

# Deep Generative Learning via Euler Particle Transport

**Yuan Gao**

*School of Mathematics and Statistics  
Xi'an Jiaotong University, Xi'an, China*

XJTUYGAO@GMAIL.COM

**Jian Huang**

*Department of Statistics and Actuarial Science  
University of Iowa, Iowa City, USA*

JIAN-HUANG@UIOWA.EDU

**Yuling Jiao**

*School of Mathematics and Statistics  
Wuhan University, Wuhan, China*

YULINGJIAOMATH@WHU.EDU.CN

**Jin Liu**

*Center of Quantitative Medicine  
Duke-NUS Medical School, Singapore*

JIN.LIU@DUKE-NUS.EDU.SG

**Xiliang Lu**

*School of Mathematics and Statistics  
Wuhan University, Wuhan, China*

XLLV.MATH@WHU.EDU.CN

**Zhijian Yang**

*School of Mathematics and Statistics  
Wuhan University, Wuhan, China*

ZJYANG.MATH@WHU.EDU.CN

**Editors:** Joan Bruna, Jan S Hesthaven, Lenka Zdeborova

## Abstract

We propose an Euler particle transport (EPT) approach to generative learning. EPT is motivated by the problem of constructing an optimal transport map from a reference distribution to a target distribution characterized by the Monge-Ampère equation. Interpreting the infinitesimal linearization of the Monge-Ampère equation from the perspective of gradient flows in measure spaces leads to a stochastic McKean-Vlasov equation. We use the forward Euler method to solve this equation. The resulting forward Euler map pushes forward a reference distribution to the target. This map is the composition of a sequence of simple residual maps, which are computationally stable and easy to train. The key task in training is the estimation of the density ratios or differences that determine the residual maps. We estimate the density ratios based on the Bregman divergence with a gradient penalty using deep density-ratio fitting. We show that the proposed density-ratio estimators do not suffer from the “*curse of dimensionality*” if data is supported on a lower-dimensional manifold. Numerical experiments with multi-mode synthetic datasets and comparisons with the existing methods on real benchmark datasets support our theoretical results and demonstrate the effectiveness of the proposed method.

**Keywords:** Density-ratio estimation; gradient flow; high-dimensional distribution; residual map; sampling; velocity fields

## 1. Introduction

The ability to efficiently sample from complex distributions plays a key role in a variety of prediction and inference tasks in machine learning and statistics (Salakhutdinov, 2015). The long-standing methodology for learning an underlying distribution relies on an explicit statistical data model, which can be difficult to specify in many applications such as image analysis, computer vision and natural language processing. In contrast, implicit generative models do not assume a specific form of the data distribution, but rather learn a nonlinear map to transform a reference distribution to the target distribution. This modeling approach has been shown to achieve impressive performance in many machine learning tasks (Reed et al., 2016; Zhu et al., 2017). *Generative adversarial networks* (GAN) (Goodfellow et al., 2014), *variational auto-encoders* (VAE) (Kingma and Welling, 2014) and *flow-based methods* (Rezende and Mohamed, 2015) are important representatives of implicit generative models.

GANs model the low-dimensional latent structure via deep nonlinear factors. They are trained by sequentially differentiable surrogates of two-sample tests, including the density-ratio test (Goodfellow et al., 2014; Nowozin et al., 2016; Mao et al., 2017; Mroueh and Sercu, 2017; Tao et al., 2018) and the density-difference test (Li et al., 2015; Sutherland et al., 2017; Li et al., 2017; Arjovsky et al., 2017; Binkowski et al., 2018), among others. VAE is a probabilistic deep latent factor model trained with variational inference and stochastic approximation. Several authors have proposed improved versions of VAE by enhancing the representation power of the learned latent codes and reducing the blurriness of the generated images in vanilla VAE (Makhzani et al., 2016; Higgins et al., 2017; Tolstikhin et al., 2018; Zhang et al., 2019). Flow-based methods learn a diffeomorphism map between the reference distribution and the target distribution by maximum likelihood using the change of variables formula. Recent work on flow-based methods has been focused on developing training methods and designing neural network architectures to trade off between the efficiency of training and sampling and the representation power of the learned map (Rezende and Mohamed, 2015; Dinh et al., 2015, 2017; Kingma et al., 2016; Papamakarios et al., 2017; Kingma and Dhariwal, 2018; Grathwohl et al., 2019).

We propose an Euler particle transport (EPT) approach to learning a generative model by integrating ideas from optimal transport, numerical ODE, density-ratio estimation and deep neural networks. EPT is motivated by the problem of finding an optimal transport from a reference distribution to the target distribution based on the quadratic Wasserstein distance. Since it is challenging to solve the Monge-Ampère equation that characterizes the optimal transport, we consider the McKean-Vlasov equation derived from the linearization of the Monge-Ampère equation, which is associated with a gradient flow converging to the target distribution. We solve the McKean-Vlasov equation using the forward Euler method. The resulting EPT that pushes forward a reference distribution to the target is a composition of a sequence of simple residual maps that are computationally stable and easy to train. The residual maps are completely determined by the density ratios between the distributions at the current iterations and the target distribution. We estimate the density ratios using neural networks based on the Bregman divergence with a gradient regularizer.

We establish bounds on the approximation errors due to linearization of the Monge-Ampère equation, Euler discretization of the McKean-Vlasov equation, and deep density-ratio estimation. Our result on the error rate for the proposed density-ratio estimators improves the minimax rate of nonparametric estimation via exploring the low-dimensional structure of the data and circumvents the “*curse of dimensionality*”. Experimental results on multi-mode synthetic data and comparisons with state-of-the-art GANs on benchmark data support our theoretical findings and demonstrate that

EPT is computationally more stable and easier to train than GANs. Using simple ReLU ResNets without batch normalization and spectral normalization, we obtained results that are better than or comparable with those using GANs trained with such tricks.

## 2. Euler particle transport: method description

Let  $X \in \mathbb{R}^m$  be a random vector with distribution  $\nu$  and let  $Z \in \mathbb{R}^m$  be a random vector with distribution  $\mu$ , where  $\nu$  is the target distribution we wish to learn and  $\mu$  is a known reference distribution. We assume that  $\mu$  has a simple form and is easy to sample from. Our goal is to construct a transformation  $\mathcal{T}$  such that  $\mathcal{T}_\# \mu = \nu$ , where  $\mathcal{T}_\# \mu$  denotes the push-forward distribution of  $\mu$  by  $\mathcal{T}$ , that is, the distribution of  $\mathcal{T}(Z)$ . Then we can sample from  $\nu$  by first generating a  $Z \sim \mu$  and calculate  $\mathcal{T}(Z)$ . In practice,  $\nu$  is unknown and only a random sample  $\{X_i\}_{i=1}^n$  i.i.d.  $\nu$  is available, our task is to construct  $\mathcal{T}$  based on the sample.

In this section we provide an overall description of EPT. The motivation and its connections with optimal transport theory and gradient flows are given in Section 3. EPT uses a sequence of residual maps to move the particles from the reference  $\mu$  gradually to the target  $\nu$ . The key question is how to determine the direction of the movement. We determine the direction via minimizing a measure of the discrepancy between the density of the particles and the density of the observations.

Specifically, suppose we have generated an initial particle  $\mathbf{X}_0 \in \mathbb{R}^m$  from the reference distribution  $\mu$ . Let  $\mu_0 = \mu$  and let  $s > 0$  be a small step size. EPT pushes forward  $\mathbf{X}_0$  to the target  $\nu$  iteratively using residual maps as follows,

$$\mathcal{T}_k = \mathbb{1} + s\mathbf{v}_k, \quad (1)$$

$$\mathbf{X}_{k+1} = \mathcal{T}_k(\mathbf{X}_k), \quad (2)$$

$$\mu_{k+1} = (\mathcal{T}_k)_\# \mu_k, \quad (3)$$

where  $\mathbb{1}$  is the identity map and  $\mathbf{v}_k$  is the velocity field at the  $k$ th step,  $k = 0, 1, \dots, K$  for some large  $K$ . As explained in Section 3, this is a discretized version of the continuous process  $\{\mathbf{X}_t\}_{t \geq 0}$  determined by the McKean-Vlasov equation (10) given below. The final transport map is the composition of the residual maps  $\mathcal{T}_0, \mathcal{T}_1, \dots, \mathcal{T}_K$ , i.e.,

$$\mathcal{T} = \mathcal{T}_K \circ \mathcal{T}_{K-1} \circ \dots \circ \mathcal{T}_0. \quad (4)$$

This updating scheme is based on the forward Euler method for solving equation (10). So we refer to the proposed method as Euler particle transport (EPT).

Given the initial particle  $\mathbf{X}_0 \sim \mu_0 \equiv \mu$ , the iteration scheme (1)-(3) is to move  $\mathbf{X}_0$  from  $\mu_0$  to  $\mu_K$  one step at a time. The processes  $\{\mathbf{X}_k\}_{k \geq 0}$  and  $\{\mu_k\}_{k \geq 0}$  defined by (1)-(3) are completely determined by the velocity fields  $\mathbf{v}_k$ . How do we determine the velocity fields  $\mathbf{v}_k$  to ensure that  $\mathbf{X}_k$  moves closer to  $\nu$  at each update and  $\mu_K \approx \nu$  approximately? The basic intuition is that we should move in the direction that decreases the discrepancy between  $\mu_k$  and the target  $\nu$ . We use an energy functional  $\mathcal{L}[\mu_k]$  to measure such discrepancy. An important energy functional is the  $f$ -divergence (Ali and Silvey, 1966),

$$\mathcal{L}[\mu_k] = \mathbb{D}_f(\mu_k \| \nu) = \int_{\mathbb{R}^m} p(\mathbf{x}) f\left(\frac{q_k(\mathbf{x})}{p(\mathbf{x})}\right) d\mathbf{x}, \quad (5)$$

where  $q_k$  is the density of  $\mu_k$ ,  $p$  is the density of  $\nu$  and  $f : \mathbb{R}^+ \rightarrow \mathbb{R}$  is a given twice-differentiable convex function with  $f(1) = 0$ . We choose  $\mathbf{v}_k$  such that  $\mathcal{L}[\mu_k]$  is minimized. Theorem 5 in Subsection 3.3 shows that this leads to

$$\mathbf{v}_k(\mathbf{x}) = -f''(r_k(\mathbf{x}))\nabla r_k(\mathbf{x}), \text{ where } r_k(\mathbf{x}) = \frac{q_k(\mathbf{x})}{p(\mathbf{x})}, \mathbf{x} \in \mathbb{R}^m.$$

For example, if we use the Pearson  $\chi^2$ -divergence with  $f(x) = (x - 1)^2/2$ , then  $\mathbf{v}_k(\mathbf{x}) = -\nabla r_k(\mathbf{x})$  is simply the negative gradient of the density ratio. Other types of velocity fields can be obtained by using different energy functionals such as the Lebesgue norm of the density difference, see Subsection 3.3 for details.

When the target  $\nu$  is unknown and only a random sample is available, it is natural to train  $\mathcal{T}$  by first estimating the velocity fields  $\mathbf{v}_k$  at the sample level and then plugging the estimator of  $\mathbf{v}_k$  in (1). For example, if we use the  $f$ -divergence as the energy functional, estimating  $\mathbf{v}_k(\mathbf{x}) = -f''(r_k(\mathbf{x}))\nabla r_k(\mathbf{x})$  boils down to estimating the density ratios  $r_k(\mathbf{x}) = q_k(\mathbf{x})/p(\mathbf{x})$  dynamically at each iteration  $k$ . Nonparametric density-ratio estimation using Bregman divergences and gradient regularizers are discussed in Section 4 below. We estimate the density ratios with deep neural networks. Let  $\hat{\mathbf{v}}_k$  be the estimated velocity field at the  $k$ th iteration. The  $k$ th estimated residual map is  $\hat{\mathcal{T}}_k = \mathbb{1} + s\hat{\mathbf{v}}_k$ . The trained map corresponding to (2) is

$$\hat{\mathcal{T}} = \hat{\mathcal{T}}_K \circ \hat{\mathcal{T}}_{K-1} \circ \dots \circ \hat{\mathcal{T}}_0. \tag{6}$$

We will use another neural network to preserve the information about this composition map so that it is convenient to generate new samples based on the reference distribution without additional training. This will also allow us to use a reference distribution with a different dimension from that of the target distribution. The details are given in Section 5.

### 3. Motivation and theoretical analysis

#### 3.1. Motivation

We describe the motivation of the proposed EPT and its connection with the optimal transport theory. Consider the quadratic Wasserstein distance between  $\mu$  and  $\nu$  defined by

$$\mathcal{W}_2(\mu, \nu) = \left\{ \inf_{\gamma \in \Gamma(\mu, \nu)} \mathbb{E}_{(Z, X) \sim \gamma} [\|Z - X\|_2^2] \right\}^{\frac{1}{2}}, \tag{7}$$

where  $\Gamma(\mu, \nu)$  denotes the set of couplings of  $(\mu, \nu)$  (Villani, 2008; Ambrosio et al., 2008). Suppose that  $\mu$  and  $\nu$  have densities  $q$  and  $p$  with respect to the Lebesgue measure, respectively. Then the optimal transport map  $\mathcal{T}$  such that  $\mathcal{T}_\# \mu = \nu$  is characterized by the Monge-Ampère equation (Brenier, 1991; McCann, 1995; Santambrogio, 2015). Specifically, the minimization problem in (7) admits a unique solution  $\gamma = (\mathbb{1}, \mathcal{T})_\# \mu$  with  $\mathcal{T} = \nabla \Psi$ ,  $\mu$ -a.e., where  $\mathbb{1}$  is the identity map and  $\nabla \Psi$  is the gradient of the potential function  $\Psi : \mathbb{R}^m \rightarrow \mathbb{R}$ . This function is convex and satisfies the Monge-Ampère equation

$$\det(\nabla^2 \Psi(\mathbf{z})) = \frac{q(\mathbf{z})}{p(\nabla \Psi(\mathbf{z}))}, \mathbf{z} \in \mathbb{R}^m. \tag{8}$$

In theory, to find the optimal transport  $\mathcal{T}$ , it suffices to solve (8) for  $\Psi$ . However, in practice it is infeasible to solve this degenerate elliptic equation in high-dimensional settings due to its highly

nonlinear nature, particularly when the target density function  $p$  is unknown and only a random sample from  $p$  is available.

