Scalable Wasserstein Gradient Flow for Generative Modeling through Unbalanced Optimal Transport

Jaemoo Choi [*1] **Jaewoong Choi** [*2] **Myungjoo Kang** [1]

Abstract

Wasserstein Gradient Flow (WGF) describes the gradient dynamics of probability density within the Wasserstein space. WGF provides a promising approach for conducting optimization over the probability distributions. Numerically approximating the continuous WGF requires the time discretization method. The most well-known method for this is the JKO scheme. In this regard, previous WGF models employ the JKO scheme and parametrize transport map for each JKO step. However, this approach results in quadratic training complexity $O(K^2)$ with the number of JKO step K. This severely limits the scalability of WGF models. In this paper, we introduce a scalable WGF-based generative model, called Semi-dual JKO (S-JKO). Our model is based on the semi-dual form of the JKO step, derived from the equivalence between the JKO step and the Unbalanced Optimal Transport. Our approach reduces the training complexity to $O(K)$. We demonstrate that our model significantly outperforms existing WGF-based generative models, achieving FID scores of 2.62 on CIFAR-10 and 5.46 on CelebA-HQ-256, which are comparable to state-of-the-art image generative models.

1. Introduction

Generative models are a class of Deep Learning models that learn the underlying distribution of training data. There are diverse approaches for generative modeling, such as Energy-based models (Zhao et al., 2016; Du & Mordatch, 2019), Diffusion models (Ho et al., 2020; Song et al., 2021b), Variational Autoencoders (Kingma & Welling, 2014), Flow models (Dinh et al., 2017; Kingma & Dhariwal, 2018), Generative Adversarial Networks (Goodfellow et al., 2020), Optimal Transport Maps (Rout et al., 2022; Choi et al., 2023a), and Wasserstein Gradient Flows. Recently, generative models achieved impressive progress, demonstrating the ability to produce high-quality samples on high-resolution image datasets. Despite these advancements, Wasserstein Gradient Flow models still face challenges in scalability to high-dimensional image datasets.

Wasserstein Gradient Flow (WGF) investigates the minimizing dynamics of probability density following the steepest descent direction of a given functional. WGF plays an important role across various areas involving optimization over probability densities, e.g. Optimal Transport (OT) (Santambrogio, 2017; Carlier et al., 2017), Physics (Carrillo et al., 2022; Adams et al., 2011), Machine learning (Lin et al., 2021; Gao et al., 2019), and Sampling (Bernton, 2018; Frogner & Poggio, 1806; Liu & Wang, 2016; Chewi et al., 2020; Glaser et al., 2021; Cheng et al., 2023b). The Jordan-Kinderlehrer-Otto (JKO) scheme is a prominent method for numerically approximating WGF (Jordan et al., 1998). The JKO scheme corresponds to the time discretization of WGF. The previous works utilized the JKO scheme and conducted optimization for every transport map at each JKO step (Gao et al., 2019; Mokrov et al., 2021; Alvarez-Melis et al., 2022; Bunne et al., 2022; Fan et al., 2022). However, this approach incurs quadratic training complexity $O(K^2)$ with the number of JKO step K. This quadratic complexity arises from the necessity to simulate the entire trajectory of the JKO scheme. This complexity significantly limited the scalability of WGF models through the prolonged training time and the limited model size for parametrization.

To overcome these challenges, we suggest a new generative algorithm by utilizing the semi-dual form of the JKO step. We refer to our model as the Semi-dual JKO (***S-JKO***). Our model consists of two components. First, we introduce the semi-dual form of the JKO step from the equivalence between the JKO step and the Unbalanced Optimal Transport problem (Chizat et al., 2018; Liero et al., 2018) (Sec 4.1). Second, we introduce the reparametrization trick to handle the complexity challenges of existing JKO models (Sec 4.2). Our model reduces the training complexity from quadratic to linear $O(K)$. Our model achieves significantly

[*]Equal contribution [1]Seoul National University [2]Korea Institute for Advanced Study. Correspondence to: Myungjoo Kang <mkang@snu.ac.kr>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

improved scalability compared to existing WGF-based models. Specifically, our S-JKO achieves FID scores of 2.62 on CIFAR-10 and 5.46 on CelebA-HQ, outperforming existing WGF-based methods by a significant margin and approaching state-of-the-art performance. Our contributions can be summarized as follows:

- We propose a WGF-based generative model based on the semi-dual form of the JKO step.
- We show that the JKO step is equivalent to the Unbalanced Optimal Transport problem. This insight leads to the semi-dual form of the JKO step.
- Our model greatly improves the scalability of WGF models until high-dimensional image datasets. To the best of our knowledge, S-JKO is the first JKO-based generative model that presents decent performance on CelebA-HQ (256×256).
- To the best of our knowledge, S-JKO is the first JKO-based generative model that achieves near state-of-the-art performance on real-world image datasets.

Notations and Assumptions Let $\mathcal{P}(\mathbb{R}^d)$ be the set of probability distributions on $\mathbb{R}^d$ that are absolutely continuous with respect to the Lebesgue measure. Throughout this paper, **we denote the source distribution as $\mu = \rho_0$ and denote the target distributions as ν.** Since our scope is on generative modeling, μ and ν correspond to the d-dimensional Gaussian distribution and the data distribution on $\mathbb{R}^d$, respectively. For a measurable map T, $T_{\#}\mu$ represents the pushforward distribution of μ. For convenience, we set $c_h(x, y) := \frac{1}{2h}\|x - y\|_2^2$. Moreover, the 2-Wasserstein distance $\mathcal{W}_2(\cdot, \cdot)$ is defined as follows:

$$\mathcal{W}_2(\rho, \xi) := \left(\min_{\pi \in \Pi(\rho,\xi)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|_2^2 d\pi(x, y) \right)^{\frac{1}{2}}, \tag{1}$$

where $\Pi(\rho, \xi)$ denotes the set of joint probability distributions on $\mathbb{R}^d \times \mathbb{R}^d$ whose marginals are ρ and ξ. Moreover, f^* indicates the convex conjugate of a function f, i.e., $f^*(y) = \sup_{x \in \mathbb{R}}\{\langle x, y\rangle - f(x)\}$ for $f : \mathbb{R} \to [-\infty, \infty]$.

2. Background

2.1. Wasserstein Gradient Flow and JKO scheme

Wasserstein Gradient Flow Given a functional $\mathcal{F}(\rho)$ on $\rho \in \mathcal{P}(\mathbb{R}^d)$, the Wasserstein Gradient Flow (WGF) (Ambrosio et al., 2005) describes the dynamics of probability density $\{\rho_t\}_{t\geq 0}$, following the steepest descent direction of $\mathcal{F}(\rho)$. Here, the metric on $\mathcal{P}(\mathbb{R}^d)$ is defined as the 2-Wasserstein distance $\mathcal{W}_2$ (Eq 1). The WGF can be explicitly written by the PDE as follows:

$$\frac{\partial \rho}{\partial t} = \nabla \cdot \left(\rho \nabla \frac{\delta \mathcal{F}}{\delta \rho} \right), \tag{2}$$

where $\frac{\delta \mathcal{F}}{\delta \rho}$ denotes the first variation of $\mathcal{F}$ with respect to standard L_2 metric (Villani et al., 2009).

When $\mathcal{F}(\rho)$ is given as the f-divergence D_f with respect to the target distribution ν, **WGF describes the trajectories of probability density $\{\rho_t\}_{t\geq 0}$ evolving from $\mu = \rho_0$ towards ν** by minimizing $\mathcal{F}(\rho)$:

$$\mathcal{F}(\rho) := D_f(\rho|\nu) = \int f\left(\frac{d\rho}{d\nu}\right) d\nu. \tag{3}$$

Specifically, when utilizing the KL divergence as the functional $\mathcal{F}(\rho) := D_{KL}(\rho|\nu)$, Eq 2 becomes the **Fokker-Plank equation** with the score $\nabla \log \nu$ (Jordan et al., 1998):

$$\frac{\partial \rho}{\partial t} = \nabla \cdot (\rho \nabla \log \nu) + \Delta \rho, \quad \rho(0, \cdot) = \rho_0, \tag{4}$$

Then, the solution ρ_t converges to ν as $t \to \infty$.

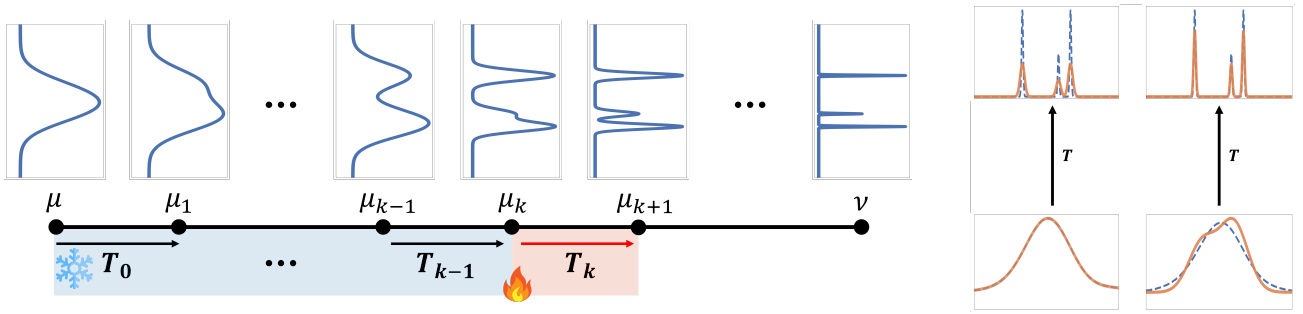
JKO scheme Computing the continuous WGF is a challenging problem. To address this, Jordan et al. (1998) proposed a time discretization scheme to approximate WGF, called the **JKO scheme** (Fig 1a). In this scheme, when given the current JKO step μ_k, the next JKO step μ_{k+1} is formally defined as follows:

$$\mu_{k+1} = \operatorname*{argmin}_{\rho \in \mathcal{P}(\mathbb{R}^d)} \left[\frac{1}{2h} \mathcal{W}_2^2(\rho, \mu_k) + \mathcal{F}(\rho) \right]. \tag{5}$$

where $\mu_0 = \mu$ is the initial condition. Intuitively, h can be understood as the step size of time discretization. When the functional is set to the KL divergence $\mathcal{F}(\rho) = D_{KL}(\rho|\nu)$, the JKO scheme converges to the solution of the Fokker-Plank equation. In other words, $\{\mu_k\}$ converges to $\{\rho_{kh}\}$ in Eq 4 as the step size $h \to 0$.

