How to Train Your Energy-Based Models

Yang Song   YANGSONG@CS.STANFORD.EDU
Stanford University
Diederik P. Kingma   DPKINGMA@GOOGLE.COM
Google Research

Abstract

Energy-Based Models (EBMs), also known as non-normalized probabilistic models, specify probability density or mass functions up to an unknown normalizing constant. Unlike most other probabilistic models, EBMs do not place a restriction on the tractability of the normalizing constant, thus are more flexible to parameterize and can model a more expressive family of probability distributions. However, the unknown normalizing constant of EBMs makes training particularly difficult. Our goal is to provide a friendly introduction to modern approaches for EBM training. We start by explaining maximum likelihood training with Markov chain Monte Carlo (MCMC), and proceed to elaborate on MCMC-free approaches, including Score Matching (SM) and Noise Constrastive Estimation (NCE). We highlight theoretical connections among these three approaches, and end with a brief survey on alternative training methods, which are still under active research. Our tutorial is targeted at an audience with basic understanding of generative models who want to apply EBMs or start a research project in this direction.

1. Introduction

Probabilistic models with a tractable likelihood are a double-edged sword. On one hand, a tractable likelihood allows for straightforward comparison between models, and straightforward optimization of the model parameters w.r.t. the log-likelihood of the data. Through tractable models such as autoregressive (Graves, 2013; Germain et al., 2015; Van Oord et al., 2016) or flow-based generative models (Dinh et al., 2014, 2016; Rezende and Mohamed, 2015), we can learn flexible models of high-dimensional data. In some cases even though the likelihood is not completely tractable, we can often compute and optimize a tractable lower bound of the likelihood, as in the framework of variational autoencoders (Kingma and Welling, 2014; Rezende et al., 2014).

Still, the set of models with a tractable likelihood is constrained. Models with a tractable likelihood need to be of a certain form: for example, in case of autoregressive models, the model distribution is factorized as a product of conditional distributions, and in flow-based generative models the data is modeled as an invertible transformation of a base distribution. In case of variational autoencoders, the data must be modeled as a directed latent-variable model. A tractable likelihood is related to the fact that these models assume that exact synthesis of pseudo-data from the model can be done with a specified, tractable procedure. These assumptions are not always natural.

Energy-based models (EBM) are much less restrictive in functional form: instead of specifying a normalized probability, they only specify the unnormalized negative log-probability,

called the *energy function*. Since the energy function does not need to integrate to one, it can be parameterized with any nonlinear regression function. In the framework of EBMs, density estimation is thus basically reduced to a nonlinear regression problem. One is generally free to choose any nonlinear regression function as the energy function, as long as it remains normalizable in principle. It is thus straightforward to leverage advances in architectures originally developed for classification or regression, and one may choose special-purpose architectures that make sense for the type of data at hand. For example, If the data are graphs (such as molecules) then one could use graph neural networks (Scarselli et al., 2008); if the data are spherical images one can in principle use spherical CNNs (Cohen et al., 2018). As such, EBMs have found wide applications in many fields of machine learning, including, among others, image generation (Ngiam et al., 2011; Xie et al., 2016; Du and Mordatch, 2019), discriminative learning (Grathwohl et al., 2020b), natural language processing (Mikolov et al., 2013; Deng et al., 2020), density estimation (Wenliang et al., 2019; Song et al., 2019) and reinforcement learning (Haarnoja et al., 2017, 2018).

Although this flexibility of EBMs can provide significant modeling advantages, both computation of the exact likelihood and exact synthesis of samples from these models are generally intractable, which makes training especially difficult. There are three major ways for training EBMs: (i) maximum likelihood training with MCMC sampling; (ii) Score Matching (SM); and (iii) Noise Constrastive Estimation (NCE). We will elaborate on these methods in order, explain their relationships to each other, and conclude by an overview to other directions for EBM training.

2. Energy-Based Models (EBMs)

For simplicity we will assume unconditional Energy-Based Models over a single dependent variable $\mathbf{x}$. It is relatively straightforward to extend the models and estimation procedures to the case with multiple dependent variables, or with conditioning variables. The density given by an EBM is

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \frac{\exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))}{Z_{\boldsymbol{\theta}}} \tag{1}$$

where $E_{\boldsymbol{\theta}}(\mathbf{x})$ (the energy) is a nonlinear regression function with parameters $\boldsymbol{\theta}$, and $Z_{\boldsymbol{\theta}}$ denotes the normalizing constant (a.k.a. the partition function):

$$Z_{\boldsymbol{\theta}} = \int \exp(-E_{\boldsymbol{\theta}}(\mathbf{x})) \, \mathrm{d}\mathbf{x} \tag{2}$$

which is constant w.r.t $\mathbf{x}$ but is a function of $\boldsymbol{\theta}$. Since $Z_{\boldsymbol{\theta}}$ is a function of $\boldsymbol{\theta}$, evaluation and differentiation of $\log p_{\boldsymbol{\theta}}(\mathbf{x})$ w.r.t. its parameters involves a typically intractable integral.

3. Maximum Likelihood Training with MCMC

The *de facto* standard for learning probabilistic models from i.i.d. data is maximum likelihood estimation (MLE). Let $p_{\boldsymbol{\theta}}(\mathbf{x})$ be a probabilistic model parameterized by $\boldsymbol{\theta}$, and $p_{\mathrm{data}}(\mathbf{x})$ be the underlying data distribution of a dataset. We can fit $p_{\boldsymbol{\theta}}(\mathbf{x})$ to $p_{\mathrm{data}}(\mathbf{x})$ by maximizing the expected log-likelihood function over the data distribution, defined by

$$\mathbb{E}_{\mathbf{x} \sim p_{\mathrm{data}}(\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x})]$$

as a function of $\boldsymbol{\theta}$. Here the expectation can be easily estimated with samples from the dataset. Maximizing likelihood is equivalent to minimizing the KL divergence between $p_{\text{data}}(\mathbf{x})$ and $p_{\boldsymbol{\theta}}(\mathbf{x})$, because

$$\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x})] = D_{KL}(p_{\text{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})) - \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[\log p_{\text{data}}(\mathbf{x})]$$
$$= D_{KL}(p_{\text{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})) - \text{constant},$$

where the second equality holds because $\mathbb{E}_{\mathbf{x} \sim p_{\text{data}}(\mathbf{x})}[\log p_{\text{data}}(\mathbf{x})]$ does not depend on $\boldsymbol{\theta}$.

We cannot directly compute the likelihood of an EBM as in the maximum likelihood approach due to the intractable normalizing constant $Z_{\boldsymbol{\theta}}$. Nevertheless, we can still estimate the gradient of the log-likelihood with MCMC approaches, allowing for likelihood maximization with gradient ascent (Younes, 1999). In particular, the gradient of the log-probability of an EBM (Eq. (1)) decomposes as a sum of two terms:

$$\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) = -\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}) - \nabla_{\boldsymbol{\theta}} \log Z_{\boldsymbol{\theta}}. \tag{3}$$

The first gradient term, $-\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x})$, is straightforward to evaluate with automatic differentiation. The challenge is in approximating the second gradient term, $\nabla_{\boldsymbol{\theta}} \log Z_{\boldsymbol{\theta}}$, which is intractable to compute exactly. This gradient term can be rewritten as the following expectation:

$$\nabla_{\boldsymbol{\theta}} \log Z_{\boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} \log \int \exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))d\mathbf{x}$$
$$\stackrel{(i)}{=} \left(\int \exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))d\mathbf{x} \right)^{-1} \nabla_{\boldsymbol{\theta}} \int \exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))d\mathbf{x}$$
$$= \left(\int \exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))d\mathbf{x} \right)^{-1} \int \nabla_{\boldsymbol{\theta}} \exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))d\mathbf{x}$$
$$\stackrel{(ii)}{=} \left(\int \exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))d\mathbf{x} \right)^{-1} \int \exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))(-\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}))d\mathbf{x}$$
$$= \int \left(\int \exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))d\mathbf{x} \right)^{-1} \exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))(-\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}))d\mathbf{x}$$
$$\stackrel{(iii)}{=} \int \exp(-E_{\boldsymbol{\theta}}(\mathbf{x}) - Z_{\boldsymbol{\theta}})(-\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}))d\mathbf{x}$$
$$\stackrel{(iv)}{=} \int p_{\boldsymbol{\theta}}(\mathbf{x})(-\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}))d\mathbf{x}$$
$$= \mathbb{E}_{\mathbf{x} \sim p_{\boldsymbol{\theta}}(\mathbf{x})}\left[-\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x}) \right],$$

where steps (i) and (ii) are due to the chain rule of gradients, and (iii) and (iv) are from definitions in Eqs. (1) and (2). Thus, we can obtain an unbiased one-sample Monte Carlo estimate of the log-likelihood gradient by

$$\nabla_{\boldsymbol{\theta}} \log Z_{\boldsymbol{\theta}} \simeq -\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}), \tag{4}$$

where $\tilde{\mathbf{x}} \sim p_{\boldsymbol{\theta}}(\mathbf{x})$, *i.e.*, a random sample from the distribution over $\mathbf{x}$ given by the EBM. Therefore, as long as we can draw random samples from the model, we have access to an

unbiased Monte Carlo estimate of the log-likelihood gradient, allowing us to optimize the parameters with stochastic gradient ascent.

