Convergence of Flow-Based Generative Models via Proximal Gradient Descent in Wasserstein Space

Xiuyuan Cheng, Jianfeng Lu, Yixin Tan, and Yao Xie, *Member, IEEE*

Abstract— **Flow-based generative models enjoy certain advantages in computing the data generation and the likelihood, and have recently shown competitive empirical performance. Compared to the accumulating theoretical studies on related score-based diffusion models, analysis of flow-based models, which are deterministic in both forward (data-to-noise) and reverse (noise-to-data) directions, remain sparse. In this paper, we provide a theoretical guarantee of generating data distribution by a progressive flow model, the so-called JKO flow model, which implements the Jordan-Kinderleherer-Otto (JKO) scheme in a normalizing flow network. Leveraging the exponential convergence of the proximal gradient descent (GD) in Wasserstein space, we prove the Kullback-Leibler (KL) guarantee of data generation by a JKO flow model to be $O(\varepsilon^2)$ when using $N \lesssim \log(1/\varepsilon)$ many JKO steps (N Residual Blocks in the flow) where ε is the error in the per-step first-order condition. The assumption on data density is merely a finite second moment, and the theory extends to data distributions without density and when there are inversion errors in the reverse process where we obtain KL-$\mathcal{W}_2$ mixed error guarantees. The non-asymptotic convergence rate of the JKO-type $\mathcal{W}_2$-proximal GD is proved for a general class of convex objective functionals that includes the KL divergence as a special case, which can be of independent interest. The analysis framework can extend to other first-order Wasserstein optimization schemes applied to flow-based generative models.**

Index Terms— **Proximal gradient descent, Kullback-Leibler divergence, Wasserstein space, generative models.**

I. INTRODUCTION

GENERATIVE models, from generative adversarial networks (GAN) [28], [30], [37] and variational autoencoder (VAE) [41], [42] to normalizing flow [44], have achieved many successes in applications and have become a central topic in deep learning. More recently, diffusion models [33], [66], [67] and closely related flow-based models [2], [3], [26], [50], [74] have drawn much research attention, given their state-of-the-art performance in image generations. Compared to score-based diffusion models, which are designed for sampling, flow models have certain advantages due to their direct capability in estimating likelihood, a basis for statistical inference. However, despite the empirical successes, the theoretical understanding and guarantees for flow-based generative models remain limited.

In this paper, we provide a theoretical guarantee of generating data distribution by a "progressive" flow model, mainly following the JKO flow model in [74] but similar models have been proposed in, e.g., [4], [54], and [70]. We prove the exponential convergence rate of such flow models in both (data-to-noise and noise-to-data) directions. Below, we give an overview of the main results. We provide a brief introduction of the most related types of flow-based models, particularly the progressive one, in Section I-A. A more complete literature survey can be found in Section I-B.

An abundance of theoretical works has provided the generation guarantee of score-based diffusion models [11], [14], [17], [20], [47], [48], [59]. In comparison, the theoretical study of flow-based generative models is much less developed. Most recent works on the topic focused on the generation guarantee of the Ordinary Differential Equation (ODE) reverse process (deterministic sampler) once a score-based model is trained from the *forward* Stochastic Differential Equation (SDE) diffusion process [16], [18], [49]. For generative models which are flow-based in the forward process, generation guarantee for flow-matching models under continuous-time formulation was shown in [3] under $\mathcal{W}_2$, and in [2] under the Kullback-Leibler (KL) divergence by incorporating additional SDE diffusion. The current paper focuses on obtaining the theoretical guarantee of the JKO flow model [74], which is progressively trained over the Residual Blocks (steps) and generates a discrete-time flow in both forward and reverse directions. The mathematical formulation of the JKO flow is summarized in Section III, where we introduce needed theoretical assumptions on the learning procedure.

Our analysis is based on first proving the convergence of the forward process (the JKO scheme by flow network), which can be viewed as an approximate proximal Gradient Decent (GD) in the Wasserstein-2 space to minimize $G(\rho)$, a functional on the space of probability distributions. While the convergence analyses of Wasserstein GD and proximal GD

have appeared previously in literature [40], [63], our setup differs in several ways, primarily in that we consider the JKO scheme, which is a "fully-backward" discrete-time GD. For the N step discrete-time proximal GD, which produces a sequence of transported distributions p_n, we prove the convergence of both $\mathcal{W}_2(p_n, q)$ and the objective gap $G(p_n) - G(q)$ at an exponential rate, where q is the global minimum of G (Theorem 10). The convergence applies to a general class of (strongly) convex G that includes the KL divergence $\mathrm{KL}(p\|q)$ as a special case. This result echos the classical proximal GD convergence in vector space where one expects an exponential convergence rate for strongly convex minimizing objectives. While exponential convergence is a natural result from the point of view of gradient flow, this convergence result of JKO-type $\mathcal{W}_2$-proximal GD did not appear in previous literature to the authors' best knowledge and can be of independent interest.

After obtaining a small $G(p_n) = \mathrm{KL}(p_n\|q)$ from the convergence of the forward process, we directly obtain the KL guarantee of the generated density from the data density by the invertibility of the flow and the data processing inequality, and this implies the total variation (TV) guarantee (Corollary 12). The requirement for data distribution is to have a finite second moment and a density (with respect to the Lebesgue measure). The TV and KL guarantees are of $O(\varepsilon)$ and $O(\varepsilon^2)$, respectively, where ε is the bound for the magnitude of the Wasserstein (sub-)gradient of the loss function (hence error in the first order condition) in each of the N JKO steps (Assumption 1), and the process achieves the error bound in $N \lesssim \log(1/\varepsilon)$ many steps (each step is a Residual Block).

To handle the situation when the data distribution only has a finite second moment but no density, we apply a short-time initial diffusion and start the forward process from the smoothed density. This short-time diffusion was adopted in practice and prior theoretical works. We then obtain KL and TV guarantee to generate the smoothed density, which can be made arbitrarily close to the data distribution in $\mathcal{W}_2$ when the initial diffusion time duration tends to zero (Corollary 13). The above results are obtained when the reverse process is computed exactly with no inversion error. Our analysis can also extend to the case of small inversion error by proving a $\mathcal{W}_2$-guarantee between the generated density from the exact reverse process and that from the actual computed one. Theoretically, the $\mathcal{W}_2$-error can be made $O(\varepsilon)$ or smaller assuming that the inversion error can be made $O(\varepsilon^\alpha)$ for some exponent α (Corollary 15).

A. Normalizing Flow Models

1) Normalizing Flow: Normalizing flow is a class of deep generative models for efficient sampling and density estimation. Compared to diffusion models, Continuous Normalizing Flow (CNF) models [44] appear earlier in the generative model literature. Largely speaking, CNF models fall into two categories: discrete-time and continuous-time. The discrete-time CNF models adopt the structure of a Residual Network (ResNet) [32] and typically consist of a sequence of mappings:

$$x_l = x_{l-1} + f_l(x_{l-1}), \quad l = 1, \cdots, L, \tag{1}$$
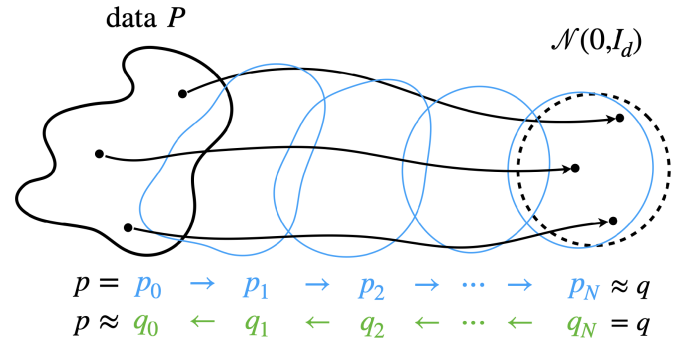

Fig. 1. The arrows indicate the forward-time flow from data distribution P to normal distribution q. The forward and reverse processes (19) consist of the sequence of transported densities at discrete time stamps.

where f_l is the neural network mapping parameterized by the l-th "Residual Block", and x_l is the output of the l-th block. Continuous-time CNFs are implemented under the neural ODE framework [15], where the neural network features $x(t)$ is computed by integrating an ODE

$$\dot{x}(t) = v(x(t), t), \quad t \in [0, T], \tag{2}$$

and $v_t(x) = v(x, t)$ is parametrized by a neural ODE network. The discrete-time CNF (1) can be viewed as computing the numerical integration of the neural ODE (2) on a sequence of time stamps via the forward Euler scheme.

In both categories, a CNF model computes a deterministic transport from the data distribution towards a target distribution q typically normal, $q = \mathcal{N}(0, I_d)$, per the name "normalizing." The forward time flow is illustrated in Figure 1. Taking the continuous-time formulation (2), let P be the data distribution with density p, $x(0) \sim p$, and denote by $p_t(x) = p(x, t)$ the probability density of $x(t)$. Then p_t solve the continuity equation (CE)

$$\partial_t p_t + \nabla \cdot (p_t v_t) = 0, \tag{3}$$

from $p_0 = p$. If the algorithm can find a v_t such that p_T at some time T is close to q, then one would expect the reverse-time flow from $t = T$ to $t = 0$ to transport from q to a distribution close to p. Note that in the continuous-time flow, invertibility is presumed since the neural ODE can be integrated in two directions of time alike. For discrete-time flow (1), invertibility needs to be ensured either by special designs of the neural network layer type [24], [25], [43], or by regularization techniques such as spectral normalization [9] or transport cost regularization [58], [72].

A notable advantage of the flow model is the computation of the likelihood. For discrete-time flow (1), this involves the computation of the log-determinant of the Jacobian of f_l. For continuous-time flow (2), this is by the relation

$$\log p_t(x(t)) - \log p_s(x(s)) = -\int_s^t \nabla \cdot v(x(\tau), \tau) d\tau,$$

which involves the time-integration of the trace of the Jacobian of v_t [29]. While these computations may encounter challenges in high dimensions, the ability to evaluate the (log) likelihood is fundamentally useful; in particular, it allows for evaluating

the maximum likelihood training objective on finite samples. This property is also adopted in the deterministic reverse process in diffusion models [67], called the "probability flow ODE" (see more in Section I-B.1), so the likelihood can be evaluated once a forward diffusion model has been trained.

2) Progressive Flow Models: Another line of works, developed around the same time as diffusion models, explored the variational form of normalizing flow as a Wasserstein gradient flow and proposed the so-called *progressive* training of the flow model.

The progressive training of ResNet, i.e., training block-wise by a per-block variational loss, was proposed by [38] at an earlier time under the GAN framework. Later on, the Jordan-Kinderleherer-Otto (JKO) scheme, as a time-discretized Wasserstein gradient flow (see more in Section II-C), was explored in several flow-based generative models: [4], [54] implemented the JKO scheme using input convex neural networks [8], [26] proposed a forward progressive flow from noise to data, showing empirical success in generating high-dimensional real datasets; [74] developed the JKO flow model under the invertible continuous-time CNF framework, achieving competitive generating performance on high-dimensional real datasets at a significantly reduced computational and memory cost from previous CNF models; an independent concurrent work [70] proposed a block-wise JKO flow model utilizing the framework of [58]. Many other flow models related to diffusion and Optimal Transport (OT) exist in the literature; see more in Section I-B.2. Our theoretical analysis will focus on the progressive flow models, and we primarily follow the invertible flow framework in [74].

To be more specific, a progressive flow model represents the flow on $[0, T]$ as the composition of N sub-flow models where each one computes the flow on a sub-interval $[t_{n-1}, t_n]$, $n = 1, \ldots, N$. The training is "progressive", meaning that at one time, only one sub-model is trained, and the training of the n-th sub-model is conducted once the previous $n-1$ sub-models are trained and fixed. The progressive block-wise training is in contrast of the end-to-end training, where the flow on $[0, T]$ (or N Residual Blocks) is trained simultaneously by a single objective. The sub-flow model on $[t_{n-1}, t_n]$ can take different forms, e.g., a ResNet block or a continuous-time neural ODE, and the N sub-intervals always provide a time-discretization of the flow. In this context, we call the sub-model on the n-th sub-interval a "Residual Block".

B. Additional Related Works

1) Score-Based Diffusion Models: In score-based diffusion models, the algorithm first simulates a forward process, which is a (time-discretized) SDE, from which a score function parameterized as a neural network is trained. The reverse (SDE or ODE) process is simulated using the learned score model to generate data samples.

a) SDE in diffusion models: As a typical example, in the *variance preserving* Denoising Diffusion Probabilistic Modeling (DDPM) process [33], [67], the forward process produces a sequence of X_n,

$$X_n = \sqrt{1 - \beta_n} X_{n-1} + \sqrt{\beta_n} Z_{n-1}, \quad n = 1, \cdots, N, \quad (4)$$

where $Z_n \sim \mathcal{N}(0, I_d)$ i.i.d. and $X_0 \sim P$ is drawn from data distribution. With large N, the continuum limit of the discrete dynamic (4) is a continuous-time SDE,

$$dX_t = -\frac{1}{2} \beta(t) X_t dt + \sqrt{\beta(t)} dW_t, \quad t \in [0, T], \quad (5)$$

where $\beta(t) > 0$ is a function and W_t is a standard Wiener process (Brownian motion). Since $\beta(t)$ in (5) corresponds to a time reparametrization of t, after changing the time ($t \mapsto \int_0^t \beta(s)/2 ds$), (5) becomes the following SDE

$$dX_t = -X_t dt + \sqrt{2} dW_t, \quad (6)$$

which is the Ornstein-Uhlenbeck (OU) process in $\mathbb{R}^d$. We consider the time-parametrization in (6) for exhibition simplicity. More generally, one can consider a diffusion process

$$dX_t = -\nabla V(X_t) dt + \sqrt{2} dW_t, \quad X_0 \sim P, \quad (7)$$

and the OU process is a special case with $V(x) = \|x\|^2/2$.

We denote by $\rho_t = \mathcal{L}_t(P)$ the marginal distribution of X_t for $t > 0$. The time evolution of ρ_t is described by the Fokker-Planck Equation (FPE) written as

$$\partial_t \rho_t = \nabla \cdot (\rho_t \nabla V + \nabla \rho_t). \quad (8)$$

b) Forward and reverse processes: When simulating the forward process, the diffusion models train a neural network to learn the score function $s_t(x) := \nabla \log \rho_t$ by score matching [36], [71]. The training objective can be expressed as the mean-squared error defined as $\int_0^T \int \|\hat{s}_t(x) - s_t(x)\|^2 \rho_t(x) dx dt$, which facilitates training and is scalable to high dimension data such as images (in the original pixel space).

Once the neural-network score function $\hat{s}_t$ is learned, the algorithm simulates a reverse-time SDE $\tilde{X}_t$ (with time discretization in practice) [67], such that from $\tilde{X}_T \sim \mathcal{N}(0, I)$ the distribution of $\tilde{X}_0$ is expected to the close to the data distribution P. It has also been proposed in [67] to compute the reverse process by integrating the following ODE reverse in time

$$\dot{\tilde{x}}(t) = -\nabla V(\tilde{x}(t)) - s_t(\tilde{x}(t)), \quad (9)$$

and (9) was called the "probability flow ODE." The validity of this ODE reverse process can be justified by the observation that the CE (3) and FPE (8) are the same when setting $v_t(x) = -(\nabla V(x) + s_t(x))$. This equivalence between density evolutions by SDE and ODE has been known in the literature of diffusion processes and solving FPE, dating back to the 90s [21], [22].

2) Flow Models Related to Diffusion and OT:

a) Flow-matching models: After diffusion models gained popularity, several flow-based models (in the reverse and forward directions) closely related to the diffusion model emerged. In particular, the Flow-Matching ODE model was proposed in [50] using the formulation conditional probability paths, where a neural ODE parameterized $\hat{v}(x, t)$ is trained to match a velocity field $v(x, t)$ whose corresponding CE (3) can evolve the density p_t towards normality. The algorithm can adopt diffusion paths, where the CE will equal the density evolution equation (8) of an SDE forward process, as well

as non-diffusion paths. A similar approach was developed under the "stochastic interpolant" framework in [3], where the terminal distribution q can be arbitrary (not necessarily the normal distribution) and only accessible via finite samples. These models train a continuous-time CNF by minimizing a "matching" objective instead of the maximum likelihood objective as in [29], thus avoiding the computational challenges of the latter.

b) Optimal transport flows: Apart from diffusion models and Wasserstein gradient flow, Wasserstein distance and OT have inspired another line of works on flow models where the Wasserstein distance, or a certain form of transport cost, is used to regularize the flow model and to compute the OT map between two distributions. Transport cost regularization of neural network models was suggested in several places: [62] provided a general framework for solving high-dimensional mean-field games (MFG) and control problems, [27] proposed a kinetic regularization aiming to stabilize neural ODE training, [58] and [72] developed the transport regularization in CNF and invertible ResNet, respectively, and [34] applied to MFG and flow models. Other works developed flow models to compute the optimal coupling or the optimal transport between two distributions. For example, Rectified Flow [51] proposed an iterative method to adjust the flow towards the optimal coupling. The method is closely related to the stochastic interpolant approach [3] which, in principle, can solve the OT trajectory if the interpolant map can be optimized. A flow model to compute the dynamic OT between two high dimensional distributions from data samples was proposed in [73] by refining the flow using the transport cost from a proper initialization. Despite the wealth of methodology developments and empirical results, the theoretical guarantees of these flow models are yet to be developed.

