Privacy Loss is defined as the difference between Train Adversarial Accuracy and Test Adversarial Accuracy, and the two adversarial accuracies stay close together, the Privacy Loss values stay almost unchanged.

The slightly negative mean values seen for several configurations also fit with the idea that there is no systematic memorization. They show that the synthetic data is not especially closer to the training set than to unseen data. Overall, the consistently low Privacy Loss gives little evidence that any of the evaluated models memorizes the training data.

However, we should read this result together with the Adversarial Accuracy values. Since both the baseline and the proposed methods stay well above the ideal adversarial accuracy of 0.5, the low Privacy Loss seems to come from the synthetic data being similarly distinguishable from both the training and holdout distributions and not from the synthetic data closely matching the real data while also avoiding memorization.

So, the proposed methods do not meaningfully change the privacy properties of the baseline model. The main limitation seems to be the generative quality of the VAE itself, rather than the specific training strategy used.

5.1.3 Utility Metrics

Balanced Accuracy Gap

Figure 5.8 reports the average Balanced Accuracy Gap for each experimental configuration across the evaluated datasets.

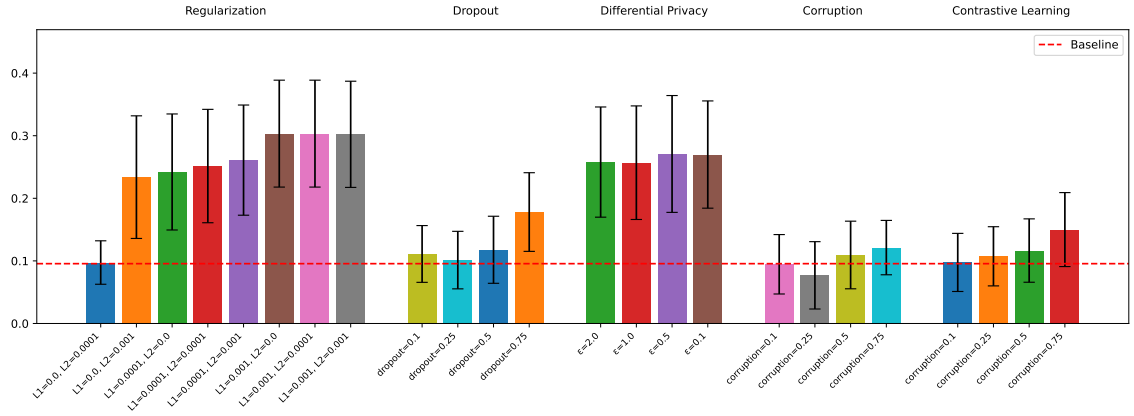


Figure 5.8: Balanced Accuracy Gap per experiment across datasets.

The baseline VAE gets one of the lowest Balanced Accuracy Gap values. This means classifiers trained on its synthetic data perform close to classifiers trained on the original data (in comparison to other methods). Among the methods tested, corruption and contrastive learning variants stay the most competitive, with several configurations showing values similar to the baseline. In contrast, regularization and differential privacy consistently produce much larger gaps, suggesting a bigger loss of predictive information in the generated data.

As the modifications get stronger, the Balanced Accuracy Gap generally gets bigger. These results show that stronger constraints on the training process tend to reduce how useful the synthetic data is for downstream classification tasks, even if they may help in other aspects. This is not surprising, as the methods that most influence the model end up generating less quality data.

Overall, the Balanced Accuracy Gap gives an indirect measure of how well the synthetic data keeps the information needed for predictive modeling. Because of this, the metric adds to the distributional quality metrics by measuring the practical usefulness of the generated data, not just its statistical similarity.

Alignment

Figure 5.9 presents the average Alignment score obtained by each method across all datasets.

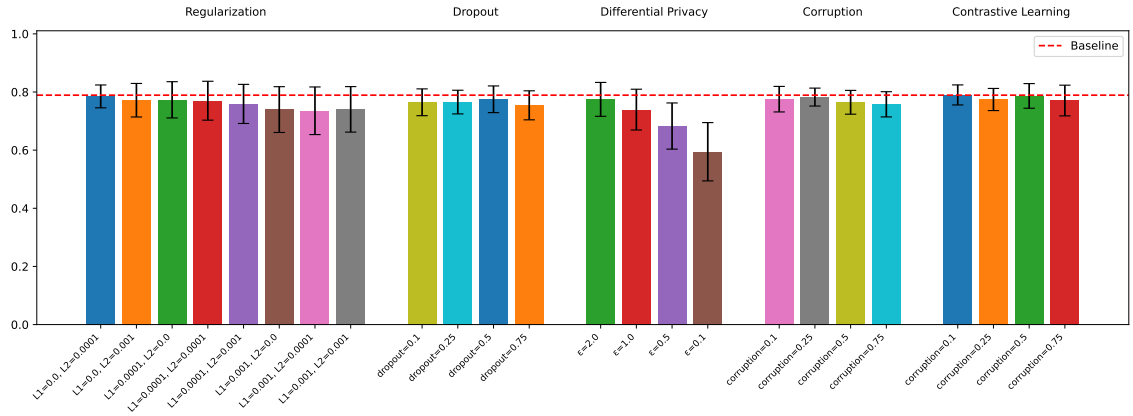


Figure 5.9: Alignment per experiment across datasets.

The standard VAE gets one of the highest Alignment scores. This means classifiers trained on the original and synthetic datasets make very similar predictions on the test set. Most of the evaluated methods stay close to the baseline, especially the corruption and contrastive learning variants, as well as the dropout configurations, whose confidence intervals mostly overlap with the baseline's. In contrast, stronger regularization and differential privacy setups cause a noticeable drop in alignment, suggesting that their synthetic datasets lead to different decision boundaries than the ones learned from the original data.

Stronger regularization generally lowers alignment, while differential privacy shows the clearest steady decline as the privacy budget decreases. In contrast, dropout, corruption, and contrastive learning keep fairly stable alignment scores across their configurations, showing that these methods preserve the original model's predictive behavior more effectively.

Overall, the consistently high Alignment scores suggest that classifiers trained on synthetic data often make similar predictions to those trained on the original data, even when there are differences in predictive performance. Since this metric measures agreement between predictions rather than correctness, it should be read together with utility metrics like the Balanced Accuracy Gap. Interestingly, the differences between the proposed methods and the baseline are considerably smaller for Alignment than for the Balanced Accuracy Gap. **This suggests that the synthetic data preserves the overall decision boundaries of the original data reasonably well, while still failing to capture less populated regions of the data distribution, such as minority classes.** Errors in these regions have a much larger impact on balanced accuracy, which explicitly accounts for class imbalance, whereas Alignment is dominated by agreement on the more prevalent patterns in the data.

5.1.4 Train and Generation Times

Dropout and corruption methods showed no significant change in training time compared to the baseline. Regularization and contrastive learning showed a small but noticeable increase. Differential privacy, however, clearly stands out as the slowest method in terms of training time, taking more than twice as long as the baseline model.

On the other hand, during generation, all models were extremely fast, always taking much less than 0.1 seconds. This highlights the difficulty of training generative models, and shows how easy and fast they can be to use during inference.

The train and generation times are available in the Appendix Tables A.11 and A.12.

5.2 Dual-Metric Results

The following figures look at pairs of evaluation metrics together, giving a fuller picture of how each method behaves than looking at each metric on its own. In each plot, the two axes represent the metrics being compared, and each marker shape stands for one of the evaluated experiment types. Within each experiment type, the marker size shows the strength of the applied modification. The larger mark represents the most aggressive hyperparameter settings, and smaller mark represents the milder one. This way of showing the data lets us see the relationship between complementary metrics, spot possible trade-offs, and compare how different methods change as their hyperparameters become more restrictive.

It is worth noting that marker size is only meaningful within a given experiment type, not across them. Dropout rate, regularization weight, corruption ratio, and the differential privacy budget act on entirely different scales, so there is no principled way to say that, for instance, the largest marker for dropout represents an equally aggressive intervention as the largest marker for regularization. The progression from small to large markers should therefore be read as showing how each method behaves as it is pushed harder on its own terms, rather than as a like-for-like comparison of aggressiveness across methods.

5.2.1 Trade-off between α-Precision and β-Recall

Figure 5.10 looks at α-Precision and β-Recall together, giving an overall view of the quality of the generated data.

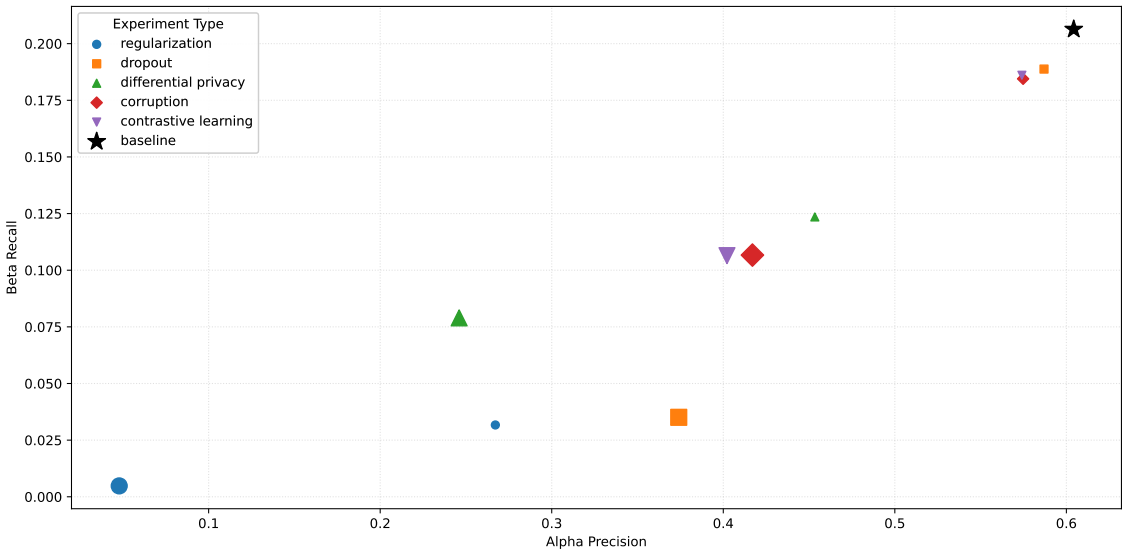


Figure 5.10: α-Precision and β-Recall per Experiment type across datasets

Unlike other pairs of metrics, we do not expect an inherent trade-off between these two. A model that captures the real data distribution more accurately should generate more faithful samples (higher α-Precision) and also better cover the diversity of the original data (higher β-Recall) at the same time. Because of this, the most desirable region of the plot is the upper-right corner, where the baseline model sits.

The evaluated methods follow a clear diagonal trend, moving from the baseline towards the lower-left region of the plot. This shows both quality metrics gradually getting worse as the training modifications become more aggressive. This matches what we would expect, since stronger regularization, more corruption, higher dropout rates, and stricter privacy constraints all add extra restrictions to the learning process, reducing the model's ability to accurately reproduce the real data distribution. So instead of revealing a trade-off between α-Precision and β-Recall, the plot shows the overall loss of generative quality caused by increasingly constrained models.

Among the methods tested, corruption and contrastive learning stay closest to the baseline across their hyperparameter ranges, showing that they better preserve the original distribution of the data. Dropout shows a similar but slightly steeper decline, while differential privacy moves further from the baseline as the privacy budget decreases. Regularization shows the sharpest decline, with its strongest configurations giving the lowest values for both metrics by some margin.

5.2.2 Trade-off between α-Precision and Authenticity

Figure 5.11 shows the relationship between α-Precision and Authenticity, highlighting the trade-off between generating samples that closely resemble the real data distribution and generating samples that are different enough from the training data.

