

A Wasserstein Minimum Velocity Approach to Learning Unnormalized Models

Ziyu Wang

Shuyu Cheng

Yueru Li

Jun Zhu

Bo Zhang

Dept. of Comp. Sci. & Tech., BNRIst Center, Institute for AI, THBI Lab, Tsinghua University

Abstract

Score matching provides an effective approach to learning flexible unnormalized models, but its scalability is limited by the need to evaluate a second-order derivative. In this paper, we present a scalable approximation to a general family of learning objectives including score matching, by observing a new connection between these objectives and Wasserstein gradient flows. We present applications with promise in learning neural density estimators on manifolds, and training implicit variational and Wasserstein auto-encoders with a manifold-valued prior.

1 INTRODUCTION

A flexible approach to density estimation is to parameterize an unnormalized density function, or energy function. In particular, unnormalized models with energy parameterized by deep neural networks have been successfully applied to density estimation (Wenliang et al., 2019; Saremi et al., 2018) and learning implicit auto-encoding models (Song et al., 2019).

Parameter estimation for such unnormalized models is highly non-trivial: the maximum likelihood objective is intractable, due to the presence of a normalization term. *SM* (Hyvärinen, 2005) is a popular alternative, yet applying score matching to complex unnormalized models can be difficult, as the objective involves the second-order derivative of the energy, rendering gradient-based optimization infeasible. In practice, people turn to scalable approximations of the score matching objective (Song et al., 2019; Hyvärinen, 2007; Vincent, 2011; Raphan and Simoncelli, 2011), or other objectives such as the kernelized Stein

discrepancy (KSD; Liu et al., 2016b; Liu and Wang, 2017). So far, approximations to these objectives are developed on a case-by-case basis, leaving important applications unaddressed; for example, there is a lack of scalable learning methods for unnormalized models on manifolds (Mardia et al., 2016).

In this work, we present a unifying perspective to this problem, and derive scalable approximations for a variety of learning objectives including score matching. We start by interpreting these objectives as the initial *velocity* of certain distribution-space gradient flows, which are simulated by common samplers. This novel interpretation leads to a scalable approximation algorithm for all such objectives, reminiscent to single-step contrastive divergence (CD-1).

We refer to any objective with the above interpretation as above as a “minimum velocity learning objective”, a term coined in the unpublished work (Movellan, 2007). Movellan (2007) focused on the specific case of SM; in contrast, our formulation generalizes theirs by lifting the concept of velocity from data space to distribution space, thus applies to different objectives as the choice of distribution space varies. For example, our method applies to score matching and Riemannian SM when we choose the 2-Wasserstein space, and to KSD when we choose the \mathcal{H} -Wasserstein space (Liu, 2017); we can also derive instances of the minimum velocity learning objective when the distribution-space gradient flow corresponds to less well-studied samplers, such as (Zhang et al., 2018; Lu et al., 2019). Another gap we fill in is the development of a practically applicable algorithm.

Our algorithm is connected to previous work using CD-1 to estimate the *gradient* of certain objectives (Hyvärinen, 2007; Movellan, 2007; Liu and Wang, 2017); however, there are important differences. From a theoretical perspective, we provide a unified derivation for all such objectives, including those not considered in previous work; our gradient-flow-based derivation is also simpler, and leads to an improved understanding of this approach. From an algorithmic perspective,

tive, we directly approximate the objective function instead of its gradient, enabling the use of regularization like early-stopping. More importantly, we identify an *infinite-variance problem* in the approximate score matching objective, which has previously rendered the approximation impractical (Hyvärinen, 2007; Saremi et al., 2018); we further present a simple fix. As a side product of our work, our fix also applies to denoising score matching (Raphael and Simoncelli, 2011; Vincent, 2011), another score matching approximation that suffers from this problem.

One important application of our method is in learning unnormalized models on manifolds, as our method leads to a scalable approximation for the Riemannian score matching objective. Density estimation on manifolds is needed in areas such as image analysis (Srivastava et al., 2007), geology (Davis and Sampson, 1986) and bioinformatics (Boomsma et al., 2008). Moreover, our approximation leads to flexible inference schemes for variational and Wasserstein auto-encoders with manifold-valued latent variables, as it enables gradient estimation for implicit variational distributions on manifolds. Auto-encoders with a manifold-valued latent space can capture the distribution of certain types of data better. For example, a hyperbolic latent space could be more suitable when the data has a hierarchical structure (Mathieu et al., 2019; Ovinnikov, 2019), and a hyperpherical prior could be more suitable for directional data (Davidson et al., 2018). As we shall see in experiments, our method improves the performance of manifold-latent VAEs and WAEs.

The rest of this paper is organized as follows: Section 2 reviews the preliminary knowledge: manifolds, gradient flows and their connection to common sampling algorithms. We present our method in Section 3 and its applications in Section 4. Section 5 contains a review of the related work, and Section 6 contains experiments. We provide our conclusions in Section 7.

2 PRELIMINARIES

2.1 Manifolds, Flows and the 2-Wasserstein Space

We recall concepts from differential manifolds.

A **(differential) manifold** \mathcal{M} is a topological space locally diffeomorphic to an Euclidean or Hilbert space. A manifold is covered by a set of *charts*, which enables the use of coordinates locally, and specifies a set of basis $\{\partial_i\}$ in the **local tangent space**. A **Riemannian manifold** further possesses a *Riemannian structure*, which assigns to each tangent space $\mathcal{T}_p\mathcal{M}$ an inner product structure.

The manifold structure enables us to differentiate a function along curves. Specifically, consider a curve $c : [0, T] \rightarrow \mathcal{M}$, and a smooth function $f : \mathcal{M} \rightarrow \mathbb{R}$. At $c(t) \in \mathcal{M}$, a **tangent vector** $\frac{dc}{dt}|_{t_0} \in \mathcal{T}_{c(t)}\mathcal{M}$ describes the velocity of c passing $c(t)$; the **differential** of the function f at $c(t)$, denoted as $(df)_{c(t)}$, is a **linear map** from $\mathcal{T}_{c(t)}\mathcal{M}$ to \mathbb{R} :

$$(df)_{c(t_0)} \left(\frac{dc}{dt} \Big|_{t_0} \right) = \frac{d}{dt} f(c(t)) \Big|_{t_0}.$$

A **tangent vector field** assigns to each $p \in \mathcal{M}$ a tangent vector $V_p \in \mathcal{T}_p\mathcal{M}$. It determines a **flow**, a set of curves $\{\phi_p(t) : p \in \mathcal{M}\}$ which all have $V_{\phi_p(t)}$ as their velocity. On Riemannian manifolds, the **gradient** of a smooth function f is a tangent vector field $p \mapsto \text{grad}_p f$ such that $\langle \text{grad}_p f, v \rangle = (df)_p(v)$ for all $v \in \mathcal{T}_p\mathcal{M}$. It determines the **gradient flow**.

We will work with two types of manifolds: the data space \mathcal{X} when we apply our method to manifold-valued data, and the space of probability distributions over \mathcal{X} . On the space of distributions, we are mostly interested in the 2-Wasserstein space $\mathcal{P}(\mathcal{X})$, a Riemannian manifold.

The properties of $\mathcal{P}(\mathcal{X})$ (Villani, 2008):

1. Its tangent space $\mathcal{T}_p\mathcal{P}(\mathcal{X})$ can be identified as a subspace of the space of vector fields on \mathcal{X} ; the Riemannian metric of $\mathcal{P}(\mathcal{X})$: $p(u) \cdot q(u)$

$$\langle X, Y \rangle_p := \mathbb{E}_{p(u)} \langle X(u), Y(u) \rangle_u, \quad (1)$$

for all $p \in \mathcal{P}(\mathcal{X})$, $X, Y \in \mathcal{T}_p\mathcal{P}(\mathcal{X})$; the inner product on the rhs above is determined by Riemannian structure of \mathcal{X} . $X: u: X \rightarrow X(u): TuX$

2. The gradient of the KL divergence functional $\text{KL}_p(q) := \text{KL}(q||p)$ in $\mathcal{P}(\mathcal{X})$ is

$$(\text{grad}_q \text{KL}_p)(u) = \text{grad}_u \log \frac{q(u)}{p(u)}. \quad (2)$$

We will also consider a few other spaces of distributions, including the Wasserstein-Fisher-Rao space (Lu et al., 2019), and the \mathcal{H} -Wasserstein space introduced in (Lin, 2017).

On the data space, we need to introduce the notion of density, i.e. the Radon–Nikodym derivative w.r.t. a suitable base measure. The Hausdorff measure is one such choice; it reduces to the Lebesgue measure when $\mathcal{X} = \mathbb{R}^n$. In most cases, distributions on manifolds are specified using their density w.r.t. the Hausdorff measure; e.g. “uniform” distributions has constant densities in this sense.

(SPOS; Zhang et al., 2018; Chen et al., 2018) combines the dynamics of SVGD and Langevin dynamics; as we will show in Appendix B.1, SPOS also has a gradient flow structure.

3 WASSERSTEIN MINIMUM VELOCITY LEARNING

2.2 Posterior Sampling by Simulation of Gradient Flows

They include diffusion-based MCMC, particle-based variational inference, and other stochastic interacting particlesystems.