A basic approach to addressing the difficulty due to nonlinearity is linearization. Instead of attempting to solve the Monge-Ampère equation (8), we linearize (8) using residual maps and solve the resulting linearized versions of (8). These linearized versions are stochastic ordinary differential equations, called the McKean-Vlasov equations given in (10) below. For a given residual map, EPT solves a discretized McKean-Vlasov equation using the forward Euler method, which leads to the particle updating scheme (1) to (3). We note that different residual maps generally lead to different linearized versions of (8). Thus although the optimal transport characterized by (8) is unique, the transport trained via EPT is not. In Subsection 3.3 below, we establish the connection between EPT and the gradient flows from  $\mu$  to  $\nu$ . Different EPT corresponding to different residuals maps all push forward  $\mu$  to  $\nu$ , although along different gradient flows. See also Proposition 2 and Remark 4 below for the connection between EPT and the optimal transport in a local sense.

We employ a linearization scheme using the residual map

$$\mathcal{T}_{t,\Phi} = \nabla\Psi = \mathbb{1} + t\nabla\Phi_t, t \geq 0, \tag{9}$$

where  $\Phi_t : \mathbb{R}^m \rightarrow \mathbb{R}^1$  is a function to be chosen such that the law of  $\mathcal{T}_{t,\Phi}(Z)$  is closer to  $\nu$  than that of  $Z$  (Villani, 2008). We then iteratively improve the approximation by repeatedly applying the residual map to the current particles. The specific forms of  $\Phi_t$  in (22) and (23) are given in Theorem 5 in Subsection 3.3.

The linearization based on (9) leads to the stochastic process  $\mathbf{X}_t : \mathbb{R}^m \rightarrow \mathbb{R}^m$  satisfying the McKean-Vlasov equation

$$\frac{d}{dt}\mathbf{X}_t(\mathbf{x}) = \mathbf{v}_t(\mathbf{X}_t(\mathbf{x})), t \geq 0, \text{ with } \mathbf{X}_0 \sim \mu, \mu\text{-a.e. } \mathbf{x} \in \mathbb{R}^m, \tag{10}$$

where  $\mathbf{v}_t$  is the velocity fields of  $\mathbf{X}_t$ . In addition, we show that  $\mathbf{v}_t = \nabla\Phi_t$ . Thus  $\mathbf{v}_t$  also determines the residual map (9). The details of the derivation are given in Theorems 5 and 6 in Subsection 3.3. Therefore, the problem of estimating the residual maps (9) is equivalent to that of estimating the velocity fields  $\mathbf{v}_t$ . The EPT updating process (1)-(3) is the Euler forward method for solving a discretized version of (10). We choose a  $\mathbf{v}_t$  to decrease the discrepancy between the distribution of  $\mathbf{X}_t$ , say  $\mu_t$ , at time  $t$  and the target  $\nu$  with respect to a properly chosen measure such as the  $f$ -divergence given in (5).

An equivalent formulation of (10) is through the gradient flow  $\{\mu_t\}_{t \geq 0}$  with  $\{\mathbf{v}_t\}_{t \geq 0}$  as its velocity fields, see Proposition 2 in Subsection 3.3. Computationally it is more convenient to work with (10). However, for the error analysis of EPT, it is useful to consider the connection between EPT and the gradient flow.

### 3.2. Summary of error analysis results

Here we provide an overview of the error bounds due to linearization, discretization and nonparametric density ratio estimation. Detailed descriptions are given in Subsection 3.3.

We establish the following bound on the approximation error due to the linearization of the Monge-Ampère equation under appropriate conditions:

$$\mathcal{W}_2(\mu_t, \nu) = \mathcal{O}(e^{-\lambda t}), \tag{11}$$

for some  $\lambda > 0$ , see Proposition 2 in Subsection 3.3. Therefore,  $\mu_t$  converges to  $\nu$  exponentially fast as  $t \rightarrow \infty$ . For an integer  $K \geq 1$  and a small  $s > 0$ , let  $\{\mu_t^s : t \in [ks, (k+1)s], k = 0, 1, \dots, K\}$  be a piecewise constant interpolation between  $\mu_{ks}$  and  $\mu_{(k+1)s}$ ,  $k = 0, 1, \dots, K$ .

Under the assumption that the velocity fields  $\mathbf{v}_t$  are Lipschitz continuous with respect to  $(\mathbf{x}, \mu_t)$ , it is shown in Proposition 8 in Subsection 3.3 that the discretization error of  $\mu_t^s$  can be bounded in a finite time interval  $[0, T)$  as follows:

$$\sup_{t \in [0, T)} \mathcal{W}_2(\mu_t, \mu_t^s) = \mathcal{O}(s). \quad (12)$$

The error bounds (11) and (12) imply that the distributions of the particles  $\mathbf{X}_k$  generated by the EPT map defined in (2) with a small  $s$  and a sufficiently large  $k$  converges to the target  $\nu$  at the rate of discretization size  $s$ .

In Theorem 11 in Section 4, we provide an error bound for the density ratio estimation. Our result improves the minimax rate of deep nonparametric estimation via exploring the low-dimensional structure of data and circumvents the “*curse of dimensionality*.” Thus deep neural networks are capable of adaptively estimating the density ratio supported on a lower-dimensional manifold.

### 3.3. Gradient flows associated with EPT

For convenience, we first define the notations used in the remaining sections. Let  $\mathcal{P}_2(\mathbb{R}^m)$  denote the space of Borel probability measures on  $\mathbb{R}^m$  with finite second moments, and let  $\mathcal{P}_2^a(\mathbb{R}^m)$  denote the subset of  $\mathcal{P}_2(\mathbb{R}^m)$  in which measures are absolutely continuous with respect to the Lebesgue measure (all distributions are assumed to satisfy this assumption hereinafter).  $\text{Tan}_\mu \mathcal{P}_2(\mathbb{R}^m)$  denotes the tangent space to  $\mathcal{P}_2(\mathbb{R}^m)$  at  $\mu$ . Let  $\text{AC}_{\text{loc}}(\mathbb{R}^+, \mathcal{P}_2(\mathbb{R}^m)) = \{\mu_t : I \rightarrow \mathcal{P}_2(\mathbb{R}^m) \text{ is absolutely continuous, } |\mu'_t| \in L^2(I), I \subset \mathbb{R}^+\}$ .  $\text{Lip}_{\text{loc}}(\mathbb{R}^m)$  denotes the set of functions that are Lipschitz continuous on any compact set of  $\mathbb{R}^m$ . For any  $\ell \in [1, \infty]$ , we use  $L^\ell(\mu, \mathbb{R}^m)$  ( $L_{\text{loc}}^\ell(\mu, \mathbb{R}^m)$ ) to denote the  $L^\ell$  space of  $\mu$ -measurable functions on  $\mathbb{R}^m$  (on any compact set of  $\mathbb{R}^m$ ). With  $\mathbb{1}$ ,  $\det$  and  $\text{tr}$ , we refer to the identity map, the determinant and the trace. We use  $\nabla$ ,  $\nabla^2$ ,  $\nabla \cdot$  and  $\Delta$  to denote the gradient or Jacobian operator, the Hessian operator, the divergence operator and the Laplace operator, respectively.

We are now ready to establish the connection between EPT and the gradient flows corresponding to the McKean-Vlasov equation (10). Let  $X \sim q$ , and let

$$\tilde{X} = \mathcal{T}_{t, \Phi}(X) = X + t \nabla \Phi(X), t \geq 0.$$

Here we let  $\Phi$  be independent of  $t$  for the moment. Denote the distribution of  $\tilde{X}$  by  $\tilde{q}$ . With a small  $t$ , the map  $\mathcal{T}_{t, \Phi}$  is invertible according to the implicit function theorem. By the change of variables formula, we have

$$\det(\nabla^2 \Psi)(\mathbf{x}) = |\det(\nabla \mathcal{T}_{t, \Phi})(\mathbf{x})| = \frac{q(\mathbf{x})}{\tilde{q}(\tilde{\mathbf{x}})}, \quad (13)$$

where

$$\tilde{\mathbf{x}} = \mathcal{T}_{t, \Phi}(\mathbf{x}). \quad (14)$$

Using the fact that the derivative  $\frac{d}{dt} \Big|_{t=0} \det(\mathbf{A} + t\mathbf{B}) = \det(\mathbf{A}) \text{tr}(\mathbf{A}^{-1}\mathbf{B})$ ,  $\forall \mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times m}$ , provided that  $\mathbf{A}$  is invertible, and applying the first order Taylor expansion to (13), we have

$$\log \tilde{q}(\tilde{\mathbf{x}}) - \log q(\mathbf{x}) = -t \Delta \Phi(\mathbf{x}) + o(t). \quad (15)$$

Let  $t \rightarrow 0$  in (14) and (15) and let  $\mathbf{x}_0$  be a realization of the random variable sampled from  $q$ . We obtain a random process  $\{\mathbf{x}_t\}_{t \geq 0}$  and its laws  $\{q_t\}_{t \geq 0}$  satisfying

$$\frac{d\mathbf{x}_t}{dt} = \nabla \Phi(\mathbf{x}_t), \quad t \geq 0, \quad (16)$$

$$\frac{d \ln q_t(\mathbf{x}_t)}{dt} = -\Delta \Phi(\mathbf{x}_t), \quad \text{with } q_0 = q. \quad (17)$$

Equations (16) and (17) resulting from linearizing the Monge-Ampère equation (8) can be interpreted as gradient flows in measure spaces (Ambrosio et al., 2008). Thanks to this connection, we can resort to solving a continuity equation characterized by a McKean-Vlasov equation, an ODE system that is easier to work with.

For  $\mu \in \mathcal{P}_2^a(\mathbb{R}^m)$  with density  $q$ , let

$$\mathcal{L}[\mu] = \int_{\mathbb{R}^m} F(q(\mathbf{x})) d\mathbf{x} : \mathcal{P}_2^a(\mathbb{R}^m) \rightarrow \mathbb{R}^+ \cup \{0\} \quad (18)$$

be an energy functional satisfying  $\nu \in \arg \min \mathcal{L}[\cdot]$ , where  $F(\cdot) : \mathbb{R}^+ \rightarrow \mathbb{R}^1$  is a twice-differentiable convex function. Among the widely used measures on  $\mathcal{P}_2^a(\mathbb{R}^m)$  in generative learning, the following two are important examples of  $\mathcal{L}[\cdot]$ : (i) the  $f$ -divergence given in (5) (Ali and Silvey, 1966); (ii) the Lebesgue norm of density difference:

$$\|\mu - \nu\|_{L^2(\mathbb{R}^m)}^2 = \int_{\mathbb{R}^m} |q(\mathbf{x}) - p(\mathbf{x})|^2 d\mathbf{x}. \quad (19)$$

**Definition 1** We call  $\{\mu_t\}_{t \geq 0} \subset \text{AC}_{\text{loc}}(\mathbb{R}^+, \mathcal{P}_2(\mathbb{R}^m))$  a gradient flow of the functional  $\mathcal{L}[\cdot]$ , if  $\{\mu_t\}_{t \geq 0} \subset \mathcal{P}_2^a(\mathbb{R}^m)$  a.e.,  $t \in \mathbb{R}^+$  and the velocity fields  $\mathbf{v}_t \in \text{Tan}_{\mu_t} \mathcal{P}_2(\mathbb{R}^m)$  satisfies  $\mathbf{v}_t \in -\partial \mathcal{L}[\mu_t]$  a.e.  $t \geq 0$ , where  $\partial \mathcal{L}[\cdot]$  is the subdifferential of  $\mathcal{L}[\cdot]$ .

The gradient flow  $\{\mu_t\}_{t \geq 0}$  of  $\mathcal{L}[\cdot]$  enjoys the following nice properties.

### Proposition 2

(i) The following continuity equation holds in the sense of distributions.

$$\frac{\partial}{\partial t} \mu_t = -\nabla \cdot (\mu_t \mathbf{v}_t) \quad \text{in } [0, \infty) \times \mathbb{R}^m \quad \text{with } \mu_0 = \mu. \quad (20)$$

(ii) Energy decay along the gradient flow:  $\frac{d}{dt} \mathcal{L}[\mu_t] = -\|\mathbf{v}_t\|_{L^2(\mu_t, \mathbb{R}^m)}^2$  a.e.  $t \geq 0$ . In addition,

$$\mathcal{W}_2(\mu_t, \nu) = \mathcal{O}(\exp^{-\lambda t}), \quad (21)$$

if  $\mathcal{L}[\mu]$  is  $\lambda$ -geodetically convex with  $\lambda > 0$ <sup>1</sup>.

---

1.  $\mathcal{L}$  is said to be  $\lambda$ -geodetically convex if there exists a constant  $\lambda > 0$  such that for every  $\mu_1, \mu_2 \in \mathcal{P}_2^a(\mathbb{R}^m)$ , there exists a constant speed geodesic  $\gamma : [0, 1] \rightarrow \mathcal{P}_2^a(\mathbb{R}^m)$  such that  $\gamma_0 = \mu_1, \gamma_1 = \mu_2$  and

$$\mathcal{L}(\gamma_s) \leq (1-s)\mathcal{L}(\mu_1) + s\mathcal{L}(\mu_2) - \frac{\lambda}{2}s(1-s)d(\mu_1, \mu_2), \quad \forall s \in [0, 1],$$

where  $d$  is a metric defined on  $\mathcal{P}_2^a(\mathbb{R}^m)$  such as the quadratic Wasserstein distance.

(iii) Conversely, if  $\{\mu_t\}_{t \geq 0}$  is the solution of continuity equation (20) in (i) with  $\mathbf{v}_t(\mathbf{x})$  specified by (22), then  $\{\mu_t\}_{t \geq 0}$  is a gradient flow of  $\mathcal{L}[\cdot]$ .

**Remark 3** In part (ii) of Proposition 2, for general  $f$ -divergences, we assume the functional  $\mathcal{L}$  to be  $\lambda$ -geodetically convex for the convergence of  $\mu_t$  to the target  $\nu$  in the quadratic Wasserstein distance. However, for the KL divergence, the convergence can be guaranteed if  $\nu$  satisfies the log-Sobolev inequality (Otto and Villani, 2000). In addition, the distributions that are strongly log-concave outside a bounded region, but not necessarily log-concave inside the region satisfy the log-Sobolev inequality, see, for example, Holley and Stroock (1987). Here the functional  $\mathcal{L}$  can even be nonconvex, an example includes the densities with double-well potential.

