

NEURAL SINKHORN GRADIENT FLOW

Anonymous authors

Paper under double-blind review

ABSTRACT

Wasserstein Gradient Flows (WGF) with respect to specific functionals have been widely used in the machine learning literature. Recently, neural networks have been adopted to approximate certain intractable parts of the underlying Wasserstein gradient flow and result in efficient inference procedures. In this paper, we introduce the Neural Sinkhorn Gradient Flow (NSGF) model, which parametrizes the time-varying velocity field of the Wasserstein gradient flow w.r.t. the Sinkhorn divergence to the target distribution starting a given source distribution. We utilize the velocity field matching training scheme in NSGF, which only requires samples from the source and target distribution to compute an empirical velocity field approximation. Our theoretical analyses show that as the sample size increases to infinity, the mean-field limit of the empirical approximation converges to the true underlying velocity field. With specific source and target data samples, our NSGF models can be used in many machine learning tasks such as unconditional/conditional image generating, style transfer, and audio-text translations. Numerical experiments with synthetic and real-world benchmark datasets support our theoretical results and demonstrate the effectiveness of the proposed method.

1 INTRODUCTION

The Wasserstein Gradient Flow (WGF) with respect to certain specific functional objective \mathcal{F} (denoted as \mathcal{F} Wasserstein gradient flow) is a powerful tool for solving optimization problems over the Wasserstein probability space. Since the seminal work of Jordan et al. (1998) which shows that the Fokker-Plank equation is the Wasserstein gradient flow with respect to the free energy, Wasserstein gradient flow w.r.t. different functionals have been widely used in various machine learning tasks such as Bayesian inference (Liu, 2017; Liu et al., 2019; di Langosco et al., 2021; Zhang et al., 2021a), reinforcement learning (Zhang et al., 2018; Martin et al., 2020; Agazzi & Lu, 2020; Zhang et al., 2021b), and mean-field games (Domingo-Enrich et al., 2020; Gao et al., 2022b; Zhang & Katsoulakis, 2023).

One recent trend in the Wasserstein gradient flow literature is to develop efficient generative modeling methods (Gao et al., 2019; 2022a; Ansari et al., 2021; Mokrov et al., 2021; Alvarez-Melis et al., 2022; Bunne et al., 2022; Fan et al., 2022). In general, these methods mimic the Wasserstein gradient flow with respect to a specific distribution metric, driving a source distribution towards a target distribution. Neural networks are typically employed to approximate the computationally challenging components of the underlying Wasserstein gradient flow such as the time-dependent transport maps. During the training process of these methods, it is common to require samples from the target distribution. After the training process, an inference procedure is often employed to generate new samples from the target distribution. This procedure involves iteratively transporting samples from the source distribution with the assistance of the trained neural network. Based on the chosen metric, these methods can be categorized into two main types.

Divergences Between Distributions With Exact Same Supports. The first class of widely used metrics is the f-divergence, such as the Kullback-Leibler divergence and the Jensen-Shannon divergence. These divergences are defined based on the density ratio between two distributions and are only well-defined when dealing with distributions that have exactly the same support. Within the scope of f-divergence Wasserstein gradient flow generative models, neural networks are commonly utilized to formulate density-ratio estimators, as demonstrated by Gao et al. (2019; 2022a); Ansari et al. (2021) and Heng et al. (2022). However, as one can only access finite samples from target distributions in the training process, the support shift between the sample collections from

the compared distributions may cause significant approximation error in the density-ratio estimators (Rhodes et al., 2020; Choi et al., 2022). An alternative approach, proposed by Fan et al. (2022), circumvents these limitations by employing a dual variational formulation of the f-divergence. In this framework, two networks are employed to approximate the optimal variational function and the transport maps. These two components are optimized alternately. It’s imperative to highlight that the non-convex and non-concave characteristics of their min-max objective can render the training inherently unstable (Arjovsky & Bottou, 2017; Hsieh et al., 2021).

Divergences Between Distributions With Possible Different Supports. Another type of generative Wasserstein gradient flow model employs divergences that are well-defined for distributions with possible different supports. This includes free energy functionals (Mokrov et al., 2021; Alvarez-Melis et al., 2022; Bunne et al., 2022), the kernel-based metrics such as the Maximum-Mean/Sobolev Discrepancy (Mroueh et al., 2019; Mroueh & Rigotti, 2020; Altekruiger et al., 2023) and sliced-Wasserstein distance (Liutkus et al., 2019; Du et al., 2023). As these divergences can be efficiently approximated with samples, neural networks are typically used to directly model the transport maps used in the inference procedure. In Wasserstein gradient flow methods, input convex neural networks (ICNNs, Amos et al. (2017)) are commonly used to approximate the transport map. However, recently, several works (Bonet et al., 2021; Korotin et al., 2021) discuss the poor expressiveness of ICNNs architecture and show that it would result in poor performance in high-dimension applications. Besides, the Maximum-Mean/Sobolev discrepancy Wasserstein gradient flow models are usually hard to train and are easy to be trapped in poor local optima in practice (Arbel et al., 2019), since the kernel-based divergences are highly sensitive to the parameters of the kernel function (Li et al., 2017; Wang et al., 2018). Liutkus et al. (2019); Du et al. (2023) consider sliced-Wasserstein WGF to build nonparametric generative Models which do not achieve high generation quality, it is an interesting work on how to combine sliced-Wasserstein WGF and neural network methods.

Contribution. In this paper, we investigate the Wasserstein gradient flow with respect to the Sinkhorn divergence, which is categorized under the second type of divergences and does not necessitate any kernel functions. We introduce the *Neural Sinkhorn Gradient Flow (NSGF)* model, which parametrizes the time-varying velocity field of the Sinkhorn Wasserstein gradient flow from a specified source distribution. The NSGF employs a velocity field matching scheme that demands only samples from the target distribution to calculate empirical velocity field approximations. Our theoretical analyses show that as the sample size approaches infinity, the mean-field limit of the empirical approximation converges to the true velocity field of the Sinkhorn Wasserstein gradient flow. Given distinct source and target data samples, our NSGF can be harnessed across a wide range of machine learning applications, including unconditional/conditional image generation, style transfer, and audio-text translation. We empirically validate NSGF on low-dimensional 2D data and benchmark images (MNIST, CIFAR-10). Our findings indicate that our models can be trained to yield commendable results in terms of generation cost and sample quality, surpassing the performance of the neural Wasserstein gradient flow methods previously tested on CIFAR-10, to the best of our knowledge.

2 RELATED WORKS

Sinkhorn Divergence in Machine Learning. Originally introduced in the domain of optimal transport, the Sinkhorn divergence emerged as a more computationally tractable alternative to the classical Wasserstein distance (Cuturi, 2013; Peyré et al., 2017; Feydy et al., 2019). Since its inception, Sinkhorn divergence has found applications across a range of machine learning tasks, including domain adaptation (Courty et al., 2014; Alaya et al., 2019; Komatsu et al., 2021), Sinkhorn barycenter (Luisse et al., 2019; Shen et al., 2020) and color transfer (Blondel et al., 2018; Pai et al., 2021). Indeed, it has already been extended to single-step generative modeling methods, such as the Sinkhorn GAN and VAE (Genevay et al., 2018; Deja et al., 2020; Patrini et al., 2020). However, to the best of our knowledge, it has yet to be employed in developing efficient generative Wasserstein gradient flow models.

Neural ODE/SDE Based Diffusion Models. Recently, diffusion models, as a class of Neural ODE/SDE Based generative methods have achieved unprecedented success, which also transforms a simple density to the target distribution, iteratively (Song & Ermon, 2019; Ho et al., 2020; Song et al., 2021). Typically, each step of diffusion models only progresses a little by denoising a sim-

ple Gaussian noise, while each step in WGF models follows the most informative direction (in a certain sense). Hence, diffusion models usually have a long inference trajectory. In recent research undertakings, there has been a growing interest in exploring more informative steps within diffusion models. Specifically, flow matching methods (Lipman et al., 2023; Liu et al., 2023; Albergo & Vanden-Eijnden, 2023) establish correspondence between the source and target via optimal transport, subsequently crafting a probability path by directly linking data points from both ends. Notably, when the source and target are both Gaussians, their path is actually a Wasserstein gradient flow. However, this property does not consistently hold for general data probabilities. Moreover, Tong et al. (2023); Pooladian et al. (2023) consider calculating the minibatch optimal transport map to guide data points connecting. Besides, Das et al. (2023) consider the shortest forward diffusion path for the Fisher metric and Shaul et al. (2023) explore the conditional Gaussian probability path based on the principle of minimizing the Kinetic Energy. Nonetheless, a commonality among many of these methods is their reliance on Gaussian paths for theoretical substantiation, thereby constraining the broader applicability of these techniques within real-world generative modeling.

3 PRELIMINARIES

3.1 NOTATIONS

We denote $\mathbf{x} = (x_1, \dots, x_d) \in \mathbb{R}^d$ and $\mathcal{X} \subset \mathbb{R}^d$ as a vector and a compact ground set in \mathbb{R}^d , respectively. For a given point $\mathbf{x} \in \mathcal{X}$, $\|\mathbf{x}\|_p := (\sum_i x_i^p)^{\frac{1}{p}}$ denotes the p -norm on euclidean space, and $\delta_{\mathbf{x}}$ stands for the Dirac (unit mass) distribution at point $\mathbf{x} \in \mathcal{X}$. $\mathcal{P}_2(\mathcal{X})$ denotes the set of probability measures on \mathcal{X} with finite second moment and $\mathcal{C}(\mathcal{X})$ denotes the space of continuous functions on \mathcal{X} . For a given functional $\mathcal{F}(\cdot) : \mathcal{P}_2(\mathcal{X}) \rightarrow \mathbb{R}$, $\frac{\delta \mathcal{F}(\mu_t)}{\delta \mu}(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}$ denotes its first variation at $\mu = \mu_t$. Besides, we use ∇ and $\nabla \cdot (\cdot)$ to denote the gradient and the divergence operator, respectively.

3.2 WASSERSTEIN DISTANCE AND SINKHORN DIVERGENCE

We first introduce the background of Wasserstein distance. Given two probability measures $\mu, \nu \in \mathcal{P}_2(\mathcal{X})$, the p -Wasserstein distance $\mathcal{W}_p(\mu, \nu) : \mathcal{P}_2(\mathcal{X}) \times \mathcal{P}_2(\mathcal{X}) \rightarrow \mathbb{R}_+$ is defined as:

$$\mathcal{W}_p(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \left(\int_{\mathcal{X} \times \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|^p d\pi(\mathbf{x}, \mathbf{y}) \right)^{\frac{1}{p}}, \quad (1)$$

where $\Pi(\mu, \nu)$ denotes the set of all probability couplings π with marginals μ and ν . The \mathcal{W}_p distance aims to find a coupling π so as to minimize the cost function $\|\mathbf{x} - \mathbf{y}\|^p$ of moving a probability mass from μ to ν . It has been demonstrated that the p -Wasserstein distance is a valid metric on $\mathcal{P}_2(\mathcal{X})$, and $(\mathcal{P}_2(\mathcal{X}), \mathcal{W}_p)$ is referred to as the Wasserstein probability space (Villani et al., 2009).

