Are We Really Learning the Score Function? Reinterpreting Diffusion Models Through Wasserstein Gradient Flow Matching

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

Diffusion models are commonly interpreted as learning the *score function*, i.e., the gradient of the log-density of noisy data. However, this assumption implies that the target of learning is a conservative vector field, which is not enforced by the neural network architectures used in practice. We present numerical evidence that trained diffusion networks violate both integral and differential constraints required of true score functions, demonstrating that the learned vector fields are not conservative. Despite this, the models perform remarkably well as generative mechanisms. To explain this apparent paradox, we advocate a new theoretical perspective: diffusion training is better understood as *flow matching* to the velocity field of a Wasserstein Gradient Flow (WGF), rather than as score learning for a reverse-time stochastic differential equation. Under this view, the "probability flow" arises naturally from the WGF framework, eliminating the need to invoke reverse-time SDE theory and clarifying why generative sampling remains successful even when the neural vector field is not a true score. We further show that non-conservative errors from neural approximation do not necessarily harm density transport. Our results advocate for adopting the WGF perspective as a principled, elegant, and theoretically grounded framework for understanding diffusion generative models.

1 Background

Diffusion models are typically described as follows: Given D-dimensional samples $x \in \mathbb{R}^D$ drawn from a data distribution μ_0, one defines a forward Itô process that gradually corrupts x into noise. Throughout this paper, we use the continuous-time Ornstein–Uhlenbeck (OU) process for concreteness:[1]

$$\mathrm{d}X_t = -X_t\,\mathrm{d}t + \sqrt{2}\,\mathrm{d}W_t, \qquad X_0 = x \sim \mu_0, \tag{1}$$

where each component of W_t is a standard Wiener process. The process (1) converges to a limiting distribution μ_∞ as $t \to \infty$, which is an isotropic Gaussian in $\mathbb{R}^D$. Because of the choice of diagonal matrices in the drift and diffusion terms, each component of X_t follows the well-studied one-dimensional OU process.

[1] [16] established the equivalence of the OU process with the discrete-time Denoising Diffusion Probabilistic Model [10] and the score-based formulation [20]. This setup is often called "variance-preserving" (VP), though this term is misleading: for each sample, the variance is not constant over time (which, in most scientific contexts, is the definition of "preserving"), but grows as $\sqrt{1-e^{-2t}}$. Our analysis extends naturally to the standard Brownian motion process $\mathrm{d}X_t = \mathrm{d}W_t$, commonly termed "variance-exploding" (VE).

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

Equivalently, the forward dynamics can be described in terms of densities. The transition kernel[2] $\rho(\xi, t|\zeta, s)$ satisfies the Fokker–Planck Equation (FPE):

$$\partial_t \rho(\xi, t|\zeta, s) = \nabla_\xi [\xi \, \rho(\xi, t|\zeta, s)] + \nabla_\xi^2 \rho(\xi, t|\zeta, s), \tag{2}$$

with the initial condition $\rho(\xi, 0) = \delta(\xi - x)$ for each of the drawn samples $x \sim \mu_0$, where $\delta(\cdot)$ denotes the Dirac delta distribution.

The modern understanding of diffusion models is grounded in Anderson's reverse-time theory [2], which guarantees the existence of a reverse-time Itô process that transforms samples from the simple distribution μ_∞ back into data-like samples as $t : \infty \to 0$:

$$\mathrm{d}X_\tau = [X_\tau + 2s\,(X_\tau, -\tau)]\,\mathrm{d}\tau + \sqrt{2}\,\mathrm{d}W_\tau, \quad X_{-\infty} \sim \mu_\infty. \tag{3}$$

Here, we define $\tau := -t$, $\tau : -\infty \to 0$, $\rho(x, t)$ denotes the forward density with initial distribution μ_0, $s(\xi, t) := \nabla_\xi \log \rho(\xi, t) \in \mathbb{R}^D$ is the score function of the corrupted (forward) distribution given initial distribution μ_0, and $\mathrm{d}W_s$ is again a multi-dimensional Wiener process. The central training objective of diffusion models is thus framed as *learning the score function* $s(x, t)$ [20]. In practice, a neural network $\mathbb{R}^D \times \mathbb{R} \to \mathbb{R}^D$ is used to approximate $s(x, t)$, which is then plugged into (3) during sampling.

A key point is that the score function has a special mathematical structure: it is a conservative field. Neural networks used in practice are not constrained to produce conservative vector fields and, therefore, do not necessarily preserve this structure. This raises the central question of this study:

> Does a trained neural network actually learn a valid *score function*, or merely a useful vector field for generative sampling?

1.1 Wasserstein Gradient Flow

Wasserstein Gradient Flow (WGF) originates from the theory of optimal transport (OT), but it has become increasingly relevant for understanding modern generative models. Here, we provide a brief overview and refer readers to the classic references [1, 7] for comprehensive materials.

Recall the forward evolution of the probability density $\rho(x, t)$ under the FPE (2). In their seminal work, Jordan, Kinderlehrer, and Otto observed that an implicit Euler discretization of the FPE can be reinterpreted as a variational problem: each timestep corresponds to minimizing a free energy functional that combines Shannon entropy with a Wasserstein-2 distance penalty [11]. This insight, known as the *JKO scheme*, shows that the FPE can be understood as a gradient flow of entropy in the space of probability measures.

Building on this idea, Otto introduced a formal Riemannian calculus on the space of probability distributions, demonstrating that the FPE defines a steepest descent in Wasserstein geometry [15]. This framework—now widely known as Otto calculus—precisely formalizes the notion that probability densities evolve like particles sliding down an energy landscape, but within the geometry induced by optimal transport. In addition, Otto also introduced the generalized Liouville equation (GLE)[3][9]. Taken together, the JKO scheme and Otto's formulation provide the foundation for WGF, unifying PDE evolution, entropy maximization, and optimal transport. One powerful result of WGF theory is:

> While the sample paths of the diffusion process that FPE describes are fundamentally *stochastic*, the marginal distribution[4] of the paths at a specific time, $\rho(\cdot, t)$, is identical to the marginal distribution of the trajectories driven by a deterministic WGF.

[2] Since the OU process decomposes into D independent one-dimensional processes, the density factorizes across coordinates: $\rho(\xi, t \mid \zeta, s) = \prod_{i=1}^D \rho_i(\xi_i, t \mid \zeta_i, s)$

[3] We distinguish GLE from the "continuity equation", a term commonly used in the field of OT. We make this distinction because continuity equations in physics can describe arbitrary conserved quantities (mass, energy, etc.), but the GLE specifically governs normalized probability density functions.

[4] $\rho(\cdot, t)$ is referred to as the marginal distribution because it is only the distribution of X_t at time t. It is a marginal distribution of the the joint distribution specified the stochastic process, $\rho\left(x_{t_1}, \ldots x_{t_N}\right)$.

