Learning Unnormalized Statistical Models via Compositional Optimization

Wei Jiang [1] **Jiayu Qin** [2] **Lingyu Wu** [1] **Changyou Chen** [2] **Tianbao Yang** [3] **Lijun Zhang** [1]

Abstract

Learning unnormalized statistical models (e.g., energy-based models) is computationally challenging due to the complexity of handling the partition function. To eschew this complexity, noise-contrastive estimation (NCE) has been proposed by formulating the objective as the logistic loss of the real data and the artificial noise. However, as found in previous works, NCE may perform poorly in many tasks due to its flat loss landscape and slow convergence. In this paper, we study *a direct approach for optimizing the negative log-likelihood* of unnormalized models from the perspective of compositional optimization. To tackle the partition function, a noise distribution is introduced such that the log partition function can be written as a compositional function whose inner function can be estimated with stochastic samples. Hence, the objective can be optimized by stochastic compositional optimization algorithms. Despite being a simple method, we demonstrate that it is more favorable than NCE by (1) establishing a fast convergence rate and quantifying its dependence on the noise distribution through the variance of stochastic estimators; (2) developing better results for one-dimensional Gaussian mean estimation by showing our objective has a much favorable loss landscape and hence our method enjoys faster convergence; (3) demonstrating better performance on multiple applications, including density estimation, out-of-distribution detection, and real image generation.

[1]National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China [2] Department of Computer Science and Engineering, University at Buffalo, New York, USA [3]Department of Computer Science and Engineering, Texas A&M University, College Station, USA. Correspondence to: Changyou Chen <changyou@buffalo.edu>, Tianbao Yang <tianbao-yang@tamu.edu>, Lijun Zhang <zhanglj@lamda.nju.edu.cn>.

Proceedings of the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023.

1. Introduction

We investigate the problem of learning unnormalized statistical models. Suppose we observe a set of training samples $\mathcal{D}_n = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ from an unknown probability density function (pdf) $p_{\text{data}}(\mathbf{x})$ and estimate this data pdf by

$$p(\mathbf{x};\theta) = \frac{p_0(\mathbf{x};\theta)}{\int p_0(\mathbf{x};\theta)d\mathbf{x}}, \tag{1}$$

where $p_0(\mathbf{x};\theta)$ is defined as an unnormalized model, and θ denotes the parameter that will be learnt to best fit the data. The term $\int p_0(\mathbf{x};\theta)d\mathbf{x}$ in equation (1) is called partition function, which is used to ensure the final model is normalized, i.e., $\int p(\mathbf{x};\theta)d\mathbf{x} = 1$. By introducing the partition function, we can use more flexible structures to represent $p_0(\mathbf{x};\theta)$. Specifically, if we set $p_0(\mathbf{x};\theta) = e^{f_0(\mathbf{x};\theta)}$, the above formulation (1) reduces to the well-known energy-based model (EBM) (LeCun et al., 2006):

$$p(\mathbf{x};\theta) = \frac{e^{f_0(\mathbf{x};\theta)}}{\int e^{f_0(\mathbf{x};\theta)}d\mathbf{x}},$$

which enjoys wide applications in machine learning (Du & Mordatch, 2019; Yu et al., 2020; Grathwohl et al., 2020a).

To find the best parameter θ for the given model $p_0(\mathbf{x};\theta)$, we can directly maximize the log-likelihood on the observed training data, i.e., optimizing the objective $\mathcal{L}(\theta)$, where

$$\mathcal{L}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}[\log p_0(\mathbf{x}_i;\theta)] + \log\int p_0(\mathbf{x};\theta)d\mathbf{x}. \tag{2}$$

However, the challenge is that the log partition function $\log\int p_0(\mathbf{x};\theta)d\mathbf{x}$ and its gradient are difficult to calculate exactly. To remedy this issue, prior works (Hinton, 2002; Tieleman, 2008; Nijkamp et al., 2019a) resort to Markov chain Monte Carlo (MCMC) sampling (Christian P. Robert, 2004). Considering the gradient of loss $\mathcal{L}(\theta)$, we have:

$$\nabla\mathcal{L}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left[\frac{\nabla p_0(\mathbf{x}_i;\theta)}{p_0(\mathbf{x}_i;\theta)}\right] + \mathbb{E}_{\mathbf{x}\sim p_\theta}\left[\frac{\nabla p_0(\mathbf{x};\theta)}{p_0(\mathbf{x};\theta)}\right],$$

where p_θ denotes the pdf $p(\mathbf{x},\theta)$. To compute the second term, previous literature applies a number of MCMC steps to sample from p_θ, and then calculates the estimated gradient. However, MCMC sampling is slow and unstable during

training (Grathwohl et al., 2020b; Ruiqi et al., 2020; Geng et al., 2021), partly because approximate samples are obtained with only a finite number of steps. As pointed out by Nijkamp et al. (2019b) and Grathwohl et al. (2020b), estimating with finite MCMC steps will produce biased estimations, leading to optimizing a different objective other than the original MLE loss.

To eschew the complexity of estimating the gradient of log partition function, noise-contrastive estimation (NCE) (Gutmann & Hyvärinen, 2012) has been proposed, which transforms the original problem into a classification task. Specifically, NCE introduces a noise distribution $q(\mathbf{x})$, and then optimizes the extended model parameter $\tau = (\theta, \alpha)$ by minimizing the following loss:

$$\mathcal{J}(\tau) = -\mathbb{E}_{\mathbf{x}\sim p_{\text{data}}} \log h(\mathbf{x}, \tau) - \mathbb{E}_{\mathbf{x}\sim q} \log(1 - h(\mathbf{x}, \tau)),$$

where $h(\mathbf{x}, \tau) = 1/(1 + e^{-(\log p_0(\mathbf{x},\theta) - \log q(\mathbf{x}) - \alpha)})$, and the parameter α is used to estimate the log partition function, i.e., $\alpha = \log \int p_0(\mathbf{x}; \theta) d\mathbf{x}$. This new objective can be interpreted as the logistic loss to distinguish between the noise and the real data. In practice, the first term is estimated by n observed training examples and the second term is evaluated by $m = \nu n$ noise samples where $\nu \geq 1$.

However, as pointed out in many works (Goodfellow et al., 2014; Ruiqi et al., 2020), if the noise distribution q is very different from the data distribution p_{data}, this classification problem would be too easy to solve and the learned model may fail to capture adequate information about the real data distribution. In particular, Liu et al. (2022a) demonstrate that when q and p_{data} are not close enough, the loss $\mathcal{J}(\tau)$ is extremely flat near the optimum, leading to the slow convergence rate of NCE method.

In this paper, we investigate an alternative approach for directly optimizing the MLE objective by converting it to a stochastic compositional optimization (SCO) problem. To deal with the intractable partition function, we introduce a noise distribution $q(\mathbf{x})$, and then convert the log partition function $\log \int p_0(\mathbf{x}; \theta) d\mathbf{x}$ into $\log \mathbb{E}_{\mathbf{x}\sim q} \left[\frac{p_0(\mathbf{x};\theta)}{q(\mathbf{x})} \right]$. Then, the objective function (2) becomes:

$$\mathcal{L}(\theta) = -\frac{1}{n} \sum_{i=1}^{n} [\log p_0(\mathbf{x}; \theta)] + \log \mathbb{E}_{\mathbf{x}\sim q} \left[\frac{p_0(\mathbf{x}; \theta)}{q(\mathbf{x})} \right],$$

which can be written as a two-level SCO problem. Since SCO has been studied extensively, state-of-the-art algorithms can be employed to solve the above problem. However, besides its simplicity, a major question remains: *What are the advantages of this approach compared with NCE and MCMC-based methods for learning unnormalized models?* Our main contributions are to demonstrate the following advantages by theoretical analysis and empirical studies:

1. We prove that our single-loop algorithm converges asymptotically to an optimal solution under the Polyak-Łojasiewicz (PL) condition, and establish a fast convergence rate in the order of $O(1/\epsilon)$ for finding an ϵ-optimal solution. In contrast, NCE and MCMC-based approaches do not provide such guarantees.
2. We prove through one-dimensional Gaussian mean estimation that the MLE objective function has a better loss landscape than the NCE objective, and establish a faster convergence rate of our algorithm than a state-of-the-art NCE-based approach (Liu et al., 2022a) for this task.
3. We illustrate the better performance of our method on different tasks, namely, density estimation, out-of-distribution detection, and real image generation. For the last task, we show that the choice of the noise distribution has a significant impact on the quality of generated data in line with our theoretical analysis.

2. Related Work

This section reviews related work on noise contrastive estimation, and other methods for learning unnormalized models, as well as stochastic compositional optimization.

2.1. Noise-contrastive Estimation

Noise-contrastive estimation (NCE) is first proposed by Gutmann & Hyvärinen (2010; 2012), and gains its popularity in machine learning applications quickly (Mnih & Kavukcuoglu, 2013; Hjelm et al., 2019; Henaff, 2020; Tian et al., 2020; Kong et al., 2020). The basic idea is to introduce a noise distribution, and then distinguish it from the data distribution by a logistic loss. As pointed out in previous works (Goodfellow et al., 2014; Ruiqi et al., 2020; Liu et al., 2022a), the choice of the noise distribution is crucial to the success of NCE, and many methods have been proposed to tune the noise automatically. For example, Bose et al. (2018) utilize a learned adversarial distribution as the noise, and develop a method named Adversarial Contrastive Estimation. At the same time, Conditional NCE is introduced by Ceylan & Gutmann (2018), which generates the noise samples with the help of the observed data. Afterward, Ruiqi et al. (2020) propose Flow Contrastive Estimation, using a flow model to learn the noise distribution by joint training. However, these methods usually introduce extra computation, and the adversarial training of the noise is slow and complex. Instead of designing more complicated noise, Liu et al. (2022a) recently propose a new objective named eNCE, which replaces the log loss in NCE as the exponential loss, and enjoys better results for exponential families. However, eNCE still suffers from the ill-behaved loss landscape, which is extremely flat near the optimum.

2.2. Other Methods for Learning Unnormalized Models

Except NCE and its variants discussed above, there are also many other approaches to solve unnormalized models. To bypass the partition function, score matching (Hyvärinen, 2005) optimizes the squared distance between the gradient of the log density of the model and that of the observed data, in which the partition function would not appear. However, the output dimension of score matching is the same as the input, and training such a model is expensive and unstable, especially for high-dimensional data (Song & Ermon, 2019; Pang et al., 2020). Existing works usually require denoising (Vincent, 2011) or slicing (Song et al., 2019) techniques to train score matching methods.

Besides NCE and score matching, optimizing MLE objective with Markov chain Monte Carlo (MCMC) sampling is also widely used. One well-known method is Contrastive Divergence (CD) introduced by Hinton (2002), which employs MCMC sampling for fixed steps. To sample more efficiently, Tieleman (2008) initializes the MCMC in each training step with the previous sampling results and names this method as persistent CD. Other variants of CD have also been proposed later, such as modified CD (Gao et al., 2018), adversarial CD (Han et al., 2019), etc. These techniques are also very popular in learning energy-based models (EBM), which is a special case of unnormalized models. Specifically, Xie et al. (2016) propose to learn EBM by using ConvNet as the energy function; then, Xie et al. (2018) and Jianwen et al. (2018) use a generator as a fast sampler to help EBM training. Note that the objective of the EBM becomes a modified contrastive divergence, and Xie et al. (2021) and Xie et al. (2022) are proposed as different variants of Jianwen et al. (2018), whose objectives are also a modified contrastive divergence. However, the main drawback of these methods is that MCMC sampling is usually time-consuming and unstable during training (Grathwohl et al., 2020b; Ruiqi et al., 2020; Geng et al., 2021).