JKO-based Models In this paragraph, we provide a brief summary of previous works based on the JKO scheme (Mokrov et al., 2021; Alvarez-Melis et al., 2022; Bunne et al., 2022; Fan et al., 2022; Vidal et al., 2023; Park et al., 2023; Lee et al., 2023; Cheng et al., 2023a; Altekrüger et al., 2023). The primary challenge in implementing the JKO scheme lies in optimizing over the probability distributions $\rho \in \mathcal{P}(\mathbb{R}^d)$. The previous works addressed this challenge by transforming it into an optimization over the transport map T from μ_k to μ_{k+1}, i.e., $T_{\#}\mu_k = \mu_{k+1}$. Note that

$$\mathcal{W}_2^2(\mu_k, \mu_{k+1}) = \min_{T_{\#}\mu_k = \mu_{k+1}} \int_{\mathbb{R}^d} \|x - T(x)\|_2^2 d\mu_k(x), \tag{6}$$



(a) Training Process of Existing JKO Models

(b) UOTM variants

Figure 1. **(a) Visualization of the Training Process of Existing JKO Models.** For each training iteration, sampling from μ_k involves sequential inference through k-networks, i.e., $T_k \circ \cdots \circ T_0(x)$ with $x \sim \mu$. This iterative network evaluation considerably slows down the training process. Formally, the training complexity becomes $O(K^2)$ where K denotes the number of JKO steps. **(b) Two Variants of the UOTMs. Left:** Source-fixed UOTM. **Right:** Both-relaxed-UOTM. For brevity, we simply call the Both-relaxed-UOTM as UOTM. UOTMs allow flexibility in marginal densities and therefore have inherent distribution errors (**Blue:** Source and Target distributions μ, ν. **Orange:** Marginal distributions of Optimal Coupling π_0, π_1.)

where T is a measurable map. The transport map T that minimizes Eq 6 is referred to as the optimal transport map from μ_k to μ_{k+1}. Using this fact, Fan et al. (2022) reparametrizes the JKO step (Eq 5) as follows:

$$\mu_{k+1} = T_{k\#}\mu_k,$$
$$T_k = \underset{T}{\operatorname{argmin}} \frac{1}{2h} \int_{\mathbb{R}^d} \|x - T(x)\|_2^2 d\mu_k(x) + \mathcal{F}\left(T_{\#}\mu_k\right). \tag{7}$$

Moreover, Brenier's theorem states that there exists a convex function ψ such that the optimal transport map T_k is a gradient of ψ, i.e., $T_k = \nabla\psi$. Leveraging this fact, several works (Mokrov et al., 2021; Alvarez-Melis et al., 2022; Bunne et al., 2022) parameterizes T as the gradient of input convex neural network (ICNN) (Amos et al., 2017). However, these JKO-based models suffered from the quadratic complexity $O(K^2)$, where K denotes the number of JKO steps. In this regard, Bonet et al. (2022) suggested the Sliced-Wasserstein Gradient Flow to mitigate this complexity to $O(K)$.

2.2. Optimal Transport-based Generative Modeling

Unbalanced Optimal Transport (UOT) The classical OT problem investigates the cost-minimizing transport map that satisfies an exact matching between two distributions (Villani et al., 2009). Recently, a new variation of the OT problem has been introduced, which is called **Unbalanced Optimal Transport (UOT)** (Chizat et al., 2018; Liero et al., 2018). Formally, the UOT problem between the source distribution μ and the target distribution ν is defined as follows:

$$\inf_{\pi \in \mathcal{M}_+} \int_{\mathcal{X}\times\mathcal{Y}} c(x,y) d\pi(x,y) + D_{\varphi_1}(\pi_0|\mu) + D_{\varphi_2}(\pi_1|\nu), \tag{8}$$

where $\mathcal{M}_+$ denotes a set of positive Radon measures on $\mathcal{X} \times \mathcal{Y} = \mathbb{R}^d \times \mathbb{R}^d$ and $c(\cdot,\cdot)$ represents the transportation cost. $D_{\varphi_1}, D_{\varphi_2}$ are two f-divergence[1] terms that penalize the dissimilarity between the marginal distributions π_0, π_1 and μ, ν, respectively. **Note that, in the UOT problem, both marginals of π are not explicitly fixed.** Instead, these marginals are softly regularized by the divergence terms. This flexibility in the marginals provides outlier robustness (Balaji et al., 2020).

Furthermore, the UOT problem is a generalization of the classical OT problem. When we choose φ_1 or φ_2 to be a convex indicator function $\iota(x) = \begin{cases} 0 & \text{if } x = 1, \\ \infty & \text{otherwise} \end{cases}$, then D_φ takes the following form:

$$D_\iota(\pi_i|\rho) = \begin{cases} 0 & \text{if } \pi_i = \rho \text{ almost-surely.} \\ \infty & \text{otherwise.} \end{cases} \tag{9}$$

Therefore, setting $\varphi_1 = \iota$ or $\varphi_2 = \iota$ means fixing the source distribution, i.e., $\pi_0 = \mu$, or the target distribution, i.e., $\pi_1 = \nu$. When we fix both distributions, the UOT problem is simplified to the OT problem. Throughout this paper, we refer to the UOT problem when $\varphi_1 = \iota$ as the **Source-fixed UOT problem**. This problem serves an important role in our work in Sec 4.

UOT-based generative models Recently, Choi et al. (2023a) proposed a class of generative models by leveraging the semi-dual form of the UOT problem. Formally,

[1] In the general case, D_φ is defined as the Csiszàr divergence (Séjourné et al., 2019) D_φ^c for the UOT problem. Note that when μ is absolutely continuous with respect to ν, the f-divergence and the Csiszàr divergence are equivalent, i.e., $D_\varphi^c(\mu|\nu) = D_\varphi(\mu|\nu)$.

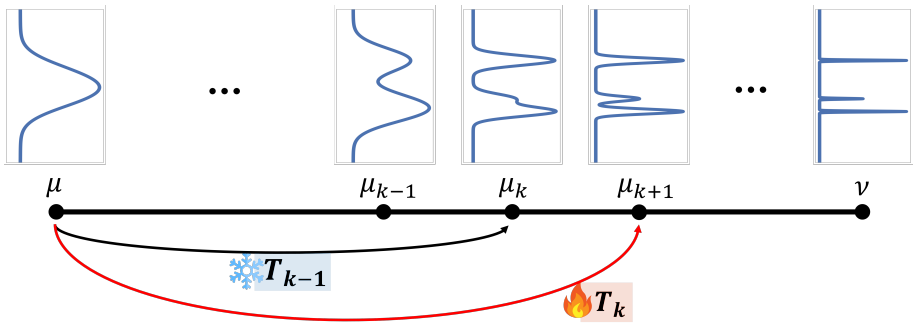


Figure 2. **Conceptual Diagram of Our Model.** During the training k-th JKO step in our model, sampling from μ_k involves only one network inference T_{k-1}, i.e., $\mu_k = T_{k-1\#}\mu_0$. This reparametrization strategy significantly reduces the overall training time. Formally, the training time complexity reduces to $O(K)$ from the $O(K^2)$ of other JKO models. Moreover, by initializing the parameters of T_k with T_{k-1}, we can further decrease the number of iterations required for training.

the semi-dual form of Eq 8 is defined as follows:

$$\sup_{v\in\mathcal{C}} \int \varphi_1^\circ \left(\inf_T \left[c\left(x, T(x)\right) - v(T(x))\right]\right) d\mu(x) + \int \varphi_2^\circ(v(y)) d\nu(y), \quad (10)$$

where $\mathcal{C}$ denotes a set of continuous functions over $\mathbb{R}^d$ and $\varphi_i^\circ(x) := -\varphi_i^*(-x)$. By parametrizing $v = v_\phi$ and $T = T_\theta$ in Eq 10 by neural networks, Choi et al. (2023a) suggested the max-min adversarial learning objective, called UOTM. In this framework, T_θ represents the (unbalanced) transport map from μ to ν, and v_ϕ serves as the potential function for discriminating between $T_{\#}\mu$ and ν. Choi et al. (2023b) demonstrated that the flexibility in distribution matching enhances the stability of the training process. However, **this flexibility also introduces inherent distribution errors to UOTM** (Choi et al., 2023a). Throughout this paper, we call the UOTM variant corresponding to the Source-fixed UOT problem as the **Source-fixed-UOTM**. Specifically, this is equivalent to choosing $D_{\varphi_1} = D_\iota$ and, thereby, $\varphi_i^\circ(x) = x$. On the contrary, when both φ_1 and φ_2 in Eq 8 are not convex indicators, we denote it as **UOTM** or **Both-relaxed-UOTM** (Fig 1b).

3. Limited Scalability of WGF Models

Quadratic Complexity of JKO models The primary challenge for WGF models lies in their limited scalability when dealing with complex high-dimensional image datasets. This limitation stems from the iterative multi-step approximation of intermediate distributions ρ_t in WGF. **This iterative approximation considerably slows down the training process quadratically, i.e., $O(K^2)$, with respect to the total number of approximation steps** K (Table 1). Consequently, the scalability of WGF models is significantly constrained. Specifically, as described in Sec 2, most WGF models employ the JKO scheme to numerically approximate WGF (Mokrov et al., 2021; Alvarez-Melis et al., 2022; Bunne et al., 2022; Fan et al., 2022; Vidal et al., 2023; Park et al., 2023; Lee et al., 2023; Cheng et al., 2023a; Altekrüger et al., 2023). Hence, these models involve the iterative estimation of $(k+1)$-th distribution μ_{k+1}, based on the k-th distribution μ_k (Eq 5). **Each estimation requires a neural network training for learning each transport map T_k from μ_k to μ_{k+1}**, i.e., $(T_k)_{\#}\mu_k = \mu_{k+1}$ (Fig 1a). Note that, for each T_k, **the source data sampling from μ_k requires inference from all $\{T_i\}_{0\le i\le k-1}$ with $x \sim \mu$**:

$$x_k \sim \rho_k \quad \Leftrightarrow \quad \left(T_{k-1} \circ T_{k-2} \circ \cdots \circ T_0\right)(x). \quad (11)$$

Table 1. **Scalability Comparison for Various JKO Schemes on CIFAR-10.** Time denotes a wall-clock training time. Complexity indicates the training complexity with respect to K and d. Here, we only consider the complexity of algorithms, not the complexity of backbone network inference. NFE refers to the number of function evaluations required to produce a sample. Note that training time is measured on 1 GPU (RTX 3090 Ti).