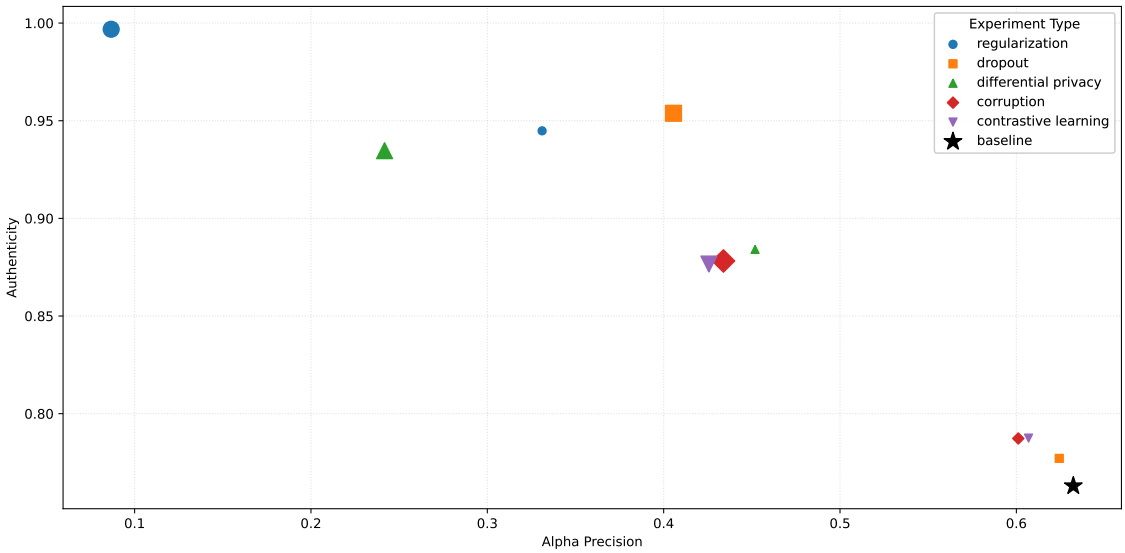


Figure 5.11: α-Precision and Authenticity per Experiment type across datasets

The baseline VAE gets the highest α-Precision but also the lowest Authenticity. This means its better fidelity comes with a stronger resemblance to the training data.

This trade-off shows up consistently across all evaluated methods. As each modification gets stronger, its points generally move towards the upper-left region of the plot, showing higher Authenticity and lower α-Precision. This is especially clear for regularization, which gives the biggest gains in Authenticity but also the biggest loss in generative fidelity. Differential privacy follows a similar trend, though the drop in α-Precision is less severe. Corruption and contrastive learning sit in between the baseline and the more aggressive privacy-preserving methods.

Overall, the results suggest that making generated samples more novel and private generally reduces how similar they are to the real data distribution. While higher Authenticity gives evidence against directly memorizing the training data, it is not automatically a good thing if it comes at the cost of faithfully representing the real distribution. Because of this, the figure shows why we need to look at quality and privacy metrics together.

5.2.3 Trade-off between Machine Learning (ML) Utility and Authenticity

Figure 5.12 highlights the trade-off between the authenticity of the generated samples and their usefulness for downstream classification tasks.

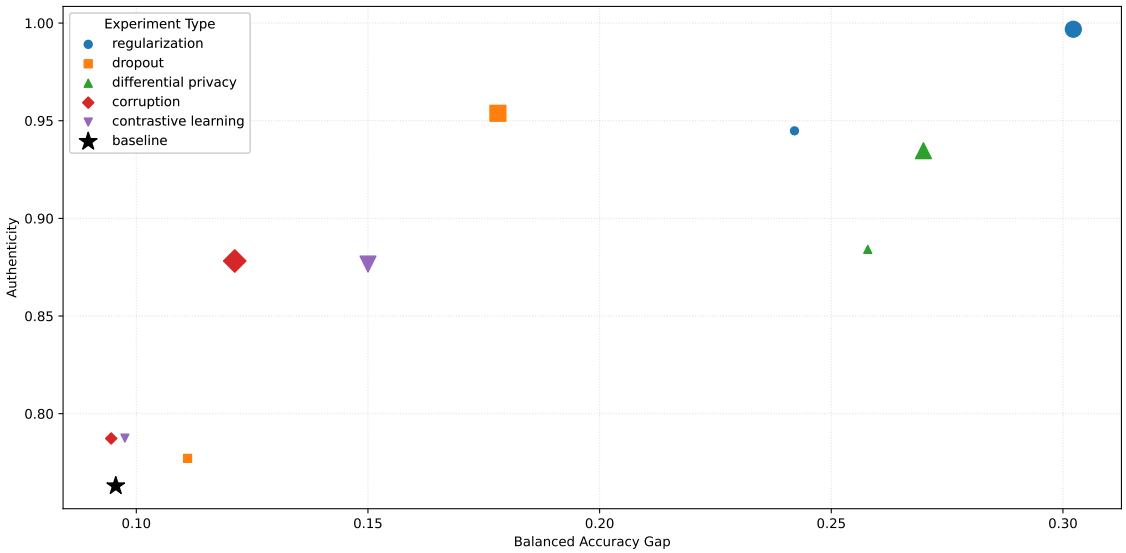


Figure 5.12: Balanced Accuracy Gap and Authenticity per Experiment type across datasets

The baseline VAE sits in the lower-left region of the plot, getting the smallest Balanced Accuracy Gap but also the lowest Authenticity. Several methods move upwards, showing better Authenticity, but often also move to the right, showing a bigger drop in predictive performance. This suggests that making synthetic samples less similar to the training samples, that is reducing memorization and improving privacy usually comes at the cost of keeping the information needed for classification.

Looking at each method's progression makes this trade-off even clearer. As the modifications get more aggressive, their points generally move towards the upper-right corner of the plot. This is especially clear for regularization and differential privacy. In contrast, corruption, contrastive learning, and dropout (to a lesser extent) show a more favorable progression as they gain noticeable Authenticity while keeping gaps close to the baseline's.

Overall, the figure suggests that corruption and contrastive learning offer the best balance between the two goals. On the other hand, while aggressive regularization and differential privacy reach the highest Authenticity scores, they do so at the cost of a big drop in utility. This highlights the inherent trade-off between reducing memorization and keeping the predictive structure of the original data.

5.3 Metafeature Correlation Results

While the earlier sections focused on comparing how well the evaluated methods perform, this analysis looks more closely at the characteristics of the synthetic datasets behind that performance. For each generated synthetic dataset, we extracted a set of general and statistical metafeatures, along with the corresponding value of the evaluation metric being studied. We then calculated correlations separately for each experiment type, which let us find the metafeatures with the strongest positive and negative links to each metric.

So, the resulting plots do not describe the properties of the original datasets, and they do not show what types of datasets a given method performs well on. Instead, they describe the synthetic datasets produced by each method, giving insight into which structural properties of the generated data are consistently linked to higher or lower performance. Because of this, these analyses should be seen as exploratory, not causal. Strong correlations do not mean that a metafeature directly causes changes in the evaluation metric, but they can reveal patterns that help explain why some generated datasets, and therefore some methods, tend to get better results.

The results of these experiments is available in Tables A.13 through A.15 for deeper review.

5.3.1 Distribution

Figure 5.13 shows the metafeatures with the strongest positive and negative correlations with α-Precision for each experiment type.

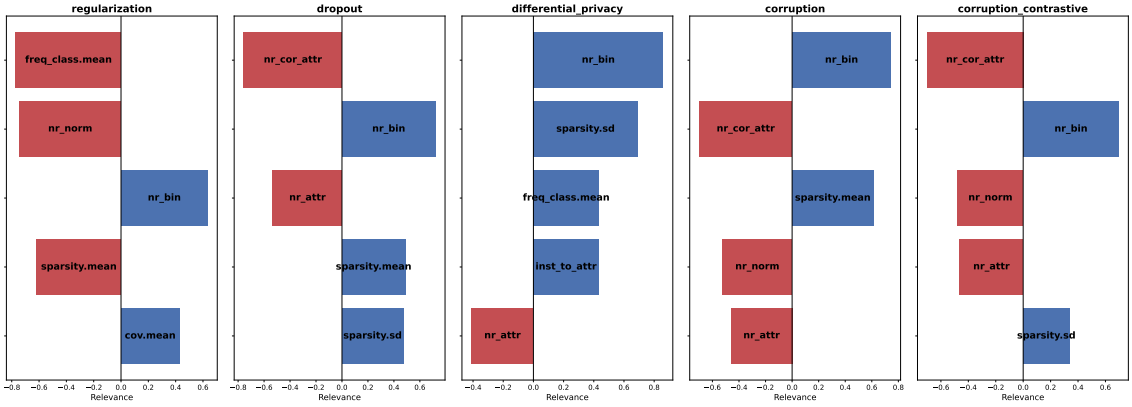


Figure 5.13: 5 most relevant dataset Metafeatures to α-Precision.

Although the exact rankings differ across methods, several metafeatures show up consistently, suggesting that higher α-Precision is linked to common structural characteristics of the generated datasets rather than to one specific generation strategy.

Among the positive correlations, the number of binary attributes (`nr_bin`) stands out the most, appearing in four of the experiment types. Similarly, the average and standard deviation of dataset sparsity (`sparsity.mean` and `sparsity.sd`), which measure the proportion and variability of zero-valued entries in the generated data, also show up repeatedly as positive indicators. On the other hand, the number of attributes (`nr_attr`) and the number of highly correlated attribute pairs (`nr_cor_attr`) consistently show negative correlations with α-Precision, while the number of approximately normally distributed attributes (`nr_norm`) shows up as a negative factor for several methods. Together, these recurring metafeatures suggest that the structural complexity of the generated datasets is closely tied to their ability to produce samples that stay within the support of the real data distribution.

Although we cannot draw causal conclusions, these patterns give some interesting insights into how the evaluated models behave. The consistent link between higher α-Precision and synthetic

datasets with more binary variables and greater sparsity may mean that methods which keep these structural characteristics are better at generating realistic samples without drifting away from the underlying data manifold. In contrast, bigger datasets and methods that produce datasets with stronger feature correlations may end up adding extra complexity, making it harder to accurately model the real data distribution.

Perhaps the most notable finding, though, is how consistent these relationships are across very different training methods. This suggests that the characteristics of the synthetic datasets themselves may tell us more about α-Precision than the mechanisms used to generate them.

5.3.2 Privacy

Figure 5.14 shows the metafeatures with the strongest positive and negative correlations with Authenticity for each experiment type.

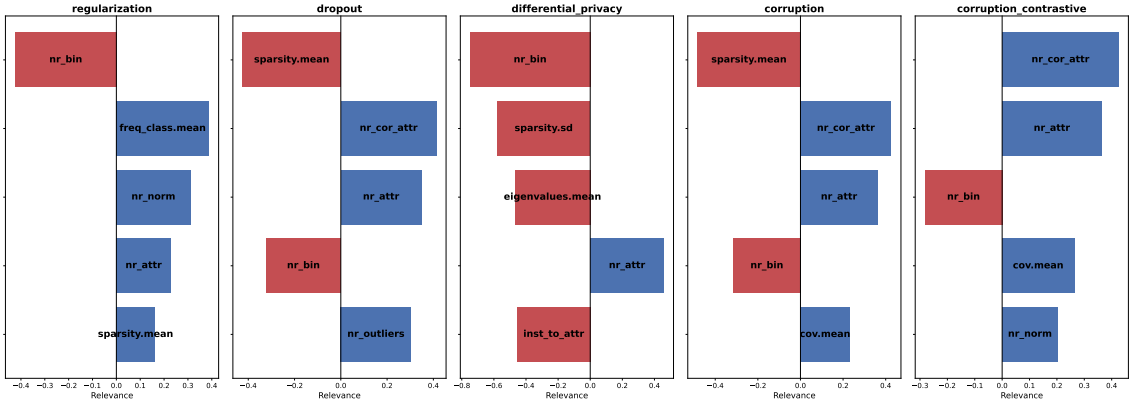


Figure 5.14: 5 most relevant dataset Metafeatures to Authenticity.

Unlike α-Precision, the patterns here are less consistent across methods, with fewer metafeatures repeating across all experiment types. This is perhaps expected, since Authenticity measures a privacy related property of the generated data, not how well it matches the real data distribution. Also, a higher Authenticity score should not be read on its own as a sign of a better synthetic dataset, since highly authentic samples could also come from unrealistic generations that simply do not resemble any real observations.

Among the patterns that do repeat, the number of attributes (`nr_attr`) and the number of highly correlated attribute pairs (`nr_cor_attr`) show up as positive correlations for dropout, corruption, and contrastive learning. This suggests that methods producing synthetic datasets with richer feature spaces and stronger dependencies between features tend to reach higher Authenticity. Measures related to covariance (`cov.mean`) also show up as positive indicators for corruption and contrastive learning.

On the other hand, the number of binary attributes (`nr_bin`) repeatedly shows up as a negative correlation, along with sparsity related measures (`sparsity.mean` and `sparsity.sd`) for

several experiment types. This means methods that generate sparser datasets with more binary features tend to get lower Authenticity.

