Flow Matching for Tabular Data Synthesis

Anonymous authors
Paper under double-blind review

Abstract

Synthetic data generation is an important tool for privacy-preserving data sharing. While diffusion models have set recent benchmarks, flow matching (FM) offers a promising alternative. This paper presents different ways to implement flow matching for tabular data synthesis. We provide a comprehensive empirical study that compares flow matching (FM and variational FM) with a state-of-the-art diffusion method (TabDDPM and TabSyn) in tabular data synthesis. We evaluate both the standard Optimal Transport (OT) and the Variance Preserving (VP) probability paths, and also compare deterministic and stochastic samplers – something possible when learning to generate using *variational* flow matching – characterising the empirical relationship between data utility and privacy risk. Our key findings reveal that flow matching, particularly TabbyFlow, outperforms diffusion baselines. Flow matching methods also achieves better performance with remarkably low function evaluations ($\leq$ 100 steps), offering a substantial computational advantage. The choice of probability path is also crucial, as using the OT path demonstrates superior performance, while VP has potential for producing synthetic data with lower disclosure risk. Lastly, our results show that making flows stochastic not only preserves marginal distributions but, in some instances, enables the generation of high utility synthetic data with reduced disclosure risk.

1 Introduction

Government bodies, particularly national statistical offices (NSOs), are confronted with a pressing challenge: how to disseminate useful tabular data while preserving individual privacy. Tabular data, which are structured data with rows and columns representing real-world entities and their attributes, are essential for decision-making in economics, healthcare, social sciences, and finance (Little et al. 2024). However, because these datasets often contain sensitive attributes, such as financial conditions or medical diagnoses, privacy protection laws such as the EU's General Data Protection Regulation (GDPR) and Indonesia's Personal Data Protection (PDP) Act limit access to them (Ruggles & Van Riper 2021). This tension between data utility and confidentiality has spurred interest in methods that can generate synthetic tabular data that preserve statistical properties while reducing the risk of disclosure (Nowok et al. 2016).

Recent advances in deep generative models have provided a promising approach to tabular data synthesis. Deep learning techniques, such as Generative Adversarial Networks (Goodfellow et al. 2014), Variational Autoencoders (Kingma & Welling 2013), and more recently, diffusion models (Sohl-Dickstein et al. 2015; Ho et al. 2020), have demonstrated impressive capabilities in generating synthetic data of various forms, including images and text, as well as tabular data (Xu et al. 2019; Kotelnikov et al. 2023).

Diffusion models, in particular, have emerged as a powerful option for generative tasks, including the synthesis of tabular data. TabDDPM (Kotelnikov et al. 2023), a diffusion model tailored for tabular data, and other models such as TabSyn (Zhang et al. 2024) have gained attention for their ability to model intricate dependencies between variables in tabular datasets by iteratively transforming noise into structured data. The main strength of these models lies in their ability to handle the high-dimensional structure of tabular data, where the relationships between variables are often nonlinear and complex.

Another recent development is flow matching (FM) (Lipman et al. 2023; Albergo et al. 2023; Liu 2022), which approximates the data distribution through a flow, i.e. through learning a velocity field that induces an ordinary differential equation transporting noise to data. Variational Flow Matching (VFM) (Eijkelboom et al. 2024) offers a reinterpretation of flow matching as a form of variational inference over trajectories, providing an elegant solution to parameterise the flow towards any kind of distribution (e.g. also discrete or constrained). This method holds significant promise for tabular data synthesis, as it ensures that the generated output not only mirrors the statistical attributes of the original data but also captures patterns that traditional generative techniques might overlook. Beyond tabular data generation (Guzmán-Cordero et al. 2025), VFM has also seen recent success in fields like molecular generation (Zaghen et al. 2025b; Eijkelboom et al. 2025; Sakalyan et al. 2025), image generation (Matişan et al. 2025), and climate modeling (Finn et al. 2025). Further discussion on related work can be found in Appendix A.

Although recent advances in tabular diffusion models (Kotelnikov et al. 2023; Zhang et al. 2024) and FM (Guzmán-Cordero et al. 2025) have shown promise, existing studies remain limited in four ways. First, they have focused primarily on open source benchmark datasets, neglecting the unique challenges of census data as a real-world representation. Second, utility and privacy risk evaluation has been restricted to computer science metrics such as ML accuracy and distance to closest record (DCR), which (i) may not align with statistical needs,(ii) are computationally intensive, and (iii) can be hard to interpret for practitioners. Third, recent flow matching studies (Eijkelboom et al. 2024; Guzmán-Cordero et al. 2025; Eijkelboom et al. 2025) have predominantly focused on optimal transport trajectories. The potential of variance-preserving trajectories - although theoretically established in the original FM formulation (Lipman et al. 2023) and proven effective in diffusion models (Ho et al. 2020) — remains underexplored. Finally, the interaction between latent vs. direct representations and deterministic vs. stochastic dynamics in FM-based tabular synthesis remains unclear, leaving open questions about which configuration produces the best synthetic data.

Research Objective. This paper's primary goal is quantify whether flow matching can offer an efficient, high-utility, and low-risk alternative to diffusion models for mixed-type tabular data. We address the goal through four specific research questions.

- **Q1.** Can FM-based models improve upon diffusion model baselines in tabular data synthesis? (Section 5.1)
- **Q2.** How efficient are FM models in terms of the utility and risk of the generated output? (Section 5.2)
- **Q3.** How do utility and risk characteristics of the synthetic data evolve with integration time? (Section 5.3)
- **Q4.** Can stochastic sampling improve synthesis quality relative to deterministic sampling? (Section 5.4)

Contributions. We explore flow matching techniques for tabular data synthesis through four implementations: (1) learning the distribution of latent variables using regular flow matching, inspired by TabSyn (Zhang et al. 2024); (2) learning the tabular data distribution directly using variational flow matching; (3) a systematic comparison of interpolation schemes; and (4) implementing stochastic dynamics in VFM.

At the application level, our work bridges flow matching with the operational needs of statistical agencies by providing a systematic analysis of the utility and risk of the synthetic data under different experimental conditions, recognising that algorithms optimising fidelity may implicitly amplify disclosure risk. The results demonstrate that flow matching can generate high-quality synthetic tabular data while protecting against statistical disclosure, thus offering a new option for responsible data generation.

2 Background

2.1 Flow Matching

Flow matching (FM) (Lipman et al. 2023; Liu et al. 2022; Albergo et al. 2023) is a simulation-free generative modelling framework that learns a velocity field v_t^θ parameterised by $\theta \in \mathbb{R}^p$. This field defines a continuous transformation of a base distribution p_0 (typically a standard Gaussian) into a target distribution p_1 (empirical data) over a pseudo-time interval $t \in [0, 1]$ through its induced ordinary differential equation.

Rather than directly estimating the target velocity field $u_t(x)$ – which we do not have access to – FM defines conditional distributions $p_t(x_t \mid x_1)$ that make an assumption on the dynamics *towards a fixed* endpoint x_1, for which the *conditional* velocity is simple to compute. Under this formulation, the marginal velocity field is given by:

$$u_t(x_t) = \int u_t(x_t \mid x_1) \frac{p_t(x_t \mid x_1) p_{\text{data}}(x_1)}{p_t(x_t)} \, \mathrm{d}x_1, \tag{2.1}$$

where $u_t(x_t \mid x_1)$ is the conditional velocity field. To avoid the computational cost of evaluating the integral in equation 2.1, FM learns the target velocity field through a Monte Carlo estimate of the conditional velocity, making the problem tractable. This approach is known as Conditional Flow Matching (CFM):

$$\mathcal{L}_{\text{CFM}}(\theta) = \mathbb{E}_{t \sim [0,1], x_1 \sim p_{\text{data}}, x_t \sim p_t(x_t \mid x_1)} \left[\left\| v_t^\theta(x_t) - u_t(x_t \mid x_1) \right\|_2^2 \right]. \tag{2.2}$$

A common way to define this conditional distribution is through interpolation, i.e. define x_t as a linear combination of x_1 and a noise sample x_0:

$$x_t = \alpha_t x_1 + \sigma_t x_0 \sim p_t(x_t \mid x_1) \quad \text{for} \quad x_0 \sim p_0, \; x_1 \sim p_1, \; t \in [0,1]. \tag{2.3}$$

where α_t, σ_t are time-dependent coefficients that specify the trajectory. We can hence write down the corresponding conditional velocity:

$$u(x_t \mid x_1) = \dot{\alpha}_t x_1 + \dot{\sigma}_t \frac{x_t - \alpha_t x_1}{\sigma_t} = \underbrace{\left(\dot{\alpha}_t - \frac{\dot{\sigma}_t}{\sigma_t} \alpha_t \right)}_{A_t(x_t)} x_1 + \underbrace{\frac{\dot{\sigma}_t}{\sigma_t}}_{B_t(x_t)} x_t. \tag{2.4}$$

Here, $A_t(x_t)$ and $B_t(x_t)$ are time-dependent coefficients that control the contributions of x_1 and x_t, respectively. This expression allows for efficient computation of the conditional velocity field without needing to perform numerical integration or simulation.

2.2 Variational Flow Matching

Although flow matching is already an efficient and simulation free generative modelling framework, it can struggle with multimodal data or data of heterogeneous types (Zhai & Hao 2025). To address this limitation, Eijkelboom et al. (2024) put forward variational flow matching (VFM), which reinterprets flow matching using the perspective of variational inference.

Unlike standard FM - which is primarily designed for continuous data - VFM allows a more flexible learning mechanism that accommodates a wider range of data types, including tabular data (Guzmán-Cordero et al. 2025). VFM learns an approximate vector field via the posterior distribution of the data using a variational distribution q_t^θ:

$$v_t^\theta(x_t) := \int u_t(x_t \mid x_1) q_t^\theta(x_1 \mid x_t) \, \mathrm{d}x_1. \tag{2.5}$$

This formulation allows the model to flexibly approximate the true conditional velocity while allowing the incorporation of data-specific constraints (e.g. discrete data) through the choice of posterior. The training objective in VFM is to minimise the divergence between the true joint distribution and the variational joint distribution, measured in the Kullback-Leibler divergence (KLD):

$$\begin{aligned}
\mathcal{L}_{\text{MF-VFM}}(\theta) &= \mathbb{E}_t \left[\mathrm{KL} \left(p_t(x_t) p_t(x_1 \mid x_t) \,\middle\|\, p_t(x_t) q_t^\theta(x_1 \mid x_t) \right) \right] && (2.6) \\
&= -\mathbb{E}_{t, x_1 x_t} \left[\log q_t^\theta(x_1 \mid x_t) \right] + \text{const} && (2.7) \\
&= -\mathbb{E}_{t, x_1, x_t} \left[\sum_{d=1}^{D} \log q_t^\theta(x_1^d \mid x_t) \right] + \text{const}, && (2.8)
\end{aligned}$$

where the final line assumes a mean-field approximation (which does not restrict expressivity as posited in the theoretical analysis of Eijkelboom et al. (2024)) factorising the variational posterior across dimensions.

Having learned the posterior distribution, the vector field can be computed using equation 2.9:

$$v_t^\theta(x_t) = \mathbb{E}_{q_t^\theta(x_1|x_t)}\left[u_t(x_t|x_1)\right] = \left(\dot{\alpha}_t - \frac{\dot{\sigma}_t}{\sigma_t}\alpha_t\right)\theta_t(x_t) + \frac{\dot{\sigma}_t}{\sigma_t}x_t \tag{2.9}$$

where $\theta_t(x_t)$ is the mean of the variational distribution q_t^θ. Crucially, this formulation recovers standard flow matching in the Gaussian case.

Stochastic Dynamics. Beyond the deterministic ODE formulation, VFM also admits a stochastic counterpart that preserves the same marginal distribution $p_t(x)$ Eijkelboom et al. (2024):

$$\mathrm{d}x_t \;=\; \Big[u_t(x_t) + \tfrac{g_t^2}{2}\,\nabla_{x_t}\log p_t(x_t)\Big]\mathrm{d}t \;+\; g_t\,\mathrm{d}w_t, \tag{2.10}$$

where $g_t \geq 0$ is a diffusion scheduler and w_t is a standard Wiener process. The scheduler controls the stochasticity of the trajectory, but not the marginal path; Setting $g_t \equiv 0$ recovers the ODE. We approximate this SDE using the variational distribution $q_t^\theta(x_1 \mid x)$ as follows:

$$\mathrm{d}x_t = \Big[v_t^\theta(x_t) + \tfrac{g_t^2}{2}\,s_t^\theta(x_t)\Big]\mathrm{d}t + g_t\,\mathrm{d}w_t, \tag{2.11}$$

where the drift and score are estimated as:

$$v_t^\theta(x_t) = \mathbb{E}_{q_t^\theta(x_1|x_t)}[u_t(x_t \mid x_1)], \qquad s_t^\theta(x_t) = \mathbb{E}_{q_t^\theta(x_1|x_t)}[\nabla_{x_t}\log p_t(x_t \mid x_1)]. \tag{2.12}$$

This formulation supports approximate stochastic flow models, enabling richer models while still leveraging the variational structure for tractable learning.