2.3. Stochastic Compositional Optimization

Stochastic Compositional Optimization (SCO) has been investigated extensively in the literature. The objective of a two-level SCO is given by $\mathbb{E}_{\xi_1}[f_{\xi_1}(\mathbb{E}_{\xi_2}[g_{\xi_2}(\theta)])]$, where ξ_1 and ξ_2 are random variables. To solve this problem, Wang et al. (2017a) develop stochastic compositional gradient descent (SCGD), which achieves a complexity of $\mathcal{O}\left(\epsilon^{-7}\right)$, $\mathcal{O}\left(\epsilon^{-3.5}\right)$, and $\mathcal{O}\left(\mu^{-14/4}\epsilon^{-5/4}\right)$ for non-convex, convex, and μ-strongly convex functions, respectively. These complexities are further improved to $\mathcal{O}\left(\epsilon^{-4.5}\right)$, $\mathcal{O}\left(\epsilon^{-2}\right)$ and $\mathcal{O}\left(\epsilon^{-1}\right)$ in a subsequent work (Wang et al., 2017b).

To obtain a better rate, Ghadimi et al. (2020) propose a method named averaged stochastic approximation (NASA), which uses momentum-update to estimate the inner function and the gradient, and attains a complexity of $\mathcal{O}\left(\epsilon^{-4}\right)$ for non-convex objectives. Recently, with the popularity of variance reduction methods such as SARAH (Nguyen et al., 2017), SPIDER (Fang et al., 2018) and STORM (Cutkosky & Orabona, 2019), this complexity is further improved to $\mathcal{O}\left(\epsilon^{-3}\right)$, by estimating the inner function and the gradient with variance-reduction techniques (Zhang & Xiao, 2019; Chen et al., 2021; Qi et al., 2021a).

3. The Proposed Method

In this section, we propose to learn unnormalized models directly by Maximum likelihood Estimation via Compositional Optimization (MECO). First, it is easy to show that, with a noise distribution $q(\mathbf{x})$, the MLE objective function (2) is equivalent to:

$$\mathcal{L}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left[\log p_0(\mathbf{x}_i;\theta)\right] + \log \mathbb{E}_{\mathbf{x}\sim q}\left[\frac{p_0(\mathbf{x};\theta)}{q(\mathbf{x})}\right].$$

However, optimizing this objective is nontrivial, because of the nested structure of the second term. Specifically, we can not acquire its unbiased estimation by sampling from $q(\mathbf{x})$, since the expectation cannot be moved out of the log function, i.e., $\mathbb{E}_{\mathbf{x}\sim q}\left[\log \frac{p_0(\mathbf{x};\theta)}{q(\mathbf{x})}\right] \neq \log \mathbb{E}_{\mathbf{x}\sim q}\left[\frac{p_0(\mathbf{x};\theta)}{q(\mathbf{x})}\right]$. Due to similar reasons, we also can not obtain an unbiased estimation of its gradient, which is the main obstacle to deriving an algorithm with convergence guarantees.

To solve this difficulty, we can treat the $\log \mathbb{E}_{\mathbf{x}\sim q}\left[\frac{p_0(\mathbf{x};\theta)}{q(\mathbf{x})}\right]$ as a compositional function, where $\log$ is the outer function and $\mathbb{E}_{\mathbf{x}\sim q}\left[\frac{p_0(\mathbf{x};\theta)}{q(\mathbf{x})}\right]$ is the inner function, which can be unbiased-estimated by sampling form $q(\mathbf{x})$. Inspired by NASA (Ghadimi et al., 2020) for solving stochastic compositional optimization (SCO), we use variance reduced estimators to approximate the inner function and the gradient, so that the estimation errors can be reduced over time. Specifically, considering the gradient of the objective function w.r.t. parameter θ, we have:

$$\nabla\mathcal{L}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\left[\frac{\nabla p_0(\mathbf{x};\theta)}{p_0(\mathbf{x};\theta)}\right] + \frac{\mathbb{E}_{\mathbf{x}\sim q}\left[\frac{\nabla p_0(\mathbf{x};\theta)}{q(\mathbf{x})}\right]}{\mathbb{E}_{\mathbf{x}\sim q}\left[\frac{p_0(\mathbf{x};\theta)}{q(\mathbf{x})}\right]}.$$

To estimate this gradient, in each training step t, we first draw sample $\mathbf{z}_t$ from the training set $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$, and sample $\widetilde{\mathbf{z}}_t$ from the noise distribution $q(\mathbf{x})$. Then, we estimate $\mathbb{E}_{\mathbf{x}\sim q}\left[\frac{p_0(\mathbf{x};\theta)}{q(\mathbf{x})}\right]$ by a function value estimator $\mathbf{u}_t$ in the style of momentum update:

$$\mathbf{u}_t = (1-\gamma_t)\mathbf{u}_{t-1} + \gamma_t \frac{p_0(\widetilde{\mathbf{z}}_t;\theta_t)}{q(\widetilde{\mathbf{z}}_t)}. \tag{3}$$

When evaluating the gradient, we would use $\mathbf{u}_t$ to approximate the term $\mathbb{E}_{\mathbf{x}\sim q}\left[\frac{p_0(\mathbf{x};\theta)}{q(\mathbf{x})}\right]$ in the denominator. Similarly,

Algorithm 1 MECO

Input: time step T, initial points $(\theta_1, \mathbf{u}_1, \mathbf{v}_1)$ sequence $\{\eta_t, \gamma_t, \beta_t\}$
for time step $t = 1$ **to** T **do**
  Sampling $\mathbf{z}_t$ from $\{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ and $\widetilde{\mathbf{z}}_t$ from $q(\mathbf{x})$
  Update estimator $\mathbf{u}_t$ according to equation (3)
  Update estimator $\mathbf{v}_t$ according to equation (4)
  Update the weight: $\theta_{t+1} = \theta_t - \eta_t \mathbf{v}_t$
end for
Choose τ uniformly at random from $\{1, \ldots, T\}$
Return θ_τ

we employ a gradient estimator $\mathbf{v}_t$ to track the overall gradient $\nabla\mathcal{L}(\theta)$ by another momentum update:

$$\begin{aligned}\mathbf{v}_t =&(1-\beta_t)\mathbf{v}_{t-1} \\ &+\beta_t\left(-\frac{\nabla p_0(\mathbf{z}_t;\theta_t)}{p_0(\mathbf{z}_t;\theta_t)}+\frac{1}{\mathbf{u}_t}\frac{\nabla p_0(\widetilde{\mathbf{z}}_t;\theta_t)}{q(\widetilde{\mathbf{z}}_t)}\right).\end{aligned} \tag{4}$$

Although the estimators $\mathbf{u}_t$ and $\mathbf{v}_t$ are still biased, it can be proved that the estimation error is reduced gradually. After obtaining the gradient estimator $\mathbf{v}_t$, we use it to update the parameter θ_t in the style of SGD. The whole algorithm is presented in Algorithm 1. Note that in the first iteration, we can simply set the estimator $\mathbf{u}_1 = \frac{p_0(\widetilde{\mathbf{z}}_1;\theta_1)}{q(\widetilde{\mathbf{z}}_1)}$ and $\mathbf{v}_1 = -\frac{\nabla p_0(\mathbf{z}_1;\theta_1)}{p_0(\mathbf{z}_1;\theta_1)} + \frac{1}{\mathbf{u}_1}\frac{\nabla p_0(\widetilde{\mathbf{z}}_1;\theta_1)}{q(\widetilde{\mathbf{z}}_1)}$.

Difference from NASA: The optimization method we used can be viewed as a modified version of NASA for SCO problems (Ghadimi et al., 2020). Compared with the original NASA applied to constrained SCO, our method does not need to project the variable θ onto the feasible set and is thus simpler. Another **big difference** from NASA lies in the improved rate we will derive for an objective satisfying the PL condition, which is missing in their original work.

Difference from energy-based model (EBM) training: Derived from SCO, our algorithm has a key difference from the standard EBM training. By setting $p_0(\mathbf{x};\theta) = e^{f_0(\mathbf{x};\theta)}$, the unnormalized model converts to the EBM, and our gradient estimator $\mathbf{v}_t$ is written as:

$$(1-\beta_t)\mathbf{v}_{t-1}+\beta_t\left(-\nabla f(\mathbf{z}_t;\theta_t)+\frac{e^{f(\widetilde{\mathbf{z}}_t;\theta_t)}}{q(\widetilde{\mathbf{z}}_t)\mathbf{u}_t}\nabla f(\widetilde{\mathbf{z}}_t;\theta_t)\right).$$

In contrast, EBM usually optimizes the objective function $\mathcal{L}'(\theta) = -\mathbb{E}_{\mathbf{z}\sim p_{\text{data}}}[f(\mathbf{z};\theta)] + \mathbb{E}_{\widetilde{\mathbf{z}}\sim q}[f(\widetilde{\mathbf{z}};\theta)]$, and its stochastic gradient is written as $\nabla\mathcal{L}'(\theta_t) = -\nabla f(\mathbf{z}_t;\theta_t) + \nabla f(\widetilde{\mathbf{z}}_t;\theta_t)$. In this sense, our method can be viewed as introducing an adaptive weight $\frac{e^{f(\widetilde{\mathbf{z}}_t;\theta_t)}}{q(\widetilde{\mathbf{z}}_t)\mathbf{u}_t}$ to the second term and applying SGD with momentum to update the model.

4. Theoretical Analysis

We first present the advantage of the MLE formulation, and then analyze the convergence rate of the proposed method, as well as its relationship with the noise distribution. Due to space limitations, all the proofs are deferred to the appendix.

4.1. Advantages of the MLE Formulation

Since we use MLE to optimize unnormalized models, our objective inherits several nice properties. Here, we analyze the behavior of the estimator $\widehat{\theta} = \arg\min_\theta \mathcal{L}(\theta)$ for large sample sizes and assume that there exists an optimal solution θ^* such that $p(\mathbf{x},\theta^*) = p_{\text{data}}(\mathbf{x})$. To compare different estimators, we introduce the definition of asymptotic relative efficiency (ARE) (Vaart, 1998) below.

Definition 1. *For any two estimators R and S with*

$$\sqrt{n}\left(R-\theta^*\right) \rightsquigarrow N\left(0, r^2\right),\ \sqrt{n}\left(S-\theta^*\right) \rightsquigarrow N\left(0, s^2\right),$$

the ARE of S to R is defined by $ARE(S,R) = r^2/s^2$.

Remark: From the definition, we know that $ARE(S,R) < 1$ indicates estimator R is more efficient than estimator S.

Theorem 1. *According to the property of MLE (Wasserman, 2004), the estimator $\widehat{\theta}$ enjoys the following guarantees:*

1. *(Consistent) The estimator $\widehat{\theta}$ converges in probability to θ^*, i.e., $\widehat{\theta} \xrightarrow{P} \theta^*$.*
2. *(Asymptotically Normal) $\sqrt{n}\left(\widehat{\theta}-\theta^*\right) \rightsquigarrow N(0, \widehat{\text{se}}^2)$, where $\widehat{\text{se}}$ can be computed analytically.*
3. *(Asymptotically Optimal) Denote that $\widetilde{\theta}$ is the output of any other estimator, then $ARE\left(\widetilde{\theta}, \widehat{\theta}\right) \le 1$.*

Remark: The last property implies that the MLE objective has the smallest variance, indicating it is better than optimizing other objectives, e.g., the NCE objective.

4.2. The Convergence of the Proposed Method

Then, we analyze the convergence of Algorithm 1. First, we define the sample complexity to measure the convergence rate, which is widely used in stochastic optimization.

Definition 2. *The sample complexity is the number of samples needed to find a point satisfying $\mathbb{E}[\|\nabla\mathcal{L}(\theta)\|] \le \epsilon$ (ϵ-stationary), or $\mathbb{E}[\mathcal{L}(\theta) - \inf_\theta \mathcal{L}(\theta)] \le \epsilon$ (ϵ-optimal).*

For notation simplicity, we denote $g(\theta) = \mathbb{E}_{\mathbf{x}\sim q}\left[\frac{p_0(\mathbf{x};\theta)}{q(\mathbf{x})}\right]$, $h(\theta) = -\frac{1}{n}\sum_{i=1}^n[\log p_0(\mathbf{x}_i;\theta)]$, $f(\cdot) = \log(\cdot)$, estimator $g(\theta;\widetilde{\mathbf{z}}) = \frac{p_0(\widetilde{\mathbf{z}};\theta)}{q(\widetilde{\mathbf{z}})}$ and $h(\theta;\mathbf{z}) = -\log p_0(\mathbf{z};\theta)$, where $\widetilde{\mathbf{z}}$ and $\mathbf{z}$ are samples drawn from q and $\mathcal{D}_n = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$.

