EFFICIENT PROJECTIVE DRIFTING MODEL VIA SLICED WASSERSTEIN METRICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Drifting models offer a promising route toward one-step generation by evolving the generated distribution during training rather than solving iterative denoising or flow dynamics at inference time. Recent Sinkhorn-based methods have improved the theoretical foundation of this paradigm by entropy-regularized optimal transport, but their dependence on repeated Sinkhorn iterations introduces additional training costs and is hampered by a trade-off between generation quality and efficiency. We propose ProjDrifting, an efficient projection-based drifting model via sliced Wasserstein metric. Instead of computing high-dimensional Sinkhorn iterations, ProjDrifting reduces distribution matching to sorting-based one-dimensional optimal transport, enabling parallel-friendly drift estimation. We further develop a gradient estimator based approach to directly estimate the drifting field from the gradient of the sliced Wasserstein energy. Further, we introduce a dynamic slicing mechanism that adaptively reweights sampled projection directions according to their transport costs, thereby focusing on more discriminative directions and reducing signal dilution of uniform averaging. Theoretical analysis and experiments establish that ProjDrifting improves the quality and efficiency of one-step generative modeling.

1 INTRODUCTION

Generating high-fidelity samples from complex data distributions is a central goal of generative modeling (Esser et al., 2024; Ma et al., 2024; Ren et al., 2025). Diffusion models (Ho et al., 2020; Song et al., 2020) achieve strong results by learning reverse denoising processes, but their iterative sampling incurs high inference cost (Lu et al., 2022; Song et al.). Flow matching (Lipman et al., 2023; Tong et al., 2023) improves efficiency by learning a continuous velocity field from a simple prior to the data distribution, yet still requires multi-step numerical integration (Liu et al., 2023).

Recently, drifting models (Deng et al., 2026) have been proposed as a one-step generative paradigm. They learn a drifting field to evolve the pushforward distribution during training, such that a single forward pass maps noise to data at convergence. While this enables efficient generation, the framework may admit spurious equilibria where distribution matching implies zero drift, but zero drift does not necessarily imply distribution matching. Sinkhorn-Drifting Models (He et al., 2026) address this issue by deriving the drift from Sinkhorn divergence (Feydy et al., 2019; Genevay et al., 2018) with two-sided entropic optimal transport (OT) couplings, leading to more principled distributional matching. Nevertheless, the entropy-regularized transport plan remains approximate, and the iterative Sinkhorn solver adds training cost and limits parallelism.

To overcome these limitations, we propose ProjDrifting, a projection-based drifting framework built on sliced Wasserstein metrics Bonneel et al. (2015). By projecting high-dimensional samples onto multiple one-dimensional subspaces, ProjDrifting exploits the closed-form Wasserstein transport in 1D, avoiding both entropic approximation bias and iterative Sinkhorn normalization. This yields a non-iterative and parallel-friendly drift computation. We further construct the drifting field with a gradient estimator that directly captures transport-induced update directions from projected distributional discrepancies. Then, an instance of ProjDrifting, denoted as ProjDrifting-A is proposed which retains the distributional guidance of OT-based drifting while offering a simpler and more efficient alternative for one-step generative modeling. However, uniform slicing dilutes the information-rich transmitted signal due to the presence of redundant or uninformative projections. We therefore pro-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

pose a variant of ProjDrifting via introducing dynamic slicing strategy Nguyen & Ho (2023), which adaptively emphasizes projections with more significant distributional discrepancies. This retains the non-entropic, closed-form benefits of one-dimensional Wasserstein matching while producing stronger drift fields. We also consider a non-smooth version of ProjDrifting-M, which only uses the slice with the highest energy information.

In summary, our contributions are as follows: (1) We propose ProjDrifting, an efficient drifting framework based on sliced Wasserstein metrics, which fully leverages the parallelizability of slice computation to achieve more accurate and efficient drift field calculations. (2) We construct a gradient estimator based drifting field that directly estimates the negative gradient of the sliced Wasserstein energy, enabling parallel-friendly distribution evolution, based on which an instance of ProjDrifting is proposed, denoted as ProjDrifting-A. (3) To mitigate the signal dilution of uniform slicing, we propose a variant ProjDrifting-D that introduces a dynamic projection mechanism to adaptively emphasize informative slices. The non-smooth version ProjDrifting-M is also considered. (4) Theoretical analysis and extensive experiments demonstrate the efficiency, stability, and effectiveness of ProjDrifting for one-step generative modeling.

2 Related Work

2.1 Generative Models

Diffusion models learn generative distributions by reversing a gradual noising process Ho et al. (2020); Song et al. (2020). Given $\mathbf{x}_0 \sim p_{\text{data}}$, the forward process perturbs data into Gaussian noise, e.g., $q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$, and the model is typically trained to predict the injected noise from the noisy sample. Sampling then recursively applies the reverse transition $\mathbf{x}_{t-1} = \Phi_\theta(\mathbf{x}_t, t)$ over many time steps, which leads to high inference cost. Flow matching learns a continuous velocity field that transports samples from a simple prior to the data distribution (Lipman et al., 2023). Given a probability path $\{p_t\}_{t\in[0,1]}$, the model matches a neural velocity field $v_\theta(\mathbf{x}_t, t)$ to the target field $u_t(\mathbf{x}_t)$ using a regression objective such as $\mathbb{E}_{t,\mathbf{x}_t}\|v_\theta(\mathbf{x}_t,t) - u_t(\mathbf{x}_t)\|_2^2$. Generation proceeds by integrating the learned ODE. Extensions such as rectified flow, stochastic interpolants, and optimal-transport conditional flow matching improve the construction of probability paths and target vector fields (Liu et al., 2023; Albergo et al., 2025; Tong et al., 2024). However, high-quality generation still commonly depends on multi-step numerical integration.

Drifting models (Deng et al., 2026) enable one-step generation by evolving the generated distribution toward the data distribution during training, avoiding iterative denoising or flow integration at inference time. Sinkhorn-Drifting (He et al., 2026) further introduces entropy-regularized optimal transport to construct more principled drift fields. However, repeated Sinkhorn iterations increase training costs, and the entropic regularization strength induces a trade-off between generation quality and efficiency.

2.2 Wasserstein Distance

Wasserstein distance measures distributional discrepancy by seeking an optimal transport plan with minimum cost (Villani et al.; Peyré & Cuturi, 2019). For two distributions p and q, it can be written:

$$\mathbb{W}(p,q) = \inf_{\pi\in\Pi(p,q)} \int_{\mathcal{X}\times\mathcal{Y}} c(\mathbf{x},\mathbf{y})\, d\pi(\mathbf{x},\mathbf{y}), \tag{1}$$

where $\Pi(p,q)$ denotes the set of joint distribution over $\mathcal{X}\times\mathcal{Y}$ whose marginals are p and q, and $c(\mathbf{x},\mathbf{y})$ is the transportation cost between $\mathbf{x}$ and $\mathbf{y}$. Wasserstein distance captures the geometry of the sample space and has been widely used in generative modeling (Mescheder et al., 2018; Gulrajani et al., 2017). However, solving high-dimensional OT is computationally expensive due to the optimization over couplings. Entropic regularization enables efficient Sinkhorn approximations (Cuturi, 2013), but introduces approximation bias and iterative normalization. Entropic optimal transport introduces an additional KL regularization term:

$$\mathbb{W}_\tau(p,q) = \inf_{\pi\in\Pi(p,q)} \int_{\mathcal{X}\times\mathcal{Y}} c(\mathbf{x},\mathbf{y})\, d\pi(\mathbf{x},\mathbf{y}) + \tau\mathrm{KL}\big(\pi \,\|\, p\otimes q\big), \tag{2}$$

where τ controls the strength of entropic regularization, and $p \otimes q$ denotes the independent coupling between p and q. To remove the entropic self-similarity bias, the Sinkhorn divergence is commonly defined as $S_\tau(p,q) = \mathbb{W}_\tau(p,q) - \frac{1}{2}\mathbb{W}_\tau(p,p) - \frac{1}{2}\mathbb{W}_\tau(q,q)$.