3) Theoretical Guarantees of Generative Models:

a) Approximation and estimation of GAN: On theoretical guarantees of generative models, earlier works focused on the approximation and estimation analysis under the GAN framework. The expressiveness of a deep neural network to approximate high dimensional distributions was established in a series of works, e.g., [46], [52], [60], and [75], among others. The neural network architectures in these universal approximation results are typically feed-forward, like the generator network (G-net) proposed in the original GAN. The approximation and estimation of the discriminator network (D-net) in GAN were studied in [19], and the problem can be cast and analyzed as the learning of distribution divergences in high dimension [68]. Convergence analysis of GAN was studied in several places, e.g., [35].

b) Guarantees of diffusion models: An earlier work [69] studied the expressiveness of a generative model using a latent diffusion process and proved guarantees for sampling and inference; however, the approach only involves a forward process and differs from the recent diffusion models. Motivated by the prevailing empirical success of score-based diffusion models, recent theoretical works centralized on the generation guarantee of such models using both SDE and ODE samplers, i.e., the reverse process.

For the SDE reverse process, the likelihood guarantee of the score-based diffusion model was first derived in [65] without time discretization. Taking into account the time discretization, which significantly influences the efficiency in practice, a series of theoretical studies have established polynomial convergence bounds for such models [11], [14], [17], [20], [47], [48], [59]. In particular, [47] were the first to attain polynomial convergence without succumbing to the curse of dimensionality, although this required a log-Sobolev Inequality on the data distribution. For a general data distribution, [17] achieved polynomial error bounds in Total Variation (TV) distance under the Lipschitz assumption, leveraging Girsanov's theorem. In parallel, [48] derived similar polynomial convergence bounds, employing a technique for converting L^{∞}-accurate score estimates into L^2-accurate score estimation. Further advancements by [14] established a more refined bound, reducing the requirement of smoothness of data distribution. Most recently, [59] improved the convergence rates under mild assumptions by introducing prediction-correction, and [11] established the first convergence bounds for diffusion models, which are linear in the data dimension (up to logarithmic factors) without requiring any smoothness of the data distribution.

c) Guarantees of ODE flows: Within the studies of score-based diffusion models (note that the forward process is always SDE), theoretical findings for the ODE reverse process are relatively fewer. To the best of our knowledge, [18] established the first non-asymptotic polynomial convergence rate where the error bound involves an exponential factor in the flow time; [16] provided the first polynomial-time convergence guarantees for the probability flow ODE implementation with a corrector step. Recently, [49] established bounds for both deterministic (ODE) and non-deterministic (SDE) samplers under certain additional assumptions on learning the score. The analysis is done by directly tracking the density ratio between the law of the diffusion process and that of the generated process in discrete time, leading to various non-asymptotic convergence rates.

Compared to score-based diffusion models, the guarantees of flow models (in both forward and reverse processes) are significantly less developed. We are aware of two recent works: The error bounds for the flow-matching model [3] were proved in [10] and applied to probability flow ODE in score-based diffusion models; for neural ODE models trained by likelihood maximization (the framework in [29]), [53] proved non-parametric statistical convergence rates to learn a distribution from data. Both works used a continuous-time formulation, and the flow models therein are trained end-to-end. Compared to end-to-end training, progressive flow models can offer advantages in training efficiency and accuracy, in addition to other advantages such as smaller model complexity. For the analysis, the formulation of progressive flow models is variational and time-discretized in nature. Theoretical studies of time-discretized ODE flow models in both forward and reverse directions remain rudimentary.

4) Optimization in Wasserstein Space: Continuing the classical literature in optimization and information geometry, several recent works established a convergence guarantee of

first-order optimization in probability space in various contexts, leveraging the connection to the Wasserstein gradient flow. These analyses can potentially be leveraged under the theoretical framework of this paper to develop new (progressive) flow models as well as theoretical guarantees of generative models.

a) Optimization in probability distribution space: Convergence and rate analysis for first-order methods for vector-space optimization, primarily gradient descent and stochastic gradient descent — sometimes referred to as the Sample Average Approximation (SAA) approach — for convex and strongly convex problems have been established in the original works [55], [56], and extended in various contexts in subsequent papers. Optimization in the space of probability distributions (which forms a manifold) naturally arises in many learning problems and has become an important field of study in statistics and machine learning. In particular, the seminal work of Amari [5], [6] introduced information geometry emerging from studies of a manifold of probability distributions. It includes convex analysis and its duality as a special but important component; however, the line of work did not develop error analysis or convergence rates for algorithms on the probabilistic manifold. More recently, a Frank-Wolfe procedure in probability space was proposed in [40] motivated by applications in nonparametric estimation and was shown to converge exponentially fast under general mild assumptions on the objective functional.

b) Wasserstein proximal gradient descent: The landmark work [39] showed the solution to the Fokker Planck equation as the gradient flow of the KL divergence under the $\mathcal{W}_2$-distance. The proof in [39] employed a time discretization of the gradient flow now recognized as the JKO scheme. Making a connection between Langevin Monte Carlo and Wasserstein gradient flow, [12] proposed a proximal version of the Unadjusted Langevin Algorithm corresponding to a splitting scheme of the discrete Wasserstein GD and derived non-asymptotic convergence analysis. To analyze the convergence of discrete-time $\mathcal{W}_2$-gradient flow, [63] introduced a Forward-Backward time discretization in the proximal Wasserstein GD and proved convergence guarantees akin to the GD algorithm in Euclidean spaces. We comment on the difference between [63] and our scheme in more detail later in Remark 3.

The JKO scheme also inspired recent studies in variational inferences (VI). In the context of Gaussian VI, [45] proposed gradient flow of the KL divergence on the Bures-Wasserstein (BW) space, namely the space of Gaussian distributions on $\mathbb{R}^d$ endowed with the $\mathcal{W}_2$-distance. The algorithm enjoys the explicit solution of the JKO scheme in the BW space, and convergence of the continuous-time gradient flow was proved. In a follow-up work [23], the forward-backward splitting was adopted in the proximal Wasserstein GD in the BW space, leading to convergence guarantees of the discrete-time GD to first-order stationary solutions. The closed-form solution of the JKO operator only applies to the BW space, while the JKO flow network tries to learn a transport map to solve the JKO scheme in each step, leveraging the expressiveness of neural networks. Theoretically, we consider distributions with finite second moments in this work.

C. Notations

Throughout the paper, we consider distributions over $\mathcal{X}$ and the domain $\mathcal{X} = \mathbb{R}^d$. We denote by $\mathcal{P}_2$, meaning $\mathcal{P}_2(\mathbb{R}^d)$, the space of probability distributions on $\mathbb{R}^d$ that has finite second moment. Specifically, for a distribution P, define $M_2(P) := \int_{\mathbb{R}^d} \|x\|^2 dP(x)$. When P has a density (with respect to the Lebesgue measure dx), we also write $M_2(P)$ as $M_2(p)$. Then $\mathcal{P}_2 = \{P \text{ on } \mathbb{R}^d, s.t., M_2(P) < \infty\}$. We denote by $\mathcal{P}_2^r$ the distributions in $\mathcal{P}_2$ that have densities, namely $\mathcal{P}_2^r = \{P \in \mathcal{P}_2, P \ll dx\}$. We also say a density $p \in \mathcal{P}_2^r$ when $dP(x) = p(x)dx$ is in $\mathcal{P}_2^r$. In this paper, we consider distributions that have densities in most places. When there is no confusion, we use the density p to stand for both the density and the distribution $dP(x) = p(x)dx$, e.g., we say that a random variable $X \sim p$.

Given a (measurable) map $v : \mathbb{R}^d \to \mathbb{R}^d$ and P a distribution on $\mathbb{R}^d$, its L^2 norm is denoted as $\|v\|_P := (\int_{\mathbb{R}^d} \|v(x)\|^2 dP(x))^{1/2}$. When P has density p, we also denote it as $\|v\|_p$. For $P \in \mathcal{P}_2$, we denote by $L^2(P)$ (and also by $L^2(p)$ when P has density p) the L^2 space of vector fields, that is, $L^2(P) := \{v : \mathbb{R}^d \to \mathbb{R}^d, \|v\|_P < \infty\}$. For $u, v \in L^2(P)$, define $\langle u, v \rangle_P := \int_{\mathbb{R}^d} u(x)^T v(x) dP(x)$, which is also denoted as $\langle u, v \rangle_p$ when p is the density. The notation $\mathrm{I_d}$ stands for the identity map, which is always in $L^2(P)$ for $P \in \mathcal{P}_2$. For $T : \mathbb{R}^d \to \mathbb{R}^d$, the *pushforward* of a distribution P is denoted as $T_\# P$, such that $T_\# P(A) = P(T^{-1}(A))$ for any measurable set A. When P has density p and $T_\# P$ also has a density, we also denote by $T_\# p$ the density of $T_\# P$. For two maps $S, T : \mathbb{R}^d \to \mathbb{R}^d$, $S \circ T$ is the function composition.

II. PRELIMINARIES

A. Wasserstein-2 Distance and Optimal Transport

We first review the definitions of the Wasserstein-2 distance and optimal transport (OT) map, which are connected by the Brenier Theorem (see, e.g., [7, Section 6.2.3]).

Given two distributions $\mu, \nu \in \mathcal{P}_2$, the Wasserstein-2 distance $\mathcal{W}_2(\mu, \nu)$ is defined as

$$\mathcal{W}_2^2(\mu, \nu) := \inf_{\pi \in \Pi(\mu, \nu)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2 \, d\pi(x, y), \quad (10)$$

where $\Pi(\mu, \nu)$ denotes the family of all joint distributions with μ and ν as marginal distributions. When P and Q are in $\mathcal{P}_2^r$ and have densities p and q respectively, we also denote $\mathcal{W}_2(P, Q)$ as $\mathcal{W}_2(p, q)$. When at least one of μ and ν has density, we have the Brenier Theorem, which allows us to define the optimal transport (OT) map from μ to ν.

Theorem 1 (Brenier Theorem): Let $\mu \in \mathcal{P}_2^r$ and $\nu \in \mathcal{P}_2$. Then

(i) There exists a unique minimizer π of (10), which is characterized by a uniquely determined μ-*a.e.* map $T_\mu^\nu : \mathbb{R}^d \to \mathbb{R}^d$ such that $\pi = (\mathrm{I_d}, T_\mu^\nu)_\# \mu$, where $(\mathrm{I_d}, T_\mu^\nu)$ maps (x, y) to $(x, T_\mu^\nu(y))$. Moreover, there exists a convex function $\varphi : \mathbb{R}^d \to \mathbb{R}$ such that $T_\mu^\nu = \nabla \varphi$ μ-a.e.

(ii) The minimum of (10) equals that of the Monge problem, namely

$$\mathcal{W}_2^2(\mu, \nu) = \inf_{T : \mathbb{R}^d \to \mathbb{R}^d, \, T_\# \mu = \nu} \int \|x - T(x)\|^2 d\mu(x).$$

(iii) If moreover $\nu \in \mathcal{P}_2^r$, then we also have the OT map T_ν^μ defined ν-a.e., and $T_\nu^\mu \circ T_\mu^\nu = \mathrm{I_d}$ μ-a.e., $T_\mu^\nu \circ T_\nu^\mu = \mathrm{I_d}$ ν-a.e.

In most places in our analysis, we will consider the OT between μ and ν both in $\mathcal{P}_2^r$, and we will frequently use the Brenier Theorem (iii) to obtain the pair of OT maps which are inverse of each other in the a.e. sense.

B. Differential and Convexity of Functionals on $\mathcal{P}_2$

Consider a proper lower semi-continuous functional $\phi :$ $\mathcal{P}_2 \to (-\infty, \infty]$ and we denote the domain to be $\mathrm{Dom}(\phi) = \{\mu \in \mathcal{P}_2, \phi(\mu) < \infty\}$. The subdifferential of ϕ was defined in the Fréchet sense, see, e.g., Definition 10.1.1 of [7]. We recall the definition of strong subdifferential as below.

Definition 2 (Strong Subdifferential): Given $\mu \in \mathcal{P}_2$, a vector field $\xi \in L^2(\mu)$ is a strong (Fréchet) subdifferential of ϕ at μ if for $v \in L^2(\mu)$,

$$\phi((\mathrm{I_d} + v)_{\#}\mu) - \phi(\mu) \ge \langle \xi, v \rangle_\mu + o(\|v\|_\mu).$$

We denote by $\partial_{\mathcal{W}_2}\phi(\mu)$ the set of strong Fréchet subdifferentials of ϕ at μ (which may be empty).

There can be different ways to introduce convexity of functions on $\mathcal{P}_2$. The most common way is the convexity along geodesics, also known as "displacement convexity." In our analysis, we technically need the notation of convexity *along generalized geodesics* (a.g.g.), which is stronger than geodesic convexity. In short, displacement convexity is along the geodesic from μ_1 to μ_2, which, in the simple case where there is a unique OT map T_1^2 from μ_1 to μ_2, is defined using interpolation $(1-t)\mathrm{I_d} + tT_1^2$ for $t \in [0, 1]$. In contrast, convexity a.g.g. involves a third distribution ν and is defined using interpolation of the two OT maps from ν to μ_1 and μ_2 respectively.

Specifically, let $\nu \in \mathcal{P}_2^r$, $\mu_i \in \mathcal{P}_2$, $i = 1, 2$, and let T_ν^i be the OT map from ν to μ_i respectively. A *general geodesic* joining μ_1 to μ_2 (with base ν) is a curve of type

$$\mu_t^{1 \to 2} := ((1-t)T_\nu^1 + tT_\nu^2)_{\#}\nu, \quad t \in [0, 1]. \tag{11}$$

Definition 3 (Convexity Along Generalized Geodesics): For $\lambda \ge 0$, a functional ϕ on $\mathcal{P}_2$ is said to be λ-convex along generalized geodesics (a.g.g.) if for any $\nu \in \mathcal{P}_2^r$ and $\mu_1, \mu_2 \in \mathcal{P}_2$ and $\forall t \in [0, 1]$,

$$\phi(\mu_t^{1 \to 2}) \le (1-t)\phi(\mu_1) + t\phi(\mu_2) - \frac{\lambda}{2}t(1-t)\mathcal{W}_\nu^2(\mu_1, \mu_2), \tag{12}$$

where $\mu_t^{1 \to 2}$ is as in (11) and

$$\mathcal{W}_\nu^2(\mu_1, \mu_2) := \int_{\mathbb{R}^d} \|T_\nu^1(x) - T_\nu^2(x)\|^2 d\nu(x) \ge \mathcal{W}_2^2(\mu_1, \mu_2). \tag{13}$$

Note that the definition implies the following property which is useful in our analysis, $\forall t \in [0, 1]$,

$$\phi(\mu_t^{1 \to 2}) \le (1-t)\phi(\mu_1) + t\phi(\mu_2) - \frac{\lambda}{2}t(1-t)\mathcal{W}_2^2(\mu_1, \mu_2). \tag{14}$$

The definition of convexity a.g.g. in [7, Section 9.2] is for the more general case when ν may not have density and the OT maps from ν to μ_i need to be replaced with optimal plans, and then the generalized geodesics may not be unique. In this paper, we only consider the case where ν has a density, so we simplify the definition, see [7, Remark 9.2.3] (and make it slightly weaker, but there is no harm for our purpose).

We also note that many functionals $\phi(\mu)$ on $\mathcal{P}_2$ that are geodesically convex actually also satisfy the convexity a.g.g. in Definition 3. Examples include $\phi(\mu) = \int V(x)d\mu(x)$ with V convex on $\mathbb{R}^d$, $\phi(\mu) = \int \int W(x_1, x_2)d\mu(x_1)d\mu(x_2)$ with convex W, and $\phi(\rho) = \int F(\rho(x))dx$, ρ being the density, where F is convex on $[0, \infty)$. (In these examples, V, W, and F need to satisfy other technical conditions.) The last example includes negative entropy as a special case, where $F(x) = x \log x$. The primary case for our work is when ϕ is the KL divergence, which will be discussed in more detail in Section IV-A. We refer to [7, Section 9] for other examples and detailed discussions of convex a.g.g. functionals.

C. JKO Scheme for Fokker-Planck Equations

Consider the diffusion process (7) starting from $P \in \mathcal{P}_2$. It is known that under generic conditions, as $t \to \infty$, ρ_t converges to the equilibrium distribution of (7) which has density

$$q \propto e^{-V}, \tag{15}$$

and the convergence is exponentially fast [13]. The function V is called the *potential function* of q.