Next, we make the following assumptions, which are commonly used in SCO problems (Wang et al., 2016; Zhang & Xiao, 2019; 2021; Wang & Yang, 2022; Jiang et al., 2022b).

Assumption 1. *We assume that (i) function $f(\cdot)$, $g(\cdot)$ are Lipchitz continuous and smooth with respect to their inputs, $h(\cdot)$ is a smooth function; (ii) $\mathcal{L}$ is lower bounded by $\mathcal{L}^*$.*

Assumption 2. *There exist $\sigma_g, \zeta_g, \zeta_h$ such that*

$$\mathbb{E}_{\widetilde{\mathbf{z}}\sim q(\widetilde{\mathbf{z}})}\left[\|g(\theta;\widetilde{\mathbf{z}}) - g(\theta)\|^2\right] \leq \sigma_g^2,$$
$$\mathbb{E}_{\widetilde{\mathbf{z}}\sim q(\widetilde{\mathbf{z}})}\left[\|\nabla g(\theta;\widetilde{\mathbf{z}}) - \nabla g(\theta)\|^2\right] \leq \zeta_g^2,$$
$$\mathbb{E}_{\mathbf{z}\sim \mathcal{D}_n}\left[\|\nabla h(\theta;\mathbf{z}) - \nabla h(\theta)\|^2\right] \leq \zeta_h^2.$$

Remark: In Assumption 1, $f(\cdot)$ is Lipchitz and smooth in terms of its input when $p_0(\widetilde{\mathbf{z}};\theta)/q(\widetilde{\mathbf{z}}) \geq c$, where c is a positive constant. Assumption 2 assumes that the variances of estimating $g(\theta)$, $\nabla g(\theta)$ and $h(\theta)$ by sampling from the corresponding distributions are bounded.

We first derive an asymptotic convergence result, showing that Algorithm 1 can converge to a stationary point of $\mathcal{L}(\theta)$.

Theorem 2. *Assume that the sequence of stepsizes satisfies $\sum_{t=1}^{\infty}\eta_t = +\infty, \sum_{t=1}^{\infty}\eta_t^2 < \infty$. Then with probability 1, accumulation point $(\theta', \mathbf{u}', \mathbf{v}')$ of the sequence $\{\theta_t, \mathbf{u}_t, \mathbf{v}_t\}$ generated by Algorithm 1 satisfies:*

$$\nabla\mathcal{L}(\theta') = 0, \quad u' = g(\theta'), \quad \mathbf{v}' = \nabla\mathcal{L}(\theta').$$

Next, we present the non-asymptotic convergence result.

Theorem 3. *For non-convex function $\mathcal{L}(\cdot)$, by setting $\eta_t = \mathcal{O}(\epsilon^2)$, $\beta_t = \gamma_t = \mathcal{O}(\epsilon^2)$, Algorithm 1 finds an ϵ-stationary point in $T = \mathcal{O}\left(\max\left\{\frac{1}{\epsilon^2}, \frac{(\sigma_g^2+\zeta_g^2+\zeta_h^2)}{\epsilon^4}\right\}\right)$ iterations.*

Remark: The analysis of above two theorems closely follows Ghadimi et al. (2020).

The main contribution of our analysis is to show that by properly setting the hyper-parameters, Algorithm 1 can enjoy a faster convergence rate when the objective satisfies the Polyak-Łojasiewicz (PL) condition (Karimi et al., 2016). We give the definition of the PL condition below.

Definition 3. *$\mathcal{L}(\theta)$ satisfies the μ-PL condition if there exists $\mu > 0$ such that $2\mu\left(\mathcal{L}(\theta) - \mathcal{L}^*\right) \leq \|\nabla\mathcal{L}(\theta)\|^2$.*

Remark: PL condition has been shown to be satisfied for deep learning under over-parameterized networks by many prior works (Allen-Zhu et al., 2019; Du et al., 2019). Under the PL condition, a stationary point θ' becomes a global optimal solution of objective $\mathcal{L}$. Thus, our algorithm asymptotically converges to an optimal solution $\widehat{\theta} = \arg\min\mathcal{L}(\theta)$ according to Theorem 2. Next, we show the improved complexity under the PL condition as stated below.

Theorem 4. *When the objective satisfies μ-PL condition, by setting $\gamma_{t+1} = \beta_{t+1} = \mathcal{O}\left(\max\{1,\mu\}\eta_t\right)$ and $1 - \mu\eta_t = \eta_t^2/\eta_{t-1}^2$, Algorithm 1 can find an ϵ-optimal solution in $T = \mathcal{O}\left(\max\left\{\frac{1}{\mu\sqrt{\epsilon}}, \frac{(\sigma_g^2+\zeta_g^2+\zeta_h^2)}{\mu\epsilon}, \frac{(\sigma_g^2+\zeta_g^2+\zeta_h^2)}{\mu^2\epsilon}\right\}\right)$ iterations.*

Remark: We emphasize that the above convergence for a single-loop algorithm is novel in SCO. Existing methods for SCO usually have to employ two-loop stagewise methods to obtain similar results under the PL condition (Zhang & Xiao, 2019; Qi et al., 2021b; Jiang et al., 2022b).

4.3. Choosing the Noise Distribution

Similar to NCE, our approach also relies on a noise distribution $q(\widetilde{\mathbf{z}})$. The difference is that the noise distribution only affects the convergence rate of our method, not the optimal solution. Theorem 2 shows that our algorithm will eventually converge to a stationary point or a global optimal solution (under PL condition) of the MLE objective, as long as $q(\widetilde{\mathbf{z}})$ is positive whenever $p_0(\widetilde{\mathbf{z}};\theta)$ is positive, no matter what the noise distribution is. On the other hand, the noise distribution would affect the objective function of NCE, and it is pointed out that if the noise distribution is too different from data distribution, the classification problem becomes too easy, which prevents the model from learning much about the data structure (Gutmann & Hyvärinen, 2012). For our method, the impact of $q(\widetilde{\mathbf{z}})$ on the convergence rate is through the variance of $g(\theta;\widetilde{\mathbf{z}})$ and $\nabla g(\theta;\widetilde{\mathbf{z}})$. From our theoretical results (Theorems 3 and 4), we can see that if the variance is zero, then the noise distribution has no impact on the convergence rate. We show the condition to satisfy the zero variance below.

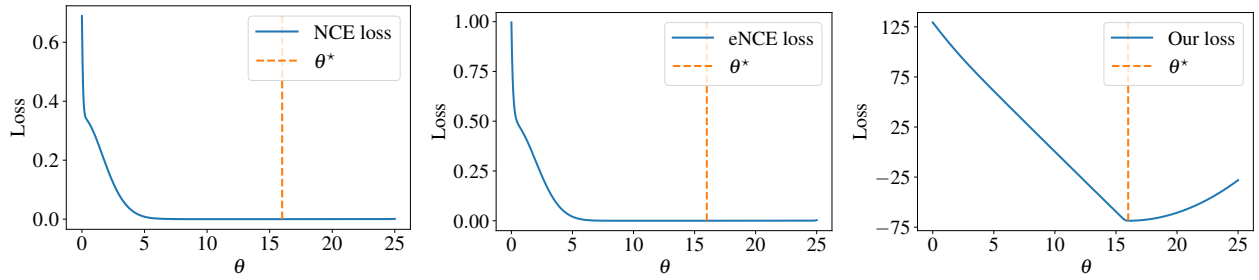
Lemma 1. *If $q(\widetilde{\mathbf{z}}) = p(\widetilde{\mathbf{z}};\theta)$, then $\sigma_g^2 = 0$ and $\zeta_g^2 = 0$.*

Remark: However, the above choice is impractical as sampling from $p(\widetilde{\mathbf{z}};\theta)$ is not easy, and the dependence of $q(\widetilde{\mathbf{z}})$ on θ would also make our convergence analysis fail. In practice, we can only hope $q(\widetilde{\mathbf{z}})$ is close to $p(\widetilde{\mathbf{z}};\theta)$.

Also, to ensure the partition function $\int p_0(\widetilde{\mathbf{z}};\theta)d\widetilde{\mathbf{z}}$ can be written as $\mathbb{E}_{\widetilde{\mathbf{z}}\sim q}\left[\frac{p_0(\widetilde{\mathbf{z}};\theta)}{q(\widetilde{\mathbf{z}})}\right]$, we should guarantee $q(\widetilde{\mathbf{z}}) > 0$ whenever $p_0(\widetilde{\mathbf{z}};\theta) > 0$. This is also required in NCE, which is not difficult to satisfy, since many continuous probability distributions can ensure their probability density functions are always positive, e.g., Gaussian, Laplace and Cauchy distributions. Thus, our analysis suggests the following:

1. Choose $q(\widetilde{\mathbf{z}})$ that can be easily sampled and computed.
2. We should ensure $q(\widetilde{\mathbf{z}}) > 0$ whenever $p_0(\widetilde{\mathbf{z}};\theta) > 0$.
3. The noise distribution should be similar to $p(\widetilde{\mathbf{z}};\theta)$ or the data distribution $p_{\text{data}}(\widetilde{\mathbf{z}})$.

To satisfy the last two properties, we can sample the real data and add some noise to the data. In particular, to sample $\widetilde{\mathbf{z}} \sim q(\widetilde{\mathbf{z}})$, we first sample $\mathbf{x}' \sim \mathcal{D}_n$, $\mathbf{z}' \sim \mathcal{N}(\mu, \Sigma)$, and then set $\widetilde{\mathbf{z}} = \mathbf{x}' + \mathbf{z}'$. Since $\mathcal{D}_n = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$ and denote the probability density function of $\mathcal{N}(\cdot;\mu,\Sigma)$ by $\kappa(\cdot)$, the noise distribution would be $q(\widetilde{\mathbf{z}}) = \frac{1}{n}\sum_{i=1}^n \kappa(\mathbf{x} - \mathbf{x}_i)$, which can be approximated within the mini-batch in practice. To ensure that the noise is similar to the data, we can also fit the parameter μ and Σ by some training data, which can



(a) The NCE objective (b) The eNCE objective (c) The MLE objective (ours)

Figure 1. The loss landscape of three objectives. Note that the optimal parameter $\theta^* = 16$.

be easily done by using the existing numpy package via numpy.mean() and numpy.cov(). In our empirical study, this is helpful for high-dimensional problems, i.e., image generation on MNIST data set in Section 6.3. For simple problems, such as density estimation and out-of-distribution detection, we can simply set the noise as a multivariate Gaussian distribution, which is very fast and easy to sample from. Since more complex noise would introduce more computations, it is a trade-off in practice.

Finally, we would like to emphasize that the impact of the noise distribution can be alleviated by increasing the mini-batch size for estimating $g(\theta)$ and $\nabla g(\theta)$. If the mini-batch size of noisy samples is B, then the variance σ_g^2 and ζ_g^2 will be scaled by B. Hence, the larger the batch size, the less impact of the noise distribution on the convergence rate.

5. Case Study: Gaussian Mean Estimation

As pointed out by Liu et al. (2022a), the reason that NCE may fail to learn a good parameter is due to its flat loss landscape. To verify this claim, they use one-dimensional Gaussian mean estimation as an example and show the slow convergence of NCE for this simple task. To demonstrate the effectiveness of our method, we use the same task to show the behavior of our loss and the proposed optimization method. Following their setup, the data and noise distributions are Gaussian distributions with mean θ^* and θ_q separately, and the variances are both fixed as 1, i.e.,

$$p_{\text{data}}(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\theta^*)^2}{2}}, \quad q(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\theta_q)^2}{2}}.$$

Then, assume that the model is another Gaussian distribution with mean θ and variance 1, which is equivalent to setting $p_0(x;\theta) = e^{\theta x - \frac{1}{2}x^2}$. The goal is to learn the parameter θ, or $\tau(\theta) = (\theta, \frac{\theta^2}{2} + \log\sqrt{2\pi})$ for NCE. First, we can see the flatness of NCE loss via the proposition below.