2.3 Trajectory Setting in Flow Matching

There are two commonly used trajectories to define the probability path in flow matching (Lipman et al. 2023).

The first is the **optimal transport (OT)** trajectory (Liu et al. 2022; Tong et al. 2024), where the mean and standard deviation of the conditional distribution evolve linearly over time. This produces a deterministic vector field with a fixed direction, resulting in straight-line (conditional) paths connecting the noise and data distributions (Liu et al. 2022). OT-based trajectories are computationally straightforward, efficient, and easy to parameterise due to their time-invariant structure, as shown in equation 2.13

$$\begin{aligned} x_t &= tx_1 + (1-(1-\sigma_{\min})t)x_0 \\ u_t(x_t \mid x_1) &= \frac{x_1 - (1-\sigma_{\min})x_t}{1-(1-\sigma_{\min})t}, \end{aligned} \tag{2.13}$$

where $\sigma_{\min}$ is added to ensure the target density has non-zero measure everywhere (or: we can imagine first convolving the data distribution with a small Gaussian).

The second type of trajectory is the **variance-preserving (VP)** path, derived from stochastic differential equations commonly used in diffusion models (Lipman et al. 2023). In this approach, noise is gradually added to the data during the forward process, while the reverse process learns to de-noise using time-dependent vector fields. Unlike OT paths, VP diffusion trajectories evolve smoothly in both magnitude and direction over time. The parameters α_t and σ_t are determined by cumulative functions of the noise scale function $\beta(t) = \beta_{\min} + t(\beta_{\max} - \beta_{\min})$, as given in equation 2.14.

$$\begin{aligned} \alpha_t = e^{-\frac{1}{2}T(t)}, \quad T(t) = \int_0^{1-t}\beta(s)ds, \quad \sigma_t = \sqrt{1-\alpha_t^2} \\ u_t(x_t|x_1) = -\frac{\dot{T}(t)}{2}\left[\frac{e^{-T(t)}x_t - e^{-\frac{1}{2}T(t)}x_1}{1-e^{-T(t)}}\right] \end{aligned} \tag{2.14}$$

The choice of trajectory – or interpolation scheme – significantly affects data synthesis performance, which we explore in this paper. Depending on the data structure or modelling objectives, either trajectory can be more suitable. Additional trajectory formulations are provided in Appendix E.1.

2.4 TabSyn: Latent Diffusion for Tabular Data

TabSyn (Zhang et al. 2024) represents a state-of-the-art generative approach for mixed-type tabular data. It integrates a transformer-based Variational Autoencoder (VAE) with a score-based diffusion model operating in latent space. The design of TabSyn addresses several key challenges in tabular data synthesis including heterogeneous feature types, complex inter-feature dependencies, and the inefficiency of diffusion processes when employed on high-dimensional structured data.

The TabSyn framework comprises two stages. First, a **VAE** maps the raw tabular data into a continuous latent space. Second, a **diffusion model** is trained to capture the underlying distribution with this latent space. To handle different data types, TabSyn uses column-wise tokenisers and detokenisers, transformer-based encoders/decoders, and an adaptive β-VAE objective. The encoder learns structured latent representations, while the decoder ensures accurate and type-consistent data reconstruction.

Within the latent space, noise is introduced through a forward process defined as: $z_t = z_0 + \sigma(t)\varepsilon$, with $\sigma(t) = t$ under a linear noise schedule. The corresponding reverse process is an SDE:

$$\mathrm{d}z_t = -2\sigma_t\dot{\sigma}_t\nabla_z \log p_t(z_t)\,\mathrm{d}t + \sqrt{2\sigma_t\dot{\sigma}(t)}\,\mathrm{d}\omega_t. \tag{2.15}$$

where ω_t denotes a Wiener process. The model is trained using denoising score matching, with a neural network $\varepsilon_\theta(z_t, t)$ that estimates the scoring function $\nabla_{z_t} \log p_t(z_t) = -\varepsilon_\theta(z_t, t)/\sigma(t)$.

A study by Zhang et al. (2024) showed that TabSyn surpasses traditional GAN and VAE based models as well as the latest diffusion models like TabDDPM and STaSy, across a variety of datasets and evaluation metrics - including unvariate distribution and pairwise correlation preservation, and downstream ML performance. Furthermore, its latent-space formulation allows for missing value imputation via latent inpainting, achieving competitive results in terms of data quality, diversity, and privacy preservation.

3 Designing Flow Matching for Tabular Data Synthesis

3.1 Motivation and Design Axes

Flow Matching provides a general recipe for learning deterministic or stochastic transformations between probability distributions. Yet, its effectiveness in practice depends not on a single equation, but on a set of design decisions that determine how distributions are represented, interpolated, and evolved over time. When applied to *tabular data synthesis*, these decisions interact in particularly complex ways: mixed continuous-categorical features challenge smooth trajectories, privacy-sensitive settings impose computational limits, and statistical agencies often require explainable and reproducible models. This section therefore reformulates Flow Matching as a *design space* rather than a single algorithmic recipe, and places tabular synthesis at its centre.

Our aim is to identify how four core axes – **representation**, **learning target**, **trajectory**, and **dynamics** – jointly shape the behaviour of tabular flow models.

1. **Representation** decides whether learning occurs in a continuous *latent space* that simplifies mixed data, or directly in the *data space*, preserving semantics.
2. **Learning target** distinguishes *Conditional Flow Matching* (velocity regression) from *Variational Flow Matching* (posterior regression), reflecting different ways of estimating the transport field.
3. **Trajectory** defines the interpolant between source and data distributions – commonly *Optimal Transport (OT)* or *Variance Preserving (VP)* paths – which control how signal and noise evolve over pseudo-time.
4. **Dynamics** determine whether the learned flow is integrated deterministically as an *ODE* or stochastically as an *SDE*, offering a continuum between precise transport and exploratory regularisation.

By systematically combining these axes, we focus on two complementary formulations. First, we introduce **TabSynFlow**, which learns deterministic ODE flows in latent space, substituting the diffusion process in

TabSyn with a learned velocity field. We also evaluate **TabbyFlow** developed by Guzmán-Cordero et al. (2025) which extends Variational Flow Matching directly to data space, supporting both numerical and categorical variables through hybrid Gaussian-categorical posteriors.

We consider a dataset of mixed-type tabular observations $x = (x_{\text{num}}, x_{\text{cat}})$, where $x_{\text{num}} \in \mathbb{R}^{D_{\text{num}}}$ are continuous features and $x_{\text{cat}} \in \{1, \dots, K_d\}^{D_{\text{cat}}}$ are categorical variables. The empirical data distribution is denoted by $p_1(x)$, while a simple prior $p_0(x)$, typically an isotropic Gaussian, serves as the source distribution. The goal of flow matching is to learn a continuous transformation that maps p_0 to p_1 through a parameterised time-dependent velocity field $v_t^\theta(x_t)$ defined on the interpolation in equation 2.3. The coefficients (α_t, σ_t) specify the probability path (for example, Optimal Transport or Variance Preserving trajectories) and induce a family of intermediate marginals $p_t(x_t)$. The generative process is recovered by integrating either a deterministic ordinary differential equation (ODE) or a stochastic differential equation (SDE) via equation 2.10. The two instantiations below, TabbyFlow and TabSynFlow, correspond to different modelling choices for the representation space and the learning target.

3.2 TabbyFlow: Variational Flow Matching in Data Spaces

TabbyFlow (Guzmán-Cordero et al. 2025) extends the Variational Flow Matching (VFM) framework of Eijkelboom et al. (2024) to structured tabular data containing both numerical and categorical features. This formulation enables direct modelling in the data space, avoiding the need for an intermediate latent representation, while explicitly accounting for heterogeneous variable types within the flow dynamics.

The TabbyFlow objective treats each feature type according to its statistical nature: continuous variables are modelled using a Gaussian distribution, while categorical variables are modelled using categorical distributions. Let $x = (x_{\text{num}}, x_{\text{cat}})$ denote a sample comprising D_{num} numerical and D_{cat} categorical features. Each categorical feature is one-hot encoded, yielding total dimensionality

$$D = D_{\text{num}} + \sum_{d=1}^{D_{\text{cat}}} K_d, \tag{3.1}$$

where K_d is the number of categories for the d-th categorical variable.

Numerical Variables. Following Theorem 3 of Eijkelboom et al. (2024), we assume a mean-field variational posterior $q_t^\theta(x_{\text{num},1} \mid x_t)$ with Gaussian structure parameterised by mean $\theta_t(x_t)$ and variance $0.5A_t(x_t)^{-2}$. The VFM loss for numerical variables is

$$\mathcal{L}_{\text{VFM-num}}(\theta) = \mathbb{E}_{t,x_t,x_1}\left[\|A_t(x_t)(x_{\text{num},1} - \theta_t(x_t))\|_2^2\right]. \tag{3.2}$$

However, we found that using $0.5A_t(x_t)^{-2}$ as variance yielded lower performance in our experiment. To mitigate this, we implemented a relaxed variance of $0.5A_t(x_t)^{-1}$. This adjustment slightly inflates the variance, therefore acting as a mild regulariser. The empirical results can be seen in appendix D.1.

Categorical Variables. For categorical features, TabbyFlow models the variational posterior as a product of categorical distributions. Let $\theta_t^{dk}(x_t)$ denote the predicted probability of category k in the d-th feature, obtained via softmax in the network output layer. The loss minimises cross-entropy with the one-hot targets $x_{\text{cat},1}^d$:

$$\mathcal{L}_{\text{VFM-cat}}(\theta) = -\mathbb{E}_{t,x_t,x_1}\left[\sum_{d=1}^{D_{\text{cat}}}\sum_{k=1}^{K_d} \mathbb{I}[x_{\text{cat},1}^d = k] \log \theta_t^{dk}(x_t)\right]. \tag{3.3}$$

Unified Objective. Combining both loss terms yields the overall TabbyFlow objective:

$$\begin{aligned}\mathcal{L}_{\text{TabbyFlow}}(\theta) = \mathbb{E}_{t,x_t,x_1}&\left[\|\sqrt{A_t(x_t)}(x_{\text{num},1} - \theta_t(x_t))\|_2^2\right] \\ &- \mathbb{E}_{t,x_t,x_1}\left[\sum_{d=1}^{D_{\text{cat}}}\sum_{k=1}^{K_d} \mathbb{I}[x_{\text{cat},1}^d = k] \log \theta_t^{dk}(x_t)\right].\end{aligned} \tag{3.4}$$

Sampling and Dynamics. During sampling, a noise sample $\tilde{x}_0$ is drawn from p_0, and the trained network predicts $\theta_t(x_t)$ at each time step. Because TabbyFlow does not directly output a velocity field, it is recovered via

$$v_t^\theta(x_t) = \left(\dot{\alpha}_t - \frac{\dot{\sigma}_t}{\sigma_t}\alpha_t\right)\theta_t(x_t) + \frac{\dot{\sigma}_t}{\sigma_t}x_t, \tag{3.5}$$

and, for the stochastic variant, the score function

$$s_t^\theta(x_t) = -\frac{x_t - \alpha_t\theta_t(x_t)}{\sigma_t^2}. \tag{3.6}$$

These components define the ODE or SDE dynamics used for synthesis (Algorithms 3–4 in Appendix B.2). TabbyFlow thus enables a unified treatment of numerical and categorical features within a single variational flow, generalising continuous-time generative modelling to structured tabular domains.

3.3 TabSynFlow: Flow Matching in Latent Spaces

TabSynFlow extends latent-space generative modelling by replacing the diffusion component in TabSyn with a deterministic probability flow trained via Conditional Flow Matching (CFM). The model learns a time-dependent velocity field that transports a simple prior to the target latent distribution through an ODE, enabling a direct and continuous transformation during both training and sampling. This deterministic integration stabilises training and accelerates synthesis relative to iterative stochastic denoising, while retaining high-fidelity latent reconstructions through the encoder–decoder pathway.