Unlike α-Precision, where simpler structural representations seemed to help, Authenticity is generally linked to methods that generate datasets with more structural complexity, that is, more attributes, stronger relationships between features, and higher covariance. One possible explanation is that these characteristics reflect generators producing samples that are further from the actual training instances, making it less likely that synthetic records closely match real observations. At the same time, the fact that these findings seem to almost perfectly mirror the ones for α-Precision, together with the findings from the previous sections, suggests that the higher Authenticity could also come from synthetic data that strays too far from the true distribution, rather than from genuinely privacy-preserving generation.

5.3.3 Utility

Figure 5.15 shows the metafeatures with the strongest positive and negative correlations with the Balanced Accuracy Gap for each experiment type.

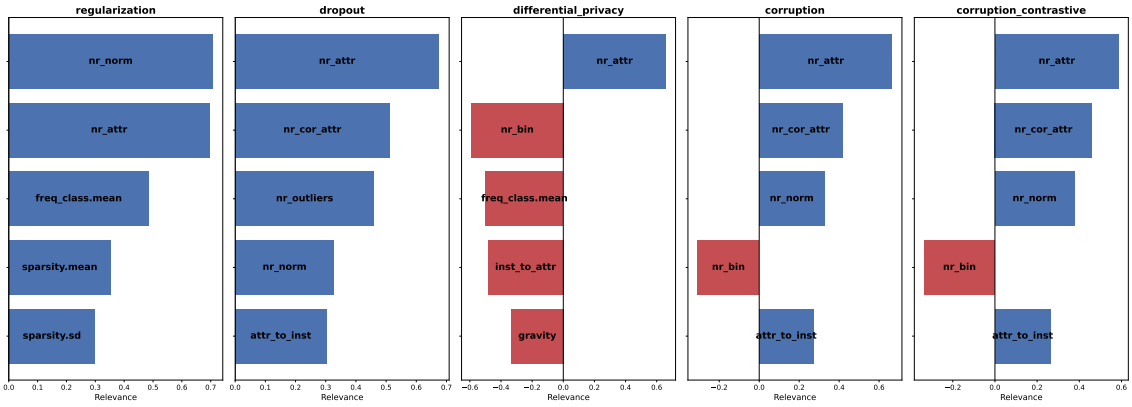


Figure 5.15: 5 most relevant dataset Metafeatures to Balanced Accuracy Gap.

Since this metric measures the difference in predictive performance between models trained on real and synthetic data, higher values mean a bigger drop in downstream machine learning utility. So, positive correlations are linked to characteristics of the generated datasets that go along with worse predictive performance, while negative correlations show properties linked to a smaller performance gap, and therefore better utility.

Several metafeatures show up consistently across the evaluated methods. The number of attributes (`nr_attr`) is the strongest positive correlation, appearing across all experiment types, often together with the number of highly correlated attribute pairs (`nr_cor_attr`) and the number of approximately normally distributed attributes (`nr_norm`). This suggests that methods producing synthetic datasets with higher dimensionality and stronger feature dependencies tend to lose more downstream predictive performance. On the other hand, the number of binary attributes (`nr_bin`) repeatedly shows up as a negative correlation. This means generators that produce

datasets with more binary variables tend to keep predictive utility better. Other metafeatures, like sparsity and class frequency statistics, show up for individual methods but not as consistently across experiment types.

These recurring patterns suggest that the structural complexity of the generated datasets is closely tied to how useful they are for downstream machine learning tasks. Methods that generate datasets with larger feature spaces, stronger correlations between features, and more normally distributed attributes seem more likely to distort the relationships needed for accurate predictive modeling, leading to bigger utility losses compared to models trained on real data.

These findings do not prove a causal relationship, but they suggest that keeping simpler and more stable structural characteristics may indirectly help maintain the predictive information needed for effective downstream learning. More broadly, since these correlations are consistent across several different generation strategies, downstream utility may depend less on the specific method used and more on the statistical characteristics of the synthetic datasets those methods end up producing, which may be a feature of the original datasets themselves.

5.4 Discussion

This chapter evaluated the proposed perturbation strategies by examining their impact on synthetic data quality, downstream utility, privacy, and the characteristics of the learned data distributions. Rather than identifying a single universally superior methodology, the results reveal how different perturbation mechanisms alter the balance between preserving utility and reducing information leakage. The findings are discussed below in relation to the research questions introduced in 1.3.3.

Before discussing the individual research questions, it is important to note that the comparisons presented throughout this chapter describe the behaviour of each perturbation strategy only for the hyperparameter configurations evaluated in this study, rather than establishing an inherent ranking of the methods themselves. Hyperparameters such as dropout rate, regularization weight, corruption ratio, and the differential privacy budget operate on different scales and influence different components of the learning process. Consequently, there is no principled way to define equivalent perturbation strengths across methods or to claim that a particular configuration of one method is directly comparable to a configuration of another. The relative behavior reported throughout this chapter should therefore be interpreted as an empirical comparison over the evaluated parameter ranges, rather than as evidence that one perturbation mechanism is inherently more or less disruptive than another.

RQ1: What impact does each method have on the distributional quality and downstream utility of the generated synthetic data compared to the unperturbed baseline VAE?

Across all evaluated methodologies, increasing perturbation strength generally reduced both distributional fidelity and downstream predictive utility relative to the unperturbed VAE. Although the extent of this degradation varied across methods, the overall trend remained consistent, with stronger perturbations producing larger reductions in synthetic data quality.

Among the evaluated approaches, dropout, corruption and contrastive learning consistently remained closest to the baseline across both the distribution based quality metrics and downstream utility measures. Overall, these results indicate that moderate perturbation strategies preserve the characteristics of the baseline VAE considerably better than more aggressive privacy oriented approaches.

RQ2: To what extent does each perturbation strategy improve privacy preservation of the generated synthetic data compared to the unperturbed baseline VAE?

Although all evaluated perturbation strategies provided some degree of additional privacy relative to the baseline VAE, the improvements were generally modest and no methodology emerged as consistently superior across the privacy evaluation. The principal reason for this appears to be that the baseline VAE itself already exhibits limited fidelity to the original data distribution. Since the perturbation strategies build upon this baseline, their privacy improvements are achieved on top of a model that already generates synthetic data with imperfect distributional quality.

Furthermore, considering the privacy and fidelity metrics jointly suggests that the observed privacy improvements are closely associated with larger deviations from the original data distribution. Rather than selectively reducing information leakage while preserving representative samples, the evaluated methods (including the baseline) appear to have good privacy performance primarily by generating synthetic observations that lie further from the true data distribution. Consequently, the measured privacy gains should be interpreted with caution, as they are likely driven, at least in part, by reduced representativeness of the generated data rather than by fundamentally more privacy-preserving data generation.

RQ3: What trade-offs between privacy and generative quality/utility emerge across the different methods studied, and which of them offer the best (or worst) such trade-off?

The experiments demonstrate a clear trade-off between privacy and utility. Methods producing greater privacy improvements generally exhibit larger reductions in distributional fidelity and downstream predictive utility, indicating that increasing privacy comes at the cost of generating synthetic data that more closely departs from the original distribution.

Among the evaluated methodologies, dropout, corruption and contrastive learning consistently provide the most favourable balance between synthetic data quality, downstream utility and privacy. These approaches remain comparatively close to the baseline while still providing measurable privacy improvements.

RQ4: How do these methods affect the underlying data generation process, and what patterns across the evaluation metrics reveal about how these approaches work?

The behavior of the evaluated methodologies provides insight into how the different perturbation mechanisms interact with the VAE learning process.

The strong performance of dropout is perhaps the most intuitive, since it introduces only a relatively mild perturbation during training and therefore has limited impact on the learned latent representation. The behavior of corruption is initially less obvious because corrupted feature values are replaced by uniformly sampled values. However, since corruption is applied after quantile normalization, the replacement values are sampled from a compressed feature range, substantially limiting how far corrupted values deviate from regions where real data are concentrated. This appears to explain why corruption introduces considerably less degradation than might otherwise be expected.

Adding contrastive learning on top of corruption produces only modest changes across the evaluated metrics and hyperparameter settings, suggesting that, within the scope of this work, the additional contrastive objective neither substantially improves nor degrades the behavior already induced by corruption.

Several evaluation metrics also reveal consistent characteristics of the learned data distributions. The disparity between α-Precision and β-Recall indicates that the generators model dense, highly populated regions considerably more accurately than less populated regions of the data distribution. Likewise, the relationship between Alignment and Balanced Accuracy Gap suggests that the information required for downstream predictive tasks is concentrated primarily within these well represented regions, in this case, in the majority classes.

RQ5: Within the synthetic data produced by each perturbation strategy, are structural properties consistently associated with better or worse performance on a given evaluation metric, helping explain why certain configurations outperform others?

The metafeature correlation analyses identify several structural properties that are consistently associated with synthetic data quality across multiple evaluation metrics. Better performing synthetic datasets generally exhibit characteristics associated with lower complexity, including fewer attributes, weaker inter-feature dependencies, greater sparsity and a higher prevalence of binary variables.

Because these metafeatures are extracted from the generated datasets themselves, they suggest that methodologies achieving stronger performance tend to produce comparatively simpler structural representations. At the same time, these structural characteristics are also influenced

by the original datasets from which the synthetic data is generated. Consequently, the observed relationships likely reflect both the intrinsic complexity of the original data and the characteristics of the learned synthetic representation.

Overall, the results consistently indicate that the adopted VAE framework models simpler data distributions more effectively, while increasingly complex distributions remain considerably more difficult to reproduce faithfully regardless of the perturbation strategy employed.

Chapter 6

Conclusion

This thesis empirically surveys and compares a set of established perturbation strategies, applied on top of a common baseline generative model, in terms of their ability to improve the privacy of synthetic tabular data while preserving its distributional quality and downstream utility. The remainder of this chapter synthesizes the main findings of the study and addresses its limitations and potential directions for future research. The chapter concludes with some final remarks on the work as a whole.

6.1 Main Findings

This work investigated the impact of several perturbation strategies on the quality, privacy and utility of synthetic tabular data generated used on top of a Variational Autoencoder. Collectively, the results provide a comprehensive empirical comparison of these approaches across multiple evaluation perspectives, including distributional fidelity, downstream predictive utility, privacy, and structural characteristics of the generated data. Rather than identifying a universally superior methodology, the evaluation highlights the strengths and weaknesses of each approach and provides insight into the privacy-utility trade-offs that arise when introducing privacy-preserving mechanisms into the data generation process.

Across the evaluated methodologies, dropout, corruption and contrastive learning consistently produced the most favorable overall performance. These approaches closely matched the baseline VAE in terms of distributional quality and downstream utility while still providing measurable improvements in privacy. Although stronger perturbation strategies generally achieved larger privacy gains, these came at the expense of substantially greater degradation in the quality of the generated data, reinforcing the inherent trade-off between privacy and utility.

The evaluation also provides insight into how these perturbation strategies influence the learned data distribution. The results suggest that the observed privacy improvements are largely associated with increasing departures from the original data distribution, while the metafeature analysis indicates that simpler synthetic data representations are consistently associated with stronger performance across multiple evaluation metrics. Together, these findings contribute to a better

understanding of the mechanisms through which different perturbation strategies affect synthetic data generation.

Finally, the results also demonstrate that the choice of the underlying generative model plays an important role in determining the overall performance of privacy-preserving synthetic data generation. The chosen baseline VAE already exhibits limitations in faithfully modeling the original data distribution, which naturally constrains the performance that can be achieved by any perturbation strategy built upon it. Nevertheless, the proposed methodologies were evaluated under a common experimental framework, allowing meaningful comparisons between them and providing valuable insight into their relative behavior. The implications of this limitation, together with potential improvements are discussed further in the following section.

6.2 Limitations and Future Work

6.2.1 Limitations

The biggest limitation of this thesis comes from the baseline generative model used throughout. As discussed in the previous section, several of the shortcomings we saw across all evaluated methods, most notably the limited distributional fidelity and the weak representation of low density regions of the data, are already present in the unperturbed baseline VAE. We deliberately kept the model simple (two fully connected layers of 128 neurons in both the encoder and decoder, and a 32-dimensional latent space as described in Section 3.1) so that any effects we saw could be attributed to the perturbation strategies rather than to architectural complexity. However, this choice also limits what any of the perturbation strategies studied could realistically achieve on top of it. A model with more capacity, additional or wider fully connected layers, a larger latent space, or a longer or more carefully tuned training schedule might give a stronger and more faithful baseline distribution. Against that stronger baseline, we could re-evaluate the same perturbation strategies with more opportunity to reveal a genuine privacy-utility trade-off, rather than one confused by an already limited baseline fit.