Since drawing random samples is far from being trivial, much of the literature has focused on methods for efficient MCMC sampling from EBMs. Some efficient MCMC methods, such as Langevin MCMC (Parisi, 1981; Grenander and Miller, 1994) and Hamiltonian Monte Carlo (Duane et al., 1987; Neal et al., 2011), make use of the fact that the gradient of the log-probability w.r.t. $\mathbf{x}$ (a.k.a., *score*) is equal to the (negative) gradient of the energy, therefore easy to calculate:

$$\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) = -\nabla_{\mathbf{x}} E_{\boldsymbol{\theta}}(\mathbf{x}) - \underbrace{\nabla_{\mathbf{x}} \log Z_{\boldsymbol{\theta}}}_{=0} = -\nabla_{\mathbf{x}} E_{\boldsymbol{\theta}}(\mathbf{x}). \tag{5}$$

For example, when using Langevin MCMC to sample from $p_{\boldsymbol{\theta}}(\mathbf{x})$, we first draw an initial sample $\mathbf{x}^0$ from a simple prior distribution, and then simulate an (overdamped) Langevin diffusion process for K steps with step size $\epsilon > 0$:

$$\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k + \frac{\epsilon^2}{2} \underbrace{\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}^k)}_{=-\nabla_{\mathbf{x}} E_{\boldsymbol{\theta}}(\mathbf{x})} + \epsilon \mathbf{z}^k, \quad k = 0, 1, \cdots, K - 1. \tag{6}$$

When $\epsilon \to 0$ and $K \to \infty$, $\mathbf{x}^K$ is guaranteed to distribute as $p_{\boldsymbol{\theta}}(\mathbf{x})$ under some regularity conditions. In practice we have to use a small finite ϵ, but the discretization error is typically negligible, or can be corrected with a Metropolis-Hastings (Hastings, 1970) step, leading to the Metropolis-Adjusted Langevin Algorithm (Besag, 1994).

Running MCMC till convergence to obtain a sample $\mathbf{x} \sim p_{\boldsymbol{\theta}}(\mathbf{x})$ can be computationally expensive. Therefore we typically need approximation to make MCMC-based learning of EBMs practical. One popular method for doing so is Contrastive Divergence (CD) (Hinton, 2002). In CD, one initializes the MCMC chain from the datapoint $\mathbf{x}$, and perform a fixed number of MCMC steps; typically fewer than required for convergence of the MCMC chain. One variant of CD that sometimes performs better is persistent CD (Tieleman, 2008), where a single MCMC chain with a persistent state is employed to sample from the EBM. In persistent CD, we do not restart the MCMC chain when training on a new datapoint; rather, we carry over the state of the previous MCMC chain and use it to initialize a new MCMC chain for the next training step. This method can be further improved by keeping multiple historical states of the MCMC chain in a replay buffer and initialize new MCMC chains by randomly sampling from it (Du and Mordatch, 2019). Other variants of CD include mean field CD (Welling and Hinton, 2002), and multi-grid CD (Gao et al., 2018).

EBMs trained with CD may not capture the data distribution faithfully, since truncated MCMC can lead to biased gradient updates that hurt the learning dynamics (Schulz et al., 2010; Fischer and Igel, 2010; Nijkamp et al., 2019). There are several methods that focus on removing this bias for improved MCMC training. For example, one line of work proposes unbiased estimators of the gradient through coupled MCMC (Jacob et al., 2017; Qiu et al., 2019); and Du et al. (2020) propose to reduce the bias by differentiating through the MCMC sampling algorithm and estimating an entropy correction term.

4

4. Score Matching (SM)

If two continuously differentiable real-valued functions $f(\mathbf{x})$ and $g(\mathbf{x})$ have equal first derivatives everywhere, then $f(\mathbf{x}) \equiv g(\mathbf{x}) + \text{constant}$. When $f(\mathbf{x})$ and $g(\mathbf{x})$ are log probability density functions (PDFs) with equal first derivatives, the normalization requirement (Eq. (1)) implies that $\int \exp(f(\mathbf{x}))\mathrm{d}\mathbf{x} = \int \exp(g(\mathbf{x}))\mathrm{d}\mathbf{x} = 1$, and therefore $f(\mathbf{x}) \equiv g(\mathbf{x})$. As a result, one can learn an EBM by (approximately) matching the first derivatives of its log-PDF to the first derivatives of the log-PDF of the data distribution. If they match, then the EBM captures the data distribution exactly. The first-order gradient function of a log-PDF is also called the *score* of that distribution. For training EBMs, it is useful to transform the equivalence of distributions to the equivalence of scores, because the score of an EBM can be easily obtained by $\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) = -\nabla_{\mathbf{x}} E_{\boldsymbol{\theta}}(\mathbf{x})$ (recall from Eq. (5)) which does not involve the typically intractable normalizing constant $Z_{\boldsymbol{\theta}}$.

Let $p_{\text{data}}(\mathbf{x})$ be the underlying data distribution, from which we have a finite number of i.i.d. samples but do not know its PDF. The basic Score Matching (Hyvärinen, 2005) (SM) objective minimizes a discrepancy between two distributions called the Fisher divergence:

$$D_F(p_{\text{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[\frac{1}{2} \left\| \nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}) \right\|^2 \right]. \tag{7}$$

The expectation w.r.t. $p_{\text{data}}(\mathbf{x})$, in this objective and its variants below, admits a trivial unbiased Monte Carlo estimator using the empirical mean of samples $\mathbf{x} \sim p_{\text{data}}(\mathbf{x})$. However, the second term of Eq. (7) is generally impractical to calculate since it requires knowing $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$. Hyvärinen (2005) shows that under certain regularity conditions, the Fisher divergence can be rewritten using integration by parts, with second derivatives of $E_{\boldsymbol{\theta}}(\mathbf{x})$ replacing the unknown first derivatives of $p_{\text{data}}(\mathbf{x})$:

$$D_F(p_{\text{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[\frac{1}{2} \sum_{i=1}^{d} \left(\frac{\partial E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i} \right)^2 + \frac{\partial^2 E_{\boldsymbol{\theta}}(\mathbf{x})}{(\partial x_i)^2} \right] + \text{constant} \tag{8}$$

where d is the dimensionality of $\mathbf{x}$. The constant does not affect optimization and thus can be dropped for training. It is shown by Hyvärinen (2005) that estimators based on Score Matching are consistent under some regularity conditions, meaning that the parameter estimator obtained by minimizing Eq. (7) converges to the true parameters in the limit of infinite data.

An important downside of the objective Eq. (8) is that, in general, computation of full second derivatives is quadratic in the dimensionality d, thus does not scale to high dimensionality. Although SM only requires the trace of the Hessian, it is still expensive to compute even with modern hardware and automatic differentiation packages (Martens et al., 2012). For this reason, the implicit SM formulation of Eq. (8) has only been applied to relatively simple energy functions where computation of the second derivatives is tractable.

Score Matching assumes a continuous data distribution with positive density over the space, but it can be generalized to discrete or bounded data distributions (Hyvärinen, 2007; Lyu, 2012). It is also possible to consider higher-order gradients of log-PDFs beyond first derivatives (Parry et al., 2012).

4.1 Denoising Score Matching (DSM)

The Score Matching objective in Eq. (8) requires several regularity conditions for $\log p_{\text{data}}(\mathbf{x})$, *e.g.*, it should be continuously differentiable and finite everywhere. However, these conditions may not always hold in practice. For example, a distribution of digital images is typically discrete and bounded, because the values of pixels are restricted to the range $\{0, 1, \cdots, 255\}$. Therefore, $\log p_{\text{data}}(\mathbf{x})$ in this case is discontinuous and is negative infinity outside the range, and therefore SM is not directly applicable.

To alleviate this difficulty, one can add a bit of noise to each datapoint: $\tilde{\mathbf{x}} = \mathbf{x} + \boldsymbol{\epsilon}$. As long as the noise distribution $p(\boldsymbol{\epsilon})$ is smooth, the resulting noisy data distribution $q(\tilde{\mathbf{x}}) = \int q(\tilde{\mathbf{x}} \mid \mathbf{x}) p_{\text{data}}(\mathbf{x}) d\mathbf{x}$ is also smooth, and thus the Fisher divergence $D_F(q(\tilde{\mathbf{x}}) \parallel p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}}))$ is a proper objective. Kingma and LeCun (2010) showed that the objective with noisy data can be approximated by the noiseless Score Matching objective of Eq. (8) plus a regularization term; this regularization makes Score Matching applicable to a wider range of data distributions, but still requires expensive second-order derivatives.