The evolution of ρ_t by FPE of the diffusion process can be interpreted as a continuous-time gradient flow under the $\mathcal{W}_2$-metric in the probability space $\mathcal{P}_2$. The JKO scheme [39] computes a Wasserstein proximal GD which is a time discretization of the gradient flow. Specifically, define $G : \mathcal{P}_2^r \to \mathbb{R}$ as the KL divergence w.r.t. q, i.e.,

$$G(\rho) = \mathrm{KL}(\rho \| q) = \mathcal{H}(\rho) + \mathcal{E}(\rho),$$
$$\mathcal{H}(\rho) = \int \rho \log \rho, \quad \mathcal{E}(\rho) = c + \int V\rho, \tag{16}$$

where c is a constant. More general G can be considered, see Section IV-A, and in this work we mainly focus on the case where G is the KL divergence as being considered in [39].

Under certain regularity condition of V, the JKO scheme computes a sequence of distributions ρ_n, $n = 0, 1, \ldots$, starting from $\rho_0 \in \mathcal{P}_2$. For a fixed step size $\gamma > 0$, and the scheme at the n-th step can be written as

$$\rho_{n+1} = \arg \min_{\rho \in \mathcal{P}_2} G(\rho) + \frac{1}{2\gamma}\mathcal{W}_2^2(\rho_n, \rho). \tag{17}$$

The scheme computes the $\mathcal{W}_2$-proximal Gradient Descent (GD) of G with step size γ, and can be written as

$$\rho_{n+1} = \mathrm{Prox}_{\gamma G}(\rho_n). \tag{18}$$

The original JKO paper [39] proved the convergence of the discrete-time solution $\{\rho_n\}$ (after interpolation over time) to the continuous-time solution ρ_t of the FPE (8) when step size $\gamma \to 0+$. In the context of flow-based generative models by neural networks, the discrete-time JKO scheme with finite γ

was adopted and implemented as a flow network in [74]. Our analysis in this work will prove the exponential convergence of ρ_n to q by the JKO scheme (including learning error), echoing the exponential convergence of the continuous-time dynamic (the FPE). This result leads to the guarantee of generating data distributions up to (TV) error $O(\varepsilon)$ in $O(\log(1/\varepsilon))$ JKO steps. We will summarize the flow model and introduce theoretical assumptions in Section III.

III. SETUP OF JKO FLOW MODEL AND ASSUMPTIONS

In this section, we summarize the mathematical setup for the JKO flow model and introduce the necessary theoretical assumptions for our analysis. The guarantee of generating the data distribution will be derived in Section V based on the exponential convergence of the $\mathcal{W}_2$-proximal GD (JKO scheme) in Section IV.

A. Forward and Reverse Processes of JKO Flow Model

As has been introduced in Section I-A, the flow model implements an ODE model (transport equation), where both the forward process and the reverse process are computed by an invertible Residual Network [9] or a neural-ODE network [15], [29]. The forward process consists of N steps, where each step is computed by a Residual Block — in the neural-ODE model, this is the neural ODE integration on a sub-time-interval $[t_n, t_{n+1}]$, and we also call it a Residual Block. The backward process consists of the N steps of the same flow network "backward in time," where each step computes the inverse map of the Residual Block, and in the neural-ODE model, this is via integrating the ODE in reverse time.

The forward and reverse processes (without inversion error) are induced by a sequence of transport maps, T_n, $n = 1, \ldots, N$, which we will define more formally later. The two processes are summarized in (19),

$$
\text{(forward)} \qquad p = p_0 \xrightarrow{T_1} p_1 \xrightarrow{T_2} \cdots \xrightarrow{T_N} p_N \approx q,
$$
$$
\text{(reverse)} \qquad p \approx q_0 \xleftarrow{T_1^{-1}} q_1 \xleftarrow{T_2^{-1}} \cdots \xleftarrow{T_N^{-1}} q_N = q. \quad (19)
$$

where p is the density of data distribution (when exists, otherwise a smoothified density by a short time diffusion), and q is the equilibrium density, typically chosen as Gaussian. Inversion error in the reverse process is considered in Section III-C.

a) Forward process: In the forward process, the algorithm learns a sequence of T_n, which transports from data distribution P to the equilibrium distribution Q, which is typically the normal distribution, $\mathcal{N}(0, I)$. We denote by q the density of Q, and p the density of the data distribution P when there is one.

Following the neural-ODE framework used in [74], each step computes a transport map $T_{n+1} : \mathbb{R}^d \to \mathbb{R}^d$ which is the solution map of the ODE from time t_n to t_{n+1}, i.e.,

$$
T_{n+1}(x_n) = x(t_{n+1}), \quad (20)
$$

where $x(t)$ solves $\dot{x}(t) = \hat{v}(x(t), t)$ on $[t_n, t_{n+1}]$, $x(t_n) = x_n$, and $\hat{v}(x, t)$ is the velocity field on $\mathbb{R}^d$ parametrized by the n-th

Residual Block. Equivalently, we have

$$
T_{n+1}(x_n) = x_n + \int_{t_n}^{t_{n+1}} \hat{v}(x(t), t) dt, \quad x(t_n) = x_n. \quad (21)
$$

In the implementation of the JKO scheme in a flow network, the learning of the N Residual Blocks is conducted progressively for $n = 1, \cdots, N$ by minimizing a training objective per step [74]. We emphasize that, unlike other normalizing flow models, which are trained end-to-end, the training procedure here is done step-wise and progressively over the N Residual Blocks.

Once T_{n+1} is learned, it pushes from p_n to p_{n+1}, i.e.,

$$
p_{n+1} = (T_{n+1})_\# p_n. \quad (22)
$$

In our problem, we want the distributions in the intermediate steps to have a density. To guarantee that this is the case for p_{n+1}, we technically need T_{n+1} to be *non-degenerate*. Intuitively, a non-degenerate map cannot collapse a set of finite (Lebesgue) measures into a set of measure zero.

Definition 4 (Non-Degenerate Map): Denote by Leb the Lebesgue measure. $T : \mathbb{R}^d \to \mathbb{R}^d$ is non-degenerate if for any set $A \subset \mathbb{R}^d$ s.t. $Leb(A) = 0$, then $Leb(T^{-1}(A)) = 0$.

If a transport map is non-degenerate, then it pushes forward a density to a distribution that also has density, as shown in the following lemma proved in Appendix A.

Lemma 5: Suppose $T : \mathbb{R}^d \to \mathbb{R}^d$ is non-degenerate, $P \ll Leb$, then $T_\# P \ll Leb$.

Assuming T_n are all non-degenerate, then the sequence of p_n produced by (22) all have densities, starting from $p_0 = p$ the data density. When data distribution has no density, we will introduce an initial short-time diffusion that mollifies the data distribution into ρ_δ which we set to be p_0 (see more in Section V-A.2). The learning aims that after N steps, the final p_N is close to the equilibrium density q.

b) Reverse process (without inversion error): The reverse process computes the inverse of the N-steps transport by inverting each T_n in the forward process. We first assume that T_n can be exactly inverted in computation, which allows for a simplified analysis. In practice, T_n^{-1} can be implemented by fixed-point iteration [9] or reverse-time ODE integration [29]. The case when the inverse cannot be exactly computed is discussed in Section III-C, where we need additional assumptions on the closeness of the computed inverse to the true inverse of T_n for our analysis.

The reverse process outputs generated samples, which are aimed to be close in distribution to the data samples, by drawing samples from q and pushing them through the reverse N steps. In terms of the sequence of probability densities generated by the process, the reverse process computes

$$
q_n = (T_{n+1}^{-1})_\# q_{n+1}, \quad (23)
$$

starting from $q_N = q$ and the output density is q_0. Theoretically, the data processing inequality for the KL divergence applied to invertible transforms (Lemma 11) guarantees that if p_N is close to $q_N = q$, then q_0 is close to p_0, which is the data density (possibly after short-time smoothing). This allows us to prove the guarantee of $q_0 \approx p_0$ once we can prove that of $p_N \approx q$, the latter following the convergence of

the $\mathcal{W}_2$-proximal GD up to a hopefully small learning error to be detailed below.

B. Learning Assumptions of the Forward Process

We consider the sequence of densities p_n in the forward process in (19). Recall from Section II-C that for fixed step-size $\gamma > 0$, the n-th step classical JKO scheme finds p_{n+1} by minimizing

$$\min_{\rho \in \mathcal{P}_2} F_{n+1}(\rho) := G(\rho) + \frac{1}{2\gamma} \mathcal{W}_2^2(p_n, \rho), \quad (24)$$

where $G(\rho) = \mathrm{KL}(\rho || q)$. The learning in the n-th Residual Block in a JKO flow network computes the minimization via parameterizing the transport T_{n+1}. Here we briefly review the rational of solving (24) by solving for T_{n+1}, which leads to our assumption of the learned forward process.

a) JKO step by learning the transport: In the right hand side of (24), when both p_n and ρ are in $\mathcal{P}_2^r$, the Brenier Theorem (Theorem 1) implies the existence of a unique OT map T from p_n to ρ. Consider the following minimization over the transport T,

$$\min_{T: \mathbb{R}^d \to \mathbb{R}^d} G(T_{\#}p_n) + \frac{1}{2\gamma} \mathbb{E}_{x \sim p_n} \|x - T(x)\|^2. \quad (25)$$

The following lemma, proved in Appendix A, shows that the minimizer T makes $T_{\#}p_n \in \mathcal{P}_2^r$:

Lemma 6: Suppose $p_n \in \mathcal{P}_2^r$ makes $G(p_n) < \infty$, and T is a minimizer of (25), then $T_{\#}p_n \in \mathcal{P}_2^r$.

Thus, in (25) it is equivalent to minimize over T that renders $T_{\#}p_n \in \mathcal{P}_2^r$ and this means that the minimizer T is the OT map. One can also verify that (25) is equivalent to (24) in the sense that a minimizer T^* of (25) makes $(T^*)_{\#}p_n$ a minimizer of (24), and for a minimizer ρ^* of (24) the OT map from p_n to ρ^* is a minimizer of (25) [74, Lemma A.1].

b) Learning error in JKO flow network: In the n-step of the JKO flow network, the transport T in (25) is parameterized by a Residual block, and the learning cannot find the T that exactly minimizes (25) (and equivalently (24)) for three reasons:

 (i) Approximation error: The minimization of (25) is over T constrained inside some neural network family $\mathcal{T}_\Theta$. When the function family $\mathcal{T}_\Theta$ is large enough to express the desired optimal transport from p_n to $\mathrm{Prox}_{\gamma G}(p_n)$, the solution can approximate the exact minimizer of (24), but this usually cannot be guaranteed.
 (ii) Finite-sample effect: The training is computed on empirical data samples, while in this analysis, we focus on the minimization of population loss.
 (iii) Imperfect optimization: The learning of neural networks is a non-convex optimization typically implemented by Stochastic Gradient Descent (SGD) over mini-batches and there is no guarantee of achieving a minimizer of the empirical loss.

As a result, the learned transport T_{n+1} finds a $p_{n+1} = (T_{n+1})_{\#}p_n$ that at most approximately minimizes (24). While the learned T_{n+1} is usually not the exact minimizer, we assume that it is regular enough such that p_{n+1} is still in $\mathcal{P}_2^r$. This

would hold if T_{n+1} is non-degenerate and also in $L^2(p_n)$ by the following lemma proved in Appendix A.

Lemma 7: Let $p \in \mathcal{P}_2^r$, then $\{T : \mathbb{R}^d \to \mathbb{R}^d, T \in L^2(p)\} = \{T : \mathbb{R}^d \to \mathbb{R}^d, T_{\#}p \in \mathcal{P}_2\}$. As a result, if $T \in L^2(p)$ and is non-degenerate, then $T_{\#}p \in \mathcal{P}_2^r$.

When p_n and p_{n+1} are both in $\mathcal{P}_2^r$, we have a unique invertible OT map from p_n to p_{n+1} by Brenier Theorem, and its (a.e.) inverse is the OT map from p_{n+1} to p_n. Specifically, we define

$$T_n^{n+1} \text{ is the OT map from } p_n \text{ to } p_{n+1}, \quad p_n\text{-a.e.},$$
$$T_{n+1}^n \text{ is the OT map from } p_{n+1} \text{ to } p_n, \quad p_{n+1}\text{-a.e.},$$
$$(26)$$

and we have $T_{n+1}^n \circ T_n^{n+1} = \mathrm{Id}$ p_n-a.e., $T_n^{n+1} \circ T_{n+1}^n = \mathrm{Id}$ p_{n+1}-a.e. Note that T_n^{n+1} differs from the learned map T_{n+1}, see Remark 2.

c) Assumption on approximate first-order condition: For our analysis, we theoretically characterize the error in learning T_{n+1} by quantifying the error in the first-order condition. Specifically, the $\mathcal{W}_2$-gradient (strictly speaking, a sub-differential) of F_{n+1} at ρ can be identified as

$$\nabla_{\mathcal{W}_2} F_{n+1}(\rho) = \nabla_{\mathcal{W}_2} G(\rho) - \frac{T_\rho^n - \mathrm{Id}}{\gamma}, \quad \rho\text{-a.e.} \quad (27)$$

where T_ρ^n is the OT map from ρ to p_n. Here we use $\nabla_{\mathcal{W}_2}\phi$ to denote the sub-differential $\partial_{\mathcal{W}_2}\phi$ assuming unique existence to simplify exhibition. The formal statement in terms of subdifferential is provided in Lemma 16 (which follows the argument of [7, Lemma 10.1.2]) for more general G (which includes KL divergence as a special case). For KL divergence G, if V is differentiable, the sub-differential $\partial_{\mathcal{W}_2} G$ is reduced to the unique $\mathcal{W}_2$-gradient written as

$$\nabla_{\mathcal{W}_2} G(\rho) = \nabla V + \nabla \log \rho, \quad (28)$$

when $\rho \in \mathcal{P}_2^r$ has a well-defined score function.

If ρ is the exact minimizer of (24), we will have $\nabla_{\mathcal{W}_2} F_{n+1}(\rho) = 0$ (and for sub-differential the condition is $0 \in \partial_{\mathcal{W}_2} F_{n+1}(\rho)$). At p_{n+1} which is pushed-forward by the learned T_{n+1}, we denote the $\mathcal{W}_2$-gradient of F_{n+1} by the following (recall the definition of T_{n+1}^n as in (26))

$$\xi_{n+1} := \nabla_{\mathcal{W}_2} F_{n+1}(p_{n+1})$$
$$= \nabla_{\mathcal{W}_2} G(p_{n+1}) - \frac{T_{n+1}^n - \mathrm{Id}}{\gamma}, \quad p_{n+1}\text{-a.e.} \quad (29)$$

and it is interpreted as sub-differential when needed. Making an analogy to the (approximate) first-order condition in vector-space optimization, we assume that p_{n+1} is close to the exact minimizer such that the (sub-)gradient ξ_{n+1} is small. In practice, the SGD algorithm to minimize the training objective (25) (assuming $\mathcal{T}_\Theta$ is expressive enough to approximate the exact minimizer T^*) would stop making progress when the $\mathcal{W}_2$-gradient vector field ξ_{n+1} evaluated on data samples collectively give a small magnitude. We characterize this by a small $L^2(p_{n+1})$ norm of the (sub-)gradient ξ_{n+1} in our theoretical assumption. The assumptions on the learned transport T_{n+1} are summarized as follows:

Assumption 1 (Approximate n-th Step Solution): The learned transport T_{n+1} is non-degenerate and in $L^2(p_n)$; it is

invertible on $\mathbb{R}^d$ and T_{n+1}^{-1} is also non-degenerate. In addition, for some $\varepsilon > 0$, $\exists \xi_{n+1} \in \partial_{\mathcal{W}_2} F_{n+1}(p_{n+1})$ s.t.

$$\|\xi_{n+1}\|_{p_{n+1}} \leq \varepsilon. \tag{30}$$

We experimentally verify the smallness of $\|\xi_{n+1}\|_{p_{n+1}}$ in neural network training of one JKO step. Figure 3 shows the decrease of the squared L^2 norm $\|\xi_{n+1}\|_{p_{n+1}}^2$ over training iterations. The Wasserstein gradient ξ_{n+1} is plotted as a vector field in Figure 4, whose magnitude gradually decreases over training. See Appendix D for more details.

The error magnitude ε can be viewed as an algorithmic parameter that reflects the accuracy of first-order methods, and similar assumptions have been made in the analysis of stochastic (noisy) gradient descent in vector space, see, e.g., [55] and [56]. We emphasize that theoretically, ε does not need to be small but will enter the final error bound. For T_{n+1} satisfying Assumption 1, from $p_n \in \mathcal{P}_2^r$, $p_{n+1} = (T_{n+1})_{\#} p_n$ is also in $\mathcal{P}_2^r$ by Lemma 7. Then the subdifferential $\partial_{\mathcal{W}_2} F_{n+1}$ can be defined at p_{n+1} and characterized by Lemma 16.

Remark 1 (Assumptions on T_n): The L^2 integrability condition of T_n is natural and together with non-degeneracy ensures that the next p_n is in $\mathcal{P}_2^r$. When T_n^{-1} is also non-degenerate, the p_n's in the forward process and q_n's in the reverse process in (19) all have densities, and this allows us to apply the data processing inequality in both directions (Lemma 11).

When there is an inversion error in the reverse process, we will further impose Lipschitzness of T_n^{-1} on $\mathbb{R}^d$ and the Lipschitz constant will theoretically enter the error bound, see more in Section V-B.