Additionally, this project only tests a single generative architecture as its baseline. While the perturbation strategies studied are, in principle, model-agnostic, we have not tested whether the specific findings of this project generalize for other generative architectures (GAN based, diffusion based, or transformer based tabular generators, among others). It is possible that a generative model with different inductive biases would respond differently to the same perturbations, or would give a baseline fit strong enough to more clearly separate genuine privacy improvements from mere loss of fidelity.

A further limitation concerns the training and preprocessing choices held fixed across all datasets and methods for the sake of a controlled comparison (Subsection 4.2.1). This was a deliberate design choice to isolate the effect of each perturbation strategy, but it also means no dataset-specific tuning was performed. Training hyperparameters such as the number of epochs,

batch size scaling, and learning rate were applied uniformly, and the quantile normalization preprocessing itself followed the same fixed procedure regardless of a given dataset's size, feature composition, or distributional shape. It is possible that some datasets would benefit from a training configuration or preprocessing choice tailored to their specific characteristics, rather than the single, uniformly applied pipeline used here.

A related limitation concerns the comparisons made between the perturbation strategies themselves. Dropout rate, regularization weight, corruption ratio, and the differential privacy budget act on different parts of the model and on entirely different scales, and there is no principled way to convert a value of one into an equivalent value of another. Because of this, statements like "regularization causes a bigger drop than dropout" describe how these specific tested configurations behaved relative to each other, not a theoretical claim that one mechanism is inherently more disruptive than another. A different, untested set of hyperparameter values for either method could change this comparison. The conclusions drawn throughout this project about which methods perform best or worst should therefore be read as empirical observations over the specific ranges tested, not as conclusive evidence that one method is objectively better than another from a theoretical standpoint.

Finally, the evaluation metrics used in this project, while covering three complementary axes, do not provide fully conclusive evidence to separate genuine privacy protection from a general loss of distributional fidelity on their own. The privacy metrics we used, Authenticity and Privacy Loss, are both based on nearest neighbor distances in a shared feature space, rather than on a direct simulation of a realistic adversary trying to infer training set membership. A dedicated membership inference attack, trained and evaluated as its own classifier, was outside the scope of this project. Similarly, the metafeature correlation analysis is explicitly exploratory and correlational, and cannot prove that the structural properties we identified are a cause, rather than just a consequence or a byproduct, of the performance patterns we saw.

6.2.2 Future Work

Building directly on the limitations above, several directions for future work follow naturally. The most important is repeating this evaluation with a baseline generative model of greater capacity to test whether the trade-offs of the studied methods found in this project remain valid, become more pronounced, or get weaker once the baseline itself fits the real data distribution more faithfully.

Since the perturbation strategies studied are model-agnostic, a natural next step is repeating this evaluation using other generative architectures as the baseline to check whether the findings of this project, particularly the apparent link between privacy gains and distributional drift, are a general property of these perturbation strategies, or are instead specific to the VAE architecture used here.

Future work could also extend the dataset scope of this thesis, both by increasing the number of datasets evaluated and by looking beyond healthcare into other sensitive domains, to check

whether the trends found here generalize to tabular data from other applications with different structural characteristics.

On the evaluation side, this project points to a broader, ongoing challenge in the field. Current metrics for jointly assessing the privacy and quality of synthetic data are not straightforward to interpret, and, as shown throughout this project, can point in different directions depending on which metric you look at independently. Future work could explore developing metrics specifically designed to separate genuine privacy protection from general distributional drift, rather than relying, as this project had to, on jointly interpreting several metrics that were not individually designed for this purpose. Along these lines, adding a dedicated membership inference attack, built and evaluated as its own adversarial classifier rather than inferred indirectly from nearest neighbor distances, would give a more direct and realistic empirical measure of privacy risk.

Finally, since the perturbation strategies studied act on different, independent levels of the model, training objective, architecture, optimization procedure, and input data, and are model-agnostic rather than tied to any specific part of the VAE, they can in principle be freely combined with one another. Future work could explore combining multiple perturbation strategies at once, for instance applying differential privacy alongside dropout or corruption, to check whether combining mechanisms that act at different levels can achieve a better privacy-utility trade-off than any single strategy applied on its own, as done throughout this project.

6.3 Final Remarks

This project started from a simple and practical problem: synthetic data promises a way to share and study sensitive health data more freely, but the same generative flexibility that makes synthetic data useful can also cause it to reveal information about the individuals present in the original training data (Section 1.1). Rather than proposing a new technique to solve this issue, this thesis surveyed several existing and conceptually different ways of addressing it, applied consistently on top of one deliberately simple baseline, and asked which of them, if any, offer an improvement in the preservation of private information with minimal impact on the broader generative task.

The answer this thesis reaches is a cautious one, and arguably a more honest reflection of where the field stands than a single clean recommendation would be. Some perturbation strategies, especially dropout, corruption, and contrastive learning, consistently matched of the baseline's distributional quality and downstream utility while still shifting privacy related metrics in a favorable direction. Others, including differential privacy, came at a steeper cost without clearly better privacy outcomes. Just as importantly, this project's results repeatedly show that the privacy and quality metrics used here are not always straightforward to read together and that some apparent privacy gains are hard to fully separate from a general drift away from the real data distribution. This is not a shortcoming unique to the methods studied here, but a reflection of a broader, still open challenge in evaluating privacy-preserving synthetic data.

If this project has one main takeaway, it is that comparing privacy-preserving techniques fairly means holding as much else constant as possible, and reading the resulting metrics together rather than on their own. Hopefully, the comparative framework developed here, and the trade-offs it reveals, offer a useful and honest reference point for future work on privacy-preserving synthetic data generation for tabular data, whether that work builds directly on the perturbation strategies studied in this thesis or turns to the further directions outlined in Subsection 6.2.2.

References

[1] European Parliament and Council of the European Union. *Regulation (EU) 2016/679 (General Data Protection Regulation)*. Official Journal of the European Union. Apr. 27, 2016. URL: https://eur-lex.europa.eu/eli/reg/2016/679/oj.

[2] U.S. Congress. *Health Insurance Portability and Accountability Act of 1996*. Public Law 104-191. 1996. URL: https://www.congress.gov/bill/104th-congress/house-bill/3103.

[3] Mikel Hernandez, Gorka Epelde, Ane Alberdi, Roldán Cilla, and Debbie Rankin. "Synthetic Data Generation for Tabular Health Records: A Systematic Review". In: *Neurocomputing* 493 (2022), pp. 344–363. DOI: 10.1016/j.neucom.2022.04.053.

[4] Khaled El Emam, Elizabeth Jonker, Luk Arbuckle, and Bradley Malin. "A systematic review of re-identification attacks on health data". In: *PloS one* 6.12 (2011), e28071.

[5] Nazia Nafis, Inaki Esnaola, Alvaro Martinez-Perez, Maria-Cruz Villa-Uriol, and Venet Osmani. "Critical challenges and guidelines in evaluating synthetic tabular data: A systematic review". In: *arXiv preprint arXiv:2504.18544* (2025).

[6] Ankai Wang, Stefanka Chukova, and Binh Nguyen. "Challenges and Opportunities of Generative Models on Tabular Data". In: *Applied Soft Computing* 166 (2024), p. 112223. DOI: 10.1016/j.asoc.2024.112223.

[7] Neha Patki, Roy Wedge, and Kalyan Veeramachaneni. "The Synthetic Data Vault". In: *IEEE International Conference on Data Science and Advanced Analytics (DSAA)*. 2016, pp. 399–410. DOI: 10.1109/DSAA.2016.49.

[8] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. "PrivBayes: Private Data Release via Bayesian Networks". In: *ACM Transactions on Database Systems* 42.4 (2017), pp. 1–41. DOI: 10.1145/3134428.

[9] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[10] 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*. PMLR. 2017, pp. 286–305.

[11] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. "Modeling Tabular Data using Conditional GAN". In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 32. 2019, pp. 7335–7345. DOI: 10.48550/arXiv.1907.00503.

[12] Julian Höllig and Michaela Geierhos. "Utility Meets Privacy: A Critical Evaluation of Tabular Data Synthesizers". In: *IEEE Access* 13 (2025), pp. 44497–44509. DOI: 10.1109/ACCESS.2025.3549680.

[13] Fida Kamal Dankar, Mahmoud K. Ibrahim, and Lina Ismail. "A Multi-Dimensional Evaluation of Synthetic Data Generators". In: *IEEE Access* 10 (2022), pp. 11147–11158. DOI: 10.1109/ACCESS.2022.3144765.

[14] Debalina Padariya. "Assessing Privacy-Utility Trade-offs in Generative Model-based Synthetic Tabular Datasets". In: *2025 30th International Conference on Automation and Computing (ICAC)*. 2025. DOI: 10.1109/ICAC65379.2025.11196133.

[15] David Watson, Kristin Blesch, Jan Kapar, and Marvin N. Wright. "Adversarial Random Forests for Density Estimation and Generative Modeling". In: *Proceedings of the 26th International Conference on Artificial Intelligence and Statistics (AISTATS)*. Vol. 206. 2023, pp. 5357–5375.

[16] Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. "Language Models are Realistic Tabular Data Generators". In: *11th International Conference on Learning Representations (ICLR)*. 2022.

[17] Jooho Lee, Migwi Kim, Yongjun Jeong, and Yong Man Ro. "Differentially Private Normalizing Flows for Synthetic Tabular Data Generation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 36. 7. 2022, pp. 7345–7353. DOI: 10.1609/aaai.v36i7.20697.

[18] Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. "Tabddpm: Modelling tabular data with diffusion models". In: *International conference on machine learning*. PMLR. 2023, pp. 17564–17579.

[19] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. "SMOTE: Synthetic Minority Over-sampling Technique". In: *Journal of Artificial Intelligence Research* 16 (2002), pp. 321–357.

[20] Diederik P Kingma and Max Welling. "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114* (2013).

[21] Benjamin Hilprecht, Martin Härterich, and Daniel Bernau. "Monte carlo and reconstruction membership inference attacks against generative models". In: *Proceedings on Privacy Enhancing Technologies* (2019).

[22] Qingrong Chen, Chong Xiang, Minhui Xue, Bo Li, Nikita Borisov, Dali Kaarfar, and Haojin Zhu. "Differentially private data generative models". In: *arXiv preprint arXiv:1812.02274* (2018).

[23] Daniel Bernau, Jonas Robl, and Florian Kerschbaum. "Assessing differentially private variational autoencoders under membership inference". In: *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer. 2022, pp. 3–14.

[24] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. "Calibrating Noise to Sensitivity in Private Data Analysis". In: *Theory of Cryptography Conference (TCC)*. Springer, 2006, pp. 265–284.

[25] Cynthia Dwork. "Differential Privacy: A Survey of Results". In: *Theory and Applications of Models of Computation*. Lecture Notes in Computer Science. Springer, 2008, pp. 1–19.

[26] 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 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. 2016, pp. 308–318. DOI: 10.1145/2976749.2978318.

[27] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. "Differentially Private Generative Adversarial Network". In: arXiv:1802.06739. 2018.

[28] Aditya Kunar, Robert Birke, Zhen Zhao, and Lydia Chen. "DTGAN: Differential Private Training for Tabular GANs". In: (2021). arXiv:2107.02521.

[29] James Jordon, Jinsung Yoon, and Mihaela van der Schaar. "PATE-GAN: Generating Synthetic Data with Differential Privacy Guarantees". In: *International Conference on Learning Representations (ICLR)*. 2019.

[30] Ming Li, Dingsu Zhuang, and Jie M. Chang. "MC-GEN: Multi-level Clustering for Private Synthetic Data Generation". In: *Knowledge-Based Systems* 264 (2023), p. 110239. DOI: 10.1016/j.knosys.2022.110239.

[31] Irinaldo Capitulino de Souza, Carolina Lopes de Carvalho Vital, Rodrigo Viana Correira de Souza, Maria Fernanda Bandeira da Silva, Eufrásio de Andrade Lima Neto, Sadraque Eneas de Figueiredo Lucena, Suellen Mary Marinho dos Santos Andrade, and Ana Hermínia Andrade e Silva. "A Multidimensional Evaluation of Synthetic Data Generators for Small Datasets: Fidelity, Utility, and the Privacy Trade-off". In: *Journal of the Brazilian Computer Society* (202X). In press.

[32] Yuchao Tao, Ryan McKenna, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. "Benchmarking Differentially Private Synthetic Data Generation Algorithms". In: (2021). arXiv:2112.09238.