Proposition 1. *(Proposition 4.2 in Liu et al. (2022a)) Denote that $R = |\theta^* - \theta_q|$ and $\mathcal{J}(\cdot)$ is the NCE loss. Then, we have $\mathcal{J}(\tau) - \mathcal{J}(\tau^*) \leq R\exp(-R^2/8)\|\tau - \tau^*\|^2$, where $\tau^* = \tau(\theta^*)$ is the optimal solution.*

Remark: If θ^* and θ_q are not close enough, the difference between $J(\tau)$ and $J(\tau^*)$ will be extremely small when τ approaches τ^*, implying a flat landscape.

However, it is not a problem for the MLE objective $\mathcal{L}(\cdot)$ that we optimize, which is shown in the following proposition.

Proposition 2. *For 1-d Gaussian mean estimation, the MLE objective satisfies that $\mathcal{L}(\theta) - \mathcal{L}(\theta^*) = \frac{1}{2}\|\theta - \theta^*\|^2$.*

Remark: Compared with NCE, the MLE objective does not have the extremely small factor $R\exp(-R^2/8)$, and thus has a much better loss landscape near the optimum.

To show this difference more vividly, we plot the loss landscape of the NCE objective and the MLE objective, as well as the newly proposed eNCE objective (Liu et al., 2022a). Following the same setup as Liu et al. (2022a), we set $\theta_q = 0$ and $\theta^* = 16$. The results are shown in Figure 1. As can be seen, both the NCE and eNCE objectives are very flat near the optimal solution, while the MLE objective is very sharp.

Besides, we provide the convergence rate of our method for this task, as stated below.

Proposition 3. *For 1-d Gaussian mean estimation, Algorithm 1 ensures $\mathcal{L}(\theta) - \mathcal{L}^* \leq \epsilon$ after $T = \mathcal{O}(\epsilon^{-1})$ iterations.*

Remark: Although Liu et al. (2022a) can ensure that $\|\tau - \tau^*\| \leq \epsilon$ after $\mathcal{O}(\epsilon^{-2})$ iterations, by using normalized gradient descent (NGD) for the NCE or eNCE objective, their assumptions are very strong. They assume the algorithm can obtain the **exact** gradient, which is impossible in practice. In contrast, our analysis only relies on stochastic gradients. Also note that NCE optimized with standard gradient descent requires an exponential number of steps to find a reasonable parameter, according to Liu et al. (2022a).

Finally, we conduct experiments to compare different methods. We choose mean square error (MSE) $\|\theta - \theta^*\|^2$ as the criterion and compare our method with NCE trained by SGD and NGD, MCMC training (Du & Mordatch, 2019), and the eNCE objective trained by NGD. The results are presented in Figure 2, which demonstrate that our method converges more quickly than other methods, and NCE is the

Table 1. Running time of each method.

NCE	NCE (NGD)	eNCE (NGD)	MCMC	Ours
176s	181s	169s	1757s	168s

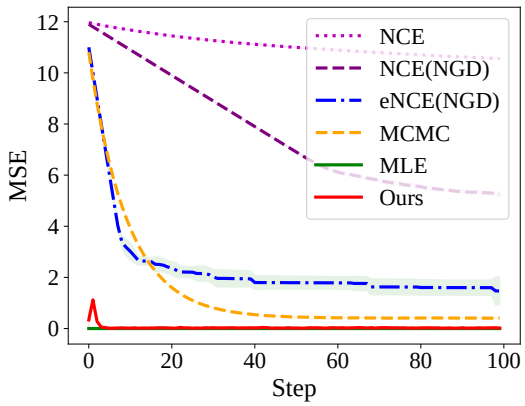


Figure 2. Results for 1-d Gaussian mean estimation.

slowest due to its flat loss landscape. Since MLE can be calculated for Gaussian distribution directly, we also include a curve for MLE as a reference, which is very close to the curve of our method after a few steps. Note that we run each method for 100 steps, and we report the running time of each method in Table 1, It can be seen that the running time of NCE and our method are very similar, while the MCMC method is much slower. As a result, it is fair to compare the convergence between NCE and our method.

6. Experiments

In this section, we conduct experiments on three different tasks, and compare our method with NCE, MCMC training, NCE and eNCE trained by NGD, etc. For our method, we set the parameter $\gamma = 0.1$ and $\beta = 0.9$. For MCMC training, the number of sampling steps is searched from the set $\{20, 50, 100\}$ and we use Langevin dynamics (Welling & Teh, 2011) as the sampling approach. For all tasks, we tune the learning rates from $\{1e{-}1, 1e{-}2, 1e{-}3, 1e{-}4\}$ and pick the best one. In this section, all methods are trained with the same training time. Experiments in Section 6.1 and 6.2 are conducted on a personal laptop, and the training time for each method is around 10 minutes and 72 minutes, respectively. Experiments on MNIST in Section 6.3 are trained on four NVIDIA Tesla V100 GPUs, and the training time is around 2.8 hours.

6.1. Density Estimation on Synthetic Data

First, we focus on density estimation on synthetic data. Following the experimental setup of the previous literature (Grathwohl et al., 2020b; Liu et al., 2022b), we sample a set of 2D data points as the training set, according to some data distribution $p_{\text{data}}(\mathbf{x})$, visualized in the top of Figure 3. Then we train unnormalized models $p_0(\mathbf{x};\theta) = e^{f_0(\mathbf{x};\theta)}$ to learn this distribution, where $f_0(\mathbf{x};\theta)$ is the multi-layer perceptrons (MLPs) with 3 hidden layers and 300 units per layer. In the experiment, We choose the SGD (or SGD-style) optimizer. For NCE, eNCE and our method, the noise distribution is selected as a multivariate Gaussian distribution, whose mean and variance are fitted on the training set.

To quantify the performance of different methods, we adopt the maximum mean discrepancy (MMD) (Gretton et al., 2012) as the criterion. The MMD metric is widely used to compare different distributions, and a lower MMD indicates that the two distributions are more similar. We sample 10000 points from the data distribution and the learned model, and the computed MMD metric is shown in Table 2. As can be seen, our method enjoys the lowest MMD among all methods for all six cases, indicating the superiority of the proposed method. Also, we report the Frechet Inception Distance (FID) score of each method, which is another widely-used measure in comparing distributions. The results are presented in Table 3, and our method enjoys better FID scores than other algorithms (smaller is better). Besides, we compare the estimated density and the ground-truth in Figure 3, showing that our method learns the density accurately in most cases. Finally, We compare different methods with the Adam optimizer and investigate the behavior of our algorithm with different values of γ and β in the appendix.

6.2. Out-of-distribution Detection

Then, we experiment on the out-of-distribution (OOD) detection, since OOD detection performance is an important measure of the density estimation quality. For this task, we choose CIFAR-10 (Krizhevsky, 2009) as the in-distribution data. We use the energy-based model as our unnormalized model, by setting $p_0(\mathbf{x};\theta) = e^{f_0(\mathbf{x};\theta)}$, where $f_0(\mathbf{x};\theta)$ is a 40-layer WideResNet (Zagoruyko & Komodakis, 2016). The noise distribution for NCE, eNCE and our method is selected as the multivariate Gaussian distribution, and we use Adam to optimize. For the OOD test dataset, we use four common benchmarks: CIFAR-10 Interp, SVHN (Netzer et al., 2011), CIFAR-100 (Krizhevsky, 2009), and LSUN (Yu et al., 2015). We decide whether a test sample $\mathbf{x}$ is anomalous or not by computing the density $p_0(\mathbf{x};\theta)$, where a higher value indicates the test sample is more likely to be a normal sample.

To measure the performance of different methods, we follow previous works (Ren et al., 2019; Havtorn et al., 2021; Geng et al., 2021) to choose three metrics to compare: (1) area under the receiver operating characteristic curve (AUROC↑); (2) area under the precision-recall curve (AUPRC↑); and (3) false positive rate at 80% true positive rate (FPR80↓),

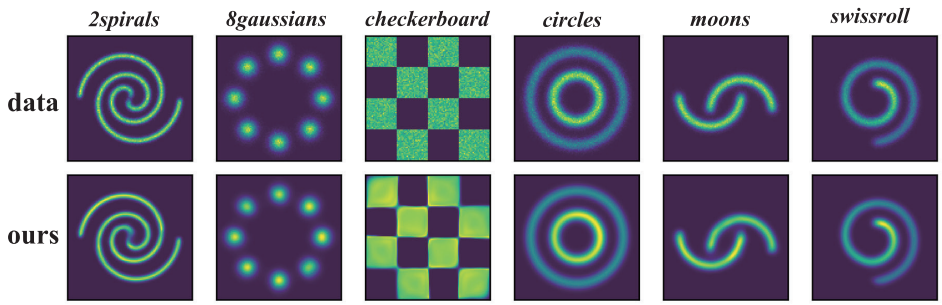


Figure 3. Visualization of density on synthetic datasets.

Table 2. Results on synthetic data in terms of MMD (lower is better).

Method	*2spirals*	*8gaussians*	*checkerboard*	*circles*	*moons*	*swissroll*
NCE	3.253 ± 0.284	0.153 ± 0.095	1.956 ± 0.469	1.223 ± 0.154	5.178 ± 0.341	2.715 ± 0.249
NCE (NGD)	3.445 ± 0.287	0.177 ± 0.102	1.963 ± 0.488	1.270 ± 0.292	5.010 ± 0.349	2.585 ± 0.201
eNCE (NGD)	3.328 ± 0.332	0.257 ± 0.144	1.810 ± 0.218	1.183± 0.188	4.728 ± 0.399	2.975 ± 0.398
MCMC	3.060 ± 0.780	0.150 ± 0.035	1.654 ± 0.217	1.154 ± 0.294	4.722 ± 0.633	2.764 ± 0.670
Score Matching	3.268 ± 0.846	0.250 ± 0.076	2.167 ± 0.703	1.302 ± 0.272	4.826 ± 0.153	2.660 ± 0.513
Contrastive Divergence	3.245 ± 0.426	0.182 ± 0.085	1.987 ± 0.470	1.161 ± 0.410	4.716 ± 0.658	2.623 ± 0.203
Ours	**3.040 ± 0.199**	**0.132 ± 0.098**	**1.645 ± 0.251**	**1.075 ± 0.169**	**4.673 ± 0.519**	**2.566 ± 0.274**

Table 3. Results on synthetic data in terms of FID (lower is better).

Method	*2spirals*	*8gaussians*	*checkerboard*	*circles*	*moons*	*swissroll*
NCE	0.103 ± 0.038	0.118 ± 0.026	0.178 ± 0.026	0.096 ± 0.018	0.114± 0.020	0.181 ± 0.041
NCE (NGD)	0.078 ± 0.024	0.143 ± 0.048	0.169 ± 0.023	0.103 ± 0.024	0.099 ± 0.029	0.171 ± 0.023
eNCE (NGD)	0.083 ± 0.040	0.128 ± 0.033	0.112 ± 0.029	0.085 ± 0.045	0.096 ± 0.015	0.212 ± 0.040
MCMC	0.072 ± 0.052	0.115 ± 0.038	0.125 ± 0.036	0.075 ± 0.042	0.109 ± 0.042	0.174 ± 0.028
Score Matching	0.109 ± 0.044	0.178 ± 0.086	0.166 ± 0.069	0.099 ± 0.026	0.117 ± 0.024	0.178 ± 0.062
Contrastive Divergence	0.089 ± 0.037	0.142 ± 0.051	0.121 ± 0.029	0.105 ± 0.039	0.110 ± 0.017	0.180 ± 0.025
Ours	**0.061 ± 0.032**	**0.104 ± 0.025**	**0.111 ± 0.024**	**0.065 ± 0.030**	**0.094 ± 0.131**	**0.163 ± 0.039**

where the arrow indicates the direction of improvement of the metrics. These three metrics are commonly used for evaluating OOD detection methods, and the results are reported in Table 4. As can be seen, our method performs the best under three different criteria in most cases, implying the effectiveness of the proposed method.

