

Adapting Noise to Data: Generative Flows from 1D Processes

J. Chemseddine* G. Kornhardt* R. Duong* G. Steidl†

October 15, 2025

Abstract

We introduce a general framework for constructing generative models using one-dimensional noising processes. Beyond diffusion processes, we outline examples that demonstrate the flexibility of our approach. Motivated by this, we propose a novel framework in which the 1D processes themselves are learnable, achieved by parameterizing the noise distribution through quantile functions that adapt to the data. Our construction integrates seamlessly with standard objectives, including Flow Matching and consistency models. Learning quantile-based noise naturally captures heavy tails and compact supports when present. Numerical experiments highlight both the flexibility and the effectiveness of our method.

1 Introduction

Flow-based generative models, especially score-based diffusion [30, 32], flow matching (FM) [1, 20, 21] and consistency models like the recently introduced inductive moment matching (IMM) [39], achieve state-of-the-art results in many applications. All these methods construct a probability flow from a simple latent distribution (noise) to a complex target (data) with a neural network trained to approximate this flow from limited target samples. In diffusion models, the *score function* directs a reverse-time SDE, while in FM, the *velocity field* is learned to compute trajectories via a flow ODE. Finally, consistency models like IMM learn to predict the jumps from noise to the data while factoring in the consistency of the flow trajectories. Usually, a Gaussian is used as latent distribution which causes difficulties when learning certain multimodal and heavy-tailed targets [13, 27], see Figure 6 for a heavy-tailed example. There exist only few approaches to learn the noising process, [3] fit the forward diffusion process via a learned invertible map that is trained end-to-end, [18] use metric flow matching, i.e., a neural network to adapt the path to a underlying Riemannian metric. In a related approach, [26] learns a componentwise Gaussian noise schedule, input-conditioned so each component receives its own noise level. On the other hand [23, 38] design heavy-tailed diffusions using Student- t latent distributions, and [29] extend the framework to the family of α -stable distributions.

*Equal contribution.

†Institut für Mathematik
Technische Universität Berlin
10623 Berlin, Germany
[{chemseddine , kornhardt , duong , steidl}@math.tu-berlin.de](mailto:{chemseddine,kornhardt,duong,steidl}@math.tu-berlin.de)

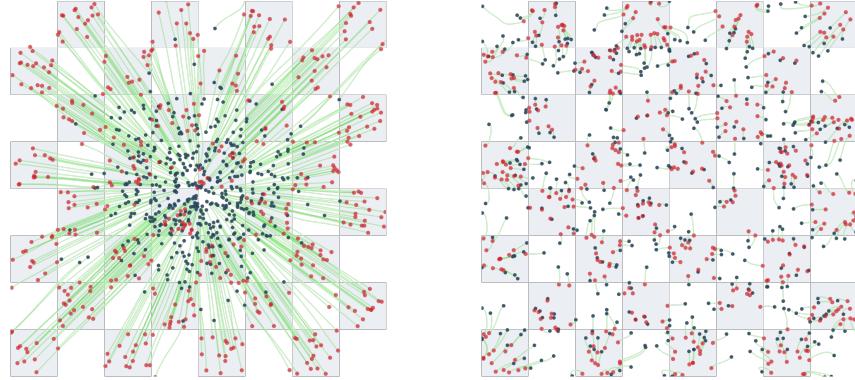


Figure 1: FM via optimal coupling with Gaussian noise (top) and our learned noise (bottom). Samples from the latent distribution are black and from generated ones are red. The left image shows their transportation paths in green. Starting from the learned latent drastically shortened the paths. (Zoom in for the paths in the bottom image).

In this paper, we present a new approach to adapt the latent distribution to the data by *learning* from its samples. The basic idea comes from the fact that all the above methods implicitly emerge as *componentwise* models. For example, denoting the target random variable by \mathbf{X}_0 and the latent by $\mathbf{X}_1 \sim \mathcal{N}(0, I_d)$, FM utilizes the process $\mathbf{X}_t = (X_t^1, \dots, X_t^d)$ with the components $X_t^i = (1-t)X_0^i + tX_1^i$ employing *one-dimensional* Gaussians $X_1^i \sim \mathcal{N}(0, 1)$. This motivated us to generally construct generative models from *1D processes and their quantile functions*.

Given any appropriate 1D process we demonstrate how to learn the componentwise neural flow by the associated conditional velocity field. We give examples besides diffusion demonstrating the flexibility of our machinery, namely the Kac process arising from the 1D damped wave equation, see [8, 14], and a process reflecting the Wasserstein gradient flow of the maximum mean discrepancy with negative distance kernel towards the uniform distribution. In contrast to diffusion, assuming a compactly supported target, these processes also have a compact support, leading to a better regularity of the corresponding velocity field. This inspired us to further adapt the process to the data and to *learn* the 1D noising process rather than choosing it manually. To this end, we exploit that 1D probability measures can be equivalently described by their quantile functions $Q^i : (0, 1) \rightarrow \mathbb{R}$ which are monotone functions, and consider quantile processes $X_t^i = (1-t)X_0^i + tQ^i(U^i)$, $i = 1, \dots, d$ with i.i.d. $U^i \sim \mathcal{U}[0, 1]$ for $t \in [0, 1]$. We learn the individual quantile functions Q_ϕ^i , $i = 1, \dots, d$ such that their componentwise concatenation $\mathbf{Q}_\phi(\mathbf{U}) := (Q_\phi^i(U^i))_{i=1}^d$ is "close" to the data.

This inspired us to minimize

$$W_2^2(\mu_0, \text{Law}(\mathbf{Q}_\phi(\mathbf{U}))), \quad \mu_0 = \text{Law}(\mathbf{X}_0).$$

with the Wasserstein distance W_2 . We combine the learning of the latent $\mathbf{Q}_\phi(\mathbf{U})$ with the learning of the velocity field via optimal coupling FM. This allows us to effectively exploit the learned noise and drastically shorten the transport paths, as illustrated in Figure 1.

The simplicity of quantile functions give us a flexible tool, which enables us to simultaneously learn the noising process and apply the FM framework. Our quantile perspective can further be extended to fit into consistency models.

Contributions. 1. We introduce a general construction method for generative neural flows by decomposing multi-dimensional flows into one-dimensional components. Ultimately, this allows us to work with *one-dimensional noising* processes in the FM framework.

2. We highlight three interesting noising processes for our framework: the Wiener process, the 1D Kac process and the 1D MMD gradient flow with negative distance kernel and uniform reference measure.

3. Based on the decomposition viewpoint, we propose to describe our 1D noising processes by their *quantile functions*. Via quantile interpolants, our framework can also be incorporated into consistency models.

4. Exploiting the simplicity of quantile functions, we propose to *learn* the quantile functions of the 1D noise simultaneously within the FM framework, aiming to fit the noise to the data. Numerical experiments demonstrate the high flexibility of our data-adapted latent noise and a shortening of transport paths.

Outline of the paper In Section 2, we start with essentials for our method, namely absolutely continuous curves in Wasserstein spaces in Subsection 2.1 followed by conditional FM in Subsection 2.2. Then, in Subsection 2.3, we introduce quite general so-called mean-reverting processes which cover standard interpolation processes in FM, but will be required in their generality when dealing with the Kac flows and quantile interpolants for consistency models.

In Section 3, we propose to decompose the noising processes into one dimensional ones. Having concrete one-dimensional processes in mind as the Kac process or the Wasserstein gradient descent flow of a special mean discrepancy functional leading to a certain Uniform process, we provide a general construction method for building accessible conditional velocity fields in FM. Finally, we consider just a scaled process, where the Wiener process and the Uniform one are special cases of. This leads us immediately to the question which one-dimensional processes or more precisely which latent noise distribution should be chosen.

Section 4, which is a main contribution of our paper, aims to learn meaningful latent distributions from the data by learning the quantile functions belonging to one-dimensional noise.

In Subsection 4.1, we recall the relation between measures on \mathbb{R} and their quantile functions, and construct quantile interpolants that may enable the integration of our framework with consistency models like inductive moment matching. Based on this, we explain in Subsection 4.2 how a data-adapted latent distribution can be learned jointly with the velocity field.

Numerical experiments demonstrating the advantages and the high potential of our approach are outlined for Section 5. On synthetic datasets, we first analyze the behavior of flow matching combined with the learned latent, and subsequently characterize its advantages and limitations on image datasets.

2 Flow Matching and Stochastic Processes

In this section, we first review absolutely continuous curves in Wasserstein spaces as basis of the subsequent FM method. Then we highlight quite general stochastic processes $(\mathbf{X}_t)_t$ “interpolating” between a noising process $(\mathbf{Y}_t)_t$ that starts in $\mathbf{Y}_0 = 0$ and ends in \mathbf{Y}_1 (our latent noise) and our target \mathbf{X}_0 . In particular, we provide the relation between the corresponding vector fields.

2.1 Absolutely Continuous Curves in Wasserstein Space

We start with a brief introduction of curves in Wasserstein spaces and basic ideas on flow matching. For more details we refer to [2] and [37]. Let $(\mathcal{P}_2(\mathbb{R}^d), W_2)$ denote the complete metric space of probability measures with finite second moments equipped with the Wasserstein distance

$$W_2^2(\mu, \nu) := \min_{\pi \in \Pi(\mu, \nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2 d\pi(x, y)$$

Here $\Pi(\mu, \nu)$ denotes the set of all probability measures on $\mathbb{R}^d \times \mathbb{R}^d$ having marginals μ and ν . The push-forward measure of $\mu \in \mathcal{P}_2(\mathbb{R}^d)$ by a measurable map $\mathcal{T} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is defined by $\mathcal{T}_\sharp \mu := \mu \circ \mathcal{T}^{-1}$. Let I be an interval in \mathbb{R} , in this paper mainly $I = [0, 1]$. A narrowly continuous curve $\mu_t : I \rightarrow \mathcal{P}_2(\mathbb{R}^d)$ is absolutely continuous, iff there exists a Borel measurable vector field $v : I \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ with $\|v_t\|_{L_2(\mathbb{R}^d, \mu_t)} \in L_2(I)$ such that (μ_t, v_t) satisfies the continuity equation

$$\partial_t \mu_t + \nabla_x \cdot (\mu_t v_t) = 0 \quad (1)$$

in the sense of distributions. If in addition $\int_I \sup_{x \in B} \|v_t(x)\| + \text{Lip}(v_t, B) dt < \infty$ for all compact $B \subset \mathbb{R}^d$, then the ODE

$$\partial_t \varphi(t, x) = v_t(\varphi(t, x)), \quad \varphi(0, x) = x, \quad (2)$$

has a solution $\varphi : I \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ and $\mu_t = \varphi(t, \cdot)_\sharp \mu_0$.

Starting in the target distribution μ_0 and ending in a simple latent distribution μ_1 , as usual in diffusion models, we can reverse the flow from the latent to the target distribution using just the opposite velocity field $-v_{1-t}$ in the ODE (2). Thus, if somebody provides us with the velocity field v_t , we can sample from a target distribution by starting in a sample from the latent one and then applying our favorite ODE solver.

2.2 Flow Matching

If we do not have a velocity field donor, we can try to approximate (learn) the velocity field by a neural network v_t^θ . Clearly, a desirable loss function would be

$$\mathcal{L}(\theta) := \mathbb{E}_{t \sim \mathcal{U}(0, 1), x \sim \mu_t} [\|v_t^\theta(x) - v_t(x)\|^2].$$

Unfortunately this loss function is not helpful, since we do not know the exact velocity field v_t nor can sample from μ_t in the empirical expectation. However, employing the law of total probabilities, as done, e.g. in [20], we see that $\mathcal{L}(\theta) = \mathcal{L}_{\text{CFM}}(\theta) + \text{const}$ with a constant not depending on θ and the Conditional Flow Matching (CFM) loss

$$\mathcal{L}_{\text{CFM}}(\theta) := \mathbb{E}_{x_0 \sim \mu_0, t \sim \mathcal{U}(0, 1), x \sim \mu_t(\cdot|x_0)} [\|v_t^\theta(x) - v_t(x|x_0)\|^2]. \quad (3)$$

The key difference is the use of the *conditional flow* $v_t(x|x_0)$ with respect to a fixed sample x_0 from our target distribution. To summarize, all you need is a *conditional* flow model with accessible velocity field $v_t(x|x_0)$ (at least along the flows trajectory), where you can easily sample from. Then you can indeed learn the velocity field v_t of the general (non-conditional) flow and finally sample from the target by the reverse ODE (2).