[33] Lucas Rosenblatt, Tianhao Liu, Julia Stoyanovich, and Carlos Scheidegger. "Epistemic Parity: Reproducibility as an Evaluation Metric for Differential Privacy". In: *Proceedings of the VLDB Endowment* 16.12 (2023), pp. 2862–2874. DOI: 10.48550/arXiv.2208.12700.

[34] Khaled El Emam, Lucy Mosquera, and Jason Bass. "Evaluating Identity Disclosure Risk in Fully Synthetic Health Data: Model Development and Validation". In: *Journal of Medical Internet Research* 22.11 (2020), e23139. DOI: 10.2196/23139.

[35] Andrew Yale, Saloni Dash, Ritik Dutta, Isabelle Guyon, Adrien Pavao, and Kristin P Bennett. "Privacy preserving synthetic health data". In: *ESANN 2019-European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*. 2019.

[36] Ahmed Alaa, Boris Van Breugel, Evgeny S Saveliev, and Mihaela Van Der Schaar. "How faithful is your synthetic data? sample-level metrics for evaluating and auditing generative models". In: *International conference on machine learning*. PMLR. 2022, pp. 290–306.

[37] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. "Membership Inference Attacks Against Machine Learning Models". In: *2017 IEEE Symposium on Security and Privacy (S&P)*. 2017, pp. 3–18.

[38] Dingfan Chen, Ning Yu, Yang Zhang, and Mario Fritz. "GAN-Leaks: A Taxonomy of Membership Inference Attacks against Generative Models". In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security*. 2020, pp. 343–362. DOI: 10.1145/3372297.3417238.

[39] Zhiyu Zhang, Chao Yan, and Bradley A. Malin. "Membership Inference Attacks Against Synthetic Health Data". In: *Journal of Biomedical Informatics* 125 (2022), p. 103977. DOI: 10.1016/j.jbi.2021.103977.

[40] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. "Synthetic Data — Anonymisation Groundhog Day?" In: *Proceedings on Privacy Enhancing Technologies*. Vol. 2022. 4. 2022, pp. 145–166. DOI: 10.48550/arXiv.2011.07018.

[41] Luc Rocher, Julien M. Hendrickx, and Yves-Alexandre De Montjoye. "Estimating the Success of Re-identifications in Incomplete Datasets Using Generative Models". In: *Nature Communications* 10.1 (2019), p. 3069. DOI: 10.1038/s41467-019-10933-3.

[42] Davinder Kaur, Matt Sobiesk, Sonal Patil, Jing Liu, Preethi Bhagat, Arpit Gupta, and Natalia Markuzon. "Application of Bayesian Networks to Generate Synthetic Health Data". In: *Journal of the American Medical Informatics Association* 28.4 (2021), pp. 801–811. DOI: 10.1093/jamia/ocaa303.

[43] Dong-Yoon Park and Il-Yeon Ko. "An Empirical Study of Utility and Disclosure Risk for Tabular Data Synthesis Models". In: *IEEE International Conference on Big Data and Smart Computing (BigComp)*. 2024, pp. 67–74. DOI: 10.1109/bigcomp60711.2024.00020.

[44] Katherine S. Button, John P. Ioannidis, Claire Mokrysz, Brian A. Nosek, Jonathan Flint, Emma S. J. Robinson, and Marcus R. Munafò. "Power Failure: Why Small Sample Size Undermines the Reliability of Neuroscience". In: *Nature Reviews Neuroscience* 14.5 (2013), pp. 365–376. DOI: 10.1038/nrn3475.

[45] Aldren Gonzales, Govind Guruswamy, and Scott R. Smith. "Synthetic Data in Health Care: A Narrative Review". In: *PLOS Digital Health* 2.1 (2023), e0000082. DOI: 10.1371/journal.pdig.0000082.

[46] Joshua Snoke, Gillian M. Raab, Beata Nowok, Chris Dibben, and Aleksandra Slavkovic. "General and Specific Utility Measures for Synthetic Data". In: *Journal of the Royal Statistical Society: Series A* 181.3 (2018), pp. 663–688. DOI: 10.48550/arXiv.1604.06651.

[47] Fida Kamal Dankar and Mahmoud Ibrahim. "Fake It Till You Make It: Guidelines for Effective Synthetic Data Generation". In: *Applied Sciences* 11.5 (2021), p. 2158. DOI: 10.3390/app11052158.

[48] Vikram S. Chundawat, Ayush K. Tarun, Murari Mandal, Monika Lahoti, and Pankaj Narang. "A Universal Metric for Robust Evaluation of Synthetic Tabular Data". In: *IEEE Transactions on Artificial Intelligence* 5.1 (2024), pp. 300–309. DOI: 10.1109/TAI.2022.3229289.

[49] Alvaro Figueira and Bruno Vaz. "Survey on Synthetic Data Generation, Evaluation Methods and GANs". In: *Mathematics* 10.15 (2022), p. 2733. DOI: 10.3390/math10152733.

[50] Yucen Lu, Meng Shen, Hao Wang, Xin Wang, Conor van Rechem, Tian Fu, and Wei Wei. "Machine Learning for Synthetic Data Generation: A Review". In: (2025). arXiv:2302.04062.

[51] Yefeng Yuan, Yuhong Liu, and Liang Cheng. "A multi-faceted evaluation framework for assessing synthetic data generated by large language models". In: *arXiv preprint arXiv:2404.14445* (2024).

[52] Michael Stenger, Robert Leppich, Ian Foster, Samuel Kounev, and André Bauer. "Evaluation is key: a survey on evaluation measures for synthetic time series". In: *Journal of Big Data* 11.1 (2024), p. 66.

[53] Andre Goncalves, Priyadip Ray, Braden Soper, Jennifer Stevens, Linda Coyle, and Adam P. Sales. "Generation and Evaluation of Synthetic Patient Data". In: *BMC Medical Research Methodology* 20 (2020), p. 108. DOI: 10.1186/s12874-020-00977-1.

[54] Huma Murtaza, Mukhtar Ahmed, and Noman Khan. "Synthetic Data Generation: State of the Art in Health Care Domain". In: *Computer Science Review* 48 (2023), p. 100546. DOI: 10.1016/j.cosrev.2023.100546.

[55] Marcos Rujas, Rodrigo Martín Gómez del Moral Herranz, Giuseppe Fico, and Beatriz Merino-Barbancho. "Synthetic Data Generation in Healthcare: A Scoping Review of Reviews on Domains, Motivations, and Future Applications". In: *International Journal of Medical Informatics* 195 (2025), p. 105763. DOI: 10.1016/j.ijmedinf.2024.105763.

[56] Steven M. Bellovin, Preetam K. Dutta, and Nathan Reitinger. "Privacy and Synthetic Datasets". In: *Stanford Technology Law Review* 22.1 (2019), pp. 1–52.

[57] Ian Goodfellow. *Deep learning*. 2016.

[58] Andrew Y Ng. "Feature selection, L 1 vs. L 2 regularization, and rotational invariance". In: *Proceedings of the twenty-first international conference on Machine learning*. 2004, p. 78.

[59] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[60] Geoffrey E Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, and Ruslan R Salakhutdinov. "Improving neural networks by preventing co-adaptation of feature detectors". In: *arXiv preprint arXiv:1207.0580* (2012).

[61] Cynthia Dwork and Aaron Roth. "The algorithmic foundations of differential privacy". In: *Foundations and trends® in theoretical computer science* 9.3-4 (2014), pp. 211–487.

[62] Ashkan Yousefpour, Igor Shilov, Alexandre Sablayrolles, Davide Testuggine, Karthik Prasad, Mani Malek, John Nguyen, Sayan Ghosh, Akash Bharadwaj, Jessica Zhao, et al. "Opacus: User-friendly differential privacy library in PyTorch". In: *arXiv preprint arXiv:2109.12298* (2021).

[63] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. "Extracting and composing robust features with denoising autoencoders". In: *Proceedings of the 25th international conference on Machine learning*. 2008, pp. 1096–1103.

[64] Dara Bahri, Heinrich Jiang, Yi Tay, and Donald Metzler. "Scarf: Self-supervised contrastive learning using random feature corruption". In: *arXiv preprint arXiv:2106.15147* (2021).

[65] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. "Representation learning with contrastive predictive coding". In: *arXiv preprint arXiv:1807.03748* (2018).

[66] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. "A simple framework for contrastive learning of visual representations". In: *International conference on machine learning*. PmLR. 2020, pp. 1597–1607.

[67] Max Little. *Parkinsons*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C59C74. 2007.

[68] Matjaz Zwitter and Milan Soklic. *Breast Cancer*. UCI Machine Learning Repository. DOI: https://doi.org/10.24432/C51P4M. 1988.

[69] Ralf Lichtinghagen, Frank Klawonn, and Georg Hoffmann. *HCV Data*. UCI Machine Learning Repository. 2020. DOI: 10.24432/C5D612. URL: https://archive.ics.uci.edu/dataset/571/hcv+data.

[70] Ehab Aboelnaga. *Diabetes*. https://www.kaggle.com/datasets/ehababoelnaga/diabetes-dataset. Kaggle dataset. Accessed: 2026-07-30. 2024.

[71] Andras Janosi, William Steinbrunn, Matthias Pfisterer, Robert Detrano, et al. *Heart Disease*. https://archive.ics.uci.edu/dataset/45/heart+disease. UCI Machine Learning Repository. Accessed: 2026-07-30. 1988.

[72] D. Ayres-de-Campos, J. Bernardes, A. Garrido, J. Marques-de-Sá, and L. Pereira-Leite. "SisPorto 2.0: A Program for Automated Analysis of Cardiotocograms". In: *Journal of Maternal-Fetal Medicine* 9.5 (2000), pp. 311–318. DOI: 10.1002/1520-6661(200009/10)9:5<311::AID-MFM7>3.0.CO;2-9.

[73] fedesoriano. *Stroke Prediction Dataset*. https://www.kaggle.com/datasets/fedesoriano/stroke-prediction-dataset. Kaggle dataset. Accessed: 2026-07-30. 2021.

[74] Dina Sulimanova. *Cardiovascular Disease Dataset*. https://www.kaggle.com/datasets/sulianova/cardiovascular-disease-dataset. Kaggle dataset. Accessed: 2026-07-30. 2019.

[75] Diederik P Kingma and Jimmy Ba. "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980* (2014).

[76] Zhaozhi Qian, Bogdan-Constantin Cebere, and Mihaela van der Schaar. *Synthcity: facilitating innovative use cases of synthetic data in different data modalities*. 2023. DOI: 10.48550/ARXIV.2301.07573. URL: https://arxiv.org/abs/2301.07573.

[77] *Synthetic Data Metrics*. Version 0.12.0. DataCebo, Inc. Oct. 2023. URL: https://docs.sdv.dev/sdmetrics/.

[78] Tianqi Chen and Carlos Guestrin. "Xgboost: A scalable tree boosting system". In: *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. 2016, pp. 785–794.

Appendix A

Additional Results

A.1 Single Metric Results

In this section, we report the complete experimental results underlying the aggregated analysis presented in the main text. Table A.1 provides a legend describing each method code used throughout this section of the appendix. The subsequent tables (one per metric) report, for each experiment, the per dataset results alongside the average and standard deviation across datasets.

Table A.1: Description of each experiment configuration.