3 METHODOLOGY

3.1 A REVISIT OF DRIFTING MODELS

Drifting Models shift the iterative evolution of generative distributions from inference time to training time, enabling one-step generation Deng et al. (2026). Given a generator G_ϕ and latent noise $\mathbf{z} \sim p_0$, the generated sample is defined as $\mathbf{x} = G_\phi(\mathbf{z})$, with distribution q_ϕ. Rather than simulating reverse diffusion or solving an ODE at inference, Drifting Models update q_ϕ during training through a drift field that moves generated samples toward the data distribution p_{data}. Upon convergence, sampling requires only a single generator evaluation. For simplicity, let $p = p_{\text{data}}$ and $q = q_\phi$. A typical drifting field follows a cross-minus-self structure. For a generated sample $\mathbf{x}$, it can be written as $V_{p,q}(\mathbf{x}) = A_p(\mathbf{x}) - A_q(\mathbf{x})$, where $A_p(\mathbf{x})$ attracts samples toward the data distribution and $A_q(\mathbf{x})$ repels them according to the current generated distribution. With a kernel $k_\tau(\mathbf{x},\mathbf{y}) = \exp\left(-\frac{1}{\tau}\|\mathbf{x}-\mathbf{y}\|\right)$, these terms can be expressed as $A_\mu(\mathbf{x}) = \frac{\int k_\tau(\mathbf{x},\mathbf{y})(\mathbf{y}-\mathbf{x})\,d\mu(\mathbf{y})}{\int k_\tau(\mathbf{x},\mathbf{y})\,d\mu(\mathbf{y})}$, where μ can be either p or q. The generator follows the drift direction through a stop-gradient regression objective:

$$\mathcal{L}_{\text{drift}} = \mathbb{E}_{\mathbf{z}\sim p_0}\left[\left\|G_\phi(\mathbf{z}) - \text{sg}\left(G_\phi(\mathbf{z}) + \eta V_{p,q}(G_\phi(\mathbf{z}))\right)\right\|_2^2\right], \tag{3}$$

where η is the drift step size and $\text{sg}(\cdot)$ denotes stop-gradient. This shifts distribution evolution to training, enabling one-step inference with a single generator forward pass.

The original drifting framework relies on one-sided normalized Gibbs kernels to construct the drifting field which lacks two-sided marginal constraints. Thus, a vanishing drift field does not necessarily imply distributional matching. Sinkhorn-Drifting Model (He et al., 2026) mitigate this issue by formulating drifting dynamics via Sinkhorn divergence, where one-sided kernels are replaced by entropy-regularized OT coupling π^* from two-sided Sinkhorn scaling, where π^* is the optimal coupling that leads to minimum Sinkhorn divergence $S_\tau(p,q)$. However, the entropic approximation and iterative Sinkhorn solver introduce extra training cost and limit parallelism.

3.2 FORMULATING DRIFTING MODEL IN SLICED WASSERSTEIN SPACE

In Sinkhorn-Drifting He et al. (2026), the drifting field $V_{p,q}$ is derived from the energy functional of Sinkhorn divergence $\mathcal{F}_\tau(q) = S_\tau(p,q)$, where p denotes the target data distribution and q denotes the current model distribution induced by the generator G_ϕ. Then the drifting field is given by:

$$\nabla_{x^i}\mathcal{F}_\tau(q_X) = \sum_{j=1}^{n}(\pi^\infty_{XY})_{ij}(x^i - y^j) - \sum_{j=1}^{n}(\pi^\infty_{XX})_{ij}(x^i - x^j) = -V_{\hat{p},\hat{q}_X}(x^i). \tag{4}$$

where $X = \{x^i\}_{i=1}^n \sim q$ denote generated samples and $Y = \{y^j\}_{j=1}^n \sim p$ denote real samples. π^∞_{XY} and π^∞_{XX} denote the Sinkhorn-scaled couplings for the cross distribution pair (X,Y) and the self pair (X,X), respectively. Despite its theoretical appeal, Sinkhorn-Drifting relies on approximate and iterative transport estimation. Its couplings π^∞_{XY} and π^∞_{XX} are entropy-regularized OT plans, yielding smoothed transport directions rather than exact Wasserstein transport. Since these couplings have no closed-form solution, they must be computed by iterative Sinkhorn normalization, which adds training cost and limits parallelism due to its sequential scaling updates.

To overcome these limitations, we propose to formulate drifting with sliced Wasserstein metrics Bonneel et al. (2015). Rather than estimating high-dimensional entropy-regularized OT couplings, we project samples onto one-dimensional subspaces, where empirical OT can be solved exactly. For a projection direction $\theta \sim \mathcal{U}(\mathbb{S}^{d-1})$, let $\theta_\# p$ and $\theta_\# q$ denote the projected distributions under $x \mapsto \theta^\top x$. The sliced Wasserstein objective is $SW(p,q) = \mathbb{E}_{\theta\sim\mathcal{U}(\mathbb{S}^{d-1})}\left[\mathbb{W}_{1d}\left(\theta_\# p, \theta_\# q\right)\right]$. Given generated samples $X = \{x^i\}_{i=1}^n \sim q$ and real samples $Y = \{y^j\}_{j=1}^n \sim p$, define their

projected values as $a_i^\theta = \theta^\top x^i$ and $b_j^\theta = \theta^\top y^j$. Then the empirical one-dimensional Wasserstein distance is:

$$\mathbb{W}_{1d}\left(\theta_{\#}\hat{q}_X, \theta_{\#}\hat{p}_Y\right) = \frac{1}{n}\sum_{r=1}^{n}\left\|a^\theta_{\sigma_X^\theta(r)} - b^\theta_{\sigma_Y^\theta(r)}\right\|^2. \tag{5}$$

where σ_X^θ and σ_Y^θ denote the sorting permutations of $\{a_i^\theta\}_{i=1}^n$ and $\{b_j^\theta\}_{j=1}^n$ in ascending order. Each projected OT subproblem can be solved exactly without entropy regularization or iterative Sinkhorn normalization, and different projections can be computed in parallel. The energy functional with sliced Wasserstein metric can be written as:

$$\mathcal{F}_s(q) = SW(p,q) = \mathbb{E}_{\theta\sim\mathcal{U}(\mathbb{S}^{d-1})}\left[\mathbb{W}_{1d}\left(\theta_{\#}p, \theta_{\#}q\right)\right]. \tag{6}$$

Unlike Sinkhorn-Drifting, which minimizes the energy $\mathcal{F}_\tau(q)$ with entropy regularization, we compute the energy from $\mathbb{W}_{1d}\left(\theta_{\#}\hat{q}_X, \theta_{\#}\hat{p}_Y\right)$ over random projections. Each projected OT subproblem can be solved exactly by sorting-based monotone matching, without regularization or iterations. The insight here is that different projections can be evaluated independently and be computed in parallel.

To build an effective drifting model based on $\mathcal{F}_s(q)$, Section 3.3 introduces a gradient-estimator-based drift field that estimates transport directions from projected Wasserstein discrepancies. Section 3.4 further proposes an variant via dynamic projection strategy for better generation.

3.3 Gradient Estimator based Drifting Field

Drifting field of ProjDrifting. Let $X = \{x^i\}_{i=1}^n \sim q$ and $Y = \{y^j\}_{j=1}^n \sim p$ denote generated and real samples. Based on Eq.(6), we instantiate the mini-batch energy as $\mathcal{F}_s(q_X) = SW(\hat{p}, \hat{q}_X)$, and define the drifting field by the negative particle gradient $V_{\hat{p},\hat{q}_X}(x^i) = -\nabla_{x^i}\mathcal{F}_s(q_X)$. Compared with Sinkhorn-Drifting, which depends on iterative Sinkhorn normalization for entropy-regularized couplings, ProjDrifting reduces transport estimation to independent one-dimensional OT problems solved exactly by sorting-based monotone matching. This allows us to directly estimate the negative particle gradient of $\mathcal{F}_s(q_X)$ and construct the drifting field efficiently. Since the projected one-dimensional OT problems are independent across projection directions, the computation can be naturally parallelized. The resulting gradient-based drifting-field solver is summarized in Algorithm 1. We have Proposition 1 that the drifting field of ProjDrifting is zero when $\theta_{\ell\#}\hat{q}_X = \theta_{\ell\#}\hat{p}$.