2.3 Stochastic Processes and Velocity Fields

Consider a continuously differentiable (noising) process $(\mathbf{Y}_t)_t$ with $\mathbf{Y}_0 \equiv 0 \in \mathbb{R}^d$ with associated velocity field $v_t = v_t^\mathbf{Y}(\cdot | 0)$ such that the pair $(\mu_t^\mathbf{Y}, v_t^\mathbf{Y})$ satisfy the continuity equation (1), where $\mu_t^\mathbf{Y}$ is the law of $(\mathbf{Y}_t)_t$ ¹. To construct a generative model we need to create a process $(\mathbf{X}_t)_t$ which can start in any sample x_0 from the target measure μ_0 . Let $\mathbf{X}_0 \sim \mu_0$. Following the lines in [8], we define the *mean-reverting* process by

$$\mathbf{X}_t := f(t) \mathbf{X}_0 + \mathbf{Y}_{g(t)}, \quad t \in [0, 1], \quad (4)$$

with smooth *scheduling functions* f, g

$$f(0) = 1, \quad f(1) = 0 \quad \text{and} \quad g(0) = 0, \quad g(1) = 1. \quad (5)$$

Then we have $\mathbf{X}_1 = \mathbf{Y}_1$, and by abuse of notation, the process \mathbf{X}_t starts in $\mathbf{X}_0 = \mathbf{X}_0$. Differentiation of (4) results in

$$\dot{\mathbf{X}}_t = \dot{f}(t) \mathbf{X}_0 + \dot{g}(t) \dot{\mathbf{Y}}_{g(t)}.$$

The conditional velocity field of \mathbf{X}_t is given by (see [37, 21])

$$\begin{aligned} v_t^\mathbf{X}(x | x_0) &= \mathbb{E}[\dot{\mathbf{X}}_t | \mathbf{X}_t = x, \mathbf{X}_0 = x_0] \\ &= \mathbb{E}[\dot{f}(t)x_0 + \dot{g}(t)\dot{\mathbf{Y}}_{g(t)} | \mathbf{Y}_{g(t)} = x - f(t)x_0] \\ &= \dot{f}(t)x_0 + \dot{g}(t)v_{g(t)}^\mathbf{Y}(x - f(t)x_0 | 0). \end{aligned} \quad (6)$$

Now, the conditional flow matching loss (3) can be minimized regarding $\mathbf{X}_0 \sim \mu_0$ and $\mathbf{X}_t \sim \mu_t$. Note that given a sample $x \sim (\mathbf{X}_t | \mathbf{X}_0 = x_0)$, we have $v_t^\mathbf{X}(x | x_0) = \dot{f}(t)x_0 + \dot{g}(t)v_{g(t)}^\mathbf{Y}(\mathbf{Y}_{g(t)} | 0)$. In general, $v^\mathbf{Y}$ might not be tractable, and only given as an conditional expectation of the time derivative $\dot{\mathbf{Y}}$. Yet, through our componentwise construction below, we will obtain easier access to it via its 1D components.

Remark 1 (Relation to FM and diffusion). *Consider the stochastic process*

$$\mathbf{X}_t^{\text{FM}} = \alpha_t \mathbf{X}_0 + \sigma_t \mathbf{X}_1, \quad \mathbf{X}_1 \sim \mathcal{N}(0, I_d). \quad (7)$$

Choosing $f(t) := \alpha_t$, $g(t) := \sigma_t^2$ and the standard Brownian motion $\mathbf{Y}_t = \mathbf{W}_t$, it holds the equality in distribution

$$\mathbf{X}_t^{\text{FM}} \stackrel{d}{=} f(t)\mathbf{X}_0 + \mathbf{W}_{g(t)} = \mathbf{X}_t.$$

Then $f(t) := 1 - t$, $g(t) := t^2$ yields (independent) FM [20], and $f(t) := \exp\left(-\frac{h(t)}{2}\right)$, $g(t) := 1 - \exp(-h(t))$, where $h(t) := \int_0^t \beta_{\min} + s(\beta_{\max} - \beta_{\min}) ds$ with, e.g., $\beta_{\min} = 0.1$, $\beta_{\max} = 20$, corresponds to processes used in score-based generative models [33], see Appendix B.

Remark 2 (Optimal Coupling). *Instead of considering (possibly independent) random variables $\mathbf{X}_0, \mathbf{X}_1$ and their induced processes such as in (7), we can also introduce a coupling between them. Let $X_0 \sim \mu_0, X_1 \sim \mu_1$ and consider the optimal coupling $\pi \in \Pi_o(\mu_0, \mu_1)$. Then define the induced curve $(e_t)_{\sharp} \pi$, where $e_t(x, y) := (1 - t)x + ty$. This curve is a geodesic between μ_0 and μ_1 in the*

¹Existence of the velocity is given under weak assumptions by [37] Theorem 6.3.

Wasserstein geometry, for more details see [37]. Using the optimal coupling $\pi \in \Pi_o(\mu_0, \mu_1)$ yields an Optimal Transport FM objective

$$\mathcal{L}_{\text{OT-CFM}}(\theta) = \mathbb{E}_{t \sim \mathcal{U}(0,1), (x,y) \sim \pi} \left[\|v_\theta((1-t)x + ty, t) - (y-x)\|_2^2 \right].$$

The velocity field minimizing this objective then satisfies the continuity equation together with $e_t \sharp \pi$. In contrast to using the independent coupling, this can lead to reduced variance in training and both shorter and straighter paths, see [35, 25].

Motivated by the fact that a multi-dimensional Wiener process $\mathbf{W}_t \in \mathbb{R}^d$ consists of *independent* (and identically distributed) 1D components $\mathbf{W}_t = (W_t^1, \dots, W_t^d)$, we propose to construct a d -dimensional flow \mathbf{Y}_t componentwise, based on independent one-dimensional processes Y_t^i .

3 From One-Dimensional to Multi-Dimensional Flows

Restricting ourselves to processes \mathbf{Y}_t that decompose into one-dimensional components allows us to propose a general construction method for accessible *conditional* flows in FM. Let Y_t^1, \dots, Y_t^d be a family of independent one-dimensional stochastic processes with time dependent laws $\mu_t^i \in \mathcal{P}_2(\mathbb{R})$. For each $i = 1, \dots, d$, let $v_t^i: \mathbb{R} \rightarrow \mathbb{R}$ be the associated velocity field such that the pair (μ_t^i, v_t^i) satisfies the one-dimensional continuity equation (1). Define the product measure $\mu_t \in \mathcal{P}_2(\mathbb{R}^d)$ by

$$\mu_t(x) = \prod_{i=1}^d \mu_t^i(x^i), \quad x = (x^1, \dots, x^d) \in \mathbb{R}^d. \quad (8)$$

For the d -dimensional process $\mathbf{Y}_t := (Y_t^1, \dots, Y_t^d)$, independence implies that its law is exactly μ_t . Moreover, by the following proposition, the corresponding d -dimensional velocity field is given componentwise, see [8].

Proposition 3. *Let μ_t be given by (8), where the μ_t^i are absolutely continuous curves in \mathbb{R} with velocity fields v_t^i . Then μ_t satisfies a multi-dimensional continuity equation (1) with a velocity field which decomposes into the univariate velocities*

$$v_t(x) := (v_t^1(x^1), \dots, v_t^d(x^d)).$$

Therefore, as long as we have access to the velocity field associated to our one-dimensional processes, we can **construct accessible conditional flows for FM**:

1. *One-dimensional noise:* Start with a one-dimensional process and an associated absolutely continuous curve μ_t with $\mu_0 = \delta_0$, $0 \in \mathbb{R}$, where you can compute the velocity field v_t in the 1D continuity equation

$$\partial_t \mu_t + \partial_x (\mu_t v_t) = 0, \quad \mu_0 = \delta_0. \quad (9)$$

2. *Multi-dimensional noise:* Set up a multi-dimensional conditional flow model starting in $\mu_0 = \delta_0$, $0 \in \mathbb{R}^d$ with possibly different, but independent 1D processes as described in Section 3.
3. *Incorporating the data:* Construct a multi-dimensional conditional flow model starting in $\mu_0 = \delta_{x_0}$ for any data point $x_0 \sim \mu_0$ by mean-reversion as shown in Section 2.3.

To outline the use of this recipe, we explore three interesting 1D (noising) processes Y_t in connection with their respective PDEs, for which our approach via reduction to one dimension is nicely applicable, namely the

- Wiener process W_t and diffusion equation,
- Kac process K_t and damped wave equation,
- Uniform process U_t and the gradient flow of the maximum mean functional $\mathcal{F}_\nu := \text{MMD}_K(\cdot, \nu)$ with negative distance kernel $K(x, y) = -|x - y|$ and $\nu = \mathcal{U}(-b, b)$.

Paths of the processes are depicted in Figure 2.

In each case, the absolutely continuous curve starting in δ_0 and the corresponding velocity field can be calculated analytically. Note that in contrast to the Wiener process W_t usually seen in diffusion and flow matching models, the latter two processes K_t, U_t do not enjoy a trivial analogue in multiple dimensions: in case of K_t the corresponding PDE (damped wave equation) is no longer mass-conserving in dimension $d \geq 3$, see [34]; in case of U_t the mere existence of the MMD gradient flow in multiple dimensions is unclear by the lack of convexity of the MMD, see [15]. Our general construction method makes these 1D processes accessible for generative modeling in arbitrary dimensions.

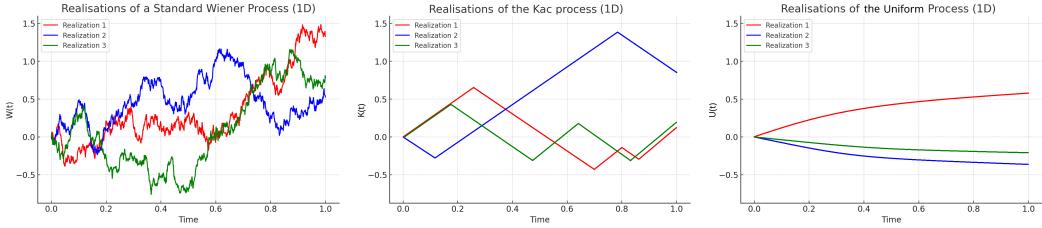


Figure 2: Three realisations of a standard Wiener process (left), the Kac process (middle), and the Uniform process (right), simulated until time $t = 1$.

These three examples highlight the adaptability of the framework and the different choices that can be made. For simplicity our approach for learning the latent distribution will only consider processes defined as a deterministic scaling of a *fixed* random variable. Therefore, in the last part of this section we also consider this case and how it fits into our framework.

3.1 Wiener Process and Diffusion Equation

First, consider the standard Wiener process (Brownian motion) $(W_t)_t$ starting in 0 whose probability density flow p_t is given by the solution of the diffusion equation

$$\partial_t p_t = \nabla \cdot (p_t \frac{1}{2} \nabla \log p_t) = \frac{1}{2} \Delta p_t, \quad t \in (0, 1], \quad \lim_{t \downarrow 0} p_t = \delta_0, \quad (10)$$

where the limit for $t \downarrow 0$ is taken in the sense of distributions. The solution is analytically known to be

$$p_t(x) = (2\pi t)^{-\frac{d}{2}} e^{-\frac{\|x\|^2}{2t}}.$$

Thus, the latent distribution is just the Gaussian $p_1 = \mathcal{N}(0, I_d)$. The velocity field in (10) reads as

$$v_t(x) = -\frac{1}{2} \nabla \log p_t = \frac{x}{2t}. \quad (11)$$

However, its L_2 -norm fulfills $\|v_t\|_{L_2(\mathbb{R}, p_t)}^2 = \frac{d}{4t}$, and is therefore not integrable over time, i.e. $\|v_t\|_{L_2(\mathbb{R}, p_t)} \notin L_2(0, 1)$. In practice, instability issues caused by this explosion at times close to the target need to be avoided by e.g. time truncations, see e.g. [19]. For a heuristic analysis also including drift-diffusion flows, we refer to [24]. Note that in the case of diffusion, there is no significant distinction between the uni- and multivariate setting. Figure 3.2 shows the generation of samples from a weighted Gaussian Mixture Model (GMM) using Flow Matching and the Wiener process as our noising process. As described in Section 2.3 we define the mean reverting process and use schedules $f(t) = 1 - t$ and $g(t) = t^2$.



Figure 3: A generated trajectory from a Flow Matching model trained using the conditional density and velocity given by the Wiener process.

3.2 Kac Process and Damped Wave Equation

The Kac process was recently used in generative modeling by some of the authors in [8]. It originates from a discrete random walk, which starts in 0 and moves with velocity parameter $c > 0$ in one direction until it reverses its direction with probability $a\Delta_t$, $a > 0$, see [17]. A continuous-time analogue is given by the Kac process which is defined using the homogeneous *Poisson point process* N_t with rate a , i.e. i) $N_0 = 0$; ii) the increments of N_t are independent, iii) $N_t - N_s \sim \text{Poi}(a(t-s))$ for all $0 \leq s < t$. Now the *Kac process starting in 0* is given by

$$K_t := B_{\frac{1}{2}} c \int_0^t (-1)^{N_s} ds, \quad (12)$$

where $B_{\frac{1}{2}} \sim \text{Ber}(\frac{1}{2})$ is a Bernoulli random variable² taking the values ± 1 . Note that in contrast to diffusion processes, the Kac process K_t persistently maintains its linear motion between changes of directions (jumps of N_t), see Figure 2.

By the following proposition, the Kac process is related to the damped wave equation, also known as telegrapher's equation, and its probability distribution admits a computable vector field such that the continuity equation is fulfilled. For a proof we refer to our paper [8].

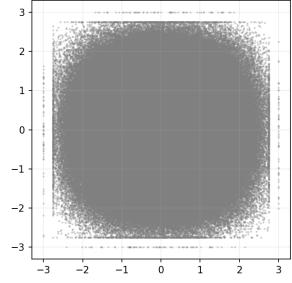


Figure 4: Samples (1M) from the distribution of the two-dimensional Kac process K_1 for $(a, c) = (9, 3)$. We can clearly observe the atomic component of its distribution derived in (13).