Vincent (2011) propose one elegant and scalable solution to the above difficulty, by showing that:

$$D_F(q(\tilde{\mathbf{x}}) \parallel p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})) = \mathbb{E}_{q(\tilde{\mathbf{x}})} \left[\frac{1}{2} \|\nabla_{\mathbf{x}} \log q(\tilde{\mathbf{x}}) - \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})\|_2^2 \right] \tag{9}$$

$$= \mathbb{E}_{q(\mathbf{x}, \tilde{\mathbf{x}})} \left[\frac{1}{2} \|\nabla_{\mathbf{x}} \log q(\tilde{\mathbf{x}}|\mathbf{x}) - \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})\|_2^2 \right] + \text{constant}, \tag{10}$$

where the expectation is again approximated by the empirical average of samples, thus completely avoiding both the unknown term $p_{\text{data}}(\mathbf{x})$ and computationally expensive second-order derivatives. The constant term does not affect optimization and can be ignored without changing the optimal solution. This estimation method is called *Denoising Score Matching* (DSM) by Vincent (2011). Similar formulations are also explored by Raphan and Simoncelli (2007, 2011) and can be traced back to Tweedie's formula (1956) and Stein's Unbiased Risk Estimation (Stein, 1981).

The major drawback of adding noise to data arises when $p_{\text{data}}(\mathbf{x})$ is already a well-behaved distribution that satisfies the regularity conditions required by Score Matching. In this case, $D_F(q(\tilde{\mathbf{x}}) \parallel p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})) \neq D_F(p_{\text{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x}))$, and DSM is not a consistent objective because the optimal EBM matches the noisy distribution $q(\tilde{\mathbf{x}})$, not $p_{\text{data}}(\mathbf{x})$. This inconsistency becomes non-negligible when $q(\tilde{\mathbf{x}})$ significantly differs from $p_{\text{data}}(\mathbf{x})$.

One way to attenuate the inconsistency of DSM is to choose $q \approx p_{\text{data}}$, *i.e.*, use a small noise perturbation. However, this often significantly increases the variance of objective values and hinders optimization. As an example, suppose $q(\tilde{\mathbf{x}} \mid \mathbf{x}) = \mathcal{N}(\tilde{\mathbf{x}} \mid \mathbf{x}, \sigma^2 I)$ and $\sigma \approx 0$. The corresponding DSM objective is

$$D_F(q(\tilde{\mathbf{x}}) \parallel p_{\boldsymbol{\theta}}(\tilde{\mathbf{x}})) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathbb{E}_{\mathbf{z} \sim \mathcal{N}(0,I)} \left[\frac{1}{2} \left\| \frac{\mathbf{z}}{\sigma} + \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x} + \sigma \mathbf{z}) \right\|_2^2 \right]$$

$$\simeq \frac{1}{2N} \sum_{i=1}^{N} \left\| \frac{\mathbf{z}^{(i)}}{\sigma} + \nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)} + \sigma \mathbf{z}^{(i)}) \right\|_2^2, \tag{11}$$

where $\{\mathbf{x}^{(i)}\}_{i=1}^N \overset{\text{i.i.d.}}{\sim} p_{\text{data}}(\mathbf{x})$, and $\{\mathbf{z}^{(i)}\}_{i=1}^N \overset{\text{i.i.d.}}{\sim} \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. When $\sigma \to 0$, we can leverage Taylor series expansion to rewrite the Monte Carlo estimator in Eq. (11) to

$$\frac{1}{2N}\sum_{i=1}^N \left[\frac{2}{\sigma}(\mathbf{z}^{(i)})^\mathsf{T}\nabla_\mathbf{x} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}) + \frac{\left\|\mathbf{z}^{(i)}\right\|_2^2}{\sigma^2}\right] + \text{constant.} \tag{12}$$

When estimating the above expectation with samples, the variances of $(\mathbf{z}^{(i)})^\mathsf{T}\nabla_\mathbf{x} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)})/\sigma$ and $\left\|\mathbf{z}^{(i)}\right\|_2^2/\sigma^2$ will both grow unbounded as $\sigma \to 0$ due to division by σ and σ^2. This enlarges the variance of DSM and makes optimization challenging.

Wang et al. (2020) propose a method to reduce the variance of Eq. (12) when $\sigma \approx 0$. Note that the terms in Eq. (12) have closed-form expectations: $\mathbb{E}_{\mathbf{x},\mathbf{z}}[2\mathbf{z}^\mathsf{T}\nabla_\mathbf{x} \log p_{\boldsymbol{\theta}}(\mathbf{x})/\sigma] = 0$ and $\mathbb{E}_{\mathbf{z}}[\|\mathbf{z}\|_2^2/\sigma^2] = d/\sigma^2$. Therefore, we can construct a variable that is, for sufficiently small σ, positively correlated with Eq. (11) while having an expected value of zero:

$$c_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z}) = \frac{2}{\sigma}\mathbf{z}^\mathsf{T}\nabla_\mathbf{x} \log p_{\boldsymbol{\theta}}(\mathbf{x}) + \frac{\|\mathbf{z}\|_2^2}{\sigma^2} - \frac{d}{\sigma^2}. \tag{13}$$

Substracting it from Eq. (11) will yield an estimator with reduced variance for DSM training:

$$\frac{1}{2N}\sum_{i=1}^N \left\|\frac{\mathbf{z}^{(i)}}{\sigma} + \nabla_\mathbf{x} \log p_{\boldsymbol{\theta}}(\mathbf{x}^{(i)} + \sigma\mathbf{z}^{(i)})\right\|_2^2 - c_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z}^{(i)}). \tag{14}$$

Variance-reducing variables like $c_{\boldsymbol{\theta}}(\mathbf{x}, \mathbf{z})$ are called *control variates* (Owen, 2013). For large σ, the Taylor series might be a bad approximator, in which case $c_{\boldsymbol{\theta}}(\mathbf{x}^{(i)}, \mathbf{z}^{(i)})$ might not be sufficiently correlated with Eq. (11), and could actually increase variance.

4.2 Sliced Score Matching (SSM)

By adding noise to data, DSM avoids the expensive computation of second-order derivatives. However, as mentioned before, the optimal EBM that minimizes the DSM objective corresponds to the distribution of noise-perturbed data $q(\tilde{\mathbf{x}})$, not the original noise-free data distribution $p_{\text{data}}(\mathbf{x})$. In other words, DSM does not give a consistent estimator of the data distribution, *i.e.*, one cannot directly obtain an EBM that exactly matches the data distribution even with unlimited data.

Sliced Score Matching (SSM) (Song et al., 2019) is one alternative to Denoising Score Matching that is both consistent and computationally efficient. Instead of minimizing the Fisher divergence between two vector-valued scores, SSM randomly samples a projection vector $\mathbf{v}$, takes the inner product between $\mathbf{v}$ and the two scores, and then compare the resulting two scalars. More specifically, Sliced Score Matching minimizes the following divergence called the sliced Fisher divergence

$$D_{SF}(p_{\text{data}}(\mathbf{x})\|p_{\boldsymbol{\theta}}(\mathbf{x})) = \mathbb{E}_{p_{\text{data}}(\mathbf{x})}\mathbb{E}_{p(\mathbf{v})}\left[\frac{1}{2}(\mathbf{v}^\mathsf{T}\nabla_\mathbf{x} \log p_{\text{data}}(\mathbf{x}) - \mathbf{v}^\mathsf{T}\nabla_\mathbf{x} \log p_{\boldsymbol{\theta}}(\mathbf{x}))^2\right],$$

where $p(\mathbf{v})$ denotes a projection distribution such that $\mathbb{E}_{p(\mathbf{v})}[\mathbf{v}\mathbf{v}^\mathsf{T}]$ is positive definite. Similar to Fisher divergence, sliced Fisher divergence has an implicit form that does not involve the

unknown $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$, which is given by

$$D_{SF}(p_{\text{data}}(\mathbf{x}) \| p_{\boldsymbol{\theta}}(\mathbf{x}))$$

$$= \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathbb{E}_{p(\mathbf{v})} \left[\frac{1}{2} \sum_{i=1}^{d} \left(\frac{\partial E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i} v_i \right)^2 + \sum_{i=1}^{d} \sum_{j=1}^{d} \frac{\partial^2 E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i \partial x_j} v_i v_j \right] + \text{constant}. \quad (15)$$

All expectations in the above objective can be estimated with empirical means, and again the constant term can be removed without affecting training. The second term involves second-order derivatives of $E_{\boldsymbol{\theta}}(\mathbf{x})$, but contrary to SM, it can be computed efficiently with a cost linear in the dimensionality d. This is because

$$\sum_{i=1}^{d} \sum_{j=1}^{d} \frac{\partial^2 E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i \partial x_j} v_i v_j = \sum_{i=1}^{d} \frac{\partial}{\partial x_i} \underbrace{\left(\sum_{j=1}^{d} \frac{\partial E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_j} v_j \right)}_{:=f(\mathbf{x})} v_i, \quad (16)$$

where $f(\mathbf{x})$ is the same for different values of i. Therefore, we only need to compute it once with $O(d)$ computation, *plus* another $O(d)$ computation for the outer sum to evaluate Eq. (16), whereas the original SM objective requires $O(d^2)$ computation.