TabSynFlow follows a two-stage framework. In the first stage, a VAE maps the raw tabular data, comprising both continuous and categorical features, into a continuous latent space. Subsequently, flow matching is applied to model the latent distribution $p(z_1)$, replacing the score-based diffusion model used in TabSyn. Gaussian noise samples $z_0 \sim \mathcal{N}(0, I)$ are continuously transformed into latent variables z_1 through an ODE parameterised by $v_t^\theta(z_t)$. The conditional velocity $u_t(z_t \mid z_1)$ is computed using Eq. equation 2.4, and the loss becomes

$$\mathcal{L}_{\text{TabSynFlow}}(\theta) = \mathbb{E}_{t,z_1,z_t}\left[\|v_t^\theta(z_t) - u_t(z_t \mid z_1)\|_2^2\right]. \tag{3.7}$$

After training, sampling proceeds by drawing $z_0 \sim \mathcal{N}(0, I)$ and integrating the learned ODE until $t = 1$. The resulting latent representation z_1 is then decoded to obtain a synthetic record $\tilde{x}$, followed by inverse transformation to the original feature space. Algorithms 1–2 in Appendix B.1 describe the procedure in detail.

Operating in latent space grants TabSynFlow efficient and stable training while retaining the expressive power of flow-based models. It supports fast, deterministic sampling and interpretable trajectories, making it a computationally efficient alternative to diffusion-based approaches for tabular data synthesis.

4 Experimental Setting

Table 1 summarises the datasets used in our empirical evaluation. Following Ran et al. (2024) and Little et al. (2024), we used four census datasets (from the UK, Fiji, Canada and Rwanda), plus one additional census data from Indonesia. We also added two further databases from UCI which are standard benchmarks for tabular data Kotelnikov et al. (2023). Unlike Ran et al. (2024), we treat age as continuous to exploit its natural ordering. Also, this choice remains compatible with TabDDPM/TabSyn, which operate on mixed-type data. To ensure a consistent and fair comparison, we adopt the same convention for TabSynFlow and TabbyFlow.

All the models in this study use a multilayer perceptron (MLP) architecture comprising four hidden layers with sizes [1024, 2048, 2048, 1024], configuration used by Zhang et al. (2024). TabSyn and TabSynFlow employ Transformer-based architectures in their VAE counterparts, diverging from the standard MLP-based tabular VAE design (Xu et al. 2019).

Training was conducted with a batch size to 4096, and the ODE was integrated until $t = 1$ using 100 steps [1]. Models were trained for up to 10,000 epochs, with early stopping based on the lowest observed training loss, which is similar to previous studies (Zhang et al. 2024; Guzmán-Cordero et al. 2025), although those studies capped training at 8000 epochs. The integration for synthesis uses Euler method. For sampling from the SDE, we set $g_t = \sigma_t$.

We focused the synthetic tabular data evaluation on preserving the statistical properties of the original data (utility) while minimising the potential to disclose sensitive information about individuals (disclosure risk) (Taub et al. 2019; Elliot et al. 2023; Little et al. 2022). **Utility** values range from 0 (no preservation of statistical properties) to 1 (perfect preservation), measuring how well synthetic data supports the same analyses as the original. **Disclosure risk** ranges from 0 (no added risk beyond baseline) to 1 (maximum disclosure risk), quantifying the probability that an adversary can correctly match sensitive information using synthetic data. Detailed evaluation methods are provided in Appendix C.

Table 1: Datasets used for the empirical study.

Abbr.	Dataset Name	#Obs.	#Num. Variables	#Cat. Variables
UK	UK Census	104267	1	14
CA	Canada Census	32149	4	21
FI	Fiji Census	84323	1	18
RW	Rwanda Census	31455	1	12
ID	Indonesia Census	177429	1	12
AD	Adult	48842	5	10
CH	Churn Modelling	10000	4	7

5 Results

In this section, we present and analyse the empirical findings of our study, highlighting the comparative performance of diffusion and flow-matching approaches for tabular data synthesis. As seen from the objective function, all algorithms optimise for utility; disclosure risk is measured post-hoc. We begin by establishing TabDDPM and TabSyn as strong baselines (Section 5.1). We then examine the number of function evaluations (NFEs), evaluating the utility and risk dynamics between computational efficiency (Section 5.2). Next, we investigate the impact of integration time (t_{ode}) on model performance, with specific attention to the evolution of probability paths (Section 5.3). Finally, we evaluate the effect of incorporating stochastic sampling within the VFM framework (Section 5.4). Collectively, these analyses provide a picture of how modelling choices shape the quality, efficiency, and reliability of synthetic tabular data generation.

5.1 Main Comparison

In this section, we present results for the census datasets which have been used for benchmarking in previous studies (e.g. Ran et al. 2024; Little et al. 2024). Table 2 summarises the comparative performance of six synthetic data generation algorithms across UK, CA, FI, and RW datasets.

TabDDPM exhibits highly inconsistent results, excelling in UK while substantially underperforming in CA, FI, and RW. On the other hand, **TabSyn** demonstrates robust utility across all datasets, achieving values above 0.6 in RW and exceeding 0.7 in UK, CA, and FI, making it a reliable choice for utility-driven synthesis.

However, **TabbyFlow** proves to be the most effective model for these benchmarks. The OT trajectory delivers the highest utility in UK and CA, while the VP trajectory attains the best scores in FI and RW. Notably, TabbyFlow-VP consistently emerges as the second-best in utility, and further distinguishes itself by achieving lower risk, particularly in FI and RW.

[1] Unless otherwise specified, this is the default setup

From regular flow matching side, **TabSynFlow-OT** outperforms TabSyn in UK and RW, attaining lower disclosure risk in their high utility synthetic data, indicative of improved synthesis quality. In CA and FI, TabSynFlow demonstrates a classic risk–utility trade off. Moreover, TabSynFlow-VP achieves the most favorable risk profiles in CA, FI, and RW at the cost of lower utility.

In summary, **TabbyFlow**—with both OT and VP trajectories—consistently attains the highest overall average scores — calculated via average of utility and 1-risk — on the main census benchmark, achieving values of 0.7038 (OT) and 0.7031 (VP). TabSyn and TabSynFlow-OT also perform competitively, with average scores of 0.6742 and 0.6716, respectively, ranking just below the TabbyFlow models. In comparison, TabSynFlow-VP yields a lower average score of 0.6150, while TabDDPM lags behind at 0.5083. These findings underscore the robustness of flow matching algorithms, particularly the TabbyFlow family, in effectively high data utility and less disclosure risk. The findings also highlight the critical role of trajectory choice (OT or VP) that modulates the relationship in some datasets, particularly the risk as post-hoc consequence. Results for the additional datasets (Indonesia, Adult, and Churn) are available in Appendix D, and further analyses of alternative interpolants for both TabSynFlow and TabbyFlow are provided in Appendix E.2 and Appendix E.3, respectively.

Table 2: Comparative Performance of six synthetic data generation algorithms across four common census datasets (UK, CA, FI, RW). Bold and underline indicate the first and second best performing algorithms for each dataset, respectively. The table emphasises best performance of TabbyFlow in common census benchmarks.

Evaluation	Algorithm	UK	CA	FI	RW
Utility	TabDDPM	0.7823±0.0207	0.2819±0.0043	0.1266±0.0076	0.4125±0.0126
	TabSyn	0.7617±0.0204	0.7366±0.0174	0.7085±0.0246	0.666±0.0203
	TabSynFlow-OT	0.7796±0.0188	0.7035±0.0129	0.6798±0.0132	0.6765±0.0221
	TabSynFlow-VP	0.6696±0.0136	0.6403±0.0088	0.6014±0.0157	0.5875±0.0175
	TabbyFlow-OT	**0.8333±0.0191**	**0.7718±0.0211**	<u>0.7263±0.0205</u>	<u>0.7284±0.0231</u>
	TabbyFlow-VP	<u>0.8066±0.0197</u>	<u>0.7472±0.0188</u>	**0.7451±0.0216**	**0.7358±0.0185**
Risk	TabDDPM	0.4993±0.0071	**0.0061±0.0214**	**0.0525±0.0739**	**0.1182±0.1838**
	TabSyn	0.4855±0.0068	0.2670±0.0122	0.5614±0.0076	0.5000±0.0104
	TabSynFlow-OT	0.4834±0.0085	0.2493±0.0149	0.5506±0.0065	0.4894±0.0107
	TabSynFlow-VP	**0.4630±0.0061**	<u>0.2073±0.0156</u>	<u>0.5208±0.0077</u>	<u>0.4655±0.0199</u>
	TabbyFlow-OT	0.4889±0.0043	0.3206±0.0146	0.5705±0.0063	0.5379±0.0139
	TabbyFlow-VP	<u>0.4765±0.0072</u>	0.3008±0.0148	0.5588±0.0074	0.5180±0.0105

5.2 Number of Function Evaluations between Tabsyn and Flow Matching Models

The analysis of computational efficiency presented in Figure 1 illustrates the relationship between the number of function evaluations (NFEs) and the performance of synthetic data generation models, highlighting the relationship between computational cost and data quality.

From the figure, it can be seen that the FM models generally converge after approximately 100 NFEs, indicating that using low steps in FM is sufficient. On the other hand, TabSyn requires more than 100 steps to outperform FM models in terms of utility, indicating a higher computational burden, in addition to costly training of VAE+diffusion.

In low-NFE settings, where flow-based models are designed to operate efficiently, TabSyn performed poorly. TabbyFlow-VP whilst yielding lower utility also achieves lower disclosure risk, and still outperforms TabSyn when NFEs are below 64. This early-stage advantage positions flow matching models, particularly using the OT path as promising candidates for applications operating under strict computational constraints.

In summary, while TabSyn can achieve competitive utility, it does so at the cost of significantly higher NFEs. Flow-matching models offer a more efficient alternative, especially in scenarios where computational efficiency is critical.

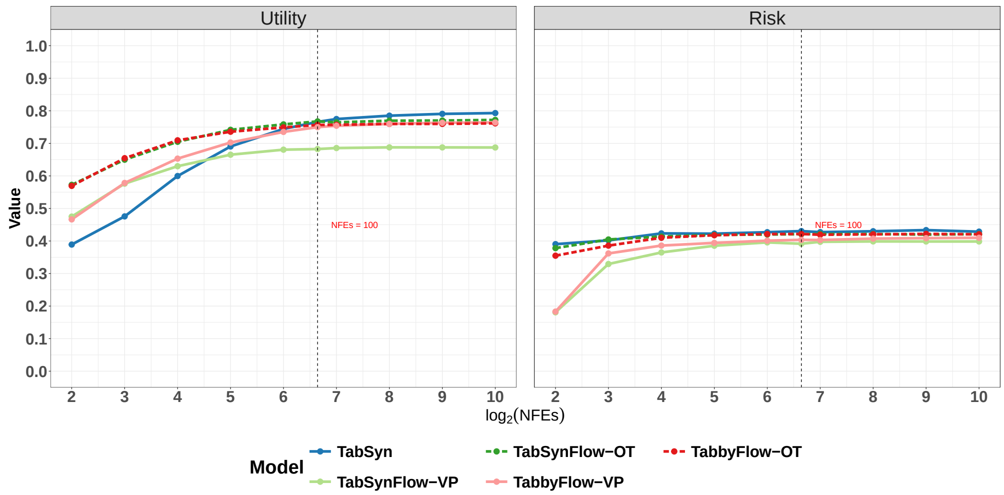

Figure 1: **Average utility (↑) and disclosure risk (↓) as a function of the number of function evaluations (NFEs) for TabSyn and flow matching models (TabSynFlow-OT, TabSynFlow-VP, TabbyFlow-OT, TabbyFlow-VP) averaged across four datasets** Solid lines represent utility, while dotted lines represent risk. Flow-matching models converge after approximately 100 NFEs, achieving competitive utility at substantially lower computational cost compared to TabSyn.

5.3 Flow Matching Results based on Integration Time

In standard flow matching implementations, the ODE is fully integrated; however, early termination is possible in practice. Figure 2 provides an analysis of the effect of ODE integration times (t_{ode}) on utility and risk, averaged across all datasets. For TabSynFlow-OT, the utility remains consistently high throughout integration times, demonstrating robust performance even with early stopping. As in $t_{\text{ode}} \to 1.0$, the performance across all methods tends to plateau.