²More precisely, $B_{\frac{1}{2}}$ is *two-point* distributed with values $\{-1, 1\}$.

Proposition 4. *The probability distribution flow of $(K_t)_t$ admits a singular and absolutely continuous part via*

$$\mu_t(x) = \frac{1}{2}e^{-at}(\delta_0(x+ct) + \delta_0(x-ct)) + \tilde{p}_t(x), \quad (13)$$

with the absolutely continuous part

$$\tilde{p}_t(x) := \frac{1}{2}e^{-at}\left(\beta ct\frac{I'_0(\beta r_t(x))}{r_t(x)} + \beta I_0(\beta r_t(x))\right)1_{[-ct,ct]}(x), \quad r_t(x) := \sqrt{c^2t^2 - x^2},$$

where $\beta := \frac{a}{c}$, and I_0 denotes the 0-th modified Bessel function of first kind. The distribution (13) is the generalized solution of the damped wave equation

$$\begin{aligned} \partial_{tt}u(t,x) + 2a\partial_tu(t,x) &= c^2\partial_{xx}u(t,x), \\ u(0,x) &= \delta_0(x), \quad \partial_tu(0,x) = 0. \end{aligned} \quad (14)$$

Further (μ_t, v_t) solves the continuity equation (9) where the velocity field is analytically given by

$$v_t(x) := \begin{cases} \frac{x}{t + \frac{r_t(x)}{c} \frac{I_0(\beta r_t(x))}{I'_0(\beta r_t(x))}} & \text{if } x \in (-ct, ct), \\ c & \text{if } x = ct, \\ -c & \text{if } x = -ct, \\ \text{arbitrary} & \text{otherwise.} \end{cases}$$

The Kac velocity field admits the boundedness $\|v_t\|_{L_2(\mathbb{R}, \mu_t)} \leq c$, and hence, $\|v_t\|_{L_2(\mathbb{R}, \mu_t)} \in L_2(0, 1)$.

Interestingly, the damped wave equation (14) is closely related to the diffusion equation via Kac' insertion method. In particular, diffusion can be seen as "an infinitely a -damped wave with infinite propagation speed c ". Note that the diffusion-related concept of particles traveling with infinite speed violates Einstein's laws of relativity and has therefore found resistance in the physics community [4, 6, 36, 34]. Figure 3.2 shows the generation of samples from a weighted Gaussian Mixture Model (GMM) using Flow Matching and the Kac process as our noising process. As described in Section 2.3 we define the mean reverting process and use schedules $f(t) = 1 - t$ and $g(t) = t^2$.



A generated trajectory from a Flow Matching model trained using the conditional density and velocity given by the Kac process with $(a, c) = (9, 3)$.

3.3 Uniform Process and Gradient Flow of MMD Functional

Wasserstein gradient flows are special absolutely continuous measure flows, whose velocity fields are negative Wasserstein (sub-)gradients of functionals \mathcal{F}_ν on $\mathcal{P}_2(\mathbb{R}^d)$ with the unique minimizer, i.e., $v_t \in -\partial\mathcal{F}_\nu(\mu_t)$. The gradient descent flow should reach this minimizer as $t \rightarrow \infty$.

In this context, the MMD functional with the non-smooth negative distance kernel $K(x, y) = -|x - y|$ given by

$$\mathcal{F}_\nu(\mu) = \text{MMD}_K^2(\mu, \nu) := -\frac{1}{2} \int_{\mathbb{R}^2} |x - y| d(\mu(x) - \nu(x)) d(\mu(y) - \nu(y)), \quad (15)$$

stands out for its flexible flow behavior between distributions of different support [15]. We have the following proposition which proof can be found in Appendix A. Note that the proof is already based on so-called quantile function, which we consider in the next section.

Proposition 5. *The Wasserstein gradient flow μ_t of the MMD functional (15) starting in $\mu_0 = \delta_0$ towards the uniform distribution $\nu = \mathcal{U}[-b, b]$ with fixed $b > 0$ reads as*

$$\mu_t = (1 - \exp(-\frac{t}{b})) \mathcal{U}[-b, b], \quad t > 0, \quad (16)$$

with corresponding velocity field

$$v_t(x) = \frac{x}{b(\exp(\frac{t}{b}) - 1)}, \quad x \in \text{supp}(\mu_t). \quad (17)$$

It holds $\|v_t\|_{L_2(\mathbb{R}, \mu_t)}^2 = \frac{2b}{3} \exp(-\frac{2t}{b})$, and hence, $\|v_t\|_{L_2(\mathbb{R}, \mu_t)} \in L_2(0, 1)$. A corresponding (stochastic) process $(U_t)_t$ is given by $U_t := b(1 - \exp(-\frac{t}{b})) U$, where $U \sim \mathcal{U}[-1, 1]$, such that $U_t \sim \mu_t$.

Figure 3.2 shows the generation of samples from a weighted Gaussian Mixture Model (GMM) using Flow Matching and the process induced by the MMD gradient flow as our noising process. As described in Section 2.3 we define the mean reverting process and use schedules $f(t) = 1 - t$ and $g(t) = t$.



Figure 5: A generated trajectory from a Flow Matching model trained using the conditional density and velocity given by the MMD gradient flow.

3.4 Scaled Latent Distributions

Finally, we consider a simple class of processes obtained by a deterministic scaling of a latent random variable. In particular, we will see that the above Wiener process and the Uniform process are of this form, while the Kac process is not. Let Z be a random variable with law $\rho_Z \in \mathcal{P}_2(\mathbb{R})$, and let $g: [0, 1] \rightarrow [0, \infty)$ be continuously differentiable with $g(0) = 0$ and $g(1) = 1$. We consider

$$Y_t := g(t) Z, \quad t \in [0, 1],$$

with $Y_t \sim \mu_t$. Supposing that μ_t has density ρ_t , we get

$$\rho_t(x) = g(t)^{-d} \rho_Z\left(\frac{x}{g(t)}\right), \quad t > 0, \quad \text{and} \quad \lim_{t \downarrow 0} \mu_t = \delta_0.$$

Then straightforward computation yields that μ_t together with the velocity field

$$v_t(x) = \frac{g'(t)}{g(t)} x, \quad x \in \text{supp}(\mu_t)$$

with the convention $v_t(0) = 0$ and arbitrary outside $\text{supp}(\mu_t)$, solves the continuity equation (9). Further, it holds

$$\int_0^1 \|v_t\|_{L_2(\mathbb{R}, \mu_t)}^2 dt = \mathbb{E}[\|Z\|^2] \int_0^1 (g'(t))^2 dt < \infty \quad \text{whenever } g' \in L_2(0, 1). \quad (18)$$

The Wiener process fits into this framework with $g(t) = \sqrt{t}$ and $Z \sim \mathcal{N}(0, 1)$, which recovers the exploding vector field $v_t(x) = \frac{1}{2t}x$ in (11). Also the Uniform process appears as a special case of the scaling process. In contrast, the Kac process does *not* belong to this class, as it is not generated by a deterministic scaling map but by persistent velocity switching, cf. (12).

In the rest of this paper, we adopt a signal-decay schedule $f(t) = 1 - t$ and the linear latent growth $g(t) = t$, so that $Y_t = tZ$ and $v_t(x) = \frac{x}{t}$, where the $L_2(\mu_t)$ -energy remains constant in t by (18). Note this corresponds to the standard linear interpolation employed in Flow Matching.

Inspired by [23] in Figure 6 we highlight how different choices of latent Z can heavily affect the sampling performance. For latent distributions without (heavy) tails, the FM model fails to capture the tails of the data distribution. This motivates us to learn a data-adapted process $Y_t := tZ$ by learning the terminal distribution Z .

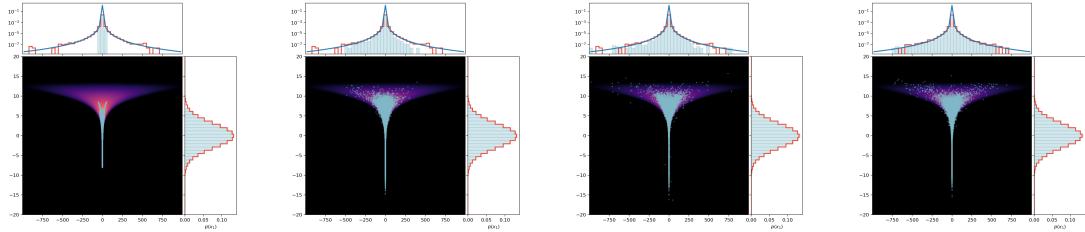


Figure 6: Sampling of Neal’s funnel with different latent distributions⁴. From left to right with uniform ($[-1, 1]$), standard Gaussian, Student-T (with parameters $(20, 4)$ inspired by the choice in [23]) and our learned distribution. The last two heavy-tailed noises perform significantly better.

4 Adapting Noise to Data

Motivated by the observed influence of the noising process on sample quality (Fig. 6), we propose to learn the noising process itself by learning a componentwise terminal distribution. We first revisit the connection between one-dimensional distributions and their quantile functions, then introduce quantile processes and quantile interpolants. Finally, we describe how the corresponding quantile functions can be learned in practice.

⁴Note that we used the independent coupling for training of these models and we pretrained our learned latent (instead of training jointly). We also used z-score normalization.

4.1 Quantile Processes and Interpolants

The restriction to componentwise noising processes \mathbf{Y}_t in (4)⁵ allows us to use the quantile functions of the 1D components. Recall that the *cumulative distribution function* (CDF) R_μ of $\mu \in \mathcal{P}_2(\mathbb{R})$ and its *quantile function* Q_μ are given by

$$R_\mu(x) := \mu((-\infty, x]), \quad x \in \mathbb{R} \quad \text{and} \quad Q_\mu(u) := \min\{x \in \mathbb{R} : R_\mu(x) \geq u\}, \quad u \in (0, 1). \quad (19)$$

In Figure 7, we exemplify the CDF and quantile function of a standard Gaussian. The quantile functions form a closed, convex cone $\mathcal{C} := \{f \in L_2(0, 1) : f \text{ increasing a.e.}\}$ in $L_2(0, 1)$. The mapping $\mu \mapsto Q_\mu$ is an isometric embedding of $(\mathcal{P}_2(\mathbb{R}), W_2)$ into $(L_2(0, 1), \|\cdot\|_{L_2})$, meaning that

$$W_2^2(\mu, \nu) = \int_0^1 |Q_\mu(s) - Q_\nu(s)|^2 ds$$

and $\mu = Q_{\mu,\#}\mathcal{L}_{(0,1)}$. Let $U \sim \mathcal{U}[0, 1]$ be uniformly distributed on $[0, 1]$. Now, any probability measure flow μ_t can be described by their respective quantile flow $Q_t := Q_{\mu_t}$, such that $\mu_t = Q_{t,\#}\mathcal{L}_{(0,1)}$ and $Q_t \circ U$ is a stochastic process with marginals μ_t .

Quantile Processes. We can therefore model any *multi-dimensional* noising process, that decomposes into its components, via quantile functions. Namely let X_0 be any component \mathbf{X}_0^i of $\mathbf{X}_0 \sim \mu_0$, and $f, g : [0, 1] \rightarrow \mathbb{R}$ smooth schedules fulfilling (5). We assume that we are given a flow $(Q_t)_t$ of quantile functions $Q_t : (0, 1) \rightarrow \mathbb{R}$, $t \in [0, 1]$, which fulfill $Q_0 \equiv 0$ and are invertible on their respective image with the inverse given by the CDF $R_t : Q_t(0, 1) \rightarrow \mathbb{R}$. We introduce the *quantile process*

$$Z_t = f(t)X_0 + Q_{g(t)}(U), \quad U \sim \mathcal{U}(0, 1), t \in [0, 1]. \quad (20)$$

The quantile process coincides (in distribution) with the components of the mean-reverting process (4), where the noising term is represented as $\mathbf{Y}_{g(t)}^i \stackrel{d}{=} Q_{\text{Law}(\mathbf{Y}_{g(t)}^i)}(U)$. In particular, the components of the process (7) are obtained via (20) using the quantile distribution Q_t of a standard Brownian motion W_t and $f(t) := \alpha_t$, $g(t) := \sigma_t^2$.

Quantile Interpolants. Let us briefly mention how our setting fits into the framework of consistency models. To this end, we define the *quantile interpolants*

$$I_{s,t}(x, y) = f(s)x + Q_{g(s)}(R_{g(t)}(y - f(t)x)), \quad s, t \in [0, 1] \quad (21)$$

which generalize the interpolants used in Denoising Diffusion Implicit Models (DDIM), see Remark 10.

⁵Besides componentwise 1D processes we may also use triangular decompositions, not addressed in this paper.

Proposition 6. For all $x, y \in \mathbb{R}$ and all $s, r, t \in [0, 1]$, it holds $I_{0,t}(x, y) = x$, $I_{t,t}(x, y) = y$, and

$$I_{s,r}(x, I_{r,t}(x, y)) = I_{s,t}(x, y).$$

Furthermore, inserting the quantile process (20) yields $I_{s,t}(Z_0, Z_t) = Z_s$.