For many choices of $p(\mathbf{v})$, part of the SSM objective (Eq. (15)) can be evaluated in closed form, potentially leading to lower variance. For example, when $p(\mathbf{v}) = \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, we have

$$\mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathbb{E}_{p(\mathbf{v})} \left[\frac{1}{2} \sum_{i=1}^{d} \left(\frac{\partial E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i} v_i \right)^2 \right] = \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \left[\frac{1}{2} \sum_{i=1}^{d} \left(\frac{\partial E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i} \right)^2 \right]$$

and as a result,

$$D_{SF}(p_{\text{data}}(\mathbf{x}) \| p_{\boldsymbol{\theta}}(\mathbf{x}))$$

$$= \mathbb{E}_{p_{\text{data}}(\mathbf{x})} \mathbb{E}_{\mathbf{v} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})} \left[\frac{1}{2} \sum_{i=1}^{d} \left(\frac{\partial E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i} \right)^2 + \sum_{i=1}^{d} \sum_{j=1}^{d} \frac{\partial^2 E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i \partial x_j} v_i v_j \right] + \text{constant}. \quad (17)$$

The above objective Eq. (17) can also be obtained by approximating the sum of second-order gradients in the standard SM objective (Eq. (8)) with the Skilling-Hutchinson trace estimator (Skilling, 1989; Hutchinson, 1989). It often (but not always) has lower variance than Eq. (15), and can perform better in some applications (Song et al., 2019).

4.3 Connection to Contrastive Divergence

Though Score Matching and Contrastive Divergence are seemingly very different approaches, they are closely connected to each other. In fact, Score Matching can be viewed as a special instance of Contrastive Divergence in the limit of a particular MCMC sampler (Hyvarinen, 2007). Moreover, the Fisher divergence optimized by Score Matching is related to the derivative of KL divergence (Cover, 1999), which is the underlying objective of Contrastive Divergence.

Contrastive Divergence requires sampling from the Energy-Based Model $E_{\boldsymbol{\theta}}(\mathbf{x})$, and one popular method for doing so is Langevin MCMC (Parisi, 1981). Recall from Section 3 that given any initial data point $\mathbf{x}^0$, the Langevin MCMC method executes the following

$$\mathbf{x}^{k+1} \leftarrow \mathbf{x}^k - \frac{\epsilon^2}{2}\nabla_{\mathbf{x}}E_{\boldsymbol{\theta}}(\mathbf{x}^k) + \epsilon\, \mathbf{z}^k,$$

iteratively for $k = 0, 1, \cdots, K-1$, where $\mathbf{z}^k \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ and $\epsilon > 0$ is the step size.

Suppose we only run one-step Langevin MCMC (*i.e.*, $K = 1$) for Contrastive Divergence. In this case, the gradient of the log-likelihood is given by

$$\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})}[\nabla_{\boldsymbol{\theta}} \log p_{\boldsymbol{\theta}}(\mathbf{x})] = -\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})}[\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x})] + \mathbb{E}_{\mathbf{x}\sim p_{\boldsymbol{\theta}}(\mathbf{x})}\left[\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x})\right]$$

$$\simeq -\mathbb{E}_{p_{\mathrm{data}}(\mathbf{x})}[\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim\mathcal{N}(\mathbf{0},\boldsymbol{I})}\left[\nabla_{\boldsymbol{\theta}} E_{\boldsymbol{\theta}}\left(\mathbf{x} - \frac{\epsilon^2}{2}\nabla_{\mathbf{x}}E_{\boldsymbol{\theta}'}(\mathbf{x}) + \epsilon\, \mathbf{z}\right)\bigg|_{\boldsymbol{\theta}'=\boldsymbol{\theta}}\right].$$

After Taylor series expansion with respect to ϵ followed by some algebraic manipulations, the above equation can be transformed to the following (see Hyvarinen (2007))

$$\frac{\epsilon^2}{2}\nabla_{\boldsymbol{\theta}} D_F(p_{\mathrm{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x})) + o(\epsilon^2).$$

When ϵ is sufficiently small, it corresponds to the re-scaled gradient of the Score Matching objective.

In general, Score Matching minimizes the Fisher divergence $D_F(p_{\mathrm{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x}))$, whereas Contrastive Divergence minimizes the KL divergence $D_{KL}(p_{\mathrm{data}}(\mathbf{x}) \parallel p_{\boldsymbol{\theta}}(\mathbf{x}))$. The above connection of Score Matching and Contrastive Divergence is a natural consequence of the connection between those two statistical divergences, as characterized by a relative version of the *de Bruijin's identity* (Cover, 1999; Lyu, 2012):

$$\frac{d}{dt} D_{KL}(q_t(\tilde{\mathbf{x}}) \parallel p_{\boldsymbol{\theta},t}(\tilde{\mathbf{x}})) = -\frac{1}{2} D_F(q_t(\tilde{\mathbf{x}}) \parallel p_{\boldsymbol{\theta},t}(\tilde{\mathbf{x}})).$$

Here $q_t(\tilde{\mathbf{x}})$ and $p_{\boldsymbol{\theta},t}(\tilde{\mathbf{x}})$ denote smoothed versions of $p_{\mathrm{data}}(\mathbf{x})$ and $p_{\boldsymbol{\theta}}(\mathbf{x})$, resulting from adding Gaussian noise to $\mathbf{x}$ with variance t; *i.e.*, $\tilde{\mathbf{x}} \sim \mathcal{N}(\mathbf{x}, t\boldsymbol{I})$.

4.4 Score-Based Generative Models

One typical application of EBMs is creating new samples that are similar to training data. Towards this end, we can first train an EBM with Score Matching, and then sample from it with MCMC approaches. Many efficient sampling methods for EBMs, such as Langevin MCMC, rely on just the score of the EBM (see Eq. (6)). In addition, Score Matching objectives (Eqs. (8), (10) and (15)) depend sorely on the scores of EBMs. Therefore, we only need a model for score when training with Score Matching and sampling with score-based MCMC, and do not have to model the energy explicitly. By building such a score model, we save the gradient computation of EBMs and can make training and sampling more efficient. These kind of models are named score-based generative models (Song and Ermon, 2019, 2020; Song et al., 2020).

Score Matching has difficulty in estimating the relative weights of two modes separated by large low-density regions (Wenliang et al., 2019; Song and Ermon, 2019), which can have
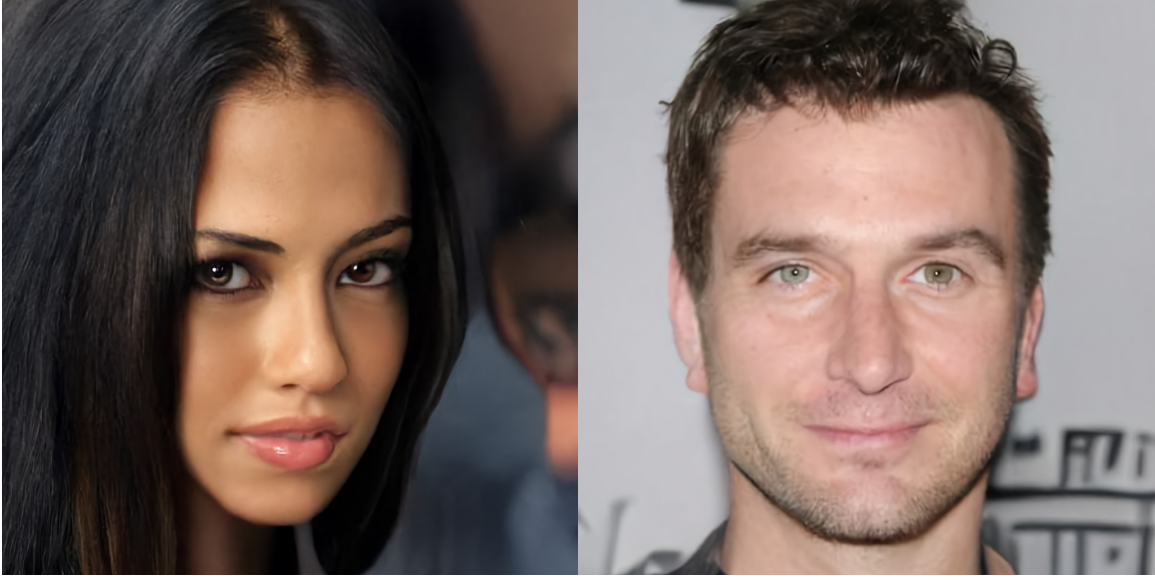
9

Figure 1: Samples from a score-based generative model trained with multiple scales of noise perturbations (resolution 1024×1024). Image credit to Song et al. (2020).

an important negative impact on sample generation. As an example, suppose $p_{\text{data}}(\mathbf{x}) = \pi p_0(\mathbf{x}) + (1 - \pi)p_1(\mathbf{x})$. Let $\mathcal{S}_0 := \{\mathbf{x} \mid p_0(\mathbf{x}) > 0\}$ and $\mathcal{S}_1 := \{\mathbf{x} \mid p_1(\mathbf{x}) > 0\}$ be the supports of $p_0(\mathbf{x})$ and $p_1(\mathbf{x})$ respectively. When they are disjoint from each other, the score of $p_{\text{data}}(\mathbf{x})$ is given by

$$\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x}) = \begin{cases} \nabla_{\mathbf{x}} \log p_0(\mathbf{x}), & \mathbf{x} \in \mathcal{S}_0 \\ \nabla_{\mathbf{x}} \log p_1(\mathbf{x}), & \mathbf{x} \in \mathcal{S}_1, \end{cases}$$

which does not depend on the weight π. Since Score Matching trains an EBM by matching its score to the score of data, $\nabla_{\mathbf{x}} \log p_{\text{data}}(\mathbf{x})$, which contains no information of π in this case, it is impossible for the learned EBM to recover the correct weight of $p_0(\mathbf{x})$ or $p_1(\mathbf{x})$. In practice, the regularity conditions of Score Matching actually require $p_{\text{data}}(\mathbf{x}) > 0$ everywhere, so $\mathcal{S}_0$ and $\mathcal{S}_1$ cannot be completely disjoint from each other, but when they are close to being mutually disjoint (which often happens in real data especially in high-dimensional space), it will be very hard to learn the weights accurately with Score Matching. When the weights are not accurate, samples will concentrate around different data modes with an inappropriate portion, leading to worse sample quality.