In contrast, both TabSynFlow and TabbyFlow using the VP path exhibit a lower initial utility at $t_{\text{ode}} = 0.6$, suggesting that VP introduces disruptive noise early in the process. Utility and risk increase substantially when $t_{\text{ode}} >= 0.6$, indicating that VP require full integration to achieve competitive utility. From these results, we can see that the consistent superiority of OT paths across integration times highlighting their robustness where early termination is desirable for computationally constrained applications. Dataset-specific results provided in Appendix D.4 show deviations around the average, particularly for $t_{\text{ode}} = [0.9, 1]$, emphasising the importance of considering dataset characteristics in selecting integration parameters.

5.4 SDE Implementation of Variational Flow Matching

In this subsection, we investigate whether stochastic differential equation (SDE) sampling yields results comparable to ODE sampling. Tables 3 and 4 present a comparative analysis of ODE and SDE formulations, with additional results provided in Appendix E.4 for alternative diffusion coefficients.

In general, it can be seen that the SDE approach performs as expected theoretically, producing marginal distributions similar to those generated by ODE Eijkelboom et al. (2024) while, in some cases, producing superior performance. This is evidenced by the close alignment of utility and risk values across both formulations. This empirical validation confirms the mathematical foundation presented in the paper.

Beyond theoretical equivalence, SDE sampling offers a practical advantage by improving the ODE results in some cases. For example, in the Fiji and Rwanda datasets across both experimental setups, SDE-based VFM simultaneously produced high utility synthetic data with lower risk - a dual improvement that is rare in practice, where gains typically only occur in one metric.

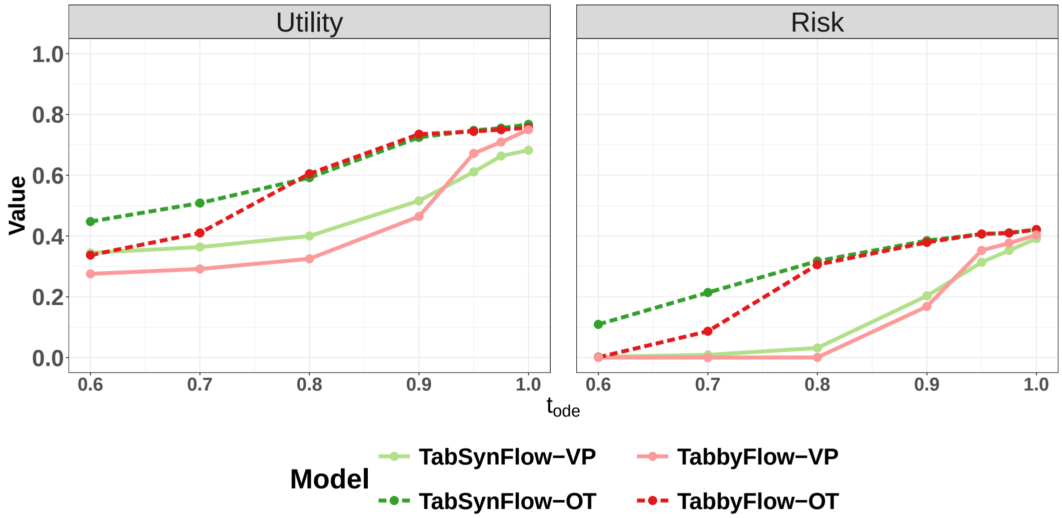

Figure 2: **Average utility (↑) and risk (↓) against ODE integration time** (t_{ode}) **for TabSynFlow and TabbyFlow** using OT and VP paths. Colours and line styles indicate method and path (VP light and solid, OT dark and dashed). OT paths enable early stopping without significant utility loss, whereas VP paths require full integration to achieve competitive performance.

Table 3: Comparison of average utility and risk between ODE and SDE performance for TabbyFlow-OT.

Dataset	ODE		SDE	
	Utility	Risk	Utility	Risk
UK	0.8333	0.4889	0.8306	0.4916
Indonesia	0.6671	0.4114	0.6629	0.4081
Canada	0.7718	0.3207	0.7785	0.3152
Fiji	0.7263	0.5705	0.7394	0.5664
Rwanda	0.7284	0.5379	0.7353	0.5361
Adult	0.7720	0.5443	0.7731	0.5508
Churn	0.7966	0.0773	0.7986	0.0784

Table 4: Comparison of average utility and risk between ODE and SDE performance for TabbyFlow-VP.

Dataset	ODE		SDE	
	Utility	Risk	Utility	Risk
UK	0.8066	0.4765	0.7903	0.4543
Indonesia	0.6488	0.3918	0.6386	0.5005
Canada	0.7472	0.3008	0.7414	0.2926
Fiji	0.7451	0.5588	0.7859	0.5343
Rwanda	0.7358	0.5180	0.7430	0.5032
Adult	0.7581	0.5677	0.7514	0.5039
Churn	0.8067	0.0753	0.7941	0.1079

6 Discussion and Concluding Remarks

6.1 Discussion

The experimental results demonstrate the performance characteristics of flow-based deep generative models for tabular data synthesis, highlighting how utility and disclosure risk vary across algorithms and datasets.

Performance Comparison (Section 5.1). Flow matching models consistently outperform TabDDPM on most datasets, with TabbyFlow generating synthetic data of the highest utility in nearly all cases. This increased fidelity is generally accompanied by elevated disclosure risk, reflecting a positive empirical correlation between utility and risk. Our most interesting finding is while it is not commonly used in FM studies, using VP path in TabbyFlow has potential to produce better optimised synthetic data with higher utility but lower disclosure risk. TabSyn remains a reliable benchmark, while TabSynFlow-OT shows targeted improvements over TabSyn in selected cases but not universally. These results highlight flow matching—particularly TabbyFlow—as a viable alternative for tabular data synthesis, with trajectory choice gives different performance in terms of data utility and disclosure risk.

Efficiency via NFEs (Section 5.2). Flow matching models, particularly TabbyFlow-OT and TabSynFlow-OT, demonstrate strong utility with a relatively small number of function evaluations (NFEs), typically between 64 and 128, whereas TabSyn requires substantially more NFEs to reach comparable performance. Notably, TabbyFlow can achieve near-optimal utility by NFEs 128, and even at very low budgets (four evaluation steps), meaningful results are observed. The actual impact of NFEs, however, depends heavily on data size, feature complexity, and model architecture. For instance, generating small datasets like CH is fast, but larger datasets such as UK or ID require more time and batch-wise processing to avoid out-of-memory errors, particularly on personal computers. Overall, lower NFEs values generally reflect greater computational efficiency, making flow-based models especially attractive for resource-constrained environments.

Integration Time and Early Stopping (Section 5.3). OT paths outperform VP paths in both TabSynFlow and TabbyFlow, maintaining high utility earlier in the generation process. VP paths require full integration to achieve competitive performance. However, both trajectories are prone to reduced utility and / or increased risk when the integration time approaches one ($t_{ode} \to 1$). Practical strategies to mitigate this include early stopping ($t_{ode} \leq 0.9$) or velocity clipping[2]. In general, OT trajectories provide a more stable and reliable approach to tabular synthesis.

Deterministic vs. stochastic sampling (Section 5.4). The application of SDE solvers within the VFM framework presents a more complex picture than initially hypothesised. While it is theoretically established that SDE recovers the same marginal distributions as ODE, our empirical results reveal numerical differences. In particular, on datasets such as Fiji and Rawnda, SDE sampling led to a simultaneous increase in utility and reduction in risk. This dual improvement suggests that the controlled stochasticity of the SDE can act as a beneficial regulariser or enable more effective exploration of the data space, rather than merely introducing destructive noise. However, since performance gains were not observed across all datasets, the utility of SDEs appears to be data-dependent. Thus, the choice between deterministic and stochastic samplers is not about universal dominance, but about suitability to specific dataset.

6.2 Concluding Remarks

This study has presented a comprehensive evaluation of flow matching models for tabular data synthesis, comparing them with state-of-the-art diffusion-based approaches. Our results demonstrate that flow matching—particularly TabbyFlow—consistently achieves higher utility than diffusion baselines, while disclosure risk follows patterns determined by the empirical relationship between model fidelity and privacy exposure, rather than optimisation for risk reduction. Across diverse datasets, the OT trajectory consistently achieves balanced performance with high utility and reduced sensitivity to integration time, while the VP trajectory enhances privacy protection with lower risk at the expense of some utility. In practice, trajectory selection should be guided by specific application requirements.

TabbyFlow further distinguishes itself through its computational efficiency, achieving competitive results with less function evaluations (NFEs). In practice, we recommend the OT path as the default configuration, with early stopping ($0.9 \leq t_{\text{ode}} < 1.0$) and 64–128 NFEs to balance between synthetic data quality and computational cost. Even ultra-low settings (4-8 NFEs) remain viable for resource-constrained environments.

The introduction of stochasticity via SDE solvers produced nuanced outcomes. While SDE formulations theoretically preserve marginal distributions, empirical performance varied between datasets. In several cases, SDE sampling simultaneously improved utility and reduced risk, demonstrating that stochastic dynamics can act as a beneficial regulariser or enhance exploration of the data space. However, these benefits were not universal, indicating that the effectiveness of SDE is context-dependent. Therefore, SDE-based sampling should be considered an optional empirical extension, subject to dataset-specific validation.

Overall, flow matching with the OT paths emerges as an efficient approach to tabular data synthesis. Future work could explore hybrid trajectories, incorporate formal privacy guarantees such as differential privacy, and investigate more thoroughly the dataset-specific factors that drive the effectiveness of both ODE and SDE solvers.

[2]e.g., `clip(1-1/(1-t), 0, m)`, m is the maximum value allowed

References

Michael S. Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants, 2023. URL https://arxiv.org/abs/2209.15571.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 2023. URL https://arxiv.org/abs/2303.08797.

Article 29 Data Protection Working Party. Opinion 05/2014 on anonymisation techniques, 2025. URL https://ec.europa.eu/justice/article-29/documentation/opinion-recommendation/files/2014/wp216_en.pdf.

Black Forest Labs. Flux.1 kontext: Flow matching for in-context image generation and editing in latent space, 2025. URL https://arxiv.org/abs/2506.15742.

Quan Dao, Hao Phung, Binh Nguyen, and Anh Tran. Flow matching in latent space, 2023. URL https://arxiv.org/abs/2307.08698.

Floor Eijkelboom, Grigory Bartosh, Christian Andersson Naesseth, Max Welling, and Jan-Willem van de Meent. Variational flow matching for graph generation, 2024. URL https://arxiv.org/abs/2406.04843.

Floor Eijkelboom, Heiko Zimmermann, Sharvaree Vadgama, Erik J Bekkers, Max Welling, Christian A. Naesseth, and Jan-Willem van de Meent. Controlled generation with equivariant variational flow matching. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=YSVSMV0lXQ.

Mark Elliot, Claire Little, and Richard Allmendinger. Do samples taken from a synthetic microdata population replicate the relationship between samples taken from an original population? In *UNECE Expert Meeting on Statistical Data Confidentiality 2023, 26-28 September 2023, Wiesbaden*, 2023.

Tobias Sebastian Finn, Marc Bocquet, Pierre Rampal, Charlotte Durand, Flavia Porro, Alban Farchi, and Alberto Carrassi. Generative ai models enable efficient and physically consistent sea-ice simulations. *arXiv preprint arXiv:2508.14984*, 2025.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. URL https://arxiv.org/abs/1406.2661.

Andrés Guzmán-Cordero, Floor Eijkelboom, and Jan-Willem van de Meent. Exponential family variational flow matching for tabular data generation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=kjtvCSkSsy.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 12 2013. URL https://arxiv.org/abs/1312.6114v10.

Akim Kotelnikov, Dmitry Baranchuk, Ivan Rubachev, and Artem Babenko. TabDDPM: Modelling tabular data with diffusion models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17564–17579. PMLR, 7 2023. URL https://proceedings.mlr.press/v202/kotelnikov23a.html.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling, 2023. URL https://arxiv.org/abs/2210.02747.

Claire Little, Mark Elliot, and Richard Allmendinger. *Comparing the Utility and Disclosure Risk of Synthetic Data with Samples of Microdata*, pp. 234–249. Springer International Publishing, 2022. ISBN 9783031139451. doi: 10.1007/978-3-031-13945-1_17. URL http://dx.doi.org/10.1007/978-3-031-13945-1_17.