Code	Description
`baseline`	Baseline VAE Model
`method1`	Regularization with l1 = 0 and l2 = 0.0001
`method2`	Regularization with l1 = 0 and l2 = 0.001
`method3`	Regularization with l1 = 0.0001 and l2 = 0
`method4`	Regularization with l1 = 0.0001 and l2 = 0.0001
`method5`	Regularization with l1 = 0.0001 and l2 = 0.001
`method6`	Regularization with l1 = 0.001 and l2 = 0
`method7`	Regularization with l1 = 0.001 and l2 = 0.0001
`method8`	Regularization with l1 = 0.001 and l2 = 0.001
`method9`	Dropout with dropout rate = 10%
`method10`	Dropout with dropout rate = 25%
`method11`	Dropout with dropout rate = 50%
`method12`	Dropout with dropout rate = 75%
`method13`	Differential Privacy with privacy budget = 2.0
`method14`	Differential Privacy with privacy budget = 1.0
`method15`	Differential Privacy with privacy budget = 0.5
`method16`	Differential Privacy with privacy budget = 0.1
`method17`	Corruption with corruption rate = 10%
`method18`	Corruption with corruption rate = 25%
`method19`	Corruption with corruption rate = 50%
`method20`	Corruption with corruption rate = 75%
`method21`	Contrastive Learning with corruption rate = 10%
`method22`	Contrastive Learning with corruption rate = 25%
`method23`	Contrastive Learning with corruption rate = 50%
`method24`	Contrastive Learning with corruption rate = 75%

A.1.1 Alpha Precision

Table A.2: Alpha Precision Results of each method on each dataset.

experiment	heart	hepatitis	stroke	parkinsons	fetal health	breast-cancer	diabetes	cardio	Average	Std
baseline	0.830	0.169	**0.926**	**0.261**	**0.551**	0.953	0.485	**0.883**	**0.632**	0.290
*method*1	0.816	0.152	0.905	0.184	0.441	0.943	0.475	0.841	0.595	0.302
*method*2	0.234	0.000	0.583	0.000	0.067	0.643	0.406	0.556	0.311	0.253
*method*3	0.443	0.000	0.750	0.000	0.067	0.680	**0.511**	0.197	0.331	0.285
*method*4	0.405	0.000	0.307	0.000	0.067	0.666	0.506	0.067	0.252	0.239
*method*5	0.243	0.000	0.067	0.000	0.067	0.439	0.229	0.067	0.139	0.143
*method*6	0.104	0.000	0.067	0.000	0.067	0.091	0.329	0.067	0.091	0.097
*method*7	0.091	0.000	0.067	0.000	0.067	0.091	0.287	0.067	0.084	0.084
*method*8	0.091	0.000	0.067	0.000	0.067	0.091	0.311	0.067	0.087	0.091
*method*9	0.851	0.183	0.915	0.194	0.523	0.955	0.492	0.882	0.624	0.300
*method*10	**0.857**	0.184	0.916	0.213	0.499	0.947	0.436	0.874	0.616	0.300
*method*11	0.607	0.161	0.910	0.212	0.393	0.913	0.413	**0.883**	0.561	0.292
*method*12	0.355	0.129	0.789	0.001	0.297	0.556	0.253	0.864	0.406	0.287
*method*13	0.587	0.074	0.873	0.065	0.410	0.865	0.052	0.689	0.452	0.331
*method*14	0.427	0.052	0.908	0.062	0.213	0.789	0.030	0.674	0.394	0.334
*method*15	0.395	0.039	0.684	0.063	0.087	0.606	0.017	0.605	0.312	0.272
*method*16	0.177	0.029	0.450	0.059	0.018	0.425	0.025	0.751	0.242	0.254
*method*17	0.830	0.138	0.908	0.220	0.523	0.937	0.430	0.822	0.601	0.296
*method*18	0.855	0.158	0.908	0.235	0.464	0.932	0.408	0.783	0.593	0.293
*method*19	0.829	0.167	0.832	0.213	0.392	0.940	0.414	0.667	0.557	0.280
*method*20	0.585	**0.194**	0.640	0.214	0.367	0.922	0.349	0.201	0.434	0.244
*method*21	0.808	0.160	0.912	0.194	0.549	**0.963**	0.457	0.813	0.607	0.296
*method*22	0.828	0.158	0.895	0.189	0.509	0.945	0.422	0.788	0.592	0.295
*method*23	0.802	0.171	0.891	0.209	0.411	0.949	0.392	0.669	0.562	0.286
*method*24	0.620	0.031	0.596	0.200	0.369	0.948	0.388	0.254	0.426	0.269

A.1.2 Beta Recall

Table A.3: Beta Recall Results of each method on each dataset.

experiment	heart	hepatitis	stroke	parkinsons	fetal health	breast-cancer	diabetes	cardio	Average	Std
baseline	**0.025**	**0.077**	**0.407**	0.041	**0.085**	**0.788**	**0.245**	0.187	**0.232**	0.242
*method*1	0.023	0.033	0.388	0.035	0.025	0.747	0.244	0.116	0.201	0.240
*method*2	0.003	0.014	0.016	0.007	0.001	0.145	0.130	0.002	0.040	0.057
*method*3	0.007	0.011	0.011	0.005	0.001	0.213	0.169	0.001	0.052	0.081
*method*4	0.005	0.015	0.004	0.004	0.001	0.180	0.165	0.000	0.047	0.073
*method*5	0.003	0.014	0.000	0.003	0.001	0.064	0.030	0.000	0.014	0.021
*method*6	0.000	0.015	0.000	0.003	0.001	0.013	0.013	0.000	0.006	0.006
*method*7	0.000	0.017	0.000	0.005	0.001	0.010	0.016	0.000	0.006	0.007
*method*8	0.001	0.015	0.000	0.005	0.001	0.012	0.009	0.000	0.005	0.005
*method*9	0.022	0.059	0.390	0.038	0.068	0.751	0.236	0.143	0.214	0.234
*method*10	0.022	0.056	0.342	0.046	0.047	0.732	0.217	0.084	0.193	0.228
*method*11	0.016	0.034	0.121	0.040	0.032	0.628	0.177	0.027	0.134	0.194
*method*12	0.004	0.020	0.043	0.010	0.007	0.180	0.053	0.006	0.041	0.055
*method*13	0.008	0.005	0.379	0.000	0.001	0.338	0.015	0.257	0.125	0.158
*method*14	0.006	0.000	0.274	0.000	0.000	0.215	0.010	0.281	0.098	0.124
*method*15	0.004	0.003	0.157	0.000	0.000	0.160	0.006	0.357	0.086	0.122
*method*16	0.002	0.004	0.063	0.000	0.000	0.115	0.004	**0.448**	0.080	0.145
*method*17	0.024	0.060	0.377	0.042	0.059	0.762	0.211	0.109	0.206	0.238
*method*18	0.022	0.066	0.240	0.045	0.042	0.728	0.207	0.052	0.175	0.222
*method*19	0.022	0.045	0.112	0.036	0.017	0.768	0.171	0.022	0.149	0.239
*method*20	0.011	0.024	0.011	0.039	0.010	0.719	0.067	0.001	0.110	0.231
*method*21	0.024	0.068	0.387	0.043	0.076	0.749	0.219	0.099	0.208	0.233
*method*22	0.023	0.065	0.244	**0.051**	0.056	0.738	0.221	0.056	0.182	0.225
*method*23	0.022	0.038	0.164	0.044	0.026	0.744	0.194	0.027	0.157	0.230
*method*24	0.016	0.022	0.017	0.038	0.010	0.705	0.094	0.004	0.113	0.225

A.1.3 Detection

Table A.4: Detection Results of each method on each dataset.

experiment	heart	hepatitis	stroke	parkinsons	fetal health	breast-cancer	diabetes	cardio	Average	Std
baseline	0.837	**0.803**	0.865	**0.785**	**0.222**	0.987	**0.832**	0.890	**0.778**	0.218
*method*1	0.826	0.784	0.822	0.658	0.108	0.872	0.781	0.874	0.716	0.238
*method*2	0.271	0.233	0.439	0.277	0.017	0.532	0.711	0.451	0.366	0.199
*method*3	0.482	0.284	0.364	0.282	0.031	0.593	0.812	0.185	0.379	0.230
*method*4	0.405	0.269	0.173	0.300	0.001	0.547	0.820	0.081	0.325	0.248
*method*5	0.259	0.208	0.094	0.326	0.000	0.161	0.377	0.091	0.190	0.120
*method*6	0.002	0.413	0.098	0.354	0.080	0.045	0.135	0.082	0.151	0.140
*method*7	0.006	0.386	0.096	0.359	0.017	0.058	0.196	0.070	0.148	0.140
*method*8	0.001	0.380	0.092	0.338	0.044	0.048	0.113	0.083	0.138	0.132
*method*9	**0.921**	0.757	0.852	0.631	0.167	**0.996**	0.802	**0.902**	0.753	0.245
*method*10	0.909	0.733	0.848	0.639	0.129	0.961	0.801	0.848	0.733	0.247
*method*11	0.638	0.741	0.681	0.612	0.096	0.861	0.794	0.690	0.639	0.219
*method*12	0.328	0.732	0.423	0.223	0.042	0.516	0.703	0.581	0.444	0.223
*method*13	0.450	0.164	0.748	0.037	0.211	0.503	0.330	0.770	0.402	0.250
*method*14	0.391	0.087	0.678	0.010	0.292	0.380	0.235	0.736	0.351	0.240
*method*15	0.248	0.049	0.567	0.010	0.172	0.227	0.092	0.672	0.255	0.226
*method*16	0.055	0.005	0.396	0.007	0.005	0.134	0.083	0.640	0.165	0.217
*method*17	0.846	0.776	**0.886**	0.709	0.162	0.966	0.805	0.874	**0.753**	0.235
*method*18	0.884	0.773	0.866	0.752	0.128	0.975	0.810	0.847	0.755	0.246
*method*19	0.795	0.763	0.752	0.664	0.087	0.940	0.761	0.770	0.692	0.239
*method*20	0.582	0.656	0.423	0.619	0.056	0.896	0.603	0.306	0.518	0.237
*method*21	0.896	0.798	0.877	0.688	0.196	0.951	0.816	0.859	0.760	0.225
*method*22	0.872	0.788	0.861	0.662	0.125	0.980	0.794	0.852	0.742	0.248
*method*23	0.846	0.738	0.752	0.680	0.083	0.940	0.758	0.763	0.695	0.243
*method*24	0.646	0.619	0.472	0.646	0.047	0.920	0.536	0.370	0.532	0.237

A.1.4 Authenticity

Table A.5: Authenticity Results of each method on each dataset.

experiment	heart	hepatitis	stroke	parkinsons	fetal health	breast-cancer	diabetes	cardio	Average	Std
baseline	0.980	0.878	0.611	0.944	0.937	0.174	0.737	0.844	0.763	0.250
*method*1	0.983	0.946	0.645	0.954	0.979	0.217	0.753	0.918	0.799	0.248
*method*2	0.998	0.994	0.985	0.996	**1.000**	0.840	0.877	0.998	0.961	0.060
*method*3	0.996	0.994	0.988	0.997	0.999	0.756	0.827	**1.000**	0.945	0.090
*method*4	0.998	0.994	0.995	0.997	0.999	0.755	0.830	**1.000**	0.946	0.091
*method*5	0.998	0.994	**1.000**	0.997	0.999	0.928	0.971	**1.000**	0.986	0.024
*method*6	0.999	0.994	**1.000**	0.999	**1.000**	**0.989**	0.993	**1.000**	**0.997**	0.004
*method*7	**1.000**	0.994	**1.000**	0.999	**1.000**	**0.989**	0.991	**1.000**	**0.997**	0.004
*method*8	0.999	0.994	**1.000**	0.999	**1.000**	0.988	0.994	**1.000**	**0.997**	0.004
*method*9	0.982	0.903	0.614	0.941	0.949	0.190	0.750	0.887	0.777	0.249
*method*10	0.982	0.907	0.686	0.944	0.962	0.197	0.772	0.936	0.798	0.247
*method*11	0.990	0.955	0.910	0.946	0.975	0.259	0.826	0.977	0.855	0.230
*method*12	0.997	0.975	0.967	0.992	0.993	0.765	0.946	0.994	0.954	0.073
*method*13	0.994	0.995	0.667	**1.000**	**1.000**	0.640	0.980	0.796	0.884	0.148
*method*14	0.996	0.999	0.761	**1.000**	**1.000**	0.727	0.989	0.774	0.906	0.118
*method*15	0.998	0.997	0.856	**1.000**	**1.000**	0.802	0.994	0.722	0.921	0.105
*method*16	**1.000**	**1.000**	0.947	**1.000**	**1.000**	0.868	**0.996**	0.667	0.935	0.110
*method*17	0.981	0.909	0.643	0.933	0.955	0.191	0.768	0.917	0.787	0.249
*method*18	0.982	0.911	0.797	0.941	0.967	0.191	0.780	0.957	0.816	0.247
*method*19	0.986	0.925	0.899	0.945	0.984	0.213	0.813	0.981	0.843	0.244
*method*20	0.992	0.960	0.987	0.945	0.989	0.232	0.921	0.999	0.878	0.246
*method*21	0.982	0.894	0.644	0.942	0.939	0.210	0.763	0.925	0.787	0.242
*method*22	0.981	0.902	0.800	0.941	0.955	0.220	0.759	0.955	0.814	0.236
*method*23	0.984	0.947	0.862	0.938	0.978	0.199	0.802	0.977	0.836	0.248
*method*24	0.990	0.974	0.982	0.942	0.988	0.237	0.903	0.997	0.877	0.244

A.1.5 Train Adversarial Accuracy

Table A.6: Train Adversarial Accuracy Results of each method on each dataset.

