Exponential Family Estimation via Adversarial Dynamics Embedding

*Bo Dai[1], *Zhen Liu[2], *Hanjun Dai[1], Niao He[3], Arthur Gretton[4], Le Song[5,6], Dale Schuurmans[1,7]
[1]Google Research, Brain Team, [2]Mila, University of Montreal,
[3]University of Illinois at Urbana Champaign, [4]University College London,
[5]Georgia Institute of Technology, [6]Ant Financial, [7]University of Alberta

Abstract

We present an efficient algorithm for maximum likelihood estimation (MLE) of exponential family models, with a general parametrization of the energy function that includes neural networks. We exploit the primal-dual view of the MLE with a *kinetics augmented model* to obtain an estimate associated with an *adversarial* dual sampler. To represent this sampler, we introduce a novel neural architecture, *dynamics embedding*, that generalizes Hamiltonian Monte-Carlo (HMC). The proposed approach inherits the flexibility of HMC while enabling tractable entropy estimation for the augmented model. By learning both a dual sampler and the primal model simultaneously, and sharing parameters between them, we obviate the requirement to design a separate sampling procedure once the model has been trained, leading to more effective learning. We show that many existing estimators, such as contrastive divergence, pseudo/composite-likelihood, score matching, minimum Stein discrepancy estimator, non-local contrastive objectives, noise-contrastive estimation, and minimum probability flow, are special cases of the proposed approach, each expressed by a different (fixed) dual sampler. An empirical investigation shows that adapting the sampler during MLE can significantly improve on state-of-the-art estimators[1].

1 Introduction

The exponential family is one of the most important classes of distributions in statistics and machine learning, encompassing undirected graphical models (Wainwright and Jordan, 2008) and energy-based models (LeCun et al., 2006; Wu et al., 2018), which include, for example, Markov random fields (Kinderman and Snell, 1980), conditional random fields (Lafferty et al., 2001) and language models (Mnih and Teh, 2012). Despite the flexibility of this family and the many useful properties it possesses (Brown, 1986), most such distributions are intractable because the partition function does not possess an analytic form. This leads to difficulty in evaluating, sampling and learning exponential family models, hindering their application in practice. In this paper, we consider a longstanding question:

> *Can a simple yet effective algorithm be developed for estimating general exponential family distributions?*

There has been extensive prior work addressing this question. Many approaches focus on approximating maximum likelihood estimation (MLE), since it is well studied and known to possess desirable statistical properties, such as consistency, asymptotic unbiasedness, and asymptotic normality (Brown, 1986). One prominent example is contrastive divergence (CD) (Hinton, 2002) and its variants (Tieleman and Hinton, 2009; Du and Mordatch, 2019). It approximates the gradient of the log-likelihood by a stochastic estimator that uses samples generated from a few Markov chain Monte Carlo (MCMC) steps. This approach has two shortcomings: first and foremost, the stochastic gradient is *biased*,

*indicates equal contribution. Email: `{bodai, hadai}@google.com, zhen.liu.2@umontreal.ca`.

[1] The code repository is available at https://github.com/lzzcd001/ade-code.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

which can lead to poor estimates; second, CD and its variants require careful design of the MCMC transition kernel, which can be challenging.

Given these difficulties with MLE, numerous learning criteria have been proposed to avoid the partition function. Pseudo-likelihood estimators (Besag, 1975) approximate the joint distribution by the product of conditional distributions, each of which only represents the distribution of a single random variable conditioned on the others. However, the the partition function of each factor is still generally intractable. Score matching (Hyvärinen, 2005) minimizes the Fisher divergence between the empirical distribution and the model. Unfortunately, it requires third order derivatives for optimization, which becomes prohibitive for large models (Kingma and LeCun, 2010; Li et al., 2019). Noise-contrastive estimation (Gutmann and Hyvärinen, 2010) recasts the problem as ratio estimation between the target distribution and a pre-defined auxiliary distribution. However, the auxiliary distribution must cover the support of the data with an analytical expression that still allows efficient sampling; this requirement is difficult to satisfy in practice, particularly in high dimensional settings. Minimum probability flow (Sohl-Dickstein et al., 2011) exploits the observation that, ideally, the empirical distribution will be the stationary distribution of transition dynamics defined under an optimal model. The model can then be estimated by matching these two distributions. Even though this idea is inspiring, it is challenging to construct appropriate dynamics that yield efficient learning.

In this paper, we introduce a novel algorithm, *Adversarial Dynamics Embedding (ADE)*, that directly approximates the MLE while achieving computational and statistical efficiency. Our development starts with the *primal-dual* view of the MLE (Dai et al., 2019) that provides a natural objective for jointly learning both a sampler and a model, as a remedy for the expensive and biased MCMC steps in the CD algorithm. To parameterize the dual distribution, Dai et al. (2019) applies a naive transport mapping, which makes entropy estimation difficult and requires learning an extra auxiliary model, incurring additional computational and memory cost.

We overcome these shortcomings by considering a different approach, inspired by the properties of Hamiltonian Monte-Carlo (HMC) (Neal, 2011):

i) HMC forms a stationary distribution with *independent* potential and kinetic variables;
ii) HMC can approximate the exponential family *arbitrarily closely*.

As in HMC, we consider an *augmented model* with latent kinetic variables in Section 3.1, and introduce a novel neural architecture in Section 3.2, called *dynamics embedding*, that mimics sampling and represents the dual distribution via parameters of the primal model. This approach shares with HMC the advantage of a *tractable* entropy function for the augmented model, while enriching the flexibility of sampler without introducing extra parameters. In Section 3.3 we develop a $\max$-$\min$ objective that allows the shared parameters in primal model and dual sampler to be learned simultaneously, which improves computational and sample efficiency. We further show that the proposed estimator subsumes CD, pseudo-likelihood, score matching, non-local contrastive objectives, noise-contrastive estimation, and minimum probability flow as special cases with hand-designed dual samplers in Section 4. Finally, in Section 5 we find that the proposed approach can outperform current state-of-the-art estimators in a series of experiments.

2 Preliminaries

Exponential family and energy-based model The natural form of the exponential family over $\Omega \subset \mathbb{R}^d$ is defined as

$$p_{f'}\left(x\right) = \exp\left(f'(x) - \log p_0\left(x\right) - A_{p_0}\left(f'\right)\right), \ \ A_{p_0}\left(f'\right) := \log \int_\Omega \exp\left(f'\left(x\right)\right) p_0\left(x\right) dx, \quad (1)$$

where $f'\left(x\right) = w^\top \phi_\varpi\left(x\right)$. The sufficient statistic $\phi_\varpi\left(\cdot\right) : \Omega \rightarrow \mathbb{R}^k$ can be any general parametric model, *e.g.*, a neural network. The $\left(w, \varpi\right)$ are the parameters to be learned from observed data. The exponential family definition (1) includes the energy-based model (LeCun et al., 2006) as a special case, by setting $f'\left(x\right) = \phi_\varpi\left(x\right)$ with $k = 1$, which has been generalized to the infinite dimensional case (Sriperumbudur et al., 2017). The $p_0\left(x\right)$ is fixed and covers the support Ω, which is usually unknown in practical high-dimensional problems. Therefore, we focus on learning $f\left(x\right) = f'\left(x\right) - \log p_0\left(x\right)$ jointly with $p_0\left(x\right)$, which is more difficult: in particular, the doubly dual embedding approach (Dai et al., 2019) is no longer applicable.

Given a sample $\mathcal{D} = \left[x_i\right]_{i=1}^N$ and denoting $f \in \mathcal{F}$ as the valid parametrization family, an exponential family model can be estimated by maximum log-likelihood, *i.e.*,

$$\max_{f \in \mathcal{F}} \ L\left(f\right) := \widehat{\mathbb{E}}_{\mathcal{D}}\left[f\left(x\right)\right] - A\left(f\right), \ \ A\left(f\right) = \log \int_\Omega \exp\left(f\left(x\right)\right) dx, \quad (2)$$

with gradient $\nabla_f L(f) = \widehat{\mathbb{E}}_{\mathcal{D}}[\nabla_f f(x)] - \mathbb{E}_{p_f(x)}[\nabla_f f(x)]$. Since $A(f)$ and $\mathbb{E}_{p_f(x)}[\nabla_f f(x)]$ are both intractable, solving the MLE for a general exponential family model is very difficult.

Dynamics-based MCMC Dynamics-based MCMC is a general and effective tool for sampling. The idea is to represent the target distribution as the solution to a set of (stochastic) differential equations, which allows samples from the target distribution to be obtained by simulating along the dynamics defined by the differential equations.

HMC (Neal, 2011) is a representative algorithm in this category, which exploits the well-known Hamiltonian dynamics. Specifically, given a target distribution $p_f(x) \propto \exp(f(x))$, the Hamiltonian is defined as $\mathcal{H}(x, v) = -f(x) + k(v)$, where $k(v) = \frac{1}{2}v^\top v$ is the kinetic energy. The Hamiltonian dynamics generate (x, v) over time t by following

$$\left[\frac{dx}{dt}, \frac{dv}{dt}\right] = [\partial_v \mathcal{H}(x, v), -\partial_x \mathcal{H}(x, v)] = [v, \nabla_x f(x)]. \tag{3}$$

Asymptotically as $t \to \infty$, x visits the underlying space according to the target distribution. In practice, to reduce discretization error, an acceptance-rejection step is introduced. The finite-step dynamics-based MCMC sampler can be used for approximating $\mathbb{E}_{p_f(x)}[\nabla_f f(x)]$ in $\nabla_f L(f)$, which leads to the CD algorithm (Hinton, 2002; Zhu and Mumford, 1998).

Primal-dual view of MLE The Fenchel duality of $A(f)$ has been exploited (Rockafellar, 1970; Wainwright and Jordan, 2008; Dai et al., 2019) as another way to address the intractability of the log-partition function.

Theorem 1 (Fenchel dual of log-partition (Wainwright and Jordan, 2008)) *Let* $H(q) := -\int_\Omega q(x)\log q(x)dx$. *Then:*

$$A(f) = \max_{q\in\mathcal{P}} \ \langle q(x), f(x)\rangle + H(q), \quad p_f(x) = \operatorname{argmax}_{q\in\mathcal{P}} \ \langle q(x), f(x)\rangle + H(q), \tag{4}$$

where $\mathcal{P}$ *denotes the space of distributions and* $\langle f, g\rangle = \int_\Omega f(x) g(x)\,dx$.

Plugging the Fenchel dual of $A(f)$ into the MLE (2), we arrive at a max-min reformulation

$$\max_{f\in\mathcal{F}} \min_{q\in\mathcal{P}} \ \widehat{\mathbb{E}}_{\mathcal{D}}[f(x)] - \mathbb{E}_{q(x)}[f(x)] - H(q), \tag{5}$$

which bypasses the explicit computation of the partition function. Another byproduct of the primal-dual view is that the dual distribution can be used for inference, however in vanila estimators this usually requires expensive sampling algorithms.

The dual sampler $q(\cdot)$ plays a vital role in the primal-dual formulation of the MLE in (5). To achieve better performance, we have several principal requirements in parameterizing the dual distribution:

i) the parametrization family needs to be *flexible* enough to achieve small error in solving the inner minimization problem;
ii) the entropy of the parametrized dual distribution should be *tractable*.

Moreover, as shown in (4) in Theorem 1, the optimal dual sampler $q(\cdot)$ is determined by primal potential function $f(\cdot)$. This leads to the third requirement:

iii) the parametrized dual sampler should *explicitly incorporate* the primal model f.

Such a dependence can potentially reduce both the memory and learning sample complexity.

A variety of techniques have been developed for distribution parameterization, such as reparametrized latent variable models (Kingma and Welling, 2014; Rezende et al., 2014), transport mapping (Goodfellow et al., 2014), and normalizing flow (Rezende and Mohamed, 2015; Dinh et al., 2017; Kingma et al., 2016). However, none of these satisfies the requirements of flexibility and a tractable density simultaneously, nor do they offer a principled way to couple the parameters of the dual sampler with the primal model.

3 Adversarial Dynamics Embedding

By augmenting the original exponential family with kinetic variables, we can parametrize the dual sampler with a *dynamics embedding* that satisfies all three requirements without effecting the MLE, allowing the primal potential function and dual sampler to both be trained adversarially. We start with the embedding of classical Hamiltonian dynamics (Neal, 2011; Caterini et al., 2018) for the dual sampler parametrization, as a concrete example, then discuss its generalization in latent space and the stochastic Langevin dynamics embedding. This technique is extended to other dynamics, with their own advantages, in Appendix B.

3.1 Primal-Dual View of Augmented MLE

As noted, it is difficult to find a parametrization of $q\left(x\right)$ in (5) that simultaneously satisfies all three requirements. Therefore, instead of directly tackling (5) in the original model, and inspired by HMC, we consider the augmented exponential family $p\left(x,v\right)$ with an auxiliary momentum variable, *i.e.*,

$$p\left(x,v\right)=\frac{\exp\left(f\left(x\right)-\frac{\lambda}{2}v^\top v\right)}{Z\left(f\right)},\quad Z\left(f\right)=\int\exp\left(f\left(x\right)-\frac{\lambda}{2}v^\top v\right)dxdv. \tag{6}$$

The MLE of such a model can be formulated as

$$\max_f L\left(f\right):=\widehat{\mathbb{E}}_{x\sim\mathcal{D}}\left[\log\int p\left(x,v\right)dv\right]=\widehat{\mathbb{E}}_{x\sim\mathcal{D}}\mathbb{E}_{p\left(v|x\right)}\left[f\left(x\right)-\frac{\lambda}{2}v^\top v-\log p\left(v|x\right)\right]-\log Z\left(f\right) \tag{7}$$

where the last equation comes from true posterior $p\left(v|x\right)=\mathcal{N}\left(0,\lambda^{-\frac{1}{2}}I\right)$ due to the independence of x and v. This independence also induces the equivalent MLE as proved in Appendix A.