Note that directly calculating \mathcal{W}_p is computationally expensive, especially for high dimensional problems (Santambrogio, 2015). Consequently, the entropy-regularized Wasserstein distance (Cuturi, 2013) is proposed to approximate equation 1 by regularizing the original problem with an entropy term:

Definition 1. *The entropy-regularized Wasserstein distance is formally defined as:*

$$\mathcal{W}_{p,\varepsilon}(\mu, \nu) = \inf_{\pi \in \Pi(\mu, \nu)} \left[\left(\int_{\mathcal{X} \times \mathcal{X}} \|\mathbf{x} - \mathbf{y}\|^p d\pi(\mathbf{x}, \mathbf{y}) \right)^{\frac{1}{p}} + \varepsilon KL(\pi | \mu \otimes \nu) \right], \quad (2)$$

where $\varepsilon > 0$ is a regularization coefficient, $\mu \otimes \nu$ denotes the product measure, i.e., $\mu \otimes \nu(\mathbf{x}, \mathbf{y}) = \mu(\mathbf{x})\nu(\mathbf{y})$, and $KL(\pi | \mu \otimes \nu)$ denotes the KL-divergence between π and $\mu \otimes \nu$.

Generally, the computational cost of $\mathcal{W}_{p,\varepsilon}$ is much lower than \mathcal{W}_p , and can be efficiently calculated with Sinkhorn algorithms (Cuturi, 2013). Without loss of generality, we fix $p = 2$ and abbreviate $\mathcal{W}_{2,\varepsilon} := \mathcal{W}_\varepsilon$ for ease of notion in the whole paper. According to Fenchel-Rockafellar theorem, the entropy-regularized Wasserstein problem \mathcal{W}_ε equation 2 has an equivalent dual formulation, which is given as follows Peyré et al. (2017):

$$\mathcal{W}_\varepsilon(\mu, \nu) = \max_{f, g \in \mathcal{C}(\mathcal{X})} \langle \mu, f \rangle + \langle \nu, g \rangle - \varepsilon \left\langle \mu \otimes \nu, \exp \left(\frac{1}{\varepsilon} (f \oplus g - C) \right) - 1 \right\rangle, \quad (3)$$

where C is the cost function in 2 and $f \oplus g$ is the tensor sum: $(x, y) \in \mathcal{X}^2 \mapsto f(x) + g(y)$. The maximizers $f_{\mu, \nu}$ and $g_{\mu, \nu}$ of 3 are called the \mathcal{W}_ε -potentials of $\mathcal{W}_\varepsilon(\mu, \nu)$. The following lemma states the optimality condition for the \mathcal{W}_ε -potentials:

Lemma 1. (*Optimality Cuturi (2013)*) *The \mathcal{W}_ε -potentials $(f_{\mu, \nu}, g_{\mu, \nu})$ exist and are unique $(\mu, \nu) -$ a.e. up to an additive constant (i.e. $\forall K \in \mathbb{R}, (f_{\mu, \nu} + K, g_{\mu, \nu} - K)$ is optimal). Moreover,*

$$\mathcal{W}_\varepsilon(\mu, \nu) = \langle \mu, f_{\mu, \nu} \rangle + \langle \nu, g_{\mu, \nu} \rangle. \quad (4)$$

We describe such the method in Appendix A for completeness. Note that, although computationally more efficient than the \mathcal{W}_p distance, the \mathcal{W}_ε distance is not a true metric, as there exists $\mu \in \mathcal{P}_2(\mathcal{X})$ such that $\mathcal{W}_\varepsilon(\mu, \mu) \neq 0$ when $\varepsilon \neq 0$, which restricts the applicability of \mathcal{W}_ε . As a result, the following Sinkhorn divergence $\mathcal{S}_\varepsilon(\mu, \nu) : \mathcal{P}_2(\mathcal{X}) \times \mathcal{P}_2(\mathcal{X}) \rightarrow \mathbb{R}$ is proposed (Peyré et al., 2017):

Definition 2. *Sinkhorn divergence:*

$$\mathcal{S}_\varepsilon(\mu, \nu) = \mathcal{W}_\varepsilon(\mu, \nu) - \frac{1}{2} (\mathcal{W}_\varepsilon(\mu, \mu) + \mathcal{W}_\varepsilon(\nu, \nu)). \quad (5)$$

$\mathcal{S}_\varepsilon(\mu, \nu)$ is nonnegative, bi-convex thus a valid metric on $\mathcal{P}_2(\mathcal{X})$ and metrize the convergence in law. Actually $\mathcal{S}_\varepsilon(\mu, \nu)$ interpolates the Wasserstein distance ($\varepsilon \rightarrow 0$) and the Maximum Mean Discrepancy ($\varepsilon \rightarrow \infty$) (Feydy et al., 2019).

3.3 GRADIENT FLOWS

Consider an optimization problem over $\mathcal{P}_2(\mathcal{X})$:

$$\min_{\mu \in \mathcal{P}_2(\mathcal{X})} \mathcal{F}(\mu) := \mathcal{D}(\mu | \mu^*). \quad (6)$$

where μ^* is the target distribution, \mathcal{D} is the divergence we choose. We consider now the problem of transporting mass from an initial distribution μ_0 to a target distribution μ^* , by finding a continuous probability path μ_t starting from $\mu_0 = \mu$ that converges to μ^* while decreasing $\mathcal{F}(\mu_t)$. To solve this optimization problem, one can consider a descent flow of $\mathcal{F}(\mu)$ in the Wasserstein space, which transports any initial distribution μ_0 towards the target distribution μ^* . Specifically, the descent flow of $\mathcal{F}(\mu)$ is described by the following continuity equation (Ambrosio et al. (2005), Villani et al. (2009), Santambrogio (2017)):

$$\frac{\partial \mu_t(x)}{\partial t} = -\nabla \cdot (\mu_t(x) \mathbf{v}_t(x)). \quad (7)$$

where $\mathbf{v}_{\mu_t} : \mathcal{X} \rightarrow \mathcal{X}$ is a velocity field that defines the direction of position transportation. To ensure a descent of $\mathcal{F}(\mu_t)$ over time t , the velocity field \mathbf{v}_{μ_t} should satisfy the following inequality ((Ambrosio et al., 2005)):

$$\frac{d\mathcal{F}(\mu_t)}{dt} = \int \left\langle \nabla \frac{\delta \mathcal{F}(\mu_t)}{\delta \mu}, \mathbf{v}_t \right\rangle d\mu_t \leq 0. \quad (8)$$

A straightforward choice of \mathbf{v}_t is $\mathbf{v}_t = -\nabla \frac{\delta \mathcal{F}(\mu_t)}{\delta \mu}$, which is actually the steepest descent direction of $\mathcal{F}(\mu_t)$. When we select this \mathbf{v}_t , we refer to the aforementioned continuous equation as the *Wasserstein gradient flow* of \mathcal{F} . We give the definition of the first variation in the appendix for the sake of completeness of the article.

4 METHODOLOGY

In this section, we first introduce the Sinkhorn Wasserstein gradient flow and investigate its convergence properties. Then, we develop our Neural Sinkhorn Gradient Flow model, which consists of a velocity field matching training procedure and a velocity field guided inference procedure. Moreover, we theoretically show that the mean-field limit of the empirical approximation used in the training procedure converges to the true velocity field of the Sinkhorn Wasserstein gradient flow.

4.1 SINKHORN WASSERSTEIN GRADIENT FLOW

Based on the definition of the Sinkhorn divergence, we construct our Sinkhorn objective $\mathcal{F}_\varepsilon(\cdot) = \mathcal{S}_\varepsilon(\cdot, \mu^*)$, where μ^* denotes the target distribution. The following theorem gives the first variation of the Sinkhorn objective.

Theorem 1. (*First variation of the Sinkhorn objective Luise et al. (2019)*) Let $\varepsilon > 0$. Let $(f_{\mu, \mu^*}, g_{\mu, \mu^*})$ be the \mathcal{W}_ε -potentials of $\mathcal{W}_\varepsilon(\mu, \mu^*)$ and $(f_{\mu, \mu}, g_{\mu, \mu})$ be the \mathcal{W}_ε -potentials of $\mathcal{W}_\varepsilon(\mu, \mu)$. The first variation of the Sinkhorn objective \mathcal{F}_ε is

$$\frac{\delta \mathcal{F}_\varepsilon}{\delta \mu} = f_{\mu, \mu^*} - f_{\mu, \mu}. \quad (9)$$

According to Theorem 1, we can construct the Sinkhorn Wasserstein gradient flow by setting the velocity field \mathbf{v}_t in the continuity equation equation 7 as $\mathbf{v}_{\mu_t}^{\mathcal{F}_\varepsilon} = -\nabla \frac{\delta \mathcal{F}_\varepsilon(\mu_t)}{\delta \mu_t} = \nabla f_{\mu_t, \mu_t} - \nabla f_{\mu_t, \mu^*}$.

Proposition 1. Consider the Sinkhorn Wasserstein gradient flow described by the following continuity equation:

$$\frac{\partial \mu_t(\mathbf{x})}{\partial t} = -\nabla \cdot (\mu_t(\mathbf{x})(\nabla f_{\mu_t, \mu_t}(\mathbf{x}) - \nabla f_{\mu_t, \mu^*}(\mathbf{x}))). \quad (10)$$

The following local descending property of \mathcal{F}_ε holds:

$$\frac{d\mathcal{F}_\varepsilon(\mu_t)}{dt} = - \int \|\nabla f_{\mu_t, \mu_t}(\mathbf{x}) - \nabla f_{\mu_t, \mu^*}(\mathbf{x})\|^2 d\mu_t, \quad (11)$$

where the r.h.s. equals 0 if and only if $\mu_t = \mu^*$.