The proof is given Appendix C. Proposition 6 allows us to also apply the concept of consistency models to our quantile process (20). The shared idea of these models is to predict the jumps from the process Z_t to the target X_0 , while factoring in the *consistency* of the trajectory of Z_t via Z_s , $0 < s < t$. In FM, this consistency of the flow is usually neglected as only single points on the FM paths are sampled. Also, consistency models as one-step or multistep samplers usually are in no need of velocity fields. In the Appendix C, we demonstrate by means of the recently proposed *inductive moment matching* (IMM) [39], that our formulation via quantile interpolants fits seamlessly into the consistency framework.

4.2 Learning Quantile Processes

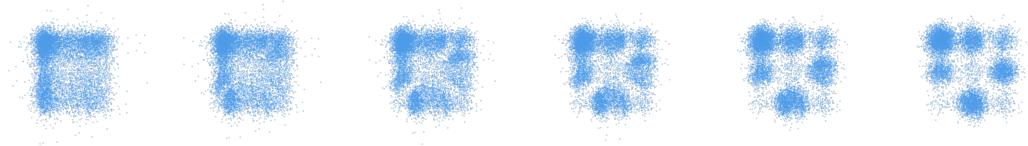


Figure 8: A generated trajectory from the learned quantile latent (left) to the unevenly weighted Gaussian mixture target (right). The adapted latent is already close to the target distribution.

We have already observed that the choice of latent distribution has a pronounced effect on sampling performance; see Figure 1 for the checkerboard distribution and Figure 6 for a heavy-tailed example. Adopting the quantile-process view from Section 4.1, we parameterize the latent distribution via coordinate-wise quantile maps

$$\mathbf{Q}_\phi := (Q_\phi^1, \dots, Q_\phi^d),$$

which yields learnable noise that, by construction, satisfies (i) data-independence $\mathbf{Q}_\phi(\mathbf{U}) \perp\!\!\!\perp \mathbf{X}_0$ and (ii) independence of components (by our 1D construction). Consequently, the induced latent belongs to the product class

$$S := \{\nu \in \mathcal{P}_2(\mathbb{R}^d) : \nu = \bigotimes_{i=1}^d \nu^i\}.$$

Adapt the Latent to Data. We learn the quantile maps \mathbf{Q}_ϕ , which by definition lie in S , by bringing the induced latent distribution

$$\nu_\phi := (\mathbf{Q}_\phi)_\# \mathcal{U}([0, 1]^d)$$

close to the data distribution μ_0 in the Wasserstein sense,

$$\mathcal{L}_{\text{AN}}(\phi) = W_2^2(\mu_0, \nu_\phi). \quad (22)$$

This choice is natural under our objective: with $(x, y) \sim \pi_\phi \in \Pi_o(\mu_0, \nu_\phi)$ an optimal coupling, one has

$$\mathbb{E}_{(x,y) \sim \pi_\phi} [\|y - x\|_2^2] = W_2^2(\mu_0, \nu_\phi),$$

so minimizing (22) shortens the average segment $\|y - x\|_2$ that the model must predict along the straight line $(1 - t)x + ty$, thereby improving conditioning. Because we restrict \mathbf{Q}_ϕ to the set S , the minimizer of (22) does not, in general, reproduce μ_0 exactly. Moreover, even the product of the target marginals need not be optimal in W_2 among product measures, as the following example shows.

Example 7. *For the measure*

$$\mu = \frac{1}{2}\delta_{(1,1)} + \frac{1}{2}\delta_{(-1,-1)} \in \mathcal{P}_2(\mathbb{R}^2), \quad \mu_{\text{marg}} = \left(\frac{1}{2}\delta_{-1} + \frac{1}{2}\delta_1\right) \otimes \left(\frac{1}{2}\delta_{-1} + \frac{1}{2}\delta_1\right),$$

one has $W_2^2(\mu, \mu_{\text{marg}}) = 2$, whereas for

$$\nu_\alpha = \left(\frac{1}{2}\delta_{-\alpha} + \frac{1}{2}\delta_\alpha\right) \otimes \left(\frac{1}{2}\delta_{-\alpha} + \frac{1}{2}\delta_\alpha\right)$$

it holds $W_2^2(\mu, \nu_\alpha) = 2(1 - \alpha + \alpha^2) = 1.5$ for $\alpha = 0.5$. Thus the W_2 -closest independent latent may contract or expand the marginals to partially account for correlations it cannot represent.

Quantile Path. Writing $\mathbf{U} \sim \mathcal{U}([0, 1]^d)$ the induced componentwise linear paths is given as

$$X_t^i = (1 - t)X_0^i + tQ_\phi^i(U^i), \quad i = 1, \dots, d, t \in [0, 1],$$

so that

$$\mathbf{X}_t = (1 - t)\mathbf{X}_0 + t\mathbf{Q}_\phi(\mathbf{U}).$$

Crucially, the independence constraint restricts ν_ϕ to per-coordinate adaptation and prevents encoding *cross-dimensional* correlations. The latter are introduced via the optimal transport coupling (x, y) and modeled by the velocity field through the target $(y - x)$. This separation lets the latent remain simple and computationally efficient while delegating dependencies to the flow.

Joint Optimization. While the quantiles can be trained independently of the vector field, we train \mathbf{Q}_ϕ jointly with v_θ to provide a consistent signal, minimizing

$$\mathcal{L}(\theta, \phi) = \mathcal{L}_{\text{OT-CFM}}(\theta, \phi) + \lambda \mathcal{L}_{\text{AN}}(\phi), \quad \lambda > 0,$$

with

$$\mathcal{L}_{\text{OT-CFM}}(\theta, \phi) = \mathbb{E}_{t \sim \mathcal{U}(0, 1), (x, y) \sim \pi_\phi} \left[\|v_\theta((1 - t)x + ty, t) - (y - x)\|_2^2 \right],$$

where $\pi_\phi \in \Pi_o(\mu_0, \nu_\phi)$, see Remark 2. In practice we optimize empirical expectations over mini-batches; we compute a mini-batch OT coupling once per batch and use it within both loss terms, see Appendix D.4.

Algorithm 1 Joint learning of 1D quantiles and FM velocity

Require: Dataset \mathcal{D} , batch size B , weight λ_Q , iterations K

Require: Quantile model \mathbf{Q}_ϕ , velocity model v_θ

- 1: **for** $k = 1$ to K **do**
 - 2: Sample $\{\mathbf{x}_i\}_{i=1}^B \sim \mathcal{D}$, $\{\mathbf{u}_i\}_{i=1}^B \sim \mathcal{U}([0, 1]^d)$, $\{t_i\}_{i=1}^B \sim \mathcal{U}(0, 1)$
 - 3: $C_{ij} \leftarrow \|\mathbf{x}_i - \mathbf{Q}_\phi(\mathbf{u}_j)\|_2^2$
 - 4: $T \leftarrow \arg \min_T \sum_{i=1}^B C_{i,T(i)}$
 - 5: $\mathcal{L}_{\text{AN}} \leftarrow \frac{1}{B} \sum_{i=1}^B \|\mathbf{x}_i - \mathbf{Q}_\phi(u_{T(i)})\|_2^2$
 - 6: $\mathbf{z}_i \leftarrow (1 - t_i)\mathbf{x}_i + t_i \mathbf{Q}_\phi(u_{T(i)})$
 - 7: $\mathcal{L}_{\text{OT-CFM}} \leftarrow \frac{1}{B} \sum_{i=1}^B \|v_\theta(\mathbf{z}_i, t_i) - \mathbf{Q}_\phi(u_{T(i)}) + \mathbf{x}_i\|_2^2$
 - 8: $\mathcal{L} \leftarrow \mathcal{L}_{\text{OT-CFM}} + \lambda \mathcal{L}_{\text{AN}}$
 - 9: Update (θ, ϕ) by a gradient step on L
 - 10: **end for**
 - 11: **return** (θ, ϕ)
-

5 Experiments

To provide intuition and validate our proposed method, we conduct experiments on both synthetic and image datasets. For each component, we model the quantile with a Rational Quadratic Spline (RQS) [12, 10] and add a learnable scale and bias. This keeps monotonicity, is parameter-efficient, and gives analytic derivatives. See Appendix D.2 for details. The code is available on GitHub at <https://github.com/TUB-Angewandte-Mathematik/Adapting-Noise>.

5.1 Synthetic Datasets

We begin by qualitatively analyzing our algorithm on several synthetic 2D distributions (see Appendix D.1), each designed to highlight a specific aspect of our approach. We provide intuition about the learned latent distribution and demonstrate that it is closer to the data in the Wasserstein sense, yields shorter transport paths, and successfully captures the tail behavior.

Gaussian Mixture Model (GMM). We first consider a 2D GMM with nine unevenly weighted modes, as visualized in Figure 8. Due to the independence assumption inherent in our factorized quantile function, the learned latent cannot perfectly replicate the target’s joint distribution and is **not the product of the correct marginals** (see also Example 7). Instead, it approximates a distribution where the components cannot further independently improve the transport cost to the target.

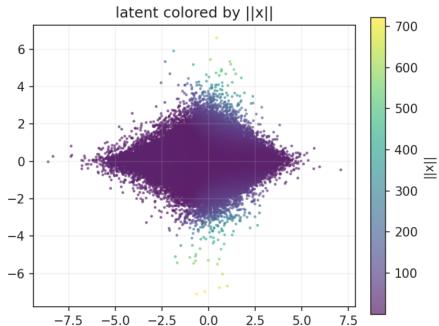


Figure 9: Samples (1M) from our learned latent of the funnel distribution. Color shows endpoint norm.

Funnel Distribution. The funnel distribution, shown in Figure 6, presents a challenge due to its heavy-tailed, conditional structure. Several methods for handling heavy-tailed datasets have already been introduced in the context of diffusion models [23, 28, 29]. For example, we compare our funnel example to [23], where the authors hand-select the parameters of a Student- t distribution in each dimension.

To visualize the effects more clearly, we use a capacity-constrained network with three layers of width 64 and no positional embeddings due to the large scale of the data. This experiment (Figure 6) highlights the importance of matching the latent tail behavior to that of the target distribution, showing that a compact latent performs worst, followed by the Gaussian latent. At the same time, we observe that our learned latent successfully **adapts to the target’s heavy tails**, see Figure 9. This enables the flow matching model to generate high fidelity samples across the distribution. Note that due to the high variance signal when training on the funnel distribution, we pre-train our quantile.

Checkerboard Distribution. In contrast to the funnel, the checkerboard distribution in Figure 1 features a compact support. Here, we demonstrate the synergy between our learned latent and an Optimal Transport (OT) coupling. Our method learns a latent that approximates a uniform distribution over the target’s support. When this adapted latent is combined with an OT coupling for FM, the resulting **transport paths are substantially shorter** than those originating from a standard Gaussian as shown in Figure 14. Further, the vector field training converges much faster, see Figure 15. This result underscores our central claim: combining a data-dependent latent with a data-dependent coupling has the potential to significantly improve model performance.

5.2 Image Datasets

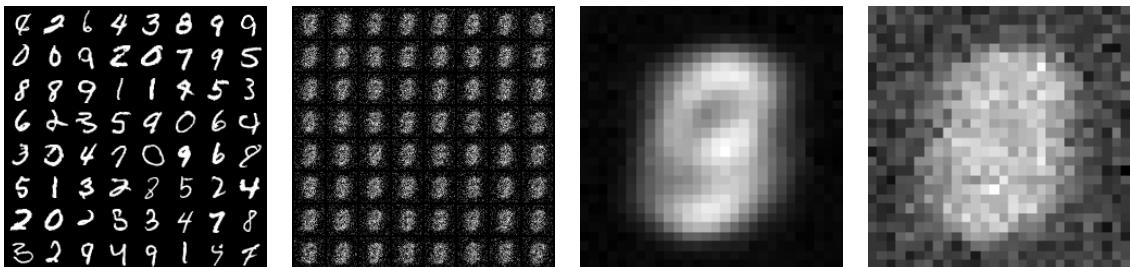


Figure 10: MNIST Dataset: From Left to Right: Generated samples, samples from the learned latent and mean and standard deviation of the learned latent.

Next we analyze our method on standard image generation benchmarks. Our quantile is extremely lightweight compared to the UNet architecture used for the flow model. We reuse the minibatch OT coupling for the latent and freeze the quantile function after a $25k$ training epochs. This strategy introduces only minimal computational overhead compared to the standard Gaussian baseline with minibatch OT coupling. Specifically, we observe an overhead of approximately 20% in runtime during joint quantile training, and about 12% after freezing the quantile parameters, measured on an NVIDIA GeForce RTX 4090.

In high-dimensional settings and given fixed batch sizes, the signal for the quantile function can be noisy, potentially leading to degenerate solutions. To mitigate this, we add a regularization term to the loss that penalizes the expected negative log-determinant of the Jacobian of the quantile. Access to analytic derivatives makes this computation efficient, for more details see D.3.

MNIST. The MNIST dataset exhibits strong marginal structure: pixels near the center are frequently active (non-zero), whereas pixels at the borders are almost always zero. Our learned quantile function successfully captures these global marginal statistics. As illustrated in Figure 10, the latent distribution learns to concentrate its mass in regions corresponding to active pixels. We also plot the mean and standard deviation in the third and fourth images of Figure 10.