Proposition 1. *If the empirical generated distribution matches the empirical data distribution, i.e., $\hat{q}_X = \hat{p}$, then the drifting field of ProjDrifting vanishes:*

$$\widehat{V}_{\hat{p},\hat{q}_X}(x^i) = 0, \qquad \forall i \in \{1, ..., n\}. \tag{7}$$

Proof. When $\hat{q}_X = \hat{p}$, their projected distributions are identical for any projection direction θ_ℓ, namely $\theta_{\ell\#}\hat{q}_X = \theta_{\ell\#}\hat{p}$. Thus, under the closed-form one-dimensional matching in Eq.(5), the sorted projected generated samples and real samples coincide. For each generated sample x^i, its matched real projected value satisfies $\tilde{b}_i^\ell = a_i^\ell$, and the projected displacement is zero, i.e., $d_i^\ell = \tilde{b}_i^\ell - a_i^\ell = 0$. Since this holds for every θ_ℓ, the aggregated sliced Wasserstein drifting field also vanishes:

$$\widehat{V}_{\hat{p},\hat{q}_X}(x^i) = \frac{1}{L}\sum_{\ell=1}^{L} d_i^\ell \theta_\ell = 0. \tag{8}$$

This proves that the drifting field of ProjDrifting is zero when $\theta_{\ell\#}\hat{q}_X = \theta_{\ell\#}\hat{p}$. □

Additionally, we have the proposition and proof in Appendix B.5 that the proposed ProjDrifting share the same property of equilibrium consistency as Sinkhorn-Drifting He et al. (2026). That is when drifting filed equals to zero, the generated distribution equals to the target distribution.

Classifier-Free Guidance. We further extend ProjDrifting to conditional generation. Let c denote the condition, e.g., class labels. Given latent noise $z \sim p_z$, the conditional generator produces samples as $x = f_\phi(z, c)$. For controllability, we adopt a classifier-free guidance strategy. For each condition c, we draw positive samples $\mathbf{y}^+$ from the conditional data distribution $p_{\text{data}}(\cdot \mid c)$ and negative samples $\mathbf{y}^-$ from the unconditional data distribution $p_{\text{data}}(\cdot \mid \varnothing)$. The positive samples provide condition-specific distributional guidance, while the negative samples characterize the unconditional data distribution. in the framework of ProjDrifting, we compute the conditional and unconditional

Algorithm 1 Dynamic Sliced Drifting Field

Require: $X \in \mathbb{R}^{N\times d}$, $Y \in \mathbb{R}^{M\times d}$, number of slices L, temperature β, constant τ
Ensure: Drift field $V \in \mathbb{R}^{N\times d}$

```
Sample directions Θ ∈ R^{d×L} from N(0, I).
Normalize Θ_{:,ℓ} ← Θ_{:,ℓ} / ‖Θ_{:,ℓ}‖_2 for ℓ = 1, ..., L.
Project X_Θ ← XΘ, where X_Θ ∈ R^{N×L}.
Project Y_Θ ← YΘ, where Y_Θ ∈ R^{M×L}.
Sort X̄_Θ ← Sort(X_Θ, dim = 0).
Sort Ȳ_Θ ← Sort(Y_Θ, dim = 0).
Compute slice costs d ← W_1D(X̄_Θ, Ȳ_Θ), where d ∈ R^L.
Normalize costs d̃ ← d / (Σ_{ℓ=1}^{L} d_ℓ + τ).
Compute weights w ← Softmax(β d̃), where w ∈ R^L.
Aggregate distance D_DS ← Σ_{ℓ=1}^{L} w_ℓ d_ℓ.
Compute drift V ← −∇_X D_DS.
return V
```

drift fields as $V^+(x) = V_{\hat{p}^+,\hat{q}_X}(x)$ and $V^-(x) = V_{\hat{p}^-,\hat{q}_X}(x)$, where $\hat{p}^+$ and $\hat{p}^-$ are the empirical distributions of $\{\mathbf{y}^+\}$ and $\{\mathbf{y}^-\}$, respectively. Following classifier-free guidance, we combine the two drift fields as $V_{\text{cfg}}(x \mid c) = (1+\omega)V^+(x) - \omega V^-(x)$, where ω is the guidance scale.

Given the guided drifting field, the conditional drifted target is defined as $\tilde{x} = x + \eta V_{\text{cfg}}(x \mid c)$, where η is the drift step size. The conditional generator is optimized by:

$$\mathcal{L}_{\text{cfg}} = \mathbb{E}_{z,c}\left[\left\| f_\phi(z,c) - \text{sg}\left(f_\phi(z,c) + \eta V_{\text{cfg}}\left(f_\phi(z,c) \mid c\right)\right)\right\|_2^2\right]. \tag{9}$$

3.4 Dynamic Projection

The ProjDrifting in Section 3.3 (denoted as ProjDrifting-A for distinction) averages projected transport directions uniformly. Although efficient and parallel-friendly, it ignores the varying informativeness of different projections. Directions with strong discrepancies may be mixed with redundant or uninformative ones, thereby diluting transport signals and weakening the resulting drift field. To address this issue, we introduce a dynamic projection strategy Nguyen & Ho (2023) that adaptively assigns larger weights to more informative projection directions and propose another variant of ProjDrifting, i.e., ProjDrifting-D. Given L projection directions $\{\theta_l\}_{l=1}^L$, the energy functional is:

$$\mathcal{F}_d(q) = \text{EBSW}(p,q) = \sum_{l=1}^{L} \widehat{w}_{p,q}(\theta_l)\, \mathbb{W}_{1d}\left(\theta_{l\#}p, \theta_{l\#}q\right), \tag{10}$$

where $\hat{w}_{p,q}(\theta_l) = \frac{w_{p,q}(\theta_l)}{\sum_{l'=1}^{L} w_{p,q}(\theta_{l'})}$. $w_{p,q}(\theta_l)$ measures the informativeness of the projection direction θ_l. Specifically, we use an abstract scoring function $w_{p,q}(\theta_l) = \Psi(\theta_l; \hat{p}, \hat{q})$ to measure the importance of each projection, and normalize it as $\widehat{w}_{p,q}(\theta_l) = w_{p,q}(\theta_l) / \sum_{l'=1}^{L} w_{p,q}(\theta_{l'})$. This allows the projective drifting model to focus more on informative slices without assuming a fixed weighting form. Based on $\mathcal{F}_d(q)$, we construct the dynamic sliced drifting field:

$$\widehat{V}_{\text{dyn}}(x^i) = \sum_{l=1}^{L} \widehat{w}_{p,q}(\theta_l) v_{\theta_l}(x^i), \tag{11}$$

where $v_{\theta_l}(x^i)$ is the projected drifting direction induced by the one-dimensional OT matching along θ_l. Compared with the uniformly averaged field, the dynamic field emphasizes projections that contain stronger transport signals, thereby reducing the dilution caused by redundant or weak slices.

Proposition 2. *If the empirical generated distribution matches the empirical data distribution, i.e., $\hat{q}_X = \hat{p}$, then the dynamic-projection drifting field of ProjDrifting vanishes:*

$$\widehat{V}_{\text{dyn}}(x^i) = 0, \qquad \forall i \in \{1, ..., n\}. \tag{12}$$

This property confirms that dynamic projection preserves the fixed point of the sliced Wasserstein drifting field. When the generated and data distributions match, all projected displacements vanish, and dynamic weighting adds no extra drift. The proof is given in Appendix B.2.

We consider a non-smooth but more efficient variant ProjDrifting-M inspired by Max-SW Deshpande et al. (2019). Instead of weighting all projection directions, it selects the most informative:

$$\theta^\star = \arg \max_{\theta_l \in \{\theta_l\}_{l=1}^L} \mathbb{W}_{1d}\left(\theta_{l\#}p, \theta_{l\#}q\right). \tag{13}$$

The energy functional of ProjDrifting-M is written as follows:

$$\mathcal{F}_m(q) = \text{Max-SW}(p, q) = \max_{\theta_l \in \{\theta_l\}_{l=1}^L} \mathbb{W}_{1d}\left(\theta_{l\#}p, \theta_{l\#}q\right). \tag{14}$$

Using this direction, the drifting field can be approximated as $\widehat{V}_{\max}(x^i) = v_{\theta^\star}(x^i)$. This strategy avoids aggregating drift directions over all projections and focuses the update on the most discriminative slice. Although the max operation is non-smooth and may be less stable than the soft dynamic weighting, it may provide a more efficient alternative when computational resources are prioritized.