4.2 VELOCITY-FIELDS MATCHING

We now present our NSGF method, the core of which lies in training a neural network to approximate the time-varying velocity field $\mathbf{v}_{\mu_t}^{\mathcal{F}_\varepsilon}$ induced by Sinkhorn Wasserstein gradient flow. Given a target probability density path $\mu_t(x)$ and its corresponding velocity field $\mathbf{v}_{\mu_t}^{S_\varepsilon}$, which generates $\mu_t(x)$, we define the velocity field matching objective as follows:

$$\min_{\theta} \mathbf{E}_{t \sim [0, T], x \sim \mu_t} \left[\|\mathbf{v}^\theta(x, t) - \mathbf{v}_{\mu_t}^{S_\varepsilon}(x)\|^2 \right]. \quad (12)$$

To construct our algorithm, we utilize independently and identically distributed (i.i.d) samples denoted as $\{Y_i\}_{i=1}^n \in \mathbb{R}^d$, which are drawn from an unknown target distribution μ^* a common practice in the field of generative modeling. Given the current set of samples $\{\tilde{X}_i^t\}_{i=1}^n \sim \mu_t$, our method calculates the velocity field using the \mathcal{W}_ε -potentials (Lemma 1) $f_{\tilde{\mu}_t, \tilde{\mu}^*}$ and $f_{\tilde{\mu}_t, \tilde{\mu}_t}$ based on samples. Here, $\tilde{\mu}_t$ and $\tilde{\mu}^*$ represent discrete Dirac distributions. Note that these potentials defined on discrete measures can be calculated efficiently with first-order stochastic gradient descent methods such as SGD and ADAM (Bottou et al., 2018; Kingma & Ba, 2014). The corresponding finite sample velocity field approximation can be computed as follows:

$$\hat{\mathbf{v}}_{\tilde{\mu}_t}^{\mathcal{F}_\varepsilon}(\tilde{X}_i^t) = \nabla_{\tilde{X}_i^t} f_{\tilde{\mu}_t, \tilde{\mu}_t}(\tilde{X}_i^t) - \nabla_{\tilde{X}_i^t} f_{\tilde{\mu}_t, \tilde{\mu}^*}(\tilde{X}_i^t). \quad (13)$$

Subsequently, we derive the particle formulation corresponding to the flow formulation 10.

$$d\tilde{X}_i^t = \hat{\mathbf{v}}_{\tilde{\mu}_t}^{\mathcal{F}_\varepsilon}(\tilde{X}_i^t) dt, i = 1, 2, \dots, n. \quad (14)$$

In the following proposition, we investigate the mean-field limit of the particle set $\{\tilde{X}_i^t\}_{i=1, \dots, M}$.

Theorem 2. (*Mean-field limits.*) Suppose the empirical distribution $\tilde{\mu}_0$ of M particles weakly converges to a distribution μ_0 when $M \rightarrow \infty$. Then, the path of equation 14 starting from $\tilde{\mu}_0$ weakly converges to a solution of the following partial differential equation starting from μ_0 when $M \rightarrow \infty$:

$$\frac{\partial \mu_t(x)}{\partial t} = -\nabla \cdot (\mu_t(x) \nabla \frac{\delta \mathcal{F}_\varepsilon(\mu_t)}{\delta \mu_t}). \quad (15)$$

which is actually the gradient flow of Sinkhorn divergence \mathcal{F}_ε in the Wasserstein space.

The following proposition shows that the goal of the velocity field matching objective equation 12 can be regarded as approximating the steepest local descent direction with neural networks.

Proposition 2. (*Steepest local descent direction.*) Consider the infinitesimal transport $T(x) = x + \lambda\phi$. The Fréchet derivative under this particular perturbation,

$$\begin{aligned} \frac{d}{d\lambda} \mathcal{F}_\varepsilon(T_\# \mu)|_{\lambda=0} &= \lim_{\lambda \rightarrow 0} \frac{\mathcal{F}_\varepsilon(T_\# \mu) - \mathcal{F}_\varepsilon(\mu)}{\lambda} \\ &= \int_{\mathcal{X}} \nabla f_{\mu, \mu^*}(\mathbf{x}) \phi(\mathbf{x}) d\mu - \int_{\mathcal{X}} \nabla f_{\mu, \mu}(\mathbf{x}) \phi(\mathbf{x}) d\mu, \end{aligned} \quad (16)$$

and the steepest local descent direction is $\phi = \frac{\nabla f_{\mu, \mu^*}(\mathbf{x}) - f_{\mu, \mu}(\mathbf{x})}{\|\nabla f_{\mu, \mu^*}(\mathbf{x}) - f_{\mu, \mu}(\mathbf{x})\|}$.

The velocity field matching training procedure is outlined in Algorithm 1. Considering the balance between expensive training costs and training quality, we opted to first build a trajectory pool of Sinkhorn gradient flow and then sample from it to construct the velocity field matching algorithm. Our method draws inspiration from experience replay, a common technique in reinforcement learning, adapting it to enhance our model’s effectiveness (Mnih et al., 2013; Silver et al., 2016). Once we calculate the time-varying velocity field $\hat{\mathbf{v}}_{\mu_t}^s(\tilde{X}_i^t)$, we can parameterize the velocity field using a straightforward regression method.

Remark 1. In the discrete case, \mathcal{W}_ε -potentials 1 can be computed by a standard method in Genevay et al. (2016). In practice, we use the efficient implementation of the Sinkhorn algorithm with GPU acceleration from the GeomLoss package (Feydy et al., 2019).

Algorithm 1: Velocity field matching training

Input : number of time steps T , batch size n , gradient flow step size $\eta > 0$, empirical or samplable distribution μ_0 and μ^* , neural network parameters θ , optimizer step size $\gamma > 0$

/ Build trajectory pool */*

while Building **do**

/ Sample batches of size n i.i.d. from the datasets */*

$\tilde{X}_i^0 \sim \mu_0, \quad \tilde{Y}_i \sim \mu^*, i = 1, 2, \dots, n.$

for $t = 0, 1, \dots, T$ **do**

 calculate $f_{\tilde{\mu}_t, \tilde{\mu}_t}(\tilde{X}_i^t), f_{\tilde{\mu}_t, \tilde{\mu}^*}(\tilde{X}_i^t)$ using the GeomLoss package .

$\hat{\mathbf{v}}_{\mu_t}^{\mathcal{F}_\varepsilon}(\tilde{X}_i^t) = \nabla f_{\tilde{\mu}_t, \tilde{\mu}_t}(\tilde{X}_i^t) - \nabla f_{\tilde{\mu}_t, \tilde{\mu}^*}(\tilde{X}_i^t).$

$\tilde{X}_i^{t+1} = \tilde{X}_i^t + \eta \hat{\mathbf{v}}_{\mu_t}^{\mathcal{F}_\varepsilon}(\tilde{X}_i^t).$

 store all $(\tilde{X}_i^t, \hat{\mathbf{v}}_{\mu_t}^{\mathcal{F}_\varepsilon}(\tilde{X}_i^t))$ pair into the pool, $i = 1, 2, \dots, n, \quad t = 0, 1, \dots, T.$

/ velocity field matching */*

while Not convergence **do**

 from trajectory pool sample pair $(\tilde{X}_i^t, \hat{\mathbf{v}}_{\mu_t}^{\mathcal{F}_\varepsilon}(\tilde{X}_i^t)).$

$\mathcal{L}(\theta) = \left\| \mathbf{v}^\theta(\tilde{X}_i^t, t) - \hat{\mathbf{v}}_{\mu_t}^{\mathcal{F}_\varepsilon}(\tilde{X}_i^t) \right\|^2,$

$\theta \leftarrow \theta - \gamma \nabla_\theta \mathcal{L}(\theta).$

Output: θ parameterize the time-varying velocity field

Once obtained a feasible velocity field approximation \mathbf{v}^θ , one can generate new samples by iteratively employing the explicit Euler discretization of the Equation 14 to drive the samples to the target. More details can be found in appendix D. Note that various other numerical schemes, such as the implicit Euler method (Platen & Bruti-Liberati, 2010) and higher-order Runge-Kutta methods (Butcher, 1964), can be employed. In this study, we opt for the first-order explicit Euler discretization method (Süli & Mayers, 2003) due to its simplicity and ease of implementation. We leave the exploration of higher-order algorithms for future research.

Algorithm	2-Wasserstein distance (10 steps)					2-Wasserstein distance (100 steps)				
	8gaussians	8gaussians-moons	moons	scurve	checkerboard	8gaussians	8gaussians-moons	moons	scurve	checkerboard
NSGF(ours)	0.285	0.144	0.077	0.117	0.252	0.278	0.144	0.067	0.110	0.147
JKO-Flow	0.290	0.177	0.085	0.135	0.269	0.274	0.167	0.085	0.123	0.160
EPT	0.295	0.180	0.082	0.138	0.277	0.289	0.176	0.080	0.118	0.163
OT-CFM	0.289	0.173	0.088	0.149	0.253	0.269	0.165	0.078	0.127	0.159
1-RF	0.427	0.294	0.107	0.169	0.396	0.415	0.293	0.099	0.136	0.166
2-RF	0.428	0.311	0.125	0.171	0.421	0.430	0.311	0.121	0.136	0.170
3-RF	0.421	0.298	0.110	0.170	0.413	0.414	0.297	0.103	0.140	0.170
SI	0.435	0.324	0.134	0.187	0.427	0.411	0.294	0.096	0.139	0.166
FM	0.423	0.292	0.111	0.171	0.417	0.415	0.290	0.097	0.135	0.165

Table 1: Comparison of neural gradient-flow-based methods and neural ODE-based diffusion models over five data sets. We provide the generated result in 10/100 Euler steps for Neural ODE Models, EPT, and our methods. The principle of steps in JKO-flow means backward Eulerian method steps (JKO steps).

5 EXPERIMENTS

We conducted an empirical investigation of the *Neural Sinkhorn Gradient Flow (NSGF)* method across a range of experiments. Initially, we demonstrate how NSGF guides the evolution and convergence of particles from the initial distribution toward the target distribution in 2D distribution experiments. Subsequently, we shift our focus to benchmarking on high-dimensional datasets, specifically the MNIST dataset (LeCun et al. (1998)) and the CIFAR-10 dataset (Krizhevsky et al. (2009)). Our method’s adaptability to high-dimensional spaces is exemplified through experiments conducted on these datasets.

5.1 2D SIMULATE DATA

We assess the performance of various generative modeling models in low dimensions. Specifically, we conduct a comparative analysis between our method, NSGF, and several neural ODE-based diffusion models, including Flow Matching (FM; Lipman et al. (2023)), Rectified Flow (1,2,3-RF; Liu et al. (2023)), Optimal Transport Condition Flow Matching (OT-CFM; Tong et al. (2023); Pooladian et al. (2023)), Stochastic Interpolant (SI; Albergo & Vanden-Eijnden (2023)), and neural gradient-flow-based models such as JKO-Flow (Fan et al., 2022) and EPT (Gao et al., 2022a). Our evaluation involves learning 2D distributions adapted from Grathwohl et al. (2018), which include multiple modes.

Table 1 provides a comprehensive overview of our 2D experimental results, clearly illustrating the generalization capabilities of NSGF. Even when employing fewer steps. It is evident that neural gradient-flow-based models consistently outperform neural ODE-based diffusion models, particularly in low-step settings. This observation suggests that neural gradient-flow-based models generate more informative paths, enabling effective generation with a reduced number of steps. Furthermore, our results showcase the best performances among neural gradient-flow-based models, indicating that we have successfully introduced a lower error in approximating Wasserstein gradient flows. More complete details of the experiment can be found in the appendix E In the absence of specific additional assertions, we adopted Euler steps as the inference steps.