Model	Time	Complexity	NFE (↓)	FID (↓)
Mokrov et al. (2021)	-	$O(K^2d^3)$	-	-
Fan et al. (2022)	$\geq$ 50h	$O(K^2)$	160	23.1
Xu et al. (2023)	$\geq$ 30h	$O(K^2)$	$\geq$ 150	29.1
Source-fixed-UOTM (*Small*)	**6h**	$O(1)$	1	14.4
Ours (*Small*)	**6h**	$O(K)$	1	**8.78**
Source-fixed-UOTM (*Large*)	50h	$O(1)$	1	7.53
Ours (*Large*)	50h	$O(K)$	1	**2.65**

Comparison to Ours Table 1 presents a comparison of the scalability (in terms of training time and complexity) of various JKO models on CIFAR-10. We compared the complexity of previous works, our JKO-based model, and the UOTM-based counterpart of our model (Source-fixed-UOTM). Note that the *Small* backbone network for our model presents a comparable size to previous JKO models (See the Appendix B for details). Therefore, **in terms of scalability, this section focuses on the comparison between the prior JKO models and our models with the**

***Small* backbone**.[2]

The prior JKO models typically utilized $K \geq 150$ JKO steps for approximating WGF on CIFAR-10 (Fan et al., 2022; Xu et al., 2023), and $K = 50 \sim 150$ JKO steps on low-dimensional (~ 100 dimensions) datasets (Fan et al., 2022; Mokrov et al., 2021). In other words, *each WGF model consisted of 50-150 small neural networks, with each neural network dedicated to approximating* T_k. In this respect, the quadratic training complexity $O(K^2)$ considerably limited the scalability of WGF models, by restricting the size of each neural network. As a result, when extended to high-dimensional image datasets of CIFAR-10, WFG models are typically adapted to op3erate on the latent space of encoder-decoder architectures (Fan et al., 2022; Xu et al., 2023). Nevertheless, these models suffer from long training time ($\geq 30h$) and non-competitive generation results of FID score (≥ 20) (Table 1). In this paper, we significantly improve the scalability of WGF models by discovering that the JKO step can be interpreted as the Unbalanced Optimal Transport problem (Sec 4.1). Compared to existing WFG models of similar size, our model outperforms them with a lower FID score of 8.78 on CIFAR-10, while requiring a much less training time of 6 hours.

4. Method

In this section, we propose a novel WGF-based generative model, called the Semi-dual JKO scheme (***S-JKO***). Our model is based on the equivalence between the JKO step and the Unbalanced Optimal Transport problem (Sec 4.1). Building upon this insight, we introduce a generative model based on the semi-dual form of the JKO step (Sec 4.2).

4.1. Equivalence between JKO step and UOT problem

In this subsection, we establish the equivalence between the JKO step (Eq 5) and the Source-fixed variant of the Unbalanced Optimal Transport problem (Eq 8). Here, we begin with the JKO step. Let $\mu = \mu_0$ and ν denote the source and target distributions, respectively. As a reminder, our primary focus is generative modeling. Hence, μ represents the prior distribution (Gaussian), and ν corresponds to the target data distribution. We define the energy functional $F(\rho)$ associated with the JKO step as $\mathcal{F}(\rho) = D_f(\rho|\nu)$. Then, the JKO step (Eq 5) can be expressed as follows:

$$\mu_{k+1} = \underset{\rho \in \mathcal{P}(\mathbb{R}^d)}{\operatorname{argmin}} \underbrace{\frac{1}{2h}\mathcal{W}_2^2(\mu_k, \rho) + D_f(\rho|\nu)}_{\mathcal{L}_\rho}. \tag{12}$$

[2]The experimental results using a *Large* backbone demonstrate that our model is scalable to the competitive backbone network (NCSN++), which is widely employed by state-of-the-art generative models, and can provide comparable performance with it. A more comprehensive discussion will be provided in Sec 5.

If we expand $\mathcal{W}_2^2(\mu_k, \rho)$ using its definition (Eq 1), then $\mathcal{L}_\rho$ can be rewritten as the follows (See the appendix for details):

$$\mathcal{L}_\rho = \min_{\pi \in \Pi(\mu_k, \rho)} \int \frac{1}{2h}\|x-y\|_2^2 d\pi(x,y) + D_f(\pi_1|\nu). \tag{13}$$

Note that $\pi_1 = \rho$ in the above equation. Therefore, when combined with the minimization over ρ in Eq 12, the JKO step is equivalent to the Source-fixed UOT problem, i.e., convex indicator $\varphi_1 = \iota$ and $\varphi_2 = f$ in Eq 8:

$$\pi^\star = \underset{\pi_0 = \mu_k}{\operatorname{argmin}} \int c_h(x,y) d\pi(x,y) + D_f(\pi_1|\nu). \tag{14}$$

$$\mu_{k+1} = \pi_1^\star. \tag{15}$$

with $c_h(x,y) = \frac{1}{2h}\|x-y\|_2^2$. Note that in the UOT problem, when μ, ν are probability distributions (i.e., positive measures with a total mass of 1), then the optimal $\pi^\star$ also has the same total mass (Gallouët et al., 2021). Therefore, performing the optimization over the positive Radon measure in Eq 8 is equivalent to performing the optimization over the joint probability distribution in Eq 14.

4.2. Generative Modeling with the Semi-dual Form of JKO step

In this subsection, we propose a generative model based on the JKO scheme for the WGF. Our model is derived through two steps: (1) Semi-dual form of the JKO scheme from the equivalence with the UOT problem and (2) Reparametrization trick for enhancing the scalability of the JKO scheme.

Semi-dual form of JKO step The semi-dual form of JKO step is obtained from its UOT interpretation (Eq 14). By setting $\varphi_1 = \iota$ and $\varphi_2 = f$ in Eq 8, we can derive the semi-dual form of JKO step from the semi-dual form of UOT (Choi et al., 2023a) as follows (See the appendix for detail):

$$\sup_{v \in \mathcal{C}} \int v^c(x) d\mu_k(x) + \int f^\circ(v(y)) d\nu(y). \tag{16}$$

where the c-transform of v is defined as $v^c(x) := \inf_y (c(x,y) - v(y))$. Here, we parametrize ΔT_k as follows (Rout et al., 2022):

$$\Delta T_k : x \mapsto \arg\min_y \left(c(x,y) - v(y)\right). \tag{17}$$

Then, ΔT_k satisfies the following:

$$v^c(x) := c(x, \Delta T_k(x)) - v(\Delta T_k(x)). \tag{18}$$

Algorithm 1 Training algorithm

Require: Transport network T_θ and the discriminator network v_ϕ.

$T_{\text{old}} = \text{Id}$
for $k = 0, 1, 2, \dots, K$ **do**
  for $i = 0, 1, 2, \dots, N$ **do**
    Sample a batch $x \sim \mu$ and $y \sim \nu$.
    $\hat{y} = T_\theta(x)$.
    Update ϕ by minimizing the objective $\mathcal{L}_v$.
$$\mathcal{L}_v = v_\phi(\hat{y}) - f^\circ\left(v_\phi(y)\right)$$
    Sample a batch $x \sim \mu$.
    $\hat{y} = T_\theta(x)$, $\hat{y}_{\text{old}} = T_{\text{old}}(x)$.
    Update θ by minimizing the objective $\mathcal{L}_T$.
$$\mathcal{L}_T = c\left(\hat{y}_{\text{old}}, \hat{y}\right) - v_\phi(\hat{y})$$
  end for
  $T_{\text{old}} \leftarrow T_\theta$
end for

Therefore, the semi-dual form of the JKO step can be represented as the following adversarial learning objective:

$$\sup_{v \in \mathcal{C}} \int \inf_{\Delta T_k} \left[c\left(x, \Delta T_k(x)\right) - v(\Delta T_k(x))\right] d\mu_k(x) + \int f^\circ(v(y)) d\nu(y). \quad (19)$$

Note that this objective for a single JKO step is equivalent to Source-fixed-UOTM (Choi et al., 2023a) between μ_k and $\mu_{k+1} = (\Delta T_k)_\# \mu_k$.

Reparametrization trick The quadratic training complexity of the JKO models (Table 1) stems from the necessity to simulate the entire trajectory of the JKO scheme $\{\mu_k\}_k$ (Fig 1a). To manage this challenge, we introduce a straightforward reparametrization trick. Suppose $T_k : \mathbb{R}^d \to \mathbb{R}^d$ be a measurable map such that

$$T_k = \Delta T_k \circ \cdots \circ \Delta T_1 \circ \Delta T_0. \quad (20)$$

T_k satisfies $T_k := \Delta T_k \circ T_{k-1}$ and $(T_k)_\# \mu = \mu_{k+1}$. Now, we introduce the reparametrization trick to μ_k in Eq 19:

$$\mathcal{L}_k = \sup_{v \in \mathcal{C}} \int \inf_{T_k} [c\left(T_{k-1}(x), T_k(x)\right) - v(T_k(x))] d\mu(x) + \int f^\circ(v(y)) d\nu(y). \quad (21)$$

Note that this reparametrized learning objective $\mathcal{L}_k$ is for the k-th JKO step. The comprehensive training procedure repeats this step for $k = 0, \cdots, K$. Our reparametrization trick transforms the learning objective from ΔT_k-training to T_k-training. Therefore, for each phase k, we possess a direct transport map T_{k-1} that connects μ to μ_k. In other words, for each phase, sampling from μ_k does not require simulating the entire trajectory. Instead, we can efficiently generate μ_k using a one-step inference. In this regard, **this reparametrization trick significantly contributes to managing the training complexity (Table 1).** Furthermore, there is an additional advantage in terms of training efficiency. For each phase transition, we initialize T_k using the previous T_{k-1}. We hypothesize that this contributes to stable training during the entire training process This training efficiency is empirically demonstrated in Sec 5.