Efficiency analysis. For a mini-batch of size n in dimension d, Sinkhorn-Drifting performs T Sinkhorn steps, with time complexity $\mathcal{O}(n^2 d + Tn^2)$. In contrast, ProjDrifting uses L one-dimensional projections and solves each projected problem exactly by sorting, yielding complexity $\mathcal{O}(Lnd + Ln\log n)$. Since the projected OT problems are solved exactly by monotone matching, require no entropy-regularization or iterative normalization, and are independent across directions, which is more parallel-friendly. When $L(d + \log n) \ll n(d + T)$, ProjDrifting provides a more efficient drifting-field estimator. A detailed derivation is given in Proposition 3 and its proof in Appendix B.3. Moreover, the finite-projection estimator used in ProjDrifting is a Monte Carlo approximation of the exact sliced Wasserstein energy. As the number of projections increases, it converges to the exact solution. The proposition and proof are provided in Appendix B.4.

4 Experiments

4.1 Experimental Setup

Datasets. To evaluate the effectiveness of our method, we conduct experiments on synthetic data, the MNIST handwritten digit dataset LeCun et al. (2002), the FFHQ dataset Karras et al. (2019), and the ImageNet 256×256 dataset Deng et al. (2009). The experiments on these datasets cover different scales of raw data and feature dimensions, with detailed information summarized in Table 1.

Table 1: Raw data dimensions and feature dimensions of experiments on different datasets.

Dataset	Raw Dim.	Feature Dim.
Synthetic	2	2
MNIST	28×28	256
FFHQ	$1024 \times 1024 \times 3$	512
ImageNet	$256 \times 256 \times 3$	512

Evaluation Metrics. We evaluate distributional closeness using the 2-Wasserstein distance (W_2), where lower values indicate better alignment between the generated and target distributions. For image generation, we adopt the Fréchet Inception Distance (FID) to assess visual generation quality, with lower FID scores indicating higher-quality generated images. To measure computational efficiency, we report the average per-iteration training time, denoted as T_{avg}. The implementation details can be found in the corresponding experimental sections and Appendix A.

4.2 Main Results

Toy Experiments. We evaluate different models on synthetic 2D datasets, mapping a standard Gaussian source $\mathcal{N}(\mathbf{0}, \mathbf{I}_2)$ to either an 8-Gaussians or 2-Moons target distribution. All models are trained with Adam optimizer Kingma & Ba (2014) for 5000 iterations, with batch size 500 and learning rate 10^{-3}. For Drifting and Sinkhorn-Drifting, we set τ to 0.01 and 0.1, respectively. The generator is a two-layer MLP with a latent dimension 256. Figure 1 shows the final generated distributions and the training curves of $\mathbb{W}_2^2$. Both Drifting and Sinkhorn-Drifting are sensitive to τ, which controls the Gibbs kernel or entropic regularization and strongly affects training stability. For Drifting, a small τ concentrates the Gibbs kernel mass on the nearest neighbors, leading to insufficient global balancing.

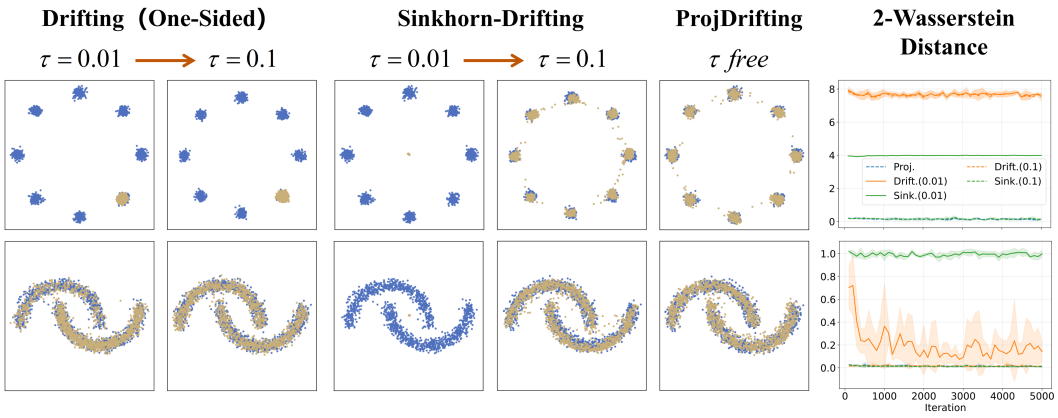


Figure 1: Unconditional generation results of different models on synthetic datasets. **The left five columns** show the final generated samples under the corresponding settings (orange), together with samples drawn from the target distribution (blue). **The rightmost column** presents the evolution of the $\mathbb{W}_2^2$ over training iterations for different methods under the same target distribution.

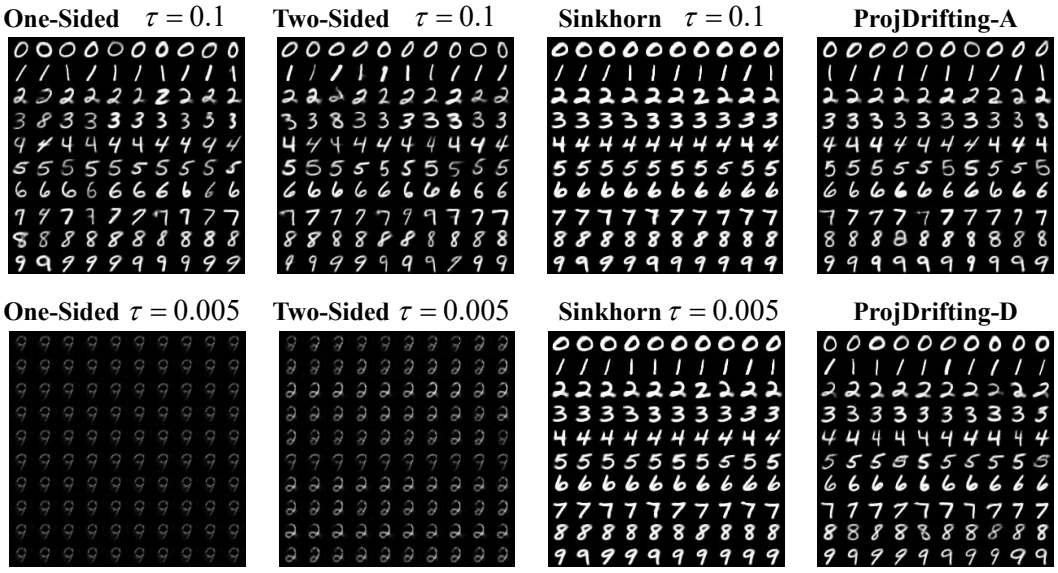


Figure 2: Final generated handwritten digit images of different generative models on the MNIST dataset under different settings. For each method, we traverse all 10 classes and generate 10 samples for each class to evaluate the accuracy and diversity of the generated handwritten digits.

This issue is particularly evident when the target distribution is 8-Gaussians. A similar phenomenon can also be observed in Sinkhorn-Drifting. However, an excessively large τ causes the Sinkhorn solution to approach a nearly uniform transport plan. In contrast, ProjDrifting-D does not rely on τ. While maintaining convergence and training stability, the generator trained with ProjDrifting-D produces samples that better fit the target distribution. More results are provided in Appendix A.2.

MNIST. We evaluate class-conditional generation on MNIST (LeCun et al., 2002) using a three-layer MLP as the generator trained for 5000 iterations with learning rate 2×10^{-4}. Table 2 reports the average EMD, i.e., the squared 2-Wasserstein distance, and conditional generation accuracy for ProjDrifting-A, ProjDrifting-D, and baselines. For Drifting and Sinkhorn-Drifting, we evaluate their performance under $\tau = 0.005$ and $\tau = 0.1$. Generated samples are shown in Figure 2. The Drifting model, including both its one-sided and two-sided variants, fails when $\tau = 0.005$. Although Sinkhorn-Drifting does not exhibit obvious mode collapse, the EMD of generated latent and image are both worse than those of ProjDrifting-A and ProjDrifting-D. Among all methods, ProjDrifting-D achieves the best generated image quality. The samples generated by the two-sided Drifting and Sinkhorn-Drifting lack sufficient diversity, as samples from the same class appear visually similar.

Table 2: For the conditional generation experiments on the MNIST dataset, we report the EMD of the generated latents and the EMD of the handwritten digit images produced by the decoder. Accuracy measures the class accuracy of the generated samples with respect to their target labels.