To see this, let us consider setting the energy functional as the sum of a quadratic potential and the negative Shannon entropy

$$E\left\{\rho\left(\cdot,t\right)\right\} := \int \frac{x^2}{2}\rho\left(x,t\right)\mathrm{d}x + \int \rho\left(x,t\right)\log\rho\left(x,t\right)\,\mathrm{d}x. \tag{4}$$

Here, the first term accounts for the drift/advection and the second for the diffusion in the FPE (2). The idea is to identify the *steepest descent* direction functions that decrease the energy the most in the space of probability density functions induced by a deterministic velocity field $v(x,t)$. Applying $\mathrm{d}/\mathrm{d}t$ to the energy functional:

$$\frac{\mathrm{d}}{\mathrm{d}t}E\left\{\rho\left(\cdot,t\right)\right\} = \int \frac{\delta E\left\{\rho\left(\cdot,t\right)\right\}}{\delta\rho(x,t)}\frac{\partial\rho\left(x,t\right)}{\partial t}\mathrm{d}x, \tag{5}$$

where the functional variation of E with respect to the density function ρ can be explicitly computed:

$$\begin{aligned}\frac{\delta E\left\{\rho\left(\cdot,t\right)\right\}}{\delta\rho(x,t)} &:= \frac{1}{\delta\rho(x,t)}\left[\int \frac{x^2}{2}\delta\rho\left(x,t\right) + \left(\rho+\delta\rho\right)\log\left(\rho+\delta\rho\right)\,\mathrm{d}x - \int \rho\left(x,t\right)\log\rho\left(x,t\right)\,\mathrm{d}x\right]\\ &\sim \frac{1}{\delta\rho(x,t)}\int\left[\frac{x^2}{2} + \log\rho(x,t) + 1\right]\delta\rho\left(x,t\right)\,\mathrm{d}x = \frac{x^2}{2} + \log\rho(x,t).\end{aligned} \tag{6}$$

In the last two equations, we neglected higher-order $\mathcal{O}(\delta\rho(x,t))$ terms (using the asymptotic symbol $\sim$) and applied the normalization condition that the functional perturbation $\int \delta\rho(x,t)\,\mathrm{d}x = 0$ because $\int \rho(x,t)\,\mathrm{d}x = 1 = \int\left(\rho+\delta\rho\right)(x,t)\,\mathrm{d}x$. Next, inserting GLE [9] (see footnote 3):

$$\partial_t\rho\left(x,t\right) = -\nabla_x\cdot\left[v(x,t)\rho(x,t)\right], \tag{7}$$

and the functional variation (6) into (5) leads to

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}E\left\{\rho\left(\cdot,t\right)\right\} &= -\int\left[\frac{x^2}{2} + \log\rho\left(x,t\right)\right]\nabla_x\cdot\left[v(x,t)\rho(x,t)\right]\,\mathrm{d}x\\ &= \int v(x,t)\cdot\left[x + \nabla_x\log\rho\left(x,t\right)\right]\,\rho(x,t)\,\mathrm{d}x,\end{aligned} \tag{8}$$

where we used integration by parts and assumed vanishing boundary terms. The above equation can be interpreted as an inner product of the functions $v(\cdot,t)$ and $\nabla_x\log\rho(\cdot,t)$ under the measure $\rho(\cdot,t)$. Clearly, the velocity field that corresponds to the steepest descent of the energy functional should align with the opposite direction of $\nabla_x\log(\cdot,t)$ (up to a global multiplicative constant):

$$v_{\text{WGF}}(x,t) := -x - \nabla_x\log\rho(x,t) = -x - s(x,t). \tag{9}$$

The probability distribution of the resulting flow system with the above velocity field evolves under the GLE:

$$\frac{\partial}{\partial t}\rho\left(x,t\right) = -\nabla_x\cdot\left[v_{\text{WGF}}(x,t)\rho(x,t)\right] = \nabla_x\left[\left(x + \nabla_x\log\rho(x,t)\right)\rho\left(x,t\right)\right], \tag{10}$$

which is exactly the FPE (2) describing the OU.

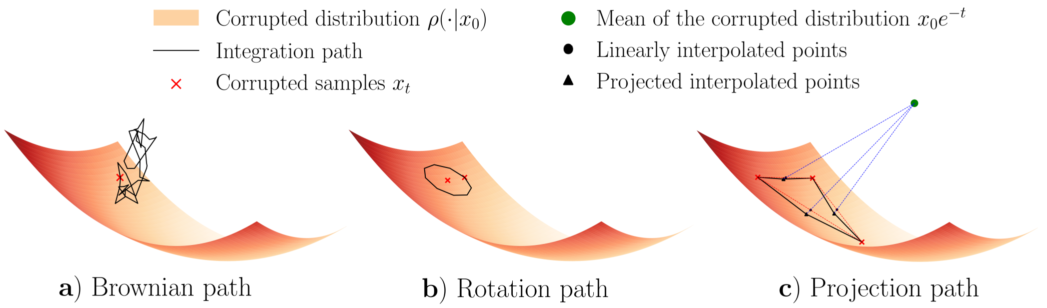
Song et al. rediscovered the WGF velocity field (9) through manipulating the FPE and noticing $\nabla_x\rho(x,t) = \rho(x,t)\,\nabla_x\log\rho(x,t)$. They used the term "probability flow", without referencing the JKO scheme, Otto calculus, and WGF. We believe it is beneficial to point out the origin of this theoretical framework, given its deeper connection to OT and the variational nature of the diffusion process.

2 Numerical experiments

We now shift our focus to numerical experiments to verify the central question we have in score-based generative modeling: *Are we learning the score function?*

Due to the definition of the score function, $s(x,t) := \nabla_x\log\rho(x,t)$, the fundamental theorem of calculus (or generalized Stokes' theorem in high dimension) states that the line integral of the score function along a closed path in the state space has to be equal to zero:

$$\oint \vec{s}(x,t)\cdot\mathrm{d}\vec{x} = 0. \tag{11}$$


Corrupted distribution $\rho(\cdot|x_0)$
Integration path
Corrupted samples x_t
Mean of the corrupted distribution x_0e^{-t}
Linearly interpolated points
Projected interpolated points

a) Brownian path **b**) Rotation path **c**) Projection path

Figure 1: Mechanisms for assessing integral constraints. Illustration of the three mechanisms we used to construct closed paths for evaluating integral constraints within the high-density regions of the data distribution.

We will refer to (11) as the *integral constraint*. The second constraint, also following directly from the definition of the score function, states:

$$\frac{\partial}{\partial x_j} s_i(x,t) = \frac{\partial}{\partial x_i} s_j(x,t), \quad \text{for any pair } (i,j) \in \{1 \dots D\}^2 . \tag{12}$$

We refer to (12) as the *differential constraint*. Our goal is to numerically investigate wether either of the constraints are met in trained diffusion models.

2.1 Models and datasets

To present a minimal working example, we trained a MNIST diffusion model using a lightweight U-Net implementation. The model is composed of ShuffleNet-style residual bottlenecks and depth-wise convolutions. The time indices are embedded, passed through an MLP, and added to the feature maps in each block. It employs simple encoder–decoder blocks with downsampling and upsampling and skip connections, keeping the model lightweight (around 4 MB). The implementation can be found at [3]. We used the cosine schedule [14] and a total discrete time index $T = 1000$, which corresponds to observing time-homogeneous OU process (1) at discrete times [16]

$$t_k = -\frac{1}{2}\log\frac{f(k)}{f(0)}, \; f(k) := \cos\left(\frac{k/T + 0.008}{1 + 0.008}\frac{\pi}{2}\right) \tag{13}$$

We also performed the same test with latent diffusion, using a VAE with an 8×8 latent space (implementation based on [18]). The diffusion process employs the same network as before but acts in the latent space of the VAE.

The purpose of this experiment is to enable a comprehensive analysis with tractable computation, especially for evaluating the differential constraints. The results are presented in the following sections. We also observed a similar behavior for the CIFAR-10 dataset (Appendix 4.2).