Claire Little, Richard Allmendinger, and Mark Elliot. Synthetic census microdata generation: A comparative study of synthesis methods examining the trade-off between disclosure risk and utility. *Journal of Official Statistics*, September 2024. ISSN 2001-7367. doi: 10.1177/0282423x241266523. URL http://dx.doi.org/10.1177/0282423X241266523.

Qiang Liu. Rectified flow: A marginal preserving approach to optimal transport, 2022. URL https://arxiv.org/abs/2209.14577.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow, 2022. URL https://arxiv.org/abs/2209.03003.

Răzvan-Andrei Matişan, Vincent Tao Hu, Grigory Bartosh, Björn Ommer, Cees GM Snoek, Max Welling, Jan-Willem van de Meent, Mohammad Mahdi Derakhshani, and Floor Eijkelboom. Purrception: Variational flow matching for vector-quantized image generation. *arXiv preprint arXiv:2510.01478*, 2025.

Markus Mueller, Kathrin Gruber, and Dennis Fok. Continuous diffusion for mixed-type tabular data. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=QPtoBPn4lZ.

Beata Nowok, Gillian M Raab, and Chris Dibben. synthpop: Bespoke Creation of Synthetic Data in R. *Journal of Statistical Software*, 74(11 SE - Articles):1–26, 10 2016. doi: 10.18637/jss.v074.i11. URL https://www.jstatsoft.org/index.php/jss/article/view/v074i11.

Nian Ran, Bahrul Nasution, Claire Little, Richard Allmendinger, and Mark Elliot. Multi-objective evolutionary GAN for tabular data synthesis. In *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 394–402, 2024. URL https://doi.org/10.1145/3638529.3654052.

Steven Ruggles and David Van Riper. The role of chance in the census bureau database reconstruction experiment. *Minnesota Population Center Working Paper Series*, 2021. doi: 10.18128/MPC2021-01.V2. URL https://assets.ipums.org/_files/mpc/wp2021-01.v2.pdf.

Kristiyan Sakalyan, Alessandro Palma, Filippo Guerranti, Fabian J Theis, and Stephan Günnemann. Modeling microenvironment trajectories on spatial transcriptomics with nicheflow. In *ICML 2025 Generative AI and Biology (GenBio) Workshop*, 2025. URL https://openreview.net/forum?id=YgDKduLX0b.

Johannes Schusterbauer, Ming Gui, Pingchuan Ma, Nick Stracke, Stefan A. Baumann, Vincent Tao Hu, and Björn Ommer. Boosting latent diffusion with flow matching. In *ECCV*, 2024.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In Francis Bach and David Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 7 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Jennifer Taub, Mark Elliot, Gillian Raab, Anne-Sophie Charest, Cong Chen, Christine M. O'Keefe, Michelle Pistner Nixon, Joshua Snoke, and Aleksandra Slavkovic. The synthetic data challenge. In *Joint UNECE/Eurostat Work Session on Statistical Data Confidentiality, 29-31 October 2019, the Hague, the Netherlands*, 2019.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport, 2024. URL https://arxiv.org/abs/2302.00482.

Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. Modeling tabular data using conditional gan. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/254ed7d2de3b23ab10936522dd547b78-Paper.pdf.

Olga Zaghen, Floor Eijkelboom, Alison Pouplin, and Erik J. Bekkers. Towards variational flow matching on general geometries, 2025a. URL https://arxiv.org/abs/2502.12981.

Olga Zaghen, Floor Eijkelboom, Alison Pouplin, Cong Liu, Max Welling, Jan-Willem van de Meent, and Erik J. Bekkers. Riemannian variational flow matching for material and protein design, 2025b. URL https://arxiv.org/abs/2502.12981.

Xuanran Zhai and Ce Hao. Vfp: Variational flow-matching policy for multi-modal robot manipulation, 2025. URL https://arxiv.org/abs/2508.01622.

Hengrui Zhang, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Xiao Qin, Christos Faloutsos, Huzefa Rangwala, and George Karypis. Mixed-type tabular data synthesis with score-based diffusion in latent space. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4Ay23yeuz0.

A Related Work

Diffusion Models for Tabular Data. Denoising diffusion probabilistic models (DDPMs) have shown great success across domains like vision and language, but their application to tabular data poses unique challenges due to feature heterogeneity and small dataset sizes. TabDDPM (Kotelnikov et al. 2023) addresses this through a dual process: Gaussian diffusion for continuous variables and multinomial diffusion for categorical features. It achieves state-of-the-art (SOTA) utility and privacy performance, outperforming the GAN and VAE baselines. However, its computational cost remains high because of iterative sampling and separate handling of feature types, which may weaken correlation modelling. TabSyn (Zhang et al. 2024) employs a transformer-VAE architecture followed by a diffusion model, while the continuous diffusion model for mixed type tabular data (CDTD) (Mueller et al. 2025) is a score-matching diffusion model that noises continuous values and categorical embeddings and samples via ODE integration. These two methods improve the utility of tabular data, particularly compared to TabDDPM and other recent SOTAs.

FM and its Variational Extensions. Flow matching (FM) (Lipman et al. 2023; Liu et al. 2022; Albergo et al. 2023) offers an alternative to diffusion by learning the deterministic dynamics of probability paths. FM has also been developed in terms of latent models for image generation (Dao et al. 2023; Schusterbauer et al. 2024). In addition, VFM (Eijkelboom et al. 2024) reformulates FM as a variational inference problem, allowing flow matching through a variational distribution. Consequently, VFM can be implemented to mixed-type data, such as tabular data, through mean-field variational inference. On the other hand, (Guzmán-Cordero et al. 2025) extended the VFM specific for the exponential family that also supports mixed-type variables through moment matching and implemented it in tabular data, called TabbyFlow. In addition to tabular, VFM has also been implemented for image generation Matişan et al. (2025), controlled molecular generation (Eijkelboom et al. 2025), non-Euclidean geometry (Zaghen et al. 2025a;b), and climate modeling (Finn et al. 2025).

FM Trajectory. Trajectory design is of paramount importance in the field of FM. The paths of optimal transport (OT) are widely preferred because of their straightforward linear interpolation. In contrast, the variance-preserving (VP) path, which takes inspiration from diffusion models, is somewhat underrated (Lipman et al. 2023; Liu et al. 2022). Recent studies also introduced different trajectories such as cosine Albergo & Vanden-Eijnden (2023), logit-normal Black Forest Labs (2025). Although the theoretical framework of FM accommodates both types of path, there is a notable gap in empirical evaluation concerning the practical application of different paths, particularly in the realm of tabular data. To address this gap, we present the potential of different paths for tabular data synthesis, despite the fact that OT is the primary implementation.

B Algorithms

B.1 TabSynFlow

Algorithm 1 An iteration of TabSynFlow training (flow matching part)

Input: Latent representation from VAE $\mathcal{Z}_1$, time function for mean α_t, time function for variance σ_t, conditional velocity field NN(.)

$z_1 \sim p_{\mathcal{Z}_1}$
$z_0 \sim \mathcal{N}(0, I)$
$t \sim \text{Uniform}(0, 1)$
$z_t = \alpha_t z_1 + \sigma_t z_0$
Calculate $u_t(z_t|z_1)$ using equation 2.4
Predict $v_t^\theta(z_t) = \text{NN}(z_t, t)$
Calculate $\mathcal{L}_{\text{TabSynFlow}}$

Algorithm 2 TabSynFlow sample generation

Input: Trained conditional velocity field NN(.), discrete step interval Δ, trained decoder from VAE Dec(.)

$\tilde{z}_0 \sim \mathcal{N}(0, I)$.
$t = 0$
while $t \leq 1$ **do**
    $\tilde{z}_{t+\Delta} = \tilde{z}_t + \Delta \cdot \text{NN}(\tilde{z}_t, t)$
    $t \mathrel{+}= \Delta$
end while
$\tilde{x} = \text{Dec}(\tilde{z}_1)$
$\tilde{\mathcal{D}} = \text{InverseTransform}(\tilde{x})$
Return $\tilde{\mathcal{D}}$

B.2 TabbyFlow

Algorithm 3 TabbyFlow training iteration

Input: Preprocessed data, Function of time for mean α_t, function of time for variance σ_t, variational parameter model NN(.)

$x_1 \sim p_{data}$
$x_0 \sim \mathcal{N}(0, I)$
$t \sim \text{Uniform}(0, 1)$
$x_t = \alpha_t x_1 + \sigma_t x_0$
Predict $\theta_t(x_t) = \text{NN}(x_t, t)$
Calculate $\mathcal{L}_{\text{TabbyFlow}}(\theta)$

Algorithm 4 TabbyFlow sample generation

Input: Time function for mean α_t, time function for variance σ_t, variational parameter model NN(.), discrete step interval Δ, noise schedule g_t (for SDE)

$\tilde{x}_0 \sim \mathcal{N}(0, I)$
$t = 0$
while $t \leq 1$ **do**
  $\tilde{\theta}_t(\tilde{x}_t) = \text{NN}(\tilde{x}_t, t)$
  Calculate $v_t^\theta(\tilde{x}_t)$ using equation 3.5
ODE:
  $\tilde{x}_{t+\Delta} = \tilde{x}_t + \Delta \cdot v_t^\theta(\tilde{x}_t)$
SDE:
  Calculate $s_t^\theta(x)$ using equation 3.6
  $\tilde{x}_{t+\Delta} = \tilde{x}_t + \Delta\left[v_t^\theta(\tilde{x}_t) + \frac{g_t^2}{2} s_t^\theta(\tilde{x}_t)\right] + g_t\sqrt{\Delta}\mathcal{N}(0, I)$
  $t{+}{=}\Delta$
end while
$\tilde{\mathcal{D}} = \text{InverseTransform}(\tilde{x}_1)$
Return $\tilde{\mathcal{D}}$

C Evaluation Details

This section provides detailed information on the evaluation metrics and implementation procedures used to assess the quality and privacy aspects of the synthetic tabular data generated in our experiments. This additional context aims to ensuring transparency and reproducibility, allowing readers to fully understand how each score is derived and how our experimental protocol aligns with established practices in tabular data synthesis evaluation.

C.1 Utility Metrics

Utility measures the extent to which synthetic data faithfully reproduces the statistical relationships and analytical properties of the original data. We employ two complementary utility metrics:

C.1.1 Ratio of Counts (ROC)

ROC evaluates the fidelity of frequency tables and cross-tabulations by comparing cell counts between original and synthetic data. For each cell, the ROC is:

$$\text{ROC}_{\text{cell}} = \frac{\min(\mathcal{Y}_{count}, \mathcal{Y}'_{count})}{\max(\mathcal{Y}_{count}, \mathcal{Y}'_{count})}, \tag{C.1}$$

where $\mathcal{Y}_{count}$ and $\mathcal{Y}'_{count}$ denote cell counts in the original and synthetic data respectively. A value of 1 indicates perfect agreement; 0 indicates extreme disagreement.

ROC is computed across two types of tabulations:

- **Univariate**: Frequency distribution of each key variable independently
- **Bivariate**: Cross-tabulation of all pairs of key variables

The final ROC_{uni} score is the mean across all univariate tables, and ROC_{biv} is the mean across all bivariate tables, i.e. cross-tabulation.

C.1.2 Confidence Interval Overlap (CIO)

CIO assesses how well statistical inference (specifically logistic regression) conducted on synthetic data recovers the confidence intervals from the original data. For each regression coefficient, CIO calculates the normalized overlap of 95% confidence intervals:

$$\text{CIO}_k = 0.5 \left[\frac{\text{overlap}}{u - l} + \frac{\text{overlap}}{u' - l'} \right], \tag{C.2}$$

where $[l, u]$ and $[l', u']$ are the confidence intervals from original and synthetic models respectively, and overlap $= \min(u, u') - \max(l, l')$. A CIO of 1 indicates identical intervals, whereas CIO < 0 indicates no overlap, therefore it is truncated to zero (Elliot et al. 2023). The final CIO score is the mean overlap across all regression coefficients and all fitted models.

C.1.3 Aggregate Utility Score

The final utility score for each dataset is computed as:

$$\text{Utility} = \frac{\text{ROC}_{\text{uni}} + \text{ROC}_{\text{biv}} + \text{CIO}}{3} \tag{C.3}$$

For each dataset, we compute ROC across all univariate and bivariate frequency tables, and CIO for regression models predicting a defined target variable. Mainly, we used logistic regression, except for target variable CreditScore and EstimatedSalary in Churn data that uses linear regression since they are continuous variables. The pseudocode to compute ROC and CIO can be seen in Algorithms 5 and 6.