Riemannian Langevin Dynamics Suppose our target distribution has density $p(x)$ w.r.t. the Hausdorff measure of \mathcal{X} . In a local chart $U \subset \mathcal{X}$, let $G : U \rightarrow \mathbb{R}^{m \times m}$ be the coordinate matrix of its Riemannian metric. Then the *Riemannian Langevin dynamics* corresponds to the following SDE in the chart¹:

$$dx = V(x)dt + \sqrt{2G^{-1}(x)}dB_t \quad (3)$$

where

$$V^i(x) = g^{ij}\partial_j \left(\log p(x) - \frac{\log |G(x)|}{2} \right) + \partial_j g^{ij}, \quad (4)$$

and (g^{ij}) is the coordinate of the matrix G^{-1} . It is known (Villani, 2008) that the Riemannian Langevin dynamics is the gradient flow of the KL functional $\text{KL}_p(q) := \text{KL}(q||p)$ in the 2-Wasserstein space $\mathcal{P}(\mathcal{X})$.

Particle-based Samplers A range of samplers approximate the gradient flow of KL_p in various spaces, using deterministic or stochastic interacting particle systems.² For instance, Stein variational gradient descent (SVGD; Liu and Wang, 2016) simulates the gradient flow in the so-called \mathcal{H} -Wasserstein space (Liu, 2017), which replaces the Riemannian structure in $\mathcal{P}(\mathcal{X})$ with the RKHS inner product. Birth-death accelerated LDs (Lu et al., 2019) is a stochastic interacting particle system that simulates to the gradient flow of KL_p in the Wasserstein-Fisher-Rao space. Finally, the stochastic particle-optimization sampler

¹ (3) differs from definitions in some works (e.g. Ma et al., 2015). This is because we define p as the density w.r.t. the Hausdorff measure of \mathcal{X} , while they use the Lebesgue measure. See also (Xifara et al., 2014; Hsu, 2008).

² There are other particle-based samplers (Liu et al., 2019b,a; Taghvaei and Mehta, 2019) corresponding to accelerated gradient flows. However, as we will be interested in the initial velocity of the flow, they do not lead to new MVL objectives.

In this section, we present our framework, which concerns all learning objectives of the following form:

$$L_{\text{mvl}}(\theta) := - \frac{d}{dt} \text{KL}(p_t || q_\theta) \Big|_{t=0}, \quad (5)$$

where p_t is defined as the gradient flow of KL_q in a suitable space of probability measures (e.g. the 2-Wasserstein space). We refer to any such objective as a “minimum velocity learning (MVL) objective”; as we shall see below, L_{mvl} equals the initial *velocity* of the gradient flow $\|\text{grad}_p \text{KL}_q\|^2$, in the corresponding distribution space.

In the following subsections, we will first set up the problem, and motivate the use of (5) by connecting it to score matching; then we present our approximation to (5), and its variance-reduced version; we also address the infinite-variance issue in two previous approximators for the score matching objective. Finally, we briefly discuss other instances of the MVL objective that our method can be applied to.

3.1 Score Matching and a Wasserstein Space View

Consider parameter estimation in the unnormalized model $q(x; \theta) \sim \exp(-\mathcal{E}(x; \theta))$.

SM:

$$D_F(p||q) := \frac{1}{2} \mathbb{E}_{p(x)} [\|\nabla_x \log p(x) - \nabla_x \log q(x; \theta)\|^2], \quad (6)$$

\iff

$$\mathbb{E}_{p(x)} \left[\Delta \log q(x; \theta) + \frac{1}{2} \|\nabla \log q(x; \theta)\|^2 \right], \quad (7)$$

Unfortunately, optimization w.r.t. second-order derivatives is prohibitively expensive when the energy is pa-

rameterized by DNNs, and scalable approximation to the SM objective must be developed.

observing

$$D_F(p|q) = \frac{1}{2} \|\text{grad}_p \text{KL}_q\|^2,$$

where the gradient and norm are defined in $\mathcal{P}(\mathcal{X})$, and the manifold \mathcal{X} inherits the Riemannian metric from \mathbb{R}^n . This follows directly from (1)-(2).

Now let p_t be the gradient flow of KL_q , i.e. $\frac{d}{dt}p_t = -\text{grad}_{p_t} \text{KL}_q$. Then

$$\begin{aligned} \|\text{grad}_p \text{KL}_q\|^2 &= d(\text{KL}_q)_p(\text{grad}_p \text{KL}_q) \\ &= -\left. \frac{d}{dt} \text{KL}(p_t \| q_\theta) \right|_{t=0} = L_{\text{mvl}}(\theta). \end{aligned} \quad (8)$$

Therefore, SM is a special case of the MVL objective (5), where the space of distributions is $\mathcal{P}(\mathcal{X})$.

3.2 Approximating the MVL Objective

While the MVL objective has a closed-form expression, it usually involves second-order derivatives. In this subsection, we will derive an efficient approximation scheme for the MVL objective. Our approximation will only involve first-order derivatives, thus it can be easily implemented using automatic differentiation softwares (e.g. TensorFlow).

First, observe that (8) holds regardless of the chosen space of distributions. Denote $\mathcal{H}[p] := \mathbb{E}_p \log p$, $\mathcal{F}[p] := \mathbb{E}_p \log q = -\mathbb{E}_p \mathcal{E}$, so $\text{KL}_q = \mathcal{H} - \mathcal{F}$,

(8) ==>

$$\begin{aligned} &\langle \text{grad}_p \mathcal{H} - \text{grad}_p \mathcal{F}, \text{grad}_p \mathcal{H} - \text{grad}_p \mathcal{F} \rangle \\ &= \|\text{grad}_p \mathcal{H}\|^2 - 2 \left\langle \text{grad}_p \mathcal{F}, \text{grad}_p \mathcal{H} - \frac{1}{2} \text{grad}_p \mathcal{F} \right\rangle \\ &= \|\text{grad}_p \mathcal{H}\|^2 - 2 \langle \text{grad}_p \mathcal{F}, \text{grad}_p \text{KL}_{q^{1/2}} \rangle. \end{aligned} \quad (9)$$

As the first term in (9) is independent of θ , the MVL objective is always equivalent to the second term. We will approximate the second term by simulating a modified gradient flow: let \tilde{p}_t be the distribution obtained by running the sampler targeting $q^{1/2}$. Then

$$\begin{aligned} \langle \text{grad}_p \mathcal{F}, -\text{grad}_p \text{KL}_{q^{1/2}} \rangle &= (d\mathcal{F})_p(-\text{grad}_p \text{KL}_{q^{1/2}}) \\ &= \lim_{\epsilon \rightarrow 0} \frac{\mathbb{E}_{\tilde{p}_t} \log q_\theta - \mathbb{E}_p \log q_\theta}{\epsilon}. \end{aligned} \quad (10)$$

(10) can be approximated by replacing the limit with a fixed ϵ , and running the corresponding sampler starting from a mini-batch of training data. The approximation becomes unbiased when $\epsilon \rightarrow 0$.

3.2.1 A Control Variate

We have derived an estimator of (10) with vanishing bias. However, the estimator will suffer from high variance when the sampler used in the MVL objective consists of Itô diffusion. Fortunately, we can solve this problem with a control variate.

To illustrate the problem as well as our solution, suppose $\{p_t\}$ corresponds to LD, and (wlog) we use a batch size of 1:

$$\hat{L}_\epsilon = \frac{2}{\epsilon} \left[\mathcal{E}(x^+) - \mathcal{E} \left(x^+ - \epsilon \nabla_x \frac{\mathcal{E}(x^+)}{2} + \sqrt{2\epsilon} Z \right) \right],$$

where x^+ is sampled from the training data, and $Z \sim \mathcal{N}(0, I)$. By Taylor expansion³, $\hat{L}_\epsilon/2$ equals

$$\begin{aligned} &\frac{1}{2} \|\nabla_x \mathcal{E}(x^+)\|^2 - Z^\top \nabla_x^2 \mathcal{E}(x^+) Z - \sqrt{\frac{2}{\epsilon}} Z^\top \nabla_x \mathcal{E}(x^+) \\ &+ o(1), \end{aligned} \quad (11)$$

and as $\epsilon \rightarrow 0$, $\text{Var } \hat{L}_\epsilon = \Theta(\epsilon^{-1}) \rightarrow \infty$.

Now we can see the need for a control variate. In this LD example, the control variate $\sqrt{2/\epsilon} Z^\top \nabla_x \mathcal{E}(x^+)$ will remove the infinite-variance term; More generally, our control variate is always the inner product of $\nabla_x \mathcal{E}(x^+)$ and the diffusion term in the sampler.

Wrapping up, our approximate MVL objective is calculated as follows:

1. Sample a mini-batch of input $\{x_i\}_{i=1}^B$.
2. Run a single step of the sampling algorithm on $\{x_i\}$ targeting $q_{1/2} \propto \exp(-\mathcal{E}(x; \theta)/2)$, with a step-size of ϵ . Denote the resulted state as $\{x_i^-\}$.
3. Return $\frac{1}{B} \sum_{j=1}^B (\mathcal{E}(x_i) - \mathcal{E}(x_i^-))$ plus the control variate.