Metric	Drifting (One-Sided)		Drifting (Two-Sided)		Sinkhorn-Drifting		ProjDrifting-A	ProjDrifting-D
	$\tau = 0.005$	$\tau = 0.10$	$\tau = 0.005$	$\tau = 0.10$	$\tau = 0.005$	$\tau = 0.10$	τ free	τ free
EMD (Latent) ↓	67.04	6.55	67.90	6.38	5.99	6.20	5.27	**3.98**
EMD (Image) ↓	74.54	25.84	68.78	25.91	25.89	25.86	25.64	**21.87**
Accuracy ↑	10.00	97.52	9.91	97.51	99.99	99.99	99.99	**99.99**

Table 3: Class-conditional face generation results of different generative models on FFHQ. We report FID↓ for each gender-age subgroup and the overall FID. The best results are highlighted.

Method	Male (FID↓)			Female (FID↓)			Average FID↓
	Children	Adult	Old	Children	Adult	Old	
Drifting ($\tau = 0.1$)	227.54	199.11	219.64	171.28	142.17	144.89	184.10
Drifting ($\tau = 1.0$)	220.65	167.46	151.15	162.57	132.45	100.30	155.76
Drifting ($\tau = 10.0$)	42.24	40.54	36.36	47.97	54.87	38.62	43.43
Sinkhorn-Drifting ($\tau = 0.1$)	36.59	35.29	29.99	40.21	43.31	80.85	44.37
Sinkhorn-Drifting ($\tau = 1.0$)	33.96	34.31	28.15	34.37	36.33	31.50	33.10
Sinkhorn-Drifting ($\tau = 10.0$)	34.32	36.06	29.59	34.85	36.47	31.85	33.86
ProjDrifting-M	64.14	61.08	47.18	60.45	64.54	59.67	59.51
ProjDrifting-A	31.33	28.87	23.10	30.87	31.90	28.00	29.01
ProjDrifting-D	**28.80**	**25.95**	**20.88**	**27.47**	**28.16**	**25.51**	**26.13**

FFHQ. FFHQ Karras et al. (2019) is a high-quality human face dataset containing diverse portrait images with large variations in age, gender, and appearance. We conduct conditional image generation experiments on FFHQ based on a pretrained Adversarial Latent Autoencoder (ALAE) Pidhorskyi et al. (2020). Our generator takes as input a $(512 + 64)$-dimensional vector, consisting of a noise vector and a label embedding, and outputs a 512-dimensional feature vector, i.e., $G : \mathbb{R}^{512+64} \to \mathbb{R}^{512}$. The generator is parameterized by a three-layer MLP with hidden width 1024. Moreover, we observe that, under more complex distributions and higher-dimensional feature spaces, the Drifting model tends to suffer from mode collapse when $\tau < 1$. Therefore, in Table 3, we report the performance of the Drifting model and Sinkhorn-Drifting for $\tau \in \{0.1, 1.0, 10.0\}$. The Drifting model suffers from mode collapse when $\tau = 1.0$ or $\tau = 0.1$, as shown in Figure 3. Our method outperforms baselines in terms of both the FID of each subclass and the average FID.

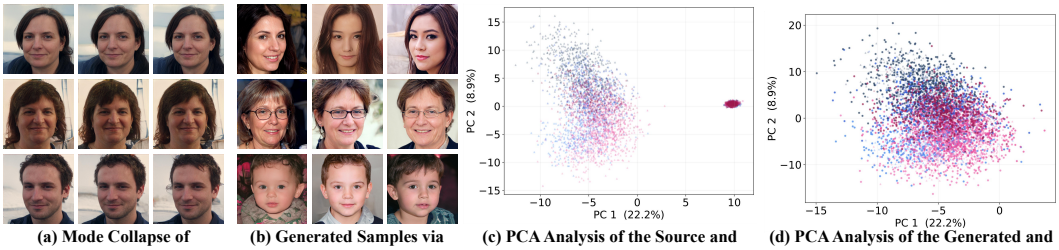
ImageNet 256×256. To verify the advantage of ProjDrifting at large scale, We evaluated the ProjDrifting and the baseline on the ImageNet 256×256 dataset Deng et al. (2009). Please refer to Appendix for implementation details and evaluation protocol. ProjDrifting-A achieves a better FID of 10.89, compared with 15.60 for Sinkhorn-Drifting. ProjDrifting-A requires 2.57 s/iteration, whereas Sinkhorn-Drifting requires 18.64 s/iteration, yielding a $7.25\times$ speedup.

Table 4: Comparisons on ImageNet 256×256.

Method	Sinkhorn-Drifting	ProjDrifting-A
FID ↓	15.60	**10.89**
Runtime ↓	18.64 s/iter	**2.57 s/iter**

4.3 Analysis

Efficiency Analysis. We effectively leverage the parallelizable nature of the slicing procedure and thereby avoid the efficiency and accuracy trade-off. In Figure 4, we provide an intuitive illustration of the computational efficiency advantage of the proposed ProjDrifting. Iterative solvers such as Sinkhorn-Drifting inevitably exhibit an almost linear increase in training latency as the number of iterations grows. ProjDrifting-A and ProjDrifting-D construct the drift field by aggregating gradients from all projection directions after parallel slicing. As a result, their training latency remains nearly unchanged over a wide range of slice numbers, from 20 to 1000. We further evaluate different ablation variants, including Sinkhorn-Drifting-G and Non-Parallel. Sinkhorn-Drifting-G is obtained by replacing the sliced Wasserstein metric in the ProjDrifting framework with the Sinkhorn metric. The time cost caused by repeated iterations and the inefficiency of backpropagating through the iterative computation jointly lead to a significant increase in training time.



(a) Mode Collapse of Baselines (b) Generated Samples via ProjDrifting-D (c) PCA Analysis of the Source and Target Distribution (d) PCA Analysis of the Generated and Target Samples

Figure 3: (a) Mode collapse of the Drifting model when τ is too small ($\tau = 0.1$). (b) Face images generated by ProjDrifting-D. (c) PCA analysis of the source distribution and the target distribution. (d) PCA analysis of the samples generated by the trained generator and the target samples.

We construct the Non-Parallel variant by implementing the slicing procedure of ProjDrifting-D in a serial manner, to demonstrate the advantage of parallelism. The latency of the Non-Parallel variant becomes large with more projections.

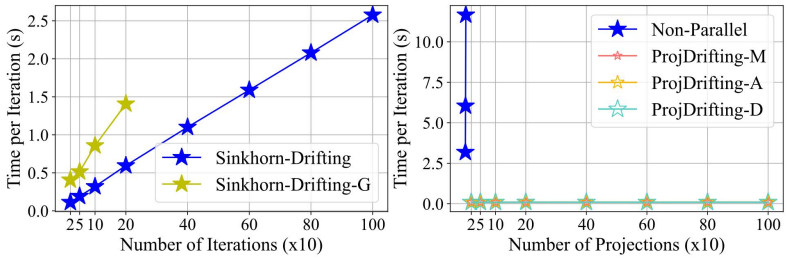


Figure 4: **The left panel** shows the latency of iterative methods. **The right pane** shows the latency of the variants of ProjDrifting, as L increases.

Variants of ProjDrifting. Under the proposed ProjDrifting framework, we implement three variants, i.e., ProjDrifting-A, ProjDrifting-D, and ProjDrifting-M. ProjDrifting-D aims at improve the generation quality of ProjDrifting-A via dynamic projections, while ProjDrifting-M is a non-smoothed variant of ProjDrifting-D. In Table 5, we compare these three variants under different cost functions, i.e., ℓ_2 distance and squared ℓ_2 distance, for analysis of generalization. ProjDrifting-D performs best across all the settings, whereas ProjDrifting-M fails in terms of generation quality.

Table 5: Comparison of variants of ProjDrifting.

Method		MNIST EMD↓	FFHQ FID↓
Model	Distance		
ProjDrifting-A	L2	25.80	30.42
	Squared L2	25.64	29.01
ProjDrifting-D	L2	**21.84**	29.31
	Squared L2	21.87	**26.13**
ProjDrifting-M	L2	22.78	29.33
	Squared L2	22.78	59.51

PCA Visualization. Figure 3 (c) and (d) show the PCA visualization of ProjDrifting-D on FFHQ. Before training, the source distribution is separated from the target distribution, indicating a large distribution gap in the ALAE latent space. After training, the generated samples move toward the target distribution and largely overlap with it in the PCA space. This suggests that ProjDrifting-D aligns the generated latents with the real data manifold while preserving diversity.