Remark 2 (T_{n+1} and T_n^{n+1}): Recall that T_{n+1} is the learned transport map and T_n^{n+1} is the OT map. In our setting (of imperfect minimization in the n-th step), both T_{n+1} and T_n^{n+1} push p_n to p_{n+1} but they are not necessarily the same. The notion of T_n^{n+1} is introduced only for theoretical purposes (the existence and invertibility are by Brenier Theorem), and our theory do not need T_{n+1} to equal T_n^{n+1}. On the other hand, one would expect T_{n+1} to approximate T_n^{n+1} when the JKO-step optimization (24) is approximately solved, leading to a small ε in (30).

C. Reverse Process With Inversion Error

Considering potential inversion error in the reverse process, we denote the sequence of transports as S_n and the transported densities as $\tilde{q}_n$, that is,

$$\tilde{q}_n = (S_{n+1})_{\#} \tilde{q}_{n+1}, \tag{31}$$

from $\tilde{q}_N = q_N = q$. The reverse process with and without inversion error is summarized as

$$
\begin{aligned}
\text{(exact reverse)} \quad & q_0 \xleftarrow{T_1^{-1}} q_1 \xleftarrow{T_2^{-1}} \cdots \xleftarrow{T_N^{-1}} q_N = q, \\
\text{(computed reverse)} \quad & \tilde{q}_0 \xleftarrow{S_1} \tilde{q}_1 \xleftarrow{S_2} \cdots \xleftarrow{S_N} \tilde{q}_N = q.
\end{aligned}
\tag{32}
$$

The computed transport S_n is not the same as T_n^{-1} but the algorithm aims to make the inversion error small. For our theoretical analysis, we make the following assumption on the error.

Assumption 2 (Inversion Error): For $n = N, \cdots, 1$, the computed reverse transport S_n is non-degenerate, in $L^2(\tilde{q}_n)$, and satisfies that

$$\|T_n \circ S_n - \mathrm{Id}\|_{\tilde{q}_n} \leq \varepsilon_{\mathrm{inv}}. \tag{33}$$

In practice, the quantity $\|T_n \circ S_n - \mathrm{Id}\|_{\tilde{q}_n}^2$ can be empirically estimated by sample average, namely the mean-squared error

$$\mathrm{MSE}_{\mathrm{inv}} = \frac{1}{n_{\mathrm{inv}}} \sum_i \|T_n \circ S_n(x_i) - x_i\|^2,$$
$$x_i = S_{n+1} \circ \cdots \circ S_{N-1}(z_i), \quad z_i \sim q,$$

computed from n_{inv} test samples. With sufficiently large n_{inv}, one can use $\mathrm{MSE}_{\mathrm{inv}}$ to monitor the inversion error of the reverse process and enhance numerical accuracy when needed. It was empirically shown in [74] that the inversion error computed on testing samples (though all the N blocks) can be made small towards the floating-point precision in the neural-ODE model.

The inversion error objective (33) resembles the "cycle consistency" loss in Cycle-GAN [76]. Our theory in Section V-B suggests that keeping the inversion error small is crucial for the success of generating a close-to-data distribution $\tilde{q}_0$ in the reverse process. Thus, there may be benefits by introducing the objective (33) as a regularization in the training of flow-based models, similar to the cycle consistency mechanisms in the GAN literature.

IV. CONVERGENCE OF FORWARD PROCESS

The current paper mainly concerns the application to flow generative networks where G is the KL divergence. In this section, we prove the exponentially fast convergence of the forward process, which applies to potentially a more general class of G as long as the subdifferential calculus in $\mathcal{P}_2$ can be conducted (see Section 10.1 of [7]) and G is strongly convex along generalized geodesics (a.g.g., see Definition 3), which may be of independent interest.

We will revisit the KL divergence G as a special case and prove the generation guarantee of the reverse process in Section V. All proofs and technical lemmas are provided in Appendix B.

A. Conditions on G and V

We introduce the more general condition of G needed by the forward process convergence.

Assumption 3 (General Condition of G): $G : \mathcal{P}_2 \rightarrow (-\infty, +\infty]$ is lower semi-continuous, $\mathrm{Dom}(G) \subset \mathcal{P}_2^r$; G is λ-convex a.g.g. in $\mathcal{P}_2$.

The first part of Assumption 3 ensures that the strong subdifferential $\partial_{\mathcal{W}_2} G(\rho)$ can be defined, see Definition 2. The strong convexity of G is used to prove the exponential convergence of the (approximated) $\mathcal{W}_2$-proximimal Gradient Descent in the forward process.

Next, we show that the KL divergence G satisfies the general condition under certain general conditions of the potential function V plus its strong convexity. We also introduce an upper bound of λ due to a rescaling argument.
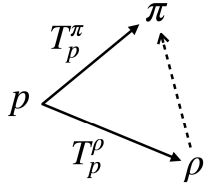
$$p, \pi, \rho \in \mathcal{P}_2^r, \ G \text{ is } \lambda\text{-convex a.g.g.,}$$

$$G(\pi) - G(\rho) \geq \left\langle \nabla_{\mathcal{W}_2} G(\rho) \circ T_p^\rho, T_p^\pi - T_p^\rho \right\rangle_p + \frac{\lambda}{2} \mathcal{W}_2^2(\pi, \rho)$$

$$\pi, \rho \in \mathbb{R}^d, \ g \text{ is } \lambda\text{-convex,}$$

$$g(\pi) - g(\rho) \geq \langle \nabla g(\rho), \pi - \rho \rangle + \frac{\lambda}{2} \|\pi - \rho\|^2$$

Fig. 2.   The monotonicity of a.g.g.-convex G in $\mathcal{P}_2$ proved in Lemma 8, as an analog to strong convexity in vector space. We remark that in the usual vector space, the convexity definition does not involve a third vector, since the inner product is uniform; while in probability space, inner product is defined at tangent space associated with p. The dotted line indicates the general geodesic between ρ and π, see the definitions in Section II-B.

1) KL Divergence and f-Divergence G: Recall that the KL divergence $G(\rho) = \mathcal{H}(\rho) + \mathcal{E}(\rho)$ as defined in (16), where $\mathcal{E}(\rho) = \int V\rho$ involves the potential function V of the equilibrium density. We introduce the following assumption on V:

Assumption 4 (Condition of V the Potential of q): The potential function $V : \mathbb{R}^d \to (-\infty, \infty]$ is proper, lower semi-continuous, and $V^- := \max\{-V, 0\}$ is bounded; $V(x)$ is λ-strongly convex on $\mathbb{R}^d$, and $q \propto e^{-V}$ is in $\mathcal{P}_2^r$.

The first part of Assumption 4 is for the sub-differential calculus of $\mathcal{E}(\rho) = \int V\rho$ in $\mathcal{P}_2$. The λ-strong convexity is used to make $\mathcal{E}(\rho)$ (and subsequently $G(\rho)$) λ-convex a.g.g. Under such condition of V, the KL divergence $G(\rho)$ satisfies Assumption 3, which is verified in Lemma 19. Thus our result applies to the KL divergence as being used in the JKO flow model. For the important case when q is standard normal, $V(x) = \|x\|^2/2$, and $\lambda = 1$.

More generally, consider the f-divergence $G(\rho) = D_f(\rho\|q) := \int f(\rho/q)q$, $f : (0, \infty) \to \mathbb{R}$ being convex, lower-semicontinuous and $f(1) = 0$. KL divergence corresponds to a special case where $f(x) = x \log x$. Under Assumption 4, we have $q \propto e^{-V}$ and is log-concave, and then $G(\rho)$ is convex a.g.g. in $\mathcal{P}_2^r$ (under additional technical conditions on f) [7, Theorem 9.4.12]. We think it is possible to establish the λ-convexity a.g.g. of f-divergence G for a certain class of f, and details are postponed here.

2) Positive and Bounded λ: Note that for strongly convex V we can use a scaling argument to make λ bounded to be $O(1)$ without loss of generality. Specifically, for V that is λ-convex on $\mathbb{R}^d$, the function $x \mapsto V(ax)$ for $a > 0$ is $(a^2\lambda)$-convex. This means that for q that has a strongly convex V as the potential function, one can rescale samples from q to make V strongly convex with $\lambda \leq 1$. In the case where G is KL divergence, the λ-convexity of G has the same λ as that of V. Thus, for the general G we assume its λ is also bounded by 1.

Assumption 5 (λ Bounded): In Assumptions 3 and 4, $0 < \lambda \leq 1$.

Our technique can potentially extend to analyze the $\lambda = 0$ case, where an algebraic $O(1/n)$ convergence rate is expected instead of the exponential rate proved in Theorem 10. For the application to flow-based generative model, one would need the equilibrium density $q \propto e^{-V}$ convenient to sample from, and thus the normal density (corresponding to $V(x) = \|x\|^2/2$) is the most common choice and the other choices usually render $\lambda > 0$ (to enable fast sampling of the starting distribution). We thus leave the $\lambda = 0$ case to future work.

B. Evolution Variational Inequality and Convergence of the Forward Process

The a.g.g. λ-convexity of G leads to the following lemma, which is important for our analysis. All proofs in this section are provided in Appendix B.

Lemma 8 (Monotonicity of G): Let $p, \rho \in \mathcal{P}_2^r$, $\pi \in \mathcal{P}_2$, and denote by T_p^ρ and T_p^π the OT maps from p to ρ and to π respectively. Suppose G satisfies Assumption 3, then for any $\eta \in \partial_{\mathcal{W}_2} G(\rho)$,

$$G(\pi) - G(\rho) \geq \left\langle \eta \circ T_p^\rho, T_p^\pi - T_p^\rho \right\rangle_p + \frac{\lambda}{2} \mathcal{W}_2^2(\pi, \rho).$$

The relationship among p, ρ, π is illustrated in Figure 2, which also includes an analog to the strong-convex function in Euclidean space. This lemma extends Lemma 5 in [63] and originally the argument in Section 10.1.1.B of [7]. We include a proof in Appendix B for completeness.

Based on the monotonicity lemma and the condition of small strong subdifferential ξ_{n+1} in Assumption 1, we are ready to derive the discrete-time *Evolution Variational Inequality* (EVI) [7, Chapter 4] for the (approximate) JKO scheme.

Lemma 9 (EVI for Approximate JKO Step): Given $\pi \in \mathcal{P}_2$, suppose G satisfies Assumption 3 with $\lambda \in (0, 1]$, and $0 < \gamma < 2$. If $p_0 \in \mathcal{P}_2^r$, and Assumption 1 holds for $n = 0, 1, \cdots$, then for all n,

$$\left(1 + \frac{\gamma\lambda}{2}\right) \mathcal{W}_2^2(p_{n+1}, \pi) + 2\gamma \left(G(p_{n+1}) - G(\pi)\right)$$

$$\leq \mathcal{W}_2^2(p_n, \pi) + \frac{2\gamma}{\lambda} \varepsilon^2. \tag{34}$$

The condition of $\lambda \leq 1$ and $\gamma < 2$ can be replaced with other constants, and our analysis will give similar results. Specifically, the upper bound 1 of λ in Assumption 5 is a generic choice and we keep it, then replacing the requirement of $\gamma < 2$ with $\gamma < \gamma_{\max}$ for another $\gamma_{\max} > 1$ will only affect the constant in the final bound and does not affect the order. For exhibition simplicity, we give our analysis under $\lambda \leq 1$ and $\gamma < 2$ without loss of generality. We also provide some rationale for upper bounding the step size γ motivated by practice: First, it has been empirically observed that successful computation of the JKO flow model in practice needs the step size not to exceed a certain maximum value, which is an algorithmic parameter [74]. Setting the step size too large may lead to difficulty in training the Residual blocks as well as in maintaining small inversion errors. Meanwhile, from the formulation of the JKO scheme, it can be seen that for large γ, the proximal GD in (17) approaches the global minimization

of $G(\rho)$, which asks for the flow to transport from the current density to the target density q in one step. Though the proximal GD (as a backward Euler scheme) does not impose a step-size constraint, the optimization problem (17) is, in principle, easier with a small (but no need to converge to zero) step size.

The EVI directly leads to the N-step convergence of the forward process, which achieves $O(\varepsilon)$ $\mathcal{W}_2$-error and $O(\varepsilon^2)$ gap from the optimal objective value in $N \lesssim \log(1/\varepsilon)$ JKO steps.

Theorem 10 (Convergence of Forward Process): Suppose $q \in \mathcal{P}_2$ is the global minimum of G, and the other assumptions are the same as Lemma 9, then for $n = 1, 2, \cdots$,

$$\mathcal{W}_2^2(p_n, q) \le \left(1 + \frac{\gamma\lambda}{2}\right)^{-n} \mathcal{W}_2^2(p_0, q) + \frac{4\varepsilon^2}{\lambda^2}. \tag{35}$$

In particular, if

$$n \ge \frac{8}{\gamma\lambda} \left(\log \mathcal{W}_2(p_0, q) + \log(\lambda/\varepsilon)\right), \tag{36}$$

then

$$\mathcal{W}_2(p_n, q) \le \sqrt{5}\frac{\varepsilon}{\lambda}, \quad G(p_{n+1}) - G(q) \le \frac{9}{2\gamma}\left(\frac{\varepsilon}{\lambda}\right)^2. \tag{37}$$

Remark 3 (Comparison to [63]): The convergence rates of Wasserstein proximal GD were previously studied in [63], and our proof techniques, namely the monotonicity of G plus discrete-time EVI, are similar to the analysis therein. However, the setups differ in several aspects: first, we consider the "fully-backward" proximal GD, i.e., the JKO scheme, while [63] focuses on the forward-backward scheme to minimize G having a decomposed form $\mathcal{E} + \mathcal{H}$ (which does cover the KL convergence as a special case). Second, [63] assumed the exact solution of the proximal step while our analysis takes into account the error ε in the first-order condition Assumption 1, which is more realistic for neural network-based learning. At last, [63] assumed L-smoothness of V, namely ∇V is L-Lipschitz, and step size $\gamma < 1/L$, as a result of the forward step in the splitting scheme, which is not needed in our fully backward scheme. The motivation for [63] is to understand the discretized Wasserstein gradient flow and to recover the same convergence rates as the (forward-backward) proximal GD in the vector space. Our analysis is motivated by the JKO flow network, and the forward process convergence is an intermediate result to prove the generation guarantee of the reverse process.

V. GENERATION GUARANTEE OF REVERSE PROCESS

In this section, we first consider the reverse process as in (19), called the *exact* reverses process, when there is no inversion error. In Section V-A, we prove a KL (and TV) guarantee of generating by q_0 any data distribution P in $\mathcal{P}_2^r$, and extend to P with no density by introducing a short-time initial diffusion.

Taking into account the inversion error, we consider the sequence $\tilde{q}_n$ induced by S_n in (32), called the *computed* reverse process, where the inversion error satisfies Assumption 2. In Section V-B, we prove a closeness bound of $\tilde{q}_0$ to q_0 in $\mathcal{W}_2$, which leads to a $\mathcal{W}_2$-KL mixed generation guarantee of $\tilde{q}_0$ based on the proved guarantee of q_0.

A. Convergence Guarantee Without Inversion Error

We start by presenting convergence analysis assuming there are no errors in the reverse process; then, we extend to the more practical situation considering inversion errors. All proofs in this section are given in Appendix C.

1) KL (TV) Guarantee of Generating P With Density: Consider the transport map over the N steps in (19) denoted as

$$T_1^N := T_N \circ \cdots \circ T_1. \tag{38}$$

Since each T_n is invertible (Assumption 1), the overall mapping T_1^N is also invertible. We have

$$p_N = (T_1^N)_{\#}p_0, \quad q_N = (T_1^N)_{\#}q_0.$$

The following lemma follows from the data processing inequality of KL, which allows us to obtain KL bound of $p_0 = p$ and q_0 from that of p_N and $q_N = q$.

Lemma 11 (Bi-Direction Data Processing Inequality): If $T : \mathbb{R}^d \to \mathbb{R}^d$ is invertible and for two densities p and q on $\mathbb{R}^d$, $T_{\#}p$ and $T_{\#}q$ also have densities, then

$$\mathrm{KL}(p||q) = \mathrm{KL}(T_{\#}p||T_{\#}q).$$

The following corollary establishes an $O(\varepsilon^2)$ KL bound in $N \lesssim \log(1/\varepsilon)$ JKO steps, which implies an $O(\varepsilon)$ TV bound by Pinsker's inequality.

Corollary 12 (KL guarantee for $P \in \mathcal{P}_2^r$): Suppose $G(\rho) = \mathrm{KL}(\rho||q)$, the potential function V satisfies Assumption 4 with $\lambda \in (0, 1]$, and $0 < \gamma < 2$. Suppose $P \in \mathcal{P}_2^r$ with density p, let $p_0 = p$, and Assumption 1 holds for some ε and all n. Then, let

$$N = \left\lceil \frac{8}{\gamma\lambda} \left(\log \mathcal{W}_2(p_0, q) + \log(\lambda/\varepsilon)\right) \right\rceil, \tag{39}$$

the generated density q_0 of the reverse process satisfies that

$$\mathrm{KL}(p||q_0) \le \frac{9}{2\gamma}\left(\frac{\varepsilon}{\lambda}\right)^2, \quad \mathrm{TV}(p, q_0) \le \frac{3}{2\sqrt{\gamma}}\frac{\varepsilon}{\lambda}. \tag{40}$$

Remark 4 (Extension to f-Divergence): As has been discussed in Section IV-A, it is possible to show that $G(\rho) = D_f(\rho||q)$ satisfies Assumption 3 (possibly under additional conditions of f), and then the convergence of the forward process, Theorem 10, extends to such f-divergence G. In addition, data processing inequality holds generally for f-divergence, and thus Lemma 11 also extends. As a result, Corollary 12 can potentially extend to certain f-divergences and show a guarantee of $D_f(p||q_0)$ in N JKO steps.