Theorem 2 (Equivalent MLE) *The MLE of the augmented model is the same as the original MLE.*

Applying the Fenchel dual to $Z\left(f\right)$ of the augmented model (6), we derive a primal-dual formulation of (7), leading to the objective,

$$L\left(f\right)\propto\min_{q\left(x,v\right)\in\mathcal{P}}\ \widehat{\mathbb{E}}_{x\sim\mathcal{D}}\left[f\left(x\right)\right]-\mathbb{E}_{q\left(x,v\right)}\left[f\left(x\right)-\frac{\lambda}{2}v^\top v-\log q\left(x,v\right)\right]. \tag{8}$$

The $q\left(x,v\right)$ in (8) contains momentum v as the latent variable. One can also exploit the latent variable model for $q\left(x\right)=\int q\left(x|v\right)q\left(v\right)dv$ in (5). However, the $H\left(q\right)$ in (5) requires marginalization, which is intractable in general, and usually estimated through variational inference with the introduction of an extra posterior model $q\left(v|x\right)$. Instead, by considering the specifically designed augmented model, (8) eliminates these extra variational steps.

Similarly, one can consider the latent variable augmented model with multiple momenta, *i.e.*, $p\left(x,\left\{v^i\right\}_{i=1}^T\right)=\frac{\exp\left(f\left(x\right)-\sum_{i=1}^T\frac{\lambda_i}{2}\left\|v^i\right\|_2^2\right)}{Z\left(f\right)}$, leading to the optimization

$$L\left(f\right)\propto\min_{q\left(x,\left\{v^i\right\}_{i=1}^T\right)\in\mathcal{P}}\widehat{\mathbb{E}}_{x\sim\mathcal{D}}\left[f\left(x\right)\right]-\mathbb{E}_{q\left(x,\left\{v^i\right\}_{i=1}^T\right)}\left[f\left(x\right)-\sum_{i=1}^T\frac{\lambda_i}{2}\left\|v^i\right\|_2^2-\log q\left(x,\left\{v^i\right\}_{i=1}^T\right)\right]. \tag{9}$$

3.2 Representing Dual Sampler via Primal Model

We now introduce the Hamiltonian dynamics embedding to represent the dual sampler $q\left(\cdot\right)$, as well as its generalization and special instantiation that satisfy all three of the principal requirements.

The vanilla HMC is derived by discretizing the Hamiltonian dynamics (3) with a leapfrog integrator. Specifically, in a single time step, the sample $\left(x,v\right)$ moves towards $\left(x',v'\right)$ according to

$$\left(x',v'\right)=\mathbf{L}_{f,\eta}\left(x,v\right):=\left(\begin{array}{c}v^{\frac{1}{2}}=v+\frac{\eta}{2}\nabla_x f\left(x\right)\\ x'=x+\eta v^{\frac{1}{2}}\\ v'=v^{\frac{1}{2}}+\frac{\eta}{2}\nabla_x f\left(x'\right)\end{array}\right), \tag{10}$$

where η is defined as the leapfrog stepsize. Let us denote the one-step leapfrog as $\left(x',v'\right)=\mathbf{L}_{f,\eta}\left(x,v\right)$ and assume the $\left(x^0,v^0\right)\sim q_\theta^0\left(x,v\right)$. After T iterations, we obtain

$$\left(x^T,v^T\right)=\mathbf{L}_{f,\eta}\circ\mathbf{L}_{f,\eta}\circ\ldots\circ\mathbf{L}_{f,\eta}\left(x^0,v^0\right). \tag{11}$$

Note that this can be viewed as a neural network with a special architecture, which we term *Hamiltonian (HMC) dynamics embedding*. Such a representation explicitly characterizes the dual sampler by the primal model, *i.e.*, the potential function f, meeting the dependence requirement.

The flexibility of the distributions HMC embedding actually is ensured by the nature of the dynamics-based samplers. In the limiting case, the proposed neural network (11) reduces to a gradient flow, whose stationary distribution is exactly the model distribution:

$$p\left(x,v\right)=\operatorname{argmax}_{q\left(x,v\right)\in\mathcal{P}}\mathbb{E}_{q\left(x,v\right)}\left[f\left(x\right)-\frac{\lambda}{2}v^\top v-\log q\left(x,v\right)\right].$$

The approximation strength of the HMC embedding is formally justified as follows:

Theorem 3 (HMC embeddings as gradient flow) *In continuous time,* i.e. *with infinitesimal stepsize $\eta\to0$, the density of particles $\left(x^t,v^t\right)$, denoted $q^t\left(x,v\right)$, follows the Fokker-Planck equation*

$$\frac{\partial q^t\left(x,v\right)}{\partial t}=\nabla\cdot\left(q^t\left(x,v\right)G\nabla\mathcal{H}\left(x,v\right)\right), \tag{12}$$

with $G=\begin{bmatrix}0&\mathbf{I}\\-\mathbf{I}&0\end{bmatrix}$, which has a stationary distribution $p\left(x,v\right)\propto\exp\left(-\mathcal{H}\left(x,v\right)\right)$ with the marginal distribution $p(x)\propto\exp\left(f(x)\right)$.

Details of the proofs are given in Appendix A. Note that this stationary distribution result is an instance of the more general dynamics described in Ma et al. (2015), showing the flexility of the induced distributions. As demonstrated in Theorem 3, the neural parametrization formed by the HMC embedding is able to well approximate an exponential family distribution on continuous variables.

Remark (Generalized HMC dynamics in latent space) The leapfrog operation in vanilla HMC works directly in the original observation space, which could be high-dimensional and noisy. We generalize the leapfrog update rule to the latent space and form a new dynamics as follows,

$$\left(x', v'\right) = \mathbf{L}_{f,\eta,S,g}\left(x, v\right) := \left(\begin{array}{c} v^{\frac{1}{2}} = v \odot \exp\left(S_v\left(\nabla_x f\left(x\right), x\right)\right) + \frac{\eta}{2} g_v\left(\nabla_x f\left(x\right), x\right) \\ x' = x \odot \exp\left(S_x\left(v^{\frac{1}{2}}\right)\right) + \eta g_x\left(v^{\frac{1}{2}}\right) \\ v' = v^{\frac{1}{2}} \odot \exp\left(S_v\left(\nabla_x f\left(x'\right), x'\right)\right) + \frac{\eta}{2} g_v\left(\nabla_x f\left(x'\right), x'\right) \end{array}\right), \tag{13}$$

where $v \in \mathbb{R}^l$ denote the momentum evolving space and $\odot$ denotes element-wise product. Specifically, the terms $S_v\left(\nabla_x f\left(x\right), x\right)$ and $S_x\left(v^{\frac{1}{2}}\right)$ rescale v and x coordinatewise. The term $g_v\left(\nabla_x f\left(x\right), x\right) \mapsto \mathbb{R}^l$ can be understood as projecting the gradient information to the essential latent space where the momentum is evolving. Then, for updating x, the latent momentum is projected back to original space via $g_x\left(v^{\frac{1}{2}}\right) \mapsto \Omega$. With these generalized leapfrog updates, the dynamical system avoids operating in the high-dimensional noisy input space, and becomes more computationally efficient. We emphasize that the proposed generalized leapfrog parametrization (13) is different from the one used in Levy et al. (2018), which is inspired from the real-NVP flow (Dinh et al., 2017).

By the generalized HMC embedding (13), we have a flexible layer $\left(x', v'\right) = \mathbf{L}_{f,\eta,S,g}\left(x, v\right)$, where $\left(S_v, S_x, g_v, g_x\right)$ will be learned in addition to the stepsize. Obviously, the classic HMC layer $\mathbf{L}_{f,\eta,M}\left(x, v\right)$ is a special case of $\mathbf{L}_{f,\eta,S,g}\left(x, v\right)$ by setting $\left(S_v, S_x\right)$ to zero and $\left(g_v, g_f\right)$ to identity functions.

Remark (Stochastic Langevin dynamics) The stochastic Langevin dynamics can also be recovered from the leapfrog step by resampling momentum in every step. Specifically, the sample $\left(x, \xi\right)$ moves according to

$$\left(x', v'\right) = \mathbf{L}^{\xi}_{f,\eta}\left(x\right) := \left(\begin{array}{c} v' = \xi + \frac{\eta}{2}\nabla_x f\left(x\right) \\ x' = x + v' \end{array}\right), \text{ with } \xi \sim q_\theta\left(\xi\right). \tag{14}$$

Hence, stochastic Langevin dynamics resample ξ to replace the momentum in leapfrog (10), ignoring the accumulated gradients. By unfolding T updates, we obtain

$$\left(x^T, \left\{v^i\right\}_{i=1}^T\right) = \mathbf{L}^{\xi^{T-1}}_{f,\eta} \circ \mathbf{L}^{\xi^{T-2}}_{f,\eta} \circ \ldots \circ \mathbf{L}^{\xi^0}_{f,\eta}\left(x^0\right) \tag{15}$$

as the derived neural network. Similarly, we can also generalize the stochastic Langevin updates $\mathbf{L}^{\xi}_{f,\eta}$ to a low-dimension latent space by introducing $g_v\left(\nabla_x f\left(x\right), x\right)$ and $g_x\left(v'\right)$ correspondingly.

One of the major advantages of the proposed distribution parametrization is its density value is also tractable, leading to tractable entropy estimation in (8) and (9). In particular, we have the following,

Theorem 4 (Density value evaluation) *If* $\left(x^0, v^0\right) \sim q^0_\theta\left(x, v\right)$*, after* T *vanilla HMC steps* (10)*, then*

$$q^T\left(x^T, v^T\right) = q^0_\theta\left(x^0, v^0\right). \tag{16}$$

For $\left(x^T, v^T\right)$ *from the generalized leapfrog steps* (13)*, we have*

$$q^T\left(x^T, v^T\right) = q^0_\theta\left(x^0, v^0\right) \prod_{t=1}^T \left(\Delta_x\left(x^t\right) \Delta_v\left(v^t\right)\right), \tag{17}$$

where $\Delta_x\left(x^t\right)$ *and* $\Delta_v\left(v^t\right)$ *denote*

$$\Delta_x\left(x^t\right) = \left|\det\left(\operatorname{diag}\left(\exp\left(2S_v\left(\nabla_x f\left(x^t\right), x^t\right)\right)\right)\right|, \Delta_v\left(v^t\right) = \left|\det\left(\operatorname{diag}\left(\exp\left(S_x\left(v^{\frac{1}{2}}\right)\right)\right)\right)\right|. \tag{18}$$

For $\left(x^T, \left\{v^i\right\}_{i=1}^T\right)$ *from the Langevin dynamics* (14) *with* $\left(x^0, \left\{\xi^i\right\}_{i=0}^{T-1}\right) \sim q^0_\theta\left(x, \xi\right) \prod_{i=i}^{T-1} q_{\theta_i}\left(\xi\right)$*, we have*

$$q^T\left(x^T, \left\{v^i\right\}_{i=1}^T\right) = q^0_\theta\left(x^0, \xi^0\right) \prod_{i=1}^{T-1} q_{\theta_i}\left(\xi^i\right). \tag{19}$$

The proof of Theorem 4 can be found in Appendix A.

The proposed dynamics embedding satisfies all three requirements: it defines a flexible family of distributions with computable entropy; and couples the learning of the dual sampler with the primal model, leading to memory and sample efficient learning algorithms, as we introduce in next section.

3.3 Coupled Model and Sampler Learning

By plugging the T-step Hamiltonian dynamics embedding (10) into the primal-dual MLE of the augmented model (8) and applying the density value evaluation (16), we obtain the proposed optimization, which learns primal potential f and the dual sampler adversarially,

$$\max_{f\in\mathcal{F}} \min_{\Theta} \ell\left(f,\Theta\right) := \widehat{\mathbb{E}}_{\mathcal{D}}\left[f\right] - \mathbb{E}_{(x^0,v^0)\sim q_\theta^0(x,v)}\left[f\left(x^T\right) - \frac{\lambda}{2}\left\|v^T\right\|_2^2\right] - H\left(q_\theta^0\right). \tag{20}$$

Here Θ denotes the learnable components in the dynamics embedding, *e.g.*, initialization q_θ^0, the stepsize (η) in the HMC/Langevin updates, and the adaptive part (S_v, S_x, g_v, g_x) in the generalized HMC. The parametrization of the initial distribution is discussed in Appendix C. Compared to the optimization in GANs (Goodfellow et al., 2014; Arjovsky et al., 2017; Dai et al., 2017), beside the reversal of $\min$-$\max$ in (20), the major difference is that our "generator" (the dual sampler) shares parameters with the "discriminator" (the primal potential function). In our formulation, the updates of the potential function automatically push the generator toward the target distribution, thus accelerating learning efficiency. Meanwhile, the tunable parameters in the dynamics embedding are learned adversarially, further promoting the efficiency of the dual sampler. These benefits will be empirically demonstrated in Section 5.