5 Conclusions

In this work, we propose ProjDrifting, an efficient projection-based drifting framework for one-step generative modeling. By replacing high-dimensional Sinkhorn iterations with sorting-based one-dimensional optimal transport induced by the sliced Wasserstein metric, ProjDrifting enables parallel-friendly and efficient drift estimation while avoiding the quality–efficiency trade-off of Sinkhorn-based drifting methods. We further introduce a gradient-estimator-based formulation together with an energy-based dynamic slicing mechanism that emphasizes more discriminative transport directions. Theoretical analysis and experiments suggest that projection-based drifting provides a scalable and effective alternative for one-step generative modeling.

AI USE STATEMENT

In this work, we have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, implementing methods, assisting with translation, cleaning and reformatting dataset, supporting qualitative and thematic data analysis, and interpreting results. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we provide detailed descriptions of the proposed ProjDrifting framework and its variants in Section 3, with the drift-field computation summarized in Algorithm 1. The experimental settings, including datasets, model architectures, training configurations, and evaluation metrics, are described in Section 4 and Appendix A. We will make the source code publicly available upon acceptance to facilitate independent reproduction of our results and support future research.

REFERENCES

Michael Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *Journal of Machine Learning Research*, 26(209):1–80, 2025.

Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision*, 51(1):22–45, 2015.

Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Mingyang Deng, He Li, Tianhong Li, Yilun Du, and Kaiming He. Generative modeling via drifting. *arXiv preprint arXiv:2602.04770*, 2026.

Ishan Deshpande, Yuan-Ting Hu, Ruoyu Sun, Ayis Pyrros, Nasir Siddiqui, Sanmi Koyejo, Zhizhen Zhao, David Forsyth, and Alexander G Schwing. Max-sliced wasserstein distance and its use for gans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10648–10656, 2019.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024.

Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trouvé, and Gabriel Peyré. Interpolating between optimal transport and mmd using sinkhorn divergences. In *The 22nd international conference on artificial intelligence and statistics*, pp. 2681–2690. PMLR, 2019.

Aude Genevay, Gabriel Peyre, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617. PMLR, 2018.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Ping He, Om Khangaonkar, Hamed Pirsiavash, Yikun Bai, and Soheil Kolouri. Sinkhorn-drifting generative models. *arXiv preprint arXiv:2603.12366*, 2026.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4401–4410, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 2002.

Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *11th International Conference on Learning Representations, ICLR 2023*, 2023.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *The Eleventh International Conference on Learning Representations (ICLR)*, 2023.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in neural information processing systems*, 35:5775–5787, 2022.

Nanye Ma, Mark Goldstein, Michael S Albergo, Nicholas M Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In *European Conference on Computer Vision*, pp. 23–40. Springer, 2024.

Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pp. 3481–3490. PMLR, 2018.

Khai Nguyen and Nhat Ho. Energy-based sliced wasserstein distance. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, pp. 18046–18075, 2023.

Gabriel Peyré and Marco Cuturi. *Computational optimal transport: With applications to data science*. Now Foundations and Trends, 2019.

Stanislav Pidhorskyi, Donald A Adjeroh, and Gianfranco Doretto. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14104–14113, 2020.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Flowar: Scale-wise autoregressive image generation meets flow matching. In *International Conference on Machine Learning*, pp. 51489–51502. PMLR, 2025.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *arXiv preprint arXiv:2302.00482*, 2023.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, pp. 1–34, 2024.

Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer.

A EXPERIMENTS

A.1 IMPLEMENTATION DETAILS

Unless otherwise specified, we use the squared ℓ_2 distance, i.e., $c(x, y) = \|x - y\|_2^2$, as the cost function, and fix the number of slices to 20 for all experiments. This choice provides a good balance between computational efficiency and approximation quality in practice. For fair comparison, all competing methods are evaluated under consistent training settings whenever possible. Additional implementation details, hyperparameter configurations, and sensitivity analyses are provided in the corresponding experimental sections and the appendix.

Datasets. We conduct experiments on multiple datasets to evaluate the effectiveness of the proposed method under different settings:

- **Synthetic datasets**: We consider commonly used 2D toy distributions, including 8-Gaussians and 2-Moons, to provide intuitive visualization of distribution alignment and convergence behavior.
- **MNIST** LeCun et al. (2002): A handwritten digit dataset consisting of 10 classes, used to evaluate conditional generation performance in terms of both accuracy and diversity.
- **FFHQ** Karras et al. (2019): A high-quality human face dataset with large variations in appearance, used to assess the scalability of our method in high-dimensional latent spaces and its ability to generate realistic images.
- **ImageNet 256×256** Deng et al. (2009): A large-scale natural image dataset spanning 1,000 object categories, used to evaluate the scalability of our method and its ability to generate diverse, high-fidelity images with complex visual structures at 256×256 resolution.

When reproducing the Sinkhorn-Drifting algorithm, we set the number of Sinkhorn iterations to 20, consistent with its official implementation He et al. (2026). Unless otherwise specified, we use the squared ℓ_2 distance, i.e., $c(x, y) = \|x - y\|_2^2$, as the cost function. More implementation details can be found in the corresponding experimental sections. We will open-source the code of the proposed ProjDrifting model once accepted.

A.2 TOY EXPERIMENTS

We evaluate different models on synthetic datasets, using a two-dimensional standard Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I}_2)$ as the source distribution and either 8-Gaussians or 2-Moons as the target distribution. All models are trained with Adam optimizer Kingma & Ba (2014) for 5000 iterations, with batch size 500 and learning rate 10^{-3}. Since both the Drifting model and the Sinkhorn-Drifting model depend on the choice of τ, we set τ to 0.01 and 0.1, respectively. In the toy experiments, the generator is parameterized by a two-layer MLP with a latent dimension of 256.

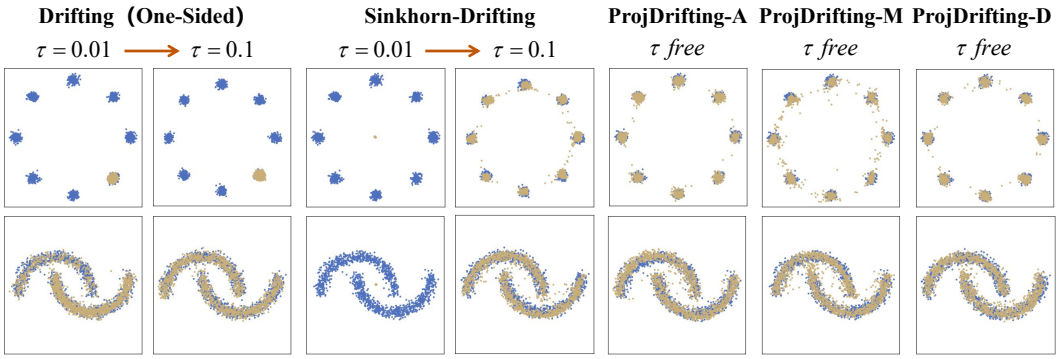


Figure 5: Generated samples on synthetic dataset by different generative methods.

A.3 CLASS-CONDITIONAL IMAGE GENERATION ON MNIST

We evaluate class-conditional generation on MNIST (LeCun et al., 2002). We use a three-layer MLP as the generator and train it for 5000 iterations with a learning rate of 2×10^{-4}.

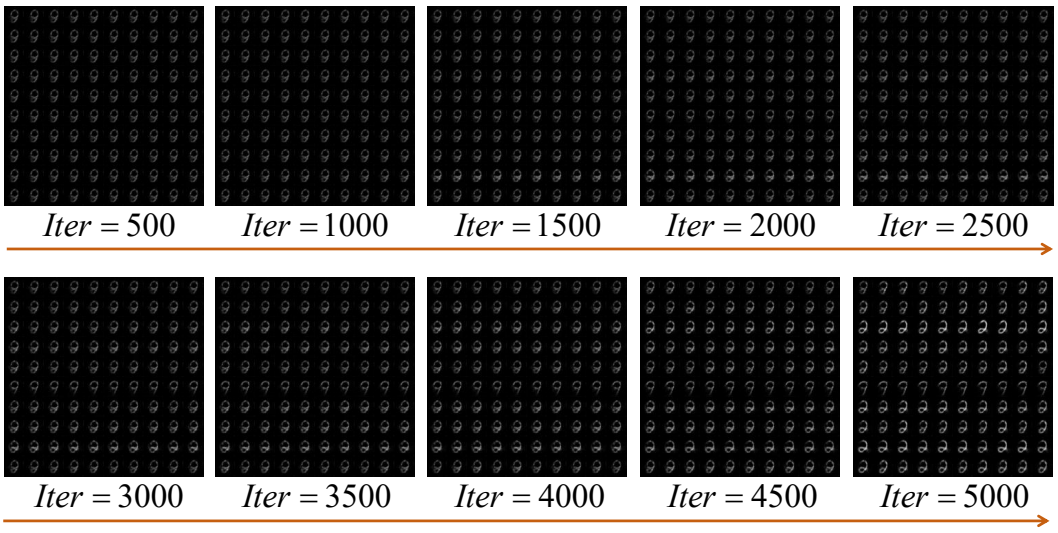


Figure 6: Mode collapse of the Drifting model when τ is too small.