2) Guarantee of Generating $P \in \mathcal{P}_2$ up to Initial Short Diffusion: For $P \in \mathcal{P}_2$ that may not have a density, we first obtain $\rho_\delta \in \mathcal{P}_2^r$ that is close to P in $\mathcal{W}_2$ by a short-time initial diffusion (specifically, the OU process as introduced in Section II-C) up to time $\delta > 0$, as shown in Lemma 20. The short-time initial diffusion was used in [48] and called "early stopping" in [14]. It is also used in practice by flow model [74] as well as score-based diffusion models to bypass the irregularity of data distribution [67]. In principle, one can also use the Brownian motion only (corresponding to convolving P with Gaussian kernel) to obtain ρ_δ. Here we use the OU process to stay in line with the literature.

The introduction of ρ_δ allows us to prove a guarantee of $\mathrm{KL}(\rho_\delta \| q_0)$ in the following corollary, which is the same type of result as [14, Theorem 2].

Corollary 13 (KL Guarantee for $P \in \mathcal{P}_2$ From ρ_δ): Suppose $P \in \mathcal{P}_2$, and the conditions on G, V, λ and γ are the same as in Corollary 12. Then $\forall \varepsilon' > 0$, there exists $\delta > 0$ s.t. $\mathcal{W}_2(P, \rho_\delta) < \varepsilon'$ and, with $p_0 = \rho_\delta$ and Assumption 1 holds for some ε and all n, let N as in (39), the generated density q_0 of the reverse process makes $\mathrm{KL}(\rho_\delta \| q_0)$ and $\mathrm{TV}(\rho_\delta, q_0)$ satisfy the same bounds as in (40).

The corollary shows that there is a density $\rho_\delta \in \mathcal{P}_2^r$ that is arbitrarily close to P in $\mathcal{W}_2$, such that the output density q_0 of the reverse process can approximate ρ_δ up to the same error as in Corollary 12. Note that the corollary holds when the potential function V of q satisfies the general condition Assumption 4. The OU process in the proof (Lemma 20) is only used in constructing ρ_δ, and there are other means to construct the surrogate initial density ρ_δ.

B. Convergence Guarantee With Inversion Error

Recall the set-up from Section III-C. To prove the $\mathcal{W}_2$ control between $\tilde{q}_0$ and q_0, we first introduce a Lipschitz condition on the inverse of the learned transport map T_n and explain the motivation.

1) Lipschitz Constant of Computed Transport Maps: Previously in Assumption 1, we required that both T_n and its inverse are non-degenerate. Here, we further require that T_n^{-1} is globally Lipschitz on $\mathbb{R}^d$ with a uniform Lipschitz constant.

Assumption 6 (Lipschitz condition on T_n^{-1}): There is $K > 0$ s.t. T_n^{-1} is Lipschitz on $\mathbb{R}^d$ with Lipschitz constant $e^{\gamma K}$ for all $n = N, \ldots, 1$.

The assumed Lipschitz constants are theoretical and motivated by neural ODE models, to be detailed below. Our analysis of $\mathcal{W}_2(\tilde{q}_0, q_0)$ applies to any type of flow network (like invertible ResNet) as long as the needed assumptions on T_n and S_n hold.

We justify the assumptions on T_n, T_n^{-1} and S_n under the framework of neural ODE flow, namely (20)(21), including the Lipschitz constant $e^{\gamma K}$ of T_n^{-1}. Specifically, by the elementary Lemma 21 proved in Appendix C, we know that if T_{n+1} can be numerically exactly computed as (21) and $\hat{v}(x,t)$ on $\mathbb{R}^d \times [t_n, t_{n+1}]$ satisfies a uniform x-Lipschitz condition with Lipschitz constant K, then both T_{n+1} and its inverse are Lipschitz on $\mathbb{R}^d$ with Lipschitz constant $e^{\gamma K}$. We will assume the same K throughout time for simplicity. In practice, Lipschitz regularization techniques can be applied to the neural network parametrized $\hat{v}(x,t)$, and the global Lipchitz bound of $\hat{v}$ on $\mathbb{R}^d$ can be achieved by "clipping" $\hat{v}$ to vanish outside some bounded domain of x. Meanwhile, note that if $T : \mathbb{R}^d \to \mathbb{R}^d$ is invertible and T^{-1} is globally Lipschitz on $\mathbb{R}^d$, then T is non-degenerate. Thus T_{n+1} and T_{n+1}^{-1} both being non-degenerate are implied by (and weaker than) the global Lipschitzness of T_{n+1} and its inverse. In addition, in a neural-ODE-based flow model, the reverse process is by integrating the neural ODE in reverse time, and thus we can expect similar properties of S_n.

While the computed transport T_n and S_n often differ from the exact numerical integration of the ODE, we still expect the Lipschitz property to retain. For general flow models, which may not be neural ODE, we impose the same theoretical assumptions. At last, the global Lipschitz condition may be theoretically relaxed by combining with truncation arguments of the probability distributions, which is postponed here.

2) $\mathcal{W}_2$-Control of the Computed Reverse Process From the Exact One:

Proposition 14: Suppose in (32), $q_N = \tilde{q}_N = q \in \mathcal{P}_2^r$, and the computed transport maps T_n and S_n satisfy Assumptions 1, 2 and 6. Then all q_n and $\tilde{q}_n$ are in $\mathcal{P}_2^r$ and

$$\mathcal{W}_2(\tilde{q}_0, q_0) \le \frac{\varepsilon_{\mathrm{inv}}}{\gamma K} e^{\gamma K(N+1)}. \tag{41}$$

A continuous-time counterpart of Proposition 14 was derived in [3, Proposition 3]. We include a proof in Appendix C for completeness. The proof uses a coupling argument of the (discrete-time) ODE flow, which as has been pointed out in [16], obtains a growing factor $e^{\gamma K N}$ in the $\mathcal{W}_2$-bound as shown in (41). To overcome this exponential factor, [16] adopted an SDE corrector step. Here, without involving any corrector step, we show that the factor $e^{\gamma K N}$ can be controlled at the order of some negative power of ε thanks to the exponential convergence in the forward process. This is because N can be chosen to be at the order of $\log(1/\varepsilon)$ as in (39), then $e^{\gamma K N}$ can be made $O(\varepsilon^{-\alpha})$ for some $\alpha > 0$. As a result, the $\mathcal{W}_2$-error (41) can be suppressed if $\varepsilon_{\mathrm{inv}}$ can be made smaller than a higher power of ε.

More specifically, combined with the analysis of the forward process, we arrive at the following guarantee, proved in Appendix C.

Corollary 15 (Mixed Bound With Inversion Error): Suppose $G(\rho) = \mathrm{KL}(\rho \| q)$, the potential function V satisfies Assumption 4 with $\lambda \in (0, 1]$, and $0 < \gamma < 2$. Suppose the computed transports maps T_n and S_n satisfy the Assumptions 1, 2, 6 for some ε and $\varepsilon_{\mathrm{inv}}$ for all n. Suppose $P \in \mathcal{P}_2^r$ with density p, let $p_0 = p$ and N as in (39), then the generated density $\tilde{q}_0$ of the computed reverse process satisfies that

$$\mathcal{W}_2(\tilde{q}_0, q_0) \le \frac{e^{2\gamma K}}{\gamma K} (\mathcal{W}_2(p_0, q)\lambda)^{8K/\lambda} \frac{\varepsilon_{\mathrm{inv}}}{\varepsilon^{8K/\lambda}}, \tag{42}$$

and q_0 satisfies the KL and TV bounds to p as in (40).

Remark 5 ($O(\varepsilon)$ error and need for small $\varepsilon_{\mathrm{inv}}$): The corollary implies that if $\varepsilon_{\mathrm{inv}}$ can be made small, then the $\mathcal{W}_2$ bound can be made equal to or smaller than ε in order. For example, if $\varepsilon_{\mathrm{inv}} = O(\varepsilon^{8K/\lambda+1})$, then we have $\mathcal{W}_2(\tilde{q}_0, q_0) = O(\varepsilon)$. This suggests that if one focuses on getting $\mathrm{KL}(p_N \| q)$ small in the forward process, then maintaining an inversion error small is crucial for the generation quality of the flow model in the reverse process.

At last, when P is merely in $\mathcal{P}_2$ and does not have density, then one can start the forward process from $p_0 = \rho_\delta$ same as in Section V-A.2. Then we have the same $\mathcal{W}_2$-bound between $\tilde{q}_0$ and q_0 as in (42), and q_0 is close to ρ_δ in the sense of Corollary 13.

VI. DISCUSSION

The work can be extended in several directions. First, it is interesting to see if the assumption on learning in the

forward process, Assumption 1, can be derived from further analysis of the neural network learning, e.g., the approximation and optimization error (c.f. the list of sources of errors in Section III-B). The current work does not contain such analysis and instead handles the goodness of the learned T_n by a single assumption. In particular, it would be of interest to theoretically justify the assumed "first order condition," i.e., the smallness of the $\mathcal{W}_2$-gradient ξ_{n+1} in (30), by analyzing the convergence of the optimization. One possibility is by showing the weak convergence of the learned p_{n+1} to the exact minimizer p^* of F_{n+1} and then utilizing the convergence of $\nabla_{\mathcal{W}_2} F_{n+1}(p)$ to $\nabla_{\mathcal{W}_2} F_{n+1}(p^*) = 0$ in a proper sense [7, Section 5.4]. Second, the current generation result only covers the case of G being the KL divergence. An extension to the cases when G is other types of divergence, such as f-divergence (see Remark 4), will broaden the scope of the result. Third, our theory uses the population quantities throughout. A finite-sample analysis, which can be based on our population analysis, will provide statistical convergence rates in addition to the current result.

Meanwhile, the JKO scheme computes a fully backward proximal GD. Given the existing convergence rates of the other Wasserstein GD [40], [63], one would expect that a variety of first-order Wasserstein GD schemes can be applied to progressive flow models and the theoretical guarantees can be derived similarly to the JKO scheme. We also note the connection between the JKO scheme and learning of the score function, at least in the limit of small step size [74, Section 3.2]. Given the growing literature on the analysis of score-based diffusion models, it can be worthwhile to investigate this connection further to develop new theories for the ODE flow models.

Finally, it would be interesting to use theory to guide practice and to develop new or improved methodologies of flow-based generative models. As discussed in Section III-C, one may consider incorporating the inversion error as part of training loss to enforce the accuracy of the reverse process. The potential theoretical extension to f-divergences also suggests utilizing more general f-divergence as the per-step training objective in JKO flow networks, e.g., by adopting techniques in f-GAN [57]. It would be interesting to explore different choices of f and the relationships among the f-divergences [31]. In addition, our theory indicates that using larger step-size γ leads to a shorter sequence of Residual Blocks in the network architecture (as long as the optimization in each JKO step can be efficiently solved). It would be natural to consider an adaptive choice of γ in practice and also in extending the theory.

APPENDIX A
PROOFS AND LEMMAS IN SECTION III

A. Lemma on the $\mathcal{W}_2$-(sub)gradient

Lemma 16: Suppose $G : \mathcal{P}_2(X) \to (-\infty, +\infty]$ is lower semi-continuous and $\mathrm{Dom}(G) \subset \mathcal{P}_2^r$. Let $\gamma > 0$, $p \in \mathcal{P}_2^r$, and

$$F(\rho) = G(\rho) + \frac{1}{2\gamma}\mathcal{W}_2^2(p, \rho). \tag{A.1}$$

If (at $\rho \in \mathcal{P}_2^r$, $\partial_{\mathcal{W}_2} F(\rho)$ is non empty and) $\xi \in \partial_{\mathcal{W}_2} F(\rho)$, then

$$\xi + \frac{T_\rho^p - \mathrm{I_d}}{\gamma} \in \partial_{\mathcal{W}_2} G(\rho).$$

The argument follows that in Lemma 10.1.2 in [7], and we include a proof for completeness.

Proof: [Proof of Lemma 16] We are to verify that

$$\eta := \xi + \frac{T_\rho^p - \mathrm{I_d}}{\gamma}$$

is a strong subdifferential of G at ρ. By Definition 2, it suffices to show that for any $v \in L^2(\rho)$ and $\delta \to 0$,

$$G((\mathrm{I_d} + \delta v)_\#\rho) - G(\rho) \geq \delta\langle\eta, v\rangle_\rho + o(\delta). \tag{A.2}$$

By construction,

$$\langle\eta, v\rangle_\rho = \langle\xi, v\rangle_\rho + \frac{1}{\gamma}\langle T_\rho^p - \mathrm{I_d}, v\rangle_\rho,$$

and since ξ is a strong subdifferential of F at ρ,

$$F((\mathrm{I_d} + \delta v)_\#\rho) - F(\rho) \geq \delta\langle\xi, v\rangle_\rho + o(\delta).$$

Combining the two and by the definition of F, we can deduce (A.2) as long as we can show that

$$\frac{1}{2}\mathcal{W}_2(p, (\mathrm{I_d}+\delta v)_\#\rho)^2 + o(\delta) \leq \frac{1}{2}\mathcal{W}_2(p, \rho)^2 - \langle T_\rho^p - \mathrm{I_d}, \delta v\rangle_\rho. \tag{A.3}$$

To show (A.3), note that by Brenier Theorem (ii),

$$\mathcal{W}_2(p, \rho)^2 = \int \|x - T_\rho^p(x)\|^2\rho(x)dx = \|\mathrm{I_d} - T_\rho^p\|_\rho^2.$$

Thus,

$$\frac{1}{2}\mathcal{W}_2(p, \rho)^2 - \langle T_\rho^p - \mathrm{I_d}, \delta v\rangle_\rho$$
$$= \frac{1}{2}\|\mathrm{I_d} - T_\rho^p\|_\rho^2 + \langle \mathrm{I_d} - T_\rho^p, \delta v\rangle_\rho$$
$$= \frac{1}{2}\|(\mathrm{I_d} + \delta v) - T_\rho^p\|_\rho^2 - \frac{1}{2}\|\delta v\|_\rho^2. \tag{A.4}$$

Note that, because $v \in L^2(\rho)$,

$$\|\delta v\|_\rho^2 = O(\delta^2)$$

and

$$\|(\mathrm{I_d} + \delta v) - T_\rho^p\|_\rho^2 = \int_{\mathbb{R}^d} \|(\mathrm{I_d} + \delta v)(x) - T_\rho^p(x)\|^2\rho(x)dx$$
$$\geq \mathcal{W}_2((\mathrm{I_d} + \delta v)_\#\rho, p)^2.$$

Putting together, this gives that the r.h.s. of (A.4) is greater than or equal to

$$\frac{1}{2}\mathcal{W}_2((\mathrm{I_d} + \delta v)_\#\rho, p)^2 + O(\delta^2)$$

which implies (A.3). □

B. Proofs of Lemmas

Proof: [Proof of Lemma 5] It suffices to show that for any A s.t. $Leb(A) = 0$, $T_\# P(A) = 0$. By the definition of push-forward, $T_\# P(A) = P(T^{-1}(A))$, which is zero because $Leb(T^{-1}(A)) = 0$ (since T is non-degenerate) and $P \ll Leb$. □

Proof: [Proof of Lemma 6] First, the minimizer makes the r.h.s. finite because $T = \mathrm{I_d}$ makes it finite: When T is identity, the r.h.s. equals $G(p_n) < \infty$. As a result, $\tilde{p} := T_\# p_n$ needs to have density because otherwise the KL divergence $G(\tilde{p}) = +\infty$.

It remains to show that $M_2(\tilde{p}) < \infty$. By definition,

$$M_2(\tilde{p}) = \int_{\mathbb{R}^d} \|x\|^2 \tilde{p}(x) dx$$
$$= \mathbb{E}_{x \sim p_n} \|T(x)\|^2$$
$$\leq 2(\mathbb{E}_{x \sim p_n} \|x\|^2 + \mathbb{E}_{x \sim p_n} \|x - T(x)\|^2),$$

where $\mathbb{E}_{x \sim p_n} \|x\|^2 = M_2(p_n) < \infty$, and, at the minimizer T, $\mathbb{E}_{x \sim p_n} \|x - T(x)\|^2$ also needs to be finite due to that it is in the 2nd term of (25). □

Proof: [Proof of Lemma 7] We first show that $T \in L^2(p)$ iff $T_\# p \in \mathcal{P}_2$. This is because $M_2(T_\# p) = \mathbb{E}_{x \sim p} \|T(x)\|^2$, and thus is finite iff. T is in $L^2(p)$.