Algorithm Finally, we present our training algorithm (Algorithm 1), called the Semi-dual JKO scheme (**S-JKO**). The adversarial learning objective $\mathcal{L}_k$ is updated through alternating gradient descent, as in GAN (Goodfellow et al., 2020). We simplified Algorithm 1 by excluding non-dependent terms for each v_ϕ and T_θ (See the appendix for the more detailed Algorithm). Additionally, **note that when we conduct training for only one phase, i.e., $K = 1$, our S-JKO is equivalent to the Source-fixed-UOTM (Choi et al., 2023a).** In Sec 5, we will provide further clarification regarding the advantages over Source-fixed-UOTM.

5. Experiments

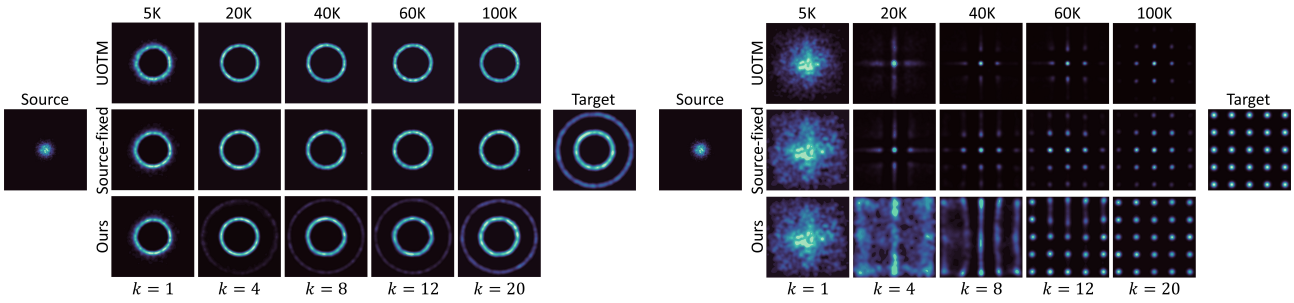
In this section, we conduct experiments on the various datasets to evaluate the following aspects of our model:

- In Sec. 5.1, we compare S-JKO with UOTMs regarding the distribution error between the generated and target distributions on synthetic datasets.
- In Sec. 5.2, we compare S-JKO with other JKO models regarding scalability on large-scale image datasets. Moreover, we demonstrate that S-JKO achieves competitive performance compared to state-of-the-art generative models.
- In Sec. 5.3, we assess the robustness of S-JKO regarding JKO hyperparameters through ablation studies, such as the step size h, the number of JKO steps K, and the functional $\mathcal{F}(\cdot)$.

Throughout this paper, we considered two functionals for $\mathcal{F}(\cdot)$: KL divergence (KLD) $\mathcal{F}(\cdot) = D_{KL}(\cdot|\nu)$ and Jensen-Shannon divergence (JSD) $\mathcal{F}(\cdot) = D_{JSD}(\cdot|\nu)$. Unless otherwise stated, $\mathcal{F}(\cdot)$ is KLD. For further implementation details, please refer to Appendix B.

5.1. Distribution Matching on Synthetic Datasets

As described in Sec 4.1, Source-fixed-UOTM is equivalent to our S-JKO with only one phase training ($K = 1$). UOTM variants exhibit prominent scalability (Choi et al., 2023a). However, the limitation of UOTM variants is that they induce inherent distribution errors (Sec 2.2). Therefore, **we evaluate whether our S-JKO can mitigate this distribu-**



(a) Two Circles (b) 25-Gaussian Mixture

Figure 3. **Generation Results for UOTM, Source-fixed UOTM, and S-JKO on Synthetic Datasets**. Each column shows the generated distribution at training iterations $\{5, 20, 40, 60, 100\}$K. k denotes the index of JKO steps corresponding to that particular iteration.

tion error through multi-phase training. We conducted experiments on two synthetic datasets: Two Circles and 25-Gaussian Mixture. To observe the clear difference, we chose multi-modal datasets as the target datasets, with densities that are spread extensively away from the origin.

Fig 3 illustrates the generated densities for every 5k training iterations. Note that k on the bottom of the figure indicates the number of JKO steps completed at that training iteration. The generated samples from UOTMs (UOTM and Source-fixed UOTM) tend to be confined to a subset of modes near the origin. In contrast, our model successfully captures and covers all modes present in complex multi-modal distributions. We interpret this phenomenon happens due to the inherent distribution errors of the UOTMs. UOTMs aim to minimize the transport cost between the source and generated distributions. Consequently, **the generated distribution from UOTMs tends to cluster around the origin**, because the source distribution is a Gaussian distribution centered at the origin. This mode collapse problem of UOTMs can be exacerbated when the target distribution is spread out far from the origin. Meanwhile, the JKO scheme gradually transforms the source distribution into the target distribution, by approximating the Wasserstein Gradient Flow towards the target distribution. As we iterate through the JKO steps, the discrepancy between the generated and the target distributions gradually decreases (Fig 3). Therefore, **our S-JKO mitigates the distribution error of its one-step variant (Source-fixed UOTM) and UOTM**.

5.2. Scalability on Image Datasets

Training Time In this paragraph, we compare the training times of existing JKO-based algorithms with our model. As aforementioned in Sec. 3, recent JKO-based approaches (Fan et al., 2022; Xu et al., 2023) parametrize the transport map for each JKO step with a separate neural network. This parametrization substantially slows down the training process. On the contrary, as discussed in Sec. 4, our model effectively reduces training time complexity through the reparametrization trick. To verify whether our model reduces the training time in practice, we measured the wall-clock training time on the single GPU of RTX 3090Ti.

As illustrated in Table 1, training our model with a *Small* backbone only requires 6 hours, which is more than 5 times faster than other comparable JKO-based models. Surprisingly, **our model with *Small* backbone not only reduces training time but also significantly outperforms other JKO-based methods**, achieving an FID score of 8.78. In contrast, other JKO-based models show FID scores over 20.

Furthermore, when compared with the Source-fixed UOTM ($K = 1$), **our model exhibits a comparable wall-clock training time to UOTMs**. Specifically, we maintain training time by decreasing the number of iterations per each JKO step (N) (See the Appendix B for the detail). As a result, our model only requires a similar number of iterations to UOTMs, which is approximately 10K iterations. This efficiency comes from our reparametrization trick that enables convergence within this decreased training iterations.

Image Generation We assessed our model on two benchmark datasets: CIFAR-10 (32×32) (Krizhevsky et al., 2009) and CelebA-HQ (256×256) (Liu et al., 2015). For the quantitative evaluation, we employed the FID (Heusel et al., 2017) score. Table 2 shows that **our model with *Large* backbone demonstrates state-of-the-art results on CIFAR-10 among existing WGF-based models, with an FID of 2.62**. (See Table 2 for a more extensive comparison with various generative models.) Our model outperforms the second-best-performing WGF-based model, NSGF (Zhu et al., 2024a), which shows an FID of 5.55, by a significant margin. Note that NSGF employs the same *Large* backbone of NCSN++ (Song et al., 2021b). Furthermore, our model achieves a competitive FID score of 5.46 on CelebA-HQ (256×256). To the best of our knowledge, our model is the first WGF-based generative model that has achieved comparable results with state-of-the-art models on image generation tasks, especially on high-resolution image datasets like CelebA-HQ.

Moreover, to validate the necessity of multiple JKO steps,

Table 2. **Image Generation on CIFAR-10.** † indicates the results conducted by ourselves.

Class	Model	FID (↓)
GAN	SNGAN+DGflow (Ansari et al., 2020)	9.62
	StyleGAN2 w/o ADA (Karras et al., 2020)	8.32
	StyleGAN2 w/ ADA (Karras et al., 2020)	2.92
	DDGAN (T=1)(Xiao et al., 2021)	16.68
	DDGAN (Xiao et al., 2021)	3.75
	RGM (Choi et al., 2023c)	**2.47**
Diffusion	NCSN (Song & Ermon, 2019)	25.3
	DDPM (Ho et al., 2020)	3.21
	Score SDE (VE) (Song et al., 2021b)	2.20
	Score SDE (VP) (Song et al., 2021b)	2.41
	DDIM (50 steps) (Song et al., 2021a)	4.67
	CLD (Dockhorn et al., 2022)	2.25
	Subspace Diffusion (Jing et al., 2022)	2.17
	LSGM (Vahdat et al., 2021)	**2.10**
Flow Matching	FM (Lipman et al., 2023)	6.35
	OT-CFM (Tong et al., 2024)	**3.74**
OT-based	WGAN (Arjovsky et al., 2017)	55.20
	WGAN-GP(Gulrajani et al., 2017)	39.40
	OTM* (*Small*) (Rout et al., 2022)	21.78
	OTM (*Large*)†	7.68
	UOTM (*Small*) (Choi et al., 2023a)	12.86
	UOTM (*Large*) (Choi et al., 2023a)	2.97±0.07
	Source-fixed UOTM (*Small*)†	14.4
	Source-fixed UOTM (*Large*)	7.53
WGF-based	JKO-Flow (Fan et al., 2022)	23.1
	JKO-iFlow (Xu et al., 2023)	29.1
	NSGF (Zhu et al., 2024a) (*Large*)	5.55
	S-JKO (*Small*)†	8.78
	S-JKO (JSD) (*Small*)†	**8.24**
	S-JKO (*Large*)†	**2.62** ±0.04
	S-JKO (JSD) (*Large*)†	2.66±0.05

Table 3. **Image Generation on CelebA-HQ.**

Class	Model	FID (↓)
Diffusion	Score SDE (VP) (Song et al., 2021b)	7.23
	Probability Flow (Song et al., 2021b)	128.13
	LSGM (Vahdat et al., 2021)	7.22
	UDM (Kim et al., 2021)	7.16
	DDGAN (Xiao et al., 2021)	7.64
	RGM (Choi et al., 2023c)	**7.15**
GAN	PGGAN (Karras et al., 2017)	8.03
	Adv. LAE (Pidhorskyi et al., 2020)	19.2
	VQ-GAN (Esser et al., 2021)	10.2
	DC-AE (Parmar et al., 2021)	15.8
	StyleSwin (Zhang et al., 2022)	**3.25**
VAE	NVAE (Vahdat & Kautz, 2020)	29.7
	NCP-VAE (Aneja et al., 2021)	24.8
	VAEBM (Xiao et al., 2020)	**20.4**
OT-based	UOTM	**6.36**
	Source-fixed UOTM†	7.36
WGF-based	**S-JKO**†	6.40
	S-JKO (JSD)†	**5.46**

Table 4. **Ablation Study on Phase Number K.**

K	10	25	50	100	200
S-JKO (KLD)	2.77	2.83	**2.62**	2.73	2.67
S-JKO (JSD)	3.15	3.23	2.86	2.83	**2.66**

Table 5. **Ablation Study on Step Size h.**

h	0.01	0.05	0.1	0.2
S-JKO (KLD)	2.91	2.71	**2.62**	6.11
S-JKO (JSD)	3.82	3.03	2.83	**2.75**

we compare our model with the Source-fixed UOTM, which is equivalent to a single-JKO step model ($K = 1$). Table 2 demonstrates that our model outperforms the Source-fixed UOTM in both architectures on CIFAR-10. Furthermore, our model surpasses the Source-fixed UOTM on CelebA-HQ (256×256) by a large margin. Combining this with the result from Sec. 5.1, we conclude that leveraging multiple JKO steps helps make the generated distribution closer to the target distribution.