The approximation becomes unbiased as $\epsilon \rightarrow 0$, and has $O(1)$ variance⁴ regardless of ϵ .

3.3 On CD-1 and Denoising SM: Pitfalls and Fixes

As a side product, we show that our variance analysis explains the pitfall of two well-known approximations to SM objective: CD-1 (Hyvärinen, 2007) and DSM (Vincent, 2011, DSM).

³We need to expand to the second order when the increment is a discretization of some Itô diffusion.

⁴under mild assumptions controlling the growth of $\nabla^3 \mathcal{E}$ (e.g. bounded by a polynomial), so that the residual term in (11) will have bounded variance when averaged over Z .

Both approximations become unbiased as a step-size hyper-parameter $\epsilon \rightarrow 0$, but did not match the performance of exact score matching in practice, as witnessed in Hyvärinen (2007); Saremi et al. (2018); Song et al. (2019). We propose novel control variates for these approximators. As we will show in Section 6.1, the variance-reduced versions of these approximations have comparable performance to the exact SM objective.

DSM DSM considers the objective

$$L_{\text{dsm}}(\theta) = \mathbb{E}_{p(x)\mathcal{N}(z|0,I)} \|x + \sigma z - (x + \psi_\theta(x + \sigma z))\|^2. \quad (12)$$

The first two terms inside the norm represent a noise corrupted sample, and ψ_θ represents a “single-step denoising direction” (Raphael and Simoncelli, 2011). It is proved that the optimal ψ satisfies $\psi = \sigma^2 \nabla \log \tilde{p}$, where \tilde{p} is the density of the corrupted distribution (Raphael and Simoncelli, 2011; Vincent, 2011).

Consider the stochastic estimator of (12). We assume a batch size of 1, and denote the data sample as x . To keep notations consistent, denote $\epsilon = \sigma^2$, $\psi_\theta(x) = \epsilon \nabla_x \mathcal{E}(x; \theta)$. ==>

$$\hat{L}_{\text{dsm}} = \|x + \sqrt{\epsilon} z - \epsilon \nabla_x \mathcal{E}(x + \sqrt{\epsilon} z; \theta) - x\|^2.$$

As is similar to Section 3.2.1, we can show by Taylor expansion (Appendix A) that

$$\lim_{\epsilon \rightarrow 0} \epsilon^{-2} \mathbb{E} \hat{L}_{\text{dsm}} = 2D_{\text{Fisher}}(p|q) + (\text{const}), \quad (13)$$

$$\lim_{\epsilon \rightarrow 0} \epsilon^{-2} \text{Var } \hat{L}_{\text{dsm}} = \infty; \quad (14)$$

furthermore, the variance reduced objective

$$\frac{\hat{L}_{\text{dsm}} - (\epsilon \|z\|^2 - 2\epsilon^{3/2} z^\top \nabla \mathcal{E}(x))}{\epsilon^2} \quad (15)$$

is unbiased with finite variance.

CD-1 with Langevin Dynamics

CD- K learning rule:

$$\theta_{\ell+1} \leftarrow \theta_\ell + \nu [\mathbb{E}_p \partial_\theta \mathcal{E} - \mathbb{E}_{p_K} \partial_\theta \mathcal{E}], \quad (16)$$

where ν is the learning rate, and p_K is obtained from p by running K steps of MCMC. (16) does not define a valid objective, since p_K also depends on θ ; however, Hyvärinen (2007) proved that when $K = 1$ and the sampler is the Langevin dynamics, (16) recovers the gradient of the score matching objective.

Using the same derivation as in Section 3.2.1, we can see that as the step-size of the sampler approaches 0 (and ν is re-scaled appropriately), the gradient produced by

CD-1 also suffers from infinite variance, and this can be fixed using the same control variate.

However, practical utility of CD-1 is still hindered by the fact that it does not correspond to a valid learning objective; consequently, it is impossible to monitor the training process for CD-1, or introduce regularizations such as early stopping⁵.

3.4 Instances of MVL Objectives

As the previous derivation is independent of the distribution space of choice, we can derive approximations to other learning objectives using samplers other than LD. An important example is the Riemannian score matching objective, which corresponds to Riemannian LD; we will discuss it in detail in Section 4.1. Another example is when we choose the sampler as SVGD. In this case, we will obtain an approximation to the kernelized Stein discrepancy, generalizing the derivation in (Liu and Wang, 2017). When the sampling algorithm is chosen as SPOS, the corresponding MVL objective will be an interpolation between KSD and the Fisher divergence. See Appendix B.2 for derivations. Finally, the use of birth-death accelerated Langevin dynamics leads to a novel learning objective.

In terms of applications, our work focuses on learning neural energy-based models, and these objectives do not improve over score matching in this aspect. However, these derivations are useful since they generalize previous discussions, and establish new connections between sampling algorithms and learning objectives. It is also possible that these approximate objectives could be useful in other scenarios, such as learning kernel exponential family models (Sriperumbudur et al., 2017), improving the training of GANs (Liu and Wang, 2017) or amortized variational inference methods (Ruiz and Titsias, 2019).

4 APPLICATIONS

We now present applications of our work, including a scalable learning algorithm for unnormalized models on manifolds, as well as its application on learning implicit auto-encoders with manifold-valued priors.

4.1 MVL on Riemannian Manifolds

Density estimation on manifolds is needed in many application areas. While it is natural to consider unnormalized models on manifolds, there has been a lack of

⁵ In practice, the term $\mathbb{E}_p \mathcal{E} - \mathbb{E}_{p_K} \mathcal{E}$ is often used to tract the training process of CD- K . It is not a proper loss; we can see from (9) that when $K = 1$ and $\epsilon \rightarrow 0$, $\mathbb{E}_p \mathcal{E} - \mathbb{E}_{p_K} \mathcal{E}$ is significantly different from the proper score matching (MVL) loss, by a term of $\frac{1}{2} \|\text{grad}_p \mathcal{F}\|^2$.

scalable learning methods. Here we address this issue, by applying our method to obtain a scalable approximation to the Riemannian SM objective (Mardia et al., 2016).

Given the data manifold \mathcal{X} , we define an unnormalized model on it by parameterizing the log density w.r.t. the Hausdorff measure, and define the density as $q(x; \theta) \propto \exp(-\mathcal{E}(x; \theta))$. The Riemannian score matching objective will have the same form as (6); although enormous in (6) is now determined by the metric on \mathcal{X} , and the base measure of the densities has changed.

It is easy to verify that the derivation in Section 3.1 still applies in the manifold case. Thus the Riemannian score matching objective is a special case of the MVL objective, in which the distribution space is still chosen as $\mathcal{P}(\mathcal{X})$. The difference is that $\mathcal{P}(\mathcal{X})$ is now defined with the non-trivial data-space metric, and the gradient flow of KL_q becomes the Riemannian Langevin dynamics (3). We can approximate the objective by doing a single step of Riemannian LD for small ϵ :

$$L_{\text{mvl-rl}} = \frac{2}{\epsilon} (\mathcal{E}(y^-; \theta) - \mathcal{E}(y; \theta) - \underbrace{\sqrt{2\epsilon} \partial_i \mathcal{E}(y) z^i}_{\text{control variate}}). \quad (17)$$

In (17), y is the local coordinates of a sampled data point, $G(y)$ is the Riemannian metric, and y^- is obtained by running Riemannian LDs⁶ targeting $q_{1/2}$:

$$(y^-)^i = y^i + \epsilon \left(-g^{ij} \partial_j \frac{\mathcal{E}(y; \theta) + \log |G(y)|}{2} + \partial_k g^{ik} \right) + \sqrt{2\epsilon} z^i, \quad z \sim \mathcal{N}(0, G^{-1}(y)). \quad (18)$$

4.2 Learning Implicit AEs with Manifold Prior

Recently, there is a surge of interest in auto-encoding models with manifold-valued priors. In this section, we present a new training method for implicit auto-encoders with manifold priors, based on the above Riemannian score matching algorithm.

Formally, auto-encoders model the observed data by marginalizing out a latent code variable, $p(x; \theta) := \int p(x|z; \theta)p(z)dz$. To enable tractable learning, they define an additional ‘‘encoder’’ distribution $q(z|x; \phi)$. We will consider two types of auto-encoders:

⁶ While readers familiar with Riemannian Brownian motion may notice that (18) is only defined before the particle escapes the local chart, this is good enough for our purpose: we are only concerned with infinitesimal time, and escape probability approaches 0 as $\epsilon \rightarrow 0$. See Appendix C.

1. VAEs with implicit encoder, which maximizes $\mathbb{E}_{p(x)} \mathbb{E}_{q(z|x; \phi)} \log \frac{p(z)p(x|z; \theta)}{q(z|x; \phi)}$, the evidence lower bound. q is a reparameterized implicit distribution, i.e. for fixed x , $q(z|x; \phi)$ is defined as the push forward measure of a simple distribution $q_0(\epsilon)$, by a DNN that takes x and ϵ as input.
2. Wasserstein auto-encoders (WAEs), which minimizes the 1-Wasserstein distance between the model and data distributions by minimizing $\mathbb{E}_{p(x)} [\mathbb{E}_{q(z|x; \phi)} c(G(z; \theta), x)] + \lambda D(\tilde{q}(z), p(z))$, where $G(z; \theta)$ is the deterministic decoder, i.e. $p(x|z; \theta) = \delta(x - G(z; \theta))$; c is a user-specified reconstruction error, $\tilde{q}(z) := \int p(x)q(z|x)dx$ is the aggregated prior, λ is a hyperparameter, and D is an arbitrary divergence. We use the exclusive KL divergence as D .