Song and Ermon (2019, 2020) and Song et al. (2020) overcome this difficulty by perturbing training data with different scales of noise, and learn a score model for each scale. For a large noise perturbation, different modes are connected due to added noise, and estimated weights between them are therefore accurate. For a small noise perturbation, different modes are more disconnected, but the noise-perturbed distribution is closer to the original unperturbed data distribution. Using a sampling method such as annealed Langevin dynamics (Song and Ermon, 2019, 2020; Song et al., 2020) or leveraging reverse diffusion processes (Sohl-Dickstein

et al., 2015; Ho et al., 2020; Song et al., 2020), we can sample from the most noise-perturbed distribution first, then smoothly reduce the magnitude of noise scales until reaching the smallest one. This procedure helps combine information from all noise scales, and maintains the correct portion of modes from larger noise perturbations when sampling from smaller ones. In practice, all score models share weights and are implemented with a single neural network conditioned on the noise scale, named a Noise-Conditional Score Network. Scores of different scales are estimated by training a mixture of Score Matching objectives, one per noise scale. This method are amongst the best generative modeling approaches for high-resolution image generation (see samples in Fig. 1), audio synthesis (Chen et al., 2020; Kong et al., 2020), and shape generation (Cai et al., 2020).

5. Noise Contrastive Estimation

A third principle for learning the parameters of EBMs is *Noise Contrastive Estimation* (NCE), introduced by Gutmann and Hyvärinen (2010). It is based on the idea that we can learn an Energy-Based Model by contrasting it with another distribution with known density.

Let $p_{\text{data}}(\mathbf{x})$ be our data distribution, and let $p_{\text{n}}(\mathbf{x})$ be a chosen distribution with known density, called a noise distribution. This noise distribution is usually simple and has a tractable PDF, like $\mathcal{N}(\mathbf{0}, \boldsymbol{I})$, such that we can compute the PDF and generate samples from it efficiently. Strategies exist to learn the noise distribution, as referenced below. Let y be a binary variable with Bernoulli distribution, which we use to define a mixture distribution of noise and data: $p_{\text{n,data}}(\mathbf{x}) = p(y=0)p_{\text{n}}(\mathbf{x}) + p(y=1)p_{\text{data}}(\mathbf{x})$. According to the Bayes' rule, given a sample $\mathbf{x}$ from this mixture, the posterior probability of $y=0$ is

$$p_{\text{n,data}}(y=0 \mid \mathbf{x}) = \frac{p_{\text{n,data}}(\mathbf{x} \mid y=0)p(y=0)}{p_{\text{n,data}}(\mathbf{x})} = \frac{p_{\text{n}}(\mathbf{x})}{p_{\text{n}}(\mathbf{x}) + \nu p_{\text{data}}(\mathbf{x})} \tag{18}$$

where $\nu = p(y=1)/p(y=0)$.

Suppose our Energy-Based Model $p_{\boldsymbol{\theta}}(\mathbf{x})$ has the form

$$p_{\boldsymbol{\theta}}(\mathbf{x}) = \exp(-E_{\boldsymbol{\theta}}(\mathbf{x}))/Z_{\boldsymbol{\theta}}.$$

Contrary to most other EBMs, $Z_{\boldsymbol{\theta}}$ is treated as a learnable (scalar) parameter in NCE. Given this model, similar to the mixture of noise and data above, we can define a mixture of noise and the model distribution: $p_{\text{n},\boldsymbol{\theta}}(\mathbf{x}) = p(y=0)p_{\text{n}}(\mathbf{x}) + p(y=1)p_{\boldsymbol{\theta}}(\mathbf{x})$. The posterior probability of $y=0$ given this noise/model mixture is

$$p_{\text{n},\boldsymbol{\theta}}(y=0 \mid \mathbf{x}) = \frac{p_{\text{n}}(\mathbf{x})}{p_{\text{n}}(\mathbf{x}) + \nu p_{\boldsymbol{\theta}}(\mathbf{x})} \tag{19}$$

In NCE, we indirectly fit $p_{\boldsymbol{\theta}}(\mathbf{x})$ to $p_{\text{data}}(\mathbf{x})$ by fitting $p_{\text{n},\boldsymbol{\theta}}(y \mid \mathbf{x})$ to $p_{\text{n,data}}(y \mid \mathbf{x})$ through a standard conditional maximum likelihood objective:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{p_{\text{n,data}}(\mathbf{x})}[D_{KL}(p_{\text{n,data}}(y \mid \mathbf{x}) \parallel p_{\text{n},\boldsymbol{\theta}}(y \mid \mathbf{x}))] \tag{20}$$

$$= \arg\max_{\boldsymbol{\theta}} \mathbb{E}_{p_{\text{n,data}}(\mathbf{x},y)}[\log p_{\text{n},\boldsymbol{\theta}}(y \mid \mathbf{x})], \tag{21}$$

which can be solved using stochastic gradient ascent. Just like any other deep classifier, when the model is sufficiently powerful, $p_{\mathrm{n},\boldsymbol{\theta}^*}(y \mid \mathbf{x})$ will match $p_{\mathrm{n,data}}(y \mid \mathbf{x})$ at the optimum. In that case:

$$p_{\mathrm{n},\boldsymbol{\theta}^*}(y = 0 \mid \mathbf{x}) \equiv p_{\mathrm{n,data}}(y = 0 \mid \mathbf{x}) \tag{22}$$

$$\Longleftrightarrow \frac{p_{\mathrm{n}}(\mathbf{x})}{p_{\mathrm{n}}(\mathbf{x}) + \nu p_{\boldsymbol{\theta}^*}(\mathbf{x})} \equiv \frac{p_{\mathrm{n}}(\mathbf{x})}{p_{\mathrm{n}}(\mathbf{x}) + \nu p_{\mathrm{data}}(\mathbf{x})} \tag{23}$$

$$\Longleftrightarrow p_{\boldsymbol{\theta}^*}(\mathbf{x}) \equiv p_{\mathrm{data}}(\mathbf{x}) \tag{24}$$

Consequently, $E_{\boldsymbol{\theta}^*}(\mathbf{x})$ is an unnormalized energy function that matches the data distribution $p_{\mathrm{data}}(\mathbf{x})$, and $Z_{\boldsymbol{\theta}^*}$ is the corresponding normalizing constant.

As one unique feature that Contrastive Divergence and Score Matching do not have, NCE provides the normalizing constant of an Energy-Based Model as a by-product of its training procedure. When the EBM is very expressive, *e.g.*, a deep neural network with many parameters, we can assume it is able to approximate a normalized probability density and absorb $Z_{\boldsymbol{\theta}}$ into the parameters of $E_{\boldsymbol{\theta}}(\mathbf{x})$ (Mnih and Teh, 2012), or equivalently, fixing $Z_{\boldsymbol{\theta}} = 1$. The resulting EBM trained with NCE will be self-normalized, *i.e.*, having a normalizing constant close to 1.

In practice, choosing the right noise distribution $p_{\mathrm{n}}(\mathbf{x})$ is critical to the success of NCE, especially for structured and high-dimensional data. As argued in Gutmann and Hirayama (2012), NCE works the best when the noise distribution is close to the data distribution (but not exactly the same). Many methods have been proposed to automatically tune the noise distribution, such as Adversarial Contrastive Estimation (Bose et al., 2018), Conditional NCE (Ceylan and Gutmann, 2018) and Flow Contrastive Estimation (Gao et al., 2020). NCE can be further generalized using Bregman divergences (Gutmann and Hirayama, 2012), where the formulation introduced here reduces to a special case.

5.1 Connection to Score Matching

Noise Contrastive Estimation provides a family of objectives that vary for different $p_{\mathrm{n}}(\mathbf{x})$ and ν. This flexibility may allow adaptation to special properties of a task with hand-tuned $p_{\mathrm{n}}(\mathbf{x})$ and ν, and may also give a unified perspective for different approaches. In particular, when using an appropriate $p_{\mathrm{n}}(\mathbf{x})$ and a slightly different parameterization of $p_{\mathrm{n},\boldsymbol{\theta}}(y \mid \mathbf{x})$, we can recover Score Matching from NCE (Gutmann and Hirayama, 2012).