experiment	heart	hepatitis	stroke	parkinsons	fetal health	breast-cancer	diabetes	cardio	Average	Std
baseline	**0.919**	0.938	**0.660**	0.969	**0.965**	0.175	**0.856**	0.884	**0.796**	0.252
*method*1	0.951	0.972	0.695	0.975	0.989	0.235	0.871	0.943	0.829	0.242
*method*2	0.999	0.997	0.992	0.998	1.000	0.750	0.938	0.999	0.959	0.081
*method*3	0.998	0.997	0.994	0.999	1.000	0.761	0.911	1.000	0.958	0.079
*method*4	0.999	0.997	0.998	0.999	1.000	0.740	0.913	1.000	0.956	0.086
*method*5	0.999	0.997	1.000	0.999	1.000	0.840	0.985	1.000	0.978	0.052
*method*6	1.000	0.997	1.000	0.999	1.000	0.995	0.996	1.000	0.998	0.002
*method*7	1.000	0.997	1.000	0.999	1.000	0.995	0.995	1.000	0.998	0.002
*method*8	1.000	0.997	1.000	0.999	1.000	0.994	0.997	1.000	0.998	0.002
*method*9	0.929	0.951	0.669	0.968	0.973	0.178	0.868	0.920	0.807	0.255
*method*10	0.953	0.952	0.742	0.969	0.980	0.179	0.881	0.958	0.827	0.256
*method*11	0.994	0.978	0.949	0.973	0.987	0.328	0.911	0.986	0.888	0.213
*method*12	0.999	0.987	0.983	0.996	0.997	0.755	0.973	0.997	0.961	0.078
*method*13	0.934	**0.965**	0.662	0.995	0.964	**0.657**	0.965	0.824	0.871	0.131
*method*14	0.954	0.981	0.746	0.996	0.972	0.729	0.979	0.799	0.894	0.108
*method*15	0.969	0.985	0.834	1.000	0.980	0.795	0.985	0.742	0.911	0.097
*method*16	0.990	0.996	0.931	0.999	0.996	0.878	0.991	**0.641**	0.928	0.116
*method*17	0.924	0.952	0.696	**0.964**	0.977	0.181	0.875	0.942	0.814	0.254
*method*18	0.936	0.954	0.852	0.969	0.982	0.195	0.885	0.973	0.843	0.249
*method*19	0.973	0.962	0.941	0.971	0.992	0.213	0.903	0.988	0.868	0.249
*method*20	0.994	0.979	0.993	0.971	0.995	0.243	0.960	0.999	0.892	0.245
*method*21	0.920	0.944	0.698	0.967	0.966	0.218	0.875	0.949	0.817	0.241
*method*22	0.920	0.948	0.851	0.968	0.976	0.209	0.872	0.971	0.839	0.242
*method*23	0.949	0.974	0.915	0.968	0.989	0.199	0.896	0.986	0.859	0.252
*method*24	0.992	0.987	0.990	0.969	0.994	0.243	0.950	0.998	0.891	0.245

A.1.6 Test Adversarial Accuracy

Table A.7: Test Adversarial Accuracy Results of each method on each dataset.

experiment	heart	hepatitis	stroke	parkinsons	fetal health	breast-cancer	diabetes	cardio	Average	Std
baseline	0.693	**0.883**	**0.633**	0.828	**0.897**	0.477	0.810	0.865	**0.761**	0.138
*method*1	0.754	0.924	0.670	0.854	0.962	0.545	0.806	0.930	0.805	0.135
*method*2	0.971	0.989	0.979	0.974	0.999	0.805	0.882	0.998	0.950	0.065
*method*3	0.949	0.988	0.982	0.974	0.999	0.785	0.839	0.999	0.940	0.076
*method*4	0.957	0.989	0.993	0.974	0.999	0.805	0.842	1.000	0.945	0.072
*method*5	0.968	0.988	0.999	0.974	0.999	0.921	0.945	1.000	0.974	0.027
*method*6	0.997	0.990	0.999	0.977	0.999	0.991	0.982	1.000	0.992	0.008
*method*7	0.996	0.991	0.999	0.977	0.999	0.989	0.973	1.000	0.991	0.010
*method*8	0.996	0.990	0.999	0.977	0.999	0.989	0.985	1.000	0.992	0.008
*method*9	0.713	0.896	0.651	0.841	0.916	0.525	0.800	0.901	0.781	0.131
*method*10	0.751	0.896	0.713	0.836	0.935	0.451	0.817	0.945	0.793	0.150
*method*11	0.870	0.934	0.916	0.849	0.948	0.542	0.838	0.979	0.859	0.129
*method*12	0.953	0.954	0.960	0.956	0.984	0.823	0.920	0.994	0.943	0.050
*method*13	0.833	0.938	0.655	0.982	0.945	0.629	0.947	**0.816**	0.843	0.128
*method*14	0.878	0.952	0.723	0.987	0.945	0.724	0.965	0.790	0.871	0.103
*method*15	0.927	0.975	0.808	0.997	0.965	0.737	0.984	0.732	0.891	0.106
*method*16	0.969	0.995	0.923	1.000	0.990	0.838	0.982	0.626	0.915	0.121
*method*17	0.709	0.899	0.681	0.831	0.920	0.489	0.811	0.929	0.784	0.141
*method*18	0.720	0.908	0.824	0.851	0.935	0.482	0.822	0.964	0.813	0.144
*method*19	0.799	0.915	0.905	0.841	0.969	**0.500**	0.845	0.982	0.844	0.143
*method*20	0.896	0.942	0.983	0.856	0.978	0.538	0.908	0.998	0.887	0.140
*method*21	**0.697**	0.893	0.672	**0.826**	0.906	0.505	**0.800**	0.936	0.779	0.137
*method*22	0.715	0.912	0.827	0.833	0.923	0.488	0.805	0.962	0.808	0.141
*method*23	0.757	0.934	0.877	0.831	0.958	0.473	0.829	0.979	0.830	0.152
*method*24	0.872	0.960	0.978	0.836	0.977	0.558	0.896	0.997	0.884	0.135

A.1.7 Privacy Loss

Table A.8: Privacy Loss Results of each method on each dataset.

experiment	heart	hepatitis	stroke	parkinsons	fetal health	breast-cancer	diabetes	cardio	Average	Std
baseline	-0.226	-0.055	-0.027	-0.141	-0.068	0.302	-0.046	-0.018	-0.035	0.143
*method*1	-0.197	-0.049	-0.025	-0.121	-0.027	0.311	-0.065	-0.014	-0.023	0.139
*method*2	-0.028	-0.008	-0.013	-0.024	-0.001	0.055	-0.056	-0.001	-0.010	0.030
*method*3	-0.049	-0.009	-0.012	-0.024	-0.001	0.024	-0.072	-0.000	-0.018	0.028
*method*4	-0.042	-0.008	-0.004	-0.024	-0.001	0.065	-0.072	-0.000	-0.011	0.037
*method*5	-0.031	-0.009	-0.001	-0.024	-0.001	0.080	-0.040	-0.000	-0.003	0.035
*method*6	-0.003	-0.007	-0.001	-0.022	-0.001	-0.004	-0.015	-0.000	-0.007	0.007
*method*7	-0.004	-0.007	-0.001	-0.022	-0.001	-0.005	-0.022	-0.000	-0.008	0.009
*method*8	-0.004	-0.007	-0.001	-0.022	-0.001	-0.005	-0.012	-0.000	-0.007	0.007
*method*9	-0.215	-0.054	-0.017	-0.127	-0.057	0.347	-0.067	-0.019	-0.026	0.154
*method*10	-0.202	-0.057	-0.029	-0.133	-0.046	0.273	-0.064	-0.013	-0.034	0.129
*method*11	-0.124	-0.044	-0.033	-0.124	-0.038	0.214	-0.073	-0.007	-0.029	0.100
*method*12	-0.046	-0.033	-0.023	-0.040	-0.012	0.068	-0.053	-0.003	-0.018	0.036
*method*13	-0.101	-0.027	-0.008	-0.013	-0.019	-0.029	-0.019	-0.008	-0.028	0.029
*method*14	-0.076	-0.029	-0.022	-0.009	-0.027	-0.005	-0.014	-0.009	-0.024	0.021
*method*15	-0.042	-0.010	-0.026	-0.003	-0.016	-0.057	-0.001	-0.010	-0.021	0.019
*method*16	-0.020	-0.001	-0.008	0.001	-0.006	-0.040	-0.008	-0.015	-0.012	0.012
*method*17	-0.215	-0.053	-0.015	-0.133	-0.056	0.308	-0.064	-0.013	-0.030	0.142
*method*18	-0.216	-0.045	-0.028	-0.117	-0.047	0.287	-0.062	-0.009	-0.030	0.135
*method*19	-0.174	-0.047	-0.037	-0.129	-0.023	0.287	-0.058	-0.007	-0.023	0.129
*method*20	-0.098	-0.037	-0.010	-0.114	-0.017	0.294	-0.052	-0.001	-0.004	0.119
*method*21	-0.223	-0.051	-0.026	-0.142	-0.060	0.287	-0.075	-0.013	-0.038	0.139
*method*22	-0.205	-0.036	-0.024	-0.135	-0.053	0.279	-0.067	-0.010	-0.031	0.132
*method*23	-0.192	-0.040	-0.038	-0.137	-0.031	0.275	-0.067	-0.007	-0.030	0.129
*method*24	-0.120	-0.027	-0.013	-0.133	-0.017	0.315	-0.054	-0.002	-0.006	0.130

A.1.8 Balanced Accuracy Gap

Table A.9: Balanced Accuracy Gap Results of each method on each dataset.

experiment	heart	hepatitis	stroke	parkinsons	fetal health	breast-cancer	diabetes	cardio	Average	Std
baseline	**0.146**	0.059	-0.017	0.202	0.197	0.047	0.029	0.102	0.096	0.075
*method*1	0.165	0.113	0.003	**0.196**	0.168	0.038	0.000	0.097	0.097	**0.072**
*method*2	0.334	0.364	0.022	0.403	0.573	0.010	-0.006	0.170	0.234	0.202
*method*3	0.296	0.364	0.008	0.403	0.573	0.089	-0.004	0.208	0.242	0.191
*method*4	0.315	0.364	0.022	0.403	0.573	0.094	0.008	0.233	0.252	0.187
*method*5	0.322	0.364	0.022	0.403	0.573	**0.005**	0.166	0.233	0.261	0.182
*method*6	0.500	0.364	0.022	0.403	0.573	0.136	0.196	0.233	0.303	0.176
*method*7	0.500	0.364	0.022	0.403	0.573	0.136	0.196	0.233	0.303	0.176
*method*8	0.491	0.364	0.022	0.403	0.573	0.136	0.196	0.233	0.302	0.175
*method*9	0.151	0.069	-0.010	0.292	0.197	0.066	0.010	0.114	0.111	0.094
*method*10	0.164	0.084	-0.014	0.261	0.210	0.021	-0.001	0.086	0.101	0.095
*method*11	0.258	0.161	0.005	0.228	0.228	-0.007	**-0.036**	0.105	0.118	0.111
*method*12	0.305	0.281	0.022	0.364	0.257	0.023	0.055	0.118	0.178	0.130
*method*13	0.347	0.368	0.023	0.403	0.573	0.159	0.152	0.039	0.258	0.182
*method*14	0.444	0.371	0.014	0.363	0.548	0.140	0.140	0.034	0.257	0.187
*method*15	0.440	0.407	0.024	0.393	0.564	0.121	0.190	**0.027**	0.271	0.193
*method*16	0.477	0.351	0.026	0.346	0.542	0.195	0.167	0.056	0.270	0.177
*method*17	0.164	0.012	-0.012	0.248	0.207	0.033	-0.019	0.124	0.095	0.098
*method*18	0.178	**-0.013**	**-0.091**	0.244	0.202	0.012	0.012	0.072	**0.077**	0.111
*method*19	0.185	0.138	-0.060	0.265	0.227	0.030	-0.031	0.122	0.109	0.112
*method*20	0.237	0.124	0.018	0.232	0.208	0.043	0.004	0.104	0.121	0.090
*method*21	0.163	0.121	-0.029	0.263	0.176	0.027	-0.020	0.080	0.098	0.096
*method*22	0.166	0.111	-0.055	0.292	**0.163**	0.072	0.018	0.091	0.107	0.098
*method*23	0.154	0.190	-0.037	0.282	0.199	0.038	-0.013	0.119	0.117	0.105
*method*24	0.241	0.334	0.010	0.256	0.216	0.037	-0.018	0.126	0.150	0.122