Both objectives are intractable, as they include the entropy of a latent-space distribution with intractable density: $H[q(z|x; \phi)]$ for VAE, and $H[\tilde{q}(z)]$ for WAE. However, it is known that to obtain $\nabla_\phi H(q)$, it suffices to estimate the score function $\nabla_z \log q(z)$. Specifically, let $q(z; \phi)$ be the push forward of $p(\epsilon)$ by $f_q(\epsilon; \phi)$. Then we have

$$\nabla_\phi H[q(z)] = -\mathbb{E}_\epsilon [\nabla_z \log q(z) \nabla_\phi f_q(\epsilon; \phi)]. \quad (19)$$

Score estimation can be done by fitting an unnormalized model $\mathcal{E}(z)$ on the distribution $q(z)$, and approximating $\nabla_z \log q(z)$ above with $-\nabla_z \mathcal{E}(z)$. (For VAE, we will fit a conditional unnormalized model to approximate the conditional entropy.)

A variant of this idea is explored in Song et al. (2019), and outperforms existing learning algorithms for implicit AEs. As argued by (Shi et al., 2018; Li and Turner, 2018), this method is advantageous as it directly estimates the *score function* of the latent-space distribution, instead of obtaining gradient from density (ratio) estimations; the latter could lead to arbitrary variations in the gradient estimate.

When the latent variables are defined on an embedded manifold (e.g. hyper-spheres), we can no longer use the Euclidean score estimators to approximate the learning objective, as the entropy of the latent-space distribution w.r.t. the Lebesgue measure is usually undefined. However, we can still approximate the objective by doing score estimation inside the manifold: let $q(z)$ be the density w.r.t. the Hausdorff measure, and H be the corresponding relative entropy functional. Then (19) will still hold; see Appendix D. We can estimate the score function in (19) by with an unnormalized model on manifold, learned with the objective (17).

Wrapping up, we obtain an efficient algorithm to train auto-encoders with a manifold-valued prior.

5 RELATED WORK

Our work concerns scalable learning algorithms for unnormalized models. This is a longstanding problem in literature, and some of the previous work is discussed in Section 1. Other notable work includes noise contrastive estimation (Gutmann and Hyvärinen, 2010) and Parzen score matching (Raphael and Simoncelli, 2011). However, to our knowledge, they have not been applied to complex unnormalized models parameterized by DNNs.

Apart from the MVL formulation used in this work, there exists other work on the connection between learning objectives of unnormalized model and infinitesimal actions of sampling dynamics (or other processes):

- The *minimum probability flow framework* (Sohl-Dickstein et al., 2011) studies the slightly different objective $\lim_{\epsilon \rightarrow 0} \frac{1}{\epsilon} \text{KL}(p_0 \| p_\epsilon)$, where $\{p_t\}$ is the trajectory of the sampler. It recovers SM as a special instance, and leads to a tractable learning objective for discrete models.
- Many of the objective functions we have considered are also instances of the *Stein discrepancy*. This interpretation is helpful in establishing theoretical properties (Gorham et al., 2019) and deriving new objectives (Barp et al., 2019).
- Lyu (2009) observes a different connection between score matching and (derivative of) KL divergence; specifically they showed $D_F(p|q) = \frac{d}{dt} \text{KL}(p_t|q_t)|_{t=0}$, where $\{p_t\}, \{q_t\}$ are obtained by doing *Brownian motion* starting from p or q .

As those formulations have different motivations compared with ours, they do not lead to scalable learning objectives for continuous models.

6 EVALUATION

6.1 Synthetic Experiments

To demonstrate the proposed estimators have small bias and variance, we first evaluate them on low-dimensional synthetic data. We will also verify that our control variate in Section 3.3 improves the performance of CD-1 and DSM.

6.1.1 Approximations to SM

In this section, we evaluate our MVL approximation to the Euclidean score matching objective (7), as well as the variance-reduced DSM objective. An experiment evaluating the variance-reduced CD-1 objective is presented in Appendix E.1.2.

We evaluate the bias and variance of our estimators by comparing them to sliced score matching (SSM), an unbiased estimator for (7). We choose the data distribution p as the 2-D banana dataset from Wenliang et al. (2018), and the model distribution q_θ as an EBM trained on that dataset. We estimate the squared bias with a stochastic upper bound using 5×10^6 samples; see Appendix E.1.1 for details.

The results are shown in Figure 1. We can see that for both estimators, the bias is negligible at $\epsilon \leq 10^{-2}$. We further use a z-test to compare the mean of the two estimators (for $\epsilon = 6 \times 10^{-5}$) with the mean of SSM. The p value is 0.48 for our estimator and 0.19 for DSM, indicating there is no significant difference in either case. The variance of the estimators, with and without our control variate, are shown in Fig.1 right. As expected, the variance grows unbounded in absence of the control variate, and is approximately constant when it is added. From the scale of the variance, we can see that it is exactly this variance problem that causes the failure of the original DSM estimator.

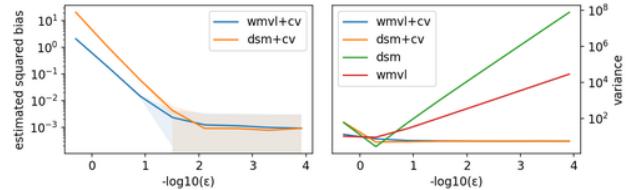


Figure 1: Estimated squared bias (left) and variance (right) of the approximate score matching objectives with varying step-size.

6.1.2 Density Estimation on Manifolds

We now evaluate our approximation to the Riemannian score matching objective, by learning neural energy-based models on S^1 and S^2 . The target distributions are mixtures of von-Mises-Fisher distributions. In Figure 2, we plot the log densities of the ground truth distribution as well as the learned model on S^1 . We can see the two functions matches closely, suggesting our method is suitable for density estimation on manifolds. Results on S^2 are similar and will be presented in E.1.3; detailed setups are deferred to Appendix E.1.1.

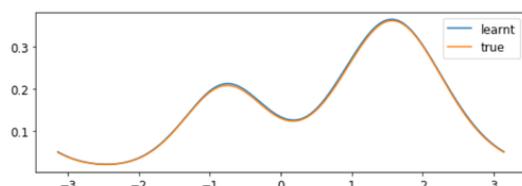


Figure 2: Density estimation on S^1 : learned energy vs ground truth in polar coordinates.

6.2 Implicit AEs with Manifold Prior

We now apply our method to train implicit auto-encoding models with manifold-valued prior. Experiment setups mainly follow Song et al. (2019); see Appendix E.2.

Note that there is an important difference from Song et al. (2019) in our implementation: for (conditional) score estimation, we parameterize an scalar energy function $\mathcal{E}_\theta(z)$ and use $\nabla_z \mathcal{E}_\theta(z)$ as the score estimate, while Song et al. (2019) directly parameterize a vector-value network $\psi_\theta(z)$. Since directly using a feed-forward network (FFN) for \mathcal{E} does not work well in practice, we parameterize the energy function as $\mathcal{E}_\theta(z) := z^\top \psi_\theta(z)$, where ψ is parameterized in the same way as Song et al. (2019). This can be seen as correcting an initial score approximation ψ to make it conservative. In addition to being conceptually desirable (as score functions are conservative fields), this approach leads to significant improvements in the WAE experiments.

6.2.1 Implicit VAEs

We apply our method to train hyperspherical VAEs (Davidson et al., 2018) with implicit encoders on the MNIST dataset. Our encoder and decoder architecture follows Song et al. (2019), with the exception that we normalize z so it lies on \mathcal{S}^n .

We consider $n_z \in \{8, 32\}$. Baseline methods include hyperspherical VAE with explicit encoders and Euclidean VAEs. We report the test log likelihood estimated with **annealed importance sampling** (Wu et al., 2016; Neal, 2001), as well as its standard deviation across 10 runs.

$n_z = 8$		$n_z = 32$	
Euc.	Sph.	Euc.	Sph.
Exp. 96.45 ± 0.10	95.47 ± 0.08	90.28 ± 0.37	91.32 ± 0.07
Imp. 95.84 ± 0.19	94.72 ± 0.16	90.33 ± 0.26	88.81 ± 0.14

Table 1: Negative log likelihood in the MNIST experiment. **Boldface** indicates the best result.

The results are summarized in Table 1. We can see that the implicit hyperspherical VAE trained with our method outperforms all other baselines. Interestingly, the explicit hyperspherical VAE could not match the performance of Euclidean VAE in higher dimensions. This is also observed in Davidson et al. (2018), who (incorrectly) conjectured that the hyperspherical prior is unsuitable in higher dimensions. From our results, we can see that the problem actually lies in the flexibility of variational posteriors. Our method thus unleashes the potential of VAEs with manifold-valued priors, and might lead to improvements in downstream tasks.