In Figure 11, we compare the learned and empirical quantiles on the MNIST dataset at different pixel locations (x, y) . Where the pixel is essentially black, the learned quantile concentrates around that value, whereas in the center regions, where uncertainty is higher, the quantiles remain spread around zero (gray), accurately reflecting the data variability.

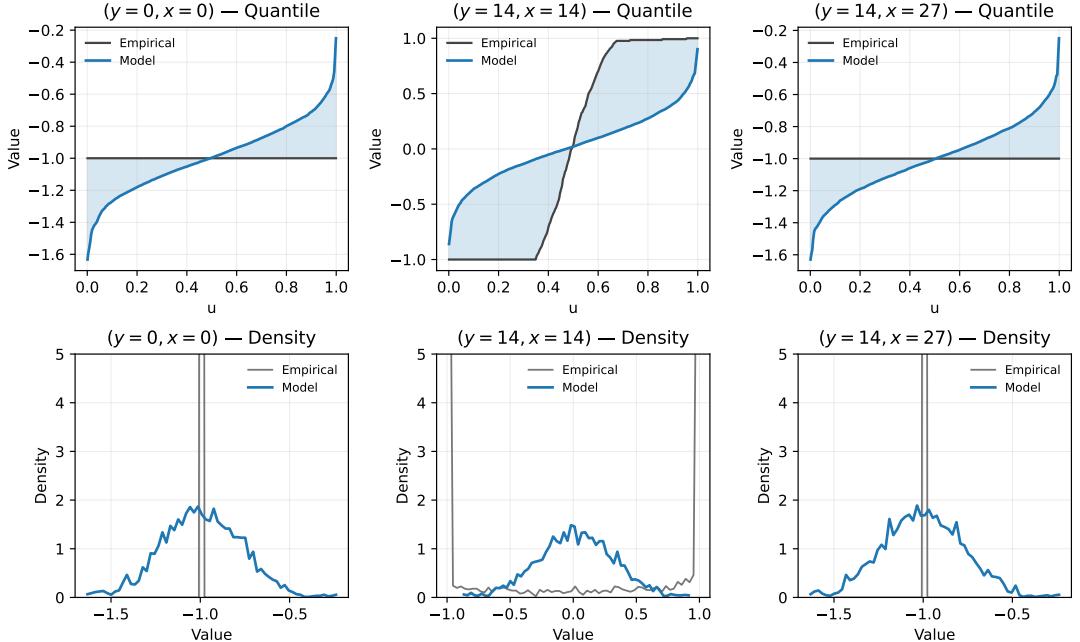


Figure 11: Comparison of the empirical and learned probability density functions and their quantile functions at different pixel locations (y, x) , averaged over images from the MNIST dataset. The blue area illustrates the difference between the quantiles, corresponding to the one-dimensional Wasserstein distance; see Eq. 4.1.

While the independence assumption prevents the model from capturing specific spatial correlations (e.g., the shape of a digit), the learned latent clearly adapts to the underlying data distribution—removing noise where unnecessary and retaining it where needed.

In Figure 12, we compare the performance under different network capacity constraints by evaluating our learned latent against a Gaussian latent. Both latents are trained using mini-batch

optimal transport. As already observed in Fig. 10, the transport paths are significantly shorter since the learned latent successfully minimizes the Wasserstein distance and removes redundant information. This enables the network to use the available parameters more efficiently and achieve better results with the same parameter count.

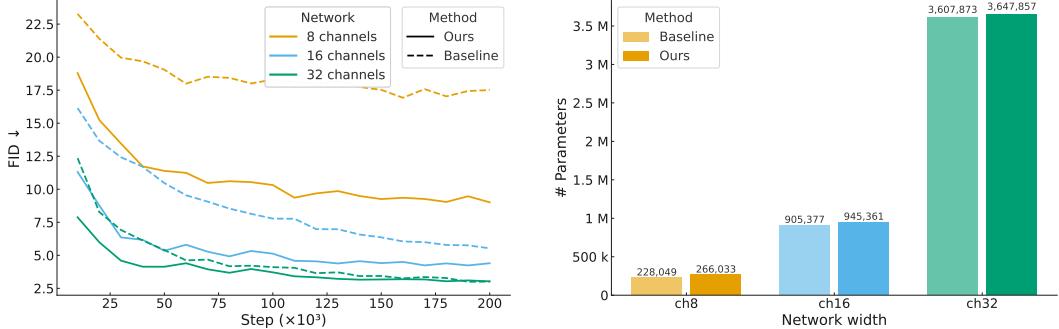


Figure 12: Ablation study over capacity of the U-Net for sampling from the MNIST dataset. In the bar chart, we see that doubling the number of channels roughly quadruples the number of parameters. The FID curves show that our method achieves significantly lower FIDs when using less channels.

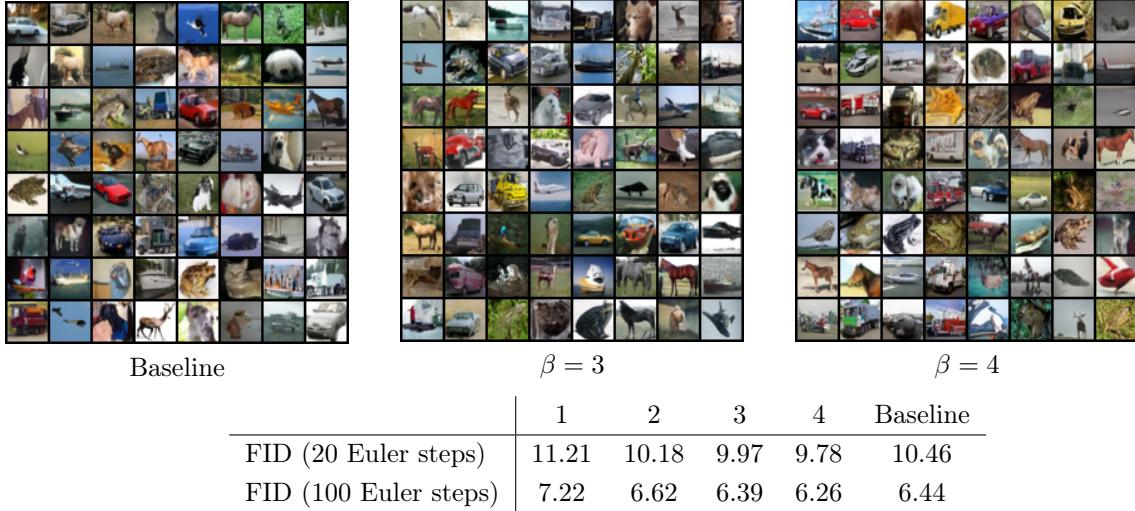


Figure 13: CIFAR results for different choices of regularization parameter and for the baseline. The visualized samples were generated using 100 Euler steps.

CIFAR-10. On the CIFAR-10 dataset, we evaluate our method in a setting characterized by strong spatial and inter-channel correlations, where product-measure approximations are inherently limited. We vary the regularization parameter β in Eq. D.3 while keeping the quantile loss weight

fixed at $\lambda = 5$ in Eq. 4.2. Figure 13 reports results for different values of β and compares them to a standard Gaussian baseline. Our results indicate that for uncorrelated noise, there exists a trade-off between the smoothness of the latent distribution and its closeness to the data. For independent noise on a highly correlated dataset, improvements remain marginal as expected since a product measure can only approximate the underlying data distribution to a limited extent. Given the substantial gains from adaptive noise on MNIST, we hypothesize that richer, correlation-aware noise models, beyond product measures, could realize similar improvements on CIFAR-10.

6 Conclusions

The result of this paper is a “quantile sandbox” for building generative models: a unifying theory and a practical toolkit that turns noise selection into a data-driven design element. Our construction plugs seamlessly into standard objectives including Flow Matching and consistency models, e.g. Inductive Moment Matching. Furthermore, our experiments demonstrate that it is possible to learn a freely parametrized, data-dependent latent distribution, beyond the usual smooth transformations of Gaussians. Our work opens several promising directions for future research. Extensions include developing time-dependent quantile functions to optimize the entire path distribution, not just the endpoint, as well as designing conditional quantile functions for tasks like class-conditional or text-to-image generation.

Acknowledgments. GS and RD acknowledge funding by the German Research Foundation (DFG) within the Excellence Cluster MATH+ and JC by project STE 571/17-2 within the *The Mathematics of Deep Learning*. GK acknowledges funding by the BMBF VIScreenPRO (ID: 100715327).

References

- [1] M. S. Albergo, N. M. Boffi, and E. Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023.
- [2] L. Ambrosio, N. Gigli, and G. Savaré. *Gradient Flows*. Lectures in Mathematics ETH Zürich. Birkhäuser, Basel, 2nd edition, 2008.
- [3] G. Bartosh, D. Vetrov, and C. A. Naesseth. Neural flow diffusion models: Learnable forward process for improved diffusion modelling, 2025.
- [4] C. Cattaneo. Sur une forme de l’équation de la chaleur éliminant le paradoxe d’une propagation instantanée. *Comptes Rendus.*, 247, 1958.
- [5] R. T. Q. Chen. torchdiffeq, 2018.
- [6] M. Chester. Second sound in solids. *Physical Review*, 131, 1963.
- [7] P. Dhariwal and A. Q. Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.
- [8] R. Duong, J. Chemseddine, P. Friz, and G. Steidl. Telegrapher’s generative model via Kac flows. *arXiv preprint arXiv::2506.20641*, 2025.

- [9] R. Duong, V. Stein, R. Beinert, J. Hertrich, and G. Steidl. Wasserstein gradient flows of MMD functionals with distance kernel and Cauchy problems on quantile functions. *ArXiv:2408.07498*, 2024.
- [10] C. Durkan, A. Bekasov, I. Murray, and G. Papamakarios. Neural spline flows. *Advances in neural information processing systems*, 32, 2019.
- [11] R. Flamary, N. Courty, A. Gramfort, M. Z. Alaya, A. Boisbunon, S. Chambon, L. Chapel, A. Corenflos, K. Fatras, N. Fournier, et al. Pot: Python Optimal Transport. *Journal of Machine Learning Research (JMLR)*, 22(1):3571–3578, 2021.
- [12] J. Gregory and R. Delbourgo. Piecewise rational quadratic interpolation to monotonic data. *IMA Journal of Numerical Analysis*, 2(2):123–130, 1982.
- [13] P. L. Hagemann and S. Neumayer. Stabilizing invertible neural networks using mixture models. *Inverse Problems*, 37(7):085002, 2021.
- [14] W. Han, C. Meng, , C. D. Manning, and S. Ermon. DistillKac: Few-step image generation via damped wave equations. *arXiv preprint arXiv:2509.215113*, 2025.
- [15] J. Hertrich, M. Gräf, R. Beinert, and G. Steidl. Wasserstein steepest descent flows of discrepancies with Riesz kernels. *Journal of Mathematical Analysis and Applications*, 531(1):127829, 2024.
- [16] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [17] M. Kac. A stochastic model related to the telegrapher’s equation. *Rocky Mountain Journal of Mathematics*, 4, 1974.
- [18] K. Kapusniak, P. Potaptchik, T. Reu, L. Zhang, A. Tong, M. Bronstein, A. J. Bose, and F. Di Giovanni. Metric flow matching for smooth interpolations on the data manifold. *arXiv preprint arXiv:2405.14780*, 2024.
- [19] D. Kim, S. Shin, K. Song, W. Kang, and I.-C. Moon. Soft truncation: A universal training technique of score-based diffusion model for high precision score estimation. *ICML*, 2022.
- [20] Y. Lipman, R. Chen, H. Ben-Hamu, M. Nickel, and M. Le. Flow matching for generative modeling. *ICLR*, 2023.
- [21] Q. Liu. Rectified flow: A marginal preserving approach to optimal transport. *arXiv preprint arXiv:2209.14577*, 2022.
- [22] R. M. Neal. Slice sampling. *The Annals of Statistics*, 31(3):705–767, 2003.
- [23] K. Pandey, J. Pathak, Y. Xu, S. Mandt, M. Pritchard, A. Vahdat, and M. Mardani. Heavy-tailed diffusion models, 2024.
- [24] J. Pidstrigach. Score-based generative models detect manifolds. *NeurIPS*, 2022.