For example, we choose the noise distribution $p_{\mathrm{n}}(\mathbf{x})$ to be a perturbed data distribution: given a small (deterministic) vector $\mathbf{v}$, let $p_{\mathrm{n}}(\mathbf{x}) = p_{\mathrm{data}}(\mathbf{x} - \mathbf{v})$. It is efficient to sample from this $p_{\mathrm{n}}(\mathbf{x})$, since we can first draw any datapoint $\mathbf{x}' \sim p_{\mathrm{data}}(\mathbf{x}')$ and then compute $\mathbf{x} = \mathbf{x}' + \mathbf{v}$. It is, however, difficult to evaluate the density of $p_{\mathrm{n}}(\mathbf{x})$ because $p_{\mathrm{data}}(\mathbf{x})$ is unknown. Since the original parameterization of $p_{\mathrm{n},\boldsymbol{\theta}}(y \mid \mathbf{x})$ in NCE (Eq. (19)) depends on the PDF of $p_{\mathrm{n}}(\mathbf{x})$, we cannot directly apply the standard NCE objective. Instead, we replace $p_{\mathrm{n}}(\mathbf{x})$ with $p_{\boldsymbol{\theta}}(\mathbf{x} - \mathbf{v})$ and parameterize $p_{\mathrm{n},\boldsymbol{\theta}}(y = 0 \mid \mathbf{x})$ with the following form

$$p_{\mathrm{n},\boldsymbol{\theta}}(y = 0 \mid \mathbf{x}) := \frac{p_{\boldsymbol{\theta}}(\mathbf{x} - \mathbf{v})}{p_{\boldsymbol{\theta}}(\mathbf{x}) + p_{\boldsymbol{\theta}}(\mathbf{x} - \mathbf{v})} \tag{25}$$

In this case, the NCE objective (Eq. (21)) reduces to

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[\log(1 + \exp(E_{\boldsymbol{\theta}}(\mathbf{x}) - E_{\boldsymbol{\theta}}(\mathbf{x} - \mathbf{v}))) + \log(1 + \exp(E_{\boldsymbol{\theta}}(\mathbf{x}) - E_{\boldsymbol{\theta}}(\mathbf{x} + \mathbf{v})))] \quad (26)$$

At $\boldsymbol{\theta}^*$, we have a solution where

$$p_{\text{n},\boldsymbol{\theta}^*}(y = 0 \mid \mathbf{x}) \equiv p_{\text{n,data}}(y = 0 \mid \mathbf{x}) \tag{27}$$

$$\Longleftrightarrow \frac{p_{\boldsymbol{\theta}^*}(\mathbf{x} - \mathbf{v})}{p_{\boldsymbol{\theta}^*}(\mathbf{x}) + p_{\boldsymbol{\theta}^*}(\mathbf{x} - \mathbf{v})} \equiv \frac{p_{\text{data}}(\mathbf{x} - \mathbf{v})}{p_{\text{data}}(\mathbf{x}) + p_{\text{data}}(\mathbf{x} - \mathbf{v})} \tag{28}$$

$$\Longleftrightarrow p_{\boldsymbol{\theta}^*}(\mathbf{x}) \equiv p_{\text{data}}(\mathbf{x}), \tag{29}$$

i.e., the optimal model matches the data distribution.

As noted in Gutmann and Hirayama (2012) and Song et al. (2019), when $\|\mathbf{v}\|_2 \approx 0$, the NCE objective Eq. (26) has the following equivalent form by Taylor expansion

$$\arg\min_{\boldsymbol{\theta}} \frac{1}{4}\mathbb{E}_{p_{\text{data}}(\mathbf{x})}\left[\frac{1}{2}\sum_{i=1}^{d}\left(\frac{\partial E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i}v_i\right)^2 + \sum_{i=1}^{d}\sum_{j=1}^{d}\frac{\partial^2 E_{\boldsymbol{\theta}}(\mathbf{x})}{\partial x_i \partial x_j}v_i v_j\right] + 2\log 2 + o(\|\mathbf{v}\|_2^2).$$

Comparing against Eq. (15), we immediately see that the above objective equals that of SSM, if we ignore small additional terms hidden in $o(\|\mathbf{v}\|_2^2)$ and take the expectation with respect to $\mathbf{v}$ over a user-specified distribution $p(\mathbf{v})$.

6. Other Methods

Aside from MCMC-based training, Score Matching and Noise Contrastive Estimation, there are also other methods for learning EBMs. Below we briefly survey some examples of them. Interested readers can learn more details from references therein.

6.1 Minimizing Differences/Derivatives of KL Divergences

The overarching strategy for learning probabilistic models from data is to minimize the KL divergence between data and model distributions. However, because the normalizing constants of EBMs are typically intractable, it is hard to directly evaluate the KL divergence when the model is an EBM (see the discussion in Section 3). One generic idea that has frequently circumvented this difficulty is to consider differences or derivatives (*i.e.*, infinitesimal differences) of KL divergences. It turns out that the unknown partition functions of EBMs are often cancelled out after taking the difference of two closely related KL divergences, or computing the derivatives.

Typical examples of this strategy include minimum velocity learning (Movellan, 2008; Wang et al., 2020), minimum probability flow (Sohl-Dickstein et al., 2011) and minimum KL contraction (Lyu, 2011). In minimum velocity learning and minimum probability flow, a Markov chain is designed such that it starts from the data distribution $p_{\text{data}}(\mathbf{x})$ and converges to the EBM distribution $p_{\boldsymbol{\theta}}(\mathbf{x}) = e^{-E_{\boldsymbol{\theta}}(\mathbf{x})}/Z_{\boldsymbol{\theta}}$. Specifically, the Markov chain satisfies $p_0(\mathbf{x}) \equiv p_{\text{data}}(\mathbf{x})$ and $p_\infty(\mathbf{x}) \equiv p_{\boldsymbol{\theta}}(\mathbf{x})$, where we denote by $p_t(\mathbf{x})$ the state distribution at time $t \geq 0$.

This Markov chain will evolve towards $p_{\boldsymbol{\theta}}(\mathbf{x})$ unless $p_{\text{data}}(\mathbf{x}) \equiv p_{\boldsymbol{\theta}}(\mathbf{x})$. Therefore, we can fit the EBM distribution $p_{\boldsymbol{\theta}}(\mathbf{x})$ to $p_{\text{data}}(\mathbf{x})$ by minimizing the modulus of the "velocity" of this evolution, defined by

$$\frac{\mathrm{d}}{\mathrm{d}t} D_{\text{KL}}(p_t(\mathbf{x}) \,\|\, p_{\boldsymbol{\theta}}(\mathbf{x}))\bigg|_{t=0} \quad \text{or} \quad \frac{\mathrm{d}}{\mathrm{d}t} D_{\text{KL}}(p_{\text{data}}(\mathbf{x}) \,\|\, p_t(\mathbf{x}))\bigg|_{t=0}$$

in minimum velocity learning and minimum probability flow respectively. These objectives typically do not require computing the normalizing constant $Z_{\boldsymbol{\theta}}$.

In minimum KL contraction (Lyu, 2011), a distribution transformation Φ is chosen such that $D_{\text{KL}}(p(\mathbf{x}) \,\|\, q(\mathbf{x})) \geq D_{\text{KL}}(\Phi\{p(\mathbf{x})\} \,\|\, \Phi\{q(\mathbf{x})\})$, with equality if and only if $p(\mathbf{x}) \equiv q(\mathbf{x})$. We can leverage this Φ to train an EBM, by minimizing

$$D_{\text{KL}}(p_{\text{data}}(\mathbf{x}) \,\|\, p_{\boldsymbol{\theta}}(\mathbf{x})) - D_{\text{KL}}(\Phi\{p_{\text{data}}(\mathbf{x})\} \,\|\, \Phi\{p_{\boldsymbol{\theta}}(\mathbf{x})\}).$$

This objective does not require computing the partition function $Z_{\boldsymbol{\theta}}$ whenever Φ is linear.

Minimum velocity learning, minimum probability flow, and minimum KL contraction are all different generalizations to Score Matching and Noise Contrastive Estimation (Movellan, 2008; Sohl-Dickstein et al., 2011; Lyu, 2011).

6.2 Minimizing the Stein Discrepancy

We can train EBMs by minimizing the Stein discrepancy, defined by

$$D_{\text{Stein}}(p_{\text{data}}(\mathbf{x}) \,\|\, p_{\boldsymbol{\theta}}(\mathbf{x})) := \sup_{\mathbf{f} \in \mathcal{F}} \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[\nabla_{\mathbf{x}} \log p_{\boldsymbol{\theta}}(\mathbf{x})^{\mathsf{T}} \mathbf{f}(\mathbf{x}) + \text{trace}(\nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}))], \tag{30}$$

where $\mathcal{F}$ is a family of vector-valued functions, and $\nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x})$ denotes the Jacobian of $\mathbf{f}(\mathbf{x})$. With some regularity conditions (Gorham and Mackey, 2015; Liu et al., 2016), we have $D_{\text{Stein}}(p_{\text{data}}(\mathbf{x}) \,\|\, p_{\boldsymbol{\theta}}(\mathbf{x})) \geq 0$, where the equality holds if and only if $p_{\text{data}}(\mathbf{x}) \equiv p_{\boldsymbol{\theta}}(\mathbf{x})$. Similar to Score Matching (Eq. (8)), the objective Eq. (30) only involves the score function of $p_{\boldsymbol{\theta}}(\mathbf{x})$, and does not require computing the EBM's partition function. Still, the trace term in Eq. (30) may demand expensive computation, and does not scale well to high dimensional data.