Algorithm 5 ROC calculation

Input: Original data $\mathcal{D}$, synthetic data $\mathcal{D}'$, key variables K

for each variable k in K **do**
  for each category c of k **do**
    Compute $\text{ROC}_{\text{uni}}[c] = \frac{\min(\mathcal{Y}_{count}(c), \mathcal{Y}'_{count}(c))}{\max(\mathcal{Y}_{count}(c), \mathcal{Y}'_{count}(c))}$
  end for
end for
$\text{ROC}_{\text{uni}} \leftarrow$ mean of $\text{ROC}_{uni}[c]$ over all variables and categories

for each pair (k_1, k_2) in K **do**
  for each cell (c_1, c_2) **do**
    Compute $\text{ROC}_{\text{biv}}[c_1, c_2] = \frac{\min(\mathcal{Y}_{count}(c_1,c_2), \mathcal{Y}'_{count}(c_1,c_2))}{\max(\mathcal{Y}_{count}(c_1,c_2), \mathcal{Y}'_{count}(c_1,c_2))}$
  end for
end for
$\text{ROC}_{\text{biv}} \leftarrow$ mean of $\text{ROC}_{\text{biv}}[c_1, c_2]$ over all pairs and cells
Return ROC_{uni}, ROC_{biv}

Algorithm 6 CIO calculation

Input: Original data $\mathcal{D}$, synthetic data $\mathcal{D}'$, key variables K, target variable T

Fit regression model (predictors K, target T) on $\mathcal{D}$
Fit regression model (predictors K, target T) on $\mathcal{D}'$
for each coefficient i **do**
  Compute $[l, u]$ from model on $\mathcal{D}$
  Compute $[l', u']$ from model on $\mathcal{D}'$
  overlap $= \max(0, \min(u, u') - \max(l, l'))$
  $\text{CIO}_i = 0.5 \left[\frac{\text{overlap}}{u-l} + \frac{\text{overlap}}{u'-l'} \right]$
end for
CIO $\leftarrow$ mean of CIO_i over all coefficients
Return CIO

C.2 Disclosure Risk Metric

We measure disclosure risk as the probability that an adversary, possessing the quasi-identifier (key variables, K), can correctly guess the sensitive value (target T) for a record by matching between the synthetic and original data. This approach, adapted from Taub et al. (2019), quantifies the additional privacy risk introduced by synthetic data at a per-record granularity, adjusting for the baseline (based on a random draw from T), defined by the distribution of T within each equivalence class (combination of K) in the original data.

For each synthetic record, we first identify the group of records in the synthetic data that share the same values for the key variables, as well as the more specific group that also matches on the target variable. The proportion is then calculated as the number of records sharing both key and target values divided by the number sharing only the key values. This proportion is used to filter which synthetic records are included in the TCAP calculation: only those in groups where the proportion meets or exceeds a specified threshold (represented by the parameter τ - which represents an assumed lower bound on the acceptable level of uncertainty for the adversary) are considered, while records in less concentrated groups are excluded. This filtering is applied solely within the synthetic data; the actual TCAP risk for each retained record is subsequently calculated based on the target distribution in the original data for the corresponding key group.

Targeted Correct Attribution Probability ($TCAP$) estimates the probability that the adversary can match the synthetic target T'_j using the equivalence class from the original data:

$$TCAP'_j = P_{\mathcal{D}}(T'_j|K'_j) = \frac{\sum_{i=1}^{n} \mathbb{I}[T_i = T'_j,\, K_i = K'_j]}{\sum_{i=1}^{n} \mathbb{I}[K_i = K'_j]} \tag{C.4}$$

Intuitively, $TCAP'_j$ shows the chance that target T'_j could arise in the real data for key K'_j—that is, it reflects the "true match" risk based on empirical frequencies in the original dataset.

Within-Equivalence Class Attribute Probability ($WEAP$) serves as a baseline: if the adversary could only guess randomly within the equivalence class K'_j, their success probability would be

$$WEAP_j = P_{\mathcal{D}}(T_j|K_j) = \frac{\sum_{i=1}^{n} \mathbb{I}[T_i = T_j, K_i = K_j]}{\sum_{i=1}^{n} \mathbb{I}[K_i = K_j]} \tag{C.5}$$

This is identical in calculation to $TCAP'_j$, but conceptually $WEAP_j$ captures the average success rate without access to synthetic information—i.e., a "random guess" adversary.

Per-record disclosure risk is then the normalised improvement of adversarial success above this baseline:

$$\text{Risk}_j = \max\left\{0, \frac{TCAP'_j - WEAP_j}{1 - WEAP_j}\right\} \tag{C.6}$$

A value near 0 indicates synthetic data provides no additional risk beyond baseline, while a value near 1 signals a maximum increase in disclosure risk. Negative values (adversary performs worse than random) are truncated to zero by convention (Ran et al. 2024; Elliot et al. 2023).

Intuition: This metric isolates the extra "leakage" in matching risk—beyond what already exists in the univariate distribution of the target — arising from access to the synthetic data. It thus accounts for singling out, linkage and inference risks (due to the empirical distribution of sensitive values per equivalence class) identified by Article 29 Data Protection Working Party (2025) as being critical determinants of risks to anonymity. The disclosure risk for the dataset is the mean of R_j over all synthetic records meeting the τ threshold. The pseudocode to calculate TCAP can be seen in Algorithm 7

Algorithm 7 TCAP calculation

Input: Original data $\mathcal{D}$, synthetic data $\mathcal{D}'$, key variables K, target variable T, threshold τ

for each synthetic record j in $\mathcal{D}'$ **do**
    Identify equivalence class K'_j and $[T'_j, K'_j]$ in synthetic data
    $p' \leftarrow \frac{\mathbb{I}[T'_j, K'_j]}{\mathbb{I}[K'_j]}$
    if $p' < \tau$ **then**
        Continue to next j ▷ Skip if class too small
    end if
    Identify all records in $\mathcal{D}$ with $K_i = K'_j$, denote as class C
    if C is empty **then**
        $R_j \leftarrow 0$, go to next j
    end if
    Calculate TCAP'_j using equation C.4
    Calculate WEAP_j using equation C.5
    Calculate Risk_j using equation C.6
end for
Risk $\leftarrow$ mean of all R_j

C.3 Reproducibility

Full implementation code (Python) and detailed parameter configurations are available in the accompanying repository on `github.com/xxxx/tabular-flow-matching`[3]. This appendix provides the procedural summary sufficient for manual replication or verification by independent researchers. Table 5- 7 showed detailed variables implemented in the utility and risk evaluation.

Table 5: Variables used for ROC evaluation in each dataset.

Dataset	**ROC Variables (Key/Categorical features)**
UK	ECONPRIM, ETHGROUP, LTILL, QUALNUM, SEX, SOCLASS, TENURE, MSTATUS
Canada	ABIDENT, SEX, TENURE, URBAN, BPLMOM, BPLPOP, CITIZEN, LANG, MARST, RELATE, MINORITY, RELIG, BPL
Fiji	PROV, TENURE, RELATE, SEX, ETHNIC, MARST, RELIGION, BPLPROV, RESPROV, RESSTAT, SCHOOL, TRAVEL
Rwanda	STATUS, SEX, URBAN, OWNERSH, DISAB2, DISAB1, RELATE, RELIG, HINS, NATION, BPL
Indonesia	OWNERSHIP, LANDOWN, RELATE, SEX, MARST, HOMEMALE, RELIGION, SCHOOL, LIT, EDATTAIND, DISABLED
Adult	workclass, education, marital-status, occupation, relationship, race, sex, native-country, income
Churn	Geography, Gender, Tenure, NumOfProducts, HasCrCard, IsActiveMember, Exited

[3] Link is anonymised on review process

Table 6: Variables used for CIO evaluation (targets and explanatory variables) in each dataset. Note that for CIO mainly uses logistic regression as the model, considering the categorical target variables, except the CreditScore and EstimatedSalary in Churn that uses linear regression since they are continuous variables.

Dataset	Target Variables	Explanatory (Key) Variables
UK	TENURE, MSTATUS	ECONPRIM, ETHGROUP, LTILL, QUALNUM, SEX, SOCLASS, TENURE, MSTATUS, AGE
Canada	TENURE, MARST	ABIDENT, CLASSWK, DEGREE, EMPSTAT, SEX, URBAN, TENURE, MARST, AGE, HRSWK, INCTOT, WKSWORK
Fiji	TENURE, MARST	CLASSWKR, ETHNIC, RELIGION, EDATTAIN, SEX, PROV, TENURE, MARST, AGE
Rwanda	OWNERSH, MARST	DISAB1, EDCERT, CLASSWK, LIT, RELIG, SEX, OWNERSH, MARST, AGE
Indonesia	OWNERSHIP, MARST	LANDOWN, RELATE, SEX, HOMEFEM, HOMEMALE, RELIGION, LIT, SCHOOL, EDATTAIND, DISABLED, OWNERSHIP, MARST, AGE
Adult	income, marital-status	workclass, education-num, marital-status, occupation, relationship, race, sex, native-country, income, age, fnlwgt, capital-gain, capital-loss, hours-per-week
Churn	Exited, CreditScore, EstimatedSalary	Geography, Gender, Tenure, NumOfProducts, HasCrCard, IsActiveMember, Exited, CreditScore, Age, Balance, EstimatedSalary

Table 7: Variables used for TCAP calculation (target and key classes) in each dataset.

Dataset	Target Variables	Key Variables for Equivalence Class Definition
UK	TENURE, MSTATUS	ECONPRIM, ETHGROUP, LTILL, QUALNUM, SEX, SOCLASS, TENURE, MSTATUS
Canada	TENURE, MARST	ABIDENT, CLASSWK, DEGREE, EMPSTAT, SEX, URBAN, TENURE, MARST
Fiji	TENURE, MARST	CLASSWKR, ETHNIC, RELIGION, EDATTAIN, SEX, PROV, TENURE, MARST
Rwanda	OWNERSH, MARST	DISAB1, EDCERT, CLASSWK, LIT, RELIG, SEX, OWNERSH, MARST
Indonesia	OWNERSHIP, MARST	LANDOWN, RELATE, SEX, HOMEFEM, HOMEMALE, RELIGION, LIT, SCHOOL, EDATTAIND, DISABLED, OWNERSHIP, MARST
Adult	income, marital-status	workclass, education-num, marital-status, occupation, relationship, race, sex, native-country, income
Churn	Exited, CreditScore, EstimatedSalary	Geography, Gender, Tenure, NumOfProducts, HasCrCard, IsActiveMember, Exited

D Additional and Detailed Results

D.1 Variance Comparison Results

The comparative analysis of utility and risk across datasets in Table 8 highlights the performance distinction between the theory-based and our approaches. In most cases, our method consistently achieves higher utility scores than the theory method, indicating better preservation of data usefulness for downstream tasks. For instance, in the UK dataset, our approach yields a utility of 0.8333 ± 0.0191 compared to 0.5137 ± 0.0072 for the theory method. However, higher utility in our approach is often accompanied by slight increased risk values. For example, the risk for our approach in the UK dataset is 0.4889 ± 0.0043, higher than that of the theoretical variance (0.3998 ± 0.0074), but still having significantly higher utility. Meanwhile, the only better performance of the theoretical approach is in Rwanda (theoretical: U=0.7374 ± 0.0155, R=0.5019ś0.0125; ours: U=0.7284ś0.0231, R=0.5379ś0.0139). Overall, our proposed approach generally provides superior performance across most datasets.

Table 8: Comparative performance of synthetic data results using theoretical (left) and relaxed (right) variance for TabbyFlow-OT. The table highlights that relaxing the theoretical variance to learn the distribution of continuous variables frequently leads to better performance.