**Remark 4** Equation (8.48) in Proposition 8.4.6 of Ambrosio et al. (2008) shows the connection (in a local sense) of the velocity  $\mathbf{v}_t$  of the gradient flow  $\mu_t$  and the optimal transport along  $\mu_t$ , i.e., let  $T_{\mu_t}^{\mu_{t+h}}$  be the optimal transport from  $\mu_t$  to  $\mu_{t+h}$  for a small  $h > 0$ , then  $T_{\mu_t}^{\mu_{t+h}} = \mathbb{1} + h\mathbf{v}_t + o(h)$  in  $L^p(\mathbb{R}^m)$ . So locally,  $\mathbb{1} + h\mathbf{v}_t$  approximates the optimal transport map from  $\mu_t$  to  $\mu_{t+h}$  on  $[t, t+h]$ . However, the global approximation property of the proposed method is not clear. This is a challenging problem that requires further study and is beyond the scope of this paper.

The following result makes the connection between the linearized Monge-Ampère equations (16)-(17) and the gradient flow defined in (20).

**Theorem 5** (i) Representation of the velocity fields: if the density  $q_t$  of  $\mu_t$  is differentiable, then

$$\mathbf{v}_t(\mathbf{x}) = -\nabla F'(q_t(\mathbf{x})) \quad \mu_t\text{-a.e. } \mathbf{x} \in \mathbb{R}^m. \quad (22)$$

(ii) If we let  $\Phi$  be time-dependent in (16)–(17), i.e.,  $\Phi_t$ , then the linearized Monge-Ampère equations (16)–(17) are the same as the continuity equation (20) by taking

$$\Phi_t(\mathbf{x}) = -F'(q_t(\mathbf{x})). \quad (23)$$

Theorem 5 and (21) in Proposition 2 imply that  $\{\mu_t\}_{t \geq 0}$ , the solution of the continuity equation (20) with  $\mathbf{v}_t(\mathbf{x}) = -\nabla F'(q_t(\mathbf{x}))$ , converges rapidly to the target distribution  $\nu$ . Furthermore, the continuity equation has the following representation under mild regularity conditions on the velocity fields.

**Theorem 6** Assume  $\|\mathbf{v}_t\|_{L^1(\mu_t, \mathbb{R}^m)} \in L^1_{\text{loc}}(\mathbb{R}^+)$  and  $\mathbf{v}_t(\cdot) \in \text{Lip}_{\text{loc}}(\mathbb{R}^m)$  with upper bound  $B_t$  and Lipschitz constant  $L_t$  such that  $(B_t + L_t) \in L^1_{\text{loc}}(\mathbb{R}^+)$ . Then the solution of the continuity equation (20) can be represented as  $\mu_t = (\mathbf{X}_t)_\# \mu$ , where  $\mathbf{X}_t(\mathbf{x}) : \mathbb{R}^+ \times \mathbb{R}^m \rightarrow \mathbb{R}^m$  satisfies the McKean-Vlasov equation (10).

As shown in Lemma 7 below, the velocity fields associated with the  $f$ -divergence (5) and the Lebesgue norm (19) are determined by density ratio and density difference respectively.

**Lemma 7** The velocity fields  $\mathbf{v}_t$  satisfy

$$\mathbf{v}_t(\mathbf{x}) = \begin{cases} -f''(r_t(\mathbf{x}))\nabla r_t(\mathbf{x}), & \mathcal{L}[\mu] = \mathbb{D}_f(\mu\|\nu), \text{ where } r_t(\mathbf{x}) = \frac{q_t(\mathbf{x})}{p(\mathbf{x})}, \\ -2\nabla d_t(\mathbf{x}), & \mathcal{L}[\mu] = \|\mu - \nu\|_{L^2(\mathbb{R}^m)}^2, \text{ where } d_t(\mathbf{x}) = q_t(\mathbf{x}) - p(\mathbf{x}). \end{cases}$$



Several methods have been developed to estimate density ratio and density difference in the literature. Examples include probabilistic classification approaches, moment matching and direct density-ratio (difference) fitting, see [Sugiyama et al. \(2012a,b\)](#); [Kanamori and Sugiyama \(2014\)](#); [Mohamed and Lakshminarayanan \(2016\)](#) and the references therein.

**Proposition 8** *For any finite  $T > 0$ , suppose that the velocity fields  $\mathbf{v}_t$  are Lipschitz continuous with respect to  $(\mathbf{x}, \mu_t)$  for  $t \in [0, T]$ , that is, there exists a finite constant  $L_v > 0$  such that*

$$\|\mathbf{v}_t(\mathbf{x}) - \mathbf{v}_{\tilde{t}}(\tilde{\mathbf{x}})\| \leq L_v[\|\mathbf{x} - \tilde{\mathbf{x}}\| + \mathcal{W}_2(\mu_t, \mu_{\tilde{t}})], t, \tilde{t} \in [0, T] \text{ and } \mathbf{x}, \tilde{\mathbf{x}} \in \mathbb{R}^m. \quad (24)$$

Then the bound (12) on the discretization error holds.

**Remark 9** *If we take  $f(x) = (x - 1)^2/2$  in Lemma 7, then the velocity fields  $\mathbf{v}_t(\mathbf{x}) = -\nabla \mathbf{r}_t(\mathbf{x})$ , where  $\mathbf{r}_t(\mathbf{x}) = q_t(\mathbf{x})/p(\mathbf{x})$ . In the proof of Theorem 5, part (ii), it is shown that  $q_t$  satisfies  $\partial q_t / \partial t = -\nabla \cdot (q_t \mathbf{v}_t)$ . Thus for this simple  $f$ -divergence function, the verification of the Lipschitz condition (24) amounts to verifying that  $\nabla \mathbf{r}_t(\mathbf{x})$  is Lipschitz continuous in the sense of (24).*

## 4. Deep density-ratio fitting

The evaluation of velocity fields depends on the dynamic estimation of a discrepancy between the push-forward distribution  $q_t$  and the target distribution  $p$ . Density-ratio and density-difference fitting with the Bregman score provides a unified framework for such discrepancy estimation without estimating each density separately ([Gneiting and Raftery, 2007](#); [Dawid, 2007](#); [Sugiyama et al., 2012a,b](#); [Kanamori and Sugiyama, 2014](#)).

Let  $r(\mathbf{x}) = q(\mathbf{x})/p(\mathbf{x})$  be the density ratio between a given density  $q(\mathbf{x})$  and the target  $p(\mathbf{x})$ . Let  $g : \mathbb{R} \rightarrow \mathbb{R}$  be a differentiable and strictly convex function. The separable Bregman score with the base probability density  $p$  for measuring the discrepancy between  $r$  and a measurable function  $R : \mathbb{R}^m \rightarrow \mathbb{R}^1$  is

$$\mathfrak{B}(r, R) = \mathbb{E}_{X \sim p}[g'(R(X))R(X) - g(R(X))] - \mathbb{E}_{X \sim q}[g'(R(X))].$$

We focus on the widely used least-squares density-ratio (LSDR) fitting with  $g(x) = (x - 1)^2$  as a working example, i.e.,

$$\mathfrak{B}_{\text{LSDR}}(r, R) = \mathbb{E}_{X \sim p}[R^2(X)] - 2\mathbb{E}_{X \sim q}[R(X)] + 1. \quad (25)$$

For other choices of  $g$ , such as  $g(x) = x \log x - (x + 1) \log(x + 1)$  corresponding to estimating  $r$  via the logistic regression (LR), and the scenario of density difference fitting will be presented in detail in Appendix B.2.1.

### 4.1. Gradient regularizer

The distributions of real data may have a low-dimensional structure with their support concentrated on low-dimensional manifolds, which may cause the  $f$ -divergence to be ill-posed due to non-overlapping supports. To exploit such underlying low-dimensional structures and avoid ill-posedness, we derive a simple weighted gradient regularizer  $\frac{1}{2}\mathbb{E}_p[g''(R)\|\nabla R\|_2^2]$ , motivated by the recent work on smoothing

via noise injection (Sønderby et al., 2017; Arjovsky and Bottou, 2017). This serves as a regularizer for deep density-ratio fitting. For example, with  $g(x) = (x - 1)^2$ , the resulting gradient regularizer is

$$\mathbb{E}_p[\|\nabla R\|_2^2], \quad (26)$$

which recovers the well-known squared Sobolev semi-norm in nonparametric statistics. Gradient regularization stabilizes and improves the long time performance of EPT. The detailed derivation is presented in Appendix B.2.2.

## 4.2. LSDR estimation with gradient regularizer

Let  $\{X_i\}_{i=1}^n$  and  $\{Y_i\}_{i=1}^n$  be two collections of i.i.d data in  $\mathbb{R}^m$  from densities  $p$  and  $q$ , respectively. Let  $\mathcal{H} \equiv \mathcal{H}_{\mathcal{D}, \mathcal{W}, \mathcal{S}, \mathcal{B}}$  be the set of ReLU neural networks  $R_\phi$  with parameter  $\phi$ , depth  $\mathcal{D}$ , width  $\mathcal{W}$ , size  $\mathcal{S}$ , and  $\|R_\phi\|_\infty \leq \mathcal{B}$ . Here the depth  $\mathcal{D}$  refers to the number of hidden layers, so the network has  $\mathcal{D} + 1$  layers in total. A  $(\mathcal{D} + 1)$ -vector  $(w_0, w_1, \dots, w_{\mathcal{D}})$  specifies the width of each layer, where  $w_0 = m$  is the dimension of the input data and  $w_{\mathcal{D}} = 1$  is the dimension of the output. The width  $\mathcal{W} = \max\{w_1, \dots, w_{\mathcal{D}}\}$  is the maximum width of the hidden layers. The size  $\mathcal{S} = \sum_{i=0}^{\mathcal{D}} [w_i \times (w_i + 1)]$  is the total number of parameters in the network. For multilayer perceptrons with equal-width hidden layers except the output layer, we have  $\mathcal{S} = \mathcal{W}(m + 1) + (\mathcal{W}^2 + \mathcal{W})(\mathcal{D} - 1) + \mathcal{W} + 1$ .

We combine the least squares loss (25) with the gradient regularizer (26) as our objective function. The resulting gradient regularized LSDR estimator of  $r = q/p$  is given by

$$\widehat{R}_\phi \in \arg \min_{R_\phi \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^n [R_\phi^2(X_i) - 2R_\phi(Y_i)] + \alpha \frac{1}{n} \sum_{i=1}^n \|\nabla R_\phi(X_i)\|_2^2, \quad (27)$$

where  $\alpha \geq 0$  is a regularization parameter.

## 4.3. Estimation error bound

We first show that the density ratio  $r$  is identifiable through the objective function by proving that, at the population level, we can recover the density ratio  $r$  via minimizing

$$\mathfrak{B}_{\text{LSDR}}^\alpha(R) = \mathfrak{B}_{\text{LSDR}}(r, R) + \alpha \mathbb{E}_p[\|\nabla R\|_2^2] + \mathcal{C},$$

where  $\mathfrak{B}_{\text{LSDR}}$  is defined in (25) and  $\mathcal{C} = \mathbb{E}_{X \sim q}[r^2(X)] - 1$ .

**Lemma 10** *For any  $\alpha \geq 0$ , we have  $r \in \arg \min_R \mathfrak{B}_{\text{LSDR}}^\alpha(R)$ . In addition,  $\mathfrak{B}_{\text{LSDR}}^\alpha(R) \geq 0$  for any  $R$  with  $\mathbb{E}_{X \sim p} R^2(X) < \infty$ , and  $\mathfrak{B}_{\text{LSDR}}^\alpha(R) = 0$  iff  $R(\mathbf{x}) = r(\mathbf{x}) = 1$  ( $q, p$ )-a.e.  $\mathbf{x} \in \mathbb{R}^m$ .*

This identifiability result shows that the target density ratio is the unique minimizer of the population version of the empirical criterion in (27). This provides a basis for establishing the convergence result of deep nonparametric density-ratio estimation.

Next we bound the nonparametric estimation error  $\|\widehat{R}_\phi - r\|_{L^2(\nu)}$  under the assumptions that the support of  $\nu$  is concentrated on a compact low-dimensional manifold and  $r$  is Lipschitz continuous. Let  $\mathfrak{M} \subseteq [-c, c]^m$  be a Riemannian manifold (Lee, 2010) with dimension  $m_*$ , condition number  $1/\tau$ , volume  $\mathcal{V}$ , geodesic covering regularity  $\mathcal{R}$ , and  $m_* \ll \mathcal{M} = \mathcal{O}(m_* \ln(m\mathcal{V}\mathcal{R}/\tau)) \ll m$ . Denote  $\mathfrak{M}_\epsilon = \{\mathbf{x} \in [-c, c]^m : \inf\{\|\mathbf{x} - \mathbf{y}\|_2 : \mathbf{y} \in \mathfrak{M}\} \leq \epsilon\}$ ,  $\epsilon \in (0, 1)$ .

**Theorem 11** Assume  $\text{supp}(r) = \mathfrak{M}_\epsilon$  and  $r(\mathbf{x})$  satisfies  $|r(\mathbf{x})| \leq B$  for a finite constant  $B > 0$  and is Lipschitz continuous with a Lipschitz constant  $L$ . Suppose the topological parameter of  $\mathcal{H}_{\mathcal{D}, \mathcal{W}, \mathcal{S}, \mathcal{B}}$  in (27) with  $\alpha = 0$  satisfies  $\mathcal{D} = \mathcal{O}(\log n)$ ,  $\mathcal{W} = \mathcal{O}(n^{\frac{\mathcal{M}}{2(2+\mathcal{M})}} / \log n)$ ,  $\mathcal{S} = \mathcal{O}(n^{\frac{\mathcal{M}-2}{\mathcal{M}+2}} / \log^4 n)$ , and  $\mathcal{B} = 2B$ . Then,

$$\mathbb{E}_{\{X_i, Y_i\}_{i=1}^n} [\|\widehat{R}_\phi - r\|_{L^2(\nu)}^2] \leq C(B^2 + cLm\mathcal{M})n^{-2/(2+\mathcal{M})},$$

where  $C$  is a universal constant.

The error bound established in Theorem 11 for the nonparametric deep density-ratio fitting is new. This result is of independent interest for nonparametric estimation with deep neural networks. Since  $\mathcal{M} = \mathcal{O}(m_* \ln(m\mathcal{V}\mathcal{R}/\tau)) \ll m$ , the convergence rate  $\mathcal{O}(n^{-\frac{2}{2+\mathcal{M}}})$  obtained in Theorem 11 is faster than the optimal rate of convergence for nonparametric estimation of a Lipschitz target in  $\mathbb{R}^m$ , where the optimal rate is  $\mathcal{O}(n^{-\frac{2}{2+m}})$  (Stone, 1982; Schmidt-Hieber, 2020), as long as the intrinsic dimension  $\mathcal{M}$  of the data is much smaller than the ambient dimension  $m$ . Therefore, the proposed density-ratio estimators circumvent the ‘‘curse of dimensionality’’ if data is supported on a lower-dimensional manifold.