We present additional comparisons between neural ODE-based diffusion models and neural gradient-flow-based models, represented by NSGF and EPT, in Figure 1, 2, which illustrates the flow at different steps from 0 to T . Our observations reveal that the velocity field induced by NSGF exhibits notably high-speed values right from the outset. This is attributed to the fact that NSGF follows the steepest descent direction within the probability space. In contrast, neural ODE-based diffusion models, particularly those based on stochastic interpolation, do not follow the steepest descent path in 2D experiments. Even with the proposed rectified flow method by Liu et al. (2023) to straighten the path, these methods still necessitate more steps to reach the desired outcome.

5.2 IMAGE BENCHMARK DATA

In this section, we illustrate the scalability of our algorithm to the high-dimensional setting by applying our methods on real image datasets where only samples from the target distribution are accessible.

Generative modeling for MNIST We evaluate NSGF on MNIST to show our generating ability. Figure 3 shows samples and their trajectories starting from Gaussian noise to target distribution and

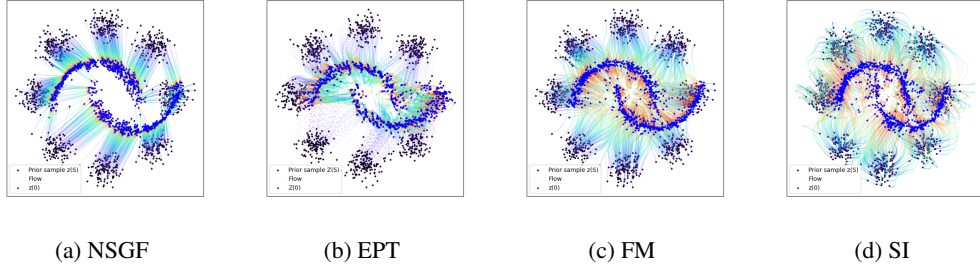


Figure 1: Visualization results for 2D generated paths. We show different methods that drive the particle from the prior distribution (black) to the target distribution (blue). The color change of the flow shows the different number of steps (from blue to red means from 0 to T).

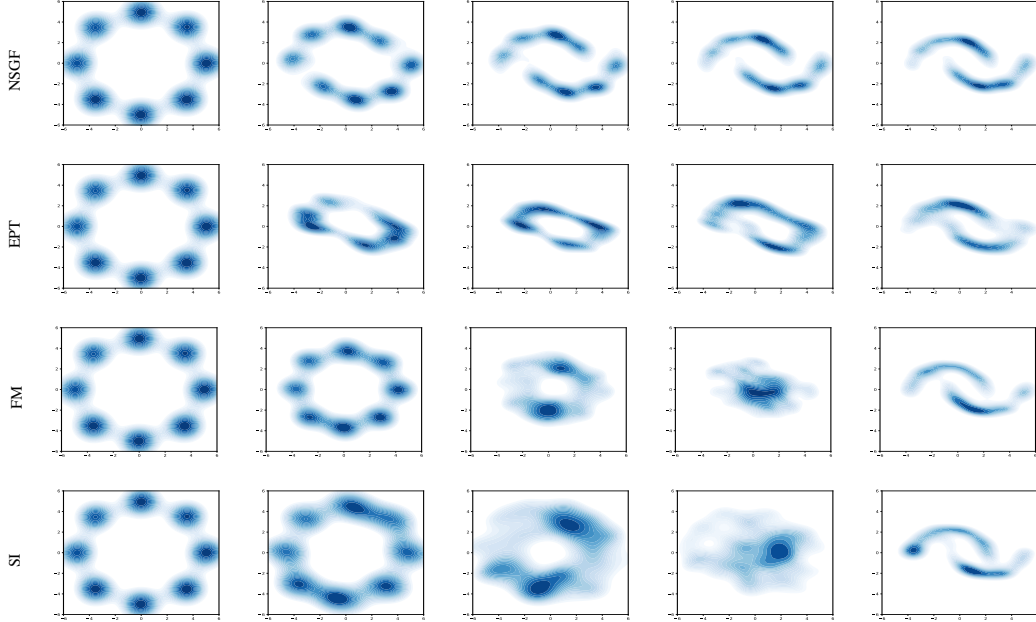


Figure 2: 2-Wasserstein Distance of the generated process utilizing neural ODE-based diffusion models and NSGF. The FM/SI methods reduce noise roughly linearly, while NSGF quickly recovers the target structure and progressively optimizes the details in subsequent steps which cause more meaningful interpolation between initial and target distributions.

demonstrates NSGF can approximate Sinkhorn gradient flow in image space empirically. We also provide sample quality using the standard Fréchet Inception Distance (FID) (Heusel et al., 2017) comparing with nonparametric gradient-flow-based methods SWGF (Sliced Wasserstein gradient flow, Liutkus et al. (2019)) and normalizing flows method SIG (Sliced iterative normalizing flows, Dai & Seljak (2020)) on appendix 3.

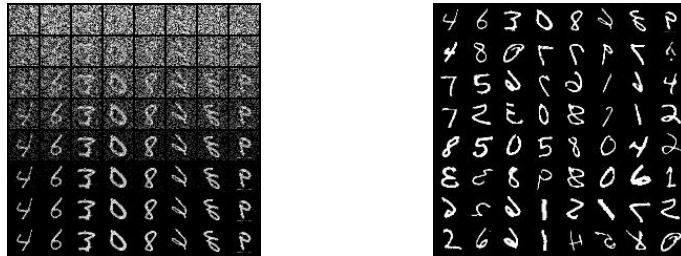


Figure 3: Trajectories and uncensored samples of our methods on MNIST

Generative modeling for CIFAR-10 We report sample quality using the standard Fréchet Inception Distance (FID) (Heusel et al., 2017), Inception Score (IS) (Salimans et al., 2016) and compute

Algorithm	CIFAR 10		
	IS(\uparrow)	FID(\downarrow)	NFE(\downarrow)
NSGF(ours)	7.56	21.6	100
EPT(Gao et al., 2022a)	/	46.63	10k
JKO-Flow(Fan et al., 2022)	7.48	23.7	>150
DGGF(Heng et al., 2022)	/	28.12	110
OT-CFM(Tong et al., 2023)	/	11.139	100
FM(Lipman et al., 2023)	/	6.35	142
1-RF(Liu et al., 2023)	9.60	2.58	127
2-RF(Liu et al., 2023)	9.24	3.36	110
3-RF(Liu et al., 2023)	9.01	3.96	104
SI(Albergo & Vanden-Eijnden, 2023)	/	10.27	/

Table 2: Comparison of Neural Wasserstein gradient flow methods and Neural ODE-based diffusion models over CIFAR-10

cost using the number of function evaluations (NFE). These are all standard metrics throughout the literature.

Table 2 presents the results, including the Fréchet Inception Distance (FID), Inception Score (IS), and the number of function evaluations (NFE), comparing the empirical distribution generated by each algorithm with the target distribution. While our current implementation may not yet rival state-of-the-art methods, it demonstrates promising outcomes, particularly in terms of generating quality (FID), outperforming neural gradient-flow-based models (EPT, Gao et al. (2022a); JKO-Flow, Fan et al. (2022); DGGF, (LSIF- \mathcal{X}^2) Heng et al. (2022)) with fewer steps. It’s essential to emphasize that this work represents an initial exploration of this particular model category and has not undergone optimization using common training techniques found in recent diffusion-based approaches. Such techniques include the use of exponential moving averages, truncations, learning rate warm-ups, and similar strategies. Furthermore, it’s worth noting that training neural gradient-flow-based models like NSGF in high-dimensional spaces can be challenging. Balancing the optimization of per-step information with the limitations of the neural network’s expressive power presents an intriguing research avenue that warrants further investigation. Figure 4 shows the trajectories of rectified flow (Liu et al., 2023) and NSGF. We still observe that NSGF quickly recovers the target structure and progressively optimizes the details in subsequent steps compared to the rectified flow.



Figure 4: Trajectories comparison between the rectified flow and the NSGF model in CIFAR-10 task. The top two rows show the trajectories of rectified flow and the bottom two rows show the trajectories of the NSGF model. We can see NSGF model quickly recovers the target structure and progressively optimizes the details in subsequent steps

6 CONCLUSION

This paper delves into the realm of Wasserstein gradient flow w.r.t. the Sinkhorn divergence as an alternative to kernel methods. Our main investigation revolves around the Neural Sinkhorn Gradient Flow (NSGF) model, which introduces a parameterized velocity field that evolves over time in the Sinkhorn gradient flow. One noteworthy aspect of the NSGF is its efficient velocity field matching, which relies solely on samples from the target distribution for empirical approximations. The combination of rigorous theoretical foundations and empirical observations demonstrates that our approximations of the velocity field converge toward their true counterparts as the sample sizes grow. Through extensive empirical experiments on well-known datasets like MNIST and CIFAR-10, we provide further evidence supporting our theoretical claims and showcasing the NSGF’s potential to surpass existing benchmarks for neural Wasserstein gradient flow. In conclusion, it becomes evident that the NSGF opens up new possibilities in the field of optimal transport, establishing a benchmark that will guide future research endeavors in this domain.

REFERENCES

- Andrea Agazzi and Jianfeng Lu. Global optimality of softmax policy gradient with single hidden layer neural networks in the mean-field regime. *arXiv preprint arXiv:2010.11858*, 2020.
- Mokhtar Z Alaya, Maxime Berar, Gilles Gasso, and Alain Rakotomamonjy. Screening sinkhorn algorithm for regularized optimal transport. *Advances in Neural Information Processing Systems*, 32, 2019.
- Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=li7qeBbCRlt>.
- Fabian Altekruiger, Johannes Hertrich, and Gabriele Steidl. Neural wasserstein gradient flows for maximum mean discrepancies with riesz kernels. *arXiv preprint arXiv:2301.11624*, 2023.
- David Alvarez-Melis, Yair Schiff, and Youssef Mroueh. Optimizing functionals on the space of probabilities with input convex neural networks. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL <https://openreview.net/forum?id=dpOYN7o8Jm>.
- Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2005.
- Brandon Amos, Lei Xu, and J Zico Kolter. Input convex neural networks. In *International Conference on Machine Learning*, pp. 146–155. PMLR, 2017.
- Abdul Fatir Ansari, Ming Liang Ang, and Harold Soh. Refining deep generative models via discriminator gradient flow. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Zbc-ue9p_rE.
- Michael Arbel, Anna Korba, Adil Salim, and Arthur Gretton. Maximum mean discrepancy gradient flow. *Advances in Neural Information Processing Systems*, 32, 2019.
- Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. *arXiv preprint arXiv:1701.04862*, 2017.
- Mathieu Blondel, Vivien Seguy, and Antoine Rolet. Smooth and sparse optimal transport. In *International conference on artificial intelligence and statistics*, pp. 880–889. PMLR, 2018.
- Clément Bonet, Nicolas Courty, François Septier, and Lucas Drumetz. Sliced-wasserstein gradient flows. *arXiv preprint arXiv:2110.10972*, 2021.
- Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *SIAM review*, 60(2):223–311, 2018.
- Charlotte Bunne, Laetitia Papaxanthos, Andreas Krause, and Marco Cuturi. Proximal optimal transport modeling of population dynamics. In *International Conference on Artificial Intelligence and Statistics*, pp. 6511–6528. PMLR, 2022.
- John C Butcher. Implicit runge-kutta processes. *Mathematics of computation*, 18(85):50–64, 1964.
- Kristy Choi, Chenlin Meng, Yang Song, and Stefano Ermon. Density ratio estimation via infinitesimal classification. In *International Conference on Artificial Intelligence and Statistics*, pp. 2552–2573. PMLR, 2022.
- Nicolas Courty, Rémi Flamary, and Devis Tuia. Domain adaptation with regularized optimal transport. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I 14*, pp. 274–289. Springer, 2014.
- Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.
- Biwei Dai and Uros Seljak. Sliced iterative normalizing flows. *arXiv preprint arXiv:2007.00674*, 2020.

- Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 447–463, 2018.
- Ayan Das, Stathi Fotiadis, Anil Batra, Farhang Nabiei, FengTing Liao, Sattar Vakili, Da-shan Shiu, and Alberto Bernacchia. Image generation with shortest path diffusion. *arXiv preprint arXiv:2306.00501*, 2023.
- Kamil Deja, Jan Dubiński, Piotr Nowak, Sandro Wenzel, Przemysław Spurek, and Tomasz Trzcinski. End-to-end sinkhorn autoencoder with noise generator. *IEEE Access*, 9:7211–7219, 2020.
- Lauro Langosco di Langosco, Vincent Fortuin, and Heiko Strathmann. Neural variational gradient descent. *arXiv preprint arXiv:2107.10731*, 2021.
- Carles Domingo-Enrich, Samy Jelassi, Arthur Mensch, Grant Rotskoff, and Joan Bruna. A mean-field analysis of two-player zero-sum games. *Advances in neural information processing systems*, 33:20215–20226, 2020.
- Chao Du, Tianbo Li, Tianyu Pang, YAN Shuicheng, and Min Lin. Nonparametric generative modeling with conditional sliced-wasserstein flows. 2023.
- Jiaojiao Fan, Qinsheng Zhang, Amirhossein Taghvaei, and Yongxin Chen. Variational wasserstein gradient flow. In *International Conference on Machine Learning*, pp. 6185–6215. PMLR, 2022.
- Kilian Fatras, Younes Zine, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Learning with minibatch wasserstein: asymptotic and gradient properties. *arXiv preprint arXiv:1910.04091*, 2019.
- Kilian Fatras, Thibault Séjourné, Rémi Flamary, and Nicolas Courty. Unbalanced minibatch optimal transport; applications to domain adaptation. In *International Conference on Machine Learning*, pp. 3186–3197. PMLR, 2021a.
- Kilian Fatras, Younes Zine, Szymon Majewski, Rémi Flamary, Rémi Gribonval, and Nicolas Courty. Minibatch optimal transport distances; analysis and applications. *arXiv preprint arXiv:2101.01792*, 2021b.
- Jean Feydy, Thibault Séjourné, François-Xavier Vialard, Shun-ichi Amari, Alain Trounev, 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.
- Gerald B Folland. *Real analysis: modern techniques and their applications*, volume 40. John Wiley & Sons, 1999.
- Yuan Gao, Yuling Jiao, Yang Wang, Yao Wang, Can Yang, and Shunkang Zhang. Deep generative learning via variational gradient flow. In *International Conference on Machine Learning*, pp. 2093–2101. PMLR, 2019.
- Yuan Gao, Jian Huang, Yuling Jiao, Jin Liu, Xiliang Lu, and Zhijian Yang. Deep generative learning via euler particle transport. In *Mathematical and Scientific Machine Learning*, pp. 336–368. PMLR, 2022a.
- Yuan Gao, Wuchen Li, and Jian-Guo Liu. Master equations for finite state mean field games with nonlinear activations. *arXiv preprint arXiv:2212.05675*, 2022b.
- Aude Genevay, Marco Cuturi, Gabriel Peyré, and Francis Bach. Stochastic optimization for large-scale optimal transport. *Advances in neural information processing systems*, 29, 2016.
- Aude Genevay, Gabriel Peyré, and Marco Cuturi. Learning generative models with sinkhorn divergences. In *International Conference on Artificial Intelligence and Statistics*, pp. 1608–1617. PMLR, 2018.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

- Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. *arXiv preprint arXiv:1810.01367*, 2018.
- Alvin Heng, Abdul Fatir Ansari, and Harold Soh. Deep generative wasserstein gradient flows. 2022.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, 30, 2017.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- Ya-Ping Hsieh, Panayotis Mertikopoulos, and Volkan Cevher. The limits of min-max optimization algorithms: Convergence to spurious non-critical sets. In *International Conference on Machine Learning*, pp. 4337–4348. PMLR, 2021.
- Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the fokker-planck equation. *SIAM journal on mathematical analysis*, 29(1):1–17, 1998.
- Iraj Kalantari. *Induction over the Continuum*, pp. 145–154. Springer Netherlands, Dordrecht, 2007. ISBN 978-1-4020-6127-1. doi: 10.1007/978-1-4020-6127-1_5. URL https://doi.org/10.1007/978-1-4020-6127-1_5.
- Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- Tatsuya Komatsu, Tomoko Matsui, and Junbin Gao. Multi-source domain adaptation with sinkhorn barycenter. In *2021 29th European Signal Processing Conference (EUSIPCO)*, pp. 1371–1375. IEEE, 2021.
- Alexander Korotin, Lingxiao Li, Aude Genevay, Justin M Solomon, Alexander Filippov, and Evgeny Burnaev. Do neural optimal transport solvers work? a continuous wasserstein-2 benchmark. *Advances in Neural Information Processing Systems*, 34:14593–14605, 2021.
- Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. Mmd gan: Towards deeper understanding of moment matching network. *Advances in neural information processing systems*, 30, 2017.
- Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=PqvMRDCJT9t>.
- Chang Liu, Jingwei Zhuo, Pengyu Cheng, Ruiyi Zhang, and Jun Zhu. Understanding and accelerating particle-based variational inference. In *International Conference on Machine Learning*, pp. 4082–4092. PMLR, 2019.
- Qiang Liu. Stein variational gradient descent as gradient flow. *Advances in neural information processing systems*, 30, 2017.
- 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*, 2023. URL <https://openreview.net/forum?id=XVjTT1nw5z>.
- Antoine Liutkus, Umut Simsekli, Szymon Majewski, Alain Durmus, and Fabian-Robert Stöter. Sliced-wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. In *International Conference on Machine Learning*, pp. 4104–4113. PMLR, 2019.

- Giulia Luise, Saverio Salzo, Massimiliano Pontil, and Carlo Ciliberto. Sinkhorn barycenters with free support via frank-wolfe algorithm. *Advances in neural information processing systems*, 32, 2019.
- John Martin, Michal Lyskawinski, Xiaohu Li, and Brendan Englot. Stochastically dominant distributional reinforcement learning. In *International conference on machine learning*, pp. 6745–6754. PMLR, 2020.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.
- Petr Mokrov, Alexander Korotin, Lingxiao Li, Aude Genevay, Justin M Solomon, and Evgeny Burnaev. Large-scale wasserstein gradient flows. *Advances in Neural Information Processing Systems*, 34:15243–15256, 2021.
- Youssef Mroueh and Mattia Rigotti. Unbalanced sobolev descent. *Advances in Neural Information Processing Systems*, 33:17034–17043, 2020.
- Youssef Mroueh, Tom Sercu, and Anant Raj. Sobolev descent. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2976–2985. PMLR, 2019.
- Gautam Pai, Jing Ren, Simone Melzi, Peter Wonka, and Maks Ovsjanikov. Fast sinkhorn filters: Using matrix scaling for non-rigid shape correspondence with functional maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 384–393, 2021.
- Giorgio Patrini, Rianne Van den Berg, Patrick Forre, Marcello Carioni, Samarth Bhargav, Max Welling, Tim Genewein, and Frank Nielsen. Sinkhorn autoencoders. In *Uncertainty in Artificial Intelligence*, pp. 733–743. PMLR, 2020.
- Gabriel Peyré, Marco Cuturi, et al. Computational optimal transport. *Center for Research in Economics and Statistics Working Papers*, (2017-86), 2017.
- Eckhard Platen and Nicola Bruti-Liberati. *Numerical solution of stochastic differential equations with jumps in finance*, volume 64. Springer Science & Business Media, 2010.
- Aram-Alexandre Pooladian, Heli Ben-Hamu, Carles Domingo-Enrich, Brandon Amos, Yaron Lipman, and Ricky Chen. Multisample flow matching: Straightening flows with minibatch couplings. *arXiv preprint arXiv:2304.14772*, 2023.
- Benjamin Rhodes, Kai Xu, and Michael U Gutmann. Telescoping density-ratio estimation. *Advances in neural information processing systems*, 33:4905–4916, 2020.
- Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016.
- Filippo Santambrogio. Optimal transport for applied mathematicians. *Birkhäuser, NY*, 55(58-63):94, 2015.
- Filippo Santambrogio. {Euclidean, metric, and Wasserstein} gradient flows: an overview. *Bulletin of Mathematical Sciences*, 7:87–154, 2017.
- Neta Shaul, Ricky TQ Chen, Maximilian Nickel, Matthew Le, and Yaron Lipman. On kinetic optimal probability paths for generative models. In *International Conference on Machine Learning*, pp. 30883–30907. PMLR, 2023.
- Zebang Shen, Zhenfu Wang, Alejandro Ribeiro, and Hamed Hassani. Sinkhorn barycenter via functional gradient descent. *Advances in Neural Information Processing Systems*, 33:986–996, 2020.
- David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016.

- Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.
- Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021. URL <https://openreview.net/forum?id=PXTIG12RRHS>.
- Endre Süli and David F Mayers. *An introduction to numerical analysis*. Cambridge university press, 2003.
- Alexander Tong, Nikolay Malkin, Guillaume Huguët, Yanlei Zhang, Jarrid Rector-Brooks, Kilian FATRAS, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. In *ICML Workshop on New Frontiers in Learning, Control, and Dynamical Systems*, 2023. URL <https://openreview.net/forum?id=HgDwiZrpVq>.
- Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.
- Wei Wang, Yuan Sun, and Saman Halgamuge. Improving mmd-gan training with repulsive loss function. *arXiv preprint arXiv:1812.09916*, 2018.
- Benjamin J Zhang and Markos A Katsoulakis. A mean-field games laboratory for generative modeling. *arXiv preprint arXiv:2304.13534*, 2023.
- Chao Zhang, Zhijian Li, Hui Qian, and Xin Du. Dpvi: A dynamic-weight particle-based variational inference framework. *arXiv preprint arXiv:2112.00945*, 2021a.
- Ruiyi Zhang, Changyou Chen, Chunyuan Li, and Lawrence Carin. Policy optimization as wasserstein gradient flows. In *International Conference on machine learning*, pp. 5737–5746. PMLR, 2018.
- Yufeng Zhang, Siyu Chen, Zhuoran Yang, Michael Jordan, and Zhaoran Wang. Wasserstein flow meets replicator dynamics: A mean-field analysis of representation learning in actor-critic. *Advances in Neural Information Processing Systems*, 34:15993–16006, 2021b.