	$0.5A_t(x_t)^{-2}$		$0.5A_t(x_t)^{-1}$ (**ours**)	
dataset	Utility	Risk	Utility	Risk
UK	0.5137 ± 0.0072	0.3998 ± 0.0074	0.8333 ± 0.0191	0.4889 ± 0.0043
Indonesia	0.5149 ± 0.0085	0.3280 ± 0.0141	0.6671 ± 0.0046	0.4114 ± 0.0149
Canada	0.7479 ± 0.0109	0.2961 ± 0.0175	0.7718 ± 0.0211	0.3206 ± 0.0146
Fiji	0.7002 ± 0.0228	0.5574 ± 0.0060	0.7263 ± 0.0205	0.5705 ± 0.0063
Rwanda	0.7319 ± 0.0199	0.5019 ± 0.0125	0.7284 ± 0.0231	0.5379 ± 0.0139
Adult	0.7374 ± 0.0155	0.4917 ± 0.0162	0.7720 ± 0.0231	0.5443 ± 0.0177
Churn	0.7613 ± 0.0103	0.0635 ± 0.0792	0.7966 ± 0.0098	0.0773 ± 0.0579

D.2 Main Comparison on ID, AD, CH Datasets

Table 9 presents a comparative evaluation of six synthetic data generation algorithms across four datasets (UK, ID, AD, CH), highlighting the two best-performing algorithms per dataset. Overall, TabSynFlow-OT consistently achieves superior utility in the ID and CH datasets compared to diffusion baselines, and matches TabSyn while delivering lower disclosure risk in AD. Notably, TabSynFlow-OT also provides reduced risk in ID and CH, making it an effective choice for synthesising tabular data which increases utility and reduces privacy risk.

By contrast, TabbyFlow-OT yields the highest utility for AD, a result that is accompanied by increased disclosure risk, while in ID, it generates synthetic data with lower utility but achieves a substantial reduction in risk. This pattern illustrates how variations in model performance and privacy exposure reflect the empirical relationship between utility and risk across different datasets. TabSyn consistently ranks as the second-best in utility across all three datasets.

By combining with the main text, these results underscore the strength of flow matching approaches for tabular data synthesis, particularly the flexibility of TabSynFlow-OT as an alternative to diffusion-based models in extreme number of observations, i.e. too high or too low.

D.3 Number of Function Evaluations

Figure 3 shows how computational cost is related to the quality of the output between methods (TabSyn, TabSynFlow and TabbyFlow) and datasets. In general, both utility and risk increase as NFEs grow. In four datasets (UK, RW, CA, FI), TabbyFlow seems to outperform TabSyn. However, in ID and CH, TabSynFlow-OT outperforms TabSyn. Therefore, if one has a limited computational budget, they may choose TabbyFlow or TabSynFlow using the OT path as the algorithm that performed best on low-step setting among all algorithms, particularly TabSyn.

D.4 Integration Time

While the original implementation of flow matching uses full ODE integration, it is possible to terminate the ODE at intermediate times—a principle explored in this section. Our analysis of ODE integration times (t_{ode}) in Figure 4, spanning datasets including Rwanda, Adult, Churn, UK, Indonesia, Canada, and Fiji, offers critical insights into how utility and risk evolve during synthetic data generation. For TabSynFlow

Table 9: Comparative Performance of six synthetic data generation algorithms across three datasets (ID, AD, CH). Bold and underline indicate the first and second best performing algorithms for each dataset, respectively. The table highlights the comparable performance of TabbyFlow and TabSyn, while TabSynFlow-OT stands out as a strong alternative—particularly in the ID dataset, where TabbyFlow exhibits higher risk in conjunction with generated high utility synthetic data.

Evaluation	Algorithm	ID	AD	CH
Utility (↑)	TabDDPM	0.7993±0.0108	0.6676±0.0092	0.8227±0.0130
	TabSyn	0.8707±0.0264	0.7596±0.0139	0.8663±0.0124
	TabSynFlow-OT	**0.8981±0.0300**	0.7560±0.0139	**0.8784±0.0184**
	TabSynFlow-VP	0.7814±0.0100	0.6843±0.0100	0.8126±0.0149
	TabbyFlow-OT	0.6671±0.0046	**0.7720±0.0231**	0.7966±0.0098
	TabbyFlow-VP	0.6488±0.0103	0.7581±0.0165	0.8066±0.0102
Risk (↓)	TabDDPM	0.6493±0.0116	0.4369±0.0134	0.0718±0.0610
	TabSyn	0.6624±0.0122	0.4805±0.0207	0.0630±0.0625
	TabSynFlow-OT	0.6616±0.0119	0.4693±0.0149	0.0429±0.0438
	TabSynFlow-VP	0.6464±0.0140	**0.4131±0.0199**	**0.0246±0.0345**
	TabbyFlow-OT	0.4114±0.0149	0.5443±0.0177	0.0773±0.0579
	TabbyFlow-VP	**0.3918±0.0244**	0.5037±0.0172	0.0753±0.0708

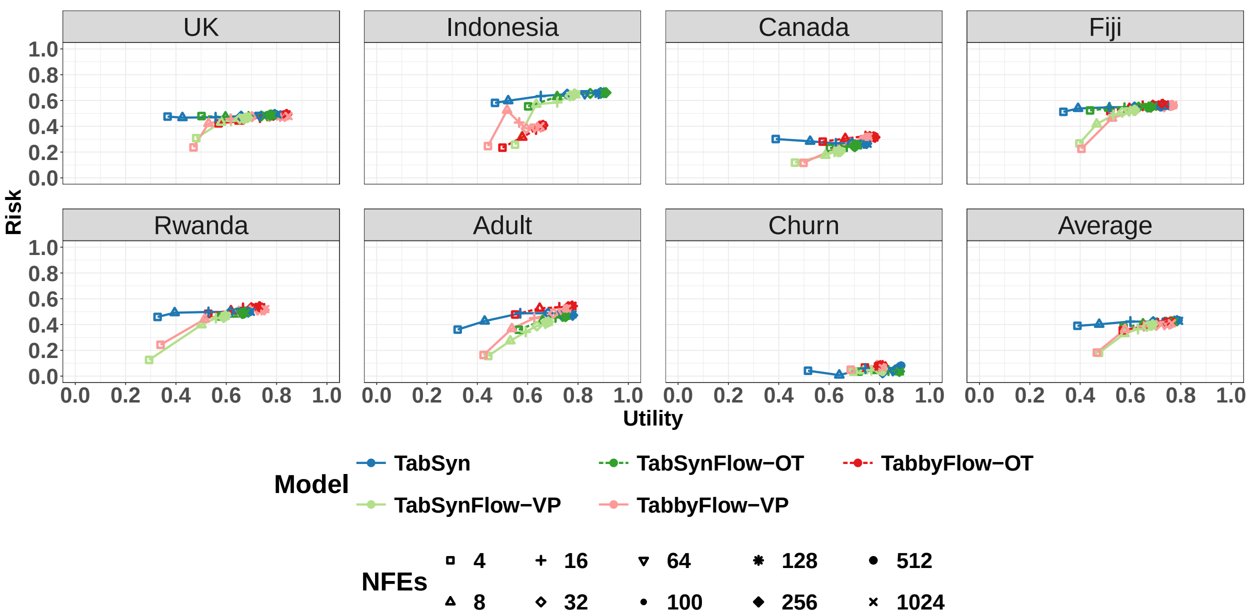

Figure 3: Utility and risk evaluation of TabSynFlow and TabbyFlow based on dataset and average using different number of evaluations with formula $2^n, n = [2, ..., 10]$. TabbyFlow-OT (and TabSynFlow-OT) achieve strong utility at low NFEs (≤ 100) and tend to converge by ≈128, whereas TabSyn requires higher NFEs to catch up—so OT + flow-matching is attractive under tight compute budgets.

with optimal transport (OT), early stopping at $t_{ode} = 0.6$ consistently produces high-utility synthetic data, whereas variance preserving (VP) paths in both TabSynFlow and TabbyFlow tend to underperform at this stage. This indicates that OT trajectories better retain the data structure in earlier integration steps, while VP introduces noise prematurely. Notably, the Rwanda dataset presents an interesting anomaly: TabbyFlow-OT achieves slightly better utility than VP methods at $t_{ode} = 0.6$, which may be attributed to the dataset's relatively simple feature distributions.

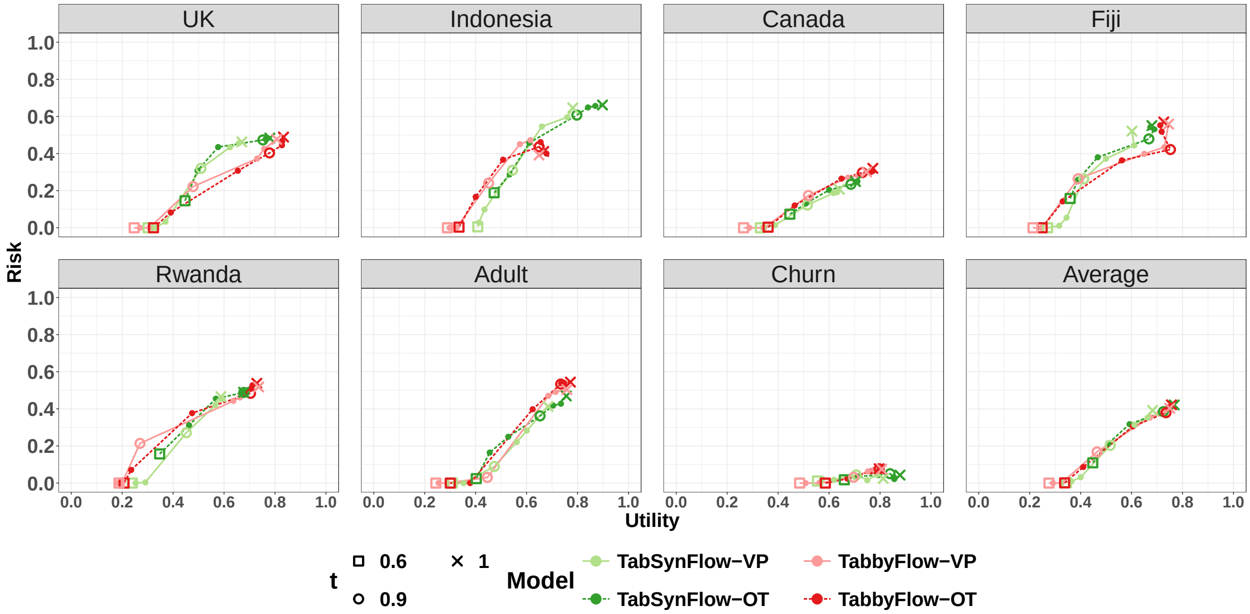

Figure 4: Effect of ODE integration time on utility–risk (per dataset + average) for TabSynFlow and TabbyFlow. Utility and risk evaluation of TabSynFlow and TabbyFlow based on dataset and average on different integration time from $t_{ode} = [0.6, 1]$ with interval 0.1. Colors encode method and path (VP vs OT). OT achieves high-utility solutions early (e.g., $t_{ode} = 0.6$), while pushing to $t_{ode} \to 1$ often reduces utility and/or increases risk—so early stopping is preferable.

Figure 5 showed the utility and risk in details when $0.9 \le t_{ode} \le 1$. At $t = 0.9$, OT methods maintain superior performance, particularly for the UK and Adult datasets, although the gap between OT and VP narrows for the latter. This implies dataset-specific sensitivity to VP's noise characteristics. Near full integration ($t_{ode} \to 1.0$), all algorithms exhibit erratic behaviour in some datasets such as Fiji: the utility drops and the risk increases. This problem suggests that flow matching is prone to quality drops when fully integrated.

E Flow Matching Results on Different Interpolants

E.1 Interpolation Path in Flow Matching

A central element in continuous-time generative modelling is the choice of *interpolation schedule*, which determines how the clean data x_1 is progressively transformed into noise, and conversely how the noise is mapped back to the data. This schedule is characterised by two functions: the signal decay factor $\alpha(t)$ and the noise scale $\sigma(t)$. Together, these define the marginal distribution.

$$x_t \sim \mathcal{N}\big(\alpha(t)x_1, \sigma^2(t)I\big),$$

as well as their time derivatives $\dot{\alpha}(t)$ and $\dot{\sigma}(t)$, which govern the dynamics in both ODE and SDE formulations.

Different schedules entail distinct trade-offs between training stability, sample quality, and computational efficiency. While the main text provides the primer for the OT and VP paths, flow matching formulations can also use alternative paths such as variance-exploding (Lipman et al. 2023), cosine (Albergo & Vanden-Eijnden 2023), or logit-normal (Black Forest Labs 2025) interpolants. In this section, we summarise these schedules in two comparative tables: Table 10 for the interpolants $(\alpha(t), \sigma(t))$ and Table 11 for their derivatives $(\dot{\alpha}(t), \dot{\sigma}(t))$.