6.2.2 Hyperspherical WAEs

We first evaluate our method on MNIST. We use the uniform distribution as $p(z)$, and choose cross entropy as the reconstruction error. We choose $n_z = 8$. We use the encoder and decoder architecture in Song et al. (2019); the architecture of the energy network is also similar to their work. We report the Frechet Inception Distance (FID; Heusel et al., 2017).

As the choice of divergence measure in the WAE objective is arbitrary, there are several methods to train WAEs with manifold latent space: using the Jensen-Shannon divergence approximated with a GAN-like discriminator (WAE-GAN), and using the maximum mean discrepancy (MMD) divergence. We choose WAE-GAN as the baseline method, as it outperforms WAE-MMD in Tolstikhin et al. (2017). To demonstrate the utility of hyperspherical priors, we also compare with models using normal priors.

Meth od	Euc.	Sph .
WAE-GAN	24.59 ± 1.16	19.81 ± 1.13
Ours	23.80 ± 1.06	18.36 ± 0.73

Table 2: FID for WAEs on MNIST.

The FID scores are reported in Table 2. We can see that hyperspherical prior leads to better sample quality compared with Euclidean prior, and our method improves the training of WAEs.

To demonstrate our method scales to higher dimensions, we also train hyperspherical WAEs on CIFAR-10 and CelebA, with larger n_z . We find that our method is comparable or better than WAE-GAN and WAE-MMD; see Appendix E.2.1.

7 CONCLUSION

We present a scalable approximation to a general family of learning objectives for unnormalized models, based on a new connection between these objectives and gradient flows. Our method can be applied to manifold density estimation and training implicit auto-encoders with manifold priors.

ACKNOWLEDGEMENT

JZ is the corresponding author. We thank Chang Liu and Jixin Shi for comments. This work was supported by the National Key Research and Development Program of China (No. 2017YFA0700904), NSFC Project (Nos. 61620106010, U1811461), Beijing NSF Project (No. L172037), Beijing Academy of Artificial Intelligence (BAAI), a grant from Tsinghua Institute for Guo Qiang, and the NVIDIA NVAIL Program with GPU/DGX Acceleration.

References

- Barp, A., Briol, F.-X., Duncan, A. B., Girolami, M., and Mackey, L. (2019). Minimum stein discrepancy estimators.
- Boomsma, W., Mardia, K. V., Taylor, C. C., Ferkinghoff-Borg, J., Krogh, A., and Hamelryck, T. (2008). A generative, probabilistic model of local protein structure. *Proceedings of the National Academy of Sciences*, 105(26):8932–8937.
- Byrne, S. and Girolami, M. (2013). Geodesic monte carlo on embedded manifolds. *Scandinavian Journal of Statistics*, 40(4):825–845.
- Chen, C., Zhang, R., Wang, W., Li, B., and Chen, L. (2018). A unified particle-optimization framework for scalable bayesian sampling. *arXiv preprint arXiv:1805.11659*.
- Davidson, T. R., Falorsi, L., De Cao, N., Kipf, T., and Tomczak, J. M. (2018). Hyperspherical variational auto-encoders. *arXiv preprint arXiv:1804.00891*.
- Davis, J. C. and Sampson, R. J. (1986). *Statistics and data analysis in geology*, volume 646. Wiley New York et al.
- Federer, H. (2014). *Geometric measure theory*. Springer.
- Gorham, J., Duncan, A. B., Vollmer, S., and Mackey, L. (2019). Measuring sample quality with diffusions. *Annals of Applied Probability*.
- Gutmann, M. and Hyvärinen, A. (2010). Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 297–304.
- Häusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, pages 6626–6637.
- Hsu, E. P. (2002). *Stochastic analysis on manifolds*, volume 38. American Mathematical Soc.
- Hsu, E. P. (2008). A brief introduction to brownian motion on a riemannian manifold. *lecture notes*.
- Hitchinson, M. F. (1990). A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 19(2):433–450.
- Hyvärinen, A. (2005). Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(Apr):695–709.
- Hyvärinen, A. (2007). Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables. *IEEE Transactions on neural networks*, 18(5):1529–1531.
- Li, Y. and Turner, R. E. (2018). Gradient estimators for implicit models. In *International Conference on Learning Representations*.
- Liu, C., Zhu, J., and Song, Y. (2016a). Stochastic gradient geodesic mcmc methods. In *Advances in Neural Information Processing Systems*, pages 3009–3017.
- Liu, C., Zhou, J., Cheng, P., Zhang, R., Zhu, J., and Carin, L. (2019a). Understanding and accelerating particle-based variational inference. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 4082–4092, Long Beach, California USA. PMLR.
- Liu, C., Zhou, J., and Zhu, J. (2019b). Understanding mcmc dynamics as flows on the wasserstein space. *arXiv preprint arXiv:1902.00282*.
- Liu, Q. (2017). Stein variational gradient descent as gradient flow. In *Advances in neural information processing systems*, pages 3115–3123.
- Liu, Q., Lee, J., and Jordan, M. (2016b). A kernelized stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pages 276–284.
- Liu, Q. and Wang, D. (2016). Stein variational gradient descent: A general purpose bayesian inference algorithm. In *Advances in neural information processing systems*, pages 2378–2386.
- Liu, Q. and Wang, D. (2017). Learning deep energy models: Contrastive divergence vs. amortized mle. *arXiv preprint arXiv:1707.00797*.
- Lu, Y., Lu, J., and Nolen, J. (2019). Accelerating Langevin Sampling with Birth-death *arXiv e-prints*, page arXiv:1905.09863.
- Lyu, S. (2009). Interpretation and generalization of score matching. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 359–366. AUAI Press.
- Ma, Y.-A., Chen, T., and Fox, E. (2015). A complete recipe for stochastic gradient mcmc. In *Advances in Neural Information Processing Systems*, pages 2917–2925.
- Mardia, K. V., Kent, J. T., and Lahiri, A. K. (2016). Score matching estimators for directional distributions. *arXiv preprint arXiv:1604.08470*.
- Mathieu, E., Lan, C. L., Maddison, C. J., Tomioka, R., and Teh, Y. W. (2019). Continuous hierarchical representations with poincaré variational auto-encoders.

- In *Advances in neural information processing systems*.
- Miyato, T., Kataoka, T., Koyama, M., and Yoshida, Y. (2018). Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*.
- Movellan, J. R. (2007). A minimum velocity approach to learning. *unpublished*.
- Neal, R. M. (2001). Annealed importance sampling. *Statistics and computing*, 11(2):125–139.
- Otto, F. (2001). The geometry of dissipative evolution equations: the porous medium equation.
- Ovinnikov, I. (2019). Poincaré wasserstein autoencoder. *arXiv preprint arXiv:1901.01427*.
- Ramachandran, P., Zoph, B., and Le, Q. V. (2017). Searching for activation functions. *arXiv preprint arXiv:1710.05941*.
- Raphael, M. and Simoncelli, E. P. (2011). Least squares estimation without priors or supervision. *Neural computation*, 23(2):374–420.
- Ruiz, F. J and Titsias, M. K. (2019). A contrastive divergence for combining variational inference and mcmc. *arXiv preprint arXiv:1905.04062*.
- Saremi, S., Mehrljou, A., Schölkopf, B., and Hyvärinen, A. (2018). Deep energy estimator networks. *arXiv preprint arXiv:1805.08306*.
- Shi, J., Sun, S., and Zhu, J. (2018). A spectral approach to gradient estimation for implicit distributions. In *Proceedings of the 35th International Conference on Machine Learning*, pages 4651–4660.
- Sohl-Dickstein, J., Battaglino, P., and DeWeese, M. R. (2011). Minimum probability flow learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 905–912. Omnipress.
- Song, Y., Garg, S., Shi, J., and Ermon, S. (2019). Sliced score matching: A scalable approach to density and score estimation. *arXiv preprint arXiv:1905.07088*.
- Sriperumbudur, B., Fukumizu, K., Gretton, A., Hyvärinen, A., and Kumar, R. (2017). Density estimation in infinite dimensional exponential families. *The Journal of Machine Learning Research*, 18(1):1830–1888.
- Srivastava, A., Jermyn, I., and Joshi, S. (2007). Riemannian analysis of probability density functions with applications in vision. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.
- Taghvaei, A. and Mehta, P. (2019). Accelerated flow for probability distributions. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6076–6085, Long Beach, California, USA. PMLR.
- Tolstikhin, I., Bousquet, O., Gelly, S., and Schölkopf, B. (2017). Wasserstein auto-encoders. *arXiv preprint arXiv:1711.01558*.
- Villani, C. (2008). *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg.
- Vincent, P. (2011). A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674.
- Wenliang, L., Sutherland, D., Strathmann, H., and Gretton, A. (2018). Learning deep kernels for exponential family densities. *arXiv preprint arXiv:1811.08357*.
- Wenliang, L., Sutherland, D., Strathmann, H., and Gretton, A. (2019). Learning deep kernels for exponential family densities. In Chaudhuri, K. and Salakhutdinov, R., editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 6737–6746, Long Beach, California, USA. PMLR.
- Wu, Y., Burda, Y., Salakhutdinov, R., and Grosse, R. (2016). On the quantitative analysis of decoder-based generative models. *arXiv preprint arXiv:1611.04273*.
- Xifara, T., Sherlock, C., Livingstone, S., Byrne, S., and Girolami, M. (2014). Langevin diffusions and the metropolis-adjusted langevin algorithm. *Statistics & Probability Letters*, 91:14–19.
- Zhang, J., Zhang, R., and Chen, C. (2018). Stochastic particle-optimization sampling and the non-asymptotic convergence theory. *arXiv preprint arXiv:1809.01293*.