- [25] A.-A. Pooladian, H. Ben-Hamu, C. Domingo-Enrich, B. Amos, Y. Lipman, and R. T. Q. Chen. Multisample flow matching: Straightening flows with batch couplings, 2023.
- [26] S. S. Sahoo, A. Gokaslan, C. De Sa, and V. Kuleshov. Diffusion models with learned adaptive noise. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.
- [27] A. Salmona, V. D. Bortoli, J. Delon, and A. Desolneux. Can push-forward generative models fit multimodal distributions? *Advances in Neural Information Processing Systems*, 35:0766–10779, 2022.
- [28] D. Shariatian, U. Simsekli, and A. Durmus. Heavy-tailed diffusion with denoising lévy probabilistic models, 2025.
- [29] D. Shariatian, U. Simsekli, and A. O. Durmus. Heavy-tailed diffusion with denoising levy probabilistic models. *ICLR 2025*, 2025.
- [30] J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In F. Bach and D. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 2256–2265, Lille, France, 07–09 Jul 2015. PMLR.
- [31] J. Song, C. Meng, and S. Ermon. Denoising diffusion implicit models. *arXiv preprint arXiv:2010.02502*, 2020.
- [32] Y. Song and S. Ermon. Generative modeling by estimating gradients of the data distribution. *ArXiv 1907.05600*, 2019.
- [33] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole. Score-based generative modeling through stochastic differential equations. *ICLR*, 2021.
- [34] R. C. Tautz and I. Lerche. Application of the three-dimensional telegraph equation to cosmic-ray transport. *Research in Astronomy and Astrophysics*, 2016.
- [35] A. Tong, K. Fatras, N. Malkin, G. Huguet, Y. Zhang, J. Rector-Brooks, G. Wolf, and Y. Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport, 2024.
- [36] P. Vernotte. Les paradoxes de la theorie continue de l'équation de la chaleur. *Comptes Rendus.*, 246, 1958.
- [37] C. Wald and G. Steidl. Flow Matching: Markov kernels, stochastic processes and transport plans. In *Variational and Information Flows in Machine Learning and Optimal Transport, Oberwolfach Seminars. Vol. 56*, pages 185–254. Birkhäuser, 2025.
- [38] T. Zhang, H. Zheng, J. Yao, X. Wang, M. Zhou, Y. Zhang, and Y. Wang. Long-tailed diffusion models with oriented calibration. In *The Twelfth International Conference on Learning Representations*, 2024.
- [39] L. Zhou, S. Ermon, and J. Song. Inductive moment matching. *ICML 2025*, 2025.

A Uniform Process and MMD Gradient Flow

We prove the proposition more general for $\nu = \mathcal{U}[a, b]$ and a flow starting in $x_0 \in [a, b]$ meaning that we show the following.

Proposition 8. *The Wasserstein gradient flow μ_t of the MMD functional (15) towards the uniform distribution $\nu = \mathcal{U}[a, b]$, $a < b$ starting in $\mu_0 = \delta_{x_0}$, $x_0 \in [a, b]$ reads as*

$$\mu_t = \mathcal{U}[a + (x_0 - a) \exp(-r(t)), b - (b - x_0) \exp(-r(t))], \quad t > 0 \quad (23)$$

with $r(t) := \frac{2t}{b-a}$ and has corresponding velocity field

$$v_t(x) = \frac{2}{b-a} \left(\frac{x - x_0}{\exp(r(t)) - 1} \right). \quad (24)$$

It holds $\|v_t\|_{L_2(\mathbb{R}^1, \mu_t)}^2 = \frac{2b}{3} \exp(-\frac{2t}{b})$, and hence, $\|v_t\|_{L_2(\mathbb{R}^1, \mu_t)} \in L_2(0, 1)$. A corresponding (stochastic) process $(U_t)_t$ is given by $U_t := b(1 - \exp(-\frac{t}{b}))U$, where $U \sim \mathcal{U}[-1, 1]$, such that $\text{Law}(U_t) = \mu_t$.

We need the relation between measures in $\mathcal{P}_2(\mathbb{R})$ and cumulative distribution functions, see (19). Using that $\nu = \mathcal{U}[a, b]$ has CDF

$$R_\nu(x) = \begin{cases} 0, & \text{if } x < a, \\ \frac{x-a}{b-a}, & \text{if } a \leq x \leq b, \\ 1, & \text{if } x > b \end{cases}$$

and $Q_\nu(s) = a(1-s) + bs$. it was shown in [15] that the functional $F_\nu: L_2(0, 1) \rightarrow \mathbb{R}$ defined by

$$F_\nu(u) := \int_0^1 \left((1-2s)(u(s) + Q_\nu(s)) + \int_0^1 |u(s) - Q_\nu(t)| dt \right) ds \quad (25)$$

fulfills $\mathcal{F}_\nu(\mu) = F_\nu(Q_\mu)$ for all $\mu \in \mathcal{P}_2(\mathbb{R})$. Moreover, we have the following equivalent characterization of Wasserstein gradient flows of \mathcal{F}_ν , which can be found in [9, Theorem 4.5].

Theorem 9. *Let \mathcal{F}_ν and F_ν be defined by (15) and (25), respectively. Then the Cauchy problem*

$$\begin{cases} \partial_t g(t) \in -\partial F_\nu(g(t)), & t \in (0, \infty), \\ g(0) = Q_{\mu_0}, \end{cases}$$

has a unique strong solution g , and the associated curve $\gamma_t := (g(t))_\# \Lambda_{(0,1)}$ is the unique Wasserstein gradient flow of \mathcal{F}_ν with $\gamma(0+) = (Q_{\mu_0})_\# \Lambda_{(0,1)}$. More precisely, there exists a velocity field v_t^* such that (γ_t, v_t^*) satisfies the continuity equation (9), and it holds the relations

$$v_t^* \circ g(t) \in -\partial F_\nu(g(t)) \quad \text{and} \quad v_t^* \in -\partial \mathcal{F}_\nu(\gamma_t). \quad (26)$$

Lastly note that here, the subdifferential $\partial F_\nu(u)$ is explicitly given by the singleton

$$-\partial F_\nu(u) = -\nabla F_\nu(u) = 2(\cdot - R_\nu \circ u) \quad \text{for all } u \in L_2(0, 1),$$

see [9, Lemma 4.3].

Proof of Proposition 5. We want to apply Theorem 9 to (μ_t, v_t) in (23) and (24). The uniform distribution in (23) has the quantile function

$$Q_{\mu_t}(s) = (1 - \exp(-r(t))) (a + (b - a)s) + x_0 \exp(-r(t)), \quad s \in (0, 1).$$

For all $t > 0$ and all $s \in (0, 1)$, we have $Q_{\mu_t}(s) \in [a, b]$ since $x_0 \in [a, b]$, and thus

$$\begin{aligned} -\nabla F_\nu(Q_{\mu_t})(s) &= 2s - 2r_\nu(Q_{\mu_t}(s)) \\ &= 2s - 2 \frac{(1 - \exp(-r(t))) (a + (b - a)s) + x_0 \exp(-r(t)) - a}{b - a} \\ &= 2 \left(s - \frac{x_0 - a}{b - a} \right) \exp(-r(t)). \end{aligned}$$

On the other hand, it holds

$$\partial_t Q_{\mu_t}(s) = -2 \frac{x_0 - a}{b - a} \exp(-r(t)) - \frac{(-2)(b - a)s}{b - a} \exp(-r(t)) = 2 \left(s - \frac{x_0 - a}{b - a} \right) \exp(-r(t)).$$

By Theorem 9, (μ_t) is the unique Wasserstein gradient flow of \mathcal{F}_ν starting in δ_0 .

Furthermore, there exists a velocity field v_t^* satisfying the continuity equation (9) and the relations (26). For $s \in (0, 1)$ and $t > 0$, let $y := g_s(t) = a + (x_0 - a) \exp(-r(t)) + (b - a)(1 - \exp(-r(t)))s$. Then, we have $s = \frac{y - a - (x_0 - a) \exp(-r(t))}{(b - a)(1 - \exp(-r(t)))}$, and thus by (26),

$$\begin{aligned} v_t^*(y) &= v_t^*(Q_{\mu_t}(s)) = 2 \left(s - \frac{x_0 - a}{b - a} \right) \exp(-r(t)) \\ &= 2 \left(\frac{y - a - (x_0 - a) \exp(-r(t))}{(b - a)(1 - \exp(-r(t)))} - \frac{x_0 - a}{b - a} \right) \exp(-r(t)) \\ &= \frac{2}{b - a} \left(\frac{y - a - (x_0 - a)}{1 - \exp(-r(t))} \right) \exp(-r(t)) \\ &= \frac{2}{b - a} \left(\frac{y - x_0}{\exp(r(t)) - 1} \right) \end{aligned}$$

for all $y \in g_s(t)(0, 1) = [a + (x_0 - a) \exp(-r(t)), b - (b - x_0) \exp(-r(t))]$. Lastly, let us compute

the action. For $t > 0$ we have

$$\begin{aligned}
\|v_t\|_{L^2(\mu_t)}^2 &= \int_{a+(x_0-a)\exp(-\frac{2t}{b-a})}^{b-(b-x_0)\exp(-\frac{2t}{b-a})} \frac{4(x-x_0)^2}{(b-a)^2 \left(\exp\left(\frac{2t}{b-a}\right) - 1\right)^2} \frac{1}{(b-a)(1-\exp(-\frac{2t}{b-a}))} dx \\
&= \frac{4}{(b-a)^3 \left(\exp\left(\frac{2t}{b-a}\right) - 1\right)^2 (1-\exp(-\frac{2t}{b-a}))} \int_{a+(x_0-a)\exp(-\frac{2t}{b-a})}^{b-(b-x_0)\exp(-\frac{2t}{b-a})} (x-x_0)^2 dx \\
&= \frac{4}{(b-a)^2 \exp\left(-\frac{2t}{b-a}\right) \left(\exp\left(\frac{2t}{b-a}\right) - 1\right)^3} \left[\frac{(x-x_0)^3}{3} \right]_{a+(x_0-a)\exp(-\frac{2t}{b-a})}^{b-(b-x_0)\exp(-\frac{2t}{b-a})} \\
&= \frac{4 \left(1-\exp\left(-\frac{2t}{b-a}\right)\right)^3}{3(b-a)^2 \exp\left(-\frac{2t}{b-a}\right) \left(\exp\left(\frac{2t}{b-a}\right) - 1\right)^3} [(b-x_0)^3 - (a-x_0)^3] \\
&= \frac{4 [(b-x_0)^3 - (a-x_0)^3]}{3(b-a)^2} \exp\left(-\frac{4t}{b-a}\right).
\end{aligned}$$

and the proof is finished. \square

Note that the fact that v_t^* is uniquely determined on $\text{supp } \mu_t = \overline{g_t(0,1)}$, correlates with the fact that the gradient $v_t^* \circ g(t) = -\nabla F_\nu(g(t))$ is a *singleton*. Outside of $\text{supp } \mu_t$, the velocity field may be arbitrarily extended, which yields a velocity $\tilde{v}_t \in -\partial \mathcal{F}_\nu(\mu_t)$ in a *non-singleton* subdifferential. The velocity v_t^* may be *uniquely* chosen from the tangent space $T_{\mu_t} \mathcal{P}_2(\mathbb{R})$, or equivalently, by choosing it to have minimal norm, i.e. $v_t^* \equiv 0$ outside of $\text{supp } \mu_t$.

B Flow Matching as Special Mean Reverting Processes

B.1 Gaussian Case

Let us shortly verify that our componentwise approach using the mean-reverting process (4), i.e.

$$\mathbf{X}_t := f(t) \mathbf{X}_0 + \mathbf{Y}_{g(t)},$$

leads to the usual FM objective. where we choose the scheduling functions $f(t) := 1-t$, $g(t) := t^2$, the target random variable $\mathbf{X}_0 \sim \mu_0$, and a standard Wiener process \mathbf{Y}_t in \mathbb{R}^d (independent of \mathbf{X}_0): First, it holds $\mathbf{Y}_{t^2} \sim \mathcal{N}(0, t^2 I_d)$, hence $\mathbf{Y}_{t^2} \stackrel{d}{=} t \mathbf{Z}$ with $\mathbf{Z} \sim \mathcal{N}(0, I_d)$, so that

$$\mathbf{X}_t \stackrel{d}{=} (1-t) \mathbf{X}_0 + t \mathbf{Z}.$$

Furthermore, by (11) the 1D components of \mathbf{Y}_t admit the velocity field $v_t^i(x^i) = \frac{x^i}{2t}$, $x^i \in \mathbb{R}$, and by Proposition 3 the multi-dimensional process \mathbf{Y}_t admits the velocity field $v_{\mathbf{Y}}(t, x) = (\frac{x^1}{2t}, \dots, \frac{x^d}{2t}) =$

$\frac{x}{2t}$, $x = (x^1, \dots, x^d) \in \mathbb{R}^d$. By the calculation (6), the conditional velocity field corresponding to \mathbf{X}_t starting in $x_0 \in \mathbb{R}^d$ reads as

$$\begin{aligned} v_{\mathbf{X}}(t, x | x_0) &= \dot{f}(t)x_0 + \dot{g}(t)v_Y(g(t), x - f(t)x_0 | 0) \\ &= -x_0 + 2t v_Y(t^2, x - (1-t)x_0 | 0) \\ &= -x_0 + \frac{x - (1-t)x_0}{t}. \end{aligned}$$

Now, if $x \sim P_{\mathbf{X}_t}(\cdot | x_0)$, i.e. $x = (1-t)x_0 + tz$ with $z \sim \mathcal{N}(0, I_d)$, then it follows

$$v_{\mathbf{X}}(t, x | x_0) = -x_0 + \frac{(1-t)x_0 + tz - (1-t)x_0}{t} = z - x_0, \quad (27)$$

which is the usual constant-in-time conditional FM velocity along the straight-line trajectories between $x_0 \sim \mu_0$ and $z \sim \mathcal{N}(0, I_d)$.