6.3. Learning on Real Image Dataset

Finally, we test our method on two real image datasets, MNIST (LeCun et al., 1998) and CIFAR-10 (Krizhevsky, 2009). We describe the setup and the results of MNIST in this subsection, and results of CIFAR-10 can be found in Appendix A. For MNIST task, following the same setup in TRE (Rhodes et al., 2020), the model takes the form of $p_0(\mathbf{x};\theta) = e^{f_0(\mathbf{x};\theta)}$, where $f_0(\mathbf{x};\theta)$ is the 18-layer ResNet (He et al., 2016). For NCE, eNCE and our method, we try three different noise distributions: 1) the empirical distribution $\mathcal{D}_n = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$; 2) multivariate Gaussian distribution, with mean and covariance fitted by the training data; 3) mixture of distribution $\mathcal{D}_n$ and a multivariate Gaussian distribution. We find the last one is the best choice for all methods. We use Adam to optimize NCE, and generate new samples via MCMC sampling after training the model. The generated samples of our method with different noises are shown in the Appendix A, and the result with the third noise distribution is presented in Figure 4.

We also evaluate the learned model via estimated average negative log-likelihood (bits per dimension) on the testing

Table 4. OOD detection results (AUROC↑, AUPRC↑ and FRP80↓) for models trained on CIFAR-10.

Dataset		AUROC↑	AUPRC↑	FRP80↓
CIFAR10-Interp	NCE	0.6468 ± 0.0122	0.5455 ± 0.0053	0.4929 ± 0.0301
	NCE (NGD)	0.6748 ± 0.0187	0.6265 ± 0.1565	0.5392 ± 0.0271
	eNCE (NGD)	0.7451 ± 0.0296	0.7052 ± 0.0310	0.4052 ± 0.0586
	MCMC	0.7077 ± 0.0154	0.6787 ± 0.0119	0.5137 ± 0.0253
	Scoring Matching	0.5329 ± 0.0024	0.5304 ± 0.0029	0.7706 ± 0.0047
	Contrastive Divergence	0.7735 ± 0.0161	0.7371 ± 0.0106	0.3733 ± 0.0382
	Ours	**0.8019 ± 0.0323**	**0.7679 ± 0.0402**	**0.3185 ± 0.0621**
SVHN	NCE	0.6425 ± 0.0107	0.3436 ± 0.0052	0.5592 ± 0.0286
	NCE (NGD)	0.6593 ± 0.0035	0.4361 ± 0.0083	0.7284 ± 0.0040
	eNCE (NGD)	0.6612 ± 0.0075	0.4633 ± 0.0109	0.6910 ± 0.0134
	MCMC	0.5359 ± 0.0078	0.2674 ± 0.0048	0.6327 ± 0.0037
	Scoring Matching	0.5601 ± 0.0068	0.3237 ± 0.0010	0.8346 ± 0.0060
	Contrastive Divergence	0.6103 ± 0.0040	0.3057 ± 0.0018	0.3758 ± 0.0025
	Ours	**0.7843 ± 0.0057**	**0.5134 ± 0.0076**	**0.3255 ± 0.2081**
CIFAR-100	NCE	0.5323 ± 0.0199	0.5129 ± 0.0095	0.7122 ± 0.0216
	NCE (NGD)	0.5513 ± 0.0218	0.5458 ± 0.0160	0.7832 ± 0.0306
	eNCE (NGD)	0.5034 ± 0.0069	0.5248 ± 0.0158	0.8263 ± 0.0095
	MCMC	0.5669 ± 0.0059	0.5604 ± 0.0138	0.7217 ± 0.0052
	Scoring Matching	0.5732 ± 0.0014	0.5343 ± 0.0070	0.6943 ± 0.0059
	Contrastive Divergence	0.5276 ± 0.0098	0.5136 ± 0.0120	0.6642 ± 0.0111
	Ours	**0.6044 ± 0.0049**	**0.6262 ± 0.0079**	**0.5795 ± 0.0113**
LSUN-C	NCE	0.5356 ± 0.0006	0.4995 ± 0.0005	0.6876 ± 0.0028
	NCE (NGD)	0.6049 ± 0.0189	0.5817 ± 0.0083	0.7119 ± 0.0408
	eNCE (NGD)	0.5470 ± 0.0039	**0.6840 ± 0.0042**	0.9450 ± 0.0028
	MCMC	0.5359 ± 0.0153	0.5107 ± 0.0157	0.7436 ± 0.0131
	Scoring Matching	0.5419 ± 0.0057	0.5221 ± 0.0030	0.7372 ± 0.0097
	Contrastive Divergence	0.5044 ± 0.0054	0.4980 ± 0.0060	0.6248 ± 0.0022
	Ours	**0.6944 ± 0.0061**	0.6267 ± 0.0046	**0.5198 ± 0.0126**

Table 5. Average negative log-likelihood (smaller is better).

NCE	NCE (NGD)	eNCE (NGD)	MCMC-OT	CF-EBM	CoopNets	Ours
1.457 ± 0.003	1.618 ± 0.010	1.600 ± 0.044	1.441 ± 0.012	1.864 ± 0.004	1.591 ± 0.022	**1.394 ± 0.007**

sets in Table 5, which is computed by the annealed importance sampling (AIS) (Nash & Durkan, 2019). We also compare our method with some other methods, including MCMC-OT (An et al., 2021), CF-EBM (Zhao et al., 2021), and CoopNets (Xie et al., 2018). As can be seen, our method attains a better likelihood than other methods, indicating the proposed method performs better in this task.

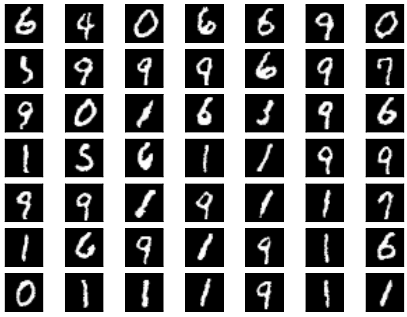


Figure 4. Generated digits using our model via MCMC sampling.

7. Conclusion

In this paper, we investigate the problem of learning unnormalized models by maximum likelihood estimation. By introducing a noise distribution, we cast the problem as compositional optimization and utilize a stochastic algorithm to solve it. We provide the convergence rate of our method and analyze its relationship with the noise distribution. Besides, we use one-dimensional Gaussian mean estimation as an example to show the better loss landscape of our loss compared with the NCE loss, and the fast convergence of the proposed method. Finally, experiments on practical problems demonstrate the effectiveness of the proposed method.

8. Acknowledgments

W. Jiang, L. Wu and L. Zhang are partially supported by the National Key R&D Program of China (2021ZD0112802), NSFC (62122037, 61921006), and the Fundamental Research Funds for the Central Universities (2023300246).

References

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 242–252, 2019.

An, D., Xie, J., and Li, P. Learning deep latent variable models by short-run mcmc inference with optimal transport correction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15410–15419, 2021.

Bose, A. J., Ling, H., and Cao, Y. Adversarial contrastive estimation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 1021–1032, 2018.

Ceylan, C. and Gutmann, M. U. Conditional noise-contrastive estimation of unnormalised models. In *Proceedings of the 35th International Conference on Machine Learning*, pp. 726–734, 2018.

Chen, T., Sun, Y., and Yin, W. Solving stochastic compositional optimization is nearly as easy as solving stochastic optimization. *IEEE Transactions on Signal Processing*, 69:4937–4948, 2021.

Christian P. Robert, G. C. *Monte Carlo Statistical Methods*. Springer, 2004.

Cutkosky, A. and Orabona, F. Momentum-based variance reduction in non-convex SGD. In *Advances in Neural Information Processing Systems 32*, pp. 15210–15219, 2019.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

Du, Y. and Mordatch, I. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems 32*, pp. 3603–3613, 2019.

Fang, C., Li, C. J., Lin, Z., and Zhang, T. Spider: Near-optimal non-convex optimization via stochastic path integrated differential estimator. *ArXiv e-prints*, arXiv:1807.01695, 2018.

Gao, R., Lu, Y., Zhou, J., Zhu, S.-C., and Wu, Y. N. Learning generative convnets via multi-grid modeling and sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9155–9164, 2018.

Geng, C., Wang, J., Gao, Z., Frellsen, J., and Hauberg, S. Bounds all around: training energy-based models with bidirectional bounds. In *Advances in Neural Information Processing Systems 34*, pp. 19808–19821, 2021.

Ghadimi, S., Ruszczynski, A., and Wang, M. A single timescale stochastic approximation method for nested stochastic optimization. *SIAM Journal on Optimization*, 30(1):960–979, 2020.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, pp. 2672–2680, 2014.

Grathwohl, W. Joint energy models. https://github.com/wgrathwohl/JEM, 2020.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., Norouzi, M., and Swersky, K. Your classifier is secretly an energy based model and you should treat it like one. In *International Conference on Learning Representations*, 2020a.

Grathwohl, W., Wang, K.-C., Jacobsen, J.-H., Duvenaud, D., and Zemel, R. Learning the stein discrepancy for training and evaluating energy-based models without sampling. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 3732–3747, 2020b.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *Journal of Machine Learning Research*, 13(25):723–773, 2012.

Guo, Z., Xu, Y., Yin, W., Jin, R., and Yang, T. On stochastic moving-average estimators for non-convex optimization. *ArXiv e-prints*, arXiv:2104.14840, 2021.

Gutmann, M. U. and Hyvärinen, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In *Proceedings of the 13th International Conference on Artificial Intelligence and Statistics*, pp. 297–304, 2010.

Gutmann, M. U. and Hyvärinen, A. Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics. *Journal of Machine Learning Research*, 13(11):307–361, 2012.

Han, T., Nijkamp, E., Fang, X., Hill, M., Zhu, S., and Wu, Y. N. Divergence triangle for joint training of generator model, energy-based model, and inferential model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8670–8679, 2019.

Havtorn, J. D., Frellsen, J., Hauberg, S., and Maaløe, L. Hierarchical VAEs know what they don't know. In *Proceedings of the 38th International Conference on Machine Learning*, pp. 4117–4128, 2021.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

Henaff, O. Data-efficient image recognition with contrastive predictive coding. In *Proceedings of the 37th International Conference on Machine Learning*, pp. 4182–4192, 2020.

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

Hjelm, R. D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., and Bengio, Y. Learning deep representations by mutual information estimation and maximization. In *International Conference on Learning Representations*, 2019.

Hyvärinen, A. Estimation of non-normalized statistical models by score matching. *Journal of Machine Learning Research*, 6(24):695–709, 2005.

Jiang, W., Li, G., Wang, Y., Zhang, L., and Yang, T. Multi-block-single-probe variance reduced estimator for coupled compositional optimization. In *Advances in Neural Information Processing Systems 35*, 2022a.

Jiang, W., Wang, B., Wang, Y., Zhang, L., and Yang, T. Optimal algorithms for stochastic multi-level compositional optimization. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 10195–10216, 2022b.

Jianwen, X., Lu, Y., Zhu, S., and Wu, Y. Cooperative training of descriptor and generator networks. *IEEE transactions on pattern analysis and machine intelligence*, 2018.

Karimi, H., Nutini, J., and Schmidt, M. Linear convergence of gradient and proximal-gradient methods under the Polyak-Łojasiewicz condition. In *Machine Learning and Knowledge Discovery in Databases*, pp. 795–811, 2016.

Kong, L., de Masson d'Autume, C., Yu, L., Ling, W., Dai, Z., and Yogatama, D. A mutual information maximization perspective of language representation learning. In *International Conference on Learning Representations*, 2020.

Krizhevsky, A. Learning multiple layers of features from tiny images. *Masters Thesis, Deptartment of Computer Science, University of Toronto*, 2009.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pp. 2278–2324, 1998.