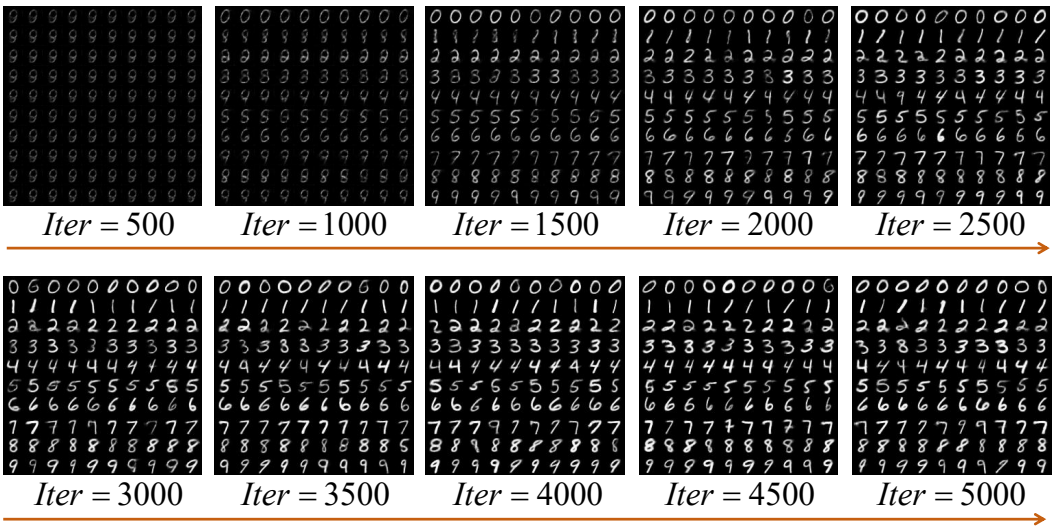


Figure 7: Generated Samples of the Drifting model (τ=0.1) with different numbers of iterations.

A.4 Class-conditional Image generation on FFHQ

FFHQ Karras et al. (2019) is a high-quality human face dataset containing diverse portrait images with large variations in age, gender, and appearance. We conduct conditional image generation experiments on FFHQ based on a pretrained Adversarial Latent Autoencoder (ALAE) Pidhorskyi et al. (2020). Specifically, both training and evaluation are performed in a 512-dimensional feature space, and the generated features are further decoded into 1024×1024 images using a frozen ALAE decoder. Our generator takes as input a $(512 + 64)$-dimensional vector, consisting of a noise vector and a label embedding, and outputs a 512-dimensional feature vector, i.e., $G : \mathbb{R}^{512+64} \to \mathbb{R}^{512}$. The generator is parameterized by a three-layer MLP with hidden width 1024.

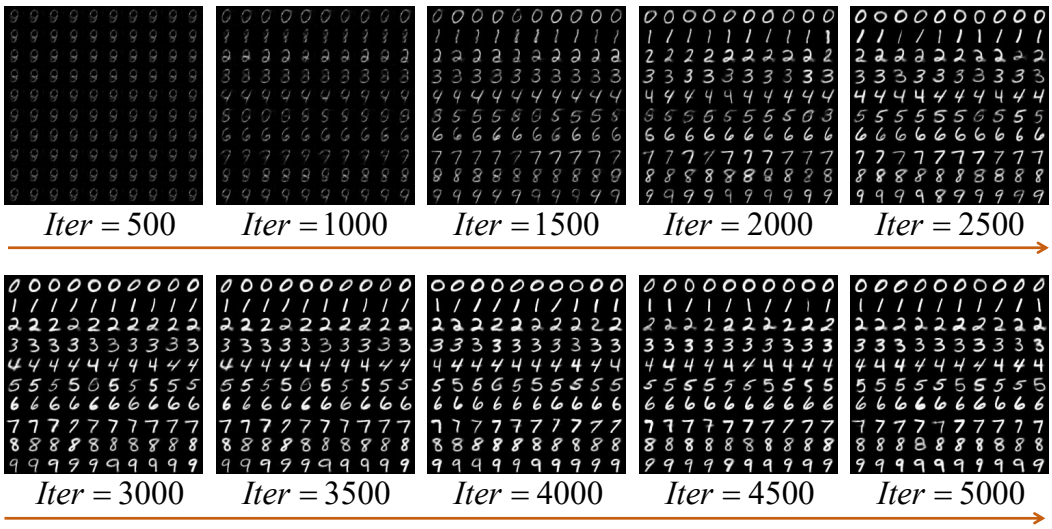


Figure 8: Generated Samples of ProjDrifting-D with different numbers of iterations.



Figure 9: More samples generated by ProjDrifting-D.

A.5 Class-conditional Image generation on ImageNet 256×256

We perform generation in the latent space of a pretrained Stable Diffusion VAE (SD-VAE), which encodes an RGB image of size $256 \times 256 \times 3$ into a latent tensor of size $32 \times 32 \times 4$. The generator takes Gaussian noise of shape $32 \times 32 \times 4$ and produces a latent tensor of the same shape, conditioned on the class label and CFG strength. The generated latent is subsequently decoded into a 256×256 RGB image using the pretrained VAE decoder. Generation requires a single generator forward pass followed by VAE decoding. We use a DiT-B/2 backbone comprising 12 Transformer blocks, with a hidden dimension of 768 and 12 attention heads. The input noise is partitioned into non-overlapping 2×2 patches, yielding 256 spatial tokens. Each patch contains $2 \times 2 \times 4 = 16$ values and is projected to a 768-dimensional embedding. The Transformer incorporates SwiGLU, rotary positional embeddings, RMSNorm, and query-key normalization. Class-label and CFG conditioning are introduced through adaptive layer normalization and 16 additional in-context conditioning tokens. Following the default configuration, 32 random style embeddings, each selected from 64 learnable entries, are summed into the conditioning vector without increasing the sequence length.

The output spatial tokens are projected back to patch values and unpatchified into a $32 \times 32 \times 4$ latent tensor. We use a pretrained, frozen ResNet-style latent-MAE encoder to extract features for the drifting objective. The encoder follows a widened ResNet-34 design with a base width of 256 and four stages containing $[3, 4, 6, 3]$ basic residual blocks, respectively. Each basic block contains two 3×3 convolutions with GroupNorm. The latent-MAE is pretrained with a 50% masking ratio for 192 epochs, without classification fine-tuning. During generator training, its parameters remain frozen, and its multi-scale features are used to construct the drifting loss. The MAE reconstruction decoder is not used for image generation.

B Proof of Propositions

B.1 Proposition 1

If the empirical generated distribution matches the empirical data distribution, i.e., $\hat{q}_X = \hat{p}$, then the drifting field of ProjDrifting vanishes:

$$\widehat{V}_{\hat{p},\hat{q}_X}(x^i) = 0, \qquad \forall i \in \{1, ..., n\}. \tag{15}$$

Proof. When $\hat{q}_X = \hat{p}$, their projected empirical distributions are also identical for any projection direction θ_ℓ, namely $\theta_{\ell\#}\hat{q}_X = \theta_{\ell\#}\hat{p}$. Therefore, under the closed-form one-dimensional OT matching in Eq.(5), the sorted projected generated samples and real samples coincide. For each generated sample x^i, its matched real projected value satisfies $\tilde{b}_i^\ell = a_i^\ell$, and thus the projected displacement is zero:

$$d_i^\ell = \tilde{b}_i^\ell - a_i^\ell = 0. \tag{16}$$

Since this holds for every projection direction θ_ℓ, the aggregated sliced Wasserstein drifting field also vanishes:

$$\widehat{V}_{\hat{p},\hat{q}_X}(x^i) = \frac{1}{L}\sum_{\ell=1}^{L} d_i^\ell \theta_\ell = 0. \tag{17}$$

This proves that the sliced Wasserstein drifting field is zero when the generated empirical distribution matches the target empirical distribution. □