## 5. Implementation

We now described how to implement EPT and train the transport map  $\mathcal{T}$  with an i.i.d. sample  $\{X_i\}_{i=1}^n \subset \mathbb{R}^m$  from an unknown target distribution  $\nu$ . The EPT map is trained via the forward Euler iteration (1)-(3) with a small step size  $s > 0$ . The resulting map is a composition of a sequence of residual maps, i.e.,  $\mathcal{T}_K \circ \mathcal{T}_{K-1} \circ \dots \circ \mathcal{T}_0$  for a large  $K$ . As implied by Theorem 11 in Section 4, each  $\mathcal{T}_k, k = 0, 1, \dots, K$  can be estimated with high accuracy by  $\widehat{\mathcal{T}}_k = \mathbb{1} + s\widehat{\mathbf{v}}_k$ , where  $\widehat{\mathbf{v}}_k(\mathbf{x}) = -f''(\widehat{R}_\phi(\mathbf{x}))\nabla\widehat{R}_\phi(\mathbf{x})$ . Here  $\widehat{R}_\phi$  is the density-ratio estimator defined in (27) below based on  $\{Y_i\}_{i=1}^n \sim q_k$  and the data  $\{X_i\}_{i=1}^n \sim p$ . Therefore, according to the EPT map (6), the particles

$$\widehat{\mathcal{T}}(\tilde{Y}_i) \equiv \widehat{\mathcal{T}}_K \circ \widehat{\mathcal{T}}_{K-1} \circ \dots \circ \widehat{\mathcal{T}}_0(\tilde{Y}_i), i = 1, 2, \dots, n$$

serve as samples drawn from the target distribution  $\nu$ , where particles  $\{\tilde{Y}_i\}_{i=1}^n \subset \mathbb{R}^m$  are sampled from a simple reference distribution  $\mu$ . The pseudocode of the basic EPT algorithm is given in Algorithm 2 in Appendix A.1.

In many applications, high-dimensional complex data such as images, texts and natural languages, tend to be supported on lower-dimensional manifolds. To learn generative models with latent low-dimensional structures, it is beneficial to have the option of first sampling particles  $\{Z_i\}_{i=1}^n$  from a low-dimensional reference distribution  $\tilde{\mu} \in \mathcal{P}_2(\mathbb{R}^\ell)$  with  $\ell \ll m$ . For this purpose, we train a generator  $G_\theta : \mathbb{R}^\ell \rightarrow \mathbb{R}^m$  together with the EPT map  $\mathcal{T}$ , where  $G_\theta$  is a neural network with parameter  $\theta$ . The generator  $G_\theta$  and the EPT map  $\mathcal{T}$  are trained iteratively as follows:

- (a) Begin outer loop: given an initial  $G_\theta$ , compute  $\tilde{Y}_i \leftarrow G_\theta(\tilde{Z}_i)$ , where  $\tilde{Z}_i \sim \tilde{\mu}, i = 1, 2, \dots, n$ ;
- (b) Inner loop: update the particles  $\{\tilde{Y}_i\}_{i=1}^n$  by iteratively using the EPT updating steps (1)-(3);
- (c) End outer loop: update  $G_\theta$  by minimizing the least squares

$$G_\theta \leftarrow \arg \min_{G_\theta} \frac{1}{n} \sum_{i=1}^n \|G_\theta(\tilde{Z}_i) - \tilde{Y}_i\|_2^2 \text{ via SGD.}$$

---

**Algorithm 1:** Euler particle transport with an optional outer loop
 

---

```

Input:  $K_I, K_O \in \mathbb{N}^*, s > 0, \ell, \alpha > 0$  // maximum inner loop count, maximum outer
loop count, step size, dimension of the reference distribution,
regularization parameter
 $X_i \sim \nu, i = 1, 2, \dots, n$  // real samples
 $\widehat{G}_\theta^0 \leftarrow G_\theta^{init}$  // initialize the transport map
 $j \leftarrow 0$ 
/* outer loop */
while  $j < K_O$  do
     $Z_i^j \sim \tilde{\mu}, i = 1, 2, \dots, n$  // latent particles
     $\tilde{Y}_i^0 = \widehat{G}_\theta^j(Z_i^j), i = 1, 2, \dots, n$  // intermediate particles
     $k \leftarrow 0$ 
    /* inner loop */
    while  $k < K_I$  do
         $\widehat{R}_\phi^k \in \arg \min_{R_\phi} \frac{1}{n} \sum_{i=1}^n [R_\phi(X_i)^2 + \alpha \|\nabla R_\phi(X_i)\|_2^2 - 2R_\phi(\tilde{Y}_i^k)]$  via SGD
        // estimate density ratio
         $\hat{\mathbf{v}}^k(\mathbf{x}) = -f''(\widehat{R}_\phi^k(\mathbf{x})) \nabla \widehat{R}_\phi^k(\mathbf{x})$  // approximate velocity fields
         $\widehat{\mathcal{T}}^k = \mathbb{1} + s \hat{\mathbf{v}}^k$  // compute forward Euler maps
         $\tilde{Y}_i^{k+1} = \widehat{\mathcal{T}}^k(\tilde{Y}_i^k), i = 1, 2, \dots, n$  // update particles
         $k \leftarrow k + 1$ 
    end
     $\widehat{G}_\theta^{j+1} \in \arg \min_{G_\theta} \frac{1}{n} \sum_{i=1}^n \|G_\theta(Z_i^j) - \tilde{Y}_i^{K_I}\|_2^2$  via SGD // fit the transport map
     $j \leftarrow j + 1$ 
end
Output:  $\widehat{G}_\theta^{K_O} : \mathbb{R}^\ell \rightarrow \mathbb{R}^d$  // transport map with a latent structure
    
```

---

Then we iterate steps (a)-(c) until the maximum number of iterations is reached. The purpose of step (c) is to preserve the information about the trained particle transport in the generator  $G_\theta$ . In addition to being able to start from a lower-dimensional reference distribution, a second benefit of using a generator  $G_\theta$  in the outer loop is that it memorizes the composition of the residual maps trained in EPT. After this generator  $G_\theta$  is well trained, it can then be used to directly transform new samples from the reference distribution to the target without the need for additional training. The pseudocode of the EPT algorithm with an outer loop (an additional generator  $G_\theta$ ) is given in Algorithm 1.

## 6. Related work

The existing generative models, such as VAEs, GANs and flow-based methods, parameterize a transform map with a neural network, say  $G$ , that solves

$$\min_G \mathfrak{D}(G_{\#}\mu, \nu), \quad (28)$$

where  $\mathfrak{D}(\cdot, \cdot)$  is an integral probability discrepancy. The original GAN (Goodfellow et al., 2014),  $f$ -GAN (Nowozin et al., 2016) and WGAN (Arjovsky et al., 2017) solve the dual form of (28) by

parameterizing the dual variable using another neural network with  $\mathfrak{D}$  as the Jensen–Shannon (JS) divergence, the  $f$ -divergence and the 1-Wasserstein distance, respectively. SWGAN (Deshpande et al., 2018) and MMDGAN (Li et al., 2017; Binkowski et al., 2018) use the sliced quadratic Wasserstein distance and the maximum mean discrepancy (MMD) as  $\mathfrak{D}$ , respectively.

Vanilla VAE (Kingma and Welling, 2014) approximately solves the primal form of (28) with the Kullback–Leibler (KL) divergence. Several authors have proposed methods that use optimal transport losses, such as various forms of Wasserstein distances between the distribution of learned latent codes and the prior distribution as the regularizer in VAE to improve performance. These methods include WAE (Tolstikhin et al., 2018), Sliced WAE (Kolouri et al., 2019) and Sinkhorn AE (Patrini et al., 2019).

Discrete time flow-based methods minimize (28) with the KL divergence loss (Rezende and Mohamed, 2015; Dinh et al., 2017; Kingma et al., 2016; Kingma and Dhariwal, 2018). Grathwohl et al. (2019) proposed an ODE flow approach for fast training in such methods using the adjoint equation (Chen et al., 2018b). By introducing the optimal transport tools into maximum likelihood training, Chen et al. (2018a) and Zhang et al. (2018) considered continuous time flows. Chen et al. (2018a) proposed a gradient flow in measure spaces in the framework of variational inference and then discretized it with the implicit movement minimizing scheme (De Giorgi, 1993; Jordan et al., 1998). Zhang et al. (2018) considered gradient flows in measure spaces with time invariant velocity fields. CFGGAN (Johnson and Zhang, 2018) derived from the perspective of optimization in the functional space is a special form of EPT with the energy functional taken as the KL divergence. SW flow (Liutkus et al., 2019) and MMD flow (Arbel et al., 2019) are gradient flows in measure spaces. MMD flow can be recovered from EPT by first choosing  $\mathcal{L}[\cdot]$  as the Lebesgue norm and then projecting the corresponding velocity fields onto reproducing kernel Hilbert spaces. However, neither SW flow nor MMD flow can model hidden low-dimensional structures with the particle sampling procedure.

SVGD in (Liu, 2017) and the proposed EPT are both particle methods based on gradient flow in measure spaces. However, SVGD samples from an unnormalized density, while EPT focuses on generative learning, i.e., learning the distribution from samples. At the population level, projecting the velocity fields of EPT with the KL divergence onto reproducing kernel Hilbert spaces will recover the velocity fields of SVGD. The proof is given in Appendix B.3. Score-based methods in (Song and Ermon, 2019, 2020; Ho et al., 2020) are also particle methods based on the unadjusted Langevin flow and deep score estimators. At the population level, the velocity fields of these score-based methods are random since they have a Brownian motion term, while the velocity fields of EPT are deterministic. At the sample level, these score-based methods need to learn a vector-valued deep score function, while in EPT we only need to estimate the density ratios which are scalar functions.

## 7. Numerical experiments

The implementation details on numerical settings, network structures, SGD optimizers and hyper-parameters are given in the appendix. All experiments are performed using NVIDIA Tesla K80 GPUs. The PyTorch code of EPT is available at <https://github.com/xjtuygao/EPT>.

### 7.1. 2D simulated data

We use EPT to learn 2D distributions adapted from Grathwohl et al. (2019) with multiple modes and density ridges. The first row in Figure 1 shows kernel density estimation (KDE) plots of 50k samples

from target distributions including (from left to right) *8Gaussians*, *pinwheel*, *moons*, *checkerboard*, *2spirals*, and *circles*. The second and third rows show the KDE plots of the learned samples via EPT with Pearson  $\chi^2$ -divergence and the surface plots of estimated density ratios after 20k iterations. The fourth and fifth rows show the KDE plots of the learned sample via EPT with Lebesgue norm of the density difference. Clearly, the generated samples via EPT are nearly indistinguishable from those of the target samples and the estimated density-ratio/ difference functions are approximately equal to 1/0, indicating the learnt distributions matches the targets well.

Next, we demonstrate the effectiveness of using the gradient penalty (26) by visualizing the transport maps learned in the generative learning tasks with the learning targets *5squares* and *large4gaussians* from *4squares* and *small4gaussians*, respectively. We use 200 particles connected with grey lines to manifest the learned transport maps. As shown in Figure 2(a), the central squares of *5squares* were learned better with the gradient penalty, which is consistent with the result on the estimated density-ratio in Figure 2(b). For *large4gaussians*, the learned transport map exhibited some optimality under quadratic Wasserstein distance due to the obvious correspondence between the samples in Figure 2(a), and the gradient penalty also improves the density-ratio estimation as expected.

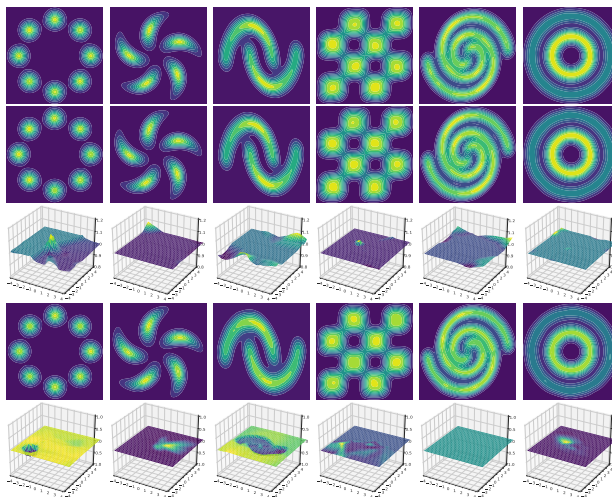


Figure 1: KDE plots and 3D surface plots for 2D distributions. KDE plots of the target samples are presented in the first row. The second and third rows show the KDE plots of the learned samples via EPT with Pearson  $\chi^2$ -divergence and the 3D surface plots of estimated density ratios after 20k iterations. The fourth and fifth rows show the KDE plots of the learned sample via EPT with Lebesgue norm of the density difference after 20k iterations.

## 7.2. Numerical convergence

We illustrate the convergence property of the learning dynamics of EPT on synthetic datasets *pinwheel*, *checkerboard* and *2spirals*. As shown in Figure 3, on the three test datasets, the dynamics of both the estimated LSDR fitting losses in (27) with  $\alpha = 0$  and the estimated value of the gradient norms  $\mathbb{E}_{X \sim q_k} [\|\nabla R_\phi(X)\|_2]$  demonstrate the estimated LSDR loss converges to the theoretical value  $-1$ .

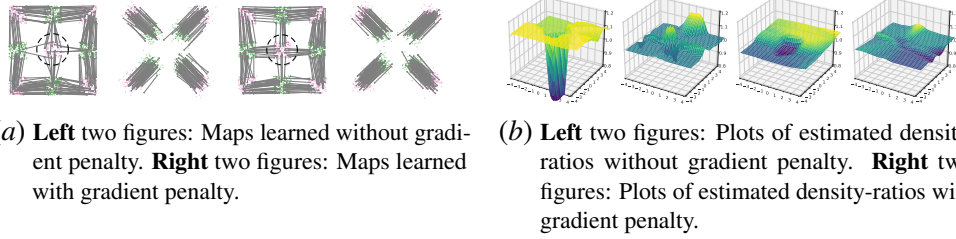


Figure 2: Learned transport maps and estimated density-ratio in learning  $5squares$  from  $4squares$ , and learning  $large4gaussians$  from  $small4gaussians$ .