Similar optimization can be derived for generalized HMC (13) with density (17). For the T-step stochastic Langevin dynamics embedding (14), we apply the density value (19) to (9), which also leads to a $\max$-$\min$ optimization with multiple momenta.

We use stochastic gradient descent to estimate f for the exponential families as well as the parameters of the dynamics embedding Θ adversarially. Note that since the generated sample $\left(x_f^T, v_f^T\right)$ depends on f, the gradient w.r.t. f should also take these variables into account as back-propagation through time (BPTT), *i.e.*,

$$\nabla_f \ell\left(f;\Theta\right) = \widehat{\mathbb{E}}_{\mathcal{D}}\left[\nabla_f f\left(x\right)\right] - \mathbb{E}_{q^0}\left[\nabla_f f\left(x^T\right)\right]$$
$$-\mathbb{E}_{q^0}\left[\nabla_x f\left(x^T\right)\nabla_f x^T + \lambda v^T \nabla_f v^T\right]. \tag{21}$$

We illustrate the MLE via HMC adversarial dynamics embedding in Algorithm 1. The same technique can be applied to alternative dynamics embeddings parametrized dual sampler as in Appendix B. Considering the dynamics embedding as an *adaptive* sampler that automatically learns w.r.t. different models and datasets, the updates for Θ can be understood as *learning to sample*.

Algorithm 1 MLE via Adversarial Dynamics Embedding (ADE)

```
Initialize $\Theta_1$ randomly, set length of steps $T$.
for iteration $k = 1, \ldots, K$ do
    Sample mini-batch $\{x_i\}_{i=1}^m$ from dataset $\mathcal{D}$ and $\left\{x_i^0, v_i^0\right\}_{i=1}^m$ from $q_\theta^0\left(x,v\right)$.
    for iteration $t = 1, \ldots, T$ do
        Compute $\left(x^t, v^t\right) = \mathbf{L}\left(x^{t-1}, v^{t-1}\right)$ for each pair of $\left\{x_i^0, v_i^0\right\}_{i=1}^m$.
    end for
    [Learning the sampler] $\Theta_{k+1} = \Theta_k - \gamma_k \hat{\nabla}_\Theta \ell\left(f_k;\Theta_k\right)$
    [Estimating the exponential family] $f_{k+1} = f_k + \gamma_k \hat{\nabla}_f \ell\left(f_k;\Theta_k\right)$.
end for
```

4 Related Work

Connections to other estimators The primal-dual view of the MLE also allows us to establish connections between the proposed estimator, *adversarial dynamics embedding* (ADE), and existing approaches, including contrastive divergence (Hinton, 2002), pseudo-likelihood (PL) (Besag, 1975), conditional composite likelihood (CL) (Lindsay, 1988), score matching (SM) (Hyvärinen, 2005), minimum (diffusion) Stein kernel discrepancy estimator (DSKD) (Barp et al., 2019), non-local contrastive objectives (NLCO) (Vickrey et al., 2010), minimum probability flow (MPF) (Sohl-Dickstein et al., 2011), and noise-contrastive estimation (NCE) (Gutmann and Hyvärinen, 2010). As summarized

Table 1: (Fix) dual samplers used in alternative estimators. We denote $p_{\mathcal{D}}$ as the empirical data distribution, x_{-i} as x without i-th coordinate, p_n as the prefixed noise distribution, $\mathcal{T}_f\left(x'|x\right)$ as the HMC/Langevin transition kernel, $T_{\mathcal{D},f}\left(x\right)$ as the Stein variational gradient descent, and $A\left(x,x'\right)$ as the acceptance ratio.

Estimators	Dual Sampler $q(x)$
CD	$\int \prod_{i=1}^T \mathcal{T}_f\left(x^i|x^{i-1}\right) A(x^i, x^{i-1}) p_{\mathcal{D}}\left(x_0\right) dx_0^{T-1}$
SM	$\int \mathcal{T}_f\left(x'|x\right) p_{\mathcal{D}}\left(x\right) dx$
DSKD	$x' = T_{\mathcal{D},f}\left(x\right)$
PL	$q(x) = \frac{1}{d}\sum_{i=1}^d p_f(x_i|x_{-i}) p_{\mathcal{D}}(x_{-i})$
CL	$q(x) = \frac{1}{m}\sum_{i=1}^m p_f(x_{A_i}|x_{-A_i}) p_{\mathcal{D}}(x_{-A_i})$ $\{A_i\}_{i=1}^m = d$ and $A_i \cap A_j = \emptyset$
NLCO	$\sum_{i=1}^m \int p_{(f,i)}\left(x\right) p\left(S_i|x'\right) p_{\mathcal{D}}\left(x'\right) dx$ $p_{(f,i)}\left(x\right) = \frac{\exp(f(x))}{Z_i(f)}, x \in S_i$
MPF	$\int \mathcal{T}_f\left(x'|x\right) \exp\left(\frac{1}{2}\left(f\left(x'\right) - f\left(x\right)\right)\right) p_{\mathcal{D}}\left(x\right) dx$
NCE	$\left(\frac{1}{2}p_{\mathcal{D}} + \frac{1}{2}p_n\right) \frac{\exp(f(x))}{\exp(f(x)) + p_n(x)}$

in Table 1, these existing estimators can be recast as the special cases of ADE, by replacing the adaptive dual sampler with hand-designed samplers, which can lead to extra error and inferior solutions. Appendix D gives detailed derivations of the connections.

Exploiting deep models for energy-based model estimation has been investigated in Kim and Bengio (2016); Dai et al. (2017); Liu and Wang (2017); Dai et al. (2019). However, the parametrization of the dual sampler should both be flexible and tractable to achieve better performance. Existing work is limited in one aspect or another. Kim and Bengio (2016) parameterized the sampler via a deep directed graphical model, whose approximation ability is restrictive and the entropy is intractable. Dai et al. (2017) proposed algorithms relying either on a heuristic approximation or a lower bound of the entropy, and requiring learning an extra auxiliary component besides the dual sampler. Dai et al. (2019) applied the Fenchel dual representation twice to reformulate the entropy term, but the algorithm requires knowing a proposal distribution with the same support, which is impractical for high-dimensional data. By contrast, ADE achieves both sufficient flexibility and tractability by exploiting the augmented model and a novel parametrization within the primal-dual view.

Learning to sample ADE also shares some similarity with meta learning for sampling (Levy et al., 2018; Feng et al., 2017; Song et al., 2017; Gong et al., 2019), where the sampler is parametrized via a neural network and learned through certain objectives. The most significant difference lies in the ultimate goal: we focus on exponential family *model estimation*, where the learned sampler *assists* with this objective. By contrast, learning to sample techniques target on a sampler for a *fixed* model. This fundamentally distinguishes ADE from methods that only learn samplers. Moreover, ADE exploits an augmented model that yields tractable entropy estimation, which has not been fully investigated in previous literature.

5 Experiments

In this section, we test ADE on several synthetic datasets in Section 5.1 and real-world image datasets in Section 5.2. The details of each experiment setting can be found in Appendix F.

5.1 Synthetic experiments

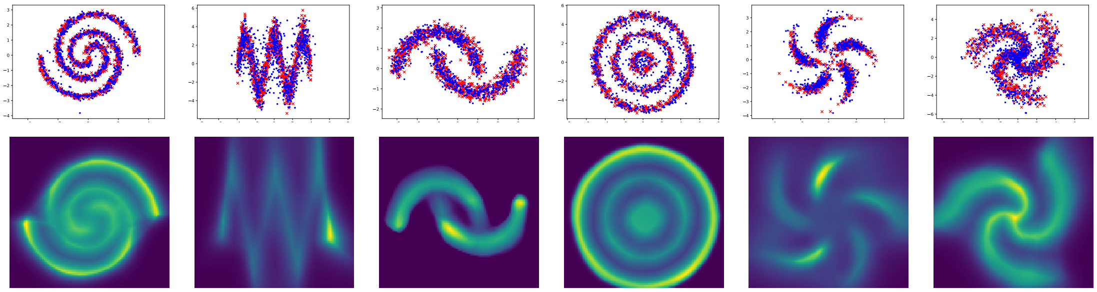
We compare ADE with SM, CD, and primal-dual MLE with the normalizing planar flow (Rezende and Mohamed, 2015) sampler (NF) to investigate the claimed benefits. SM, CD and primal-dual with NF can be viewed as special cases of our method, with either a fixed sampler or restricted parametrized q_θ. Thus, this also serves as an ablation study of ADE to verify the significance of its different subcomponents. We keep the model sizes the same in NF and ADE (10 planar layers). Then we perform 5-steps stochastic Langevin steps to obtain the final samples x^T with standard Gaussian noise in each step, and without incurring extra memory cost. For fairness, we conduct CD with 15 steps. This setup is preferable to CD with an extra acceptance-rejection step. We emphasize that, by comparison to SM and CD, ADE learns the sampler and exploits the gradients through the sampler. In comparison to primal-dual with NF, dynamics embedding achieves more flexibility without introducing extra parameters. Complete experiment details are given in Appendix F.1.

Table 2: Comparison on synthetic data using maximum mean discrepancy (MMD $\times 1e^{-3}$).

Dataset	SM	NF	CD-15	ADE
2spirals	5.09	0.69	-0.45	**-0.61**
Banana	8.10	0.88	-0.31	**-0.99**
circles	4.90	0.76	-0.83	**-1.13**
cos	10.36	0.91	7.15	**-0.55**
Cosine	8.34	2.15	0.78	**-1.09**
Funnel	13.07	**-0.92**	-0.38	-0.75
swissroll	19.93	1.97	0.20	**-0.36**
line	10.28	0.39	10.5	**-1.30**
moons	41.34	0.80	2.21	**-1.10**
Multiring	2.01	0.30	-0.38	**-1.02**
pinwheel	18.41	3.01	**-1.03**	-0.95
Ring	9.22	161.89	0.12	**-0.91**
Spiral	9.48	5.96	-0.41	**-0.81**
Uniform	5.88	0.00	**-1.17**	-0.94

In Figure 1, we visualize the learned distribution using both the learned dual sampler and the unnormalized exponential model on several synthetic datasets. Overall, the sampler almost perfectly recovers the distribution, and the learned f captures the landscape of the distribution. We also plot the convergence behavior in Figure 2. We observe that the samples are smoothly converging to the true data distribution. As the learned sampler depends on f, this figure also indirectly suggests good convergence behavior for f. More results for the learned models can be found in Figure 5 in Appendix G.

A quantitative comparison in terms of the MMD (Gretton et al., 2012) of the samplers is in Table 2. To compute the MMD, for NF and ADE, we use 1,000 samples from their sam-



(a) `2spirals` (b) `Cosine` (c) `moons` (d) `Multiring` (e) `pinwheel` (f) `Spiral`

Figure 1: We illustrated the learned samplers from different synthetic datasets in the first row. The × denotes training data and ● denotes the ADE samplers. The learned potential functions f are illustrated in the second row.

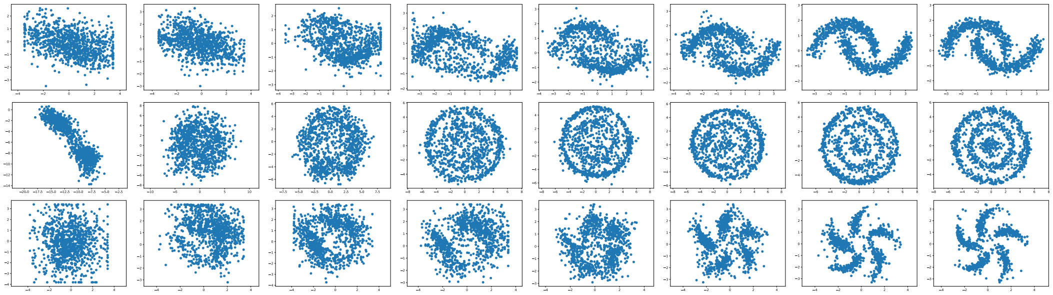


Figure 2: Convergence behavior of sampler on `moons`, `Multiring`, `pinwheel` synthetic datasets.

pler with Gaussian kernel. The kernel bandwidth is chosen using median trick (Dai et al., 2016). For SM, since there is no such sampler available, we use vanilla HMC to get samples from the learned model f, and use them to estimate MMD as in Dai et al. (2019). As we can see from Table 2, ADE obtains the best MMD in most cases, which demonstrates the flexibility of dynamics embedding compared to normalizing flow, and the effectiveness of adversarial training compared to SM and CD.

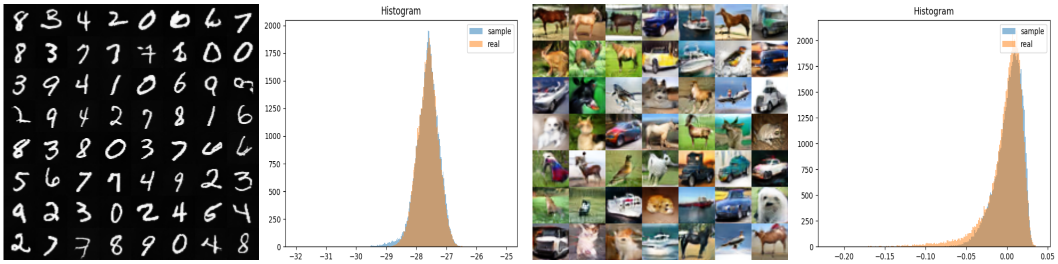
We also investigate the parameters recovery of ADE on the multivariate Gaussians with different dimensions where we know the potential functions. The empirical results can be found in Table 5 in Appendix G. In this simple task, the SM is proven to be consistent and achieve the same estimator as MLE (Hyvärinen, 2005). The objective of ADE can be non-convex due to the learning of the sampler parametrization, therefore, it losses the theoretical guarantees and incurs extra cost. However, as we can see the ADE still achieves comparable performances.