A COMPUTATION OF \mathcal{W}_ε -POTENTIALS

The \mathcal{W}_ε -potentials is the cornerstone to conduct NSGF. Hence, a key component of our method is to efficiently compute this quantity. Genevay et al. (2016) provided an efficient method when both μ and ν are discrete measures so that we can calculate \mathcal{W}_ε -potential in terms of samples. In particular, when μ is discrete, f can be simply represented by a finite-dimensional vector since only its values on $\text{supp}(\mu)$ matter.

Remark 2. *To more clearly explain the relationship between the calculation of \mathcal{W}_ε -potentials and the composition of our algorithm, we provide the following explanation: In practice, we actually calculate the \mathcal{W}_ε -potentials for the empirical distribution of discrete minibatches and construct Sinkhorn WGF based on this. Therefore, in fact, the μ and ν in the subsequent text refer to $(\tilde{X}_i^t)_{i=1}^n$ and $(\tilde{Y}_i^t)_{i=1}^n$ in the Algorithm 1.*

We first introduce another property of the entropy-regularized optimal transport problem.

Lemma 2. *Define the Sinkhorn mapping: $\mathcal{A} : \mathcal{C}(\mathcal{X}) \times \mathcal{M}_1^+(\mathcal{X}) \rightarrow \mathcal{C}(\mathcal{X})$*

$$\mathcal{A}(f, \mu)(\mathbf{y}) = -\varepsilon \log \int_{\mathcal{X}} \exp((f(\mathbf{x}) - c(\mathbf{x}, \mathbf{y}))/\varepsilon) d\mu(\mathbf{x}). \quad (17)$$

The pair $(f_{\mu, \nu}, g_{\mu, \nu})$ are the \mathcal{W}_ε -potentials of the entropy-regularized optimal transport problem 2 if they satisfy:

$$f_{\mu, \nu} = \mathcal{A}(g_{\mu, \nu}, \nu), \mu - a.e. \quad \text{and} \quad g_{\mu, \nu} = \mathcal{A}(f_{\mu, \nu}, \mu), \nu - a.e., \quad (18)$$

or equivalently

$$\int_{\mathcal{X}} h(\mathbf{x}, \mathbf{y}) d\nu(\mathbf{y}) = 1, \mu - a.e., \quad \int_{\mathcal{X}} h(\mathbf{x}, \mathbf{y}) d\mu(\mathbf{x}) = 1, \nu - a.e., \quad (19)$$

where $h(\mathbf{x}, \mathbf{y}) := \exp \frac{1}{\gamma} (f(\mathbf{x}) + g(\mathbf{x}) - c(\mathbf{x}, \mathbf{y}))$

To be more precise, by plugging in the optimality condition on $g_{\mu, \nu}$ in 1, the dual problem 2 becomes:

$$\text{OT}_\varepsilon(\mu, \nu) = \max_{f \in \mathcal{C}} \langle f, \mu \rangle + \langle \mathcal{A}(f, \mu), \nu \rangle \quad (20)$$

Viewing the discrete measure μ as a weight vector \mathbf{w}_μ on $\text{supp}(\mu)$, we have:

$$\text{OT}_\varepsilon(\mu, \nu) = \max_{\mathbf{f} \in \mathbb{R}^d} \left\{ F(\mathbf{f}) := \mathbf{f}^\top \mathbf{w}_\mu + \mathbb{E}_{y \sim \nu} [\mathcal{A}(\mathbf{f}, \mu)(y)] \right\}, \quad (21)$$

that is, we get a standard concave stochastic optimization problem, where the randomness of the problem comes from ν (Genevay et al., 2016). Hence, the problem can be solved using stochastic gradient descent (SGD). In our methods, we can treat the computation of \mathcal{W}_ε -potentials as a Blackbox. In practice, we use the efficient implementation of the Sinkhorn algorithm with GPU acceleration from the GeomLoss package (Feydy et al., 2019).

B THEORY OF SINKHORN WASSERSTEIN GRADIENT FLOW

Definition 3. *(First variation of Functionals over Probability). Given a functional $\mathcal{F} : \mathcal{P}(\mathcal{X}) \rightarrow \mathbb{R}^+$, we shell perturb measure μ with a perturbation χ so that $\mu + t\chi$ belongs to $\mathcal{P}(\mathcal{X})$ for small t ($\int d\chi = 0$). We treat $\mathcal{F}(\mu)$, as a functional over probability in its second argument and compute its first variation as follows:*

$$\left. \frac{d}{dt} \mathcal{F}(\mu + t\chi) \right|_{t=0} = \lim_{t \rightarrow 0} \frac{\mathcal{F}(\mu + t\chi) - \mathcal{F}(\mu)}{t} := \int \frac{\delta \mathcal{F}}{\delta \mu}(\mu) d\chi. \quad (22)$$

B.1 PROOF OF THEOREM 1

Proof. According to definition 3, given $\mathcal{F}_\varepsilon(\cdot) = \mathcal{S}_\varepsilon(\cdot, \mu^*)$ and t in a neighborhood of 0, we define $\mu_t = \mu + t\delta\mu$

$$\begin{aligned} \lim_{t \rightarrow 0} \frac{1}{t} (\mathcal{F}_\varepsilon(\mu_t) - \mathcal{F}_\varepsilon(\mu)) &= \lim_{t \rightarrow 0} \frac{1}{t} (\underbrace{\mathcal{W}_\varepsilon(\mu_t, \mu^*) - \mathcal{W}_\varepsilon(\mu, \mu^*)}_{\Delta_t^{\text{first part}}}) \\ &\quad - \lim_{t \rightarrow 0} \frac{1}{2t} (\underbrace{\mathcal{W}_\varepsilon(\mu_t, \mu_t) - \mathcal{W}_\varepsilon(\mu, \mu)}_{\Delta_t^{\text{second part}}}) \end{aligned}$$

We first analysis $\Delta_t^{\text{first part}} := \lim_{t \rightarrow 0} \frac{1}{t} (\mathcal{W}_\varepsilon(\mu_t, \mu^*) - \mathcal{W}_\varepsilon(\mu, \mu^*))$. First, let us remark that (f, g) is the a suboptimal pair of dual potentials $\mathcal{W}_{\varepsilon, \mu^*}(\mu)$ for short. Recall 3,

$$\mathcal{W}_\varepsilon \geq \langle \mu_t, f \rangle + \langle \mu^*, g \rangle - \varepsilon \left\langle \mu_t \otimes \mu^*, \exp \left(\frac{1}{\varepsilon} (f \oplus g - C) \right) - 1 \right\rangle,$$

and thus, since

$$\mathcal{W}_\varepsilon \geq \langle \mu, f \rangle + \langle \mu^*, g \rangle - \varepsilon \left\langle \mu \otimes \mu^*, \exp \left(\frac{1}{\varepsilon} (f \oplus g - C) \right) - 1 \right\rangle,$$

one has

$$\begin{aligned} \Delta_t^{\text{first part}} &\geq \langle \delta\mu, f \rangle - \varepsilon \left\langle \delta\mu \otimes \mu^*, \exp \left(\frac{1}{\varepsilon} (f \oplus g - C) \right) \right\rangle + o(1) \\ &\geq \langle \delta\mu, f - \varepsilon \rangle + o(1) \end{aligned}$$

Conversely, let us denote by (f_t, g_t) the optimal pair of potentials for $\mathcal{W}_\varepsilon(\mu_t, \mu^*)$ satisfying $g_t(x_o) = 0$ for some arbitrary anchor point $x_o \in \mathcal{X}$. As (f_t, g_t) are suboptimal potentials for $\mathcal{W}_\varepsilon(\mu, \mu^*)$ we get that

$$\mathcal{W}_\varepsilon \geq \langle \mu, f_t \rangle + \langle \mu^*, g_t \rangle - \varepsilon \left\langle \mu \otimes \mu^*, \exp \left(\frac{1}{\varepsilon} (f \oplus g - C) \right) - 1 \right\rangle,$$

and thus, since

$$\mathcal{W}_\varepsilon \geq \langle \mu_t, f_t \rangle + \langle \mu^*, g_t \rangle - \varepsilon \left\langle \mu_t \otimes \mu^*, \exp \left(\frac{1}{\varepsilon} (f_t \oplus g - C) \right) - 1 \right\rangle,$$

one has

$$\begin{aligned} \Delta_t^{\text{first part}} &\geq \langle \delta\mu, f_t \rangle - \varepsilon \left\langle \delta\mu \otimes \mu_t^*, \exp \left(\frac{1}{\varepsilon} (f_t \oplus g - C) \right) \right\rangle + o(1) \\ &\geq \langle \delta\mu, f_t - \varepsilon \rangle + o(1) \end{aligned}$$

Now, let us remark that as t goes to 0, $\mu + t\delta\mu \rightarrow \mu$. f_t and g_t converge uniformly towards f and g according to Proposition 13 Feydy et al. (2019). we get

$$\Delta_t^{\text{first part}} = \langle \delta\mu, f \rangle$$

Similar to analysis $\Delta_t^{\text{first part}} := \lim_{t \rightarrow 0} \frac{1}{t} (\mathcal{W}_\varepsilon(\mu_t, \mu^*) - \mathcal{W}_\varepsilon(\mu, \mu^*))$ we define $\Delta_t^{\text{second part}} := \lim_{t \rightarrow 0} \frac{1}{2t} (\mathcal{W}_\varepsilon(\mu_t, \mu_t) - \mathcal{W}_\varepsilon(\mu, \mu))$, we have:

$$\Delta_t^{\text{second part}} = \langle \delta\mu, f' \rangle$$

to be more clearly, we denote $f = f_{\mu, \mu^*}$ and $f' = f_{\mu, \mu}$ thus,

$$\lim_{t \rightarrow 0} \frac{1}{t} (\mathcal{F}_\varepsilon(\mu_t) - \mathcal{F}_\varepsilon(\mu)) = \langle \delta\mu, f_{\mu, \mu^*} - f_{\mu, \mu} \rangle.$$

So the first variation of \mathcal{F}_ε is:

$$\frac{\delta \mathcal{F}_\varepsilon}{\delta \mu} = f_{\mu, \mu^*} - f_{\mu, \mu}.$$

□

B.2 PROOF OF PROPOSITION 2

Following the lines of our proof in Theorem 1, we give the following proof.