B.2 Uniform Case

Now consider any component of the mean-reverting process (4) with $f(t), g(t)$ to be chosen, X_0 being a component of $\mathbf{X}_0 \sim \mu_0$, and Y_t given by the MMD gradient flow (16), i.e. $Y_t := b(1 - \exp(-\frac{t}{b}))U$, where $U \sim \mathcal{U}[-1, 1]$. Let v_Y be the corresponding velocity field from (17). Then, we have

$$\begin{aligned} v_X(t, x | x_0) &= \dot{f}(t)x_0 + \dot{g}(t)v_Y(g(t), |x - f(t)x_0|) \frac{x - f(t)x_0}{|x - f(t)x_0|} \\ &= \dot{f}(t)x_0 + \dot{g}(t) \frac{x - f(t)x_0}{b \left(\exp\left(\frac{g(t)}{b}\right) - 1 \right)}. \end{aligned}$$

Now, along the trajectory $x \sim P_{X_t}(\cdot | x_0)$, i.e.

$$x = f(t)x_0 + b \left(1 - \exp\left(-\frac{g(t)}{b}\right) \right) u =: \alpha_t x_0 + \sigma_t u, \quad (28)$$

with $u \sim \mathcal{U}(-1, 1)$, the velocity calculates as

$$\begin{aligned} v_X(t, x | x_0) &= \dot{f}(t)x_0 + \dot{g}(t) \frac{b \left(1 - \exp\left(-\frac{g(t)}{b}\right) \right) u}{b \left(\exp\left(\frac{g(t)}{b}\right) - 1 \right)} \\ &= \dot{f}(t)x_0 + \dot{g}(t) \exp\left(\frac{-g(t)}{b}\right) u \\ &= \dot{\alpha}_t x_0 + \dot{\sigma}_t u, \end{aligned} \quad (29)$$

where $\alpha_t := f(t)$ and $\sigma_t := b \left(1 - \exp\left(-\frac{g(t)}{b}\right) \right)$. Hence, in order to minimize the CFM loss, we only need to sample $t \sim \mathcal{U}[0, 1]$, $x_0 \sim X_0$, and $u \sim \mathcal{U}(-1, 1)$. Note the similarity between the MMD path (28) and the FM/diffusion path (7); by choosing $b = 1$, $f(t) := 1 - t$ and $g(t) := -\log(1 - t)$ it follows $\alpha(t) = 1 - t$, $\sigma(t) = t$, and we obtain in (29) the FM-velocity along the trajectory (27), where the Gaussian noise $z \sim \mathcal{N}(0, 1)$ is just replaced by a uniform noise $u \sim \mathcal{U}(-1, 1)$.

Finally we want to note that the MMD functional (15) loses its convexity (along generalized geodesics) in multiple dimensions [15], and the general existence of its Wasserstein gradient flows is unclear in the multivariate case. This yields another reason to work in 1D.

C IMM with Quantile Interpolants

In this section, we want to demonstrate how the IMM framework proposed in [39] can be realized by our quantile approach. Note that in the following – for notational simplicity – we consider the one-dimensional case $X_0, Z_t \in \mathbb{R}$ where we can employ quantile functions. By combining the 1D components into a multivariate model $\mathbf{X}_0 = (X_0^1, \dots, X_0^d)$, $\mathbf{Z}_t = (Z_t^1, \dots, Z_t^d)$, the results of this chapter trivially extend to \mathbb{R}^d .

Recall our definition of the *quantile process*

$$Z_t = f(t)X_0 + Q_{g(t)}(U), \quad U \sim \mathcal{U}(0, 1), t \in [0, 1]. \quad (30)$$

and the *quantile interpolants*

$$I_{s,t}(x, y) = f(s)x + Q_{g(s)}(R_{g(t)}(y - f(t)x)), \quad s, t \in [0, 1]. \quad (31)$$

Note that by the assumptions (5) it holds $Z_0 = X_0$ and $Z_1 = Q_1(U)$.

By the following remark, our quantile interpolants generalize the interpolants used in Denoising Diffusion Implicit Models (DDIM).

Remark 10 (Relation to DDIM). *The interpolants used in Denoising Diffusion Implicit Models (DDIMs) [31] are given by*

$$\text{DDIM}_{s,t}(x, y) := \left(\alpha_s - \frac{\sigma_s}{\sigma_t} \alpha_t \right) x + \frac{\sigma_s}{\sigma_t} y. \quad (32)$$

Now let $f(t) := 1 - t$, $g(t) := t^2$ and let Q_t be the quantile of the law of a standard Brownian motion W_t .

First we obtain

$$Q_{g(t)}(p) = Q_{t^2}(p) = Q_{\mathcal{N}(0, t^2)}(p) = t\sqrt{2} \operatorname{erf}^{-1}(2p - 1) = t Q_{\mathcal{N}(0, 1)}(p), \quad p \in (0, 1),$$

with the error function erf . Hence, (30) exactly becomes (not only in distribution)

$$Z_t = (1 - t)Y_0 + t Q_{\mathcal{N}(0, 1)}(U) = (1 - t)Y_0 + tZ,$$

where $Z := Q_{\mathcal{N}(0, 1)}(U) \sim \mathcal{N}(0, 1)$, i.e. the components of (7) with the choice $\alpha_t = 1 - t$, $\sigma_t = t$. Furthermore, since $R_{t^2}(z) = R_{\mathcal{N}(0, t^2)}(z) = \frac{1}{2}(1 + \operatorname{erf}\left(\frac{z}{t\sqrt{2}}\right))$, the quantile interpolant (21) reads as

$$\begin{aligned} I_{s,t}(x, y) &= (1 - s)x + s\sqrt{2} \operatorname{erf}^{-1}\left(\operatorname{erf}\left(\frac{y - (1 - t)x}{t\sqrt{2}}\right)\right) = (1 - s)x + \frac{s}{t}(y - (1 - t)x) \\ &= ((1 - s) - \frac{s}{t}(1 - t))x + \frac{s}{t}y. \end{aligned}$$

which is exactly $\text{DDIM}_{s,t}(x, y)$ in (32) with $\alpha_t = f(t)$ and $\sigma_t^2 = g(t)$. \diamond

Exactly as the DDIM interpolants, our quantile interpolants (31) satisfy the following crucial interpolation properties.

Proposition 11 (a.k.a Proposition 6). *For all $x, y \in \mathbb{R}$ and all $s, r, t \in [0, 1]$, it holds*

$$I_{0,t}(x, y) = x, \quad I_{t,t}(x, y) = y, \quad (33)$$

and

$$I_{s,r}(x, I_{r,t}(x, y)) = I_{s,t}(x, y).$$

Furthermore, inserting the quantile process (20) yields

$$I_{s,t}(Z_0, Z_t) = Z_s. \quad (34)$$

Proof. By assumptions it holds

$$I_{0,t}(x, y) = f(0)x + Q_{g(0)}(R_{g(t)}(y - f(t)x)) = x,$$

and

$$I_{t,t}(x, y) = f(t)x + Q_{g(t)}(R_{g(t)}(y - f(t)x)) = y.$$

Furthermore, it holds the interpolation/consistency property

$$\begin{aligned} I_{s,r}(x, I_{r,t}(x, y)) &= f(s)x + Q_{g(s)}(R_{g(r)}(I_{r,t}(x, y) - f(r)x)) \\ &= f(s)x + Q_{g(s)}\left(R_{g(r)}\left(\cancel{f(r)x} + \cancel{Q_{g(r)}(R_{g(t)}(y - f(t)x))} - \cancel{f(r)x}\right)\right) \\ &= f(s)x + Q_{g(s)}(R_{g(t)}(y - f(t)x)) \\ &= I_{s,t}(x, y) \end{aligned}$$

for all $x, y \in \mathbb{R}$. Also note that inserting the random variables Z_0, Z_t yields

$$\begin{aligned} I_{s,t}(Z_0, Z_t) &= f(s)Z_0 + Q_{g(s)}(R_{g(t)}(Z_t - f(t)Z_0)) \\ &= f(s)Z_0 + Q_{g(s)}(U) \\ &= Z_s. \end{aligned}$$

This finishes the proof. □

Proposition 11 represents the key observation which allows us to utilize our quantile process (30) in the IMM framework the same way as [39] employ the DDIM interpolants (32):

For this, let us now recall the basic idea of inductive moment matching and the corresponding loss functions. Let us distinguish between real numbers written in small letters ($x_0, u, z_t \in \mathbb{R}$) and random variables written with capital letters (X_0, U, Z_t, \dots). We assume that the probability distributions have densities:

$$\frac{\text{Law}(X_0)}{\rho_0(x_0)} \mid \frac{\text{Law}(Z_t)}{\rho_t(z_t)} \mid \frac{\text{Law}(Z_s|X_0 = x_0, Z_t = z_t)}{\rho_{s|0,t}(z_s|x_0, z_t)} \mid \frac{\text{Law}(Z_t|X_0 = x_0, U = u)}{\rho_{t|0,1}(z_t|x_0, u)} \mid \frac{\text{Law}(X_0|Z_t = z_t)}{\rho_{0|t}(x_0|z_t)}$$

Note that by (34) we have $\rho_{s|0,t}(z_s|x_0, z_t) = \text{Law}(I_{s,t}(x_0, z_t))(z_s) = \delta(z_s - I_{s,t}(x_0, z_t))$, hence sampling from $\rho_{s|0,t}(z_s|x_0, z_t)$ is just applying $I_{s,t}(x_0, z_t)$. Similarly, sampling from $\rho_{t|0,1}(z_t|x_0, u)$ is just evaluating $I_{t,1}(x_0, Q_1(u))$.

The following proposition follows directly from Proposition 11 as in [39]. It is essential for deriving the appropriate loss functions.

Proposition 12. For all $0 \leq s \leq r \leq t \leq 1$, the quantile interpolant (31) is self-consistent, i.e.

$$\rho_{s|0,t}(z_s|x_0, z_t) = \int_{\mathbb{R}} \rho_{s|0,r}(z_s|x_0, z_r) \rho_{r|0,t}(z_r|x_0, z_t) dz_r,$$

and the quantile process (30) is marginal preserving, i.e.

$$\rho_s(z_s) = \mathbb{E}_{z_t \sim \rho_t, x_0 \sim \rho_{0|t}(\cdot|z_t)} [\rho_{s|0,t}(z_s|x_0, z_t)].$$

Learning. The conditional probability $\rho_{0|t}(\cdot|z_t)$ is now approximated by a network p_{s,t,z_t}^θ where the parameter s describes the dependence on ρ_s such that

$$\rho_s \approx \mathbb{E}_{z_t \sim \rho_t, x_0 \sim p_{s,t,z_t}^\theta} [\rho_{s|0,t}(\cdot|x_0, z_t)] =: p^\theta(s, t). \quad (35)$$

Then it is proposed in [39, Eq. (7)] to minimize the so-called *naïve objective*

$$\mathcal{L}_{\text{naive}}(\theta) := \mathbb{E}_{s,t} [D(\rho_s, p^\theta(s, t))], \quad (36)$$

with an appropriate metric D , e.g. MMD. The procedure is now as follows: starting in a sample x_0 from X_0 , we can sample z_s, z_t from Z_s, Z_t by (30), respectively; then given z_t we sample \tilde{x}_0 from p_{s,t,z_t}^θ , and finally we can evaluate $\tilde{z}_s = I(\tilde{x}_0, z_t)$ from (34), which is then compared with z_s .

Inference. The following iterative multi-step sampling can be applied: for chosen decreasing $t_k \in (0, 1]$, $k = 0, \dots, T$ with $t_0 = 1$, starting with $x_0^{(0)} \sim p_{0,1,z_1}^\theta$, we compute

$$z_{t_k} = I_{t_k, t_{k-1}} \left(x_0^{(k-1)}, z_{t_{k-1}} \right), \quad x_0^{(k)} \sim p_{0,t_k,z_{t_k}}^\theta, \quad k = 1, \dots, T.$$

Although for marginal-preserving interpolants, a minimizer of $\mathcal{L}_{\text{naive}}$ exists with minimum 0, the authors of [39] object that directly optimizing (36) faces practical difficulties when t is far away from s . Instead, they propose to apply the following “inductive bootstrapping” technique:

Bootstrapping. Instead of minimizing (36), we consider the *general objective*

$$\mathcal{L}_{\text{general}}(\theta) := \mathbb{E}_{s,t} [w(s, t) \text{MMD}_K^2(p^{\theta_{n-1}}(s, r), p^{\theta_n}(s, t))], \quad (37)$$

with a weighting function $w(s, t)$ to be chosen. The kernel K of the squared MMD distance can be chosen as e.g. the (time-dependent) Laplace kernel. Importantly, the value r is chosen to be a function $r = r_{s,t} \in [s, t]$ being “close to t ” and fulfilling a suitable monotonicity property.