There are two common methods to sidestep this difficulty. Gorham and Mackey (2015) and Liu et al. (2016) discovered that when $\mathcal{F}$ is a unit ball in a Reproducing Kernel Hilbert Space (RKHS) with a fixed kernel, the Stein discrepancy becomes kernelized Stein discrepancy, where the trace term is a constant and does not affect optimization. Otherwise, $\text{trace}(\nabla_{\mathbf{x}} \mathbf{f}(\mathbf{x}))$ can be approximated with the Skilling-Hutchinson trace estimator (Skilling, 1989; Hutchinson, 1989; Grathwohl et al., 2020c).

6.3 Adversarial Training

Recall from Section 3 that when training EBMs with maximum likelihood estimation (MLE), we need to sample from the EBM per training iteration. However, sampling using multiple MCMC steps is expensive and requires careful tuning of the Markov chain. One way to avoid this difficulty is to use non-MLE methods that do not need sampling, such as Score Matching and Noise Contrastive Estimation. Here we introduce another family of methods that sidestep costly MCMC sampling by learning an auxiliary model through adversarial training, which allows fast sampling.

Using the definition of EBMs, we can rewrite the maximum likelihood objective by introducing a variational distribution $q_\phi(\mathbf{x})$ parameterized by ϕ:

$$\begin{aligned}
\mathbb{E}_{p_{\text{data}}(\mathbf{x})}[\log p_{\boldsymbol{\theta}}(\mathbf{x})] &= \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[-E_{\boldsymbol{\theta}}(\mathbf{x})] - \log Z_{\boldsymbol{\theta}} \\
&= \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[-E_{\boldsymbol{\theta}}(\mathbf{x})] - \log \int e^{-E_{\boldsymbol{\theta}}(\mathbf{x})} \mathrm{d}\mathbf{x} \\
&= \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[-E_{\boldsymbol{\theta}}(\mathbf{x})] - \log \int q_\phi(\mathbf{x}) \frac{e^{-E_{\boldsymbol{\theta}}(\mathbf{x})}}{q_\phi(\mathbf{x})} \mathrm{d}\mathbf{x} \\
&\overset{(i)}{\leq} \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[-E_{\boldsymbol{\theta}}(\mathbf{x})] - \int q_\phi(\mathbf{x}) \log \frac{e^{-E_{\boldsymbol{\theta}}(\mathbf{x})}}{q_\phi(\mathbf{x})} \mathrm{d}\mathbf{x} \\
&= \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[-E_{\boldsymbol{\theta}}(\mathbf{x})] - \mathbb{E}_{q_\phi(\mathbf{x})}[-E_{\boldsymbol{\theta}}(\mathbf{x})] - H(q_\phi(\mathbf{x})), \quad (31)
\end{aligned}$$

where $H(q_\phi(\mathbf{x}))$ denotes the entropy of $q_\phi(\mathbf{x})$. Step (i) is due to Jensen's inequality. Eq. (31) provides an upper bound to the expected log-likelihood. For EBM training, we can first minimize the upper bound Eq. (31) with respect to $q_\phi(\mathbf{x})$ so that it is closer to the likelihood objective, and then maximize Eq. (31) with respect to $E_{\boldsymbol{\theta}}(\mathbf{x})$ as a surrogate for maximizing likelihood. This amounts to using the following maximin objective

$$\max_{\boldsymbol{\theta}} \min_{\phi} \mathbb{E}_{q_\phi(\mathbf{x})}[E_{\boldsymbol{\theta}}(\mathbf{x})] - \mathbb{E}_{p_{\text{data}}(\mathbf{x})}[E_{\boldsymbol{\theta}}(\mathbf{x})] - H(q_\phi(\mathbf{x})). \quad (32)$$

Optimizing the above objective is similar to training Generative Adversarial Networks (GANs) (Goodfellow et al., 2014) and can be achieved by adversarial training. The variational distribution $q_\phi(\mathbf{x})$ should allow both fast sampling and efficient entropy evaluation to make Eq. (32) tractable. This limits the model family of $q_\phi(\mathbf{x})$, and usually restricts our choice to invertible probabilistic models, such as inverse autoregressive flow (Kingma et al., 2016), and NICE/RealNVP (Dinh et al., 2014, 2016). See Dai et al. (2019b) for an example on designing $q_\phi(\mathbf{x})$ and training EBMs with Eq. (32).

Kim and Bengio (2016) and Zhai et al. (2016) propose to represent $q_\phi(\mathbf{x})$ with neural samplers, like the generator of GANs. A neural sampler is a deterministic mapping g_ϕ that maps a random Gaussian noise $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$ directly to a sample $\mathbf{x} = g_\phi(\mathbf{z})$. When using a neural sampler as $q_\phi(\mathbf{x})$, it is efficient to draw samples through the deterministic mapping, but $H(q_\phi(\mathbf{x}))$ is intractable since the density of $q_\phi(\mathbf{x})$ is unknown. Kim and Bengio (2016) and Zhai et al. (2016) propose several heuristics to approximate this entropy function. Kumar et al. (2019) propose to estimate the entropy through its connection to mutual information: $H(q_\phi(\mathbf{z})) = I(g_\phi(\mathbf{z}), \mathbf{z})$, which can be estimated from samples with variational lower bounds (Nguyen et al., 2010; Nowozin et al., 2016; Belghazi et al., 2018). Dai et al. (2019a) notice that when defining $p_{\boldsymbol{\theta}}(\mathbf{x}) = p_0(\mathbf{x}) e^{-E_{\boldsymbol{\theta}}(\mathbf{x})}/Z_{\boldsymbol{\theta}}$, with $p_0(\mathbf{x})$ being a fixed base distribution, the entropy term $-H(q_\phi(\mathbf{x}))$ in Eq. (32) equates $D_{\text{KL}}(q_\phi(\mathbf{x}) \parallel p_0(\mathbf{x}))$, which can likewise be approximated with variational lower bounds using samples from $q_\phi(\mathbf{x})$ and $p_0(\mathbf{x})$, without requiring the density of $q_\phi(\mathbf{x})$.

Grathwohl et al. (2020a) represent $q_\phi(\mathbf{x})$ as a noisy neural sampler, where samples are obtained via $g_\phi(\mathbf{z}) + \sigma\boldsymbol{\epsilon}$, assuming $\mathbf{z}, \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. With a noisy neural sampler, $\nabla_\phi H(q_\phi(\mathbf{x}))$ becomes particularly easy to estimate, which allows gradient-based optimization for the maximin objective in Eq. (32). A related approach is proposed in Xie et al. (2018), where authors train a noisy neural sampler with samples obtained from MCMC, and initialize new

15

MCMC chains with samples generated from the neural sampler. This cooperative sampling scheme improves the convergence of MCMC, but may still require multiple MCMC steps for sample generation. It does not directly optimize the objective in Eq. (31).

When using both adversarial training and MCMC sampling, Yu et al. (2020) observe that EBMs can be trained with an arbitrary f-divergence, including KL, reverse KL, total variation, Hellinger, *etc*. The method proposed by Yu et al. (2020) allows us to explore the trade-offs and inductive bias of different statistical divergences for more flexible EBM training.

7. Conclusion

We reviewed some of the modern approaches for EBM training. In particular, we focused on maximum likelihood estimation with MCMC sampling, Score Matching, and Noise Contrastive Estimation. We emphasized their mutual connections, and concluded by a short review on other EBM training approaches that do not directly fall into these three categories. The contents of this tutorial is of course limited to the authors' knowledge and bias in the field; we did not cover many other important aspects of EBMs, including EBMs with latent variables, and various downstream applications of EBMs. Training techniques are crucial to problem solving with EBMs, and will remain an active direction for future research.

References

Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeshwar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and Devon Hjelm. Mutual information neural estimation. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 531–540, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR. URL http://proceedings.mlr.press/v80/belghazi18a.html.

JE Besag. Comments on "representations of knowledge in complex systems" by u. grenander and mi miller. *J. Roy. Statist. Soc. Ser. B*, 56:591–592, 1994.

Avishek Joey Bose, Huan Ling, and Yanshuai Cao. Adversarial contrastive estimation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1021–1032, 2018.

Ruojin Cai, Guandao Yang, Hadar Averbuch-Elor, Zekun Hao, Serge Belongie, Noah Snavely, and Bharath Hariharan. Learning gradient fields for shape generation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

Ciwan Ceylan and Michael U Gutmann. Conditional noise-contrastive estimation of unnormalised models. In *International Conference on Machine Learning*, pages 726–734, 2018.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. *arXiv preprint arXiv:2009.00713*, 2020.

Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical CNNs. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=Hkbd5xZRb.

Thomas M Cover. *Elements of information theory.* John Wiley & Sons, 1999.

Bo Dai, Hanjun Dai, Arthur Gretton, Le Song, Dale Schuurmans, and Niao He. Kernel exponential family estimation via doubly dual embedding. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 2321–2330. PMLR, 2019a.