2.2 Integral constraints

To numerically check the integral constraint (11), we introduce three different mechanisms for generating closed paths on which the integral is evaluated:

- **Brownian path.** Starting from a corrupted sample $x_t \in \mathbb{R}^D$ generated by the forward diffusion, we perform a random walk on $\mathbb{R}^D$ using a Brownian bridge, which generates a path in $\mathbb{R}^D$ starting and ending at x_t. The path of Brownian bridge is $X_u^{\mathrm{BB}} = W_u - uW_U/U$ with a fictitious time $u \in [0, U)$. We choose $U = 9$, uniformly sample 1,000 discrete time steps in between, and add the resulting path to a forward sample x_t, i.e., $y_{u;t} = x_t + X_u^{\mathrm{BB}}$. This method does not guarantee that the path stays close to the the typical region induced by the forward process, as illustrated in Fig. 1 (a). We include this path as a way to study the behavior of out-of-distribution samples.
- **Rotation path.** Following the typical application of image corruption process, the corrupted sample $x_t = x_0e^{-t} + \sqrt{1-e^{-2t}}\varepsilon$, where $\varepsilon \sim \mathcal{N}(0, I)$. We randomly pair each of the D

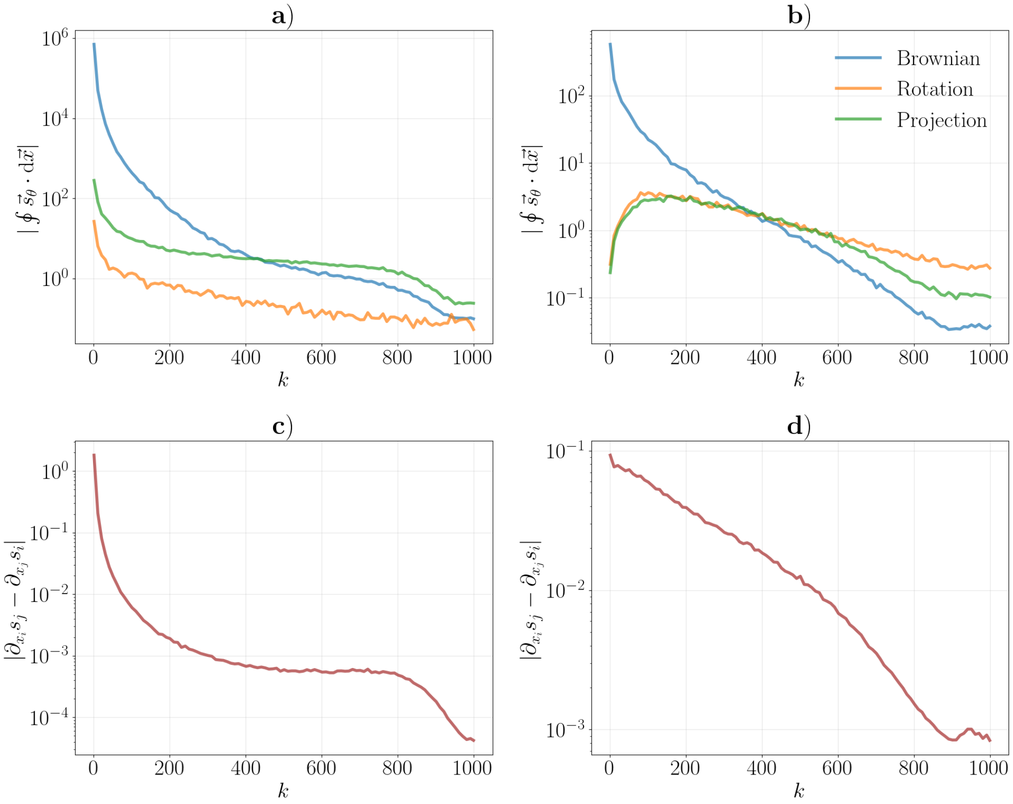


Figure 2: Results of integral and differential constraints, as functions of discrete time index k: **a)** shows the absolute value of the integral condition $\oint \vec{s}_\theta \cdot \mathrm{d}\vec{x}$; **b)** presents the same quantity but for the latent dynamics; **c)** reports the differential condition $|\partial_{x_i} s_j - \partial_{x_j} s_i|$ in normal diffusion; **d)** shows the corresponding differential condition in latent diffusion.

components of ε, so $(\varepsilon_i, \varepsilon_j)$ forms a two-dimensional vector. Then, we rotate each of the $D/2$ two-dimensional vectors with respect to the origin, i.e., $\varepsilon'_i(u) = \cos(2\pi u)\varepsilon_i + \sin(2\pi u)\varepsilon_j$ and $\varepsilon'_j(2\pi u) = -\sin(2\pi u)\varepsilon_i + \cos(2\pi u)\varepsilon_j$. Note that we rotate all $D/2$ pairs with the same "angular velocity". The resulting vector is used to generate a closed loop in the x-space, i.e., $y_{u;t} = x_0 e^{-t} + \sqrt{1-e^{-2t}}\varepsilon'(u)$, $u : 0 \to 1$. With this construct, the probability density of noise realization $\varepsilon'(u)$ is identical to that of the original noise realization ε, ensuring the closed path in the x-space sits in the region where most of the probability mass is.

- **Projection path.** We first generate multiple corrupted samples x_t from the same initial x_0, then find a way to connect these points such that the connections lie in the typical set of corrupted distribution. In order to achieve this, we propose a simple mechanism: to connect two corrupted samples x_t and x'_t, we first generate points that linearly interpolate between the two samples, and then project the interpolated points back to the corrupted distribution. Since Gaussian diffusion in high-dimensional space induces the structure of a thin shell around the clean samples, the projection can be carried out by projecting the samples radially back to the shell in $\mathbb{R}^D$, whose radius is estimated either through Monte Carlo sampling (which we also know would be $\approx \sqrt{D}$ from asymptotic analysis). An illustrative schematic diagram is provided in Fig. 1 (c).

Figures 2 (a) and (b) show the results of evaluating the integral constraint using these three methods of generating closed paths. Summary statistics of these distributions are provided in Fig. 4 in the Appendix.

Clearly, the integral condition is not satisfied in the trained neural network. One may argue whether the magnitude matters to the reverse-time dynamics. To answer this, we notice that the score-induced drift $2s(x,t)$ is added to a linear term $x(t)$ in (3); this provides us a non-dimensional quantity:

$$\frac{2\oint \vec{s}(\vec{y},t)\cdot \mathrm{d}\vec{y}}{\oint |\vec{y}|\,|\mathrm{d}\vec{y}|}, \tag{14}$$

where $\vec{y}$ is a dummy vector looping over the generated path. Results of this quantity are presented in Figs. 5 and 6 in Appendix, showing a significant deviation from 0.

2.3 Differential constraints

Due to the intensive resources required to compute the full Jacobian matrix, we instead randomly sample 64 components of the predicted score $s(x,t)$ and 64 components of the corrupted samples x_t to compute a 64×64 sub-Jacobian matrix. The statistics were collected from 256 samples for each time step, and are presented in Fig. 2 (c) and (d), both showing non-zero contributions.

3 Discussion

The numerical evidence clearly suggests that *the trained neural network* **does not** *learn the score function*, which is a conservative field. However, the trained network can definitely perform the generative task. The observation raises an interesting question: what is the trained neural network actually learning in order to perform the generative task?