5.2 Real-world Image Datasets

We apply ADE to `MNIST` and `CIFAR-10` data. In both cases, we use a CNN architecture for the discriminator, following Miyato et al. (2018), with spectral normalization added to the discriminator layers. In particular, for the discriminator in the `CIFAR-10` experiments, we replace all downsampling operations by average pooling, as in Du and Mordatch (2019). We parametrize the initial distribution $p_0(x, v)$ with a deep Gaussian latent variable model (Deep LVM), specified in Appendix C. The output sample is clipped to $[0, 1]$ after each HMC step and the Deep LVM initialization. The detailed architectures and experimental configurations are described in Appendix F.2.

Table 3: Inception scores of different models on `CIFAR-10` (unconditional).

Model	Inception Score
WGAN-GP (Gulrajani et al., 2017)	6.50
Spectral GAN (Miyato et al., 2018)	7.42
Langevin PCD (Du and Mordatch, 2019)	6.02
Langevin PCD (10 ensemble) (Du and Mordatch, 2019)	6.78
ADE: Deep LVM init w/o HMC	7.26
ADE: Deep LVM init w/ HMC	**7.55**



(a) Samples on MNIST (b) Histogram on MNIST (c) Samples on CIFAR-10 (d) Histogram on CIFAR-10

Figure 3: The generated images on MNIST and CIFAR-10 and the comparison between energies of generated samples and real images. The blue histogram illustrates the distribution of $f(x)$ on generated samples, and the orange histogram is generated by $f(x)$ on testing samples. As we can see, the learned potential function $f(x)$ matches the empirical dataset well.

We report the inception scores in Table 3. For ADE, we train with Deep LVM as the initial q_θ^0 with/without HMC steps for an ablation study. The HMC embedding greatly improves the performance of the samples generated by the initial q_θ^0 alone. The proposed ADE not only achieves better performance, compared to the fixed Langevin PCD for energy-based models reported in (Du and Mordatch, 2019), but also enables the generator to outperform the Spectral GAN.

We show some of the generated images in Figure 3(a) and (c); additional sampled images can be found in Figure 6 and 7 in Appendix G. We also plot the potential distribution (unnormalized) of the generated samples and that of the real images for MNIST and CIFAR-10 (using 1000 data points for each) in Figure 3(b) and (d). The energy distributions of both the generated and real images show significant overlap, demonstrating that the obtained energy functions have successfully learned the desired distributions.

Since ADE learns an energy-based model, the learned model and sampler can also be used for image completion. To further illustrate the versatility of ADE, we provide several image completions on MNIST in Figure 4. Specifically, we estimate the model with ADE on fully observed images. For the input images, we mask the lower half with uniform noise. To complete the corrupted images, we perform the learned dual sampler steps to update the lower half of images with the upper half images fixed. We visualize the output from each of the 20 HMC runs in Figure 4. Further details are given in Appendix F.2.

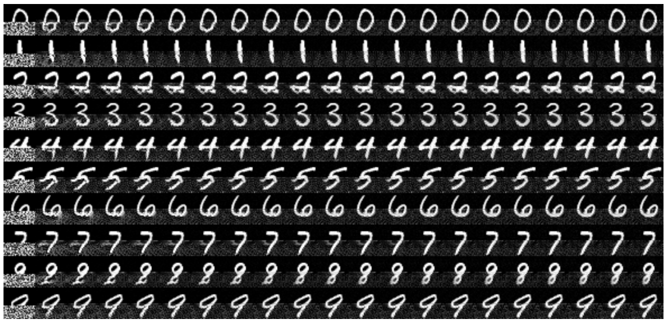


Figure 4: Image completion with the ADE learned model and sampler on MNIST.

6 Conclusion

We proposed Adversarial Dynamics Embedding (ADE) to efficiently perform MLE with general exponential families. In particular, by utilizing the primal-dual formulation of the MLE for an augmented distribution with auxiliary kinetic variables, we incorporate the parametrization of the dual sampler into the estimation process in a fully differentiable way. This approach allows for shared parameters between the primal and dual, achieving better estimation quality and inference efficiency. We also established the connection between ADE and existing estimators. Our empirical results on both synthetic and real data illustrate the advantages of the proposed approach.

Acknowledgments

We thank David Duvenaud, Arnaud Doucet, George Tucker and the Google Brain team, as well as the anonymous reviewers for their insightful comments and suggestions. L.S. was supported in part by NSF grants CDS&E-1900017 D3SC, CCF-1836936 FMitF, IIS-1841351, CAREER IIS-1350983.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein GAN. In *International Conference on Machine Learning*, 2017.

Alessandro Barp, Francois-Xavier Briol, Andrew B. Duncan, Mark Girolami, and Lester Mackey. Minimum Stein Discrepancy Estimators. *arXiv preprint arXiv:1906.08283*, 2019.

Dimitri Bertsekas. *Nonlinear Programming*. Athena Scientific, Belmont, MA, 1995.

Julian Besag. Statistical analysis of non-lattice data. *The Statistician*, 24:179–195, 1975.

Christos Boutsidis, Petros Drineas, Prabhanjan Kambadur, Eugenia-Maria Kontopoulou, and Anastasios Zouzias. A randomized algorithm for approximating the log determinant of a symmetric positive definite matrix. *Linear Algebra and its Applications*, 533:95–117, 2017.

Lawrence D. Brown. *Fundamentals of Statistical Exponential Families*, volume 9 of *Lecture notes-monograph series*. Institute of Mathematical Statistics, Hayward, Calif, 1986.

Anthony L Caterini, Arnaud Doucet, and Dino Sejdinovic. Hamiltonian variational auto-encoder. In *Advances in Neural Information Processing Systems*, 2018.

Bo Dai, Niao He, Hanjun Dai, and Le Song. Provable bayesian inference via particle mirror descent. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 985–994, 2016.

Bo Dai, Hanjun Dai, Niao He, Weiyang Liu, Zhen Liu, Jianshu Chen, Lin Xiao, and Le Song. Coupled variational bayes via optimization embedding. In *Advances in Neural Information Processing Systems*, 2018.

Bo Dai, Hanjun Dai, Arthur Gretton, Le Song, Dale Schuurmans, and Niao He. Kernel exponential family estimation via doubly dual embedding. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, pages 2321-2330, 2019.

Zihang Dai, Amjad Almahairi, Philip Bachman, Eduard Hovy, and Aaron Courville. Calibrating energy-based generative adversarial networks. In *International Conference on Learning Representations* , 2017.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real NVP. In *International Conference on Learning Representations* , 2017.

Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. *arXiv preprint arXiv:1903.08689*, 2019.

Yihao Feng, Dilin Wang, and Qiang Liu. Learning to draw samples with amortized stein variational gradient descent. In *Conference on Uncertainty in Artificial Intelligence*, 2017.

Wenbo Gong, Yingzhen Li, and José Miguel Hernández-Lobato. Meta-learning for stochastic gradient MCMC. In *International Conference on Learning Representations* , 2019.

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*, pages 2672–2680, 2014.

Will Grathwohl, Ricky TQ Chen, Jesse Betterncourt, Ilya Sutskever, and David Duvenaud. FFJORD: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations* , 2019.

Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *JMLR*, 13:723–773, 2012.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017.

Michael Gutmann and Aapo Hyvärinen. 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, 2010.

Insu Han, Dmitry Malioutov, and Jinwoo Shin. Large-scale log-determinant computation through stochastic chebyshev expansions. In *International Conference on Machine Learning*, pages 908–917, 2015.

Geoffrey E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.

Aapo Hyvärinen. Estimation of non-normalized statistical models using score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.

Aapo Hyvärinen. Connections between score matching, contrastive divergence, and pseudolikelihood for continuous-valued variables. *IEEE Transactions on neural networks*, 18(5):1529-1531, 2007.

Taesup Kim and Yoshua Bengio. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*, 2016.

Ross Kindermann and J. Laurie Snell *Markov Random Fields and their applications*. Amer. Math. Soc., Providence, RI, 1980.

Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1×1 convolutions. In *Advances in Neural Information Processing Systems*, 2018.

Diederik P Kingma and Yann LeCun. Regularized estimation of image statistics by score matching. In *NIPS*, 2010.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. In *International Conference on Learning Representations*, 2014.

Diederik P Kingma, Tim Salimans, Rafal Jozefowicz, Xi Chen, Ilya Sutskever, and Max Welling. Improved variational inference with inverse autoregressive flow. In *Advances in Neural Information Processing Systems*, pages 4743–4751, 2016.

John Lafferty, Andrew McCallum, and Fernando C.N. Pereira. Conditional random fields: Probabilistic modeling for segmenting and labeling sequence data. In *Proceedings of International Conference on Machine Learning*, volume 18, pages 282–289, San Francisco, CA, 2001. Morgan Kaufmann.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006.

Daniel Levy, Matthew D Hoffman, and Jascha Sohl-Dickstein. Generalizing hamiltonian monte carlo with neural networks. In *International Conference on Learning Representations* , 2018.

Bruce G. Lindsay. Composite likelihood methods. *Contemporary Mathematics*, 80(1):221–239, 1988.

Qiang Liu and Dilin Wang. Learning deep energy models: Contrastive divergence vs. amortized MLE. *arXiv preprint arXiv:1707.00797*, 2017.

Yi-An Ma, Tianqi Chen, and Emily Fox A complete recipe for stochastic gradient MCMC. In In *Advances in Neural Information Processing Systems*, pages 2917–2925, 2015.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations* , 2018.

Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Conference on International Conference on Machine Learning*, 2012.

Radford M Neal. MCMC using Hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2 (11), 2011.

Danilo J Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1278–1286, 2014.

Danilo Jimenez Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, 2015.

R. Tyrrell Rockafellar ockafellar *Convex Analysis*, volume 28 of *Princeton Mathematics Series*. Princeton University Press, Princeton, NJ, 1970.

Jascha Sohl-Dickstein, Peter Battaglino, and Michael R DeWeese. Minimum probability flow learning. In *Proceedings of the 28th International Conference on Machine Learning*, pages 905–912, 2011.

Jiaming Song, Shengjia Zhao, and Stefano Ermon. A-nice-MC: Adversarial training for mcmc. In *Advances in Neural Information Processing Systems*, pages 5140–5150, 2017.

Bharath Sriperumbudur, Kenji Fukumizu, Arthur Gretton, Aapo Hyvärinen, and Revant Kumar. Density estimation in infinite dimensional exponential families. *The Journal of Machine Learning Research*, 18(1):1830–1888, 2017.

Tijmen Tieleman. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the International Conference on Machine Learning*, 2008.

Tijmen Tieleman and Geoffrey Hinton. Using fast weights to improve persistent contrastive divergence. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1033–1040. ACM, 2009.

David Vickrey, Cliff Chiung-Yu Lin, and Daphne Koller. Non-local contrastive objectives. In *Proceedings of the International Conference on Machine Learning*, 2010.

Martin J. Wainwright and Michael I. Jordan Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, $1(1-2)$:1–305, 2008.

Li Wenliang, Dougal Sutherland, Heiko Strathmann, and Arthur Gretton. Learning deep kernels for exponential family densities. In *International Conference on Machine Learning*, 2019.

Ying Nian Wu, Jianwen Xie, Yang Lu, and Song-Chun Zhu. Sparse and deep generalizations of the frame model. *Annals of Mathematical Sciences and Applications*, 3(1):211–254, 2018.

Linfeng Zhang, Weinan E, and Lei Wang. Monge-Ampère flow for generative modeling. *arXiv preprint arXiv:1809.10188*, 2018.

Song Chun Zhu and David Mumford. Grade: Gibbs reaction and diffusion equations. In *Sixth International Conference on Computer Vision (IEEE Cat. No. 98CH36271)*, pages 847–854. IEEE, 1998.

Appendix

A Proof of Theorems in Section 3

Theorem 2 (Equivalent MLE) *The MLE of the augmented model is the same as the original MLE.*

Proof The conclusion is straightforward from independence between x and v. We rewrite the MLE (7) in another way as follows

$$\max_f L\left(f\right) = \widehat{\mathbb{E}}_{x\sim\mathcal{D}}\left[\log\int p\left(x,v\right)dv\right] \tag{22}$$

$$= \widehat{\mathbb{E}}_{x\sim\mathcal{D}}\left[\log\left(p\left(x\right)\int p\left(v\right)dv\right)\right] \tag{23}$$

$$= \widehat{\mathbb{E}}_{x\sim\mathcal{D}}\left[\log p\left(x\right)+\underbrace{\log\int p\left(v\right)dv}_{\log 1=0}\right] = \widehat{\mathbb{E}}_{x\sim\mathcal{D}}\left[\log p\left(x\right)\right], \tag{24}$$

where the second equation comes from the definition of the $p\left(x,v\right)$ in (6) with independent x and v. ■

Theorem 3 (HMC embeddings as gradient flow) *In continuous time,* i.e. *with infinitesimal stepsize* $\eta\to 0$*, the density of particles* $\left(x^t,v^t\right)$*, denoted* $q^t\left(x,v\right)$*, follows the Fokker-Planck equation*

$$\tfrac{\partial q^t\left(x,v\right)}{\partial t} = \nabla\cdot\left(q^t\left(x,v\right)G\nabla\mathcal{H}\left(x,v\right)\right), \tag{25}$$

with $G=\begin{bmatrix}0 & \mathbf{I}\\ -\mathbf{I} & 0\end{bmatrix}$*, which has a stationary distribution* $p\left(x,v\right)\propto\exp\left(-\mathcal{H}\left(x,v\right)\right)$ *with the marginal distribution* $p(x)\propto\exp\left(f\left(x\right)\right)$.