Bo Dai, Zhen Liu, Hanjun Dai, Niao He, Arthur Gretton, Le Song, and Dale Schuurmans. Exponential family estimation via adversarial dynamics embedding. In *Advances in Neural Information Processing Systems*, pages 10979–10990, 2019b.

Yuntian Deng, Anton Bakhtin, Myle Ott, Arthur Szlam, and Marc'Aurelio Ranzato. Residual energy-based models for text generation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1l4SgHKDH.

Laurent Dinh, David Krueger, and Yoshua Bengio. NICE: Non-linear independent components estimation. *arXiv preprint arXiv:1410.8516*, 2014.

Laurent Dinh, Jascha Sohl-Dickstein, and Samy Bengio. Density estimation using real nvp. *arXiv preprint arXiv:1605.08803*, 2016.

Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 3608–3618. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8619-implicit-generation-and-modeling-with-energy-based-models.pdf.

Yilun Du, Shuang Li, Joshua Tenenbaum, and Igor Mordatch. Improved contrastive divergence training of energy based models. *arXiv preprint arXiv:2012.01316*, 2020.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. *Physics letters B*, 195(2):216–222, 1987.

Asja Fischer and Christian Igel. Empirical analysis of the divergence of gibbs sampling based learning algorithms for restricted boltzmann machines. In *International conference on artificial neural networks*, pages 208–217. Springer, 2010.

Ruiqi Gao, Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Learning grid-like units with vector representation of self-position and matrix representation of self-motion. *arXiv preprint arXiv:1810.05597*, 2018.

Ruiqi Gao, Erik Nijkamp, Diederik P Kingma, Zhen Xu, Andrew M Dai, and Ying Nian Wu. Flow contrastive estimation of energy-based models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7518–7528, 2020.

Mathieu Germain, Karol Gregor, Iain Murray, and Hugo Larochelle. Made: Masked autoencoder for distribution estimation. In *International Conference on Machine Learning*, pages 881–889, 2015.

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.

Jackson Gorham and Lester Mackey. Measuring sample quality with stein's method. In *Advances in Neural Information Processing Systems*, pages 226–234, 2015.

Will Grathwohl, Jacob Kelly, Milad Hashemi, Mohammad Norouzi, Kevin Swersky, and David Duvenaud. No mcmc for me: Amortized sampling for fast and stable training of energy-based models. *arXiv preprint arXiv:2010.04230*, 2020a.

Will Grathwohl, Kuan-Chieh Wang, Joern-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020b. URL https://openreview.net/forum?id=Hkxzx0NtDB.

Will Grathwohl, Kuan-Chieh Wang, Jorn-Henrik Jacobsen, David Duvenaud, and Richard Zemel. Cutting out the middle-man: Training and evaluating energy-based models without sampling. *arXiv preprint arXiv:2002.05616*, 2020c.

Alex Graves. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*, 2013.

Ulf Grenander and Michael I Miller. Representations of knowledge in complex systems. *Journal of the Royal Statistical Society: Series B (Methodological)*, 56(4):549–581, 1994.

Michael Gutmann and Jun-ichiro Hirayama. Bregman divergence as general framework to estimate unnormalized statistical models. *arXiv preprint arXiv:1202.3727*, 2012.

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.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1352–1361, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870, 2018.

W Keith Hastings. Monte carlo sampling methods using markov chains and their applications. 1970.

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

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *arXiv preprint arXiv:2006.11239*, 2020.

Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.

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

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

Aapo Hyvärinen. Some extensions of score matching. *Computational statistics & data analysis*, 51(5):2499–2512, 2007.

Pierre E Jacob, John O'Leary, and Yves F Atchadé. Unbiased markov chain monte carlo with couplings. *arXiv preprint arXiv:1708.03625*, 2017.

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

Diederik P Kingma and Yann LeCun. Regularized estimation of image statistics by score matching. In *Advances in neural information processing systems*, pages 1126–1134, 2010.

Diederik P Kingma and Max Welling. Auto-encoding variational Bayes. *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.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

Rithesh Kumar, Sherjil Ozair, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. *arXiv preprint arXiv:1901.08508*, 2019.

Qiang Liu, Jason Lee, and Michael Jordan. A kernelized stein discrepancy for goodness-of-fit tests. In *International conference on machine learning*, pages 276–284, 2016.

Siwei Lyu. Unifying non-maximum likelihood learning objectives with minimum kl contraction. In *Advances in Neural Information Processing Systems*, pages 64–72, 2011.

Siwei Lyu. Interpretation and generalization of score matching. *arXiv preprint arXiv:1205.2629*, 2012.

James Martens, Ilya Sutskever, and Kevin Swersky. Estimating the hessian by back-propagating curvature. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 963–970, 2012.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models. In *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pages 419–426, 2012.

Javier R Movellan. A minimum velocity approach to learning. *unpublished draft, Jan*, 2008.

Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of markov chain monte carlo*, 2(11):2, 2011.

Jiquan Ngiam, Zhenghao Chen, Pang W Koh, and Andrew Y Ng. Learning deep energy models. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 1105–1112, 2011.

XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model. In *Advances in Neural Information Processing Systems*, pages 5232–5242, 2019.

Sebastian Nowozin, Botond Cseke, and Ryota Tomioka. f-gan: Training generative neural samplers using variational divergence minimization. In *Advances in neural information processing systems*, pages 271–279, 2016.

Art B. Owen. *Monte Carlo theory, methods and examples*. 2013.

Giorgio Parisi. Correlation functions and computer simulations. *Nuclear Physics B*, 180(3):378–384, 1981.

Matthew Parry, A Philip Dawid, Steffen Lauritzen, et al. Proper local scoring rules. *The Annals of Statistics*, 40(1):561–592, 2012.

Yixuan Qiu, Lingsong Zhang, and Xiao Wang. Unbiased contrastive divergence algorithm for training energy-based latent variable models. In *International Conference on Learning Representations*, 2019.

Martin Raphan and Eero P Simoncelli. Learning to be bayesian without supervision. In *Advances in neural information processing systems*, pages 1145–1152, 2007.

Martin Raphan and Eero P Simoncelli. Least squares estimation without priors or supervision. *Neural computation*, 23(2):374–420, 2011.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International Conference on Machine Learning*, pages 1530–1538, 2015.

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

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE Transactions on Neural Networks*, 20(1):61–80, 2008.

Hannes Schulz, Andreas Müller, and Sven Behnke. Investigating convergence of restricted boltzmann machine learning. In *NIPS 2010 Workshop on Deep Learning and Unsupervised Feature Learning*, volume 1, pages 6–1, 2010.

John Skilling. The eigenvalues of mega-dimensional matrices. In *Maximum Entropy and Bayesian Methods*, pages 455–466. Springer, 1989.

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

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems*, pages 11895–11907, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *arXiv preprint arXiv:2006.09011*, 2020.

Yang Song, Sahaj Garg, Jiaxin Shi, and Stefano Ermon. Sliced score matching: A scalable approach to density and score estimation. In *Proceedings of the Thirty-Fifth Conference on Uncertainty in Artificial Intelligence, UAI 2019, Tel Aviv, Israel, July 22-25, 2019*, page 204, 2019. URL http://auai.org/uai2019/proceedings/papers/204.pdf.

Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020.

Charles M Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981.

Tijmen Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071, 2008.

Aaron Van Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International Conference on Machine Learning*, pages 1747–1756, 2016.

Pascal Vincent. A connection between score matching and denoising autoencoders. *Neural computation*, 23(7):1661–1674, 2011.

Ziyu Wang, Shuyu Cheng, Li Yueru, Jun Zhu, and Bo Zhang. A wasserstein minimum velocity approach to learning unnormalized models. In Silvia Chiappa and Roberto Calandra, editors, *Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics*, volume 108 of *Proceedings of Machine Learning Research*, pages 3728–3738, Online, 26–28 Aug 2020. PMLR. URL http://proceedings.mlr.press/v108/wang20j.html.

Max Welling and Geoffrey E Hinton. A new learning algorithm for mean field boltzmann machines. In *International Conference on Artificial Neural Networks*, pages 351–357. Springer, 2002.

Li Wenliang, Dougal Sutherland, Heiko Strathmann, and Arthur Gretton. Learning deep kernels for exponential family densities. In *International Conference on Machine Learning*, pages 6737–6746, 2019.

Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *International Conference on Machine Learning*, pages 2635–2644, 2016.

Jianwen Xie, Yang Lu, Ruiqi Gao, and Ying Nian Wu. Cooperative learning of energy-based model and latent variable model via mcmc teaching. In *AAAI*, volume 1, page 7, 2018.

Laurent Younes. On the convergence of markovian stochastic algorithms with rapidly decreasing ergodicity rates. *Stochastics: An International Journal of Probability and Stochastic Processes*, 65(3-4):177–228, 1999.

Lantao Yu, Yang Song, Jiaming Song, and Stefano Ermon. Training deep energy-based models with f-divergence minimization. *arXiv preprint arXiv:2003.03463*, 2020.

Shuangfei Zhai, Yu Cheng, Rogerio Feris, and Zhongfei Zhang. Generative adversarial networks as variational training of energy based models. *arXiv preprint arXiv:1611.01799*, 2016.