We here propose a bold hypothesis, leveraging the WGF theory, to understand what happens in the "score-matching" generative modeling. Our assertion is:

> Existing diffusion modeling is better understood as modeling a normalizing flow [5], through performing flow matching [12] to the WGF velocity (9), rather than learning the reverse stochastic differential equation established by [2], popularized by [20].

Contrary to typical flow-based models [5] which learn the velocity field by maximizing the end-to-end likelihood, the flow-matching method [12] matches the neural velocity field to a target velocity field. The target velocity field is often analytically derived for a prescribed transport from the data distribution to an easy-to-sample distribution (often isotropic Gaussian distribution in high dimension), and evaluated on sampled training data. Here, we use the WGF induced by the energy functional (4) as the prescribed transport, and match the velocity field (9). More precisely, we only match the flow induced by the entropic term in (4).

There are several advantages to understand the diffusion model as the flow-matching WGF. First, the "probability flow" is naturally included in the WGF framework. Secondly, we can formally bypass the necessity to invoke the reverse-time Itô process, which can be confusing and counterintuitive—as will be seen below, within the WGF and Otto calculus framework, the deterministic probability flow ODE arises naturally, bypassing the need to explicitly route through Anderson's reverse-time SDE. Finally, flow-matching WGF naturally explains why the trained neural flow, which fails to obey the differential and integral score conditions, can still perform in generative modeling.

To see this, let us illustrate a self-consistent narrative of a flow-matching problem:

1. **Optimization objective.** Our goal is to learn (9) through flow-matching. We choose to minimize the L^2 error between the neural velocity and the entropy-induced velocity field in (9)

$$\min_{\theta} \mathbb{E}_{k\sim \text{Unif}(\{1,2\ldots T\})}\mathbb{E}_{x\sim\rho(\cdot,t)} \left\|v_\theta(\cdot,t_k) - s\left(\cdot,t_k\right)\right\|_2 \tag{15}$$

2. **Data generation.** Samples to perform Monte Carlo approximation of the above L^2-norm will be drawn from the distribution at time t, induced by the energy function (4). Instead of using the WGF in the forward dynamics, which involves estimating $\log\rho$ in high dimension, we use the equivalent OU process (1) to generate sample and more importantly, to compute *analytically exact* $s(x,t)$ for matching the neural velocity field.
3. **Sampling/Inference.** To perform the generative task, terminal samples drawn from the isotropic Gaussian are transported from $t\to\infty$ to $t=0$ by integrating the ordinary differential equation backward in time. That is, $\mathrm{d}x(\tau)/\mathrm{d}\tau = -v_{\text{WGF}}(x(\tau)) = x(\tau) + \text{NN}(x(\tau),-\tau)$, where $\tau\equiv -t$,

so signs flip relative to forward time. $x(\infty) \sim \mathcal{N}(0, I)$ and $\tau : -\infty \to 0$. The corresponding GLE [9] is

$$\frac{\partial}{\partial \tau}\rho\left(x,t\right) = -\nabla_x\left[\left(x + v_{\theta^*}\left(x,-\tau\right)\right)\rho\left(x,t\right)\right], \tag{16}$$

where θ^* stands for the trained neural weights.

Operationally, the above descriptions are identical to applying the "score-matching" for training and performing "probability flow" for inference [20]. However, because of the deterministic nature of the WGF, we would not need to invoke the reverse-time stochastic process [2]. The simplicity is the first benefit of recognizing the existing approach as a Wasserstein Gradient Flow-Matching problem.

By framing the learning as a flow-matching problem, it is most natural to weight each time equally, which is the *de facto* training procedure for both discrete-time [10] and continuous-time [20] diffusion models. The procedure would seem *ad hoc* if one aims to parameterize a neural network for learning the reverse-time diffusion process by a more theoretically grounded log-likelihood (more precisely, the bound of which) maximization as shown in [19]. As DDPM [10] pointed out, the log-likelihood approach involved weights which are not uniform in time; by removing such non-uniform weights, DDPM achieved a better performance by effectively solving a flow-matching problem.

Next, assuming that we learn the WGF perfectly, we can treat the reverse-time WGF as a dynamical system:

$$\frac{\mathrm{d}}{\mathrm{d}\tau}x(\tau) = x(\tau) + \mathrm{NN}(x(\tau),-\tau) = x(\tau) + \nabla_x \log(x(\tau),-\tau). \tag{17}$$

This system is identical to a Wasserstein Gradient Flow with the energy functional,

$$\begin{aligned} E\left\{\rho\left(\cdot,\tau\right)\right\} &= -\int \frac{x^2}{2}\rho\left(x,\tau\right)\mathrm{d}x - \int \rho\left(x,s\right)\log\rho\left(x,\tau\right)\,\mathrm{d}x \\ &= \underbrace{-\int \frac{x^2}{2}\rho\left(x,\tau\right)\mathrm{d}x - 2\int \rho\left(x,\tau\right)\log\rho\left(x,\tau\right)\,\mathrm{d}x}_{\text{Reverse-time drift}} + \underbrace{\int \rho\left(x,\tau\right)\log\rho\left(x,\tau\right)\,\mathrm{d}x}_{\text{Reverse-time diffusion}} \end{aligned} \tag{18}$$

which is equivalent to the reverse-time Itô process (3). This suggests that we would not need to invoke Anderson's seminal proof of the existence of the reverse diffusion [2] for generative task. This justifies the second advantage of the WGF framework. We remark, however, that to rigorously establish the equivalence of the forward and reverse *path measures*, Anderson's theory remains necessary. Nevertheless, because generative diffusion models only require consistency at the level of marginal densities, it is not necessary to invoke path measures in practice. We emphasize that our results concern density transport (marginals). We do not make claims about sample-path equivalence, which requires Anderson's reverse-time construction. However, the corresponding reverse-time Itô process not only can be used as a stochastic process for sampling, but also coincidentally the reverse-time process established by Anderson [2].

Finally, as suggested by our numerical analysis, the neural network is *not* learning a gradient of a scalar potential, i.e. $\mathrm{NN}(x,t) \neq s(x,t)$ for all t, both globally (because it violates the integral conditions) or locally (because it violates the differential conditions.) It is thus puzzling and challenging to analyze how the violations affect the reverse-time diffusion, and consequently the quality of the generated samples. The flow representation can bring some insight here. Suppose we use the trained, yet imperfect neural velocity field $\mathrm{NN}(x,t) \approx \nabla_x \log\rho(x,t)$. Denote the error by $e(x,t) := s(x,t) - \mathrm{NN}(x,t)$. Then, the GLE governing the distribution driven by the neural velocity field is

$$\begin{aligned} \frac{\partial}{\partial \tau}\rho(x,\tau) &= -\frac{\partial}{\partial x}\left[\left(x + \mathrm{NN}\left(x,-\tau\right)\right)\rho\left(x,-\tau\right)\right] \\ &= -\frac{\partial}{\partial x}\left[\left(x + s\left(x,-\tau\right)\right)\rho\left(x,-\tau\right)\right] + \frac{\partial}{\partial x}\left[e(x,-\tau)\rho\left(x,-\tau\right)\right] \\ &= -\frac{\partial}{\partial x}\left[\left(x + s\left(x,-\tau\right)\right)\rho\left(x,-\tau\right)\right] \\ &\quad + \left[\nabla_x \cdot e(x,-\tau) + s^T\left(x,-\tau\right)\cdot e\left(x,-\tau\right)\right]\rho(x,-\tau). \end{aligned} \tag{19}$$