As a result, when $T \in L^2(p)$, $T_\# p \in \mathcal{P}_2$. If T is also non-degenerate, Lemma 5 implies that $T_\# p$ also has density. This proves that $T_\# p \in \mathcal{P}_2^r$. □

APPENDIX B
PROOFS AND LEMMAS IN SECTION IV

A. Technical Lemmas in Section IV-A

Lemma 17: $\mathcal{H}(\rho)$ is convex a.g.g. in $\mathcal{P}_2$.

Proof: The a.g.g.-convexity of functional in the form of $\mathcal{F}(\rho) = \int F(\rho(x)) dx$ in $\mathcal{P}_2$ is established in Proposition 9.3.9 of [7] when $F : [0, +\infty) \rightarrow (-\infty, \infty]$ is a proper, lower semi-continuous convex function satisfying that $s \mapsto s^d F(s^{-d})$ is convex and non-increasing on $(0, +\infty)$. The entropy $\mathcal{H}(\rho) = \mathcal{F}(\rho)$ with $F(s) = s \log s$, and this F satisfies the above conditions. □

Lemma 18: Under Assumption 4, $\mathcal{E}(\rho)$ is λ-convex a.g.g. in $\mathcal{P}_2$.

Proof: This is a direct result of Proposition 9.3.2(i) of [7], noting that assuming the boundedness of V^- implies the growth condition needed in Section 9.3 therein. The proof of Proposition 9.3.2(i) shows that $\mathcal{E}(\rho)$ is λ-convex along any interpolation curve which implies λ-convexity a.g.g. □

Lemma 19: Under Assumptions 4, the KL divergence $G(\rho)$ defined in (16) satisfies Assumption 3.

Proof: The lower semi-continuity follows from that of $\mathcal{H}(\rho)$ and the condition on V in Assumption 4. The domain of G is restricted to ρ with density because $\mathcal{H}(\rho)$ diverges otherwise. The a.g.g. λ-convexity of G directly follows from Lemma 17 and Lemma 18. □

B. Proofs in Section IV-B

Proof: [Proof of Lemma 8] The unique existences of T_p^ρ and T_p^π are by Brenier Theorem. Since $\rho \in \mathcal{P}_2^r$, the map T_p^ρ

has an inverse denoted by T_ρ^p which is defined ρ-a.e. Under Assumption 3 first part, the strong subdifferential of $\partial_{\mathcal{W}_2} G(\rho)$ is well-defined, and we assume η is one of them.

Let $v := T_p^\pi \circ T_\rho^p - \mathrm{I_d}$. One can verify that $v \in L^2(\rho)$, since $\|T_p^\pi \circ T_\rho^p\|_\rho^2 = M_2(\pi)$, $\|\mathrm{I_d}\|_\rho^2 = M_2(\rho)$, and both are finite. By definition, for $\delta \in [0, 1]$,

$$(\mathrm{I_d} + \delta v)_\# \rho = (\mathrm{I_d} + \delta(T_p^\pi \circ T_\rho^p - \mathrm{I_d}))_\# \rho$$
$$= (T_p^\rho + \delta(T_p^\pi - T_p^\rho))_\# p. \quad \text{(A.5)}$$

We also have

$$\langle \eta, v \rangle_\rho = \langle \eta \circ T_p^\rho, T_p^\pi - T_p^\rho \rangle_p. \quad \text{(A.6)}$$

Since $v \in L^2(\rho)$, by that $\eta \in \partial_{\mathcal{W}_2} G(\rho)$ and the definition of strong subdifferential (Definition 2), with $\delta \rightarrow 0+$ we have

$$G\left((\mathrm{I_d} + \delta v)_\# \rho\right) \geq G(\rho) + \delta \langle \eta, v \rangle_\rho + o(\delta). \quad \text{(A.7)}$$

Combined with (A.5)(A.6), this gives

$$G\left((T_p^\rho + \delta(T_p^\pi - T_p^\rho))_\# p\right) - G(\rho)$$
$$\geq \delta \langle \eta \circ T_p^\rho, T_p^\pi - T_p^\rho \rangle_p + o(\delta). \quad \text{(A.8)}$$

Meanwhile, by the λ-convexity of G a.g.g. (Definition 3), and specifically (14), we have

$$G\left((T_p^\rho + \delta(T_p^\pi - T_p^\rho))_\# p\right)$$
$$\leq (1 - \delta) G(\rho) + \delta G(\pi) - \frac{\lambda}{2} \delta(1 - \delta) \mathcal{W}_2(\rho, \pi)^2. \quad \text{(A.9)}$$

Comparing (A.8) and (A.9), we have

$$G(\pi) - G(\rho) \geq \langle \eta \circ T_p^\rho, T_p^\pi - T_p^\rho \rangle_p$$
$$+ \frac{\lambda}{2}(1 - \delta) \mathcal{W}_2(\rho, \pi)^2 + o(1).$$

We get the conclusion by letting $\delta \rightarrow 0+$. □

Proof: [Proof of Lemma 9] In the n-th step, T_{n+1} is in $L^2(p_n)$ and is non-degenerate under Assumption 1, thus from $p_n \in \mathcal{P}_2^r$, $p_{n+1} = (T_{n+1})_\# p_n$ is also in $\mathcal{P}_2^r$ by Lemma 7. This holds for $n = 0, \cdots, N - 1$, and thus all p_n are in $\mathcal{P}_2^r$, where $p_0 \in \mathcal{P}_2^r$ is by the lemma assumption. By the Brenier Theorem, the OT map from p_n to p_{n+1} is denoted as T_n^{n+1}, which is uniquely defined p_n-a.e. Let T_{n+1}^n be the OT map from p_{n+1} to p_n, and it is also the p_{n+1}-a.e. inverse of T_n^{n+1}. We use the short-hand notation

$$X_{n+1} := T_n^{n+1}.$$

Under the assumption on G, Lemma 16 applies which gives the relationship between $\partial_{\mathcal{W}_2} F_{n+1}$ and $\partial_{\mathcal{W}_2} G$. Together with the assumption on ξ_{n+1} by Assumption 1, we have that for each n, $\exists \eta_{n+1} \in \partial_{\mathcal{W}_2} G(p_{n+1})$ s.t.

$$\gamma \xi_{n+1} - \gamma \eta_{n+1} = \mathrm{I_d} - T_{n+1}^n, \quad p_{n+1}\text{-a.e.}$$

and equivalently,

$$\mathrm{I_d} - X_{n+1} = \gamma(\eta_{n+1} - \xi_{n+1}) \circ X_{n+1}. \quad p_n\text{-a.e.} \quad \text{(A.10)}$$

Denote by T_n^π the unique OT map from p_n to π. Expanding $\|X_{n+1} - T_n^\pi\|_{p_n}^2$ as

$$\|X_{n+1} - T_n^\pi\|_{p_n}^2$$

$$= \|(\mathrm{I_d} - X_{n+1}) - (\mathrm{I_d} - T_n^\pi)\|_{p_n}^2$$
$$= \|\mathrm{I_d} - T_n^\pi\|_{p_n}^2 - 2\langle \mathrm{I_d} - T_n^\pi, \mathrm{I_d} - X_{n+1}\rangle_{p_n}$$
$$+ \|\mathrm{I_d} - X_{n+1}\|_{p_n}^2$$
$$= \|\mathrm{I_d} - T_n^\pi\|_{p_n}^2 - 2\langle X_{n+1} - T_n^\pi, \mathrm{I_d} - X_{n+1}\rangle_{p_n}$$
$$- \|\mathrm{I_d} - X_{n+1}\|_{p_n}^2$$
$$\le \|\mathrm{I_d} - T_n^\pi\|_{p_n}^2 - 2\langle X_{n+1} - T_n^\pi, \mathrm{I_d} - X_{n+1}\rangle_{p_n},$$

where in the last inequality we use that $\|\mathrm{I_d} - X_{n+1}\|_{p_n}^2 \ge 0$. By that

$$\|\mathrm{I_d} - T_n^\pi\|_{p_n}^2 = \mathcal{W}_2(p_n, \pi)^2,$$

and together with (A.10), we have

$$\|X_{n+1} - T_n^\pi\|_{p_n}^2 \le \mathcal{W}_2(p_n, \pi)^2$$
$$- 2\gamma\langle X_{n+1} - T_n^\pi, (\eta_{n+1} - \xi_{n+1}) \circ X_{n+1}\rangle_{p_n}. \quad (A.11)$$

Applying Lemma 8 with $p = p_n$ and $\rho = p_{n+1}$, we have

$$G(\pi) - G(p_{n+1})$$
$$\ge \langle T_n^\pi - X_{n+1}, \eta_{n+1} \circ X_{n+1}\rangle_{p_n} + \frac{\lambda}{2}\mathcal{W}_2(p_{n+1}, \pi)^2. \quad (A.12)$$

Meanwhile, by Cauchy Schwartz,

$$|\langle X_{n+1} - T_n^\pi, \xi_{n+1} \circ X_{n+1}\rangle_{p_n}|$$
$$\le \|X_{n+1} - T_n^\pi\|_{p_n}\|\xi_{n+1} \circ X_{n+1}\|_{p_n}$$
$$\le \varepsilon\|X_{n+1} - T_n^\pi\|_{p_n}$$

where the 2nd inequality is by that $\|\xi_{n+1} \circ X_{n+1}\|_{p_n} = \|\xi_{n+1}\|_{p_{n+1}} \le \varepsilon$ (Assumption 1). Since $\lambda > 0$, we have

$$\varepsilon\|X_{n+1} - T_n^\pi\|_{p_n} \le \frac{\varepsilon^2}{\lambda} + \frac{\lambda}{4}\|X_{n+1} - T_n^\pi\|_{p_n}^2. \quad (A.13)$$

Putting together, this gives

$$|\langle X_{n+1} - T_n^\pi, \xi_{n+1} \circ X_{n+1}\rangle_{p_n}| \le \frac{\varepsilon^2}{\lambda} + \frac{\lambda}{4}\|X_{n+1} - T_n^\pi\|_{p_n}^2. \quad (A.14)$$

Inserting (A.12)(A.14) into (A.11) gives

$$(1 - \frac{\gamma\lambda}{2})\|X_{n+1} - T_n^\pi\|_{p_n}^2$$
$$\le \mathcal{W}_2(p_n, \pi)^2 + 2\gamma\left(G(\pi) - G(p_{n+1}) - \frac{\lambda}{2}\mathcal{W}_2(p_{n+1}, \pi)^2\right)$$
$$+ \frac{2\gamma}{\lambda}\varepsilon^2. \quad (A.15)$$

Because $(X_{n+1}, T_n^\pi)_{\#}p_n$ is a coupling between p_{n+1} and π, we have

$$\mathcal{W}_2(p_{n+1}, \pi)^2 \le \|X_{n+1} - T_n^\pi\|_{p_n}^2. \quad (A.16)$$

Under the condition of the lemma, $0 < \gamma\lambda < 2$, and thus $1 - \frac{\gamma\lambda}{2} > 0$ and then the l.h.s. of (A.15) $\ge (1 - \frac{\gamma\lambda}{2})\mathcal{W}_2(p_{n+1}, \pi)^2$. This proves (34). $\square$

Proof: [Proof of Theorem 10] Taking $\pi = q$ and apply Lemma 9, by that $2\gamma(G(p_{n+1}) - G(\pi)) \ge 0$, (34) gives that for all n,

$$\left(1 + \frac{\gamma\lambda}{2}\right)\mathcal{W}_2^2(p_{n+1}, q) \le \mathcal{W}_2^2(p_n, q) + \frac{2\gamma}{\lambda}\varepsilon^2. \quad (A.17)$$

Define the numbers ρ and α as

$$\rho := \left(1 + \frac{\gamma\lambda}{2}\right)^{-1}, \quad 0 < \rho < 1, \quad \alpha := \sqrt{\frac{2\gamma}{\lambda}}\varepsilon,$$

and define

$$E_n := \mathcal{W}_2(p_n, q)^2,$$

then (A.17) can be written as

$$E_{n+1} \le \rho(E_n + \alpha).$$

Recursively applying from 0 to $n - 1$ gives that

$$E_n \le \rho^n E_0 + \alpha\frac{\rho(1 - \rho^n)}{1 - \rho} \le \rho^n E_0 + \alpha\frac{\rho}{1 - \rho},$$

which by definition is equivalent to (35).

By (35), one will have $\mathcal{W}_2(p_n, q)^2 \le 5\varepsilon^2/\lambda^2$ if

$$\left(1 + \frac{\gamma\lambda}{2}\right)^{-n}\mathcal{W}_2^2(p_0, q) \le \frac{\varepsilon^2}{\lambda^2},$$

which is fulfilled as long as

$$n \ge \frac{2\left(\log\mathcal{W}_2(p_0, q) + \log(\lambda/\varepsilon)\right)}{\log(1 + \gamma\lambda/2)}.$$

This requirement of n is satisfied under (36) by that $0 < \gamma\lambda < 2$ and the elementary relation that $\log(1 + x) \ge x/2$ for $x \in (0, 1)$. We have proved the $\mathcal{W}_2$-error bound.

To show the smallness of the objective gap $G(p_n) - G(q)$, we use (34) again, and by that $\mathcal{W}_2^2(p_{n+1}, q) \ge 0$,

$$2\gamma(G(p_{n+1}) - G(q)) \le \mathcal{W}_2^2(p_n, q) + \frac{2\gamma}{\lambda}\varepsilon^2. \quad (A.18)$$

When n already makes $\mathcal{W}_2(p_n, q)^2 \le 5\varepsilon^2/\lambda^2$, we have

$$2\gamma(G(p_{n+1}) - G(q)) \le (5 + 2\gamma\lambda)\frac{\varepsilon^2}{\lambda^2} \le 9\frac{\varepsilon^2}{\lambda^2}, \quad (A.19)$$

where in the 2nd inequality we use that $\gamma\lambda < 2$ because $0 < \lambda \le 1$ and $0 < \gamma < 2$. This proves the bound of $G(p_n) - G(q)$ in (37). $\square$

APPENDIX C
PROOFS IN SECTION V

A. Proofs in Section V-A.1

Proof: [Proof of Lemma 11] Let $X_1 \sim p$, $X_2 \sim q$, and

$$Y_1 = T(X_1), \quad Y_2 = T(X_2).$$

Then Y_1 and Y_2 also have densities, $Y_1 \sim \tilde{p} := T_{\#}p$ and $Y_2 \sim \tilde{q} := T_{\#}q$. By the data processing inequality concerning two probability distributions through the same stochastic transformation for the KL divergence (see, e.g., the introduction of [61]),

$$\mathrm{KL}(\tilde{p}||\tilde{q}) \le \mathrm{KL}(p||q).$$

In the other direction, $X_i = T^{-1}(Y_i)$, $i = 1, 2$, then data processing inequality also implies

$$\mathrm{KL}(p||q) \le \mathrm{KL}(\tilde{p}||\tilde{q}).$$

$\square$

Proof: [Proof of Corollary 12] Under Assumption 4, the KL divergence $G(\rho)$ satisfies Assumption 3 (Lemma 19), also $q \in \mathcal{P}_2^r$ and $G(q) = 0$ is the global minimum of G. The needed assumptions of Theorem 10 are all satisfied, by which we have that for the N defined in the corollary,

$$\text{KL}(p_N\|q) = G(p_N) \le \frac{9}{2\gamma}\left(\frac{\varepsilon}{\lambda}\right)^2. \qquad (\text{A.20})$$

Under Assumption 1, T_n are all invertible, and thus T_1^N as defined in (38) is invertible. In addition, by Definition 4, one can verify that if T_1 and T_2 are non-degenerate, then so is $T_2 \circ T_1$. Using the arguments for $N - 1$ times, we have that T_1^N is non-degenerate. Similarly, since each T_n^{-1} is non-degenerate, we have that $(T_1^N)^{-1}$ is also non-degenerate. Now, $p_0 = p$ has density, and then $p_N = (T_1^N)_\# p_0$ also has density (Lemma 5). Meanwhile, $q_N = q$ has density (Assumption 4), then $q_0 = ((T_1^N)^{-1})_\# q_N$ also has density. We now have that $p_0, q_0, p_N = (T_1^N)_\# p_0$ and $q_N = (T_1^N)_\# q_0$ all have densities. Then Lemma 11 gives that

$$\text{KL}(p_0\|q_0) = \text{KL}(p_N\|q_N) = \text{KL}(p_N\|q),$$

which, by (A.20), is bounded as stated in the corollary. The TV bound is followed by Pinsker's inequality. $\qquad \square$

B. Proofs in Section V-A.2

Lemma 20 (ρ_δ and $\mathcal{W}_2$ Closeness): Suppose $P \in \mathcal{P}_2$, and ρ_t is the density of X_t in an OU process as in (6), then
    (i) $\rho_t \in \mathcal{P}_2^r$ for any $t > 0$,
    (ii) $\forall \varepsilon > 0$, $\exists \delta > 0$ s.t. $\mathcal{W}_2(\rho_\delta, P) < \varepsilon$. In this case, one can choose $\delta \sim \varepsilon^2$.