Supplementary Material

A Derivation of (13)-(15)

Denote $\tilde{x} := x + \sqrt{\epsilon}z$.

$$\hat{L}_{\text{dsm}} = \|x + \sqrt{\epsilon}z - \epsilon \nabla \mathcal{E}(x + \sqrt{\epsilon}z) - x\|^2 \quad (20)$$

$$= \epsilon \|z\|^2 + \epsilon^2 \|\nabla \mathcal{E}(\tilde{x})\|^2 - 2\epsilon^{3/2} \langle z, \nabla \mathcal{E}(\tilde{x}) \rangle \quad (21)$$

$$= \epsilon \|z\|^2 + \epsilon^2 \|\nabla \mathcal{E}(\tilde{x})\|^2 - 2\epsilon^{3/2} \langle z, \nabla \mathcal{E}(x) + (\nabla^2 \mathcal{E}(x))(\sqrt{\epsilon}z) + O(\epsilon) \rangle \quad (22)$$

$$= \underbrace{\epsilon^2 (\|\nabla \mathcal{E}(\tilde{x})\|^2 - 2z^\top (\nabla^2 \mathcal{E}(x))z)}_A + \underbrace{\epsilon \|z\|^2 - 2\epsilon^{3/2} z^\top \nabla \mathcal{E}(x) + o(\epsilon^2)}_B \quad (23)$$

Notice

$$\mathbb{E}_z(z^\top \nabla^2 \mathcal{E}(x)z) = \Delta \mathcal{E}(x)$$

which is known as the **Hitchinson's trick** (Hitchinson, 1990), so $\lim_{\epsilon \rightarrow 0} \epsilon^{-2} \mathbb{E}(A)$ is two times the Fisher divergence $D_F(p|q)$. But $\text{Var}(B) = O(\epsilon^2)$, so as $\epsilon \rightarrow 0$, the rescaled estimator $\epsilon^{-2} \hat{L}_{\text{dsm}}$ becomes unbiased with *infinite variance*; and subtracting (B) from (A) results in a finite-variance estimator.

B On SPOS and MVL

Notations In this section, let the parameter space be d -dimensional, and define $L_2(\rho \mathcal{X} \rightarrow \mathbb{R}^d)$ as the space of d -dimensional functions $\{f : \mathbb{E}_{\rho(x)} \|f(x)\|^2 < \infty\}$.

While in the main text, we identified the tangent space of $\mathcal{P}(\mathcal{X})$ as a subspace of $L_2(\rho \mathcal{X} \rightarrow \mathbb{R}^d)$ for clarity, here we use the equivalent definition $\mathcal{T}_p(\mathcal{P}(\mathcal{X})) := \{s \in L_2(\rho \mathcal{X} \rightarrow \mathbb{R}) : \mathbb{E}_p s = 0\}$ following (Otto, 2001). The two definitions are connected by the transform $s = -\nabla \cdot (\rho p)$ for $p \in L_2(\rho \mathcal{X} \rightarrow \mathbb{R}^d)$. Using the new definition, the differential of the KL divergence functional is then $(d\text{KL}_\phi)_p(s) := \int s(x) \log \frac{\rho(x)}{\phi(x)} dx$.

B.1 SPOS as Gradient Flow

$$\begin{aligned} X \rightarrow s \rightarrow p &= \langle \text{DKL}, X \rangle \\ &= \langle \log \rho/\phi, s \rangle \\ &= \langle \text{Dlog } \rho/\phi, p \rangle \end{aligned}$$

In this section, we give a formal derivation of SPOS as the gradient flow of the KL divergence functional, wrt a new metric.

Recall the SPOS sampler targeting distribution (with density) ϕ corresponds to the following density evolution:

$$\partial_t \rho_t = -\nabla \cdot (\rho_t(x) \underbrace{(\phi_{\rho_t, \phi}^*(x) + \alpha \nabla \log(\phi/\rho))}_{\nu_t(x)})$$

where $\alpha > 0$ is a hyperparameter, and

$$\phi_{\rho_t, \phi}^*(x) := \mathbb{E}_{\rho_t(x')} (S_\phi \otimes k)(x', x) := \mathbb{E}_{\rho_t(x')} [(\nabla_{x'} \log \phi(x'))k(x', x) + \nabla_{x'} k(x', x)]$$

is the SVGD update direction (Liu and Wang, 2016; Liu, 2017). Fix ρ , define the integral operator

$$K_\rho[f](x) := \mathbb{E}_{\rho(x')} k(x', x)f(x),$$

and define the tensor product operator $K_\rho^{\otimes d} : L^2(\mathcal{X} \rightarrow \mathbb{R}^d) \rightarrow L^2(\mathcal{X} \rightarrow \mathbb{R}^d)$ accordingly. Then the SVGD update direction satisfies

$$\phi_{\rho, \phi}^* = K_\rho^{\otimes d} [\nabla \log(\phi/\rho)], \quad (24)$$

which we will derive at the end of this subsection for completeness. (24) ==>

$$\nu_t(x) = (\alpha \text{Id} + K_\rho^{\otimes d}) [\nabla \log(\phi/\rho)]. \quad (25)$$

The rest of our derivation follows (Otto, 2001; Liu, 2017): consider the function space $\mathcal{H}_{\rho,\alpha} := \{(\alpha \text{Id} + K_{\rho_t}^{\otimes d})[\nabla h]\}$, where $h : \mathcal{X} \rightarrow \mathbb{R}$ is any square integrable and differentiable function. It connects to the tangent space of $\mathcal{P}(\mathcal{X})$ if we consider $s = -\nabla \cdot (\rho \tilde{p})$ for any $\tilde{p} \in \mathcal{H}_{\rho,\alpha}$. Define on $\mathcal{H}_{\rho,\alpha}$ the inner product

$$\langle f, g \rangle_{\mathcal{H}_{\rho,\alpha}} := \langle f, (\alpha \text{Id} + K_{\rho}^{\otimes d})^{-1}[g] \rangle_{L_2(\rho \mathcal{X} \rightarrow \mathbb{R}^d)}. \quad (26)$$

It then determines a Riemannian metric on the function space. For $\tilde{p} \in \mathcal{H}_{\rho,\alpha}$ and $s = -\nabla \cdot (\rho \tilde{p})$, by (25) we have

$$\langle \nu_t, \tilde{p} \rangle_{\mathcal{H}_{\rho,\alpha}} = \mathbb{E}_{\rho_t(x)} \langle \nabla \log(\phi/\rho_t)(x), \tilde{p}(x) \rangle = - \int \log \frac{\phi}{\rho_t} (\nabla \cdot (\tilde{p} \rho)) dx = -(d \text{KL}_{\phi})(s),$$

i.e. wrt the metric (26), SPOS is the gradient flow minimizing the KL divergence functional.

Derivation of (24) let $(\lambda_i, \psi_i)_{i=1}^\infty$ be its eigendecomposition (i.e. the Mercer representation). For $j \in [d]$

let $\psi_{i,j} := \psi_i \mathbf{e}_j$ where $\{\mathbf{e}_j\}_{j=1}^d$ is the coordinate basis in \mathbb{R}^d , so $\{\lambda^{-1/2} \psi_{i,j}\}$: orthonormal basis in $\mathcal{H}^{\otimes d}$.
 \iff

$$\begin{aligned} \langle \phi_{\rho,\phi}^*, \psi_{i,j} \rangle_{L_2(\rho)} &= \mathbb{E}_{\rho(x)} \mathbb{E}_{\rho(x')} \langle (\nabla_{x'} \log \phi(x')) k(x', x) + \nabla_{x'} k(x', x), \psi_{i,j}(x) \rangle \\ &= \mathbb{E}_{\rho(x')} [\langle \nabla_{x'} \log \phi(x'), (K_\rho[\psi_{i,j}])(x') \rangle + \nabla \cdot ((K_\rho[\psi_{i,j}])(x'))] \\ &=: \mathbb{E}_{\rho(x')} [S_\phi(K_\rho[\psi_{i,j}])(x')]. \end{aligned} \quad (27)$$

S_ϕ is known to satisfy the *Stein's identity*

$$\mathbb{E}_\rho S_\rho(g) = 0$$

for all $g \in \mathcal{H}$. Thus we can subtract $\mathbb{E}_\rho S_\rho(K_\rho[\psi_{i,j}])$ from the rhs of (27) without changing its value,
 \iff

$$\begin{aligned} &\mathbb{E}_{\rho(x')} [S_\phi(K_\rho[\psi_{i,j}])(x')] - \mathbb{E}_{\rho(x')} [S_\rho(K_\rho[\psi_{i,j}])(x')] \\ &= \mathbb{E}_{\rho(x')} \left[\left\langle \nabla_{x'} \log \frac{\phi(x')}{\rho(x')}, (K_\rho[\psi_{i,j}])(x') \right\rangle \right] \\ &= \lambda_k \mathbb{E}_{\rho(x')} \left[\left\langle \nabla_{x'} \log \frac{\phi(x')}{\rho(x')}, \psi_{i,j}(x') \right\rangle \right]. \end{aligned}$$

As the equality holds for all i, k , we completed the derivation of (24).