Immediately, we can identify a condition that if the error field $e(x,t)$ satisfies

$$0 = \nabla_x \cdot e(x,t) + s^T\left(x,t\right)\cdot e\left(x,t\right), \tag{20}$$

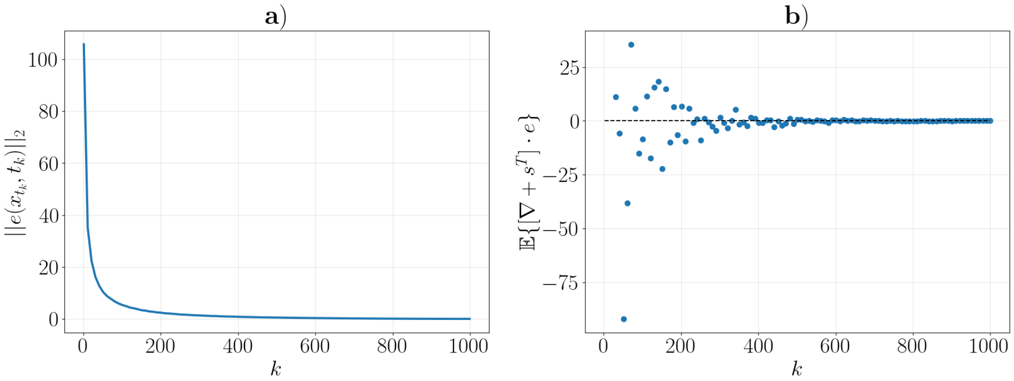


Figure 3: **a)** L2 norm of $e(x,t)$ and **b)** Stein operator value of $e(x,t)$.

the induced distribution is identical to the true distribution. In other words, if $e(x,t)$ lives in the null kernel of the operator $\nabla_x + s^T(x,t)$, the trained neural network can perfectly perform the generative task, even if it is not perfectly capturing the score function. We remark that this vector operator is related to the Stein operator [13] and is the key construct in several recent papers on sampling [4, 6, 21]. In Fig. 3, we computed the error field on a trained latent diffusion model using forward generated samples, showing that indeed a significant $e(x,t)$ is induced (which is of order 10^2, significant compared to the order 10^0 of deterministic decaying flow, $\dot{x}(t) = -x(t)$), but the error field is statistically confined[5] in the null kernel. This analysis suggests that:

> Even when $\mathrm{NN}(x,t)$ is not the score function $\nabla_x \log p(x,t)$, the trained neural network can still be effective to perform generative modeling.

We remark that this analysis is only possible by recognizing the underlying flow structure. We humbly acknowledge that we are not the first to propose the equivalence between diffusion and flow models: Song et al. [20] recognized the “probability flow/ODE” and even suggested density and likelihood estimation, and very recently, Gao et al. pointed out the resemblance between diffusion and flow models [8]. Nevertheless, to our best knowledge, there have been no studies connecting diffusion models and normalizing flow parametrized by flow matching through the elegant theory of WGF and Otto calculus. The existing theories neither connect the flow operator to the Stein operator [13]. Furthermore, the identification of a unified description between the diffusion models and WGF could inspire new forward random sampling (“data generation”) for training and regularizing flow-based models.

To conclude, we advocate for this theoretical framework because, first, it was developed over 20 years ago, and yet has been largely ignored in the machine learning literature, and second, the setup is self-consistent, simple, concise, and elegant. We dedicate this work to the pioneers of WGF theory—Jordan, Kinderlehrer, and Otto—whose foundational insights continue to shape and inspire cutting-edge machine learning research today.

References

[1] Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*. 2nd ed. Basel: Birkhäuser, 2008. ISBN: 978-3-7643-8722-8.

[2] Brian D.O. Anderson. “Reverse-time diffusion equation models”. en. In: *Stochastic Processes and their Applications* 12.3 (May 1982), pp. 313–326. ISSN: 03044149. DOI: 10.1016/0304-4149(82)90051-5.

[3] bot66. *MNISTDiffusion: Implement a MNIST (also minimal) version of denoising diffusion probabilistic model from scratch*. https://github.com/bot66/MNISTDiffusion. 2022.

[5] We averaged over 256 randomly generated forward samples x_t. For each sample, the sufficient condition does not seem to be met but the average seems to agree, noting the significant variance for small k.

[4] Peng Chen and Omar Ghattas. *Projected Stein Variational Gradient Descent*. June 2020. arXiv: `2002.03469 [cs]`.

[5] Ricky T. Q. Chen et al. *Neural Ordinary Differential Equations*. Dec. 2019. arXiv: `1806.07366 [cs, stat]`.

[6] Mingzhou Fan et al. "Path-Guided Particle-based Sampling". In: *Proceedings of the 41st International Conference on Machine Learning*. PMLR, July 2024, pp. 12916–12934.

[7] Alessio Figalli and Federico Glaudo. *An Invitation to Optimal Transport, Wasserstein Distances, and Gradient Flows*. Second edition. Berlin, Germany: EMS Press, 2023. ISBN: 978-3-98547-550-6.

[8] Ruiqi Gao et al. "Diffusion Models and Gaussian Flow Matching: Two Sides of the Same Coin". In: *The Fourth Blogpost Track at ICLR 2025*. Feb. 2025.

[9] G. Gerlich. "Die verallgemeinerte Liouville-Gleichung". en. In: *Physica* 69.2 (Nov. 1973), pp. 458–466. ISSN: 0031-8914. DOI: `10.1016/0031-8914(73)90083-9`.

[10] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. Dec. 16, 2020. arXiv: `2006.11239 [cs, stat]`.

[11] Richard Jordan, David Kinderlehrer, and Felix Otto. "The Variational Formulation of the Fokker–Planck Equation". In: *SIAM Journal on Mathematical Analysis* 29.1 (Jan. 1998), pp. 1–17. ISSN: 0036-1410. DOI: `10.1137/S0036141096303359`.

[12] Yaron Lipman et al. "Flow Matching for Generative Modeling". en. In: Sept. 2022. URL: `https://openreview.net/forum?id=PqvMRDCJT9t`.

[13] Qiang Liu and Dilin Wang. "Stein Variational Gradient Descent: A General Purpose Bayesian Inference Algorithm". In: *Advances in Neural Information Processing Systems*. Ed. by D. Lee et al. Vol. 29. Curran Associates, Inc., 2016.

[14] Alex Nichol and Prafulla Dhariwal. *Improved Denoising Diffusion Probabilistic Models*. Feb. 18, 2021. arXiv: `2102.09672 [cs, stat]`.

[15] Felix Otto. "The Geometry of Dissipative Evolution Equations: The Porous Medium Equation". In: *Communications in Partial Differential Equations* 26.1-2 (2001), pp. 101–174. DOI: `10.1081/PDE-100002243`.

[16] Javier E. Santos and Yen Ting Lin. *Understanding Denoising Diffusion Probabilistic Models and their Noise Schedules via the Ornstein–Uhlenbeck Process*. Oct. 2023. arXiv: `2311.17673 [stat,cond-mat, cs,math-ph]`.

[17] Won Seong. *Simple Latent Diffusion Model*. `https://huggingface.co/spaces/JuyeopDang/KoFace-AI`. 2024.

[18] sksq96. *A CNN Variational Autoencoder in PyTorch*. `https://github.com/sksq96/pytorch-vae/blob/master/vae.py`. 2018.

[19] Jascha Sohl-Dickstein et al. *Deep Unsupervised Learning Using Nonequilibrium Thermodynamics*. Nov. 18, 2015. arXiv: `1503.03585 [cond-mat, q-bio, stat]`.