Let us assume the simplest case $r_{s,t} := \max\{s, t - \varepsilon\}$ with a small fixed $\varepsilon > 0$ and hereby demonstrate the bootstrapping technique: Fix $s \in [0, 1]$. Then, it holds for all $t \in [s, s + \varepsilon]$ that $r_{s,s} = s$. By the definition (35) and property (33), it holds (independently of θ) that $p^\theta(s, s)(z_s) = \rho_s(z_s)$. Hence, minimizing (37) in the first step $n = 1$ yields

$$0 = \text{MMD}_K^2(p^{\theta_0}(s, s), p^{\theta_1}(s, t_1)) = \text{MMD}_K^2(\rho_s, p^{\theta_1}(s, t_1)) \quad \text{for all } t_1 \in [s, s + \varepsilon].$$

In the second step $n = 2$, it holds for all $t_2 \in [s, s + 2\varepsilon]$ that $r_{s,t_2} \in [s, s + \varepsilon]$. Hence, minimizing (37) in the second step yields, together with the first step,

$$0 = \text{MMD}_K^2(p^{\theta_1}(s, r_{s,t_2}), p^{\theta_2}(s, t_2)) = \text{MMD}_K^2(\rho_s, p^{\theta_2}(s, t_2)) \quad \text{for all } t_2 \in [s, s + 2\varepsilon].$$

Thus, for the number of steps $n \rightarrow \infty$, it holds $0 = \text{MMD}_K^2(\rho_s, p^{\theta_n}(s, t_n))$ even for the entire interval $t_n \in [s, 1]$. Hence, minimizing the general objective (37) with a large number of steps eventually minimizes the naïve objective (36), see [39, Theorem 1] for more details.

D Adapting Noise to Data

D.1 Toy Target Distributions



Figure 14: A generated sample path from the learned quantile latent to the checkerboard. The adapted latent (left) is already close to the target distribution.

We use three standard challenging low-dimensional distributions: Neal’s funnel, a 3×3 Gaussian mixture, and a checkerboard.

Funnel. For the toy illustration in Figure 6, we work with the dataset known as Neals Funnel [22]. The distribution of Neal’s funnel is defined as follows:

$$p(x_1, x_2) = \mathcal{N}(x_1; 0, 3) \mathcal{N}(x_2; 0, \exp(x_1/2)).$$

Grid Gaussian Mixture. We give more details about the mixture of Gaussian we consider in our experiment. It is designed in a grid pattern in $[-1, 1]^2$, as follows:

$$\sum_{i=1}^9 w_i \cdot \mathcal{N}(\mu_i, \sigma^2 I_2),$$

where $(w_i)_{i=1}^9 = (0.01, 0.1, 0.3, 0.2, 0.02, 0.15, 0.02, 0.15, 0.05)$, $\mu_i = (\mu_1, \mu_2)$ with $\mu_1 = (i \bmod 3) - 1$, $\mu_2 = \lfloor \frac{i}{3} \rfloor - 1$, and $\sigma = 0.025$.

Checkerboard. Fix $\ell < h$ and domain $\Omega = [\ell, h]^2$. Define the support

$$\mathcal{S} = \{(x, y) \in \Omega : |x| + |y| \text{ is even}\}.$$

The checkerboard distribution is uniform on \mathcal{S} and zero elsewhere:

$$p_{\text{Checker}}(x, y) = \begin{cases} \frac{1}{\text{area}(\mathcal{S})}, & (x, y) \in \mathcal{S}, \\ 0, & \text{otherwise.} \end{cases}$$

For integer ℓ, h with even side length (e.g. $\ell = -4, h = 4$), exactly half of Ω is active, hence

$$p_{\text{Checker}}(x, y) = \frac{2}{(h - \ell)^2} \mathbf{1}_{\mathcal{S}}(x, y).$$

D.2 Details on the Architecture of the Learned Quantiles

We implement each one-dimensional quantile function with rational-quadratic splines (RQS) [12, 10]. We explored several ways to map $u \in (0, 1)$ into the spline input; the two variants below consistently performed well and are used in our experiments. For every coordinate i , we write

$$Q_\phi^i(u) = S_\phi^i(\psi(u)), \quad u \in (0, 1),$$

where $S_\phi^i : \mathbb{R} \rightarrow \mathbb{R}$ is a strictly increasing RQS with an interior knot interval $(-B, B)$ (with K bins) and linear tails outside $\pm B$ that are C^1 -matched at the boundaries. The two settings differ only in the “activation” ψ :

$$(A) \text{ Logit: } \psi(u) = \text{logit}(u), \quad (B) \text{ Affine: } \psi(u) = \alpha_B(u) = B(2u - 1).$$

Thus, both (A) and (B) share exactly the same spline S_ϕ^i architecture—including the bounded interior $(-B, B)$ and slope-matched linear tails—and differ only in how $(0, 1)$ is mapped into the spline’s input. In (A), $\psi(u) \in \mathbb{R}$ and the linear tails of S_ϕ^i are used whenever $|\text{logit}(u)| > B$; in (B), $\psi(u) \in (-B, B)$ so the forward pass never touches the tails (they remain important for invertibility and out-of-range evaluation).

Parameterization and constraints. Each spline S_ϕ^i is parameterized by raw bin widths, heights, and knot slopes. We pass these raw parameters through softplus, normalize widths and heights to sum to one (scaled to the domain span $2B$ and the learned range span, respectively), and add a small constant $s_{\min} > 0$ to each slope to enforce a positive lower bound. The linear tail slopes (left/right) are learned in the same way and are chosen so that both function value and slope agree at $\pm B$. These constraints guarantee strict monotonicity, hence Q_ϕ^i is strictly increasing on $(0, 1)$ under both (A) and (B). Closed-form formulas for the spline pieces and their (log-)derivatives are available; by the chain rule,

$$\frac{d}{du} Q_\phi^i(u) = S_\phi^{ii}(\psi(u)) \psi'(u), \quad \text{with} \quad \psi'(u) = \begin{cases} \frac{1}{u(1-u)} & \text{for (A),} \\ 2B & \text{for (B).} \end{cases}$$

Per-component affine wrapper (scale/bias). After computing $Q_\phi^i(u)$, we add a tiny affine head per coordinate:

$$\tilde{Q}_\phi^i(u) = s_i Q_\phi^i(u) + b_i, \quad s_i = \text{softplus}(\log \alpha_i), \quad b_i = \beta_i,$$

where $\alpha_i > 0$ and $\beta_i \in \mathbb{R}$ are learned per component. Using $\text{softplus}(\log \alpha_i)$ keeps $s_i > 0$ with a convenient dynamic range; this preserves monotonicity and adds only one scale and one bias parameter per component.

D.3 Regularization via Expected Negative Log-Jacobian

Let $Q_\phi : (0, 1)^d \rightarrow \mathbb{R}^d$ be the componentwise map with affine heads, $Q_\phi(u) = (\tilde{Q}_\phi^1(u_1), \dots, \tilde{Q}_\phi^d(u_d))$. Since the construction is per-coordinate, the Jacobian is diagonal with entries $\partial_{u_i} \tilde{Q}_\phi^i(u_i) > 0$. We

regularize with the expected negative log-determinant of the Jacobian:

$$\begin{aligned}\mathcal{L}_{\text{reg}}(\phi) &= \lambda_{\text{reg}} \mathbb{E}_{u \sim p_U} [-\log \det J_{Q_\phi}(u)] \\ &= \lambda_{\text{reg}} \mathbb{E}_{u \sim p_U} \left[- \sum_{i=1}^d \log (\partial_{u_i} \tilde{Q}_\phi^i(u_i)) \right].\end{aligned}$$

Here $p_U = \text{Unif}((0, 1)^d)$. In practice, we evaluate the log-derivatives in closed form.

D.4 Minibatch Optimal Transport

Since the learned latent distribution is close to the data distribution, we can exploit this improved matching via an optimal transport coupling. For training, the minibatch OT is computed empirically as follows: draw a minibatch $\{\mathbf{x}_0^{(i)}\}_{i=1}^B \sim \mu_0$ and $\{\mathbf{u}^{(j)}\}_{j=1}^B \sim \mathcal{U}([0, 1]^d)$, set $\mathbf{y}^{(j)} = \mathbf{Q}_\phi(\mathbf{u}^{(j)})$, and define the empirical measures

$$\hat{\mu}_0^B = \frac{1}{B} \sum_{i=1}^B \delta_{\mathbf{x}_0^{(i)}}, \quad \hat{\nu}_\phi^B = \frac{1}{B} \sum_{j=1}^B \delta_{\mathbf{y}^{(j)}}.$$

The minibatch objective is

$$\hat{\mathcal{E}}_{\mathbf{Q}}(\phi) = D(\hat{\mu}_0^B, \hat{\nu}_\phi^B),$$

and gradients backpropagate through $\mathbf{y}^{(j)} = \mathbf{Q}_\phi(\mathbf{u}^{(j)})$.

Furthermore, we use the linear path $\mathbf{x}_t^{(j)} = (1-t_j)\mathbf{x}_0^{(j)} + t_j \mathbf{y}^{(T(j))}$, $j = 1, \dots, B$, with $t_j \sim \mathcal{U}(0, 1)$, the target velocity $\mathbf{y}^{(\pi(j))} - \mathbf{x}_0^{(j)}$, and we optimize the empirical versions

$$\hat{\mathcal{E}}_{\mathbf{FM}}(\theta; \phi) = \frac{1}{B} \sum_{j=1}^B \|v_\theta(\mathbf{x}_t^{(j)}, t_j) - (\mathbf{y}_\phi^{(T(j))} - \mathbf{x}_0^{(j)})\|_2^2, \quad \hat{\mathcal{L}}_{\text{joint}} = \hat{\mathcal{E}}_{\mathbf{FM}} + \lambda_Q \hat{\mathcal{E}}_Q.$$

E Implementation Details

We support baseline flow matching, optional quantile pretraining, and joint quantile+velocity optimisation. Pretraining fits the RQS transport before optionally freezing it; joint training updates both modules simultaneously. Once the quantile learning rate decays to zero we freeze its weights and continue optimising the velocity field only.

The coupling plans are calculated using the Python Optimal Transport package [11]. For inference simulate the corresponding ODEs using the torchdiffeq [5] package. For all models we only used the batch size 128 and learning rate $2e-4$ for the velocities. Quantile transports are parameterised by stacked rational-quadratic splines following [10].

E.1 Synthetic Examples

All models include a sinusoidal time embedding and SiLU activation functions.

Funnel. For all models we used 3 hidden layers with width 64. We used a batch size of 128, a learning rate of $2e - 4$ and exponential moving average on the network weights of 0.999. The baselines were trained for 200,000 iterations. We pretrain our quantiles and use the frozen quantiles during flow matching. We trained our quantile for 50,000 steps and to compensate we trained our velocity for only 150,000 steps. For the RQS we choose logit activation, 32 bins, a bound of 500 and one layer.

Grid Gaussian Mixture and Checker. The quantiles were trained for the first 20,000 steps, after which the learning rate was linearly decayed to 0 by step 25,000. For both datasets, we trained the velocity model with 4 layers and a hidden width of 256 for 100,000 steps. For the RQS we choose the parameters number of bins 32, bound 5, layers 3.

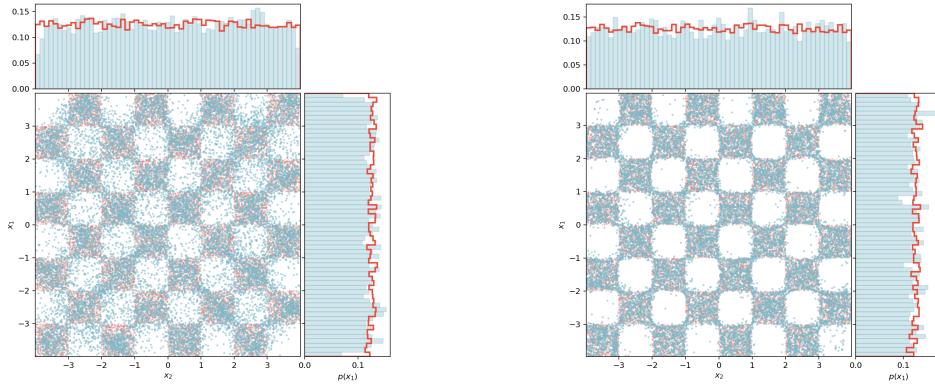


Figure 15: Flow Matching with optimal coupling using Gaussian noise (left) and our learned noise (right) after **20k** training steps with identical parameters. Generated samples are shown in blue, and ground-truth samples in red

E.2 Image Experiments

For both image datasets, we adapt the U-Net from [7] to parametrize our velocity field.

MNIST. For the MNIST dataset we use the U-Net with base width 64, channel multipliers (1, 2, 4), two residual blocks per resolution, attention at 7×7 , 1 attention head, and dropout 0.1. We clip the gradient norm to 1 and use exponential moving averaging with a decay of 0.99. The quantiles were trained for the first 20,000 steps, after which the learning rate was linearly decayed to 0 by step 25,000.

CIFAR. Here we use the U-Net with base width 128, channel multipliers (1, 2, 2, 2), two residual blocks per resolution, attention at 16×16 , four attention heads , and dropout 0.1. We clip the gradient norm to 1 and use exponential moving averaging with a decay of 0.9999. To evaluate our

results, we use the Fréchet inception distance (FID) [16]. The quantiles were trained for the first 20,000 steps, after which the learning rate was linearly decayed to 0 by step 25,000.

CIFAR-10 inputs are normalized to $[-1, 1]$ with random horizontal flips.