Lemma 3. (*Fréchet derivative of entropy-regularized Wasserstein distance*) Let $\varepsilon > 0$. We shall fix in the following a measure μ^* and let $(f_{\mu, \mu^*}, g_{\mu, \mu^*})$ be the \mathcal{W}_ε potentials of $\mathcal{W}_\varepsilon(\mu, \mu^*)$ according to lemma 1. Consider the infinitesimal transport $T(x) = x + \lambda\phi$. We have the Fréchet derivative under this particular perturbation:

$$\begin{aligned} \frac{d}{d\lambda} \mathcal{W}_\varepsilon(T_\# \mu, \mu^*)|_{\lambda=0} &= \lim_{\lambda \rightarrow 0} \frac{\mathcal{W}_\varepsilon(T_\# \mu, \mu^*) - \mathcal{W}_\varepsilon(\mu, \mu^*)}{\lambda} \\ &= \int_{\mathcal{X}} \nabla f_{\mu, \mu^*}(\mathbf{x}) \phi(\mathbf{x}) d\mu(\mathbf{x}). \end{aligned} \quad (23)$$

Proof. let $f = f_{\mu, \mu^*}$ and $g = g_{\mu, \mu^*}$ be the \mathcal{W}_ε -potentials 1 to $\mathcal{W}_\varepsilon(\mu, \mu^*)$ for short. By 3 and the optimality of (f, g) , we have follows:

$$\mathcal{W}_\varepsilon(\mu, \mu^*) = \langle f, \mu \rangle + \langle g, \mu^* \rangle.$$

However, (f, g) are not necessarily the optimal dual variables for $\mathcal{W}_\varepsilon(T_\# \mu, \mu^*)$, recall the lemma 2:

$$\mathcal{W}_\varepsilon(T_\# \mu, \mu^*) \geq \langle f, T_\# \mu \rangle + \langle g, \mu^* \rangle - \varepsilon \langle h - 1, T_\# \mu \otimes \mu^* \rangle,$$

where $\int_{\mathcal{X}} h(\mathbf{x}, \mathbf{y}) d\mu^*(\mathbf{y}) = 1$ and hence $\langle h - 1, T_\# \mu \otimes \mu^* \rangle = 0$. Thus:

$$\mathcal{W}_\varepsilon(T_\# \mu, \mu^*) - \mathcal{W}_\varepsilon(\mu, \mu^*) \geq \langle f, T_\# \mu - \mu \rangle.$$

Use the change-of-variables formula of the push-forward measure to obtain:

$$\frac{1}{\lambda} \langle f, T_\# \mu - \mu \rangle = \frac{1}{\lambda} \int_{\mathcal{X}} ((f \circ T)(\mathbf{x}) - f(\mathbf{x})) d\mu(\mathbf{x}) = \int_{\mathcal{X}} \nabla f(x + \lambda' \phi(x)) \phi(x) d\mu(x),$$

where $\lambda' \in [0, \lambda]$ is from the mean value theorem. Here we assume ∇f is Lipschitz continuous follow Proposition 12 in Feydy et al. (2019) and Lemma A.4 from Shen et al. (2020). We have:

$$\lim_{\lambda \rightarrow 0} \frac{1}{\lambda} \langle f, T_\# \mu - \mu \rangle = \int_{\mathcal{X}} \nabla f(\mathbf{x}) \phi(\mathbf{x}) d\mu(\mathbf{x}).$$

Hence:

$$\lim_{\lambda \rightarrow 0} \frac{1}{\lambda} (\mathcal{W}_\varepsilon(T_\# \mu, \mu^*) - \mathcal{W}_\varepsilon(\mu, \mu^*)) \geq \int_{\mathcal{X}} \nabla f(\mathbf{x}) \phi(\mathbf{x}) d\mu(\mathbf{x}).$$

Similarly, let f' and g' be the \mathcal{W}_ε potentials to $\mathcal{W}_\varepsilon(T_\# \mu, \mu^*)$, we have:

$$\mathcal{W}_\varepsilon(\mu, \mu^*) \geq \langle f', \mu \rangle + \langle g, \mu^* \rangle - \varepsilon \langle h - 1, \mu \otimes \mu^* \rangle,$$

where $\int_{\mathcal{X}} h(\mathbf{x}, \mathbf{y}) d\mu^*(\mathbf{y}) = 1$ and hence $\langle h - 1, \mu \otimes \mu^* \rangle = 0$. Thus:

$$\mathcal{W}_\varepsilon(T_\# \mu, \mu^*) - \mathcal{W}_\varepsilon(\mu, \mu^*) \leq \langle f', T_\# \mu - \mu \rangle.$$

Same as above, use the change-of-variables formula and the mean value theorem:

$$\frac{1}{\lambda} \langle f', T_\# \mu - \mu \rangle = \int_{\mathcal{X}} \nabla f'(x + \lambda' \phi(x)) \phi(x) d\mu(x),$$

Thus:

$$\lim_{\lambda \rightarrow 0} \frac{1}{\lambda} (\mathcal{W}_\varepsilon(T_\# \mu, \mu^*) - \mathcal{W}_\varepsilon(\mu, \mu^*)) \leq \int_{\mathcal{X}} \lim_{\lambda \rightarrow 0} \nabla f'(x + \lambda' \phi(x)) \phi(x) d\mu(x).$$

Assume that $\nabla f'$ is Lipschitz continuous and $f' \rightarrow f$ as $\lambda \rightarrow 0$. Consequently we have $\lim_{\lambda \rightarrow 0} \nabla f'(x + \lambda' \phi(x)) = \nabla f(x)$ and hence:

$$\lim_{\lambda \rightarrow 0} \frac{1}{\lambda} (\mathcal{W}_\varepsilon(T_\# \mu, \mu^*) - \mathcal{W}_\varepsilon(\mu, \mu^*)) = \int_{\mathcal{X}} \nabla f(\mathbf{x}) \phi(\mathbf{x}) d\mu(\mathbf{x}).$$

According to lemma 3, we have:

$$\begin{aligned} \left. \frac{d}{d\lambda} \mathcal{F}_\varepsilon(T_\# \mu) \right|_{\lambda=0} &= \int_{\mathcal{X}} \nabla f_{\mu, \mu^*}(\mathbf{x}) \phi(\mathbf{x}) d\mu(\mathbf{x}) - \frac{1}{2} \cdot \int_{\mathcal{X}} 2 \nabla f_{\mu, \mu}(\mathbf{x}) \phi(\mathbf{x}) d\mu(\mathbf{x}) \\ &= \int_{\mathcal{X}} \nabla f_{\mu, \mu^*}(\mathbf{x}) \phi(\mathbf{x}) d\mu - \int_{\mathcal{X}} \nabla f_{\mu, \mu}(\mathbf{x}) \phi(\mathbf{x}) d\mu. \end{aligned}$$

□

B.3 PROOF OF THEOREM 2

Proof. First, we define $\Psi(\mu) = \int h d\mu$ where $h : \mathbb{R}^d \rightarrow \mathbb{R}$ is an arbitrary bounded and continuous function and $\frac{\delta \Psi(\mu)}{\delta \mu}(x)$ denotes the first variation of functional Ψ at μ satisfying:

$$\int \frac{\delta \Psi(\mu)}{\delta \mu}(x) \xi(x) dx = \lim_{\epsilon \rightarrow 0} \frac{\Psi(\mu + \epsilon \xi) - \Psi(\mu)}{\epsilon}$$

for all signed measure $\int \xi(x) dx = 0$. We also have the following:

$$\frac{\delta \Psi(\mu)}{\delta \mu}(\cdot) = \frac{\delta \int h d\mu}{\delta \mu}(\cdot) = h(\cdot)$$

Assume μ_t is a flow satisfies the following:

$$\partial_t \Psi[\mu_t] = (\mathcal{L}\Psi)[\mu_t],$$

where,

$$\mathcal{L}\Psi[\mu_t] = - \int \langle \nabla \frac{\delta \mathcal{F}_\varepsilon(\mu_t)}{\delta \mu}(x), \nabla_x \frac{\delta \Psi(\mu_t)}{\delta \mu}(x) \rangle \mu_t(x) dx \quad (24)$$

Notably, μ_t is a solution of equation 2.

Next, let $\tilde{\mu}_t^M$ be the distribution produced by the equation 14 at time t . Under mild assumption of $\tilde{\mu}_0^M \rightharpoonup \mu_0$, we want to show that the mean-field limit of $\tilde{\mu}_t^M$ as $M \rightarrow \infty$ is μ_t by showing that $\lim_{M \rightarrow \infty} \Psi(\mu_t^M) = \Psi(\mu_t)$ (Folland, 1999).

For the measure valued flow $\tilde{\mu}_t^M$ equation 14, the infinitesimal generator of Ψ w.r.t. $\tilde{\mu}_t^M$ is defined as follows:

$$(\mathcal{L}\Psi)[\tilde{\mu}_t^M] := \lim_{\epsilon \rightarrow 0^+} \frac{\Psi(\tilde{\mu}_{t+\epsilon}^M) - \Psi(\tilde{\mu}_t^M)}{\epsilon},$$

According to the definition of first variation, it can be calculated that

$$\begin{aligned} (\mathcal{L}\Psi)[\tilde{\mu}_t^M] &= \lim_{\epsilon \rightarrow 0^+} \frac{\Psi[\sum_{i=1}^M \frac{1}{M} \delta_{x_{t+\epsilon}^i}] - \Psi(\sum_{i=1}^M \frac{1}{M} \delta_{x_t^i})}{\epsilon} \\ &= \int \frac{\delta \Psi(\tilde{\mu}_t^M)}{\delta \mu}(x) \sum_{i=1}^M \frac{1}{M} \partial_t \rho(x_t^i) dx \end{aligned}$$

Then we adopt the *Induction over the Continuum* to prove $\lim_{n \rightarrow \infty} \Psi(\tilde{\mu}_t^M) = \Psi(\mu_t)$ for all $t > 0$. Here $t \in \mathbb{R}^+$ satisfy the requirement of well ordering and the existence of a greatest lower bound for non-empty subsets, so *Induction over the Continuum* is reasonable (Kalantari, 2007).

1. As for $t = 0$, our assumption of $\tilde{\mu}_0^M \rightharpoonup \mu_0$ suffice.
2. For the case of $t = t^*$, we first hypothesis that for $t < t^*$, $\tilde{\mu}_t^M \rightharpoonup \mu_t$ as $M \rightarrow \infty$. Then for $t < t^*$ we have:

$$\begin{aligned} \lim_{M \rightarrow \infty} (\mathcal{L}\Psi)[\tilde{\mu}_t^M] &= \lim_{M \rightarrow \infty} \int \frac{\delta \Psi(\tilde{\mu}_t^M)}{\delta \mu}(x) \sum_{i=1}^M \frac{1}{M} \partial_t \rho(x_t^i) dx \\ &= - \lim_{M \rightarrow \infty} \int \langle \nabla \frac{\delta \mathcal{F}_\varepsilon(\mu_t)}{\delta \mu}(x), \nabla_x \frac{\delta \Psi(\tilde{\mu}_t^M)}{\delta \mu}(x) \rangle \tilde{\mu}_t^M(x) dx \\ &= - \int \langle \nabla \frac{\delta \mathcal{F}_\varepsilon(\mu_t)}{\delta \mu}(x), \nabla_x \frac{\delta \Psi(\mu_t)}{\delta \mu}(x) \rangle \mu_t(x) dx. \end{aligned}$$

Because $\lim_{M \rightarrow \infty} \Psi(\tilde{\mu}_0^M) = \Psi(\mu_0)$ at $t = 0$ and $\lim_{M \rightarrow \infty} (\partial_t \Psi)[\tilde{\mu}_t^M] = (\partial_t \Psi)[\mu_t]$ for all $t < t^*$, we have $\lim_{M \rightarrow \infty} \Psi(\tilde{\mu}_{t^*}^M) = \Psi(\mu_{t^*})$.