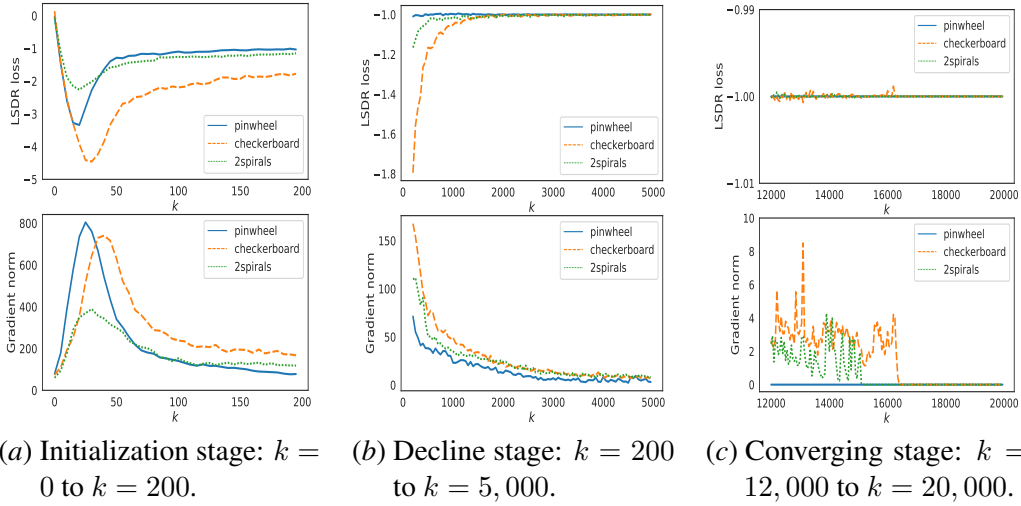


Figure 3: The numerical convergence of EPT on simulated datasets. First row: LSDR loss (27) with  $\alpha = 0$  v.s. iterations on *pinwheel*, *checkerboard* and *2spirals*. Second row: Estimation of the gradient norm  $\mathbb{E}_{X \sim q_k} [\|\nabla R_\phi(X)\|_2]$  v.s. iterations on *pinwheel*, *checkerboard* and *2spirals*.

### 7.3. Benchmark Image Data

Finally, we show the performance of applying EPT to benchmark data MNIST (LeCun et al., 1998), CIFAR10 (Krizhevsky and Hinton, 2009) and CelebA (Liu et al., 2015) using ReLU ResNets without batch normalization and spectral normalization. The particle evolutions on MNIST and CIFAR10 without using the outer loop (see Algorithm 2 in Appendix A.1) are shown in Figure 4. Clearly, EPT can transport samples from a multivariate normal distribution into a target distribution.

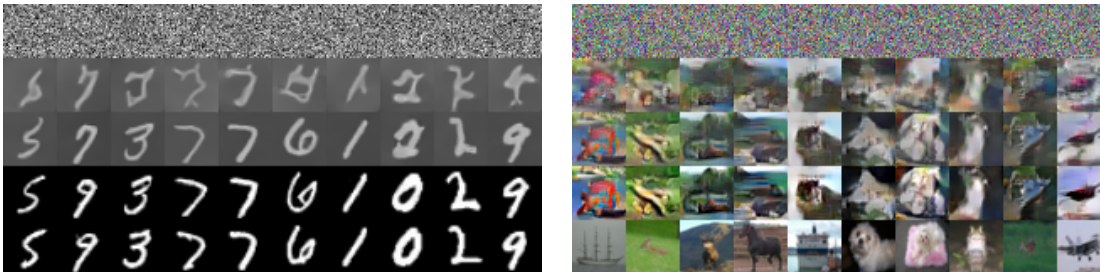


Figure 4: Particle evolution of EPT on MNIST and CIFAR10 datasets. As shown in the first row, initial particles are sampled from a standard normal distribution. The following three rows show the particle evolution process. The most similar real data (measured with the Frobenius norm) are also presented for the corresponding generated particles in the last row.

We further compare EPT using the outer loop with the generative models including WGAN, SNGAN and MMDGAN. We considered different  $f$ -divergences, including Pearson  $\chi^2$ , KL, JS and logD (Gao et al., 2019) and different deep density-ratio fitting methods (LSDR and LR in Section 4). Table 1 shows FID (Heusel et al., 2017) evaluated with five bootstrap sampling of EPT with four divergences on CIFAR10. We can see that EPT using ReLU ResNets without batch normalization and spectral normalization attains (usually better) comparable FID scores with the state-of-the-art generative models. Comparisons of the real samples and learned samples on MNIST, CIFAR10 and CelebA are shown in Figure 5, where high-fidelity learned samples are comparable to real samples visually.

Table 1: Mean (standard deviation) of FID scores on CIFAR10. The FID score of NSCN is reported in Song and Ermon (2019) and results in the right table are adapted from Arbel et al. (2018).

Models	CIFAR10 (50k)	Models	CIFAR10 (50k)
EPT-LSDR- $\chi^2$	<b>24.9 (0.1)</b>	WGAN-GP	31.1 (0.2)
EPT-LR-KL	25.9 (0.1)	MMDGAN-GP-L2	31.4 (0.3)
EPT-LR-JS	25.3 (0.1)	SN-GAN	26.7 (0.2)
EPT-LR-logD	<b>24.6 (0.1)</b>	SN-SMMDGAN	<b>25.0 (0.3)</b>
NCSN	25.3		





Figure 5: Visual comparisons between real images (left 3 panels) and generated images (right 3 panels) by EPT-LSDR- $\chi^2$  on MNIST, CIFAR10 and CelebA datasets.

## 8. Conclusion and future work

EPT is a new approach for generative learning via training a transport map that pushes forward a reference to the target. Because EPT is easy to train, computationally stable, and enjoys strong theoretical guarantees, we expect it to be a useful addition to the methods for generative learning. There are two important ingredients in EPT: the velocity field and density-ratio estimation. With a suitable choice of the velocity and a density-ratio estimation procedure, EPT can recover several existing generative models such as MMD flow and SVGD. Thus our theoretical results also provide insights into the properties of these methods. Simulation results on multi-mode synthetic datasets and comparisons with the existing methods on real benchmark datasets using simple ReLU ResNets support our theoretical analysis and demonstrate the effectiveness of the proposed method.

Some aspects and results in this paper are of independent interest. For example, density-ratio estimation is an important problem and of general interest in machine learning and statistics. The estimation error bound established in Theorem 11 for the nonparametric deep density-ratio fitting procedure is new. We show that the proposed density-ratio estimators do not suffer from the “*curse of dimensionality*” if data is supported on a lower-dimensional manifold. This provides an important example showing that deep nonparametric estimation can circumvent the curse of dimensionality via exploring the underlying structure of the data.

EPT is motivated from the Monge-Ampère equation that characterizes the optimal transport map. As we described in Subsection 3.1, EPT solves a sequence of linearized versions of the Monge-Ampère equation (8), but not the Monge-Ampère equation itself. The transport maps learned with EPT are not unique, since different residual maps used for linearization will lead to different gradient flows. However, they all push forward the reference distribution to the target, albeit along different

gradient flows. How to consistently estimate the unique Monge-Ampère optimal map when only a random sample from the target distribution is available remains a challenging and open problem.

## Acknowledgments

J. Huang is partially supported by the U.S. NSF grant DMS-1916199. Y. Jiao is supported in part by the National Science Foundation of China under Grant 11871474 and by the research fund of KLATASDSMOE of China. J. Liu is supported by the grants MOE2018-T2-1-046 and MOE2018-T2-2-006 from the Ministry of Education, Singapore. X. Lu is partially supported by the National Science Foundation of China (No. 11871385), the National Key Research and Development Program of China (No.2018YFC1314600) and the Natural Science Foundation of Hubei Province (No. 2019CFA007), and by the research fund of KLATASDSMOE of China. Z. Yang is supported by National Science Foundation of China (No. 12071362 and 11671312), the National Key Research and Development Program of China (No. 2020YFA0714200), the Natural Science Foundation of Hubei Province (No. 2019CFA007). The numerical studies of this work were done on the supercomputing system in the Supercomputing Center of Wuhan University.

## References

- Syed Mumtaz Ali and Samuel D Silvey. A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28 (1):131–142, 1966.
- Luigi Ambrosio, Nicola Gigli, and Giuseppe Savaré. *Gradient flows: in metric spaces and in the space of probability measures*. Springer Science & Business Media, 2008.
- Martin Anthony and Peter L Bartlett. *Neural Network Learning: Theoretical Foundations*. Cambridge University Press, 2009.
- Michael Arbel, Dougal Sutherland, Mikolaj Binkowski, and Arthur Gretton. On gradient regularizers for MMD GANs. In *NIPS*, 2018.
- Michael Arbel, Anna Korba, Adil Salim, and Arthur Gretton. Maximum mean discrepancy gradient flow. In *NeurIPS*, 2019.
- Martin Arjovsky and Léon Bottou. Towards principled methods for training generative adversarial networks. In *ICLR*, 2017.
- Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *ICML*, 2017.
- Vladimir Igorevich Arnold. *Geometrical Methods in the Theory of Ordinary Differential Equations*, volume 250. Springer Science & Business Media, 2012.
- Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

- Peter L Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20:1–17, 2019.
- Mikolaj Binkowski, Dougal J Sutherland, Michael Arbel, and Arthur Gretton. Demystifying MMD GANs. In *ICLR*, 2018.
- Yann Brenier. Polar factorization and monotone rearrangement of vector-valued functions. *Communications on Pure and Applied Mathematics*, 44(4):375–417, 1991.
- Changyou Chen, Chunyuan Li, Liqun Chen, Wenlin Wang, Yunchen Pu, and Lawrence Carin Duke. Continuous-time flows for efficient inference and density estimation. In *ICML*, 2018a.
- Tian Qi Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. In *NIPS*, 2018b.
- Frank H Clarke. *Optimization and Nonsmooth Analysis*, volume 5. SIAM, 1990.
- A Philip Dawid. The geometry of proper scoring rules. *Annals of the Institute of Statistical Mathematics*, 59(1):77–93, 2007.
- E De Giorgi. New problems on minimizing movements, boundary value problems for partial differential equations. *Results in Applied Mathematics*, 29:81–98, 1993.
- Ishan Deshpande, Ziyu Zhang, and Alexander G Schwing. Generative modeling using the sliced wasserstein distance. In *CVPR*, 2018.
- Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. In *ICLR*, 2015.
- Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using Real NVP. In *ICLR*, 2017.
- Yuan Gao, Yuling Jiao, Yang Wang, Yao Wang, Can Yang, and Shunkang Zhang. Deep generative learning via variational gradient flow. In *ICML*, 2019.
- Izrail Moiseevitch Gelfand and Sergei Vasilevich Fomin. *Calculus of Variations*. Dover Publications, 2000.
- Tilmann Gneiting and Adrian E Raftery. Strictly proper scoring rules, prediction, and estimation. *Journal of the American statistical Association*, 102(477):359–378, 2007.
- Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.
- Will Grathwohl, Ricky TQ Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *ICLR*, 2019.
- Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*, 2017.

- Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner.  $\beta$ -VAE: Learning basic visual concepts with a constrained variational framework. In *ICLR*, 2017.
- Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020.
- Richard Holley and Daniel Stroock. Logarithmic sobolev inequalities and stochastic ising models. *Journal of Statistical Physics*, 46:1159–1194, 1987.
- Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015.
- Rie Johnson and Tong Zhang. Composite functional gradient learning of generative adversarial models. In *ICML*, 2018.
- Richard Jordan, David Kinderlehrer, and Felix Otto. The variational formulation of the Fokker-Planck equation. *SIAM Journal on Mathematical Analysis*, 29(1):1–17, 1998.
- Takafumi Kanamori and Masashi Sugiyama. Statistical analysis of distance estimators with density differences and density ratios. *Entropy*, 16(2):921–942, 2014.
- Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014.
- Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *NIPS*, 2018.
- Durk P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *NIPS*, 2016.
- Soheil Kolouri, Phillip E Pope, Charles E Martin, and Gustavo K Rohde. Sliced-Wasserstein autoencoder: An embarrassingly simple generative model. In *ICLR*, 2019.
- Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.
- Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- John Lee. *Introduction to Riemannian Manifolds*. Springer, 2010.
- Chun-Liang Li, Wei-Cheng Chang, Yu Cheng, Yiming Yang, and Barnabás Póczos. MMD GAN: Towards deeper understanding of moment matching network. In *NIPS*, 2017.
- Yujia Li, Kevin Swersky, and Rich Zemel. Generative moment matching networks. In *ICML*, 2015.
- Qiang Liu. Stein variational gradient descent as gradient flow. In *Advances in Neural Information Processing Systems*, volume 30, pages 3115–3123. Curran Associates, Inc., 2017.
- Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

- Antoine Liutkus, Umut Simsekli, Szymon Majewski, Alain Durmus, Fabian-Robert Stöter, Kamalika Chaudhuri, and Ruslan Salakhutdinov. Sliced-Wasserstein flows: Nonparametric generative modeling via optimal transport and diffusions. In *ICML*, 2019.
- Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, and Ian Goodfellow. Adversarial autoencoders. In *ICLR Workshop*, 2016.
- Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *ICCV*, 2017.
- Robert J. McCann. Existence and uniqueness of monotone measure-preserving maps. *Duke Mathematical Journal*, 80(2):309–324, 1995.
- Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *ICLR*, 2018.
- Shakir Mohamed and Balaji Lakshminarayanan. Learning in implicit generative models. *arXiv preprint arXiv:1610.03483*, 2016.
- Youssef Mroueh and Tom Sercu. Fisher GAN. In *NIPS*, 2017.
- Sebastian Nowozin, Botond Cseke, and Ryota Tomioka.  $f$ -GAN: Training generative neural samplers using variational divergence minimization. In *NIPS*, 2016.
- F. Otto and C. Villani. Generalization of an inequality by talagrand and links with the logarithmic sobolev inequality. *Journal of Functional Analysis*, 173:261–400, 2000.
- George Papamakarios, Theo Pavlakou, and Iain Murray. Masked autoregressive flow for density estimation. In *NIPS*, 2017.
- Giorgio Patrini, Samarth Bhargav, Rianne van den Berg, Max Welling, Patrick Forré, Tim Genewein, Marcello Carioni, KFU Graz, Frank Nielsen, and CSL Sony. Sinkhorn autoencoders. In *UAI*, 2019.
- Scott Reed, Zeynep Akata, Xinchun Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. In *ICML*, 2016.
- Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *ICML*, 2015.
- Kevin Roth, Aurelien Lucchi, Sebastian Nowozin, and Thomas Hofmann. Stabilizing training of generative adversarial networks through regularization. In *NIPS*, 2017.
- Ruslan Salakhutdinov. Learning deep generative models. *Annual Review of Statistics and Its Application*, 2:361–385, 2015.
- Filippo Santambrogio. *Optimal Transport for Applied Mathematicians*. Springer, 2015.
- Johannes Schmidt-Hieber. Nonparametric regression using deep neural networks with relu activation function. *The Annals of Statistics*, in press, 2020.