5.3. Ablation Studies

In this section, **we conduct ablation studies to assess the robustness of our model on the main hyperparameters for the JKO scheme.** These parameters include the number of JKO step K, the step size h, and the functional $\mathcal{F}(\cdot)$.

Ablation on Phase Number K We conducted an ablation study on the number of JKO steps K. To maintain the total training iterations, we adjusted the number of iterations N per JKO step accordingly. We tested $K \in \{10, 25, 50, 100, 200\}$ for two functionals (KLD and JSD). For each KLD and JSD experiment, we fixed the total number of iterations to 10K and 8K, respectively (See the appendix for details). Table 4 shows that our model with both KLD and JSD shows similar performance across diverse K, which demonstrates that our model is robust to K. Moreover, we observed a marginal improvement in performance as the number of steps increased, achieving FID scores of 2.60 and 2.66 in the KLD and JSD experiments, respectively. We interpret this phenomenon through WGF. A sufficient number of steps are required to converge to the complex data distribution.

Ablation on Step Size h We performed an ablation study on step size h (Table 5). We experimented $h \in \{0.01, 0.05, 0.1, 0.2\}$ while fixing $K = 50$. Both S-JKOs employing KLD and JSD showed the best results around $h = 0.1$ and comparable performance at $h = 0.05$. However, the performance on a too-small $h = 0.01$ slightly declines. We hypothesize that this is because too small h is insufficient to transport the source distribution to the target distribution, within the fixed number of JKO steps. Moreover, S-JKO-KLD exhibited sharp degradation at $h = 0.2$. We interpret this is because of the discretization error of the JKO step. Interestingly, this error is much smaller for JSD. Investigating this difference is beyond the scope of this work. However, we believe this would be an interesting

Table 6. **logSymKL(↓) between Ground-Truth WGF and Each method at $t = 0.5$ for dimensions $d = 2, \cdots, 10$.**

Model	Dual JKO	EM 50K	EM PR 10K	ICNN JKO	Ours
$d = 2$	-1.4	-2.1	-2.0	$\mathbf{-2.6}$	-2.3
$d = 4$	-0.3	-1.0	-0.8	$\mathbf{-2.1}$	-0.9
$d = 6$	0.1	-0.4	-0.2	$\mathbf{-1.8}$	$\mathbf{-1.8}$
$d = 10$	0.6	0.4	0.6	$\mathbf{-1.8}$	-0.1

Table 7. **logSymKL(↓) between Ground-Truth WGF and Each method at $t = 0.9$ for dimensions $d = 2, \cdots, 10$.**

Model	Dual JKO	EM 50K	EM PR 10K	ICNN JKO	Ours
$d = 2$	-1.1	-2.3	-1.9	$\mathbf{-2.4}$	$\mathbf{-2.4}$
$d = 4$	-0.7	-1.0	-0.8	$\mathbf{-2.1}$	-1.2
$d = 6$	-0.5	-0.3	-0.1	$\mathbf{-2.2}$	-0.8
$d = 10$	0.1	0.4	0.4	$\mathbf{-1.8}$	-0.1

future research.

Ablation on f-divergence During our ablation study on other hyperparameters, we also examined the impact of f-divergence: KLD and JSD. In summary, both f-divergences significantly outperform other JKO models and the Source-fixed UOTM on CIFAR-10 (Table 2), and outperform the Source-fixed UOTM on CelebA-HQ (Table 3). S-JKO-KLD achieves slightly better results than S-JKO-JSD on both datasets. However, S-JKO-JSD is more robust to larger h.

5.4. Numerical Comparison to Ground-Truth WGF

We numerically compared our model to the ground truth solution of WGF to assess its accuracy. We followed the experimental settings of the Ornstein-Uhlenbeck process experiments in Mokrov et al. (2021). Specifically, we measured the symmetric KL divergence between the ground-truth solution and the approximate WGF recovered by each method. (See Mokrov et al. (2021) for the details of each method). Table 6 and 7 present the results. ICNN JKO Mokrov et al. (2021) presents the best symmetric KL divergence to the ground truth solution, while our approach demonstrates the second-best results. However, ICNN JKO requires additional cubic complexity of $O(K^2d^3)$ for approximating functional $\mathcal{F}(\rho)$ (Table 1), where K denotes the number of JKO steps and d refers to the data dimension. Hence, it is challenging to apply ICNN JKO to high-dimensional data, such as image datasets in our paper. In this respect, our method provides competitive scalability to high-dimensional datasets, demonstrating a favorable trade-off between scalability and accuracy.

6. Conclusion

In this paper, we introduce S-JKO, a generative model based on the semi-dual form of the JKO scheme. Our work addresses the scalability challenges in previous JKO-based approaches by leveraging (i) the semi-dual form of the JKO scheme, and (ii) by reparametrizing the transport map. Additionally, we explore the relationship with UOTM and enhance the distributional matching between the generated distribution and the target distribution. Through comprehensive experiments on 2D synthetic datasets and large-scale benchmark datasets like CIFAR-10 and CelebA-HQ, we demonstrate that our proposed model generates high-quality samples in large-scale data while faithfully capturing the underlying distribution.

Acknowledgements

This work was supported by KIAS Individual Grant [AP087501] via the Center for AI and Natural Sciences at Korea Institute for Advanced Study, and NRF grant[RS-2024-00421203].

Impact Statement

The advancement of image generation techniques carries the significant potential to influence various scientific and industrial domains, such as machine learning, finance, image synthesis, health care, and anomaly detection. Our method will improve the models in this area since our method addresses the main challenge of generative models: Generating high-quality samples in large-scale datasets while faithfully transporting the distribution to the data. As a result, we anticipate that our model can play a role in addressing the negative social impacts associated with existing generative models that struggle to capture the full diversity of data. On the other hand, the potential negative societal impact of our work is that generative models tend to learn dependencies on the semantics of data, potentially amplifying existing biases. Thus, deploying such models in real-world applications necessitates vigilant monitoring to prevent the reinforcement of societal biases present in the data. It is crucial to meticulously control the training data and modeling process of generative models to mitigate potential negative societal impacts.

References

Adams, S., Dirr, N., Peletier, M. A., and Zimmer, J. From a large-deviations principle to the wasserstein gradient flow: a new micro-macro passage. *Communications in Mathematical Physics*, 307:791–815, 2011.

Altekrüger, F., Hertrich, J., and Steidl, G. Neural wasserstein gradient flows for discrepancies with riesz kernels. In *International Conference on Machine Learning*, pp. 664–690. PMLR, 2023.

Alvarez-Melis, D., Schiff, Y., and Mroueh, Y. Optimizing functionals on the space of probabilities with input convex

neural networks. *Transactions on Machine Learning Research*, 2022.

Ambrosio, L., Gigli, N., and Savaré, G. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.

Amos, B., Xu, L., and Kolter, J. Z. Input convex neural networks. In *International Conference on Machine Learning*, pp. 146–155. PMLR, 2017.

Aneja, J., Schwing, A., Kautz, J., and Vahdat, A. A contrastive learning approach for training variational autoencoder priors. *Advances in neural information processing systems*, 34:480–493, 2021.

Ansari, A. F., Ang, M. L., and Soh, H. Refining deep generative models via discriminator gradient flow. *arXiv preprint arXiv:2012.00780*, 2020.

Arjovsky, M., Chintala, S., and Bottou, L. Wasserstein generative adversarial networks. In *International conference on machine learning*, pp. 214–223. PMLR, 2017.

Balaji, Y., Chellappa, R., and Feizi, S. Robust optimal transport with applications in generative modeling and domain adaptation. *Advances in Neural Information Processing Systems*, 33:12934–12944, 2020.

Bernton, E. Langevin monte carlo and jko splitting. In *Conference on learning theory*, pp. 1777–1798. PMLR, 2018.

Bonet, C., Courty, N., Septier, F., and Drumetz, L. Efficient gradient flows in sliced-wasserstein space. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856.

Bunne, C., Papaxanthos, L., Krause, A., and Cuturi, M. Proximal optimal transport modeling of population dynamics. In *International Conference on Artificial Intelligence and Statistics*, pp. 6511–6528. PMLR, 2022.

Carlier, G., Duval, V., Peyré, G., and Schmitzer, B. Convergence of entropic schemes for optimal transport and gradient flows. *SIAM Journal on Mathematical Analysis*, 49(2):1385–1418, 2017.

Carrillo, J. A., Craig, K., Wang, L., and Wei, C. Primal dual methods for wasserstein gradient flows. *Foundations of Computational Mathematics*, pp. 1–55, 2022.

Cheng, X., Lu, J., Tan, Y., and Xie, Y. Convergence of flow-based generative models via proximal gradient descent in wasserstein space. *arXiv preprint arXiv:2310.17582*, 2023a.

Cheng, Z., Zhang, S., Yu, L., and Zhang, C. Particle-based variational inference with generalized wasserstein gradient flow. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b.

Chewi, S., Le Gouic, T., Lu, C., Maunu, T., and Rigollet, P. Svgd as a kernelized wasserstein gradient flow of the chi-squared divergence. *Advances in Neural Information Processing Systems*, 33:2098–2109, 2020.

Chizat, L., Peyré, G., Schmitzer, B., and Vialard, F.-X. Unbalanced optimal transport: Dynamic and kantorovich formulations. *Journal of Functional Analysis*, 274(11): 3090–3123, 2018.

Choi, J., Choi, J., and Kang, M. Generative modeling through the semi-dual formulation of unbalanced optimal transport. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023a.