Proof: [Proof of Lemma 20] For the OU process, we have $V(x) = \|x\|^2/2$ in (7). Then for any $t > 0$, $\rho_t = \mathcal{L}_t(P)$ has the expression as, with $\sigma_t^2 := 1 - e^{-2t}$,

$$\rho_t(x) = \int_{\mathbb{R}^d} \frac{1}{(2\pi\sigma_t^2)^{d/2}} e^{-\|x - e^{-t}y\|^2/(2\sigma_t^2)} dP(y). \quad (\text{A.21})$$

Equivalently, ρ_t is the probability density of the random vector

$$Z_t := e^{-t}X_0 + \sigma_t Z, \quad Z \sim \mathcal{N}(0, I_d),$$

where Z is independent from X_0. Since $\mathbb{E}\|Z_t\|^2 = e^{-2t}M_2(P) + \sigma_t^2 d < \infty$, we have $\rho_t \in \mathcal{P}_2$ and this proves (i).

To prove (ii): Because the law of (Z_t, X_0) is a coupling of ρ_t and P,

$$\begin{aligned}
\mathcal{W}_2(\rho_t, P)^2 &\le \mathbb{E}\|Z_t - X_0\|^2 \\
&= \mathbb{E}\|(e^{-t} - 1)X_0 + \sigma_t Z\|^2 \\
&= (1 - e^{-t})^2 M_2(P) + (1 - e^{-2t})d \\
&\le t^2 M_2(P) + 2td,
\end{aligned}$$

where in the last inequality we used that $1 - e^{-x} \le x, \forall x \ge 0$. Since $M_2(P) < \infty$, we have bounded $\mathcal{W}_2(\rho_t, P)^2$ to be $O(t)$. $\qquad \square$

Proof: [Proof of Corollary 13] For the ε in Assumption 1, the existence of δ to make $\mathcal{W}_2(\rho_\delta, P) < \varepsilon$ is by Lemma 20, and we also have $\rho_\delta \in \mathcal{P}_2^r$. The rest of the proof is the same as in Corollary 12 by starting from $p_0 = \rho_\delta$. $\qquad \square$

C. Proofs and Lemmas in Section V-B

Lemma 21 (Lipschitz Bound of ODE Solution Map): Suppose for $\gamma > 0$, $\hat{v}(x, t)$ is C^1 in (x, t) and Lipschitz in x uniformly on $\mathbb{R}^d \times [0, \gamma]$ with Lipschitz constant $K \ge 0$. Let $x(t)$ be the solution to the ODE

$$\dot{x}(t) = \hat{v}(x(t), t), \quad t \in [0, \gamma], \qquad (\text{A.22})$$

and define the solution map from 0 to γ as $T : \mathbb{R}^d \to \mathbb{R}^d$, that is,

$$T(x_0) = x_0 + \int_0^\gamma \hat{v}(x(t), t)dt, \quad x(0) = x_0. \quad (\text{A.23})$$

Then T is invertible on $\mathbb{R}^d$, and both T and T^{-1} are Lipschitz on $\mathbb{R}^d$ with Lipschitz constant $e^{\gamma K}$.

Proof: [Proof of Lemma 21] Let $x_1(t)$ and $x_2(t)$ be the solution to the ODE (A.22) from $x_1(0) = y$, and $x_2(0) = z$ respectively. By definition,

$$T(y) = x_1(\gamma), \quad T(z) = x_2(\gamma).$$

Under the condition of $\hat{v}$, the ODE is well-posed [64]. This implies the invertibility of T, and T^{-1} is the solution map of the reverse time ODE from $t = \gamma$ to $t = 0$.

We now prove the Lipschitz constant of T on $\mathbb{R}^d$, and that of T^{-1} can be proved similarly by considering the reverse time ODE. We want to show that

$$\|T(y) - T(z)\| \le e^{\gamma K}\|y - z\|, \quad \forall y, z \in \mathbb{R}^d,$$

and this is equivalent to that for any $x_1(0), x_2(0) \in \mathbb{R}^d$,

$$\|x_1(\gamma) - x_2(\gamma)\| \le e^{\gamma K}\|x_1(0) - x_2(0)\|. \quad (\text{A.24})$$

For fixed $x_1(0), x_2(0)$, define

$$E(t) := \frac{1}{2}\|x_1(t) - x_2(t)\|^2,$$

then $E(0) = \|x_1(0) - x_2(0)\|^2/2$, and

$$\begin{aligned}
\dot{E}(t) &= (x_1(t) - x_2(t))^T(\dot{x}_1(t) - \dot{x}_2(t)) \\
&= (x_1(t) - x_2(t))^T(\hat{v}(x_1(t), t) - \hat{v}(x_2(t), t)).
\end{aligned}$$

Thus, by that $\|\hat{v}(x_1(t), t) - \hat{v}(x_2(t), t)\| \le K\|x_1(t) - x_2(t)\|$, we have

$$\dot{E}(t) \le K\|x_1(t) - x_2(t)\|^2 = 2KE(t).$$

By Grönwall's inequality, $E(t) \le E(0)e^{2Kt}$, and this gives

$$\|x_1(t) - x_2(t)\|^2 \le e^{2Kt}\|x_1(0) - x_2(0)\|^2, \quad t \in [0, \gamma].$$

Setting $t = \gamma$ proves (A.24). $\qquad \square$

Lemma 22: Suppose $p \in \mathcal{P}_2^r$, and $T : \mathbb{R}^d \to \mathbb{R}^d$ is Lipschitz on $\mathbb{R}^d$, then $T \in L^2(p)$.

Proof: We are to show that

$$\mathbb{E}_{x \sim p}\|T(x)\|^2 < \infty.$$

Suppose T is L-Lipschitz on $\mathbb{R}^d$, then $\forall x \in \mathbb{R}^d$,

$$\|T(x)\| \le \|T(0)\| + \|T(x) - T(0)\| \le \|T(0)\| + L\|x\|.$$

Thus,

$$\mathbb{E}_{x \sim p}\|T(x)\|^2 \le 2(\|T(0)\|^2 + L^2\mathbb{E}_{x \sim p}\|x\|^2)$$

$$= 2(\|T(0)\|^2 + L^2 M_2(p)) < \infty,$$

because $M_2(p) < \infty$. $\square$

Proof: [Proof of Proposition 14] By construction, for $n = 1, \cdots, N$,

$$q_{n-1} = (T_n^{-1})_\# q_n, \quad \tilde{q}_{n-1} = (S_n)_\# \tilde{q}_n.$$

We also have that $q_N = \tilde{q}_N = q \in \mathcal{P}_2^r$ by assumption.

For the sequence of q_n, we know that T_n^{-1} is non-degenerate (Assumption 1). Meanwhile, for $q_n \in \mathcal{P}_2^r$, T_n^{-1} being globally Lipschitz on $\mathbb{R}^d$ also implies that it is in $L^2(q_n)$ (Lemma 22). Then by Lemma 7, $q_{n-1} = (T_n^{-1})_\# q_n$ is also in $\mathcal{P}_2^r$. For the sequence of $\tilde{q}_n$, by Assumption 2, S_n is non-degenerate and in $L^2(\tilde{q}_n)$, thus from $\tilde{q}_n \in \mathcal{P}_2^r$, $\tilde{q}_{n-1} = (S_n)_\# \tilde{q}_n$ is also in $\mathcal{P}_2^r$ by Lemma 7 again. Thus by induction, we have that q_n and $\tilde{q}_n$ are all in $\mathcal{P}_2^r$.

For each n, we have

$$\begin{aligned} \mathcal{W}_2(\tilde{q}_{n-1}, q_{n-1}) &= \mathcal{W}_2((S_n)_\# \tilde{q}_n, (T_n^{-1})_\# q_n) \\ &\leq \underbrace{\mathcal{W}_2((S_n)_\# \tilde{q}_n, (T_n^{-1})_\# \tilde{q}_n)}_{①} \\ &\quad + \underbrace{\mathcal{W}_2((T_n^{-1})_\# \tilde{q}_n, (T_n^{-1})_\# q_n)}_{②}. \end{aligned}$$

To bound ①, we use Assumption 2 and the Lipschitzness of T_n^{-1}. Define $L := e^{\gamma K}$. Using $(S_n, T_n^{-1})_\# \tilde{q}_n$ as the coupling, we have

$$\begin{aligned} &\mathcal{W}_2^2((S_n)_\# \tilde{q}_n, (T_n^{-1})_\# \tilde{q}_n) \\ &\leq \int_{\mathbb{R}^d} \|S_n(x) - T_n^{-1}(x)\|^2 \tilde{q}_n(x) dx \\ &\leq \int_{\mathbb{R}^d} L^2 \|T_n \circ S_n(x) - x\|^2 \tilde{q}_n(x) dx \\ &= L^2 \|T_n \circ S_n - \mathrm{I_d}\|_{\tilde{q}_n}^2, \end{aligned}$$

where for the second inequality, we use the fact that T_n^{-1} is L-Lipschitz. Thus,

$$① \leq L \varepsilon_{\mathrm{inv}}. \tag{A.25}$$

To bound ②, we use that T_n^{-1} is L-Lipschitz on $\mathbb{R}^d$ again. Specifically, let Y_n be the unique OT map from q_n to $\tilde{q}_n$ which is well-defined by the Brenier Theorem, then $(T_n^{-1} \circ Y_n, T_n^{-1})_\# q_n$ is a coupling of $(T_n^{-1})_\# \tilde{q}_n$ and $(T_n^{-1})_\# q_n$. We have that

$$\begin{aligned} &\mathcal{W}_2^2((T_n^{-1})_\# \tilde{q}_n, (T_n^{-1})_\# q_n) \\ &\leq \int_{\mathbb{R}^d} \|T_n^{-1} \circ Y_n(x) - T_n^{-1}(x)\|^2 q_n(x) dx \\ &\leq \int_{\mathbb{R}^d} L^2 \|Y_n(x) - x\|^2 q_n(x) dx \\ &= L^2 \mathcal{W}_2^2(\tilde{q}_n, q_n). \end{aligned}$$

Thus

$$② \leq L \mathcal{W}_2(\tilde{q}_n, q_n). \tag{A.26}$$

Putting together, we have

$$\mathcal{W}_2(\tilde{q}_{n-1}, q_{n-1}) \leq e^{\gamma K}(\varepsilon_{\mathrm{inv}} + \mathcal{W}_2(\tilde{q}_n, q_n)).$$
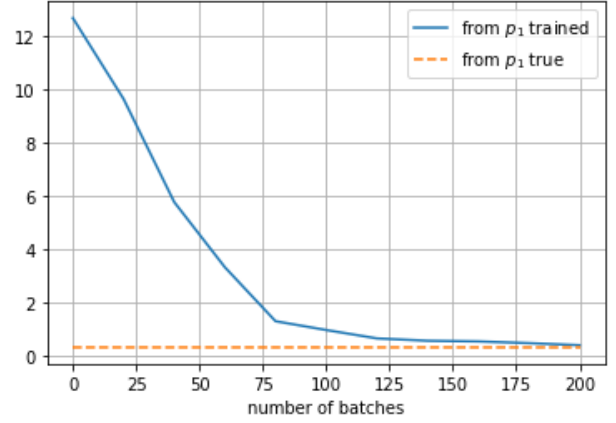

Fig. 3. Computed values of $\|\nabla_{\mathcal{W}_2} F_{n+1}(p_{n+1})\|_{p_{n+1}}^2$ from $N = 2000$ samples, where p_{n+1} is pushforwarded by a trained neural network transport T from a Gaussian initial p_n in $\mathbb{R}^2$, $n = 0$. The blue line shows the value as the training progresses, and the dashed line is a base value computed from the analytical solution p_{n+1}^{true} (where the Wasserstein gradient vanishes).

Note that $\mathcal{W}_2(\tilde{q}_N, q_N) = 0$ by that $\tilde{q}_N = q_N = q$. Applying recursively from $n = N$ to $n = 1$ gives that

$$\mathcal{W}_2(\tilde{q}_0, q_0) \leq \varepsilon_{\mathrm{inv}} \frac{e^{\gamma K}(e^{\gamma KN} - 1)}{e^{\gamma K} - 1}, \tag{A.27}$$

which proves (41) by that $e^x - 1 \geq x$ for any $x \in \mathbb{R}$. $\square$

Proof: [Proof of Corollary 15] Under the condition of the corollary, Corollary 12 applies to bound $\mathrm{KL}(p\|q_0)$ and $TV(p, q_0)$ as in (40), and Proposition 14 applies to bound $\mathcal{W}_2(\tilde{q}_0, q_0)$ as in (41). It suffices to show that the r.h.s. of (41) is less than or equal to that of (42).

By the choice of N in (39),

$$N \leq \frac{8}{\gamma \lambda} \left(\log \mathcal{W}_2(p_0, q) + \log(\lambda/\varepsilon) \right) + 1,$$

and thus

$$e^{\gamma K(N+1)} \leq e^{2\gamma K} \left(\mathcal{W}_2(p_0, q) \frac{\lambda}{\varepsilon} \right)^{8K/\lambda},$$

which proves the needed inequality. $\square$

APPENDIX D
NUMERICAL EVIDENCE TO SUPPORT ASSUMPTION 1

We conduct training of one JKO block to verify (30) in Assumption 1. The code is available at https://github.com/yixintan-zeta/jko_wass_grad.

Data is in $\mathbb{R}^2$. Let $n = 0$, $p_0 = \mathcal{N}([3, 3]^T, I)$, $q = \mathcal{N}(0, I)$, and $G(\rho) = \mathrm{KL}(\rho\|q)$. The step size $\gamma = 0.5$. We train a JKO flow network block to minimize F_{n+1} in (24) via the training objective (25), where T is parametrized by a neural ODE block consisting of two hidden layers with 128 hidden dimensions and using the softplus activation ($\beta = 20$). We use 10,000 training samples, batch size 2000, and 200 total iterations, with learning rate 10^{-4}. The implementation of the JKO flow network follows the setup in [74].
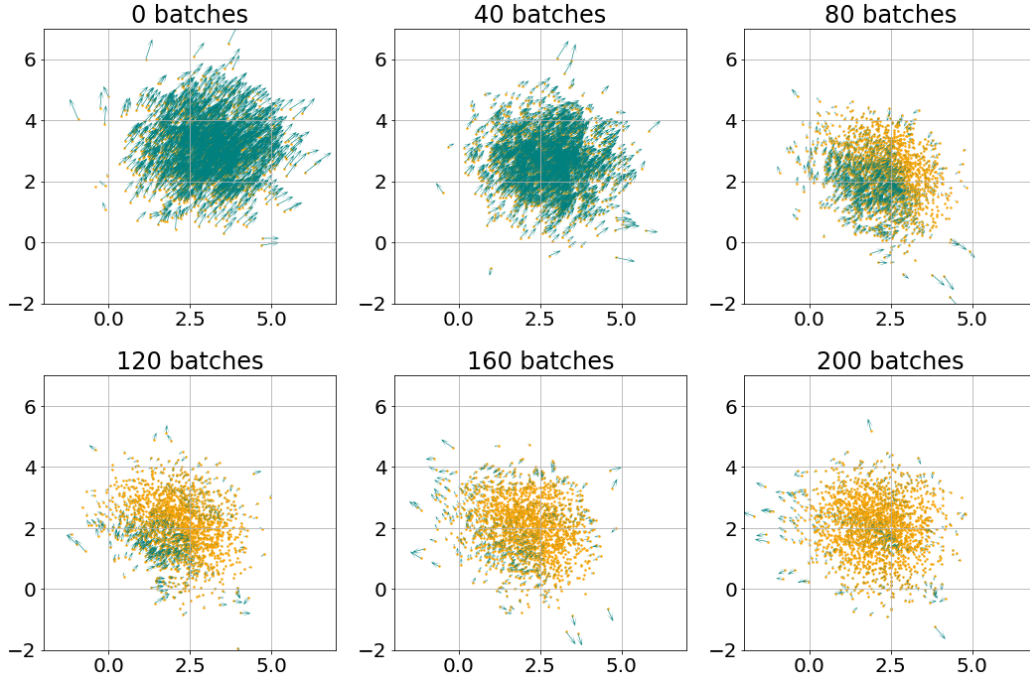
Fig. 4.   The Wasserstein gradient vector field ξ at samples $x_i^{(1)} = T(x_i)$ (shown by green arrows), where T is the trained neural network transport map, plotted as the training progresses. The yellow dots are samples $x_i^{(1)}$. The length of the arrow is proportional to the magnitude of $\|\xi(x_i^{(1)})\|$.

Making use of the explicit expression of $\nabla_{\mathcal{W}_2} G(\rho)$, let ξ be ξ_1, then (27) gives that

$$\xi = (\nabla V + \nabla \log p_1) - \frac{T_{p_1}^{p_0} - \mathrm{I_d}}{\gamma},$$

where $V(x) = \|x\|^2/2$. To compute ξ as a vector field, we used $N = 2000$ samples $x_i \sim p_0$, and then the neural-network trained T will push-forward x_i to $T(x_i) \sim p_1$. Let $X_0 = \{x_i^{(0)} := x_i\}_{i=1}^N$ and $X_1 = \{x_i^{(1)} := T(x_i)\}_{i=1}^N$ be two data clouds. We approximate the OT map $T_{p_1}^{p_0}$ evaluated on $x_i^{(i)}$ by the solution of a discrete OT problem, computed by the Python POT package [1]. The score function $\nabla \log p_1$ is approximately computed by a kernel approach, where we used a Gaussian kernel with a properly chosen bandwidth parameter. Once $\xi(x_i^{(1)})$ is computed at every sample, we can approximately compute the squared L^2 norm $\|\xi\|_{p_1}^2$ by a sample average.