LeCun, Y., Chopra, S., Hadsell, R., Ranzato, A., and Huang, F. J. A tutorial on energy-based learning. *Predicting structured data*, 2006.

Liu, B., Rosenfeld, E., Ravikumar, P. K., and Risteski, A. Analyzing and improving the optimization landscape of noise-contrastive estimation. In *International Conference on Learning Representations*, 2022a.

Liu, M., Liu, H., and Ji, S. Gradient-guided importance sampling for learning binary energy-based models. *ArXiv e-prints*, arXiv:2210.05782, 2022b.

Mnih, A. and Kavukcuoglu, K. Learning word embeddings efficiently with noise-contrastive estimation. In *Advances in Neural Information Processing Systems 26*, pp. 2265–2273, 2013.

Nash, C. and Durkan, C. Autoregressive energy machines. In *Proceedings of the 36th International Conference on Machine Learning*, pp. 1735–1744, 2019.

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., and Ng, A. Y. Reading digits in natural images with unsupervised feature learning. In *Advances in Neural Information Processing Systems Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Nguyen, L. M., Liu, J., Scheinberg, K., and Takac, M. SARAH: A novel method for machine learning problems using stochastic recursive gradient. In *Proceedings of the 34th International Conference on Machine Learning*, pp. 2613–2621, 2017.

Nijkamp, E., Hill, M., Han, T., Zhu, S.-C., and Wu, Y. N. On the anatomy of MCMC-based maximum likelihood learning of energy-based models. In *Proceedings of the 33th AAAI Conference on Artificial Intelligence*, 2019a.

Nijkamp, E., Hill, M., Zhu, S.-C., and Wu, Y. N. Learning non-convergent non-persistent short-run MCMC toward energy-based model. In *Advances in Neural Information Processing Systems 32*, pp. 5233–5243, 2019b.

Pang, T., Xu, K., LI, C., Song, Y., Ermon, S., and Zhu, J. Efficient learning of generative models via finite-difference score matching. In *Advances in Neural Information Processing Systems 33*, pp. 19175–19188, 2020.

Qi, Q., Guo, Z., Xu, Y., Jin, R., and Yang, T. An online method for a class of distributionally robust optimization with non-convex objectives. *ArXiv e-prints*, arXiv:2006.10138, 2021a.

Qi, Q., Luo, Y., Xu, Z., Ji, S., and Yang, T. Stochastic optimization of areas under precision-recall curves with provable convergence. In *Advances in Neural Information Processing Systems 34*, pp. 1752–1765, 2021b.

Ren, J., Liu, P. J., Fertig, E., Snoek, J., Poplin, R., Depristo, M., Dillon, J., and Lakshminarayanan, B. Likelihood ratios for out-of-distribution detection. In *Advances in*

Neural Information Processing Systems 32, pp. 14680–14691, 2019.

Rhodes, B., Xu, K., and Gutmann, M. U. Telescoping density-ratio estimation. In *Advances in Neural Information Processing Systems 33*, pp. 4905–4916, 2020.

Ruiqi, G., Nijkamp, E., Kingma, D., Xu, Z., Dai, A., and Wu, Y. Flow contrastive estimation of energy-based models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7518–7528, 2020.

Seitzer, M. pytorch-fid: FID Score for PyTorch, 2020.

Song, Y. and Ermon, S. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems 32*, pp. 11895–11907, 2019.

Song, Y., Garg, S., Shi, J., and Ermon, S. Sliced score matching: A scalable approach to density and score estimation. In *Proceedings of the 35th Conference on Uncertainty in Artificial Intelligence*, 2019.

Tian, Y., Krishnan, D., and Isola, P. *Contrastive Multiview Coding*. Springer, 2020.

Tieleman, T. Training restricted Boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning*, 2008.

Vaart, A. W. v. d. *Asymptotic Statistics*. Cambridge University Press, 1998.

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

Wang, B. and Yang, T. Finite-sum coupled compositional stochastic optimization: Theory and applications. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 23292–23317, 2022.

Wang, M., Liu, J., and Fang, E. Accelerating stochastic composition optimization. In *Advances in Neural Information Processing Systems 29*, pp. 1714–1722, 2016.

Wang, M., Fang, E. X., and Liu, H. Stochastic compositional gradient descent: algorithms for minimizing compositions of expected-value functions. *Mathematical Programming*, 161(1-2):419–449, 2017a.

Wang, M., Liu, J., and Fang, E. X. Accelerating stochastic composition optimization. *Journal of Machine Learning Research*, 18:105:1–105:23, 2017b.

Wasserman, L. *All of Statistics*. Springer, 2004.

Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on Machine Learning*, 2011.

Xie, J., Lu, Y., Zhu, S.-C., and Wu, Y. N. A theory of generative convnet. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, pp. 2635–2644, 2016.

Xie, J., Lu, Y., Gao, R., and Wu, Y. N. Cooperative learning of energy-based model and latent variable model via mcmc teaching. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.

Xie, J., Zheng, Z., and Li, P. Learning energy-based model with variational auto-encoder as amortized sampler. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

Xie, J., Zhu, Y., Li, J., and Li, P. A tale of two flows: Cooperative learning of langevin flow and normalizing flow toward energy-based model. In *International Conference on Learning Representations*, 2022.

Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., and Xiao, J. LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *ArXiv e-prints*, arXiv:1506.03365, 2015.

Yu, L., Song, Y., Song, J., and Ermon, S. Training deep energy-based models with f-divergence minimization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 10957–10967, 2020.

Zagoruyko, S. and Komodakis, N. Wide residual networks. In *Proceedings of the British Machine Vision Conference*, 2016.

Zhang, J. and Xiao, L. A stochastic composite gradient method with incremental variance reduction. In *Advances in Neural Information Processing Systems 32*, pp. 9075–9085, 2019.

Zhang, J. and Xiao, L. Multilevel composite stochastic optimization via nested variance reduction. *SIAM Journal on Optimization*, 31(2):1131–1157, 2021.

Zhao, Y., Xie, J., and Li, P. Learning energy-based generative models via coarse-to-fine expanding and sampling. In *International Conference on Learning Representations*, 2021.

A. Omitted Experimental Results

In this section, we present omitted experimental results on the task of density estimation and image generation.

A.1. More Results on Density Estimation

Results for Adam optimizer First, we show the performance of different methods with Adam (or Adam-style) optimizer in the task of Density Estimation on Synthetic Data. The results are shown in Table 6 and Table 7. As can be seen, our method performs better than other algorithms in most cases.

Table 6. Results on synthetic data in terms of MMD with Adam optimizer (lower is better).

Method	*2spirals*	*8gaussians*	*checkerboard*	*circles*	*moons*	*swissroll*
NCE	3.230 ± 0.460	0.130 ± 0.055	1.710 ± 0.190	1.045 ± 0.208	4.588 ± 0.688	1.510 ± 0.498
NCE (NGD)	2.533 ± 0.716	0.128 ± 0.020	1.807 ± 0.451	0.947 ± 0.265	4.202 ± 0.474	1.576 ± 0.589
eNCE (NGD)	2.585 ± 0.841	0.146 ± 0.082	1.690 ± 0.264	0.968 ± 0.342	3.852 ± 0.957	1.604 ± 0.490
MCMC	**2.342 ± 0.658**	0.178 ± 0.105	1.710 ± 0.394	0.937 ± 0.387	3.680 ± 0.916	1.438 ± 0.375
Score Matching	2.727 ± 0.579	0.135 ± 0.071	3.130 ± 0.497	1.706 ± 0.309	5.296 ± 0.221	2.874 ± 0.688
Contrastive Divergence	2.516 ± 0.984	0.173 ± 0.066	1.683 ± 0.904	0.927 ± 0.432	3.904 ± 0.914	1.498 ± 0.368
Ours	2.488 ± 0.356	**0.117 ± 0.086**	**1.651 ± 0.166**	**0.886 ± 0.129**	**3.644 ± 0.490**	**1.338 ± 0.131**

Table 7. Results on synthetic data in terms of FID with Adam optimizer (lower is better).

Method	*2spirals*	*8gaussians*	*checkerboard*	*circles*	*moons*	*swissroll*
NCE	0.053 ± 0.030	0.096 ± 0.042	0.156 ± 0.121	0.068 ± 0.028	0.046 ± 0.031	0.159 ± 0.069
NCE (NGD)	0.067 ± 0.315	0.104 ± 0.038	0.167 ± 0.091	0.061 ± 0.030	0.029 ± 0.017	0.171 ± 0.065
eNCE (NGD)	0.072 ± 0.053	0.127 ± 0.085	0.102 ± 0.081	0.066 ± 0.046	0.038 ± 0.032	0.205 ± 0.035
MCMC	0.047 ± 0.025	0.107 ± 0.073	0.097 ± 0.084	0.049 ± 0.028	0.052 ± 0.021	0.161 ± 0.088
Score Matching	0.070 ± 0.024	0.087 ± 0.050	0.231 ± 0.110	0.086 ± 0.034	0.037 ± 0.018	0.190 ± 0.148
Contrastive Divergence	0.064 ± 0.025	0.092 ± 0.034	0.093 ± 0.034	0.076 ± 0.038	0.062 ± 0.022	0.148 ± 0.029
Ours	**0.040 ± 0.026**	**0.067 ± 0.030**	**0.086 ± 0.101**	**0.040 ± 0.035**	**0.028 ± 0.037**	**0.144 ± 0.078**

Results with different γ and β Then, we investigate the behavior of our algorithm with different values of γ and β. In the previous experiment, we simply set $\gamma = 0.1$ and $\beta = 0.9$. Here, we first fix $\gamma = 0.1$ and enumerate β from the set $\{0.8, 0.9, 0.99\}$. Then, we fix $\beta = 0.9$ and enumerate γ from the set $\{0.01, 0.1, 0.2\}$. The results are reported in Table 8 and Table 9, which indicates that our method is not very sensitive to the choice of β and γ within a certain range.

Table 8. Results on synthetic data in terms of MMD (lower is better).

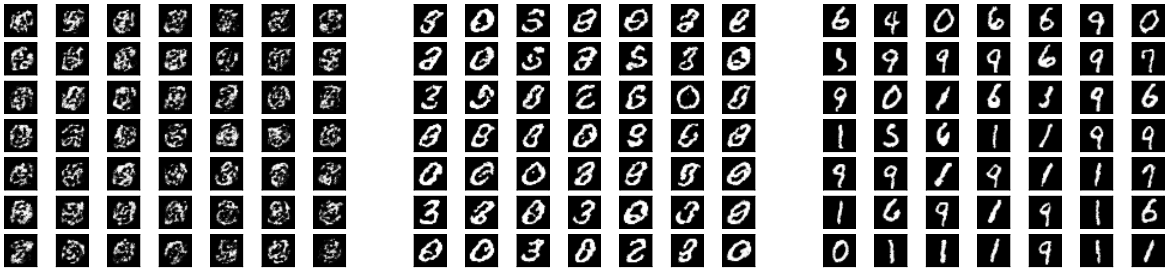
value	*2spirals*	*8gaussians*	*checkerboard*	*circles*	*moons*	*swissroll*
$\beta = 0.8$	3.355 ± 0.781	0.168 ± 0.098	1.889 ± 0.506	1.355 ± 0.164	4.762 ± 0.192	2.784 ± 0.618
$\beta = 0.9$	3.040 ± 0.199	0.132 ± 0.098	1.645 ± 0.251	1.075 ± 0.169	4.673 ± 0.519	2.566 ± 0.274
$\beta = 0.99$	3.038 ± 0.395	0.150 ± 0.078	1.603 ± 0.258	1.120 ± 0.211	4.816 ± 0.292	2.528 ± 0.371

Table 9. Results on synthetic data in terms of MMD (lower is better).