A.1.9 Alignment

Table A.10: Alignment Results of each method on each dataset.

experiment	heart	hepatitis	stroke	parkinsons	fetal health	breast-cancer	diabetes	cardio	Average	Std
baseline	**0.855**	0.781	0.937	0.718	0.767	0.816	0.734	0.705	0.789	0.073
*method*1	0.835	0.838	0.950	0.697	0.789	0.729	0.740	0.700	0.785	0.081
*method*2	0.668	0.908	**0.989**	0.774	0.788	0.725	0.728	0.592	0.772	0.119
*method*3	0.703	0.908	0.984	0.774	0.788	0.751	0.755	0.520	0.773	0.129
*method*4	0.684	0.908	**0.989**	0.774	0.788	0.751	0.772	0.492	0.770	0.138
*method*5	0.679	0.908	**0.989**	0.774	0.788	0.758	0.665	0.509	0.759	0.139
*method*6	0.513	0.908	**0.989**	0.774	0.788	0.773	0.677	0.493	0.740	0.163
*method*7	0.513	0.908	**0.989**	0.774	0.788	0.773	0.677	0.460	0.735	0.169
*method*8	0.518	0.908	**0.989**	0.774	0.788	0.773	0.677	0.494	0.740	0.162
*method*9	0.850	0.715	0.962	0.641	0.783	0.751	0.734	0.681	0.765	0.095
*method*10	0.836	0.699	0.948	0.667	0.745	0.776	0.738	0.713	0.765	0.084
*method*11	0.741	0.854	0.963	0.636	0.769	0.805	0.746	0.685	0.775	0.095
*method*12	0.693	0.671	**0.989**	0.764	**0.791**	0.794	0.660	0.671	0.754	0.103
*method*13	0.657	0.891	0.988	0.774	0.785	0.620	0.650	0.833	0.775	0.120
*method*14	0.555	0.874	0.986	0.600	0.796	0.620	0.647	0.837	0.739	0.145
*method*15	0.562	0.714	0.985	0.467	0.753	0.589	0.543	0.850	0.683	0.164
*method*16	0.521	0.256	0.939	0.682	0.666	0.387	0.508	0.797	0.594	0.207
*method*17	0.837	0.817	0.955	0.656	0.739	0.779	0.754	0.665	0.775	0.091
*method*18	0.821	0.795	0.914	0.692	0.741	0.809	0.745	0.743	0.783	0.064
*method*19	0.817	0.799	0.930	0.662	0.700	0.791	0.754	0.664	0.765	0.084
*method*20	0.763	0.739	0.988	0.713	0.732	0.715	0.724	0.687	0.758	0.090
*method*21	0.838	0.740	0.934	0.687	**0.818**	0.790	0.779	0.732	**0.790**	0.071
*method*22	0.834	0.774	0.929	0.641	0.775	0.769	0.759	0.712	0.774	0.079
*method*23	0.846	0.842	0.949	0.667	0.767	0.769	**0.780**	0.672	0.786	0.088
*method*24	0.759	**0.866**	0.971	0.595	0.751	0.808	0.756	0.658	0.771	0.109

A.1.10 Train and Generation Times

Table A.11: Train Time of each method on each dataset (in seconds).

experiment	heart	hepatitis	stroke	parkinsons	fetal health	breast-cancer	diabetes	cardio	Average	Std
baseline	220.393	226.259	286.740	301.948	226.765	221.084	242.056	**504.863**	278.763	90.322
*method*1	328.721	349.341	437.230	439.636	371.474	337.953	378.359	716.235	419.869	118.629
*method*2	293.500	355.230	419.142	470.005	317.771	356.366	320.606	683.090	401.964	119.212
*method*3	279.178	259.373	364.192	324.936	274.040	284.880	293.178	567.343	330.890	94.614
*method*4	279.965	258.732	361.562	325.217	275.082	285.184	292.407	566.703	330.606	94.250
*method*5	279.106	258.321	362.340	324.932	274.405	285.280	292.744	567.820	330.619	94.759
*method*6	279.884	258.973	361.688	325.437	274.699	285.363	291.963	568.202	330.776	94.746
*method*7	279.726	259.016	361.061	326.094	274.346	285.100	292.586	564.996	330.366	93.729
*method*8	280.322	259.412	362.111	325.407	274.957	285.172	292.922	567.295	330.950	94.352
*method*9	221.514	226.432	290.752	304.395	233.108	220.105	243.803	527.603	283.464	97.023
*method*10	219.455	222.945	285.201	299.923	229.584	216.681	235.827	520.045	278.708	95.822
*method*11	223.048	212.640	277.624	288.885	214.371	226.914	221.083	514.617	272.398	95.554
*method*12	225.851	**208.216**	**277.367**	**284.381**	**209.657**	233.139	**219.762**	508.880	**270.907**	93.947
*method*13	363.776	328.084	558.728	335.969	423.292	306.923	382.107	1980.611	584.936	532.685
*method*14	362.518	327.365	557.365	334.290	421.043	304.586	380.572	1982.734	583.809	533.906
*method*15	361.754	327.242	557.505	332.347	421.079	304.020	378.360	1983.088	583.174	534.323
*method*16	361.060	327.686	558.260	333.178	421.429	303.801	379.251	1987.401	584.008	535.650
*method*17	**212.001**	224.534	282.868	301.185	224.067	219.108	233.731	547.425	280.615	105.266
*method*18	215.658	222.711	285.271	300.114	224.096	**214.691**	234.183	553.144	281.234	107.203
*method*19	222.524	224.383	302.341	300.266	228.433	220.283	237.065	562.783	287.260	108.926
*method*20	231.922	227.848	316.366	300.418	231.532	231.571	253.914	564.953	294.815	106.998
*method*21	246.700	259.699	364.951	370.340	245.328	260.804	284.136	879.961	363.990	200.626
*method*22	248.057	262.525	367.206	369.155	250.102	260.263	282.787	881.234	365.166	200.508
*method*23	258.950	268.574	379.453	368.171	261.542	263.683	286.705	887.654	371.842	200.205
*method*24	272.838	284.739	395.031	368.895	272.415	281.603	311.485	894.857	385.233	197.395

Table A.12: Generation Time of each method on each dataset (in seconds).

experiment	heart	hepatitis	stroke	parkinsons	fetal health	breast-cancer	diabetes	cardio	Average	Std
baseline	0.003	0.000	0.007	0.000	0.003	0.000	0.007	0.030	0.006	0.009
*method*1	0.001	0.009	0.015	0.000	0.008	0.000	0.002	0.034	0.009	0.011
*method*2	0.003	0.002	0.003	0.004	0.010	0.003	0.003	0.064	0.012	0.020
*method*3	0.000	0.003	0.000	0.000	0.003	0.001	0.001	0.025	0.004	0.008
*method*4	0.003	0.000	0.000	0.000	0.000	0.000	0.003	0.023	0.004	0.007
*method*5	0.000	0.001	0.002	0.002	0.004	0.000	0.000	0.030	0.005	0.010
*method*6	0.000	0.000	0.002	0.000	0.000	0.000	0.000	0.028	0.004	0.009
*method*7	0.000	0.000	0.003	0.000	0.001	0.003	0.000	0.025	0.004	0.008
*method*8	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.029	0.004	0.009
*method*9	0.000	0.000	0.003	0.000	0.003	0.000	0.003	0.023	0.004	0.007
*method*10	0.000	0.000	0.001	0.003	0.003	0.000	0.000	0.027	0.004	0.009
*method*11	0.003	0.000	0.000	0.000	0.003	0.000	0.000	0.025	0.004	0.008
*method*12	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.029	0.004	0.009
*method*13	0.000	0.000	0.000	0.000	0.003	0.000	0.000	0.022	0.003	0.007
*method*14	0.006	0.000	0.000	0.000	0.006	0.000	0.003	0.027	0.005	0.009
*method*15	0.000	0.000	0.001	0.000	0.004	0.000	0.000	0.027	0.004	0.009
*method*16	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.026	0.003	0.008
*method*17	0.003	0.000	0.000	0.000	0.000	0.000	0.000	0.025	0.004	0.008
*method*18	0.000	0.000	0.004	0.000	0.000	0.000	0.003	0.027	0.004	0.009
*method*19	0.000	0.000	0.000	0.000	0.003	0.003	0.000	0.022	0.004	0.007
*method*20	0.003	0.000	0.003	0.000	0.000	0.000	0.003	0.024	0.004	0.008
*method*21	0.000	0.000	0.003	0.000	0.000	0.000	0.000	0.031	0.004	0.010
*method*22	0.000	0.003	0.003	0.000	0.000	0.000	0.000	0.023	0.004	0.007
*method*23	0.000	0.000	0.006	0.000	0.000	0.000	0.000	0.024	0.004	0.008
*method*24	0.000	0.000	0.007	0.000	0.000	0.000	0.003	0.024	0.004	0.008

A.2 Dataset Feature Relevance Results

A.2.1 Alpha Precision

Table A.13: Dataset Meta-feature Relevance for Performance in Alpha Precision.

experiment_type	feature	joint_relevance
regularization	freq_class.mean	-0.769
regularization	nr_norm	-0.744
regularization	nr_bin	0.634
regularization	sparsity.mean	-0.617
regularization	cov.mean	0.427
dropout	nr_cor_attr	-0.757
dropout	nr_bin	0.720
dropout	nr_attr	-0.532
dropout	sparsity.mean	0.488
dropout	sparsity.sd	0.471
differential_privacy	nr_bin	0.860
differential_privacy	sparsity.sd	0.691
differential_privacy	freq_class.mean	0.435
differential_privacy	inst_to_attr	0.432
differential_privacy	nr_attr	-0.410
corruption	nr_bin	0.745
corruption	nr_cor_attr	-0.699
corruption	sparsity.mean	0.613
corruption	nr_norm	-0.524
corruption	nr_attr	-0.460
corruption_contrastive	nr_cor_attr	-0.701
corruption_contrastive	nr_bin	0.697
corruption_contrastive	nr_norm	-0.478
corruption_contrastive	nr_attr	-0.463
corruption_contrastive	sparsity.sd	0.338

A.2.2 Authenticity

Table A.14: Dataset Meta-feature Relevance for Performance in Authenticity.

experiment_type	feature	joint_relevance
regularization	nr_bin	-0.424
regularization	freq_class.mean	0.388
regularization	nr_norm	0.310
regularization	nr_attr	0.228
regularization	sparsity.mean	0.161
dropout	sparsity.mean	-0.425
dropout	nr_cor_attr	0.413
dropout	nr_attr	0.348
dropout	nr_bin	-0.323
dropout	nr_outliers	0.303
differential_privacy	nr_bin	-0.749
differential_privacy	sparsity.sd	-0.579
differential_privacy	eigenvalues.mean	-0.470
differential_privacy	nr_attr	0.458
differential_privacy	inst_to_attr	-0.455
corruption	sparsity.mean	-0.483
corruption	nr_cor_attr	0.426
corruption	nr_attr	0.362
corruption	nr_bin	-0.315
corruption	cov.mean	0.231
corruption_contrastive	nr_cor_attr	0.426
corruption_contrastive	nr_attr	0.363
corruption_contrastive	nr_bin	-0.282
corruption_contrastive	cov.mean	0.263
corruption_contrastive	nr_norm	0.203

A.2.3 Balanced Accuracy Gap

Table A.15: Dataset Meta-feature Relevance for Performance in Balanced Accuracy Gap.

experiment_type	feature	joint_relevance
regularization	nr_norm	0.706
regularization	nr_attr	0.696
regularization	freq_class.mean	0.485
regularization	sparsity.mean	0.353
regularization	sparsity.sd	0.299
dropout	nr_attr	0.675
dropout	nr_cor_attr	0.511
dropout	nr_outliers	0.460
dropout	nr_norm	0.328
dropout	attr_to_inst	0.302
differential_privacy	nr_attr	0.659
differential_privacy	nr_bin	-0.591
differential_privacy	freq_class.mean	-0.498
differential_privacy	inst_to_attr	-0.485
differential_privacy	gravity	-0.334
corruption	nr_attr	0.666
corruption	nr_cor_attr	0.416
corruption	nr_norm	0.326
corruption	nr_bin	-0.308
corruption	attr_to_inst	0.270
corruption_contrastive	nr_attr	0.586
corruption_contrastive	nr_cor_attr	0.460
corruption_contrastive	nr_norm	0.376
corruption_contrastive	nr_bin	-0.336
corruption_contrastive	attr_to_inst	0.263