B.2 Proposition 2

If the empirical generated distribution matches the empirical data distribution, i.e., $\hat{q}_X = \hat{p}$, then the dynamic-projection drifting field of ProjDrifting vanishes:

$$\widehat{V}_{\mathrm{dyn}}(x^i) = 0, \qquad \forall i \in \{1, ..., n\}. \tag{18}$$

Proof. When $\hat{q}_X = \hat{p}$, their projected empirical distributions are also identical for any projection direction θ_l:

$$\theta_{l\#}\hat{q}_X = \theta_{l\#}\hat{p}. \tag{19}$$

Therefore, the closed-form one-dimensional OT matching along θ_l induces zero projected displacement for every generated sample. That is, the projected drifting direction satisfies

$$v_{\theta_l}(x^i) = 0, \qquad \forall i,\ l. \tag{20}$$

Since the dynamic drifting field is a weighted aggregation of these projected directions, we have

$$\widehat{V}_{\mathrm{dyn}}(x^i) = \sum_{l=1}^{L} \widehat{w}_{p,q}(\theta_l) v_{\theta_l}(x^i) = 0. \tag{21}$$

This proves the claim. □

B.3 PROPOSITION 3

We compare the computational cost of Sinkhorn-Drifting and ProjDrifting on a mini-batch of size n in dimension d. Let T denote the number of Sinkhorn iterations and L denote the number of projection directions used in sliced Wasserstein estimation.

Proposition 3. *Evaluating the Sinkhorn-divergence energy in Sinkhorn-Drifting has time complexity*

$$\mathcal{O}(n^2 d + Tn^2), \tag{22}$$

whereas evaluating the sliced Wasserstein energy in ProjDriftinghas time complexity

$$\mathcal{O}(Lnd + Ln\log n). \tag{23}$$

Consequently, when $L(d+\log n) \ll n(d+T)$, ProjDriftingis computationally more efficient than Sinkhorn-Drifting.

Proof. For Sinkhorn-Drifting, the energy functional is defined by the Sinkhorn divergence, which consists of a constant number of entropy-regularized OT terms. For each OT term, constructing the pairwise cost matrix between two mini-batches requires computing n^2 distances in $\mathbb{R}^d$, leading to a cost of $\mathcal{O}(n^2 d)$. Given this cost matrix, Sinkhorn normalization performs T rounds of alternating row and column scaling. Each round involves matrix-vector operations over an $n \times n$ matrix and costs $\mathcal{O}(n^2)$. Therefore, evaluating one entropy-regularized OT term costs $\mathcal{O}(n^2 d + Tn^2)$.. Since the Sinkhorn divergence contains only a constant number of such terms, its total time complexity remains $\mathcal{O}(n^2 d + Tn^2)$.

For ProjDrifting, the energy functional is computed by averaging one-dimensional OT costs over L projection directions. For each direction, projecting n samples from $\mathbb{R}^d$ to $\mathbb{R}$ costs $\mathcal{O}(nd)$. The one-dimensional OT cost is then obtained by sorting the projected generated and real samples, which costs $\mathcal{O}(n\log n)$. After sorting, the monotone matching and cost computation require only a linear scan, whose cost is $\mathcal{O}(n)$ and is dominated by the sorting term. Thus, one projection costs $\mathcal{O}(nd + n\log n)$, and all L projections cost $\mathcal{O}(Lnd + Ln\log n)$. □

B.4 PROPOSITION 4

Proposition 4 (Consistency with Infinite Projections)**.** *Let p and q be probability measures on $\mathbb{R}^d$ with finite second moments. For projection directions $\{\theta_\ell\}_{\ell=1}^L$ independently sampled from $\mathcal{U}(\mathbb{S}^{d-1})$, define the Monte Carlo estimator of the sliced Wasserstein energy as*

$$\widehat{\mathcal{F}}_s^L(q) = \frac{1}{L}\sum_{\ell=1}^{L} \mathbb{W}_{1d}\left(\theta_{\ell\#}p, \theta_{\ell\#}q\right). \tag{24}$$

Then, as $L \to \infty$, we have

$$\widehat{\mathcal{F}}_s^L(q) \xrightarrow{\text{a.s.}} \mathcal{F}_s(q) = SW(p,q) = \mathbb{E}_{\theta\sim\mathcal{U}(\mathbb{S}^{d-1})}\left[\mathbb{W}_{1d}\left(\theta_{\#}p, \theta_{\#}q\right)\right]. \tag{25}$$

Proof. Let

$$Z_\ell = \mathbb{W}_{1d}\left(\theta_{\ell\#}p, \theta_{\ell\#}q\right). \tag{26}$$

Since θ_ℓ are independently sampled from $\mathcal{U}(\mathbb{S}^{d-1})$, the random variables $\{Z_\ell\}_{\ell=1}^L$ are i.i.d. Moreover, because p and q have finite second moments, the projected one-dimensional Wasserstein cost is integrable, i.e., $\mathbb{E}_\theta[|Z_\ell|] < \infty$. Therefore, by the strong law of large numbers,

$$\frac{1}{L}\sum_{\ell=1}^{L} Z_\ell \xrightarrow{\text{a.s.}} \mathbb{E}_{\theta\sim\mathcal{U}(\mathbb{S}^{d-1})}\left[Z_\ell\right]. \tag{27}$$

Substituting the definition of Z_ℓ gives

$$\widehat{\mathcal{F}}_s^L(q) \xrightarrow{\text{a.s.}} \mathbb{E}_{\theta\sim\mathcal{U}(\mathbb{S}^{d-1})}\left[\mathbb{W}_{1d}\left(\theta_{\#}p, \theta_{\#}q\right)\right] = \mathcal{F}_s(q). \tag{28}$$

This proves the claim. □

B.5 PROPOSITION 5

Proposition 5 (Equilibrium Consistency). *Let $p, q \in \mathcal{P}_2(\mathbb{R}^d)$, and define*

$$\mathcal{F}_p(q) := SW_2^2(q,p) = \int_{\mathbb{S}^{d-1}} W_2^2(\theta_\# q, \theta_\# p)\, d\sigma(\theta). \tag{29}$$

Assume that the first variation of $\mathcal{F}_p$ with respect to q exists and that the ProjDrifting-A field is given by

$$v_q(x) = -\nabla_x \frac{\delta \mathcal{F}_p}{\delta q}(x). \tag{30}$$

If $v_q(x) = 0$ for q-almost every x, and this condition characterizes a stationary point of $\mathcal{F}_p$, then $q = p$.

Proof. We first show that $\mathcal{F}_p$ is convex with respect to its first measure argument. Let $q_0, q_1 \in \mathcal{P}_2(\mathbb{R}^d)$ and $t \in [0,1]$. Since the push-forward operation is linear,

$$\theta_\#\big((1-t)q_0 + tq_1\big) = (1-t)\theta_\# q_0 + t\theta_\# q_1. \tag{31}$$

Moreover, the squared Wasserstein distance is convex in each measure argument under linear interpolation. Therefore, for every $\theta \in \mathbb{S}^{d-1}$,

$$\begin{aligned} &W_2^2\left(\theta_\#\big((1-t)q_0 + tq_1\big), \theta_\# p\right) \\ &\quad \le (1-t)W_2^2(\theta_\# q_0, \theta_\# p) + tW_2^2(\theta_\# q_1, \theta_\# p). \end{aligned} \tag{32}$$

Integrating this inequality over θ gives

$$\mathcal{F}_p\big((1-t)q_0 + tq_1\big) \le (1-t)\mathcal{F}_p(q_0) + t\mathcal{F}_p(q_1). \tag{33}$$

Hence, $\mathcal{F}_p$ is convex in q.

Under the stated regularity assumption, $v_q = 0$ implies that q is a stationary point of $\mathcal{F}_p$. Because $\mathcal{F}_p$ is convex, every stationary point is a global minimizer. Consequently,

$$\mathcal{F}_p(q) = \min_{\mu \in \mathcal{P}_2(\mathbb{R}^d)} \mathcal{F}_p(\mu). \tag{34}$$

Since $\mathcal{F}_p(p) = SW_2^2(p,p) = 0$ and $\mathcal{F}_p(q) \ge 0$, the minimum value is zero. Thus,

$$SW_2^2(q,p) = 0. \tag{35}$$

Finally, the Sliced Wasserstein distance separates probability measures:

$$SW_2(q,p) = 0 \quad \Longleftrightarrow \quad q = p. \tag{36}$$

Therefore, $q = p$, which establishes the claimed equilibrium consistency. □