Method	*2spirals*	*8gaussians*	*checkerboard*	*circles*	*moons*	*swissroll*
$\gamma = 0.01$	2.982 ± 0.304	0.141 ± 0.106	1.730 ± 0.716	1.345 ± 0.198	4.650 ± 0.302	2.595 ± 0.493
$\gamma = 0.1$	3.040 ± 0.199	0.132 ± 0.098	1.645 ± 0.251	1.075 ± 0.169	4.673 ± 0.519	2.566 ± 0.274
$\gamma = 0.2$	3.168 ± 0.360	0.145 ± 0.052	1.718 ± 0.418	1.152 ± 0.109	4.772 ± 0.651	2.648 ± 0.601

A.2. More Results on Image Generation

Training EBM on MNIST To verify our analysis for the noise distribution, we compare the image generation results with different noises: 1) the empirical distribution $\mathcal{D}_n = \{\mathbf{x}_1, \cdots, \mathbf{x}_n\}$; 2) Gaussian distribution with its mean and variance fitted on the training data; 3) mixture of empirical distribution $\mathcal{D}_n$ and a fitted Gaussian noise. The results are shown in Figure 5. As can be seen, the first two cases perform poorly and only the third case generates images similar to the original MNIST dataset, which is consistent with our suggestions.



(a) empirical distribution (b) fitted Gaussian distribution (c) empirical distribution + noise

Figure 5. Generated samples with different noise distributions.

Training EBM on CIFAR-10 We further test our method on the more complex CIFAR-10 dataset. We use the codebase of the JEM model (Grathwohl et al., 2020a), which performs generation and classification simultaneously. We replace the standard EBM with Langevin dynamics with our proposed method, and follow exactly the same parameter setting as specified in the codebase (Grathwohl, 2020). We test our method for both classification and image generation tasks, and compare with the original JEM method. For this sets of experiments on more complex natural images, we find that it is more important to design an appropriate noise distribution. Our *empirical distribution + noise* method does not work well in this case, partially due to the fact that the empirical data are too sparsely located at the complex data manifold. Consequently, the estimated density can induce too large variance, leading to slow convergence. We mitigate the limitation by introducing a few MCMC steps (around 10) by directly starting from a pool of updating negative samples, and adopt the kernel density estimation to approximate the density values (as they do not have closed forms after MCMC steps). We find this is essential for learning to generate high-quality real images. Note our adopted method still has limitations, for example, it still requires MCMC steps that we want to avoid, and the estimated densities are not accurate enough. We thus leave designing a good noise estimation for more advanced image generation as interesting future work. Following the instruction in the codebase, we achieve a classification accuracy of 90.72% with a loss of 0.34, compared to an accuracy of 89.38% and a loss of 0.39 from the JEM. Regarding the generated image quality, we measure it with the Inception Score (IS) and Fréchet Inception Distance (FID) score (Seitzer, 2020), based on 10k generated images. Our method obtains an improved FID score from 54.76 to 51.77, while maintaining comparable IS scores, *i.e.*, 4.79 (ours) versus 4.83 (JEM). Note that these numbers are a little worse than what were reported in the JEM paper, mostly due to different experimental settings[1]. Some generated samples from our method are plotted in Figure 6.

B. Analysis

In this section, we present the proof of all theorems.

B.1. Proof of Theorem 1

Here, we prove the first property of Theorem 1. The other properties — asymptotically normal and asymptotically optimal can be obtained according to the property of MLE (for example, see Chapter 9 in Wasserman (2004)).

[1]As confessed by the JEM author, their reported results need heavily manual tuning, which is not easily reproduced (see issue #7 in the codebase). The results are also far from the current state-of-the-arts. However, our goal is only to demonstrate the effectiveness of our method compared to the standard EBM training under a simple setting. Thus, our results are reasonable and the comparison is fair.

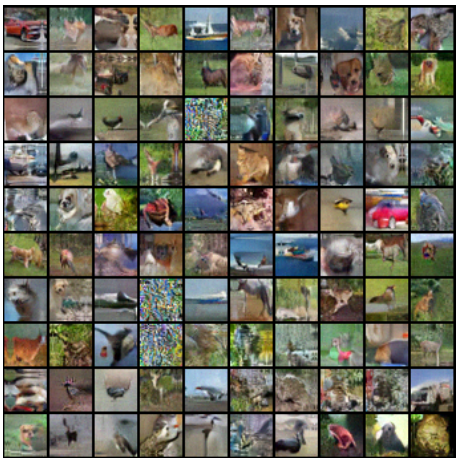


Figure 6. Generated samples on the CIFAR-10 dataset.

We note that the minimizing of $\mathcal{L}(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\log p(\mathbf{x}_i;\theta)$ is equivalent to minimizing

$$M_n(\theta) = -\frac{1}{n}\sum_{i=1}^{n}\log\frac{p(\mathbf{x}_i;\theta)}{p(\mathbf{x}_i;\theta^*)} = \frac{1}{n}\sum_{i=1}^{n}\log\frac{p(\mathbf{x}_i;\theta^*)}{p(\mathbf{x}_i;\theta)}.$$

By the law of large numbers, as the sample size increases, $M_n(\theta)$ converges to

$$M(\theta) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\left[\log\frac{p(\mathbf{x};\theta^*)}{p(\mathbf{x};\theta)}\right] = \int p(\mathbf{x};\theta^*)\log\frac{p(\mathbf{x};\theta^*)}{p(\mathbf{x};\theta)}d\mathbf{x} = D(\theta^*,\theta),$$

where $D(\theta^*,\theta)$ means the Kullback-Leibler distance between $p(\mathbf{x};\theta^*)$ and $p(\mathbf{x};\theta)$. Hence, we have $M_n(\theta) \xrightarrow{P} D(\theta^*,\theta)$. Then, we prove the first property — consistent is satisfied under the following conditions.

$$1)\ \sup_{\theta}|M_n(\theta) - M(\theta)| \xrightarrow{P} 0; \qquad 2)\ \sup_{\theta:|\theta-\theta^*|\geq\epsilon} M(\theta^*) < M(\theta),\ \text{for every } \epsilon > 0.$$

Since $\widehat{\theta}$ minimizes $M_n(\theta)$, we have $M_n(\widehat{\theta}) \leq M_n(\theta^*)$ and

$$\begin{aligned} M(\widehat{\theta}) - M(\theta^*) &= M_n(\theta^*) - M(\theta^*) + M(\widehat{\theta}) - M_n(\theta^*) \\ &\leq M_n(\theta^*) - M(\theta^*) + M(\widehat{\theta}) - M_n(\widehat{\theta}) \\ &\leq M_n(\theta^*) - M(\theta^*) + \sup_{\theta}|M(\theta) - M_n(\theta)| \\ &\xrightarrow{P} 0. \end{aligned}$$

It follows that, for any $\delta > 0$,

$$\mathbb{P}\left(M(\theta^*) < M(\widehat{\theta}) - \delta\right) \to 0.$$

Select any $\epsilon > 0$. By condition 2), there exists $\delta > 0$ such that $|\theta - \theta^*| \geq \epsilon$ implies that $M(\theta^*) < M(\widehat{\theta}) - \delta$. Hence,

$$\mathbb{P}\left(\left|\widehat{\theta} - \theta^*\right| > \epsilon\right) \leq \mathbb{P}\left(M(\theta^*) < M(\widehat{\theta}) - \delta\right) \to 0.$$

B.2. Proof of Theorem 2

First, we prove the following supporting lemmas. We simplify the notation $g(\theta;\widetilde{\mathbf{z}})$ as $\widetilde{g}(\theta_t)$ and $h(\theta;\mathbf{z})$ as $\widetilde{h}(\theta_t)$.

Lemma 2. *We show that the estimation errors of $\mathbf{u}_t$ and $\mathbf{v}_t$ would be reduced over time.*

$$\mathbb{E}\left[\|\mathbf{u}_{t+1} - g(\theta_{t+1})\|^2\right] \leq (1-\gamma_{t+1})\|\mathbf{u}_t - g(\theta_t)\|^2 + \frac{C^2\eta_t^2}{r_{t+1}}\|\mathbf{v}_t\|^2 + \gamma_{t+1}^2\sigma_g^2;$$
$$\mathbb{E}\left[\|\mathbf{v}_{t+1} - \nabla\mathcal{L}(\theta_{t+1})\|^2\right] \leq (1-\beta_{t+1})\mathbb{E}\left[\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2\right] + \frac{2L_0^2 + 18C^4L^2}{\beta_{t+1}}\eta_t^2\|\mathbf{v}_t\|^2$$
$$+ 18\beta_{t+1}C^2L^2\|\mathbf{u}_t - g(\theta_t)\|^2 + 2\beta_{t+1}^2\zeta_h^2 + 6C^2L^2\beta_{t+1}^2\sigma_g^2 + 2\beta_{t+1}^2C^2\zeta_g^2.$$

Proof. Consider the update $\mathbf{u}_{t+1} = (1-\gamma_{t+1})\mathbf{u}_t + \gamma_{t+1}\widetilde{g}(\theta_{t+1})$, $\theta_{t+1} = \theta_t - \eta_t\mathbf{v}_t$, and set $p = \frac{\gamma_{t+1}}{1-\gamma_{t+1}}$.

$$\begin{aligned}
&\mathbb{E}\left[\|\mathbf{u}_{t+1} - g(\theta_{t+1})\|^2\right] \\
=&\mathbb{E}\left[\|(1-\gamma_{t+1})(\mathbf{u}_t - g(\theta_{t+1})) + \gamma_{t+1}(\widetilde{g}(\theta_{t+1}) - g(\theta_{t+1}))\|^2\right] \\
=&\|(1-\gamma_{t+1})(\mathbf{u}_t - g(\theta_{t+1}))\|^2 + \mathbb{E}\left[\|\gamma_{t+1}(\widetilde{g}(\theta_{t+1}) - g(\theta_{t+1}))\|^2\right. \\
&\left.+\gamma_{t+1}(1-r_{t+1})(\mathbf{u}_t - g(\theta_{t+1}))(\widetilde{g}(\theta_{t+1}) - g(\theta_{t+1}))\right] \\
=&\|(1-\gamma_{t+1})(\mathbf{u}_t - g(\theta_{t+1}))\|^2 + \gamma_{t+1}^2\sigma_g^2 \\
\leq&(1-\gamma_{t+1})^2\|\mathbf{u}_t - g(\theta_{t+1})\|^2 + \gamma_{t+1}^2\sigma_g^2 \\
\leq&(1-\gamma_{t+1})^2(1+p)\|\mathbf{u}_t - g(\theta_t)\|^2 + (1-\gamma_{t+1})^2(1+\frac{1}{p})\|g(\theta_t) - g(\theta_{t+1})\|^2 + \gamma_{t+1}^2\sigma_g^2 \\
=&(1-\gamma_{t+1})\|\mathbf{u}_t - g(\theta_t)\|^2 + \frac{C^2\eta_t^2}{r_{t+1}}\|\mathbf{v}_t\|^2 + \gamma_{t+1}^2\sigma_g^2,
\end{aligned}$$

where the last inequality is due to the fact that $(a+b)^2 \leq (1+p)a^2 + (1+\frac{1}{p})b^2$.