[20] Yang Song et al. *Score-Based Generative Modeling through Stochastic Differential Equations*. Comment: ICLR 2021 (Oral). Feb. 10, 2021. arXiv: `2011.13456 [cs, stat]`.

[21] Yifeng Tian, Nishant Panda, and Yen Ting Lin. "Liouville Flow Importance Sampler". In: *Forty-First International Conference on Machine Learning*. June 2024.

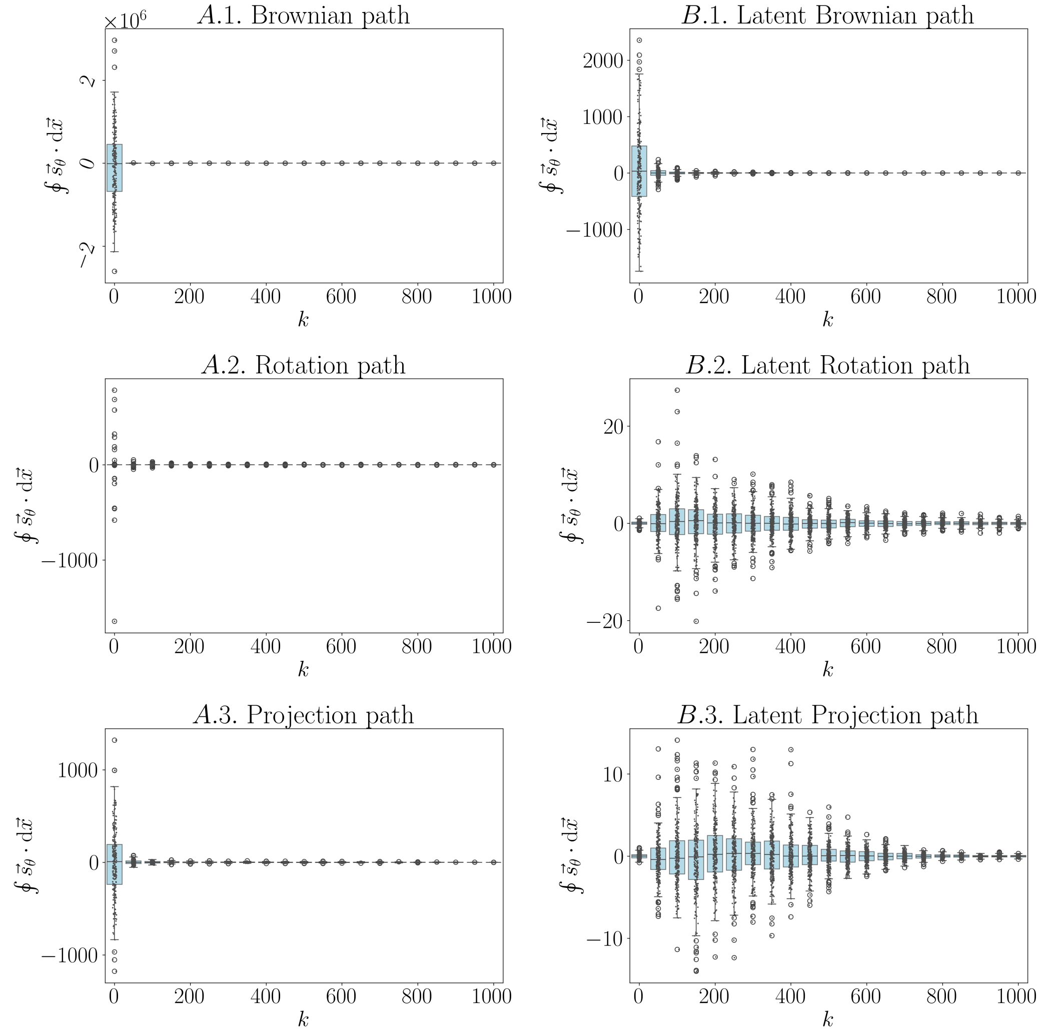


Figure 4: (MNIST) Summary statistics of $\oint \vec{s}_\theta \cdot \mathrm{d}\vec{x}$ calculated by different path-generating mechanisms, in normal and latent diffusions.

4 Appendix

We provide more statistics of the non-dimensionalized quantity $|\oint \vec{s}_\theta \mathrm{d}\vec{x}| / \oint |\vec{x}_t| \cdot |\mathrm{d}\vec{x}|$ (14), as well as experiment results on the CIFAR-10 dataset.

4.1 More numerical results on MNIST

Refer to Figs. 4, 5, 6.

4.2 Numerical results on CIFAR-10

For CIFAR-10, we utilized the models from [17], it implements the standard DDPM and VAE with latent dimension of $3 \times 16 \times 16$. We also tried training these models from scratch, which exhibits similar behaviors to the pretrained ones. Results are presented in Figs. 7, 8, 9, 10.

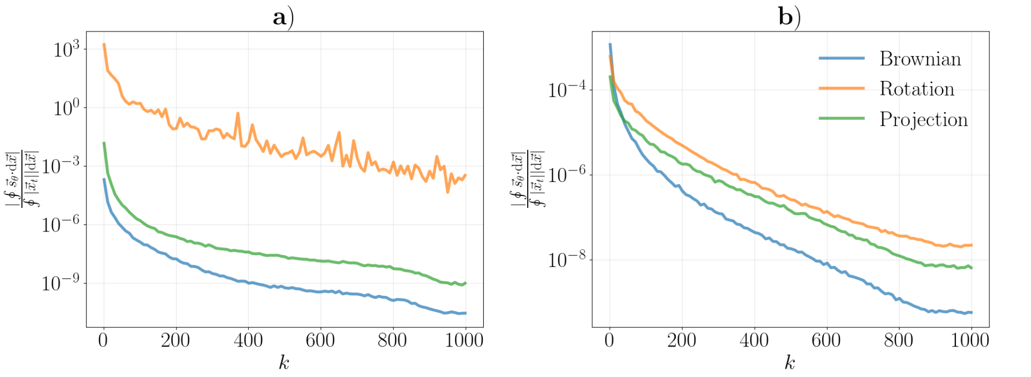


Figure 5: (MNIST) Results of integral constraints, as functions of discrete time index k: **a)** shows the absolute value of the integral condition $\oint \vec{s}_\theta \cdot \mathrm{d}\vec{x}$ normalized by the path length and the strength of the deterministic flow, $\oint |\vec{x}_t||\mathrm{d}\vec{x}|$; **b)** presents the same quantity but for the latent dynamics.