- Zuwei Shen, Haizhao Yang, and Shijun Zhang. Deep network approximation characterized by number of neurons. *arXiv preprint arXiv:1906.05497*, 2019.
- Casper Kaae Sønderby, Jose Caballero, Lucas Theis, Wenzhe Shi, and Ferenc Huszár. Amortised map inference for image super-resolution. In *ICLR*, 2017.
- Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.
- Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems*, volume 33. Curran Associates, Inc., 2020.
- Charles J. Stone. Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics*, 10(4):1040–1053, 1982.
- Masashi Sugiyama, Takafumi Kanamori, Taiji Suzuki, Marthinus D Plessis, Song Liu, and Ichiro Takeuchi. Density-difference estimation. In *NIPS*, 2012a.
- Masashi Sugiyama, Taiji Suzuki, and Takafumi Kanamori. *Density ratio estimation in machine learning*. Cambridge University Press, 2012b.
- Dougal J Sutherland, Hsiao-Yu Tung, Heiko Strathmann, Soumyajit De, Aaditya Ramdas, Alex Smola, and Arthur Gretton. Generative models and model criticism via optimized maximum mean discrepancy. In *ICLR*, 2017.
- Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.
- Chenyang Tao, Liqun Chen, Ricardo Henao, Jianfeng Feng, and Lawrence Carin Duke. Chi-square generative adversarial network. In *ICML*, 2018.
- I Tolstikhin, O Bousquet, S Gelly, and B Schölkopf. Wasserstein auto-encoders. In *ICLR*, 2018.
- Roman Vershynin. *High-dimensional Probability: An Introduction with Applications in Data Science*, volume 47. Cambridge university press, 2018.
- Cédric Villani. *Optimal Transport: Old and New*, volume 338. Springer Science & Business Media, 2008.
- Linfeng Zhang, Weinan E, and Lei Wang. Monge-Ampère flow for generative modeling. *arXiv preprint arXiv:1809.10188*, 2018.
- Shunkang Zhang, Yuan Gao, Yuling Jiao, Jin Liu, Yang Wang, and Can Yang. Wasserstein-Wasserstein auto-encoders. *arXiv preprint arXiv:1902.09323*, 2019.
- Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

---

**Algorithm 2:** EPTv1: Euler particle transport

---

```

Input:  $K_I \in \mathbb{N}^*$ ,  $s > 0$ ,  $\alpha > 0$  // maximum loop count, step size,
regularization coefficient
 $X_i \sim \nu$ ,  $\tilde{Y}_i^0 \sim \mu$ ,  $i = 1, 2, \dots, n$  // real samples, initial particles
 $k \leftarrow 0$ 
while  $k < K_I$  do
     $\hat{R}_\phi^k \in \arg \min_{R_\phi} \frac{1}{n} \sum_{i=1}^n [R_\phi(X_i)^2 + \alpha \|\nabla R_\phi(X_i)\|_2^2 - 2R_\phi(\tilde{Y}_i^k)]$  via SGD
    // determine the density ratio
     $\hat{\mathbf{v}}^k(\mathbf{x}) = -f''(\hat{R}_\phi^k(\mathbf{x})) \nabla \hat{R}_\phi^k(\mathbf{x})$  // approximate velocity fields
     $\hat{\mathcal{T}}^k = \mathbb{1} + s \hat{\mathbf{v}}^k$  // define the forward Euler map
     $\tilde{Y}_i^{k+1} = \hat{\mathcal{T}}^k(\tilde{Y}_i^k)$ ,  $i = 1, 2, \dots, n$  // update particles
     $k \leftarrow k + 1$ 
end
Output:  $\tilde{Y}_i^{K_I} \sim \tilde{\mu}_{K_I}$ ,  $i = 1, 2, \dots, n$  // transported particles

```

---

**APPENDIX**

In the appendix, we provide the implementation details on numerical settings, network structures, SGD optimizers, and hyper-parameters in the paper. We give the proofs of the results in Sections 3 to 4. We also show that SVGD can be derived from EPT by choosing an appropriate  $f$ -divergence.

**Appendix A. Implementation details of numerical experiments**

**A.1. Algorithm details**

We present the details of the basic EPT algorithm without the outer loop in Algorithm 2.

**A.2. Additional figures**

We provide additional figures on simulated data in this part. Figure A1 includes 2D surface plots to show the efficiency of deep density-ratio (density-difference) fitting.

**A.3. Implementation details, network structures, hyper-parameters**

A.3.1. 2D EXAMPLES

Experiments on 2D examples in our work were performed with deep LSDR fitting and the Pearson  $\chi^2$  divergence. We use the EPTv1 (Algorithm 2) without outer loops. In inner loops, only a multilayer perceptron (MLP) was utilized for dynamic estimation of the density ratio between the model distribution  $q_k$  and the target distribution  $p$ . The network structure and hyper-parameters in EPT and deep LSDR fitting were shared in all 2D experiments. We adopt EPT to push particles from a pre-drawn pool consisting of 50k i.i.d. Gaussian particles to evolve in 20k steps. We used RMSProp with the learning rate 0.0005 and the batch size 1k as the SGD optimizer. The details are given in Table A1 and Table A2. We note that  $s$  is the step size,  $n$  is the number of particles,  $\alpha$  is the penalty coefficient, and  $T$  is the mini-batch gradient descent times of deep LSDR fitting or deep logistic regression in each inner loop hereinafter.  $K_I$  indicates the maximum (inner) loop count of EPTv1.

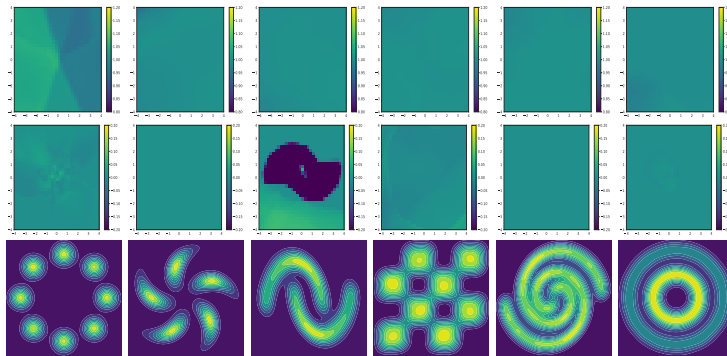


Figure A1: 2D surface plots of the estimated density ratio (the first row) and the estimated density difference (the second row) after 20k EPT iterations and KDE plots of the corresponding simulated data (the last row).

Table A1: MLP for deep LSDR fitting.

Layer	Details	Output size
1	Linear, ReLU	64
2	Linear, ReLU	64
3	Linear, ReLU	64
4	Linear	1

### A.3.2. REAL IMAGE DATA

**Datasets.** We evaluated EPT on three benchmark datasets including two small datasets MNIST, CIFAR10 and one large dataset CelebA from GAN literature. MNIST contains a training set of 60k examples and a test set of 10k examples as  $28 \times 28$  bilevel images which were resized to  $32 \times 32$  resolution. There are a training set of 50k examples and a test set of 10k examples as  $32 \times 32$  color images in CIFAR10. We randomly divided the 200k celebrity images in CelebA into two sets for training and test according to the ratio 9:1. We also pre-processed CelebA images by first taking a  $160 \times 160$  central crop and then resizing to the  $64 \times 64$  resolution. Only the training sets are used to train our models.

**Evaluation metrics.** *Fréchet Inception Distance* (FID) (Heusel et al., 2017) computes the Wasserstein distance  $\mathcal{W}_2$  with summary statistics (mean  $\mu$  and variance  $\Sigma$ ) of real samples  $\mathbf{x}s$  and generated samples  $\mathbf{g}s$  in the feature space of the Inception-v3 model (Szegedy et al., 2016), i.e.,  $\text{FID} = \|\mu_{\mathbf{x}} - \mu_{\mathbf{g}}\|_2^2 + \text{Tr}(\Sigma_{\mathbf{x}} + \Sigma_{\mathbf{g}} - 2(\Sigma_{\mathbf{x}}\Sigma_{\mathbf{g}})^{\frac{1}{2}})$ . Here, FID is reported with the TensorFlow implementation and lower FID is better.

**Network architectures and hyper-parameter settings.** We employed the ResNet architectures used by Gao et al. (2019) in our EPT algorithm. Especially, the batch normalization (Ioffe and Szegedy, 2015) and the spectral normalization (Miyato et al., 2018) of networks were omitted for EPT-LSDR- $\chi^2$ . To train neural networks, we set SGD optimizers as RMSProp with the learning



Table A2: Hyper-parameters in EPT on 2D examples.

Parameter	$s$	$n$	$\alpha$	$T$	$K_I$
Value	0.005	50k	0 or 0.5	5	20k

rate 0.0001 and the batch size 100. Inputs  $\{Z_i\}_{i=1}^n$  in EPT (Algorithm 1) were vectors generated from a 128-dimensional standard normal distribution on all three datasets. Hyper-parameters are listed in Table A3 where  $K_I$  expresses the maximum inner loop count in each outer loop and  $K_O$  shows the maximum outer loop count. Even without outer loops, EPTv1 (Algorithm 2) can generate images on MNIST and CIFAR10 as well by making use of a large set of particles. Table A4 shows the hyper-parameters.

Table A3: Hyper-parameters in EPT **with** outer loops on real image datasets.

Parameter	$\ell$	$s$	$n$	$\alpha$	$T$	$K_I$	$K_O$
Value	128	0.5	1k	0	1	20	10k

Table A4: Hyper-parameters in EPT **without** outer loops on real image datasets.

Parameter	$s$	$n$	$\alpha$	$T$	$K_I$
Value	0.5	4k	0	5	10k

#### A.4. Learning and inference

The learning process of EPT performs particle evolution via solving the McKean-Vlasov equation using forward Euler iterations. The iterations rely on the estimation of the density ratios (difference) between the push-forward distributions and the target distribution. To make the inference of EPTv1 more amendable, we propose EPT based on EPTv1. EPT takes advantage of a neural network to fit the pushforward map. The inference of EPT is fast since the pushforward map is parameterized as a neural network and only forward propagation is involved. These aspects distinguish EPT from score-based generative models (Song and Ermon, 2019, 2020) which simulate Langevin dynamics to generate samples.

## Appendix B. Proofs

### B.1. Proofs of the results in Section 3

#### B.1.1. PROOF OF PROPOSITION 2.

(i) The continuity equation (20) follows from the definition of the gradient flow directly, see, page 281 in (Ambrosio et al., 2008). (ii) The first equality follows from the chain rule and integration

by part, see, Theorem 24.2 of [Villani \(2008\)](#). The second one on linear convergence follows from Theorem 24.7 of [Villani \(2008\)](#), where the assumption on  $\lambda$  in equation (24.6) is equivalent to the  $\lambda$ -geodetically convex assumption here. (iii) Similar to (i) see, page 281 in [Ambrosio et al. \(2008\)](#). ■

### B.1.2. PROOF OF THEOREM 5.

(i) Recall  $\mathcal{L}[\mu]$  is a functional on  $\mathcal{P}_2^a(\mathbb{R}^m)$ . By the classical results in calculus of variation ([Gelfand and Fomin, 2000](#)),

$$\frac{\partial \mathcal{L}[q]}{\partial q}(\mathbf{x}) = \frac{d}{dt} \mathcal{L}[q + tg] \Big|_{t=0} = F'(q(\mathbf{x})),$$

where  $\frac{\partial \mathcal{L}[q]}{\partial q}$  denotes the first order of variation of  $\mathcal{L}[\cdot]$  at  $q$ , and  $q, g$  are the densities of  $\mu$  and an arbitrary  $\xi \in \mathcal{P}_2^a(\mathbb{R}^m)$ , respectively. Let

$$L_F(z) = zF'(z) - F(z) : \mathbb{R}^1 \rightarrow \mathbb{R}^1.$$

Some algebra shows,

$$\nabla L_F(q(\mathbf{x})) = q(\mathbf{x}) \nabla F'(q(\mathbf{x})).$$

Then, it follows from Theorem 10.4.6 in ([Ambrosio et al., 2008](#)) that

$$\nabla F'(q(\mathbf{x})) = \partial^\circ L(\mu),$$

where,  $\partial^\circ L(\mu)$  denotes the one in  $\partial L(\mu)$  with minimum length. The above display and the definition of gradient flow implies the representation of the velocity fields  $\mathbf{v}_t$ .

(ii) The time dependent form of (16)-(17) reads

$$\begin{aligned} \frac{d\mathbf{x}_t}{dt} &= \nabla \Phi_t(\mathbf{x}_t), \quad \text{with } \mathbf{x}_0 \sim q, \\ \frac{d \ln q_t(\mathbf{x}_t)}{dt} &= -\Delta \Phi_t(\mathbf{x}_t), \quad \text{with } q_0 = q. \end{aligned}$$

By chain rule and substituting the first equation into the second one, we have

$$\begin{aligned} \frac{1}{q_t} \left( \frac{dq_t}{dt} + \frac{dq_t}{d\mathbf{x}_t} \frac{d\mathbf{x}_t}{dt} \right) &= \frac{1}{q_t} \left( \frac{dq_t}{dt} + \nabla q_t \nabla \Phi_t(\mathbf{x}_t) \right) \\ &= -\Delta \Phi_t(\mathbf{x}_t), \end{aligned}$$

which implies,

$$\frac{dq_t}{dt} = -q_t \Delta \Phi_t(\mathbf{x}_t) - \nabla q_t \nabla \Phi_t(\mathbf{x}_t) = -\nabla \cdot (q_t \nabla \Phi_t).$$

By (22), the above display coincides with the continuity equation (20) with  $\mathbf{v}_t = \nabla \Phi_t = -\nabla F'(q_t(\mathbf{x}))$ . ■

### B.1.3. PROOF OF THEOREM 6.

The Lipschitz assumption of  $\mathbf{v}_t$  implies the existence and uniqueness of the McKean-Vlasov equation (10) according to the classical results in ODE ([Arnold, 2012](#)). By the uniqueness of the continuity equation, see Proposition 8.1.7 in [Ambrosio et al. \(2008\)](#), it is sufficient to show that  $\mu_t = (\mathbf{X}_t)_\# \mu$  satisfies the continuity equation (20) in a weak sense. This can be done by the standard test function and smoothing approximation arguments, see, Theorem 4.4 in [Santambrogio \(2015\)](#) for details. ■

## B.1.4. PROOF OF LEMMA 7

By definition,

$$F(q_t(\mathbf{x})) = \begin{cases} p(\mathbf{x})f\left(\frac{q_t(\mathbf{x})}{p(\mathbf{x})}\right), & \mathcal{L}[\mu] = \mathbb{D}_f(\mu|\nu), \\ (q_t(\mathbf{x}) - p(\mathbf{x}))^2, & \mathcal{L}[\mu] = \|\mu - \nu\|_{L^2(\mathbb{R}^m)}^2. \end{cases}$$