We compute ξ at p_1 induced by trained T not only at the end of training but also during intermediate iterations. This will show the change in the Wasserstein gradient as the neural network training progresses. At 0, 20, ..., 200 batches, the estimated L^2 norm is shown in Figure 3. The dash-line shows the computed value of the Wasserstein gradient at the true solution $p_1^{\mathrm{true}} = \mathcal{N}([2,2]^T, I)$, since the JKO step is from a Gaussian p_0 and thus the true population minimizer p_1 is analytically available. The numerical value is not exactly zero because it is also computed on $N = 2000$ finite samples. The dash-line shows a baseline of the numerical L^2 norm, and it can be seen that at the end of training, the neural network learned p_1 archives a comparable value.

We further illustrate the vector field ξ on $x_i^{(1)}$'s in Figure 4, which shows the evolution of the Wasserstein gradient over

training iterations. It can be seen that the magnitude of the vector field decreases, and the values are getting small at least within the region where the distribution density p_1 has a large value. In the outskirts, the vector field ξ does not numerically get small because it is at the tail of the Gaussian distribution, where a small L^2 norm does not imply pointwise smallness of ξ in these regions and the estimation of the vector field may also be of lower accuracy.

REFERENCES

[1] *Python POT Package, Version (0.9.4)*. Accessed: Jun. 2024. [Online]. Available: https://pythonot.github.io/

[2] M. S. Albergo, N. M. Boffi, and E. Vanden-Eijnden, "Stochastic interpolants: A unifying framework for flows and diffusions," 2023, *arXiv:2303.08797*.

[3] M. S. Albergo and E. Vanden-Eijnden, "Building normalizing flows with stochastic interpolants," in *Proc. 11th Int. Conf. Learn. Represent.*, 2023, pp. 1–29.

[4] D. Alvarez-Melis, Y. Schiff, and Y. Mroueh, "Optimizing functionals on the space of probabilities with input convex neural networks," *Trans. Mach. Learn. Res.*, 2022.

[5] S.-I. Amari, "Information geometry and its applications: Convex function and dually flat manifold," in *LIX Fall Colloquium on Emerging Trends in Visual Computing*. Berlin, Germany: Springer, 2008, pp. 75–102.

[6] S.-I. Amari, *Information Geometry and Its Applications*, vol. 194. Berlin, Germany: Springer, 2016.

[7] L. Ambrosio, N. Gigli, and G. Savaré, *Gradient Flows: In Metric Spaces and in the Space of Probability Measures*. Berlin, Germany: Springer, 2005.

[8] B. Amos, L. Xu, and J. Z. Kolter, "Input convex neural networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 146–155.

[9] J. Behrmann, W. Grathwohl, R. T. Chen, D. Duvenaud, and J.-H. Jacobsen, "Invertible residual networks," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2019, pp. 573–582.

[10] J. Benton, G. Deligiannidis, and A. Doucet, "Error bounds for flow matching methods," *Trans. Mach. Learn. Res.*, 2024.

[11] J. Benton, V. De Bortoli, A. Doucet, and G. Deligiannidis, "Nearly d-linear convergence bounds for diffusion models via stochastic localization," in *Proc. 12th Int. Conf. Learn. Represent.*, 2024, pp. 1–21.

[12] E. Bernton, "Langevin Monte Carlo and JKO splitting," in *Proc. 31st Conf. Learn. Theory*, vol. 75, 2018, pp. 1777–1798.

[13] F. Bolley, I. Gentil, and A. Guillin, "Convergence to equilibrium in Wasserstein distance for Fokker–Planck equations," *J. Funct. Anal.*, vol. 263, no. 8, pp. 2430–2457, Oct. 2012.

[14] H. Chen, H. Lee, and J. Lu, "Improved analysis of score-based generative modeling: User-friendly bounds under minimal smoothness assumptions," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 4735–4763.

[15] R. T. Chen, Y. Rubanova, J. Bettencourt, and D. K. Duvenaud, "Neural ordinary differential equations," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–13.

[16] S. Chen, S. Chewi, H. Lee, Y. Li, J. Lu, and A. Salim, "The probability flow ode is provably fast," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 36, 2024, pp. 1–24.

[17] S. Chen, S. Chewi, J. Li, Y. Li, A. Salim, and A. Zhang, "Sampling is as easy as learning the score: Theory for diffusion models with minimal data assumptions," in *Proc. 11th Int. Conf. Learn. Represent.*, 2022, pp. 1–29.

[18] S. Chen, G. Daras, and A. Dimakis, "Restoration-degradation beyond linear diffusions: A non-asymptotic analysis for ddim-type samplers," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 4462–4484.

[19] X. Cheng and A. Cloninger, "Classification logit two-sample testing by neural networks for differentiating near manifold densities," *IEEE Trans. Inf. Theory*, vol. 68, no. 10, pp. 6631–6662, Oct. 2022.

[20] V. De Bortoli, "Convergence of denoising diffusion models under the manifold hypothesis," *Trans. Mach. Learn. Res.*, 2022.

[21] P. Degond and S. Mas-Gallic, "The weighted particle method for convection-diffusion equations. I. The case of an isotropic viscosity," *Math. Comput.*, vol. 53, no. 188, pp. 485–507, 1989.

[22] P. Degond and F.-J. Mustieles, "A deterministic approximation of diffusion equations using particles," *SIAM J. Scientific Stat. Comput.*, vol. 11, no. 2, pp. 293–310, Mar. 1990.

[23] M. Z. Diao, K. Balasubramanian, S. Chewi, and A. Salim, "Forward-backward Gaussian variational inference via JKO in the bures-Wasserstein space," in *Proc. Int. Conf. Mach. Learn.*, 2023, pp. 7960–7991.

[24] L. Dinh, D. Krueger, and Y. Bengio, "NICE: Non-linear independent components estimation," 2014, *arXiv:1410.8516*.

[25] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real NVP," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–32.

[26] J. Fan, Q. Zhang, A. Taghvaei, and Y. Chen, "Variational Wasserstein gradient flow," in *Proc. Int. Conf. Mach. Learn.*, 2022, pp. 6185–6215.

[27] C. Finlay, J. Jacobsen, L. Nurbekyan, and A. M. Oberman, "How to train your neural ODE: The world of Jacobian and kinetic regularization," in *Proc. 37th Int. Conf. Mach. Learn.*, 2020, pp. 3154–3164.

[28] I. J. Goodfellow et al., "Generative adversarial nets," in *Proc. NIPS*, 2014, pp. 1–9.

[29] W. Grathwohl, R. T. Chen, J. Bettencourt, I. Sutskever, and D. Duvenaud, "FFJORD: Free-form continuous dynamics for scalable reversible generative models," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.

[30] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. NIPS*, 2017, pp. 1–11.

[31] P. Harremoes and I. Vajda, "On pairs of f-divergences and their joint range," *IEEE Trans. Inf. Theory*, vol. 57, no. 6, pp. 3230–3235, Jun. 2011.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[33] J. Ho, A. Jain, and P. Abbeel, "Denoising diffusion probabilistic models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 6840–6851.

[34] H. Huang, J. Yu, J. Chen, and R. Lai, "Bridging mean-field games and normalizing flows with trajectory regularization," *J. Comput. Phys.*, vol. 487, Aug. 2023, Art. no. 112155.

[35] J. Huang, Y. Jiao, Z. Li, L. Shiao, W. Yang, and Y. Yang, "An error analysis of generative adversarial networks for learning distributions," *J. Mach. Learn. Res.*, vol. 23, no. 1, pp. 5047–5089, 2022.

[36] A. Hyvärinen and P. Dayan, "Estimation of non-normalized statistical models by score matching," *J. Mach. Learn. Res.*, vol. 6, no. 4, pp. 1–15, 2005.

[37] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5967–5976.

[38] R. Johnson and T. Zhang, "A framework of composite functional gradient methods for generative adversarial models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 17–32, Jan. 2021.

[39] R. Jordan, D. Kinderlehrer, and F. Otto, "The variational formulation of the fokker-Planck equation," *SIAM J. Math. Anal.*, vol. 29, no. 1, pp. 1–17, Jan. 1998.

[40] C. Kent, J. Li, J. Blanchet, and P. W. Glynn, "Modified Frank Wolfe in probability space," in *Proc. 35th Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 14448–14462.

[41] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*.

[42] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.

[43] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1×1 convolutions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–10.

[44] I. Kobyzev, S. J. D. Prince, and M. A. Brubaker, "Normalizing flows: An introduction and review of current methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 3964–3979, Nov. 2021.

[45] M. Lambert, S. Chewi, F. Bach, S. Bonnabel, and P. Rigollet, "Variational inference via Wasserstein gradient flows," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 14434–14447.

[46] H. Lee, R. Ge, T. Ma, A. Risteski, and S. Arora, "On the ability of neural nets to express distributions," in *Proc. Conf. Learn. Theory*, 2017, pp. 1271–1296.

[47] H. Lee, J. Lu, and Y. Tan, "Convergence for score-based generative modeling with polynomial complexity," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 35, 2022, pp. 22870–22882.

[48] H. Lee, J. Lu, and Y. Tan, "Convergence of score-based generative modeling for general data distributions," in *Proc. Int. Conf. Algorithmic Learn. Theory*, 2023, pp. 946–985.

[49] G. Li, Y. Wei, Y. Chen, and Y. Chi, "Towards non-asymptotic convergence for diffusion-based generative models," in *Proc. 12th Int. Conf. Learn. Represent.*, 2024, pp. 1–12.

[50] Y. Lipman, R. T. Q. Chen, H. Ben-Hamu, M. Nickel, and M. Le, "Flow matching for generative modeling," in *Proc. 11th Int. Conf. Learn. Represent.*, 2023, pp. 1–28.

[51] Q. Liu, "Rectified flow: A marginal preserving approach to optimal transport," 2022, *arXiv:2209.14577*.

[52] Y. Lu and J. Lu, "A universal approximation theorem of deep neural networks for expressing probability distributions," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 3094–3105.

[53] Y. Marzouk, Z. Ren, S. Wang, and J. Zech, "Distribution learning via neural differential equations: A nonparametric statistical perspective," *J. Mach. Learn. Res.*, 2024.

[54] P. Mokrov, A. Korotin, L. Li, A. Genevay, J. M. Solomon, and E. Burnaev, "Large-scale Wasserstein gradient flows," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 15243–15256.

[55] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro, "Robust stochastic approximation approach to stochastic programming," *SIAM J. Optim.*, vol. 19, no. 4, pp. 1574–1609, Jan. 2009.

[56] A. Nemirovsky and D. Yudin, *Problem Complexity and Method Efficiency in Optimization*. Chichester, U.K.: Wiley, 1983.

[57] S. Nowozin, B. Cseke, and R. Tomioka, "f-GAN: Training generative neural samplers using variational divergence minimization," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 1–9.

[58] D. Onken, S. W. Fung, X. Li, and L. Ruthotto, "OT-flow: Fast and accurate continuous normalizing flows via optimal transport," in *Proc. AAAI Conf. Artif. Intell.*, vol. 35, 2021, pp. 9223–9232.

[59] F. Pedrotti, J. Maas, and M. Mondelli, "Improved convergence of score-based diffusion models via prediction-correction," *Trans. Mach. Learn. Res.*, 2024.

[60] D. Perekrestenko, L. Eberhard, and H. Bölcskei, "High-dimensional distribution generation through deep neural networks," *Partial Differ. Equ. Appl.*, vol. 2, no. 5, p. 64, Oct. 2021.

[61] M. Raginsky, "Strong data processing inequalities and Φ-Sobolev inequalities for discrete channels," *IEEE Trans. Inf. Theory*, vol. 62, no. 6, pp. 3355–3389, Jun. 2016.

[62] L. Ruthotto, S. J. Osher, W. Li, L. Nurbekyan, and S. W. Fung, "A machine learning framework for solving high-dimensional mean field game and mean field control problems," *Proc. Nat. Acad. Sci. USA*, vol. 117, no. 17, pp. 9183–9193, Apr. 2020.

[63] A. Salim, A. Korba, and G. Luise, "The Wasserstein proximal gradient algorithm," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 12356–12366.

[64] T. C. Sideris, *Ordinary Differential Equations and Dynamical Systems*. Paris, France: Atlantis Press, 2013.

[65] Y. Song, C. Durkan, I. Murray, and S. Ermon, "Maximum likelihood training of score-based diffusion models," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 1415–1428.

[66] Y. Song and S. Ermon, "Generative modeling by estimating gradients of the data distribution," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, 2019, pp. 1–13

[67] Y. Song, J. Sohl-Dickstein, D. P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," in *Proc. Int. Conf. Learn. Represent.*, 2021, pp. 1–36.

[68] S. Sreekumar and Z. Goldfeld, "Neural estimation of statistical divergences," *J. Mach. Learn. Res.*, vol. 23, no. 126, pp. 1–75, 2022.

[69] B. Tzen and M. Raginsky, "Theoretical guarantees for sampling and inference in generative models with latent diffusions," in *Proc. Conf. Learn. Theory*, 2019, pp. 3084–3114.

[70] A. Vidal, S. Wu Fung, L. Tenorio, S. Osher, and L. Nurbekyan, "Taming hyperparameter tuning in continuous normalizing flows using the JKO scheme," *Sci. Rep.*, vol. 13, no. 1, p. 4501, Mar. 2023.

[71] P. Vincent, "A connection between score matching and denoising autoencoders," *Neural Comput.*, vol. 23, no. 7, pp. 1661–1674, Jul. 2011.

[72] C. Xu, X. Cheng, and Y. Xie, "Invertible neural networks for graph prediction," *IEEE J. Select. Areas Inf. Theory*, vol. 3, no. 3, pp. 454–467, Sep. 2022.

[73] C. Xu, X. Cheng, and Y. Xie, "Computing high-dimensional optimal transport by flow neural networks," 2023, *arXiv:2305.11857*.

[74] C. Xu, X. Cheng, and Y. Xie, "Normalizing flow neural networks by JKO scheme," in *Proc. Conf. Neural Inf. Process. Syst.*, 2023, pp. 1–11.

[75] Y. Yang, Z. Li, and Y. Wang, "On the capacity of deep generative networks for approximating distributions," *Neural Netw.*, vol. 145, pp. 144–154, Jan. 2022.

[76] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2223–2232.

Xiuyuan Cheng received the Ph.D. degree from Princeton University in 2013. She was a Post-Doctoral Researcher with the Ecole Normale Superieure, Paris, France, from 2013 to 2015. She was a Gibbs Assistant Professor with Yale from 2015 to 2017. She is currently an Associate Professor in mathematics with Duke University. She works on theoretical and computational techniques to solve problems in high-dimensional data analysis, signal processing, and machine learning. Her work has been recognized by a Sloan Fellowship and a NSF CAREER Award.

Jianfeng Lu received the Ph.D. degree in applied mathematics from Princeton University in 2009. He is currently a James B. Duke Professor in mathematics, physics, and chemistry with Duke University. He was a Courant Instructor with New York University from 2009 to 2012. He works on mathematical analysis and algorithm development for problems and challenges arising from computational physics, theoretical chemistry, materials science, high-dimensional PDEs, and machine learning. He is a fellow of AMS. His work has been recognized by a Sloan Fellowship, a NSF CAREER Award, the IMA Prize in Mathematics and its Applications, and the Feng Kang Prize.

Yixin Tan received the Ph.D. degree from Duke University in 2024. His research focuses on the analysis of diffusion-based sampling methods and their applications on generative models.

Yao Xie (Member, IEEE) received the Ph.D. degree in electrical engineering (minor in mathematics) from Stanford University in 2012. She was a Research Scientist with Duke University. She is currently the Coca-Cola Foundation Chair, a Professor with Georgia Institute of Technology, H. Milton Stewart School of Industrial and Systems Engineering, and the Associate Director of the Machine Learning Center. From September 2017 to May 2023, she was the Harold R. and Mary Anne Nash Early Career Professor. Her research interests include the intersection of statistics, machine learning, and optimization in providing theoretical guarantees and developing computationally efficient and statistically powerful methods for problems motivated by real-world applications. She received the National Science Foundation (NSF) CAREER Award in 2017, the INFORMS Wagner Prize Finalist in 2021, and the INFORMS Gaver Early Career Award for Excellence in Operations Research in 2022. She is currently an Associate Editor of IEEE TRANSACTIONS ON INFORMATION THEORY, *Journal of the American Statistical Association*, *Theory and Methods*, *Operations Research*, *Sequential Analysis: Design Methods and Applications*, and *INFORMS Journal on Data Science*; and the Area Chair of NeurIPS, ICML, and ICLR.