Choi, J., Choi, J., and Kang, M. Analyzing and improving ot-based adversarial networks. In *The Twelfth International Conference on Learning Representations*, 2023b.

Choi, J., Park, Y., and Kang, M. Restoration based generative models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202. PMLR, 2023c.

Dinh, L., Sohl-Dickstein, J., and Bengio, S. Density estimation using real NVP. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Dockhorn, T., Vahdat, A., and Kreis, K. Score-based generative modeling with critically-damped langevin diffusion. *The International Conference on Learning Representations*, 2022.

Du, Y. and Mordatch, I. Implicit generation and modeling with energy based models. *Advances in Neural Information Processing Systems*, 32, 2019.

Esser, P., Rombach, R., and Ommer, B. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 12873–12883, 2021.

Fan, J., Zhang, Q., Taghvaei, A., and Chen, Y. Variational wasserstein gradient flow. In *proceedings of international conference on machine learning*, 2022.

Frogner, C. and Poggio, T. Approximate inference with wasserstein gradient flows (2018). *arXiv preprint arXiv:1806.04542*, 1806.

Gallouët, T., Ghezzi, R., and Vialard, F.-X. Regularity theory and geometry of unbalanced optimal transport. *arXiv preprint arXiv:2112.11056*, 2021.

Gao, R., Song, Y., Poole, B., Wu, Y. N., and Kingma, D. P. Learning energy-based models by diffusion recovery likelihood. *Advances in neural information processing systems*, 2021.

Gao, Y., Jiao, Y., Wang, Y., Wang, Y., Yang, C., and Zhang, S. Deep generative learning via variational gradient flow. In *International Conference on Machine Learning*, pp. 2093–2101. PMLR, 2019.

Glaser, P., Arbel, M., and Gretton, A. Kale flow: A relaxed kl gradient flow for probabilities with disjoint support. *Advances in Neural Information Processing Systems*, 34: 8018–8031, 2021.

Gong, X., Chang, S., Jiang, Y., and Wang, Z. Autogan: Neural architecture search for generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3224–3234, 2019.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. C. Improved training of wasserstein gans. *Advances in neural information processing systems*, 30, 2017.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Jiang, Y., Chang, S., and Wang, Z. Transgan: Two transformers can make one strong gan. *arXiv preprint arXiv:2102.07074*, 1(3), 2021.

Jing, B., Corso, G., Berlinghieri, R., and Jaakkola, T. Subspace diffusion generative models. *arXiv preprint arXiv:2205.01490*, 2022.

Jordan, R., Kinderlehrer, D., and Otto, F. The variational formulation of the fokker–planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.

Karras, T., Aila, T., Laine, S., and Lehtinen, J. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., and Aila, T. Training generative adversarial networks with limited data. *Advances in Neural Information Processing Systems*, 33:12104–12114, 2020.

Kim, D., Shin, S., Song, K., Kang, W., and Moon, I.-C. Score matching model for unbounded data score. *arXiv preprint arXiv:2106.05527*, 2021.

Kingma, D. P. and Dhariwal, P. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. In Bengio, Y. and LeCun, Y. (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. 2009.

Lee, W., Wang, L., and Li, W. Deep jko: time-implicit particle methods for general nonlinear gradient flows. *arXiv preprint arXiv:2311.06700*, 2023.

Liero, M., Mielke, A., and Savaré, G. Optimal entropy-transport problems and a new hellinger–kantorovich distance between positive measures. *Inventiones mathematicae*, 211(3):969–1117, 2018.

Lin, A. T., Li, W., Osher, S., and Montúfar, G. Wasserstein proximal of gans. In *International Conference on Geometric Science of Information*, pp. 524–533. Springer, 2021.

Lipman, Y., Chen, R. T. Q., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023.

Liu, Q. and Wang, D. Stein variational gradient descent: A general purpose bayesian inference algorithm. *Advances in neural information processing systems*, 29, 2016.

Liu, Z., Luo, P., Wang, X., and Tang, X. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pp. 3730–3738, 2015.

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

Mokrov, P., Korotin, A., Li, L., Genevay, A., Solomon, J. M., and Burnaev, E. Large-scale wasserstein gradient flows. *Advances in Neural Information Processing Systems*, 34: 15243–15256, 2021.

Park, M. S., Kim, C., Son, H., and Hwang, H. J. The deep minimizing movement scheme. *Journal of Computational Physics*, 494:112518, 2023.

Parmar, G., Li, D., Lee, K., and Tu, Z. Dual contradistinctive generative autoencoder. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 823–832, 2021.

Pidhorskyi, S., Adjeroh, D. A., and Doretto, G. Adversarial latent autoencoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14104–14113, 2020.

Roth, K., Lucchi, A., Nowozin, S., and Hofmann, T. Stabilizing training of generative adversarial networks through regularization. *Advances in neural information processing systems*, 30, 2017.

Rout, L., Korotin, A., and Burnaev, E. Generative modeling with optimal transport maps. In *International Conference on Learning Representations*, 2022.

Santambrogio, F. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 7:87–154, 2017.

Séjourné, T., Feydy, J., Vialard, F.-X., Trouvé, A., and Peyré, G. Sinkhorn divergences for unbalanced optimal transport. *arXiv e-prints*, pp. arXiv–1910, 2019.

Song, J., Meng, C., and Ermon, S. Denoising diffusion implicit models. *Advances in Neural Information Processing Systems*, 2021a.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. *Advances in Neural Information Processing Systems*, 32, 2019.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. *The International Conference on Learning Representations*, 2021b.

Tong, A., FATRAS, K., Malkin, N., Huguet, G., Zhang, Y., Rector-Brooks, J., Wolf, G., and Bengio, Y. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856.

Vacher, A. and Vialard, F.-X. Semi-dual unbalanced quadratic optimal transport: fast statistical rates and convergent algorithm. In *International Conference on Machine Learning*, pp. 34734–34758. PMLR, 2023.

Vahdat, A. and Kautz, J. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33:19667–19679, 2020.

Vahdat, A., Kreis, K., and Kautz, J. Score-based generative modeling in latent space. *Advances in Neural Information Processing Systems*, 34:11287–11302, 2021.

Van Oord, A., Kalchbrenner, N., and Kavukcuoglu, K. Pixel recurrent neural networks. In *International conference on machine learning*, pp. 1747–1756. PMLR, 2016.

Vidal, A., Wu Fung, S., Tenorio, L., Osher, S., and Nurbekyan, L. Taming hyperparameter tuning in continuous normalizing flows using the jko scheme. *Scientific Reports*, 13(1):4501, 2023.

Villani, C. et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

Xiao, Z., Kreis, K., Kautz, J., and Vahdat, A. Vaebm: A symbiosis between variational autoencoders and energy-based models. *arXiv preprint arXiv:2010.00654*, 2020.

Xiao, Z., Kreis, K., and Vahdat, A. Tackling the generative learning trilemma with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.

Xu, C., Cheng, X., and Xie, Y. Normalizing flow neural networks by JKO scheme. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

Zhang, B., Gu, S., Zhang, B., Bao, J., Chen, D., Wen, F., Wang, Y., and Guo, B. Styleswin: Transformer-based gan for high-resolution image generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11304–11314, 2022.

Zhao, J., Mathieu, M., and LeCun, Y. Energy-based generative adversarial networks. In *International Conference on Learning Representations*, 2016.

Zhu, H., Wang, F., Zhang, C., Zhao, H., and Qian, H. Neural sinkhorn gradient flow. *arXiv preprint arXiv:2401.14069*, 2024a.

Zhu, Y., Xie, J., Wu, Y. N., and Gao, R. Learning energy-based models by cooperative diffusion recovery likelihood. In *The Twelfth International Conference on Learning Representations*, 2024b.

A. Derivation & Algorithm

In this section, we provide a derivation of our optimization problem (Eq 21) and provide the precise algorithms for our model.

Notations and Asssumptions Assume that all the distributions including μ, μ_k and ν be absolutely continuous with respect to the Lebesgue measure. Throughout the paper, we assume that every function f under the subscript of divergence D, i.e. D_f, is a convex, differentiable, and nonnegative function defined on $\mathbb{R}^+$. f^* is a convex conjugate of f, i.e. $f^*(y) := \sup_x (\langle x, y\rangle - f(x))$. For convenience, we define $f^\circ(x) := -f^*(-x)$. Moreover, we set $c_h(x, y) := \frac{1}{2h}\|x-y\|_2^2$.

Derivation of the Optimization Problem Before the derivation, we start with the following Lemma:

Lemma A.1. *(Chizat et al., 2018; Vacher & Vialard, 2023; Gallouët et al., 2021; Choi et al., 2023a) Consider the following optimization problem:*

$$\inf_{\pi\in\mathcal{M}_+} \int_{\mathcal{X}\times\mathcal{Y}} c_h(x, y)d\pi(x, y) + D_{\varphi_1}(\pi_0|\mu_k) + D_{\varphi_2}(\pi_1|\nu). \tag{22}$$

Then, the semi-dual formulation of Eq 22 is given as

$$\sup_{v\in\mathcal{C}} \int \varphi_1^\circ \left(v^c(x)\right) d\mu_k(x) + \int \varphi_2^\circ(v(y))d\nu(y), \tag{23}$$

where $v^c(x) = \inf_y [c(x, y) - v(y)]$. Moreover, the strong duality holds.