Proof The first part of the theorem is trivial. When $\eta\to 0$, the HMC follows the dynamical system

$$\left[\frac{dx}{dt},\frac{dv}{dt}\right] = \left[\partial_v\mathcal{H}\left(x,v\right),-\partial_x\mathcal{H}\left(x,v\right)\right] = G\nabla\mathcal{H}\left(x,v\right).$$

By applying the Fokker-Planck equation, we obtain

$$\frac{\partial q^t\left(x,v\right)}{\partial t} = \nabla\cdot\Big(q^t\left(x,v\right)G\nabla\mathcal{H}\left(x,v\right)\Big). \tag{26}$$

To show that the stationary distribution of such dynamical system converges to $p\left(x,v\right)\propto\exp\left(-\mathcal{H}\left(x,v\right)\right)$, recall the fact that

$$\nabla\cdot\Big(G\nabla q^t\left(x,v\right)\Big) = -\partial_x\partial_v q^t\left(x,v\right)+\partial_v\partial_x q^t\left(x,v\right)=0. \tag{27}$$

The Fokker-Planck equation can be rewritten as

$$\frac{\partial q^t\left(x,v\right)}{\partial t} = \nabla\cdot\Big(q^t\left(x,v\right)G\nabla\mathcal{H}\left(x,v\right)+G\nabla q^t\left(x,v\right)\Big). \tag{28}$$

Substitute $p\left(x,v\right)\propto\exp\left(-\mathcal{H}\left(x,v\right)\right)$ into (28) and notice

$$\exp\left(-\mathcal{H}\left(x,v\right)\right)\nabla\mathcal{H}\left(x,v\right)+\nabla\exp\left(-\mathcal{H}\left(x,v\right)\right)=0,$$

we have $\partial p\left(x,v\right)=0$, which means $p\left(x,v\right)\propto\exp\left(-\mathcal{H}\left(x,v\right)\right)$ is a stationary distribution, and thus $p\left(x\right)\propto\exp\left(f\left(x\right)\right)$. ■

Theorem 4 (Density value evaluation) *If* $\left(x^0,v^0\right)\sim q^0_\theta\left(x,v\right)$*, after* T *vanilla HMC steps* (10)*, we have*

$$q^T\left(x^T,v^T\right)=q^0_\theta\left(x^0,v^0\right).$$

For the $\left(x^T,v^T\right)$ *from the generalized leapfrog steps* (13)*, we have*

$$q^T\left(x^T,v^T\right)=q^0_\theta\left(x^0,v^0\right)\prod_{t=1}^T\left(\Delta_x\left(x^t\right)\Delta_v\left(v^t\right)\right),$$

where $\Delta_x\left(x^t\right)$ and $\Delta_v\left(v^t\right)$ are defined in (29).

For the $\left(x^T, \left\{v^i\right\}_{i=1}^T\right)$ from the stochastic Langevin dynamics (14) *with $\left(x^0, \left\{\xi^i\right\}_{i=0}^{T-1}\right) \sim q_\theta^0\left(x, \xi\right) \prod_{i=1}^{T-1} q_{\theta_i}\left(\xi^i\right)$, we have*

$$q^T\left(x^T, \left\{v^i\right\}_{i=1}^T\right) = q_\theta^0\left(x^0, \xi^0\right) \prod_{i=1}^{T-1} q_{\theta_i}\left(\xi^i\right).$$

Proof The claim can be obtained by simply applying the change-of-variable rule, *i.e.*,

$$q^T\left(x^T, v^T\right) = q_\theta^0\left(x^0, v^0\right) \prod_{t=1}^{T} \left|\det \nabla \mathbf{L}_{f,M}\left(x^t, v^t\right)\right|.$$

The Jacobian of the transformation from (x, v) to $\left(x, v^{-\frac{1}{2}}\right)$ is $\begin{bmatrix}\mathbf{I} & \mathbf{0}\\ \frac{\eta}{2}\nabla_x^2 f\left(x\right) & \mathbf{I}\end{bmatrix}$, whose determinant is 1. Similarly, the determinant of the Jacobian of the transform from $\left(x, v^{-\frac{1}{2}}\right)$ to (x', v') is also 1. Therefore, $\left|\det\left(\nabla \mathbf{L}_{f,M}\left(x^t, v^t\right)\right)\right| = 1, \forall i = 1, \ldots, T$, and we prove the first claim.

The second claim can also be obtained in a similar way. By simple algebraic manipulations, we have that the Jacobians of the transformation are all diagonal matrices. Thus,

$$\begin{aligned}\Delta_x\left(x^t\right) &= \left|\det\left(\operatorname{diag}\left(\exp\left(2S_v\left(\nabla_x f\left(x^t\right), x^t\right)\right)\right)\right)\right|,\\ \Delta_v\left(v^t\right) &= \left|\det\left(\operatorname{diag}\left(\exp\left(S_x\left(v^{\frac{1}{2}}\right)\right)\right)\right)\right|.\end{aligned} \tag{29}$$

Similarly, we calculate the Jacobian for the stochastic Langevin update. Specifically, during the t-th step, the Jacobian of the transformation from $\left(x^{t-1}, \left\{v^i\right\}_{i=1}^{t-1}, \xi^{t-1}\right)$ to $\left(x^{t-1}, \left\{v^i\right\}_{i=1}^{t-1}, v^t\right)$ is $\begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{I} & \mathbf{0}\\ \frac{\eta}{2}\nabla_x^2 f\left(x\right) & \mathbf{0} & \mathbf{I}\end{bmatrix}$, whose determinant is 1. Similarly, the Jacobian of the transformation from $\left(x^{t-1}, \left\{v^i\right\}_{i=1}^{t-1}, v^t\right)$ to $\left(x^t, \left\{v^i\right\}_{i=1}^{t-1}, v^t\right)$ is $\begin{bmatrix}\mathbf{I} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{I} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \mathbf{I}\end{bmatrix}$, whose determinant is also 1. Therefore $\left|\det\left(\nabla \mathbf{L}_f\left(x^t, \left\{v^i\right\}_{i=1}^t\right)\right)\right| = 1$, which implies

$$q^t\left(x^t, \left\{v^i\right\}_{i=1}^{t-1}, v^t\right) = q^{t-1}\left(x^{t-1}, \left\{v^i\right\}_{i=1}^{t-1}, \xi^{t-1}\right) = q^{t-1}\left(x^{t-1}, \left\{v^i\right\}_{i=1}^{t-1}\right) q_{\theta^{t-1}}\left(\xi^{t-1}\right).$$

Apply the same argument for $\forall t = 1, \ldots, T$, we obtain the third claim. ■

B Variants of Dynamics Embedding

Besides the vanilla Hamiltonian/Langevin embedding and its generalized version we introduced in the main text, we can also embed alternative dynamics, *i.e.*, deterministic Langevin dynamics and its continuous and generalized version.

B.1 Deterministic Langevin Embedding

We embed the *deterministic Langevin dynamics* to form $x' = \mathbf{L}_{f,M}\left(x\right)$ as $x' = x + \eta \nabla_x f\left(x\right)$ with $x^0 \sim q_\theta^0\left(x\right)$. By the change-of-variable rule, we have $q_{f,M}^T\left(x^T\right) = q_\theta^0\left(x_0\right) \prod_{t=1}^T \left|\det \frac{\partial x^t}{\partial x^{t-1}}\right|$. The deterministic Langevin embedding has been exploited in variational auto-encoder (Dai et al., 2018), in which the variational technique has been applied to bypass the calculation of $\prod_{t=1}^T \left|\det \frac{\partial x^t}{\partial x^{t-1}}\right|$.

Plug such parametrization of the dual distribution into (5), we achieve the alternative objective

$$\max_{f\in\mathcal{F}}\min_{\theta,M,\eta} \ell\left(f;\theta,M,\eta\right) := \widehat{\mathbb{E}}_{\mathcal{D}}\left[f\right] - \mathbb{E}_{x^0\sim q_\theta^0(x)}\left[f\left(x^T\right) - \log q_\theta^0\left(x\right) - \sum_{t=1}^{T}\log\left|\det\frac{\partial x^t}{\partial x^{t-1}}\right|\right]. \tag{30}$$

For the log-determinant term, $\log\left|\det\frac{\partial x^t}{\partial x^{t-1}}\right| = \log\left|\det\left(I+\eta\mathbf{H}^f\left(x^t\right)\right)\right|$, where $\mathbf{H}^f_{i,j} = \frac{\partial^2 f(x)}{\partial x_i\partial x_j}$. Then, the gradient $\frac{\partial\log\left|\det\left(I+\eta\mathbf{H}^f\left(x^t\right)\right)\right|}{\partial f} = \eta\,\mathrm{tr}\left(\left(I+\eta\mathbf{H}^f\left(x_t\right)\right)^{-1}\frac{\partial H^f\left(x^t\right)}{\partial f}\right)$. However, the computation of the log-determinant and its derivative w.r.t. f are expensive. We can apply the polynomial expansion to approximate it.

Denoting δ as the bound of the spectrum of $\mathbf{H}^f\left(x^t\right)$ and $C := \frac{\eta\delta}{1+\eta\delta}I - \frac{1}{1+\eta\delta}\mathbf{H}^f\left(x^t\right)$, we have $\lambda\left(C\right)\in\left(-1,1\right)$. Then,

$$\log\left|\det\left(I+\eta\mathbf{H}^f\left(x^t\right)\right)\right| = d\log\left(1+\eta\delta\right) + \mathrm{tr}\left(\log\left(I-C\right)\right).$$

We can apply Taylor expansion or Chebyshev expansion to approximate the $\mathrm{tr}\left(\log\left(I-C\right)\right)$. Specifically, we have

- Stochastic Taylor Expansion (Boutsidis et al., 2017) Recall $\log\left(1-x\right) = -\sum_{k=1}^{\infty}\frac{x^k}{k}$, we have the Taylor expansion

  $$\mathrm{tr}\left(\log\left(I-C\right)\right) = -\sum_{i=1}^{k}\frac{\mathrm{tr}\left(C^i\right)}{i}.$$

  To avoid the matrix-matrix multiplication, we further approximate the $\mathrm{tr}\left(C\right) = \mathbb{E}_z\left[z^\top Cz\right]$ with z as Rademacher random variables, *i.e.*, Bernoulli distribution with $p=\frac{1}{2}$.

  Particularly, if we set $i=1$, recall the $\mathrm{tr}\left(\mathbf{H}^f\left(x\right)\right) = \nabla_x^2 f\left(x\right)$, we can directly calculate without the Hutchinson approximation.

- Stochastic Chebyshev Expansion (Han et al., 2015) We can approximate with Chebyshev polynomial, *i.e.*,

  $$\mathrm{tr}\left(\log\left(I-C\right)\right) = \sum_{i=1}^{k}c_i\,\mathrm{tr}\left(R_i\left(C\right)\right),$$

  where $R\left(\cdot\right)$ denotes the Chebshev polynomial as $R_i\left(x\right) = 2xR_{i-1}\left(x\right) - R_{i-2}\left(x\right)$ with $R_1\left(x\right) = x$ and $R_0\left(x\right) = 1$. The $c_i = \frac{2}{k+1}\sum_{j=0}^{k}\log\left(1-s_j\right)R_i\left(s_j\right)$ if $i\geqslant 1$, otherwise $c_0 = \frac{1}{n+1}\sum_{j=0}^{k}\log\left(1-s_j\right)$ where $s_j = \cos\left(\frac{\pi\left(k+\frac{1}{2}\right)}{k+1}\right)$ for $j=0,1,\ldots,k$.

  Similarly, we can use the Hutchinson approximation to avoid matrix-matrix multiplication.