Combining (1) and (2), we can reach to the conclusion that $\lim_{M \rightarrow \infty} \Psi(\mu_t^M) = \Psi(\mu_t)$ for all t . which indicates that $\tilde{\mu}_t^M \rightharpoonup \mu_t$ if $\tilde{\mu}_0^M \rightharpoonup \mu_0$. Since μ_t solves the partial differential equation 10, we conclude that the path of equation 14 starting from $\tilde{\mu}_0^M$ weakly converges to a solution of the partial differential equation 10 starting from μ_0 as $M \rightarrow \infty$.

□

B.4 DESCENDING PROPERTY

Proposition 3. *Consider the Sinkhorn gradient flow 10, the differentiation of $\mathcal{F}_\varepsilon(\mu_t)$ with respect to the time t satisfies:*

$$\frac{d\mathcal{F}_\varepsilon(\mu_t)}{dt} = - \int \left\| \nabla \left(\frac{\delta \mathcal{F}_\varepsilon(\mu_t)}{\delta \mu_t} \right) \right\|^2 d\mu_t \leq 0 \quad (25)$$

Proof. By substituting $\Psi(\cdot) = \mathcal{F}_\varepsilon(\cdot)$ in equation 24, we directly reach to the above equality. \square

C MINIBATCH OPTIMAL TRANSPORT

For large datasets, the computation and storage of the optimal transport plan can be challenging due to OT’s cubic time and quadratic memory complexities relative to the number of samples (Cuturi, 2013; Genevay et al., 2016; Peyré et al., 2017). The minibatch approximation offers a viable solution for enhancing calculation efficiency. Theoretical analysis of using the minibatch approximation for transportation plans is provided by Fatras et al. (2019; 2021b). Although minibatch OT introduces some errors compared to the exact OT solution, its efficiency in computing approximate OT is clear, and it has seen successful applications in domains like domain adaptation (Damodaran et al., 2018; Fatras et al., 2021a) and generative modeling (Genevay et al., 2018).

More recently, (Pooladian et al., 2023; Tong et al., 2023) introduced OT-CFM and empirically demonstrated that using minibatch approximation of optimal transport in flow matching methods (Liu et al., 2023; Lipman et al., 2023) can straighten the flow’s trajectory and yield more consistent samples. OT-CFM specifically focuses on minibatch initial and target samples, continuing to use random linear interpolation paths. In contrast, NSGF leverages minibatch \mathcal{W}_ε -potentials to construct Sinkhorn gradient flows in minibatches. Our method also involves performing velocity field matching on the flow’s discretized form, marking a separate and innovative direction in the field.

D INFERENCE ALGORITHM

Algorithm 2: Inference via velocity field

Input : number of time steps T , inference step size η , time-varying velocity field \mathbf{v}^θ , prior samples

$\tilde{X}_i^0 \sim \tilde{\mu}_0$
for $t = 0, 1, \dots, T$ **do**
 $\tilde{X}_i^{t+1} = \tilde{X}_i^t + \eta \mathbf{v}^\theta(\tilde{X}_i^t, t), i = 1, 2, \dots, n.$

Output: \tilde{X}_i^T as the results.

E EXPERIMENTS

E.1 2D SIMULATE DATA

For the 2D experiments, we closely follow Tong et al. (2023) and the released code at <https://github.com/atong01/conditional-flow-matching> (code released under MIT license), and use the same synthetic datasets and the 2-Wasserstein distance between the test set and samples simulated using NSGF as the evaluation metric. We use 1024 samples in the test set since we find the We use a simple MLP with 3 hidden layers and 256 hidden units to parameterize the velocity matching networks. We use batch size 256 and 10/100 steps with a uniform schedule at sampling time. For both Neural gradient-flow-based models and Neural ODE-based Models, we train for 20000 steps in total. Note that FM cannot be used for the 8gaussians-moons task since it requires a Gaussian source, but we still conducted experiments with the algorithm and found competitive experimental results. We believe that this is because FM is essentially very close to 1-RF in its algorithmic design, and that the Gaussian source condition can be meaningfully relaxed in practice, as confirmed

Algorithm	MNIST FID(\downarrow)
NSGF(ours)	3.24
VGrow(Outerloop)(Gao et al., 2019)	3.32
SWGf(Liutkus et al., 2019)	225.1
SIG(Dai & Seljak, 2020)	4.5

Table 3: Comparison of Neural Wasserstein gradient flow methods over MNIST

in Tong et al. (2023). The experiments are run using one 3090 GPU and take approximately less than 60 minutes (for both training and testing).

For the neural gradient-flow-based models, we solely implemented the EPT without the outer loop, as the outer loop can be likened to a GAN-like distillation approach (Goodfellow et al., 2014). Notably, the original EPT (Gao et al., 2022a) recommends iterating for 20,000 rounds with an exceedingly small step size; however, to ensure a fair comparison, we employed the same number of steps as the other methods while adapting the step size accordingly. It’s worth mentioning that for the JKO-Flow, we used the recommended parameter setting of 10 steps, as suggested in Fan et al. (2022), but we also provide results for 100 steps for comparative purposes. All the results for Neural Gradient flow-based models were trained and sampled following the standard procedures outlined in their respective papers.

E.2 IMAGE BENCHMARK DATA

For the MNIST/CIFAR-10 experiments, we summarize the setup here, where the exact parameter choices can be seen in the source code. For the calculation of \mathcal{W}_ϵ -potentials, we use the Geom-Loss package (Feydy et al., 2019) with blur = 0.05, 0.1 or 1, scaling = 0.85 or 0.95 depends on learning rate of Sinkhorn gradient flow. We also find that training the NSGF model in MNIST can use blur = 1, scaling ≤ 0.85 but still get good performance. To minimize the computational cost, we choose blur = 1, scaling = 0.75. We also find using an incremental lr scheme will improve training performance. More detailed experiments we will leave for future work. We used the AdamW optimizer with weight decay = 10^{-4} . We used the UNet architecture from <https://github.com/zoubohao/DenoisingDiffusionProbabilityModel-ddpm>- we find that this UNet architecture is slightly different from the architecture used in Ho et al. (2020). The training parameters are listed as flows: batch size = 128, learning rate = 10^{-5} , For sampling, we use 50-step Euler integration for MNIST and 100-step Euler integration for CIFAR10. For the MNIST/CIFAR-10 experiments, a considerable amount of storage space is required when establishing the trajectory pool during the first phase of the algorithm. For instance, with CIFAR-10, setting the batch size to 128 and saving all minibatch Sinkhorn gradient flow trajectories while traversing the entire dataset requires about 115GB of storage space. In situations where storage space is limited, we suggest dynamically adding and removing trajectories in the trajectory pool to meet the training requirements. Identifying a more effective trade-off between training time and storage space utilization is a direction for future improvement.

E.3 SUPPLEMENTARY EXPERIMENTAL RESULTS

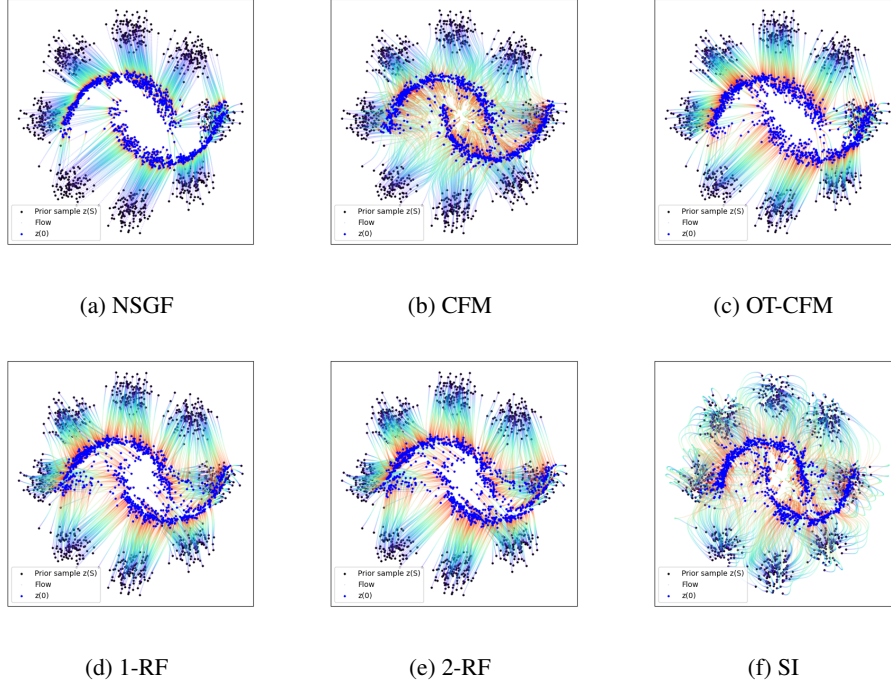


Figure 5: Visualization results for 2D generated paths. We show different methods that drive the particle from the prior distribution (black) to the target distribution (blue). The color change of the flow shows the different number of steps (from blue to red means from 0 to T). We can see NSGF using fewer steps than OT-CFM



Figure 6: Uncurated samples of our methods on MNIST (Without data augmentation)



Figure 7: Samples of our methods on building trajectory CIFAR10



Figure 8: Uncurated samples of our NSGF methods on CIFAR10

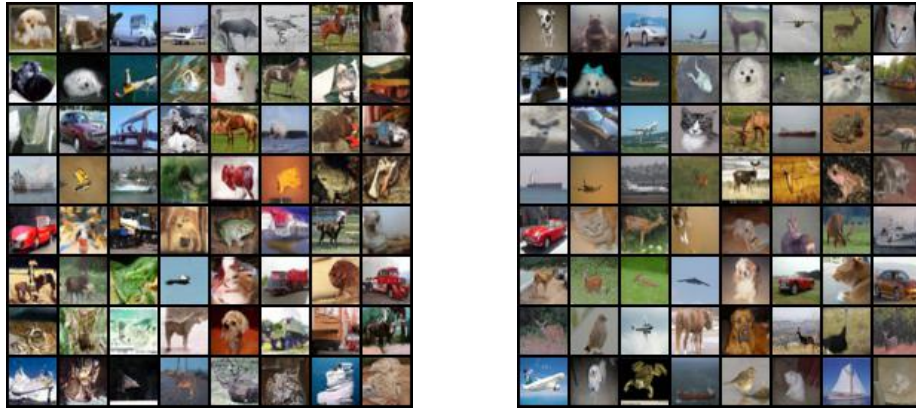


Figure 9: Uncurated samples on CIFAR10 and L_2 -nearest neighbors from the training set (left: Samples, right: real)