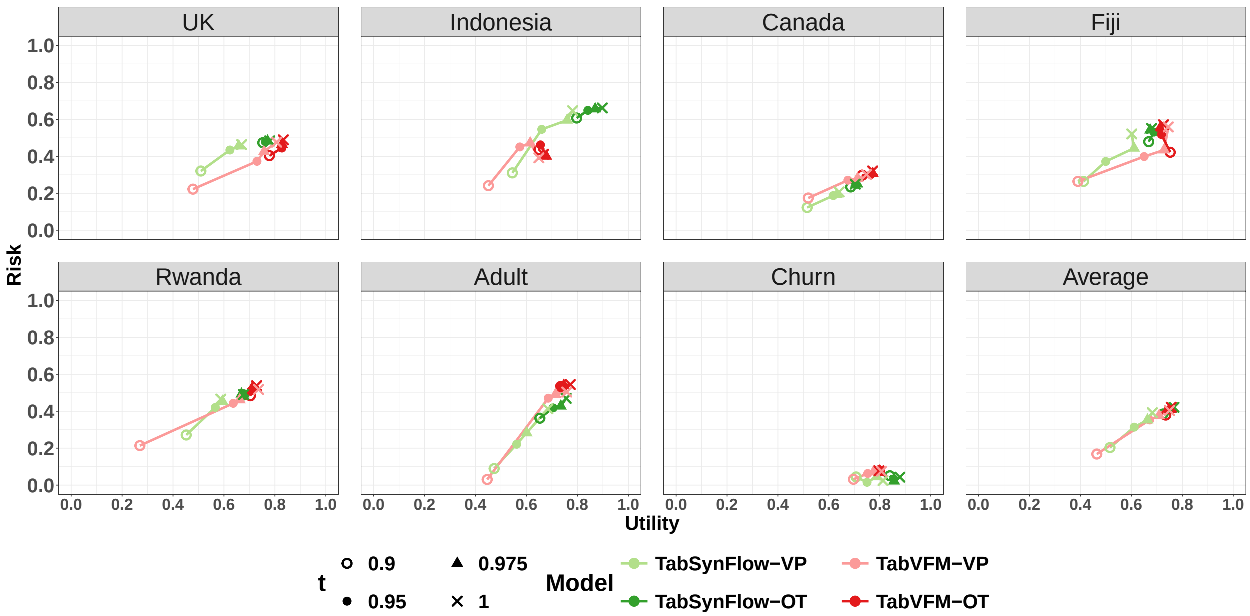

Figure 5: Utility and risk evaluation of TabSynFlow and TabbyFlow based on dataset and average on late integration time $t = [0.9, 0.95, 0.975, 1]$. For selected datasets, we zoom into the late integration window using the same color/marker scheme as Figure 4. Take-home: Near full integration, instability emerges across several datasets (e.g., Fiji), consistent with utility drops and risk increases; OT remains comparatively robust but still degrades as $t_{ode} \to 1$.

Table 10: Interpolants $\alpha(t)$ and $\sigma(t)$ for different schedules.

Interpolation	$\alpha(t)$	$\sigma(t)$
Linear	$1 - t$	t
Variance-Preserving	$\exp\Big(-\frac{1}{2}\int_0^{1-t}\beta(s)\,ds\Big)$	$\sqrt{1-\alpha^2(t)}$
Variance-Exploding	1	$\sigma_{\min}\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)^t$
Cosine	$\sin\big(\frac{\pi}{2}t\big)$	$\cos\big(\frac{\pi}{2}t\big)$
Logit-Normal	$\frac{e^\mu}{e^\mu + (\frac{1}{ct}-1)^\lambda}$ $\mu = \log 3,\ \lambda = 1$	$1 - \alpha(t)$

E.2 Empirical Results on TabSynFlow

Table 12 disseminated the simulation results of TabSynFlow using different paths. Based on the table, it seems that the comparative analysis shows clear distinctions between the paths. optimal transport (OT) offers the best balance of utility and risk. variance preserving (VP) and Cosine paths reduce risk further, making them well suited for privacy-sensitive contexts, though at a slight cost to utility. In contrast, the variance exploding (VE) and Logit paths underperforms on both measures. Overall, OT is a strong default, VP is preferable when privacy is prioritised, and Cosine remains a viable alternative, although path choice should be validated for each dataset.

Table 11: Derivative of interpolants for different schedules.

Interpolation	$\dot{\alpha}(t)$	$\dot{\sigma}(t)$
Linear	-1	1
Variance-Preserving	$-\frac{1}{2}\beta(t)\alpha(t)$	$\frac{1}{2}\,\frac{\beta(t)\alpha^2(t)}{\sigma(t)}$
Variance-Exploding	0	$\sigma(t)\,\ln\left(\frac{\sigma_{\max}}{\sigma_{\min}}\right)$
Cosine	$\frac{\pi}{2}\,\cos\left(\frac{\pi}{2}t\right)$	$-\frac{\pi}{2}\,\sin\left(\frac{\pi}{2}t\right)$
Logit-Normal	$\frac{\lambda\exp(\mu)}{ct^2(e^\mu+(\frac{1}{ct}-1)^\lambda)^2}\left(\frac{1}{ct}-1\right)^{\lambda-1}$	$-\dot{\alpha}(t)$

Table 12: Comparison of utility and risk across different paths on TabSynFlow.

Dataset	OT		VP		VE		Cosine		Logit-Normal	
	Utility	Risk	Utility	Risk	Utility	Risk	Utility	Risk	Utility	Risk
UK	0.7796	0.4834	0.6696	0.4630	0.2387	0.5263	0.7554	0.4802	0.5435	0.4622
Indonesia	0.8981	0.6616	0.7814	0.6464	0.4467	0.6186	0.8732	0.6547	0.5206	0.4782
Canada	0.7035	0.2493	0.6403	0.2073	0.2841	0.3827	0.7021	0.2485	0.4875	0.2233
Fiji	0.6798	0.5506	0.6014	0.5208	0.2800	0.5376	0.7115	0.5526	0.3465	0.4593
Rwanda	0.6765	0.4894	0.5875	0.4655	0.2723	0.5250	0.6681	0.4923	0.4528	0.4513
Adult	0.7560	0.4693	0.6843	0.4131	0.2745	0.5023	0.7466	0.4675	0.4429	0.2489
Churn	0.8784	0.0429	0.8126	0.0246	0.4922	0.0216	0.8284	0.0562	0.7189	0.0630
Average	0.7674	0.4209	0.6824	0.3915	0.3269	0.4449	0.7551	0.4217	0.5018	0.3409

E.3 Empirical Results on TabbyFlow

The results on Table 13 highlights clear differences of the TabbyFlow results between paths. optimal transport (OT) delivers strong overall performance and serves as a reliable default, while Logit performs comparably and in some cases slightly better. VP and Cosine stand out as the least risk, making them suitable when privacy is prioritised, though they sacrifice a little utility. Variance exploding (VE), by contrast, shows worst performance, limiting its general usefulness. These findings highlight that trajectory choice should align with application needs: OT offers balanced performance, VP prioritises privacy protection, while Logit and cosine present viable alternatives to OT and VP, respectively.

Table 13: Comparison of Utility and Risk Across Different Paths on TabbyFlow

Dataset	OT		VP		VE		Cosine		Logit-Normal	
	Utility	Risk	Utility	Risk	Utility	Risk	Utility	Risk	Utility	Risk
UK	0.8333	0.4889	0.8066	0.4765	0.3238	0.5331	0.8001	0.4721	0.8201	0.4887
Indonesia	0.6671	0.4114	0.6488	0.3918	0.4519	0.3955	0.6319	0.3856	0.6520	0.3888
Canada	0.7718	0.3206	0.7472	0.3008	0.4083	0.3437	0.7626	0.3102	0.7619	0.3032
Fiji	0.7263	0.5705	0.7451	0.5588	0.3052	0.6000	0.7121	0.5341	0.7303	0.5694
Rwanda	0.7284	0.5379	0.7358	0.5180	0.3267	0.5222	0.7440	0.5138	0.7396	0.5271
Adult	0.7720	0.5443	0.7581	0.5037	0.3097	0.5666	0.7205	0.5107	0.7665	0.5580
Churn	0.7966	0.0773	0.8066	0.0753	0.5563	0.0481	0.7913	0.0841	0.7982	0.0767
Average	0.7565	0.4215	0.7498	0.4036	0.3831	0.4299	0.7375	0.4015	0.7526	0.4160

E.4 SDE Performance on Different g_t

Tables 14 and 15 disseminate the comparison of the difference in utility (ΔU) and risk (ΔR) when using SDE with different g_t on TabbyFlow compared to its ODE counterparts as a baseline (See Table 13). The tables reveal consistently small deviations across all datasets, with the majority of differences within $[-0.02, +0.02]$. This minimal variation strongly validates the theoretical equivalence between the SDE and ODE marginals, demonstrating that SDEs preserve the same distributional properties regardless of the specific g_t function employed. The results confirm that the controlled stochasticity introduced through SDE sampling maintains distributional fidelity while offering additional flexibility in the sampling process.

In particular, several configurations show the desirable outcome of simultaneously improving both utility and reducing risk. For example, Rwanda in the OT configuration achieves ΔU $= +0.0105$ and ΔR $= -0.0054$, while Fiji in the VP configuration achieves ΔU $= +0.0067$ and ΔR $= -0.0017$. These dual improvements indicate that SDE-based approaches can not only match the performance of ODE but, in some cases, produced high utility data with lower privacy risk. This dataset-dependent behavior underscores the importance of empirical selection for g_t in synthesiser design, as the configuration can vary based on feature complexity, data distribution, or presence of rare categories.

Table 14: Difference of utility and risk across different g_t setting in SDE w.r.t. the ODE results on TabbyFlow-OT.

Dataset	OT		VP		VE		Cos		Logit	
	ΔU(↑)	ΔR(↓)	ΔU(↑)	ΔR(↓)	ΔU(↑)	ΔR(↓)	ΔU(↑)	ΔR(↓)	ΔU(↑)	ΔR(↓)
UK	-0.0073	-0.0011	-0.0044	0.0000	-0.0066	-0.0007	-0.0024	0.0003	-0.0103	0.0022
Indonesia	-0.0044	-0.0130	-0.0105	-0.0106	-0.0028	-0.0072	-0.0075	-0.0041	-0.0027	-0.0005
Canada	-0.0011	-0.0020	0.0060	0.0007	0.0004	-0.0026	0.0047	-0.0010	0.0034	-0.0076
Fiji	0.0006	-0.0024	0.0067	-0.0017	0.0042	-0.0027	0.0010	-0.0052	0.0029	-0.0001
Rwanda	0.0105	-0.0054	0.0142	0.0004	0.0093	-0.0050	0.0123	-0.0039	0.0112	-0.0023
Adult	0.0077	0.0012	0.0027	0.0040	0.0047	0.0021	0.0067	0.0038	0.0039	0.0045
Churn	0.0036	-0.0156	0.0021	0.0070	0.0025	-0.0047	0.0024	0.0112	0.0032	0.0210

Table 15: Difference of utility and risk across different g_t setting in SDE w.r.t. the ODE results on TabbyFlow-VP.

Dataset	OT		VP		VE		Cos		Logit	
	ΔU(↑)	ΔR(↓)	ΔU(↑)	ΔR(↓)	ΔU(↑)	ΔR(↓)	ΔU(↑)	ΔR(↓)	ΔU(↑)	ΔR(↓)
UK	-0.0035	-0.0005	-0.0070	-0.0024	-0.0101	-0.0020	-0.0072	-0.0022	-0.0045	-0.0010
Indonesia	-0.0034	0.0028	-0.0069	0.0001	-0.0025	0.0077	-0.0030	0.0023	-0.0034	0.0017
Canada	0.0015	0.0029	0.0025	-0.0027	0.0014	0.0017	0.0026	0.0007	0.0048	0.0076
Fiji	0.0038	-0.0012	0.0011	-0.0017	0.0097	-0.0022	0.0021	-0.0026	0.0095	-0.0006
Rwanda	0.0103	0.0012	0.0038	0.0000	0.0032	-0.0030	0.0084	0.0054	0.0098	-0.0017
Adult	0.0043	-0.0035	0.0082	-0.0021	-0.0006	-0.0028	0.0055	0.0010	0.0004	0.0006
Churn	0.0001	-0.0102	0.0002	0.0018	0.0009	-0.0014	-0.0009	0.0061	-0.0011	-0.0022