B.2 Continuous-time Langevin Embedding

We discuss several discretized dynamics embedding above. In this section, we take the continuous-time limit $\eta\to 0$ in the deterministic Langevin dynamics, *i.e.*, $\frac{dx}{dt} = \nabla_x f\left(x\right)$. Follow the change-of-variable rule, we obtain

$$\begin{aligned} & q\left(x'\right) = p\left(x\right)\det\left(I+\eta\mathbf{H}^f\left(x\right)\right) \\ \Rightarrow\quad & \log q\left(x'\right) - \log p\left(x\right) = -\,\mathrm{tr}\log\left(I+\eta\mathbf{H}^f\left(x\right)\right) = -\eta\nabla_x^2 f\left(x\right) + \mathcal{O}\left(\eta^2\right). \end{aligned}$$

As $\eta\to 0$, we have

$$\frac{d\log q\left(x,t\right)}{dt} = -\nabla_x^2 f\left(x\right). \tag{31}$$

Remark (connections to Fokker-Planck equation) Consider the $\frac{dx}{dt} = \nabla_x f\left(x\right)$ as a SDE with zero diffusion term, by Fokker-Planck equation, we obtain the PDE w.r.t. $q\left(x,t\right)$ as

$$\frac{\partial q\left(x,t\right)}{\partial t} = -\nabla\cdot\left(\nabla_x f\left(x\right)q\left(x,t\right)\right).$$

Alternatively, we can also derive the (31) from the Fokker-Planck equation by explicitly writing the derivative. Specifically,

$$\begin{aligned}
\frac{dq\left(x,t\right)}{dt} &= \frac{\partial q\left(x,t\right)}{\partial x}\frac{\partial x}{\partial t} + \frac{\partial q\left(x,t\right)}{\partial t} \\
&= \frac{\partial q\left(x,t\right)}{\partial x}\nabla_x f\left(x\right) - \nabla\cdot\left(\nabla_x f\left(x\right) q\left(x,t\right)\right) \\
&= \frac{\partial q\left(x,t\right)}{\partial x}\nabla_x f\left(x\right) - \nabla_x^2 f\left(x\right) q\left(x,t\right) - \nabla_x f\left(x\right)\frac{\partial q\left(x,t\right)}{\partial t} \\
&= -\nabla_x^2 f\left(x\right) q\left(x,t\right).
\end{aligned}$$

Therefore, we have

$$\frac{1}{q\left(x,t\right)}\frac{dq\left(x,t\right)}{dt} = -\nabla_x^2 f\left(x\right) \Rightarrow \begin{bmatrix} \frac{d\log q(x,t)}{dt} = -\nabla_x^2 f\left(x\right) \\ \frac{dx}{dt} = \nabla_x f\left(x\right) \end{bmatrix}. \tag{32}$$

Based on (31), we can obtain the samples and its density value by

$$\begin{bmatrix} x^t \\ \log q\left(x^t\right) - \log p_\theta^0\left(x^0\right) \end{bmatrix} = \int_{t_0}^{t_1} \begin{bmatrix} \nabla_x f\left(x\left(t\right)\right) \\ -\nabla_x^2 f\left(x(t)\right) \end{bmatrix} dt := \mathbf{L}_{f,t_0,t_1}\left(x\right). \tag{33}$$

We emphasize that this dynamics is different from the continuous-time flow proposed in Grathwohl et al. (2019), where we have $\nabla_x^2 f\left(x\right)$ in the ODE rather than a trace operator, which requires one more Hutchinson stochastic approximation. We noticed that Zhang et al. (2018) also exploits the Monge-Ampère equation to design the flow-based model for unsupervised learning. However, their learning algorithm is totally different from ours. They use the parameterization as a new flow and fit the model by matching a *separate* distribution; while in our case, the exponential family and flow share the same parameters and match each other automatically.

We can approximate the integral using a numerical quadrature methods. One can approximate the $\nabla_{(f,t_0,t_1)}\ell\left(f;t_0,t_1\right)$ by the derivative through the numerical quadrature. Alternatively, we denote $g\left(t\right) = -\frac{\partial\ell(f,t_0,t_1)}{\partial x(t)}$, by the adjoint method, the $\frac{\ell(f,t_0,t_1)}{\partial f}$ is also characterized by ODE

$$\frac{\partial\ell\left(f,t_0,t_1\right)}{\partial f} = \int_{t_0}^{t_1} -g\left(t\right)^\top \nabla_f \cdot \nabla_x f\left(x\right) dt, \tag{34}$$

and can be approximated by numerical quadrature too.

We can combine the discretized and continuous-time Langevin dynamics by simply stacking several layers of $\mathbf{L}_{f,t_0,t_1}$.

B.3 Generalized Continuous-time Langevin Embedding

We generalize the continuous-time Langevin dynamics by introducing more learnable space as

$$\frac{dx}{dt} = h\left(\xi_f\left(x\right)\right), \tag{35}$$

where h can be arbitrary smooth function and $\xi_f\left(x\right) = \left(\nabla_x f\left(x\right), f\left(x\right), x\right)$. We now derive the distributions formed by such flows following the change-of-variable rule, *i.e.*,

$$\begin{aligned}
& q\left(x'\right) = p\left(x\right)\det\left(I + \eta\nabla_x h\left(\xi_f\left(x\right)\right)\right) \\
\Rightarrow\quad & \log q\left(x'\right) - \log p\left(x\right) = -\operatorname{tr}\log\left(I + \eta\nabla_x h\left(\xi_f\left(x\right)\right)\right) = -\eta\operatorname{tr}\left(\nabla_x h\left(\xi_f\left(x\right)\right)\right) + \mathcal{O}\left(\eta^2\right).
\end{aligned}$$

As $\eta \to 0$, we have

$$\frac{d\log q\left(x,t\right)}{dt} = -\operatorname{tr}\left(\nabla_x h\left(\xi_f\left(x\right)\right)\right). \tag{36}$$

Similarly, we can compute the samples and its density value by

$$\begin{bmatrix} x^t \\ \log q\left(x^t\right) - \log p_\theta^0\left(x^0\right) \end{bmatrix} = \int_{t_0}^{t_1} \begin{bmatrix} h\left(\xi_f\left(x\right)\right) \\ -\operatorname{tr}\left(\nabla_x h\left(\xi_f\left(x\right)\right)\right) \end{bmatrix} dt := \mathbf{L}_{f,t_0,t_1}\left(x\right). \tag{37}$$

C Practical Algorithm

In this section, we discuss several key components in the implementation of the Algorithm 1, including the gradient computation and the parametrization of the initialization $q_\theta\left(x, v\right)$.

C.1 Gradient Estimator

The gradient w.r.t. f is illustrated in (21). The computation of the gradient needs to compute back-propagated through time, therefore, the computational cost is proportional to the number of sampling steps T.

By Denskin's theorem (Bertsekas, 1995), if the samples (x, v) from the optimal solution $p\left(x, v\right) \propto \exp\left(-\mathcal{H}\left(x, v\right)\right)$, the third term in (21) exactly vanish to zero, *i.e.*,

$$\nabla_f \ell\left(f; \Theta\right) = \widehat{\mathbb{E}}_{\mathcal{D}}\left[\nabla_f f\left(x\right)\right] - \mathbb{E}_{(x,v)\sim p(x,v)}\left[\nabla_f f\left(x\right)\right], \tag{38}$$

whose computational cost is independent to T.

Recall Theorem 3 that as $\eta \to 0$ and $T \to \infty$, the HMC embedding converges to the optimal solution. Therefore, we can approximate the BPTT estimator (21) with the truncated gradient (38). As T increasing, the corresponding dual sampler approaches the optimal solution, and the truncation bias becomes smaller.

C.2 Initialization Distribution Parametrization

In our algorithm, the dual distribution are parametrized via dynamics sampling method with an initial distribution $q^0_\theta\left(x, v\right)$, whose density value is available. There are several possible parametrization:

- **Flow-based model:** The most straightforward parametrization for $q^0_\theta\left(x, v\right)$ is utilizing flow-based model (Rezende and Mohamed, 2015; Dinh et al., 2017; Kingma and Dhariwal, 2018). For simplicity, we can decompose $q^0_\theta\left(x, v\right) = q^0_{\theta_1}\left(x\right) q^0_{\theta_2}\left(v\right)$ and parametrized both $q^0_{\theta_1}\left(x\right)$ and $q^0_{\theta_2}\left(v\right)$ separately.

- **Variants of deterministic Langevin embedding:** The expression ability of flow-based models is still restricted. We can exploit the deterministic Langevin embedding with separate potential function as the initialization. Specifically, we can also decompose $q^0_\theta\left(x, v\right) = q^0_{\theta_1}\left(x\right) q^0_{\theta_2}\left(v\right)$, for the sampler x, we exploit

  $$x^{t+1} = x^t + \epsilon\phi^t\left(x^t\right).$$

  Although we do not have the explicit $\log q^0_{\theta_1}\left(x\right)$, we can approximate it via either Taylor expansion or Chebyshev expansion as Section B.1. It should be emphasized that in such parametrization, in each layer we use different ϕ^t for $t = \{1, \ldots, T\}$, which are all different from $\nabla_x f\left(x\right)$.

- **Deep latent variable model:** We can also consider the model

  $$\begin{aligned} v &\sim q^0_{\theta_2}\left(v\right), & (39)\\ x &= \phi_{\theta_1}\left(v\right) + \epsilon, \quad \epsilon \sim \mathcal{N}\left(0, \Sigma\right), & (40)\end{aligned}$$

  where $q^0_{\theta_2}\left(v\right)$ is some known distribution with θ_2 as parameter and ϕ_{θ_1} denotes the neural network with θ_1 as parameter. Therefore, we have the distribution as

  $$q^0_\theta\left(x, v\right) = \mathcal{N}\left(x; \phi^0_{\theta_1}\left(v\right), \Sigma\right) q^0_{\theta_2}\left(v\right).$$

  For vanilla HMC with leap-frog, the auxiliary variable v should be the same size as x. However, for generalized HMC, the dimension of v can be smaller than that of x.

- **Nonparametric model:** We can also prefix the $q^0\left(x, v\right) = q^0\left(x\right) q^0\left(v\right)$ without learning. Specifically, we set $q^0\left(x\right)$ as the empirical $p_{\mathcal{D}}\left(x\right)$ and $q^0\left(v\right) = \mathcal{N}\left(0, \mathbf{I}\right)$. Since the initial distribution is fixed, the learning objective (8) reduces to

  $$\max_{f\in q} \min_{\Theta}\ \ell\left(f, \Theta\right) \propto \widehat{\mathbb{E}}_{\mathcal{D}}\left[f\right] - \mathbb{E}_{(x^0,v^0)\sim q^0(x,v)}\left[f\left(x^T\right) - \frac{1}{2}\left\|v^T\right\|_2^2\right]. \tag{41}$$

D Details for Connections to Other Estimators

We provide the details for recasting the existing estimators as special cases of our ADEas listed in Table 1.

D.1 Connection to Contrastive Divergence

The CD algorithm (Hinton, 2002) is a special case of the proposed algorithm. By Theorem 1, the optimal solution to the inner optimization is $p(x, v) \propto \exp(-\mathcal{H}(x, v))$. Applying Danskin's theorem (Bertsekas, 1995), the gradient of $L(f)$ w.r.t. f is

$$\nabla_f L(f) = \widehat{\mathbb{E}}_{\mathcal{D}} \left[\nabla_f f(x)\right] - \mathbb{E}_{p_f(x)} \left[\nabla_f f(x)\right]. \tag{42}$$

To estimate the integral $\mathbb{E}_{p_f}\left[\nabla_f f(x)\right]$, the CD algorithm approximates the negative term in (42) stochastically with a finite MCMC step away from empirical data.

In the proposed dual sampler, by setting $p_\theta^0(x)$ to be the empirical distribution and eliminating the sampling learning, the dynamic embedding will collapse to CD with T-HMC steps if we remove gradient through the sampler, *i.e.*, ignoring the third term in (21). Similarly, the persistent CD (PCD) (Tieleman, 2008) and recent ensemble CD (Du and Mordatch, 2019) can also be recast as special cases by setting the negative sampler to be MCMC with initial samples from previous model and ensemble of MCMC samplers, respectively.

From this perspective, the CD and PCD algorithms induce errors not only from the sampler, but also from the gradient back-propagation truncation. The proposed algorithm escapes these sources of bias by learning to sample, and by adopting true gradients, respectively. Therefore, the proposed estimator is expected to achieve better performance than CD as demonstrated in the empirical experiments Section 5.2.

D.2 Connection to Score Matching

The score matching (Hyvärinen, 2005) estimates the exponential family by minimizing the Fisher divergence, *i.e.*,

$$L_{SM}(f) := -\mathbb{E}_{\mathcal{D}}\left[\sum_{i=1}^{d}\left(\frac{1}{2}\left(\partial_i f(x)\right)^2\right) + \partial_i^2 f(x)\right]. \tag{43}$$

As explained in Hyvärinen (2007), the objective (43) can be derived as the 2nd-order Taylor approximation of the MLE with 1-step Langevin Monte Carlo as the dual sampler. Specifically, the Langevin Monte Carlo generates samples via

$$x' = x + \frac{\eta}{2}\nabla_x f(x) + \sqrt{\eta}\xi, \quad \xi \sim \mathcal{N}(0, I),$$

then, a simple Taylor expansion gives

$$\log p_f(x') = \log p_f(x) + \sum_{i=1}^{d} \partial_i f(x)\left(\frac{\eta}{2}\partial_i f(x) + \sqrt{\eta}\xi_i\right) + \eta\sum_{i,j=1}^{d} \xi_i\xi_j\partial_{ij}^2 f(x) + o(\eta).$$

Plug such into the negative expectation in $L(f)$, leading to

$$L(f) \approx \widehat{\mathbb{E}}_{\mathcal{D}}\left[\log p_f(x) - \mathbb{E}_{x'|x}\left[\log p_f(x')\right]\right] \approx -\eta\mathbb{E}_{\mathcal{D}}\left[\sum_{i=1}^{d}\left(\frac{1}{2}\left(\partial_i f(x)\right)^2\right) + \partial_i^2 f(x)\right],$$

which is exactly the scaled $L_{SM}(f)$ defined in (43).