B.2 MVL Objective Derived from SPOS

By (25) and (26), the MVL objective derived from SPOS is

$$\|\text{grad}_\rho \text{KL}_{\phi}\|_{\mathcal{H}_{\rho,\alpha}}^2 = \langle \nabla \log(\phi/\rho_t), (\alpha \text{Id} + K^{\otimes d}) \nabla \log(\phi/\rho_t) \rangle_{L_2(\rho \mathcal{X} \rightarrow \mathbb{R}^d)}.$$

In the rhs above, the first term in the summation is the Fisher divergence, and the second is the kernelized Stein discrepancy (Liu et al., 2016b, Definition 3.2).

We note that a similar result for SVGD has been derived in (Liu and Wang, 2017), and our derivations connect to the observation that Langevin dynamics can be viewed as SVGD with a Dirac function kernel (thus SPOS also corresponds to SVGD with generalized-function-valued kernels).

C Justification of the Use of Local Coordinates in (17)

In this section, we prove in Proposition C.1 that the local coordinate representation lead to valid approximation to the MVL objective in the compact case. We also argue in Remark C.2 that the use of local coordinate does not lead to numerical instability.

Remark C.1. While a result more general than Proposition C.1 is likely attainable (e.g. by replacing compactness of \mathcal{X} with quadratic growth of the energy), this is out of the scope of our work; for our purpose, it is sufficient to note that the proposition covers manifolds like S^n , and the local coordinate issue will not exist in manifolds possessing a global chart, such as H^n .

Lemma C.1. (*Theorem 3.6.1 in (Hsu, 2002)*) For any manifold \mathcal{M} , $x \in \mathcal{M}$, and a normal neighborhood B of x , there exists constant $C > 0$ such that the first exit time τ from B , of the Riemannian Brownian motion starting from x , satisfies

$$P\left(\tau \leq \frac{C}{L}\right) \leq e^{-L/2}$$

for any $L \geq 1$.

Proposition C.1. Assume the data manifold \mathcal{X} is compact, and for all θ , $\mathcal{E}(\cdot; \theta)$ is in C^1 . Let $\tilde{L}_{\text{mvl_rld}}$ be defined as in (17), X_t following the true Riemannian Langevin dynamics targeting $q^{1/2}$. Then

$$\frac{1}{2} \lim_{\epsilon \rightarrow 0} \mathbb{E}(\tilde{L}_{\text{mvl_rld}}) = \frac{d}{dt} \mathbb{E}(\mathcal{E}(X_t)) \Big|_{t=0},$$

i.e. (17) recovers true WMVL objective.

Proof. By the tower property of conditional expectation, it suffices to prove the result when $P(X_0 = x) = 1$ for some x . Choose a normal neighborhood B centered at x such that B is contained by our current chart, and has distance from the boundary of the chart bounded by some $\delta > 0$. Let $C, \bar{\tau}$ be defined as in Lemma C.1. Recall the Riemannian LD is the sum of a drift and the Riemannian BM. Since \mathcal{X} is compact and \mathcal{E} is in C^1 , the drift term in the SDE will have norm bounded by some finite C . Thus the first exit time of the Riemannian LD is greater than $\min(\bar{\tau}, \delta/C) =: \tau$.

Let X_t follow the true Riemannian LD, $\bar{X}_t = X_t$ when $t < \tau$, and be such that $\mathcal{E}(\bar{X}_t) = 0$ afterwards.⁷ By Hsu (2008), until τ , \bar{X}_t follows the local coordinate representation of Riemannian LD (3), thus on the event $\{\epsilon \leq \tau\}$, X_ϵ would correspond to y^- in (18). As \mathcal{X} is compact, the continuous energy function \mathcal{E} is bounded by $|\mathcal{E}(\cdot)| \leq A$. Then for sufficiently small ϵ ,

$$\begin{aligned} \frac{1}{2} \mathbb{E}(\tilde{L}_{\text{mvl_rld}}) &= \frac{\mathbb{E}(\mathcal{E}(\bar{X}_\epsilon) - \mathcal{E}(X_0))}{\epsilon} = \frac{\mathbb{E}(\mathcal{E}(X_\epsilon) - \mathcal{E}(X_0))}{\epsilon} + \frac{\mathbb{E}(\mathcal{E}(\bar{X}_\epsilon) - \mathcal{E}(X_\epsilon))}{\epsilon} \\ &= \frac{\mathbb{E}(\mathcal{E}(X_\epsilon) - \mathcal{E}(X_0))}{\epsilon} + \frac{\mathbb{E}(-\mathcal{E}(X_\epsilon) \mathbf{1}_{\{\tau \leq \epsilon\}})}{\epsilon}. \end{aligned}$$

In the above the first term converges to $\frac{d}{dt} \mathbb{E}(\mathcal{E}(X_t)) \Big|_{t=0}$ as $\epsilon \rightarrow 0$, and $\left| \frac{\mathbb{E}(-\mathcal{E}(X_\epsilon) \mathbf{1}_{\{\tau \leq \epsilon\}})}{\epsilon} \right| \leq \frac{A \mathbb{P}(\tau \leq \epsilon)}{\epsilon} = \frac{A \mathbb{P}(\bar{\tau} \leq \epsilon)}{\epsilon} \leq \frac{A e^{-C/2\epsilon}}{\epsilon} \rightarrow 0$ when $\epsilon \rightarrow 0$. \square

Remark C.2. It is argued that simulating diffusion-based MCMC in local coordinates leads to numeric instabilities (Byrne and Girolami, 2013; Liu et al., 2016a). We stress that in our setting of approximating MVL objectives, this is not the case. The reason is that we only need to do a single step of MCMC, with arbitrarily small step-size. Therefore, we could use different step-size for each sample, based on the magnitude of g and $\log q$ in their locations. We can also choose different local charts for each sample, which is justified by the proposition above.

D Derivation of (19) in the Manifold Case

In this section we derive (19), where the latent-space distribution $q_\phi(z)$ is defined on a p -dimensional manifold embedded in some Euclidean space, and $H[q_\phi(z)]$ is the relative entropy w.r.t. the Hausdorff measure. The derivation is largely similar to the Euclidean case, and we only include it here for completeness.

(19) holds because

$$\begin{aligned} \nabla_\phi \mathbb{H}[q_\phi(z)] &\stackrel{(i)}{=} -\nabla_\phi \mathbb{E}_{p(\epsilon)} [\log q_\phi(f(\epsilon, \phi))] \\ &= -\mathbb{E}_{p(\epsilon)} [\nabla_\phi \log q_\phi(f(\epsilon, \phi))] \\ &= -\mathbb{E}_{p(\epsilon)} \left[\nabla_\phi \log q_\phi(z) \Big|_{z=f(\epsilon, \phi)} + \nabla_f \log q(f(\epsilon, \phi)) \nabla_\phi f(\epsilon, \phi) \right] \\ &\stackrel{(ii)}{=} -\mathbb{E}_{p(\epsilon)} [\nabla_z \log q_\phi(z) \nabla_\phi f(\epsilon, \phi)], \end{aligned}$$

⁷ This is conceptually similar to the standard augmentation used in stochastic process texts; from a algorithmic perspective it can be implemented by modifying the algorithm so that in the very unlikely event when y^- escapes the chart, we return 0 as the corresponding energy. We note that this is unnecessary for manifolds like S^n , since the charts can be extended to \mathbb{R}^d and hence $\tau = \infty$.

where (i) follows from Theorem 2.10.10 in Federer (2014), and (ii) follows from the same theorem as well as the fact that $\mathbb{E}_{q_\phi(z)}[\nabla_\phi \log q_\phi(z)] = \nabla_\phi \int q_\phi(z) dz = 0$.

E Experiment Details and Additional Results

Code will be available at <https://github.com/thu-ml/wmvl>.

E.1 Synthetic Experiments

E.1.1 Experiment Details

Experiment Details in Section 6.1.1 The (squared) bias is estimated as follows: denote the SSM estimator and ours as $\mathbb{E}_{p(x)\mathcal{N}(\epsilon|0,1)}[L_F^{\text{ssm}}(x; \epsilon)]$ and $\mathbb{E}_{p(x)\mathcal{N}(\epsilon|0,1)}[L_F^{\text{mvl}}(x; \epsilon)]$, respectively. One could verify that both methods estimate (7). Our estimate for the squared bias is now $\frac{1}{K} \sum_{k=1}^K \left(\frac{1}{M} \sum_{j=1}^M (L_F^{\text{ssm}}(x^{(k)}; \epsilon^{(j)}) - L_F^{\text{mvl}}(x^{(k)}; \epsilon^{(j)}))^2 \right)$, where $x^{(k)} \sim p(x)$, $\epsilon^{(j)} \sim \mathcal{N}(0, 1)$ are i.i.d. draws. The expectation of this estimate upper bounds the true squared bias by Cauchy's inequality, and the bias $\rightarrow 0$ as $K, M \rightarrow 0$. We choose $K = 100$, $M = 50000$ and plot the confidence interval. We also use these samples to estimate the variance of our estimator.