Direct calculation shows

$$F'(q_t(\mathbf{x})) = \begin{cases} f'\left(\frac{q_t(\mathbf{x})}{p(\mathbf{x})}\right), & \mathcal{L}[\mu] = \mathbb{D}_f(\mu|\nu), \\ 2(q_t(\mathbf{x}) - p(\mathbf{x})), & \mathcal{L}[\mu] = \|\mu - \nu\|_{L^2(\mathbb{R}^m)}^2. \end{cases}$$

Then, the desired result follows from the above display and (22). ■

## B.1.5. PROOF OF PROPOSITION 8.

Without loss of generality let  $K = \frac{T}{s} > 1$  be an integer. Recall  $\{\mu_t^s \mid t \in [ks, (k+1)s)\}$  is the piecewise constant interpolation between  $\mu_k$  and  $\mu_{k+1}$  defined as

$$\mu_t^s = (\mathcal{T}_t^{k,s})_{\#}\mu_k,$$

where,

$$\mathcal{T}_t^{k,s} = \mathbf{1} + (t - ks)\mathbf{v}_k,$$

$\mu_k$  is defined in (16)-(18) with  $\mathbf{v}_k = \mathbf{v}_{ks}$ , i.e., the continuous velocity in (22) at time  $ks$ ,  $k = 0, \dots, K-1$ ,  $\mu_0 = \mu$ . Under assumption (24) we can first show in a way similar to the proof of Lemma 10 in Arbel et al. (2019) that

$$\mathcal{W}_2(\mu_{ks}, \mu_k) = \mathcal{O}(s). \quad (29)$$

Let  $\Gamma$  be the optimal coupling between  $\mu_k$  and  $\mu_{ks}$ , and  $(X, Y) \sim \Gamma$ . Let  $X_t = \mathcal{T}_t^{k,s}(X)$  and  $Y_t$  be the solution of (10) with  $\mathbf{X}_0 = Y$  and  $t \in [ks, (k+1)s)$ . Then

$$X_t \sim \mu_t^s, \quad Y_t \sim \mu_t$$

and

$$Y_t = Y + \int_{ks}^t \mathbf{v}_{\tilde{t}}(Y_{\tilde{t}}) d\tilde{t}.$$

It follows that

$$\begin{aligned} \mathcal{W}_2^2(\mu_t, \mu_{ks}) &\leq \mathbb{E}[\|Y_t - Y\|_2^2] \\ &= \mathbb{E}\left[\left\|\int_{ks}^t \mathbf{v}_{\tilde{t}}(Y_{\tilde{t}}) d\tilde{t}\right\|_2^2\right] \\ &\leq \mathbb{E}\left[\left(\int_{ks}^t \|\mathbf{v}_{\tilde{t}}(Y_{\tilde{t}})\|_2 d\tilde{t}\right)^2\right] \\ &\leq \mathcal{O}(s^2). \end{aligned} \quad (30)$$

where, the first inequality follows from the definition of  $\mathcal{W}_2$ , and the last equality follows from the the uniform bounded assumption of  $\mathbf{v}_t$ . Similarly,

$$\begin{aligned}\mathcal{W}_2^2(\mu_k, \mu_t^s) &\leq \mathbb{E}[\|X - X_t\|_2^2] \\ &= \mathbb{E}[\|(t - ks)\mathbf{v}_k(X)\|_2^2] \\ &\leq \mathcal{O}(s^2).\end{aligned}\tag{31}$$

Then,

$$\begin{aligned}\mathcal{W}_2(\mu_t, \mu_t^s) &\leq \mathcal{W}_2(\mu_t, \mu_{ks}) + \mathcal{W}_2(\mu_{ks}, \mu_k) + \mathcal{W}_2(\mu_k, \mu_t^s) \\ &\leq \mathcal{O}(s),\end{aligned}$$

where the first inequality follows from the triangle inequality, see for example Lemma 5.3 in Santambrogio (2015), and the second one follows from (29)-(31).  $\blacksquare$

## B.2. Derivation of the results in Section 4.

### B.2.1. BREGMAN SCORE FOR DENSITY RATIO/DIFFERENCE

The separable Bregman score with the base probability measure  $p$  to measure the discrepancy between a measurable function  $R : \mathbb{R}^m \rightarrow \mathbb{R}^1$  and the density ratio  $r$  is

$$\begin{aligned}\mathfrak{B}_{\text{ratio}}(r, R) &= \mathbb{E}_{X \sim p}[g'(R(X))(R(X) - r(X)) - g(R(X))] \\ &= \mathbb{E}_{X \sim p}[g'(R(X))R(X) - g(R(X))] - \mathbb{E}_{X \sim q}[g'(R(X))].\end{aligned}$$

It can be verified that  $\mathfrak{B}_{\text{ratio}}(r, R) \geq \mathfrak{B}_{\text{ratio}}(r, r)$ , where the equality holds iff  $R = r$ .

For deep density-difference fitting, a neural network  $D : \mathbb{R}^m \rightarrow \mathbb{R}^1$  is utilized to estimate the density-difference  $d(\mathbf{x}) = q(\mathbf{x}) - p(\mathbf{x})$  between a given density  $q$  and the target  $p$ . The separable Bregman score with the base probability measure  $w$  to measure the discrepancy between  $D$  and  $d$  can be derived similarly,

$$\begin{aligned}\mathfrak{B}_{\text{diff}}(d, D) &= \mathbb{E}_{X \sim p}[w(X)g'(D(X))] - \mathbb{E}_{X \sim q}[w(X)g'(D(X))] \\ &\quad + \mathbb{E}_{X \sim w}[g'(D(X))D(X) - g(D(X))].\end{aligned}$$

Here, we focus on the widely used least-squares density-ratio (LSDR) fitting with  $g(c) = (c - 1)^2$  as a working example for estimating the density ratio  $r$ . The LSDR loss function is

$$\mathfrak{B}_{\text{LSDR}}(r, R) = \mathbb{E}_{X \sim p}[R(X)^2] - 2\mathbb{E}_{X \sim q}[R(X)] + 1.$$

### B.2.2. GRADIENT PENALTY

We consider a noise convolution form of  $\mathfrak{B}_{\text{ratio}}(r, R)$  with Gaussian noise  $\epsilon \sim \mathcal{N}(\mathbf{0}, \alpha\mathbf{I})$ ,

$$\mathfrak{B}_{\text{ratio}}^\alpha(r, R) = \mathbb{E}_{X \sim p}\mathbb{E}_\epsilon[g'(R(X + \epsilon))R(X + \epsilon) - g(R(X + \epsilon))] - \mathbb{E}_{X \sim q}\mathbb{E}_\epsilon[g'(R(X + \epsilon))].$$

Taylor expansion applied to  $R$  gives

$$\mathbb{E}_\epsilon[R(\mathbf{x} + \epsilon)] = R(\mathbf{x}) + \frac{\alpha}{2}\Delta R(\mathbf{x}) + \mathcal{O}(\alpha^2).$$

Using equations (13)-(17) in [Roth et al. \(2017\)](#), we get

$$\mathfrak{B}_{\text{ratio}}^\alpha(r, R) \approx \mathfrak{B}_{\text{ratio}}(r, R) + \frac{\alpha}{2} \mathbb{E}_p[g''(R) \|\nabla R\|_2^2],$$

i.e.,  $\frac{1}{2} \mathbb{E}_p[g''(R) \|\nabla R\|_2^2]$  serves as a regularizer for deep density-ratio fitting when  $g$  is twice differentiable.

### B.2.3. PROOF LEMMA 10

By definition, it is easy to check

$$\mathfrak{B}_{\text{LSDR}}^0(R) = \mathfrak{B}_{\text{ratio}}(r, R) - \mathfrak{B}_{\text{ratio}}(r, r),$$

where  $\mathfrak{B}_{\text{ratio}}(r, R)$  is the Bregman score with the base probability measure  $p$  between  $R$  and  $r$ . Then  $r \in \arg \min_{\text{measureable } R} \mathfrak{B}_{\text{LSDR}}^0(R)$  follow from the fact  $\mathfrak{B}_{\text{ratio}}(r, R) \geq \mathfrak{B}_{\text{ratio}}(r, r)$  and the equality holds iff  $R = r$ . Since

$$\mathfrak{B}^\alpha(R) = \mathfrak{B}_{\text{LSDR}}^0(R) + \alpha \mathbb{E}_p[\|\nabla R\|_2^2] \geq 0,$$

Then,

$$\mathfrak{B}^\alpha(R) = 0$$

iff

$$\mathfrak{B}_{\text{LSDR}}^0(R) = 0 \text{ and } \mathbb{E}_p[\|\nabla R\|_2^2] = 0,$$

which is further equivalent to

$$R = r = \text{constant } (q, p)\text{-a.e.},$$

and the constant = 1 since  $r$  is a density ratio. ■

### B.2.4. PROOF OF THEOREM 11

We use  $\mathfrak{B}(R)$  to denote  $\mathfrak{B}_{\text{LSDR}}^0 - C$  for simplicity, i.e.,

$$\mathfrak{B}(R) = \mathbb{E}_{X \sim p}[R(X)^2] - 2\mathbb{E}_{X \sim q}[R(X)]. \quad (32)$$

Rewrite (20) with  $\alpha = 0$  as

$$\widehat{R}_\phi \in \arg \min_{R_\phi \in \mathcal{H}_{\mathcal{D}, \mathcal{W}, \mathcal{S}, \mathcal{B}}} \widehat{\mathfrak{B}}(R_\phi) = \sum_{i=1}^n \frac{1}{n} (R_\phi(X_i)^2 - 2R_\phi(Y_i)). \quad (33)$$

By Lemma 10 and Fermat's rule ([Clarke, 1990](#)), we know  $\mathbf{0} \in \partial \mathfrak{B}(r)$ . Then,  $\forall R$  direct calculation yields,

$$\|R - r\|_{L^2(\nu)}^2 = \mathfrak{B}(R) - \mathfrak{B}(r) - \langle \partial \mathfrak{B}(r), R - r \rangle = \mathfrak{B}(R) - \mathfrak{B}(r). \quad (34)$$

$\forall \bar{R}_\phi \in \mathcal{H}_{\mathcal{D}, \mathcal{W}, \mathcal{S}, \mathcal{B}}$  we have,

$$\begin{aligned} \|\widehat{R}_\phi - r\|_{L^2(\nu)}^2 &= \mathfrak{B}(\widehat{R}_\phi) - \mathfrak{B}(r) \\ &= \mathfrak{B}(\widehat{R}_\phi) - \widehat{\mathfrak{B}}(\widehat{R}_\phi) + \widehat{\mathfrak{B}}(\widehat{R}_\phi) - \widehat{\mathfrak{B}}(\bar{R}_\phi) \\ &\quad + \widehat{\mathfrak{B}}(\bar{R}_\phi) - \mathfrak{B}(\bar{R}_\phi) + \mathfrak{B}(\bar{R}_\phi) - \mathfrak{B}(r) \\ &\leq 2 \sup_{R \in \mathcal{H}_{\mathcal{D}, \mathcal{W}, \mathcal{S}, \mathcal{B}}} |\mathfrak{B}(R) - \widehat{\mathfrak{B}}(R)| + \|\bar{R}_\phi - r\|_{L^2(\nu)}^2, \end{aligned} \quad (35)$$

where the inequality uses the definition of  $\widehat{R}_\phi$ ,  $\bar{R}_\phi$  and (34). We prove the theorem by upper bounding the expected value of the right hand side term in (35). To this end, we need the following three inequalities (36)-(38). First, we show that

$$\mathbb{E}_{\{Z_i\}_i^n}[\sup_R |\mathfrak{B}(R) - \widehat{\mathfrak{B}}(R)|] \leq 4C_1(2\mathcal{B} + 1)\mathfrak{G}(\mathcal{H}), \quad (36)$$

where

$$\mathfrak{G}(\mathcal{H}) = \mathbb{E}_{\{Z_i, \epsilon_i\}_i^n} \left[ \sup_{R \in \mathcal{H}_{\mathcal{D}, \mathcal{W}, \mathcal{S}, \mathcal{B}}} \frac{1}{n} \sum_{i=1}^n \epsilon_i R(Z_i) \right]$$

is the Gaussian complexity of  $\mathcal{H}_{\mathcal{D}, \mathcal{W}, \mathcal{S}, \mathcal{B}}$  (Bartlett and Mendelson, 2002).