Therefore, the score matching can be viewed as applying Taylor expansion approximation with fixed 1-step Langevin sampler in our framework, which is compared in Section 5.1.

D.3 Connection to Minimum Stein Discrepancy Estimator

The minimum Stein discrepancy estimator (Barp et al., 2019) is obtained by minimizing the Stein discrepancy, including the diffusion kernel Stein discrepancy (DKSD) and diffusion score matching. Without loss of the generality, for simplicity, we recast the DKSD with an identity diffusion matrix as a special approximation to the MLE.

The identity DKSD maximizes the following objective,

$$L_{DKSD}\left(f\right) := -\sup_{h\in\mathcal{H}_k, \|h\|_{\mathcal{H}_k}\leqslant 1} \widehat{\mathbb{E}}_{\mathcal{D}}\left[\mathcal{S}_f h\left(x\right)\right] = -\widehat{\mathbb{E}}_{x,x'\sim\mathcal{D}}\left[\mathcal{S}_f\left(x,\cdot\right)\otimes_k \mathcal{S}_f\left(x',\cdot\right)\right] \tag{44}$$

where $\mathcal{S}_f h\left(x\right) := \left\langle \mathcal{S}_f\left(x,\cdot\right), h\right\rangle = \left\langle \nabla_x f\left(x\right)^\top k\left(x,\cdot\right) + \nabla k\left(x,\cdot\right), h\right\rangle$.

In fact, the objective (44) can be derived as the Taylor approximation of the MLE with Stein variational gradient descent (SVGD) as the dual sampler. Specifically, the SVGD generates samples via

$$x' = T_{\mathcal{D},f}\left(x\right) := x + \eta h^*_{\mathcal{D},f}\left(x\right), \quad x\sim p_{\mathcal{D}}\left(x\right),$$

where $h^*_{\mathcal{D},f}\left(\cdot\right) \propto \mathbb{E}_{y\sim\mathcal{D}}\left[\mathcal{S}_f\left(y,\cdot\right)\right]$. Then, by Taylor-expansion, we have

$$f\left(x'\right) = f\left(x\right) + \eta\nabla_x f^\top\left(x\right) h^*_{\mathcal{D},f}\left(x\right) + o\left(\eta\right).$$

We apply the change-of-variable rule, leading to $q\left(x'\right) = p_{\mathcal{D}}\left(x\right)\det\left|\frac{\partial x}{\partial x'}\right|$, therefore,

$$\begin{aligned}
\log q(x') &= \log p_{\mathcal{D}}\left(x\right) + \log\det\left|\frac{\partial x}{\partial x'}\right| \\
&= \log p_{\mathcal{D}}\left(x\right) - \log\det\left|\frac{\partial x'}{\partial x}\right| \\
&= \log p_{\mathcal{D}}\left(x\right) - \log\det\left|I + \eta\nabla_x h^*_{\mathcal{D}}\left(x\right)\right| \\
&= \log p_{\mathcal{D}}\left(x\right) - \eta\operatorname{tr}\left(\nabla_x h^*_{\mathcal{D}}\left(x\right)\right),
\end{aligned}$$

where the last equation comes from Taylor expansion.

Plug these into the primal-dual view of MLE (5) with the fixed SVGD dual sampler, we have

$$\begin{aligned}
L\left(f\right) &\approx \widehat{\mathbb{E}}_{x\sim\mathcal{D}}\left[f\left(x\right) - f\left(x'\right) + \log q\left(x'\right)\right] \\
&= \widehat{\mathbb{E}}_{x\sim\mathcal{D}}\left[-\eta\nabla_x f^\top\left(x\right) h^*_{\mathcal{D},f}\left(x\right) - \eta\operatorname{tr}\left(\nabla_x h^*_{\mathcal{D}}\left(x\right)\right)\right] + \widehat{\mathbb{E}}_{x\sim\mathcal{D}}\left[\log p_{\mathcal{D}}\left(x\right)\right] + o\left(\eta\right) \\
&= -\eta\underbrace{\widehat{\mathbb{E}}_{x,x'\sim\mathcal{D}}\left[\mathcal{S}_f\left(x,\cdot\right)\otimes_k\mathcal{S}_f\left(x',\cdot\right)\right]}_{L_{DSKD}(f)} + \mathtt{const} + o\left(\eta\right),
\end{aligned}$$

which is the scaled $L_{DSKD}\left(f\right)$ defined in (44).

Therefore, the (diffusion) Stein kernel estimator can be viewed as Taylor expansion with fixed 1-step Stein variational gradient descent dual sampler in our framework.

D.4 Connection to Pseudo-Likelihood and Conditional Composite Likelihood

The pseudo-likelihood estimation (Besag, 1975) is a special case of the proposed algorithm by restricting the parametrization of the dual distribution. Specifically, denote the $p_f\left(x_i|x_{-i}\right) = \frac{\exp(f(x_i,x_{-i}))}{Z(x_{-i})}$ with $Z\left(x_{-i}\right) := \int\exp\left(f\left(x_i, x_{-i}\right)\right)dx_i$, instead of directly maximizing likelihood, the pseudo-likelihood estimator is maximizing

$$L_{PL}\left(f\right) := \widehat{\mathbb{E}}_{\mathcal{D}}\left[\sum_{i=1}^d \log p_f\left(x_i|x_{-i}\right)\right]. \tag{45}$$

Then, the f is updated by the following the gradient of $L_{pl}\left(f\right)$, *i.e.*,

$$\nabla_f L_{PL}\left(f\right) \propto \widehat{\mathbb{E}}_{\mathcal{D}}\left[\nabla_f f\left(x\right)\right] - \mathbb{E}_{i\sim\mathcal{U}(d)}\widehat{\mathbb{E}}_{x_{-i}}\mathbb{E}_{p_f(x_i|x_{-i})}\left[\nabla_f f\left(x_i, x_{-i}\right)\right].$$

The pseudo-likelihood estimator can be recast as a special case of the proposed framework if we fix the dual sampler as **i)**, sample $i\in\{1,\ldots,d\}$ uniformly; **ii)**, sample $x\sim\mathcal{D}$ and mask x_i; **iii)**, sample $x_i\sim p_f\left(x_i|x_{-1}\right)$ and compose $\left(x_i, x_{-i}\right)$.

The conditional composite likelihood (Lindsay, 1988) is a generalization of pseudo-likelihood by maximizing

$$L_{CL}\left(f\right) := \widehat{\mathbb{E}}_{\mathcal{D}}\left[\sum_{A_i=1}^m \log p_f\left(x_{A_i}|x_{-A_i}\right)\right], \tag{46}$$

where $\{A_i\}_{i=1}^m = d$ and $A_i\cap A_j = \emptyset$. Similarly, the composite likelihood is updating with prefixed conditional block sampler for negative sampling.

Same as CD, the prefixed sampler and the biased gradient in pseudo-likelihood and composite likelihood estimator will induce extra errors and lead to inferior solution. Moreover, the pseudo-likelihood may not applicable to the general exponential family with continuous variables, whose conditional distribution is also intractable.

D.5 Connection to Non-local Contrastive Objectives

The non-local contrastive estimator (Vickrey et al., 2010) is obtained by maximizing

$$L_{NCO}\left(f\right) := \widehat{\mathbb{E}}_{\mathcal{D}}\left[\sum_{i=1}^{m} w\left(x, S_i\right)\left(f\left(x\right) - \log Z_i\left(f\right)\right)\right], \tag{47}$$

where $\left[S_i\right]_{i=1}^{m}$ denotes some prefixed partition of Ω, $Z_i\left(f\right) = \int_{x\in S_i} \exp\left(f\left(x\right)\right) dx$, and $w\left(x, S_i\right) = P\left(x \in S_i | x\right)$ with $\sum_{i=1}^{m} w\left(x, S_i\right) = 1$. The objective (47) leads to the update direction as

$$\nabla_f L_{NCO}\left(f\right) = \widehat{\mathbb{E}}_{\mathcal{D}}\left[\nabla_f f\left(x\right)\right] - \mathbb{E}_{q_f(x)}\left[\nabla_f f\right], \tag{48}$$

where $q_f\left(x\right) = \sum_{i=1}^{m} \int p_{(f,i)}\left(x\right) w\left(x', S_i\right) p_{\mathcal{D}}\left(x'\right) dx'$ with $p_{\mathcal{D}}$ as the empirical distribution and $p_{(f,i)}\left(x\right) = \frac{\exp(f(x))}{Z_i(f)}$, $x \in S_i$. Therefore, the non-local contrastive objective is a special case of the proposed framework with the dual sampler as **i)**, sample x' uniformly from $\mathcal{D}$; **ii)**, sample S_i conditional on x' according to $w\left(x, S_i\right)$; **iii)**, sample $x_i \sim p_{(f,i)}\left(x\right)$ within S_i. Such negative sampling method is also not applicable to the general exponential family with continuous variables.

D.6 Connection to Minimum Probability Flow

In the continuous state model, the minimum probability flow (Sohl-Dickstein et al., 2011) estimates the exponential family by maximizing

$$L_{MPF}\left(f\right) := -\widehat{\mathbb{E}}_{x\sim\mathcal{D}}\mathbb{E}_{x'\sim\mathcal{T}_f(x'|x)}\left[\exp\left(\frac{1}{2}\left(f\left(x'\right) - f\left(x\right)\right)\right)\right],$$

where $\mathcal{T}_f$ is a *hand-designed* symmetric transition kernel based on the potential function $f\left(x\right)$, *e.g.*, Hamiltonian or Langevin simulation. Then, the MPF update direction can be rewritten as

$$\widehat{\mathbb{E}}_{x\sim\mathcal{D}}\mathbb{E}_{x'\sim\Gamma(x'|x)}\left[\nabla_f f\left(x\right) - \nabla_f f\left(x'\right) - \nabla_x f\left(x'\right)\nabla_f x'\right]. \tag{49}$$

where $\Gamma\left(x'|x\right) := \mathcal{T}_f(x'|x)\exp\left(\frac{1}{2}\left(f\left(x'\right) - f\left(x\right)\right)\right)$. The probability flow operator $\Gamma\left(x'|x\right)$ actually defines a Markov chain sampler that achieves the following balance equation,

$$\Gamma\left(x'|x\right) p_f\left(x\right) = \Gamma\left(x|x'\right) p_f\left(x'\right).$$

Similar to CD and score matching, the MPF exploits the 1-step MCMC. Moreover, the gradient in MPF also considers the effects in sampler as the third term in (49). Therefore, the MPF can be recast as a special case of our algorithm with the prefixed dual sampler as $x \sim \mathcal{D}$ and $x' \sim \Gamma\left(x'|x\right)$.

D.7 Connection to Noise-Contrastive Estimator

Instead of directly estimating the f in the exponential family, Gutmann and Hyvärinen (2010) propose the noise-contrastive estimation (NCE) for the density ratio between the exponential family and some user defined reference distribution $p_n\left(x\right)$, from which the parameter f can be reconstructed. Specifically, the NCE considers an alternative representation of exponential family distribution as $p_f\left(x\right) = \exp\left(f\left(x\right)\right)$, which explicitly enforces $\int \exp\left(f\left(x\right)\right) dx = 1$. The NCE is obtained by maximizing

$$L_{NCE}\left(f\right) := \widehat{\mathbb{E}}_{\mathcal{D}}\left[\log h\left(x\right)\right] + \mathbb{E}_{p_n(x)}\left[\log\left(1 - h\left(x\right)\right)\right], \tag{50}$$

where $h\left(x\right) = \frac{\exp(f(x))}{\exp(f(x)) + p_n(x)}$. Then, we have the gradient of $L_{NCE}\left(f\right)$ as

$$\nabla_f L_{NCE}\left(f\right) = \widehat{\mathbb{E}}_{\mathcal{D}}\left[\nabla_f f\left(x\right)\right] - \mathbb{E}_{\frac{1}{2}p_{\mathcal{D}} + \frac{1}{2}p_n}\left[h\left(x\right)\nabla_f f\left(x\right)\right]. \tag{51}$$

The negative sampler in the (51) can be understood as an approximate importance sampling algorithm where the proposal is $\frac{1}{2}p_{\mathcal{D}} + \frac{1}{2}p_n$ and the reweighting part is $h\left(x\right)$. As the $\exp\left(f\right)$ approaching $p_{\mathcal{D}}$, the $h\left(x\right)$ will approach the true ratio $\frac{\exp(f(x))}{p_{\mathcal{D}} + p_n(x)}$, and thus, the negative samples will converge to true model samples.

The NCE can be understood as learning an important sampler. However, the performance of NCE highly relies on the quality $h(x)$, *i.e.*, the choice of $p_n(x)$. It is required to cover the support of $p_{\mathcal{D}}(x)$, which is non-trivial in practical high-dimensional applications.