Proof. See Choi et al. (2023a) for the proof. □

This Lemma enables us to reformulate the JKO optimization problem into the semi-dual formulation of UOT. Suppose $\mu = \mu_0$ and ν are source and target distributions, respectively, and let $\mathcal{F}(\rho) := D_f(\rho|\nu)$. Then, our optimization problem Eq 5 is as the follows:

$$\mu_{k+1} = \operatorname*{argmin}_{\rho\in\mathcal{P}(\mathbb{R}^d)} \underbrace{\frac{1}{2h}\mathcal{W}_2^2(\mu_k, \rho) + D_f(\rho|\nu)}_{\mathcal{L}_\rho}. \tag{24}$$

Recall the definition of 2-Wasserstein distance $\mathcal{W}_2$:

$$\mathcal{W}_2^2(\rho, \xi) := \min_{\pi\in\Pi(\rho,\xi)} \int_{\mathbb{R}^d\times\mathbb{R}^d} \|x - y\|_2^2 d\pi(x, y). \tag{25}$$

By the definition of $\mathcal{W}_2$, the objective function $\mathcal{L}_\rho$ of Eq 12 can be rewritten as the follows:

$$\mathcal{L}_\rho = \min_{\pi\in\Pi(\mu_k,\rho)} \int_{\mathbb{R}^d\times\mathbb{R}^d} c_h(x, y)d\pi + D_f(\pi_1|\nu). \tag{26}$$

Then, by combining Eq 12 and Eq 26, we obtain the following optimization problem:

$$\inf_{\pi_0=\mu_k} \left[\int_{\mathbb{R}^d\times\mathbb{R}^d} c_h(x, y)d\pi(x, y) + D_f(\pi_1|\nu)\right]. \tag{27}$$

By configuring $\varphi_1 = \iota$ and $\varphi_2 = f$ in Eq 22, it boils down to Eq 27. Thus, by applying Lemma A.1, the semi-dual of Eq 27 is written as follows:

$$\sup_{v\in\mathcal{C}} \int v^c(x)d\mu_k(x) + \int f^\circ(v(y))d\nu(y). \tag{28}$$

Note that $v^c(x) := \inf_y(c(x, y) - v(y))$. Equivalently, we can define $v^c(x) := \inf_{\Delta T}(c(x, \Delta T(x)) - v(\Delta T(x)))$. Thus, Eq 28 can be rewritten as follows:

$$\sup_{v\in\mathcal{C}} \int \inf_{\Delta T} \left[c\left(x, \Delta T(x)\right) - v(\Delta T(x))\right] d\mu_k(x) + \int f^\circ(v(y))d\nu(y). \tag{29}$$

Now, suppose $T_k : \mathbb{R}^d \to \mathbb{R}^d$ be a measurable map such that $T_{k\#}\mu = \mu_k$. Then, by reparametrizing $T := \Delta T \circ T_k$, Eq 29 is equivalent to

$$\sup_{v\in\mathcal{C}} \int \inf_{T} \left[c\left(T_k(x), T(x)\right) - v(T(x))\right] d\mu(x) + \int f^\circ(v(y))d\nu(y). \tag{30}$$

Algorithm 2 Training algorithm of S-JKO with KLD

Require: Transport network T_θ and the discriminator network v_ϕ.

1: $T_{\text{old}} = \text{Id}$
2: **for** $k = 0, 1, 2, \ldots, K$ **do**
3: **for** $i = 0, 1, 2, \ldots, N$ **do**
4: Sample a batch $x \sim \mu$, $y \sim \nu$, and $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
5: $\hat{y} = T_\theta(x, z)$.
6: Update ϕ by using the loss $\mathcal{L}_v$.

$$\mathcal{L}_v = v_\phi(\hat{y}) + f^*\left(-v_\phi(y)\right) + \lambda \|\nabla v_\phi(y)\|_2^2.$$

7: Sample a batch $x \sim \mu$, and $z_1, z_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
8: $\hat{y} = T_\theta(x, z_1)$, $\hat{y}_{\text{old}} = T_{\text{old}}(x, z_2)$.
9: Update θ by using the loss $\mathcal{L}_T$.

$$\mathcal{L}_T = c\left(\hat{y}_{\text{old}}, \hat{y}\right) - v_\phi(\hat{y})$$

10: **end for**
11: $T_{\text{old}} \leftarrow T_\theta$
12: **end for**

Algorithm 3 Training algorithm of S-JKO with JSD

Require: Transport network T_θ and the discriminator network w_ϕ.

1: $T_{\text{old}} = \text{Id}$
2: **for** $k = 0, 1, 2, \ldots, K$ **do**
3: **for** $i = 0, 1, 2, \ldots, N$ **do**
4: Sample a batch $x \sim \mu$, $y \sim \nu$, and $z \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
5: $\hat{y} = T_\theta(x, z)$.
6: Update ϕ by using the loss $\mathcal{L}_v$.

$$\mathcal{L}_v = S\left(w_\phi(\hat{y})\right) + S\left(-w_\phi(y)\right) + \lambda \|\nabla w_\phi(y)\|_2^2.$$

7: Sample a batch $x \sim \mu$, and $z_1, z_2 \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$.
8: $\hat{y} = T_\theta(x, z_1)$, $\hat{y}_{\text{old}} = T_{\text{old}}(x, z_2)$.
9: Update θ by using the loss $\mathcal{L}_T$.

$$\mathcal{L}_T = c\left(\hat{y}_{\text{old}}, \hat{y}\right) + S\left(-w_\phi(\hat{y})\right)$$

10: **end for**
11: $T_{\text{old}} \leftarrow T_\theta$
12: **end for**

Algorithm In image generation tasks, we slightly modify Algorithm 1 by following implementations in Choi et al. (2023a;b). We demonstrate the precise training algorithm with KLD in Algorithm 2. As shown in lines 4-5 and lines 7-8, we additionally plugged auxiliary variable z into the network T_θ. This strategy is known to yield additional improvements in performance (Xiao et al., 2021; Choi et al., 2023c;a;b). Moreover, we incorporate R_1 regularizer (Roth et al., 2017), i.e. $\lambda \|\nabla v_\phi(y)\|_2^2$ in line 5, which is a popular regularization employed in various studies (Mescheder et al., 2018; Xiao et al., 2021; Choi et al., 2023c;a;b). Furthermore, note that the cost function in line 9 is $c(x, y) = \frac{1}{2dh}\|x - y\|_2^2$.

Algorithm 3 demonstrates the exact algorithm for our model with JSD. In image generation tasks on JSD, we also employ additional auxiliary variables and R_1 regularizer. Suppose D_f is JSD, then the convex conjugate of f is

$$f^*(x) = \begin{cases} -\log(2 - e^x), & \text{if } x < \log 2, \\ \infty, & \text{if } x \geq \log 2. \end{cases}$$

Since the $f^*(v_\phi(y))$ is infinite whenever $v_\phi(y) \geq \log 2$, the reparametrization for v_ϕ is inevitable. Thus, we introduce w_ϕ, which is a reparametrization of v_ϕ as defined as follows:

$$w_\phi(y) = \left(\sigma^{-1} \circ \exp\right)\left(v_\phi(y) - \log 2\right), \tag{31}$$

where σ is a sigmoid function. Then, the objective $\mathcal{L}_v$ and $\mathcal{L}_T$ in lines 5 and 9 in Algorithm 1 can be written as the follows:

$$\begin{aligned}\mathcal{L}_v &= S\left(w_\phi(\hat{y})\right) + S\left(-w_\phi(y)\right)\\ \mathcal{L}_T &= c(\hat{y}_{\text{old}}, \hat{y}) - S\left(w_\phi(\hat{y})\right),\end{aligned} \tag{32}$$

where $S(x)$ is a softplus function, i.e. $S(x) := \log(1 + e^x)$. However, technically, it is well-known that the gradient of the generator T saturates when trained with $\mathcal{L}_T$ (Goodfellow et al., 2020). Thus, by following the technical modification introduced in Goodfellow et al. (2020), we modify the object as follows:

$$\begin{aligned}\mathcal{L}_v &= S\left(w_\phi(\hat{y})\right) + S\left(-w_\phi(y)\right)\\ \mathcal{L}_T &= c(\hat{y}_{\text{old}}, \hat{y}) + S\left(-w_\phi(\hat{y})\right).\end{aligned} \tag{33}$$

Finally, we obtain the training algorithm of S-JKO with JSD as the Algorithm 3.

B. Implementation Details

Unless otherwise stated, the source distribution μ is a d-dimensional standard Gaussian distribution and the target distribution ν is a data distribution.

B.1. Synthetic data

Two Circles Suppose P is a uniform distribution on the circles of radius 4 and 8. Then, we generate 2D "Two Circle" data as follows:

$$x + 0.2z \quad x \sim P \quad z \sim \mathcal{N}(0, \mathbf{I}).$$

25-Gaussian Mixture Let P be a uniform distribution on $\{(3i, 3j) : i, j \in \{-2, -1, 0, 1, 2\}\}$. Then, we generate 2D "25-Gaussian Mixture" data as follows:

$$x + 0.005z \quad x \sim P \quad z \sim \mathcal{N}(0, \mathbf{I}).$$

Implementation Details For all synthetic experiments, we used the same architectures for the transport map T_θ and the potential network v_ϕ. We followed the architectures and hyperparameters of Choi et al. (2023b) unless otherwise stated. We used a batch size of 400 and a learning rate of 10^{-4} and 10^{-5} for the transport and potential networks, respectively. We trained the networks for 100K iterations for UOTMs. For our model, we trained for 20 JKO steps ($K = 20$), and 5K iterations for each JKO step ($N = 5000$). Thus, our models are also trained for 100K iterations. Moreover, we set $h = 5$ for the 25-Gaussian Mixture, and $h = 2$ for the Two Circles data. We do not use any regularizations.

B.2. Image Generation

Otherwise stated, all the implementation details including preprocessing, hyperparameters, and architectures follow the implementation of Choi et al. (2023b). The DCGAN model, which is written as *Small* throughout the manuscript, follows the architecture employed in Rout et al. (2022). For *Large* model, we follow the implementation of Choi et al. (2023a). For all implementations, We employ a batch size of 256, Adam optimizer with $(\beta_1, \beta_2) = (0.5, 0.9)$, and the learning rate of 2×10^{-4} and 10^{-4} for the T_θ and v_ϕ networks, respectively. Moreover, we used R_1 regularization of $\lambda = 0.2$ for CIFAR-10 experiments, and $\lambda = 20$ for CelebA-256 experiments. For the implementation of our model with KL divergence, we trained for 50 JKO steps ($K = 50$), 10K iterations for the first JKO step, and 2K iterations for other JKO steps ($N = 2000$). In total, we train for 110K iterations. For the implementation of our model with Jensen-Shannon divergence, we trained for 35 JKO steps ($K = 50$), 10K iterations for the first JKO step, and 2K iterations for other JKO steps ($N = 2000$). In total, we train for 80K iterations. For the implementation of ablation on K, we adjusted the number of iterations for each JKO step, i.e. N, to fix the total number of training iterations.

Number of network parameters In this paragraph, we compare the number of network parameters between comparison models ((Fan et al., 2022; Xu et al., 2023)) to our S-JKO on the CIFAR-10 experiments. For *Small* and *Large* architecture, we use approximately 0.4M and 48M number of parameters for T_θ, respectively. Fan et al. (2022) employs more than 30M parameters. Moreover, since Xu et al. (2023) use encoder-decoder networks, they can save the number of parameters to approximately 2-3M.