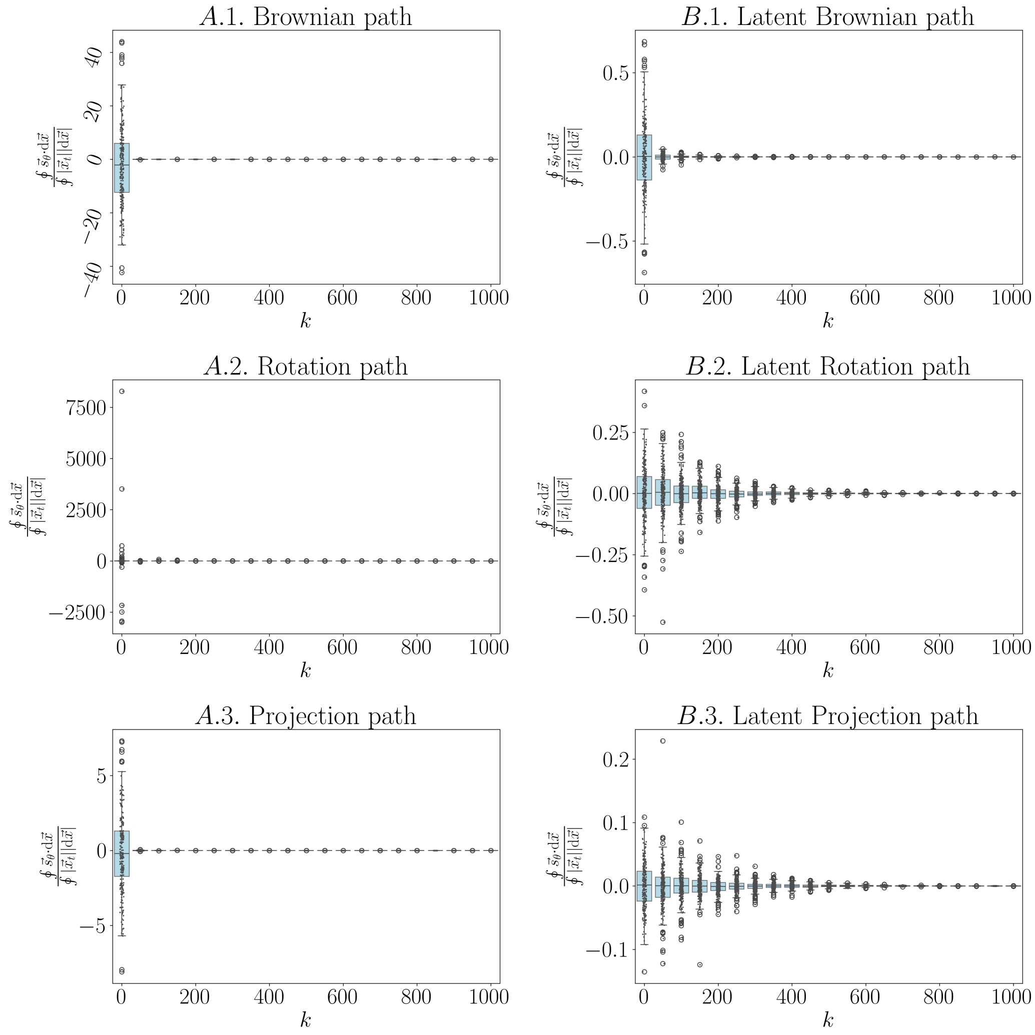


Figure 6: (MNIST) Summary statistics of $|\oint \vec{s}_\theta \cdot \mathrm{d}\vec{x}|/\oint |\vec{x}_t||\mathrm{d}\vec{x}|$ calculated by different path-generating mechanisms, in normal and latent diffusions.

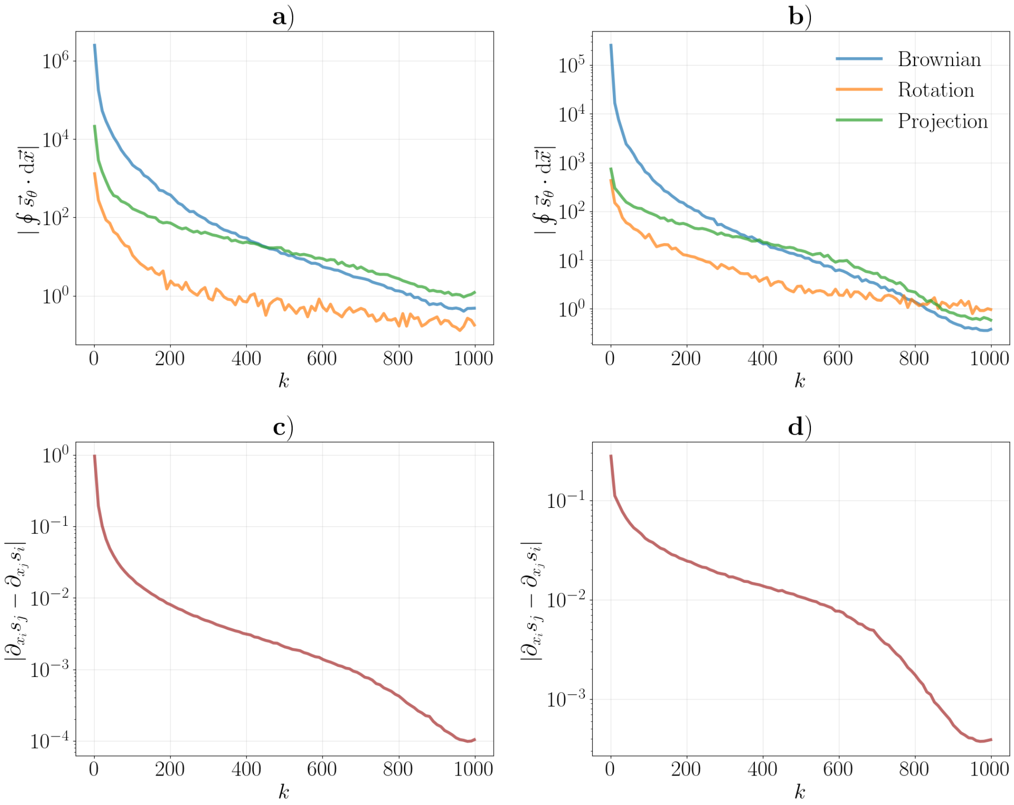


Figure 7: (CIFAR-10) Results of integral and differential constraints, as functions of discrete time index k: **a)** shows the absolute value of the integral condition $\oint \vec{s}_\theta \cdot \mathrm{d}\vec{x}$; **b)** presents the same quantity but for the latent dynamics; **c)** reports the differential condition $|\partial_{x_i} s_j - \partial_{x_j} s_i|$ in normal diffusion; **d)** shows the corresponding differential condition in latent diffusion.