**Proof of (36).** Let  $g(c) = c^2 - c$ ,  $\mathbf{z} = (\mathbf{x}, \mathbf{y}) \in \mathbb{R}^m \times \mathbb{R}^m$ ,

$$\widetilde{R}(\mathbf{z}) = (g \circ R)(\mathbf{z}) = R^2(\mathbf{x}) - R(\mathbf{y}).$$

Denote  $Z = (X, Y)$ ,  $Z_i = (X_i, Y_i)$ ,  $i = 1, \dots, n$  with  $X, X_i$  i.i.d.  $\sim p$ ,  $Y, Y_i$  i.i.d.  $\sim q$ . Let  $\widetilde{Z}_i$  be an i.i.d. copy of  $Z_i$ , and  $\sigma_i(\epsilon_i)$  be i.i.d. Rademacher random (standard normal) variables that are independent of  $Z_i$  and  $\widetilde{Z}_i$ . Then,

$$\mathfrak{B}(R) = \mathbb{E}_Z[\widetilde{R}(Z)] = \frac{1}{n} \mathbb{E}_{\widetilde{Z}_i}[\widetilde{R}(\widetilde{Z}_i)],$$

and

$$\widehat{\mathfrak{B}}(R) = \frac{1}{n} \sum_{i=1}^n \widetilde{R}(Z_i).$$

Denote

$$\mathfrak{A}(\mathcal{H}) = \frac{1}{n} \mathbb{E}_{\{Z_i, \sigma_i\}_i^n} \left[ \sup_{R \in \mathcal{H}_{\mathcal{D}, \mathcal{W}, \mathcal{S}, \mathcal{B}}} \sum_{i=1}^n \sigma_i R(Z_i) \right]$$

as the Rademacher complexity of  $\mathcal{H}_{\mathcal{D}, \mathcal{W}, \mathcal{S}, \mathcal{B}}$  (Bartlett and Mendelson, 2002). Then,

$$\begin{aligned} \mathbb{E}_{\{Z_i\}_i^n}[\sup_R |\mathfrak{B}(R) - \widehat{\mathfrak{B}}(R)|] &= \frac{1}{n} \mathbb{E}_{\{Z_i\}_i^n} \left[ \sup_R \left| \sum_{i=1}^n (\mathbb{E}_{\widetilde{Z}_i}[\widetilde{R}(\widetilde{Z}_i)] - \widetilde{R}(Z_i)) \right| \right] \\ &\leq \frac{1}{n} \mathbb{E}_{\{Z_i, \widetilde{Z}_i\}_i^n} \left[ \sup_R |\widetilde{R}(\widetilde{Z}_i) - \widetilde{R}(Z_i)| \right] \\ &= \frac{1}{n} \mathbb{E}_{\{Z_i, \widetilde{Z}_i, \sigma_i\}_i^n} \left[ \sup_R \left| \sum_{i=1}^n \sigma_i (\widetilde{R}(\widetilde{Z}_i) - \widetilde{R}(Z_i)) \right| \right] \\ &\leq \frac{1}{n} \mathbb{E}_{\{Z_i, \sigma_i\}_i^n} \left[ \sup_R \left| \sum_{i=1}^n \sigma_i \widetilde{R}(Z_i) \right| \right] + \frac{1}{n} \mathbb{E}_{\{\widetilde{Z}_i, \sigma_i\}_i^n} \left[ \sup_R \left| \sum_{i=1}^n \sigma_i \widetilde{R}(\widetilde{Z}_i) \right| \right] \\ &= 2\mathfrak{A}(g \circ \mathcal{H}) \\ &\leq 4(2\mathcal{B} + 1)\mathfrak{A}(\mathcal{H}) \\ &\leq 4C_1(2\mathcal{B} + 1)\mathfrak{G}(\mathcal{H}), \end{aligned}$$

where, the first inequality follows from the Jensen's inequality, and the second equality holds since the distribution of  $\sigma_i(\widetilde{R}(\widetilde{Z}_i) - \widetilde{R}(Z_i))$  and  $\widetilde{R}(\widetilde{Z}_i) - \widetilde{R}(Z_i)$  are the same, and the last equality holds

since the distribution of the two terms are the same, and last two inequality follows from the Lipschitz contraction property where the Lipschitz constant of  $g$  on  $\mathcal{H}_{\mathcal{D},\mathcal{W},\mathcal{S},\mathcal{B}}$  is bounded by  $2\mathcal{B} + 1$  and the relationship between the Gaussian complexity and the Rademacher complexity, see for Theorem 12 and Lemma 4 in [Bartlett and Mendelson \(2002\)](#), respectively. Next, we bound the Gaussian complexity

$$\mathfrak{G}(\mathcal{H}) \leq C_2\mathcal{B}\sqrt{\frac{n}{\mathcal{D}\mathcal{S}\log\mathcal{S}}}\log\frac{n}{\mathcal{D}\mathcal{S}\log\mathcal{S}}\exp(-\log^2\frac{n}{\mathcal{D}\mathcal{S}\log\mathcal{S}}). \quad (37)$$

**Proof of (37).** Since  $\mathcal{H}$  is negation closed,

$$\begin{aligned} \mathfrak{G}(\mathcal{H}) &= \mathbb{E}_{\{Z_i, \epsilon_i\}_i^n} \left[ \sup_{R \in \mathcal{H}_{\mathcal{D},\mathcal{W},\mathcal{S},\mathcal{B}}} \frac{1}{n} \sum_{i=1}^n \epsilon_i R(Z_i) \right] \\ &= \mathbb{E}_{Z_i} \left[ \mathbb{E}_{\epsilon_i} \left[ \sup_{R \in \mathcal{H}_{\mathcal{D},\mathcal{W},\mathcal{S},\mathcal{B}}} \frac{1}{n} \sum_{i=1}^n \epsilon_i R(Z_i) \right] \middle| \{Z_i\}_{i=1}^n \right]. \end{aligned}$$

Conditioning on  $\{Z_i\}_{i=1}^n, \forall R, \tilde{R} \in \mathcal{H}_{\mathcal{D},\mathcal{W},\mathcal{S},\mathcal{B}}$  it easy to check

$$\mathbb{V}_{\epsilon_i} \left[ \frac{1}{n} \sum_{i=1}^n \epsilon_i (R(Z_i) - \tilde{R}(Z_i)) \right] = \frac{d_2^{\mathcal{H}}(R, \tilde{R})}{\sqrt{n}},$$

where,  $d_2^{\mathcal{H}}(R, \tilde{R}) = \frac{1}{\sqrt{n}} \sqrt{\sum_{i=1}^n (R(Z_i) - \tilde{R}(Z_i))^2}$ . Observing the diameter of  $\mathcal{H}_{\mathcal{D},\mathcal{W},\mathcal{S},\mathcal{B}}$  under  $d_2^{\mathcal{H}}$  is at most  $\mathcal{B}$ , we have

$$\begin{aligned} \mathfrak{G}(\mathcal{H}) &\leq \frac{C_3}{\sqrt{n}} \mathbb{E}_{\{Z_i\}_{i=1}^n} \left[ \int_0^{\mathcal{B}} \sqrt{\log \mathcal{N}(\mathcal{H}, d_2^{\mathcal{H}}, \delta)} d\delta \right] \\ &\leq \frac{C_3}{\sqrt{n}} \mathbb{E}_{\{Z_i\}_{i=1}^n} \left[ \int_0^{\mathcal{B}} \sqrt{\log \mathcal{N}(\mathcal{H}, d_{\infty}^{\mathcal{H}}, \delta)} d\delta \right] \\ &\leq \frac{C_3}{\sqrt{n}} \int_0^{\mathcal{B}} \sqrt{\text{VC}_{\mathcal{H}} \log \frac{6\mathcal{B}n}{\delta \text{VC}_{\mathcal{H}}}} d\delta, \\ &\leq C_4\mathcal{B} \left( \frac{n}{\text{VC}_{\mathcal{H}}} \right)^{1/2} \log \left( \frac{n}{\text{VC}_{\mathcal{H}}} \right) \exp(-\log^2 \left( \frac{n}{\text{VC}_{\mathcal{H}}} \right)) \\ &\leq C_2\mathcal{B} \sqrt{\frac{n}{\mathcal{D}\mathcal{S}\log\mathcal{S}}}\log\frac{n}{\mathcal{D}\mathcal{S}\log\mathcal{S}}\exp(-\log^2\frac{n}{\mathcal{D}\mathcal{S}\log\mathcal{S}}) \end{aligned}$$

where the first inequality follows from the chaining Theorem 8.1.3 in [Vershynin \(2018\)](#), the second inequality holds due to  $d_2^{\mathcal{H}} \leq d_{\infty}^{\mathcal{H}}$ , in the third inequality we used the relationship between the metric entropy and the VC-dimension of the ReLU networks  $\mathcal{H}_{\mathcal{D},\mathcal{W},\mathcal{S},\mathcal{B}}$  ([Anthony and Bartlett, 2009](#)), i.e.,

$$\log \mathcal{N}(\mathcal{H}, d_{\infty}^{\mathcal{H}}, \delta) \leq \text{VC}_{\mathcal{H}} \log \frac{6\mathcal{B}n}{\delta \text{VC}_{\mathcal{H}}},$$

the fourth inequality follows by some calculation, and the last inequality holds due to the upper bound of VC-dimension for the ReLU network  $\mathcal{H}_{\mathcal{D},\mathcal{W},\mathcal{S},\mathcal{B}}$  satisfying

$$\text{VC}_{\mathcal{H}} \leq C_5\mathcal{D}\mathcal{S}\log\mathcal{S},$$

see [Bartlett et al. \(2019\)](#).

The third inequality we is the following. For any two integer  $M, N$ , there exists a  $\bar{R}_\phi \in \mathcal{H}_{\mathcal{D}, \mathcal{W}, \mathcal{S}, \mathcal{B}}$  with width  $\mathcal{W} = \max\{8\mathcal{M}N^{1/\mathcal{M}} + 4\mathcal{M}, 12N + 14\}$  and depth  $\mathcal{D} = 9M + 12$ , and  $\mathcal{B} = 2B$ , such that

$$\|r - \bar{R}_\phi\|_{L^2(\nu)}^2 \leq C_6 c L m \mathcal{M} (NM)^{-4/\mathcal{M}}. \quad (38)$$

**Proof of (38).** We use Lemma 4.1, Theorem 4.3, 4.4 and following the proof of Theorem 1.3 in [Shen et al. \(2019\)](#). Let  $\mathbf{A}$  be the random orthoprojector in Theorem 4.4, then it is to check  $\mathbf{A}(\mathfrak{M}_\epsilon) \subset \mathbf{A}([-c, c]^m) \subset [-c\sqrt{m}, \sqrt{m}c]^M$ . Let  $\tilde{r}$  be an extension of the restriction of  $r$  on  $\mathfrak{M}_\epsilon$ , which is defined similarly as  $\tilde{g}$  on page 30 in [Shen et al. \(2019\)](#). Since we assume the target  $r$  is Lipschitz continuous with the bound  $B$  and the Lipschitz constant  $L$ , let  $\epsilon$  small enough, then by Theorem 4.3, there exist a ReLU network  $\tilde{R}_\phi \in \mathcal{H}_{\mathcal{D}, \mathcal{W}, \mathcal{S}, \mathcal{B}}$  with width

$$\mathcal{W} = \max\{8\mathcal{M}N^{1/\mathcal{M}} + 4\mathcal{M}, 12N + 14\},$$

and depth

$$\mathcal{D} = 9M + 12,$$

and  $\mathcal{B} = 2B$ , such that

$$\|\tilde{r} - \tilde{R}_\phi\|_{L^\infty(\mathfrak{M}_\epsilon \setminus \mathcal{N})} \leq 80cL\sqrt{m\mathcal{M}}(NM)^{-2/m},$$

and

$$\|\tilde{R}_\phi\|_{L^\infty(\mathfrak{M}_\epsilon)} \leq B + 3Lc\sqrt{m\mathcal{M}},$$

where,  $\mathcal{N}$  is a  $\nu$ -negligible set with  $\nu(\mathcal{N})$  can be arbitrary small. Define  $\bar{R}_\phi = \tilde{R}_\phi \circ \mathbf{A}$ . Then, following the proof after equation (4.8) in Theorem 1.3 of [Shen et al. \(2019\)](#), we get our (38) and

$$\|\bar{R}_\phi\|_{L^\infty(\mathfrak{M}_\epsilon \setminus \mathcal{N})} \leq 2B, \|\bar{R}_\phi\|_{L^\infty(\mathcal{N})} \leq 2B + 3cL\sqrt{m\mathcal{M}}.$$

Let  $\mathcal{D}\mathcal{S} \log \mathcal{S} < n$ , combing the results (35) - (38), we have

$$\begin{aligned} & \mathbb{E}_{\{X_i, Y_i\}_1^n} [\|\hat{R}_\phi - r\|_{L^2(\nu)}^2] \\ & \leq 8C_1(2B + 1)\mathfrak{G}(\mathcal{H}) + C_6 c L m \mathcal{M} (NM)^{-4/\mathcal{M}} \\ & \leq 8C_1(2B + 1)C_2 B \sqrt{\frac{\mathcal{D}\mathcal{S} \log \mathcal{S}}{n}} \log \frac{n}{\mathcal{D}\mathcal{S} \log \mathcal{S}} \\ & \quad + C_6 c L m \mathcal{M} (NM)^{-4/\mathcal{M}} \\ & \leq C(B^2 + cLm\mathcal{M})n^{-2/(2+\mathcal{M})}, \end{aligned}$$

where, last inequality holds since we choose  $M = \log n$ ,  $N = n^{\frac{\mathcal{M}}{2(2+\mathcal{M})}} / \log n$ ,  $\mathcal{S} = n^{\frac{\mathcal{M}-2}{\mathcal{M}+2}} / \log^4 n$ , i.e.,  $\mathcal{D} = 9 \log n + 12$ ,  $\mathcal{W} = 12n^{\frac{\mathcal{M}}{2(2+\mathcal{M})}} / \log n + 14$ .  $\blacksquare$



### B.3. Proof of the relation between EPT and SVGD

Here we show that SVGD can be derived from EPT.

**Proof** Let  $f(x) = x \log x$  in (5). With this  $f$ -divergence function, the velocity fields  $\mathbf{v}_t = -f''(r_t)\nabla r_t = -\frac{\nabla r_t(\mathbf{x})}{r_t(\mathbf{x})}$ . Let  $\mathbf{g}$  in a Stein class associated with  $q_t$ .

$$\begin{aligned}
 & \langle \mathbf{v}_t, \mathbf{g} \rangle_{\mathcal{H}(q_t)} \\
 &= - \int \mathbf{g}(\mathbf{x})^T \frac{\nabla r_t(\mathbf{x})}{r_t(\mathbf{x})} q_t(\mathbf{x}) d\mathbf{x} \\
 &= - \int \mathbf{g}(\mathbf{x})^T \nabla \log r_t(\mathbf{x}) q_t(\mathbf{x}) d\mathbf{x} \\
 &= - \mathbb{E}_{\mathbf{X} \sim q_t(\mathbf{x})} [\mathbf{g}(\mathbf{x})^T \nabla \log q_t(\mathbf{X}) + \mathbf{g}(\mathbf{x})^T \nabla \log p(\mathbf{X})] \\
 &= - \mathbb{E}_{\mathbf{X} \sim q_t(\mathbf{x})} [\mathbf{g}(\mathbf{x})^T \nabla \log q_t(\mathbf{X}) + \nabla \cdot \mathbf{g}(\mathbf{x})] \\
 &\quad + \mathbb{E}_{\mathbf{X} \sim q_t(\mathbf{x})} [\mathbf{g}(\mathbf{x})^T \nabla \log p(\mathbf{X}) + \nabla \cdot \mathbf{g}(\mathbf{x})] \\
 &= - \mathbb{E}_{\mathbf{X} \sim q_t(\mathbf{x})} [\mathcal{T}_{q_t} \mathbf{g}] + \mathbb{E}_{\mathbf{X} \sim q_t(\mathbf{x})} [\mathcal{T}_p \mathbf{g}] \\
 &= \mathbb{E}_{\mathbf{X} \sim q_t(\mathbf{x})} [\mathcal{T}_p \mathbf{g}],
 \end{aligned}$$

where the last equality is obtained by restricting  $\mathbf{g}$  in a Stein class associated with  $q_t$ , i.e.,  $\mathbb{E}_{\mathbf{X} \sim q_t(\mathbf{x})} \mathcal{T}_{q_t} \mathbf{g} = 0$ . This is the velocity fields of SVGD (Liu, 2017). ■