E More Related Work

The parametrization of the dual sampler should be both flexible enough and density tractable to achieve better performance. Pioneering works are limited in either one aspect or other. Kim and Bengio (2016) parameterize the sampler via a deep directed graphical model, whose approximation ability is limited to known distributions. Meanwhile, they fit q by minimizing the KL-divergence with an approximation of the entropy term, leading to unclear relationship to MLE. Due to the difficulty of the entropy term for general transport mapping parametrization, a variety of approximate surrogates have been proposed to relax the density value tractability requirement. Liu and Wang (2017) learn the sampler q to mimic the Stein variational gradient descent sampling procedure without a consistent objective; Dai et al. (2017) propose algorithms relying on either a heuristic approximation or a lower bound of the entropy, with extra auxiliary component introduced to be learned; Dai et al. (2019) apply a second Fenchel dual representation to reformulate the entropy term, at the cost of introducing another auxiliary function to be estimated. Meanwhile, the second Fenchel duality parametrization relies on a proposal distribution with the same support for numerical stability, which is impractical for high-dimensional data. In contrast to these existing methods, the proposed dynamics embedding achieves both flexibility and tractability of entropy estimation with less independent auxiliary parameters introduced.

One of our major contributions is learning a sampling strategy for the exponential family estimation through the primal-dual view of MLE. The proposed algorithm shares some similarities with recent advances in meta learning for sampling (Levy et al., 2018; Feng et al., 2017; Song et al., 2017; Gong et al., 2019), in which the sampler is parametrized via neural network and will be learned through certain objectives. However, we emphasize that the most significant difference lies in the ultimate goals: we focus on exponential family *model estimation* and the learned sampler is only introduced to *assist* with this objective. By contrast, the learning to sample techniques are targeting on learning a *fixed* model that is already given. This fundamentally distinguishes the proposed ADE from methods that only learns samplers, leading to totally different learning criterion and algorithm updates, *i.e.*, the primal model will be learned back through the learned sampler, from which perspective the proposed algorithm can be understood as meta^2-learning.

F Experiment Details

F.1 Synthetic Experiments Details

We parametrize the potential function f with fully connected multi-layer perceptron with 3 hidden layers. Each hidden layer has 128 hidden units. We use ReLU to do the nonlinear activation in each hidden layer. We clip the norm of $\nabla_x f$ when updating v, and clip v when updating x. The coefficient λ in (20) is tuned in $\{0.1, 0.5, 1\}$. For the NF baseline, we tune the number of layers in $\{10, 15, 20\}$. For our ADE, we fix the number of normalizing flow layers to be 10, and then perform at most 10 steps of dynamics updates. So finally, the number of steps for sampling is comparable, while the ADE maintains less memory cost.

To make the training stable, we also tried several tricks, including:

1. clip samples in HMC. This helps stabilize the training; We assume the final output has limited support over 2D space.
2. gradient penalty for $f(\cdot)$. We use a small penalty coefficient 0.01 for this, which is not very important though.
3. variance of proposal gaussian distribution. While we use 1 in general, a standard deviation of 0.5 would be more helpful in some cases.
4. penalty of momentum term in HMC. This is equivalent to the variance of prior of the latent variable we introduced.

The dataset generators are collect from several open-source projects [2] [3]. During training, we use this generator to generate the data from the true distribution on the fly. To get a quantitative comparison, we also generate 1,000 data samples for held-out evaluation. We illustrate the unnormalized model $\exp(c \cdot f)$ in Figure 1 and 5, where c is a constant that is tuned within $[0.01, 10]$.

To compute the MMD, for NF and ADE, we use 1,000 samples from their sampler with Gaussian kernel. The kernel bandwidth is chosen using median trick (Dai et al., 2016). For SM, since there is no such sampler available, we directly use vanilla HMC to get sample from the learned model f, and use them to estimate MMD.

Parameter estimation experiments In the experiment of recovering parameters of a given graphical model from data, we use high dimensional gaussian distribution with diagonal covariance. Here the energy function to be estimated $f(x) = -0.5(x-\mu)^\top \Sigma^{-1}(x-\mu)$, where Σ is a diagonal matrix.

For our method, we use a 2-layer MLP as initial proposal with 3 of HMC steps afterwards. The step size in HMC is learned end-to-end. For CD, we use up to 15 steps of HMC, where the step size is adaptively adjusted according to the rejection rate. For all the methods, we average the parameters estimated in the last 5 epochs during training, and report the best results in this parameter estimation procedure.

F.2 Real-world Experiments Details

Table 4: Our architectures for both potential function $f(x)$ and initial dual sampler $p^0_\theta(x, v)$ used in `MNIST` and `CIFAR-10` experiments.

Potential function $f(\cdot)$
3x3 conv, 64
3x3 conv, 128
2x2 avg pool
3x3 conv, 128
3x3 conv, 256
2x2 avg pool
3x3 conv, 256
7x7 avg pool
fc, 256 → 1

(a) Potential function $f(\cdot)$

Initial dual sampler
fc, 512 → $4 \times 4 \times 512$
Reshape to 4×4 Feature Map
2x2 Deconv, 256, stride 2
2x2 Deconv, 128, stride 2
2x2 Deconv, 64, stride 2
3x3 Deconv, 3, stride 1

(b) initial dual sampler

We used the standard spectral normalization on the discriminator to stabilize the training process, and Adam with learning rate 10^{-4} and $\beta_1 = 0.0$ to optimize our model. For stability, we use a separate Adam optimizer for the hmc parameters and set the epsilon to $1e-5$. We trained the models with 200000 iterations with batch size being 64. For better performance, we used generalized HMC (13), where we set $S_v(\cdot) = 0$, $S_x(\cdot) = 0$, $g_v(v) = \text{clip}(v, -0.01, 0.01)$ and $g_x(v^{1/2}) = v^{1/2}$. We fix η to be 0.5. The step sizes for our HMC sampler are independently learned for all HMC dimensions but shared among all time steps, and the values are all initialized to 10. We set the number of HMC steps to 30. The coefficient of the entropy regularization term is set to 10^{-5} and that of the L_2 regularization on the momentum vector in the last HMC step is set to 10^{-5}.

We demonstrate the architectures of potential function f and initial Deep LVM in Table 4. A leaky ReLU follows each convolutional/deconvolutional layer in both the discriminator and generator. For the discriminator, we use spectral normalization for all layers in the discriminator. In addition, there is no activation function after the final fully-connected layer. For each deconvolution layer in the generator, we insert a batch normalization layer before passing the output to the leaky ReLU.

We generate the image from the model and illustrated in Figure 3, Figure 6 and Figure 7. We also compared in terms of inception score with other energy-model training algorithm and several state-of-the-art GAN algorithm in Table 3, where the ADE achieves the best performances. Also,

[2] https://github.com/rtqichen/ffjord.

[3] https://github.com/kevin-w-li/deep-kexpfam.

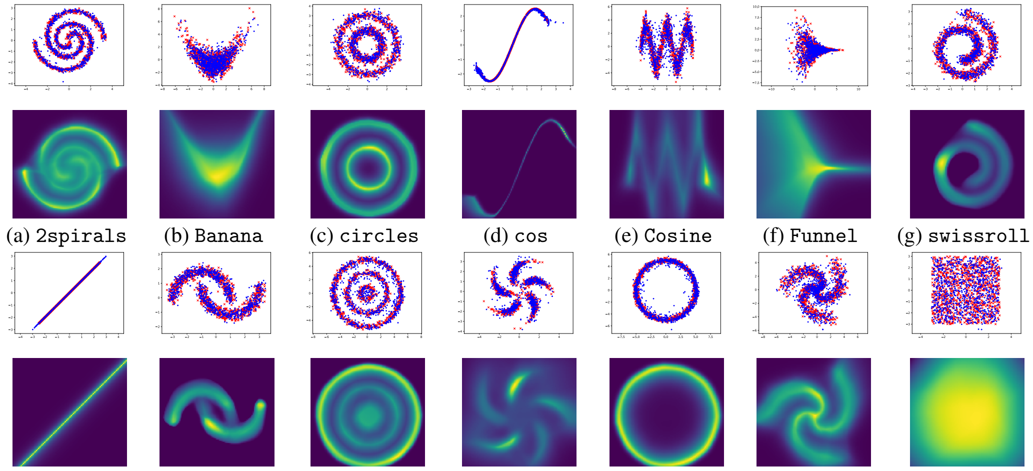
with simple importance sampling and proposal distribution being uniform distribution on $[-1, 1]^{n_d}$ (n_d is the dimension of images), the log likelihood (in nats) on `CIFAR-10` is estimated to be around 2100.

We also trained a non-parametric ADE on `MNIST` dataset for image completion to verify our algorithm. Specifically, we use with the same discriminator architecture used in parametric ADE for `MNIST`. The model is trained with fully observed images. We used generalized HMC (13), where we set $S_v(v)$ being a learnable logit (so that $\exp(S_v(\cdot)) \in [0, 1]$), $g_v(v) = \text{clip}(v, -0.1, 0.1)$, $S_x(\cdot) = 0$ and $g_x(\cdot) = 1$. Both S_v and η will be learned, with η initialized to $\sqrt{10}$ and S_v initialized to a small number close to 0. We unfold 60 steps of HMC in the dual samplers. As in Du and Mordatch (2019), we used a replay buffer of size 10000. We added extra amount of noise into the dataset to make the training process more stable. We trained the model with Adam optimizer ($\beta_1 = 0.0, \beta_2 = 0.999$) for 60000 iterations.

We tested the ADE by image completion where we covered the lower half of images with uniform noise and used them as input to the learned HMC operators. We repeatedly apply the learned HMC with the learned model to lower half of these images for 20 steps, with the upper half images fixed, and obtain $\text{HMC}^{(20)}(x_0; S_v, \eta)$. We visualize the output from each of the 20 HMC runs in Figure 4.

G More Experiment Results

More results on synthetic datasets We visualized the learned models and samplers on all the synthetic datasets in Figure 5.



(a) `2spirals` (b) `Banana` (c) `circles` (d) `cos` (e) `Cosine` (f) `Funnel` (g) `swissroll`

(h) `line` (i) `moons` (j) `Multiring` (k) `pinwheel` (l) `Ring` (m) `Spiral` (n) `Uniform`

Figure 5: Learned samplers in odd row and potential function f in even row from different synthetic datasets. In the sampler illustration in odd rows, the × denotes training data and • denotes the ADE samplers.

More results on parameters recovery We have conducted empirical comparion between ADE, CD and SM on multivariate Gaussians with different dimensions, where we know the potential functions, to investigate the effect of the number of dimensionality and complexity of the potential function on these algorithms.

Table 5: Parameter recovering on Multivariate Gaussians.

Dataset	SM	CD-5	ADE
`2D-Gaussian`	$\mathbf{2.18 \times 10^{-3}}$	5.67×10^{-3}	2.28×10^{-3}
`5D-Gaussian`	3.17×10^{-3}	4.19×10^{-1}	$\mathbf{3.09 \times 10^{-3}}$
`10D-Gaussian`	3.90×10^{-3}	6.36×10^{-1}	$\mathbf{3.23 \times 10^{-3}}$

The 5 runs average results, in terms of RMSE between learned parameters and the true parameters, are reported in Table 5.

More results on real-world image generation We illustrated additional generated images by the proposed ADE on MNIST and CIFAR-10 in Figure 6 and Figure 7, respectively.

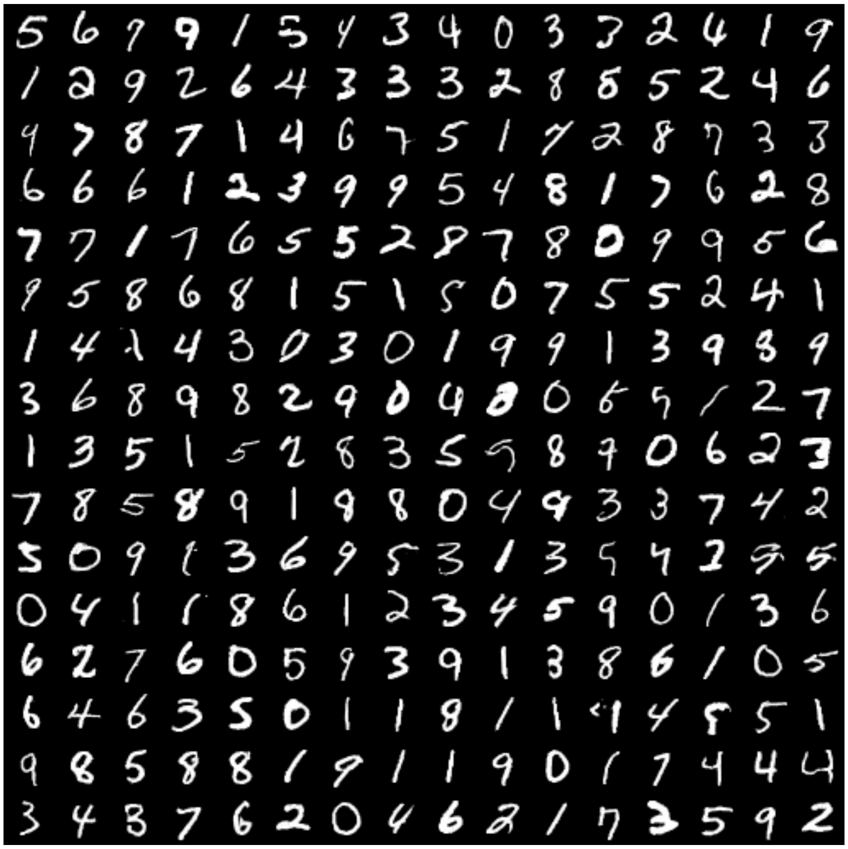


Figure 6: Generated images for MNIST by ADE.

Figure 7: Generated images for CIFAR-10 by ADE.