For the model distribution q , we choose an EBM as stated in the main text. The energy of the model is parameterized as follows: we parameterize a d -dimensional vector $\psi(x; \theta)$ using a feed-forward network, then return $x^\top \psi(x; \theta)$ as the energy function. This is inspired by the “score network” parameterization in (Song et al., 2019); we note that this choice has little influence on the synthetic experiments (and is merely chosen here for consistency), but leads to improved performance in the AE experiments. Finally, $\psi(x; \theta)$ is parameterized with 2 hidden layers and Swish activation (Ramachandran et al., 2017), and each layer has 100 units. We apply spectral normalization (Miyato et al., 2018) to the intermediate layers. We train the EBM for 400 iterations with our approximation to the SM objective, using a batch size of 200 and a learning rate of 4×10^{-3} . The choice of training objective is arbitrary; changing it to sliced SM does not lead to any notable difference, as is expected from this experiment.

The same procedure is applied to the DSM estimator.

Experiment Details in Section 6.1.2 For this experiment, the data distribution is chosen as

$$p(x) = 0.7p_{vM}(x|(0, 1), 2) + 0.3p_{vM}(x|(0.5, -0.5), 3),$$

where p_{vM} is the von Mises density

$$p_{vM}(x|\mu, \sigma) \propto e^{\frac{1}{\sigma^2} \cos(x-\mu)}.$$

For the model distribution, the energy function is parameterized with a feed-forward network, using the same score-network-inspired parameterization as in the last experiment. The network uses tanh activation and has 2 hidden layers, each layer with 100 units.

We generate 50,000 samples from $p(x)$ for training. We use full batch training and train for 6,000 iterations, using a learning rate of 5×10^{-4} . The step-size hyperparameter in the MVL approximation is set to 10^{-5} .

E.1.2 On the Variance Problem in CD-1

To verify our control variate also solves the variance issue in CD-1, we train EBMs using CD-1 with varying step-size, with and without our control variate, and compare the score matching loss to EBMs trained with our method as well as sliced score matching. We use a separate experiment for CD-1 since it only estimates the gradient of the score matching loss.

The SM loss is calculated using SSM on training set, and averaged over 3 separate runs. We use the cosine dataset in (Wenliang et al., 2018); the energy parameterization is the same as in Section 6.1.1. The results are shown in Figure 3. We can see that with the introduction of the control variate, CD-1 performs as well as other SM methods.

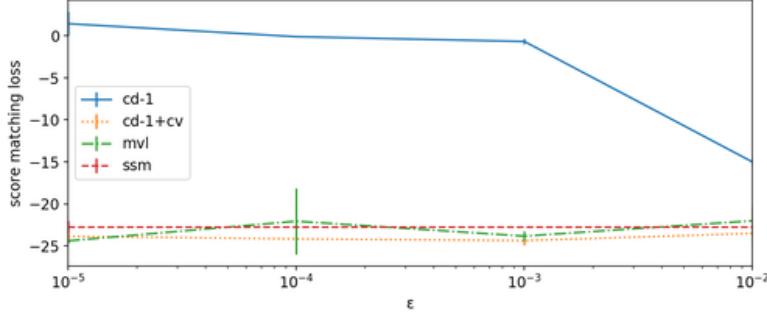


Figure 3: Score matching loss for different methods, with varying step-size. Lower is better.

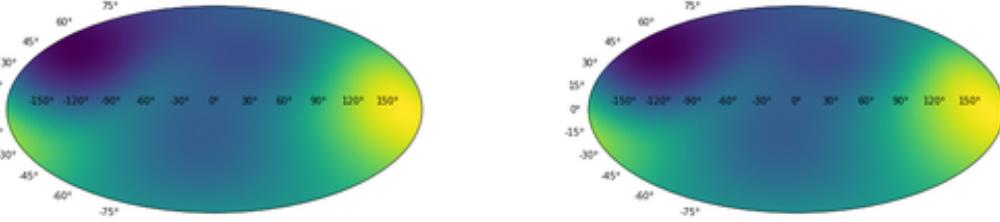


Figure 4: Mollweide projections of the ground truth (left) and learnt (right) energy functions on S^2 .

E.1.3 Learning EBMs on S^2

As a slightly more involved test case for our Riemannian score matching approximation, we consider learning EBMs on S^2 . The target distribution is a mixture of 4 von-Mises-Fisher distributions. The ground truth and learnt energy functions are plotted in Figure 4; we can see that our method leads to a good fit.

E.2 Auto-Encoder Experiments

In all auto-encoder experiments, setup follows from (Song et al., 2019) whenever possible. The only difference is that for score estimation, we parameterize the energy function, and use its gradient as the score estimate, as opposed to directly parameterizing the score function as done in (Song et al., 2019). This modification makes our method applicable; essentially, it corrects the score estimation in (Song et al., 2019) so that it constitutes a conservative field, which is a desirable property since score functions should be conservative.

For this reason, we re-implement all experiments for Euclidean-prior auto-encoders to ensure a fair comparison. The results are slightly worse than (Song et al., 2019) for the VAE experiment, but significantly better for WAE experiments. It should be also noted that in the VAE experiment, our implicit hyperspherical VAE result is still better than the implicit Euclidean VAE result reported in (Song et al., 2019).

VAE Experiment The (conditional) energy function in this experiment is parameterized using the score-net-inspired method described in Appendix E.1.1, with a feed-forward network. The network has 2 hidden layers, each with 256 hidden units. We use tanh activation for the network, and do not apply spectral normalization. When training the energy network, we add a L2 regularization term for the energy scale, with coefficient 10^{-4} . The coefficient is determined by grid search on $\{10^{-3}, 10^{-4}, 10^{-5}\}$, using AIS-estimated likelihood on a heldout set created from the training set. The step-size of the MVL approximation is set to 10^{-3} ; we note that the performance is relatively insensitive w.r.t. the step-size inside the range of $[10^{-4}, 10^{-2}]$, as suggested by the synthetic experiment. Outside this range, using a smaller step-size makes the result worse, presumably due to floating point errors.

For implicit models, the test likelihood is computed with annealed importance sampling, using 1,000 intermediate distributions, following (Song et al., 2019). The transition operator in AIS is HMC for Euclidean-space latents, and Riemannian LD for hyperspherical latents.

The training setup follows from (Song et al., 2019): for all methods, we train for 100,000 iterations using RMSProp

use a batch size of 128, and a learning rate of 10^{-3} .

WAE Experiment on MNIST For our method, the energy network is parameterized in the same way as in the VAE experiments. When training the energy network, we use a step-size of 10^{-3} , and apply L2 regularization on the energy scale with coefficient 10^{-5} . For the WAE-GAN baseline, we parameterize the GAN discriminator as a feed-forward network with 2 hidden layers, each with 256 units. We use tanh activation, and apply L2 regularization with coefficient 10^{-5} . All models are trained for 200,000 iterations using RMSProp, using a batch size of 128, and a learning rate of 10^{-3} . The Lagrange multiplier hyperparameter λ in the WAE objective is fixed at 10. FID scores are calculated using the implementation in (Heusel et al., 2017).

Sampled Generations in the Auto-encoder Experiments See Figure 7 - 9.

E.2.1 WAE Experiments in Higher Dimensions

In this section, we present results of hyperspherical WAEs on CIFAR-10 and CelebA, with larger n_z .

For CelebA we follow the setup in Song et al. (2019): $n_z = 32$, RMSProp, learning rate 10^{-4} , train for 100,000 iterations. In addition, we apply spectral normalization and L2 regularization with coefficient 10^{-4} . The step-size in the MVL approximation is set to 10^{-4} . The FID scores, averaged over 5 runs, are 50.82 ± 0.50 for our method and 51.20 ± 0.59 for WAE-GAN.

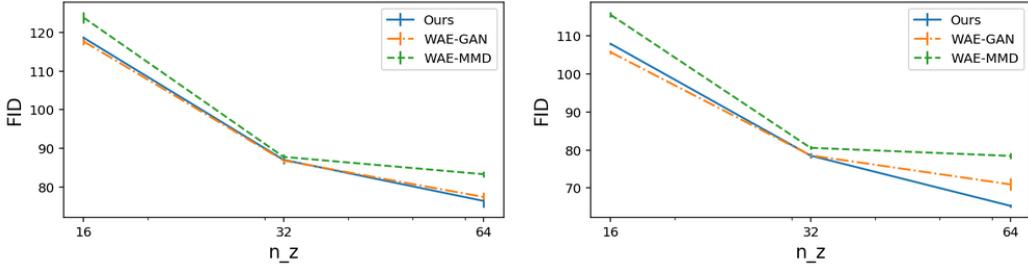


Figure 5: FID on CIFAR-10, with varying n_z . Left: after 10^5 iterations; right: after 2×10^5 iterations.

For CIFAR-10, we modify the auto-encoder architecture and remove one scaling block to account for its lower resolution. We do not use spectral normalization which leads to slightly worse results. The FID scores for varying n_z are presented in Figure 5, where we can see our method compares favorably to all baselines.