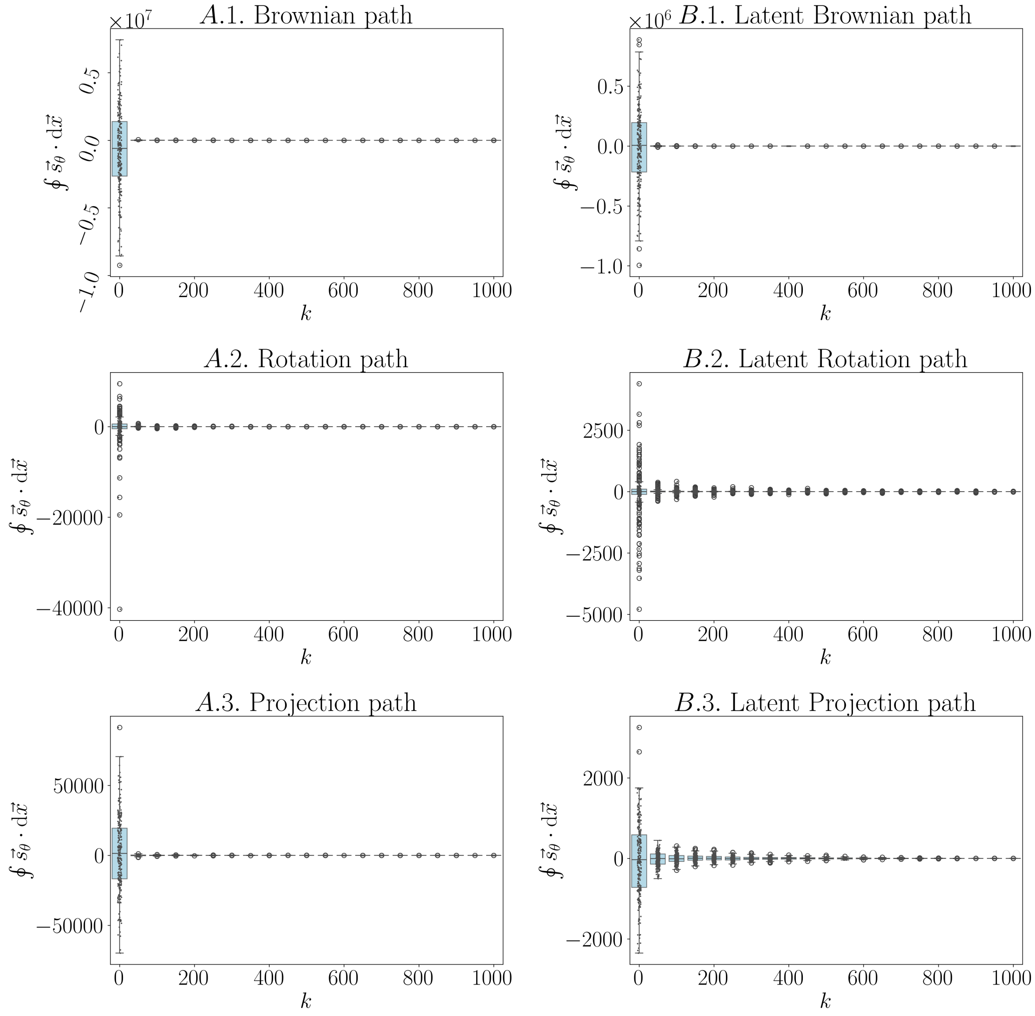


Figure 8: (CIFAR-10) Summary statistics of $\oint \vec{s}_\theta \cdot \mathrm{d}\vec{x}$ calculated by different path-generating mechanisms, in normal and latent diffusions.

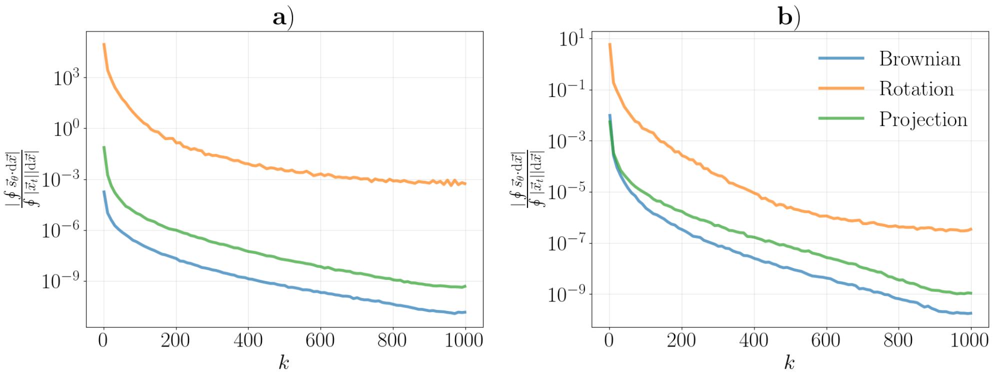


Figure 9: (CIFAR-10) Results of integral constraints, as functions of discrete time index k: **a)** shows the absolute value of the integral condition $\oint \vec{s}_\theta \cdot \mathrm{d}\vec{x}$ normalized by the path length and the strength of the deterministic flow, $\oint |\vec{x}_t||\mathrm{d}\vec{x}|$; **b)** presents the same quantity but for the latent dynamics.

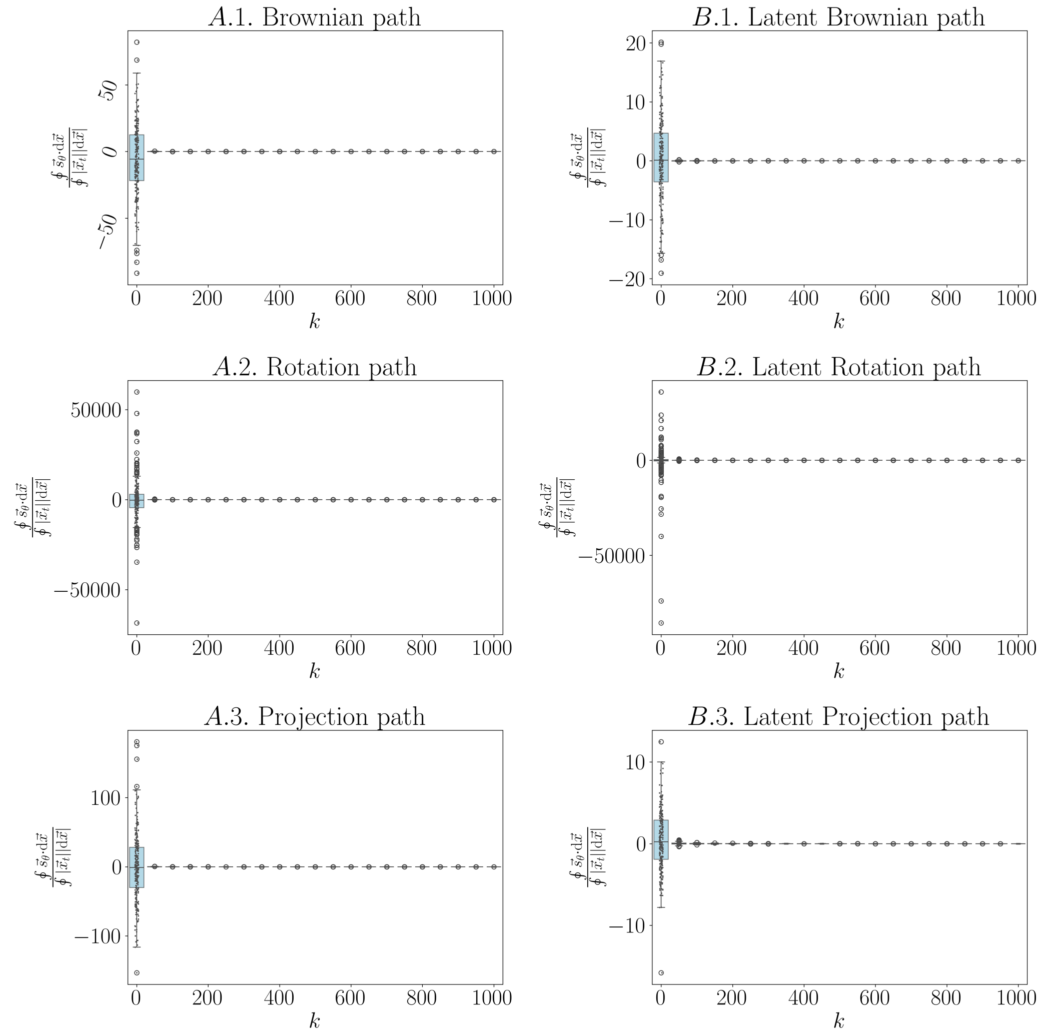


Figure 10: (CIFAR-10) Summary statistics of $|\oint \vec{s}_\theta \cdot \mathrm{d}\vec{x}| \oint |\vec{x}_t||\mathrm{d}\vec{x}|$ calculated by different path-generating mechanisms, in normal and latent diffusions.