Figure 4. **CIFAR-10 trajectories from S-JKO (KLD)** for $K = 5j + 1$ ($0 \leq j \leq 9$).

Evaluation Metric We used 50,000 generated samples to measure FID (Heusel et al., 2017) scores.

C. Additional Results

C.1. Training dynamics

Through Fig 4 and 5, we visualize the trajectories of S-JKO trained on CIFAR-10. We sampled a batch $x \sim \mu$ and visualized $\{T_{5j+1}(x)\}$ for a non-negative integer j.

C.2. Additional Qualitative Results

Through Fig 6-11, we present generated samples for S-JKO trained on CIFAR-10 and CelebA-HQ (256×256).

Figure 5. **CIFAR-10 trajectories from S-JKO (JSD)** for $K = 5j + 1$ ($0 \leq j \leq 7$).

Figure 6. **Generated samples from S-JKO (KLD)** trained on CIFAR-10 (32×32) with *Large* model.

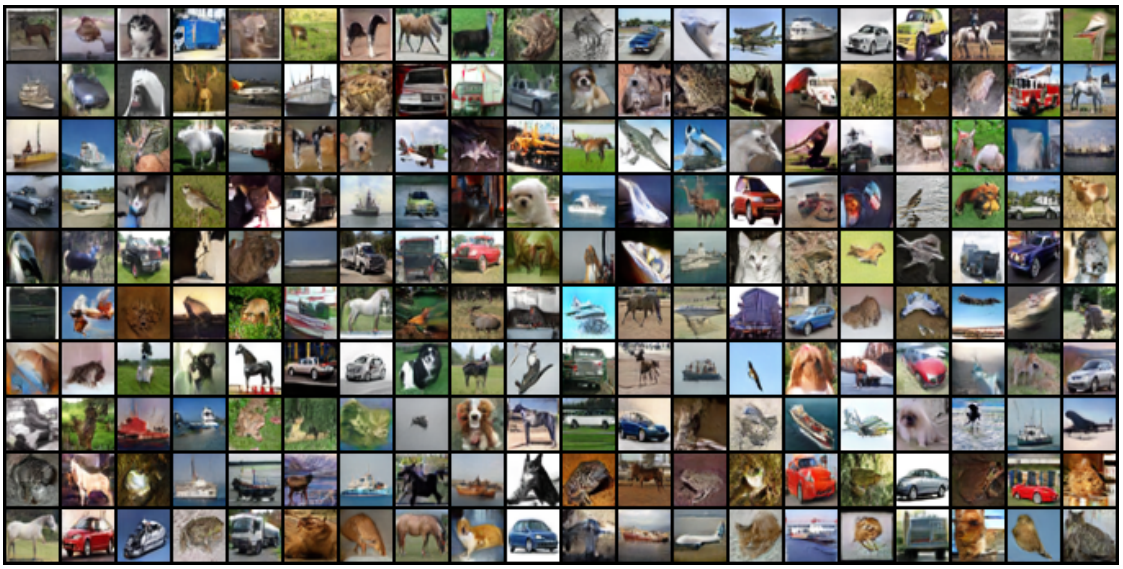


Figure 7. **Generated samples from S-JKO (JSD)** trained on CIFAR-10 (32×32) with *Large* model.

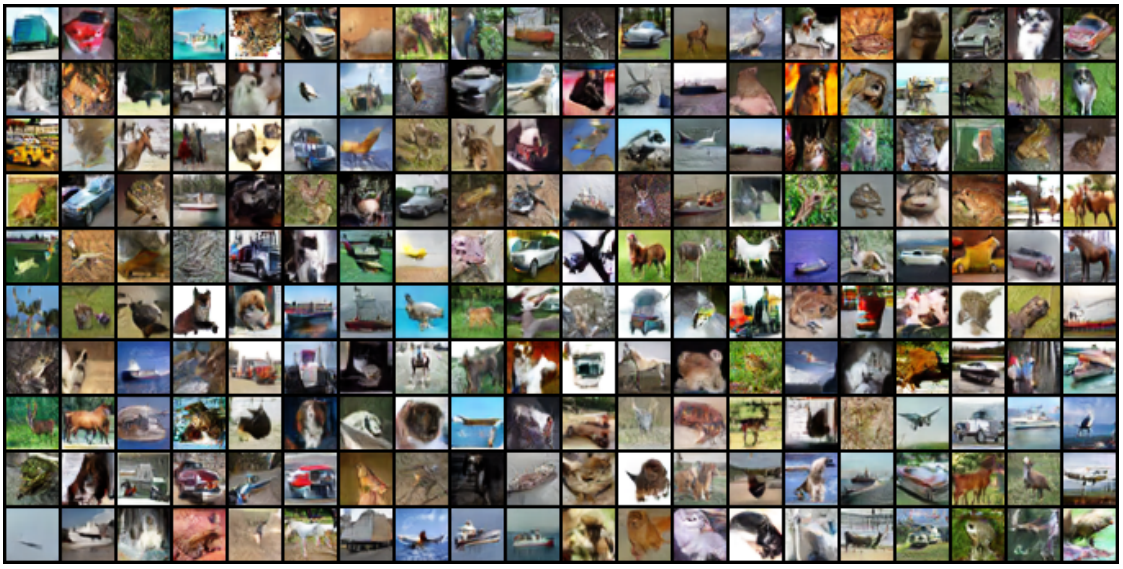


Figure 8. **Generated samples from S-JKO (KLD)** trained on CIFAR-10 (32×32) with *Small* model.

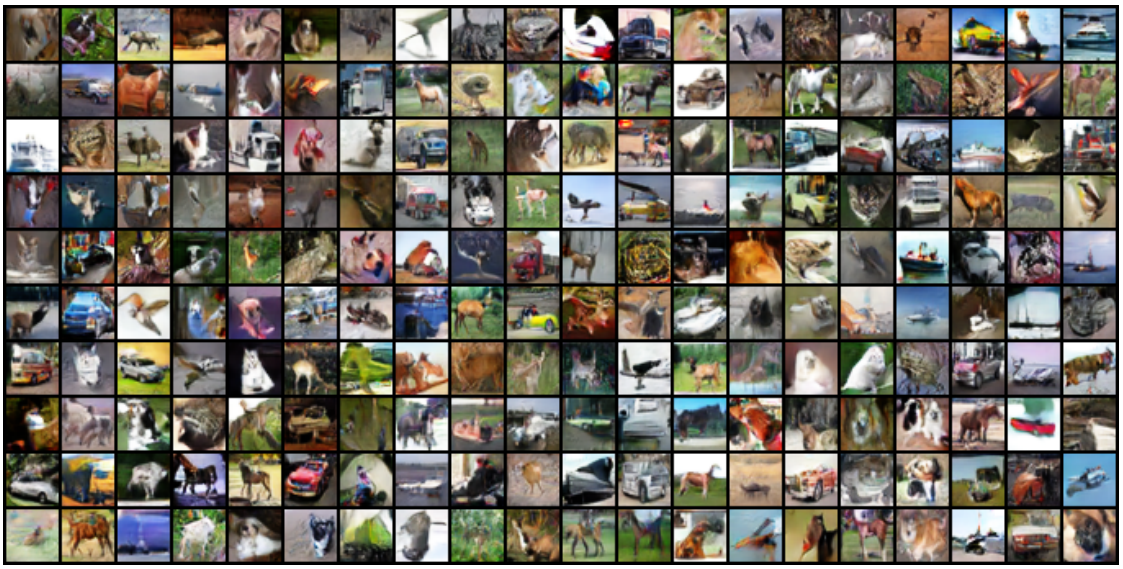


Figure 9. **Generated samples from S-JKO (JSD)** trained on CIFAR-10 (32×32) with *Small* model.

Figure 10. **Generated samples from S-JKO (KLD)** trained on CelebA-HQ (256×256).

Figure 11. **Generated samples from S-JKO (JSD)** trained on CelebA-HQ (256×256).

Table 2. **Extensive Comparison with Diverse Generative Models on Image Generation on CIFAR-10.** † indicates the results conducted by ourselves.

Class	Model	FID (↓)
GAN	SNGAN+DGflow (Ansari et al., 2020)	9.62
	AutoGAN (Gong et al., 2019)	12.4
	TransGAN (Jiang et al., 2021)	9.26
	StyleGAN2 w/o ADA (Karras et al., 2020)	8.32
	StyleGAN2 w/ ADA (Karras et al., 2020)	2.92
	DDGAN (T=1)(Xiao et al., 2021)	16.68
	DDGAN (Xiao et al., 2021)	3.75
	RGM (Choi et al., 2023c)	**2.47**
Diffusion	NCSN (Song & Ermon, 2019)	25.3
	DDPM (Ho et al., 2020)	3.21
	Score SDE (VE) (Song et al., 2021b)	2.20
	Score SDE (VP) (Song et al., 2021b)	2.41
	DDIM (50 steps) (Song et al., 2021a)	4.67
	CLD (Dockhorn et al., 2022)	2.25
	Subspace Diffusion (Jing et al., 2022)	2.17
	LSGM (Vahdat et al., 2021)	**2.10**
Flow Matching	FM (Lipman et al., 2023)	6.35
	OT-CFM (Tong et al., 2024)	**3.74**
VAE&EBM	NVAE (Vahdat & Kautz, 2020)	23.5
	Glow (Kingma & Dhariwal, 2018)	48.9
	PixelCNN (Van Oord et al., 2016)	65.9
	VAEBM (Xiao et al., 2020)	12.2
	Recovery EBM (Gao et al., 2021)	9.58
	CDRL-large (Zhu et al., 2024b)	**3.68**
OT-based	WGAN (Arjovsky et al., 2017)	55.20
	WGAN-GP(Gulrajani et al., 2017)	39.40
	OTM (*Large*)†	7.68
	Source-fixed UOTM (*Large*)	7.53
	UOTM (*Large*) (Choi et al., 2023a)	**2.97**
WGF-based	JKO-Flow (Fan et al., 2022)	23.1
	JKO-iFlow (Xu et al., 2023)	29.1
	NSGF (Zhu et al., 2024a) (*Large*)	5.55
	S-JKO (*Large*)†	**2.62**
	S-JKO (JSD) (*Large*)†	2.66