According to Assumption 1, we know that $\mathcal{L}(\cdot)$ is also C_0-Lipchitz continuous and L_0-smooth, where $C_0 = (C+1)C$ and $L_0 = \left(C^2 + C + 1\right)L$. By setting $p = \frac{\beta_{t+1}}{1-\beta_{t+1}}$, we have:

$$\begin{aligned}
&\mathbb{E}\left[\|\mathbf{v}_{t+1} - \nabla\mathcal{L}(\theta_{t+1})\|^2\right] \\
&= \mathbb{E}\left[\left\|(1-\beta_{t+1})(\mathbf{v}_t - \nabla\mathcal{L}(\theta_{t+1})) + \beta_{t+1}[\nabla\widetilde{h}(\theta_{t+1}) + \nabla f(\mathbf{u}_{t+1})\nabla\widetilde{g}(\theta_{t+1}) - \nabla h(\theta_{t+1}) - \nabla f(g(\theta_{t+1}))\nabla g(\theta_{t+1})]\right\|^2\right] \\
&= \mathbb{E}\left[\|(1-\beta_{t+1})(\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)) + (1-\beta_{t+1})(\nabla\mathcal{L}(\theta_t) - \nabla\mathcal{L}(\theta_{t+1})) + \beta_{t+1}\nabla g(\theta_{t+1})(\nabla f(\mathbf{u}_{t+1}) - \nabla f(g(\theta_{t+1})))\right. \\
&\quad\left.+\beta_{t+1}[\nabla\widetilde{h}(\theta_{t+1}) + \nabla f(\mathbf{u}_{t+1})\nabla\widetilde{g}(\theta_{t+1}) - \nabla h(\theta_{t+1}) - \nabla f(\mathbf{u}_{t+1})\nabla g(\theta_{t+1})]\right\|^2\right] \\
&\leq \mathbb{E}\left[\|(1-\beta_{t+1})(\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)) + (1-\beta_{t+1})(\nabla\mathcal{L}(\theta_t) - \nabla\mathcal{L}(\theta_{t+1})) + \beta_{t+1}\nabla g(\theta_{t+1})(\nabla f(\mathbf{u}_{t+1}) - \nabla f(g(\theta_{t+1})))\|^2\right] \\
&\quad+ \mathbb{E}\left[2\beta_{t+1}^2\left\|\nabla\widetilde{h}(\theta_{t+1}) - \nabla h(\theta_{t+1})\right\|^2 + 2\beta_{t+1}^2\|\nabla f(\mathbf{u}_{t+1})(\nabla\widetilde{g}(\theta_{t+1}) - \nabla g(\theta_{t+1}))\|^2\right] \\
&\leq \mathbb{E}\left[(1-\beta_{t+1})^2(1+p)\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2 + 2(1-\beta_{t+1})^2(1+\frac{1}{p})\|\nabla\mathcal{L}(\theta_t) - \nabla\mathcal{L}(\theta_{t+1})\|^2\right] \\
&\quad+ 2\beta_{t+1}^2(1+\frac{1}{p})C^2\|\nabla f(\mathbf{u}_{t+1}) - \nabla f(g(\theta_{t+1}))\|^2 + 2\beta_{t+1}^2\zeta_h^2 + 2\beta_{t+1}^2C^2\zeta_g^2 \\
&\leq (1-\beta_{t+1})\mathbb{E}\left[\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2\right] + \frac{2L_0^2}{\beta_{t+1}}\eta_t^2\|\mathbf{v}_t\|^2 + 2\beta_{t+1}C^2L^2\|\mathbf{u}_{t+1} - g(\theta_{t+1})\|^2 + 2\beta_{t+1}^2\zeta_h^2 + 2\beta_{t+1}^2C^2\zeta_g^2 \\
&\leq (1-\beta_{t+1})\mathbb{E}\left[\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2\right] + \frac{2L_0^2 + 18C^4L^2}{\beta_{t+1}}\eta_t^2\|\mathbf{v}_t\|^2 + 18\beta_{t+1}C^2L^2\|\mathbf{u}_t - g(\theta_t)\|^2 \\
&\quad+ 2\beta_{t+1}^2\zeta_h^2 + 6C^2L^2\beta_{t+1}^2\sigma_g^2 + 2\beta_{t+1}^2C^2\zeta_g^2
\end{aligned}$$

□

To finish the proof, we also need to introduce the following lemma.

Lemma 3. *(Guo et al., 2021) Suppose function $\mathcal{L}$ is L_0-smooth and consider the update $\theta_{t+1} := \theta_t - \eta_t \mathbf{v}_t$. With $\eta_t L_0 \leq \frac{1}{2}$, we have:*

$$\mathcal{L}(\theta_{t+1}) \leq \mathcal{L}(\theta_t) - \frac{\eta_t}{2}\|\nabla\mathcal{L}(\theta_t)\|^2 + \frac{\eta_t}{2}\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2 - \frac{\eta_t}{4}\|\mathbf{v}_t\|^2$$

According to above lemmas, we have:

$$\begin{aligned}
\sum_{t=1}^{T}\gamma_{t+1}\mathbb{E}\left[\|\mathbf{u}_t - g(\theta_t)\|^2\right] &\leq \|\mathbf{u}_1 - g(\theta_1)\|^2 + \sum_{t=1}^{T}\frac{C^2\eta_t^2}{r_{t+1}}\|\mathbf{v}_t\|^2 + \sum_{t=1}^{T}\gamma_{t+1}^2\sigma_g^2 \\
\sum_{t=1}^{T}\beta_{t+1}\mathbb{E}\left[\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2\right] &\leq \mathbb{E}\left[\|\mathbf{v}_1 - \nabla\mathcal{L}(\theta_1)\|^2\right] + \sum_{t=1}^{T}\frac{2L_0^2 + 18C^4L^2}{\beta_{t+1}}\eta_t^2\|\mathbf{v}_t\|^2 \\
&\quad + 18C^2L^2\sum_{t=1}^{T}\beta_{t+1}\|\mathbf{u}_t - g(\theta_t)\|^2 + \sum_{t=1}^{T}\beta_{t+1}^2\left(2\zeta_h^2 + 6C^2L^2\sigma_g^2 + 2C^2\zeta_g^2\right) \\
\sum_{t=1}^{T}\frac{\eta_t}{2}\|\nabla\mathcal{L}(\theta_t)\|^2 &\leq \mathcal{L}(\theta_1) + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2 - \sum_{t=1}^{T}\frac{\eta_t}{4}\|\mathbf{v}_t\|^2
\end{aligned}$$

Summing up, we have:

$$\begin{aligned}
&\sum_{t=1}^{T}\gamma_{t+1}\|\mathbf{u}_t - g(\theta_t)\|^2 + \beta_{t+1}\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2 + \frac{\eta_t}{2}\|\nabla\mathcal{L}(\theta_t)\|^2 \\
\leq& L_1 + \sum_{t=1}^{T}\left(\frac{C^2\eta_t}{r_{t+1}} + \frac{(2L_0^2 + 18C^4L^2)\eta_t}{\beta_{t+1}} - \frac{1}{4}\right)\eta_t\|\mathbf{v}_t\|^2 + L_1\sum_{t=1}^{T}\left(\gamma_{t+1}^2 + \beta_{t+1}^2\right) \\
&+ 18C^2L^2\sum_{t=1}^{T}\beta_{t+1}\|\mathbf{u}_t - g(\theta_t)\|^2 + \sum_{t=1}^{T}\frac{\eta_t}{2}\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2,
\end{aligned}$$

where $L_1 = \max\left\{\|\mathbf{u}_1 - g(\theta_1)\|^2 + \|\mathbf{v}_1 - \nabla\mathcal{L}(\theta_1)\|^2 + \mathcal{L}(\theta_1), \sigma_g^2, 2\zeta_h^2 + 6C^2L^2\sigma_g^2 + 2C^2\zeta_g^2\right\}$

By setting $\beta_{t+1} = L_2\eta_t$ and $\gamma_{t+1} = L_3\beta_t$, where $L_2 = (16L_0^2 + 144C^4L^2 + 1)$ and $L_3 = 36C^2(L^2+1)$, we have:

$$\sum_{t=1}^{T}\frac{L_2L_3}{2}\eta_t\|\mathbf{u}_t - g(\theta_t)\|^2 + \frac{L_2}{2}\eta_t\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2 + \frac{\eta_t}{2}\|\nabla\mathcal{L}(\theta_t)\|^2 \leq L_1 + L_1(L_2^2 + L_2^2L_3^2)\sum_{t=1}^{T}\eta_{t+1}^2$$

According to the analysis of previous works, such as NASA (Ghadimi et al., 2020), since $\sum_{t=1}^{T}\eta_{t+1} = \infty$ a.s., and $\sum_{t=1}^{T}\eta_{t+1}^2 < \infty$, we would have $\mathbf{u}_t \to g(\theta_t)$, $\mathbf{v}_t \to \nabla\mathcal{L}(\theta_t)$, $\|\nabla\mathcal{L}(\theta_t)\|^2 \to 0$, otherwise the left part of above equation tends to infinity, while the right part is not.

B.3. Proof of Theorem 3

With supporting lemmas in previous theorem, by setting $\beta_t = \gamma_t = \beta, \eta_t = \eta$, we have:

$$\begin{aligned}
\mathbb{E}\left[\beta\sum_{t=1}^{T}\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2\right] \leq& \mathbb{E}\left[\|\mathbf{v}_1 - \nabla\mathcal{L}(\theta_1)\|^2\right] + \frac{2L_0^2 + 18C^4L^2}{\beta}\sum_{t=1}^{T}\eta^2\|\mathbf{v}_t\|^2 \\
&+ 18\beta C^2L^2\sum_{t=1}^{T}\|\mathbf{u}_t - g(\theta_t)\|^2 + (2\zeta_h^2 + 6C^2L^2\sigma_g^2 + 2C^2\zeta_g^2)\beta^2T,
\end{aligned}$$

where $\mathbb{E}\left[\|\mathbf{v}_1 - \nabla\mathcal{L}(\theta_1)\|^2\right] \leq 3\zeta_h^2 + 3C^2\zeta_g^2 + 3C^2L^2\sigma_g^2$.

Similarly, for the term $\|\mathbf{u}_t - g(\theta_t)\|^2$, we have:

$$\sum_{t=1}^{T}\|\mathbf{u}_t - g(\theta_t)\|^2 \leq \frac{1}{\beta}\|\mathbf{u}_1 - g(\theta_1)\|^2 + \frac{C^2\eta^2}{\beta^2}\sum_{t=1}^{T}\|\mathbf{v}_t\|^2 + \beta\sigma_g^2 T \leq \frac{\sigma_g}{\beta} + \frac{C^2\eta^2}{\beta^2}\sum_{t=1}^{T}\|\mathbf{v}_t\|^2 + \beta\sigma_g^2 T.$$

So, we have

$$\begin{aligned}
&\mathbb{E}\left[\sum_{t=1}^{T}\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2\right] \\
&\leq \frac{3\zeta_h^2 + 3C^2\zeta_g^2 + 21C^2L^2\sigma_g^2}{\beta} + \frac{2L_0^2 + 36C^4L^2}{\beta^2}\eta^2\sum_{t=1}^{T}\|\mathbf{v}_t\|^2 + (2\zeta_h^2 + 2C^2\zeta_g^2 + 24C^2L^2\sigma_g^2)\beta T \\
&= \frac{L_2}{\beta} + \frac{L_4}{\beta^2}\eta^2\sum_{t=1}^{T}\|\mathbf{v}_t\|^2 + L_3\beta T,
\end{aligned}$$

where L_2, L_3, L_4 is the constant.

Define the $\beta = \min\{\frac{\epsilon^2}{3L_3}, 1\}, \eta = \min\{\frac{\epsilon^2}{3\sqrt{2L_4}L_3}, \frac{1}{2L_0}, \frac{\beta}{\sqrt{2L_4}}\}, T = \max\{\frac{L_2}{\beta\epsilon^2}, \frac{2\mathcal{L}(\theta_1)}{\eta\epsilon^2}\}$. We can have:

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\|\nabla\mathcal{L}(\theta_t)\|^2\right] &\leq \frac{2\mathcal{L}(\theta_1)}{\eta T} + \frac{1}{T}\mathbb{E}\left[\sum_{t=1}^{T}\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2\right] - \frac{1}{2T}\sum_{t=1}^{T}\|\mathbf{v}_t\|^2 \\
&\leq \frac{2\mathcal{L}(\theta_1)}{\eta T} + \frac{L_2}{\beta T} + \left(\frac{L_4\eta^2}{\beta^2 T} - \frac{1}{2T}\right)\sum_{t=1}^{T}\|\mathbf{v}_t\|^2 + L_3\beta \\
&\leq \epsilon^2.
\end{aligned}$$

To hide the constant, we finally have $T = \max\left\{\mathcal{O}\left(\frac{1}{\epsilon^2}\right), \mathcal{O}\left(\frac{\zeta_h^2+\zeta_g^2+\sigma_g^2}{\epsilon^4}\right)\right\}$. We can easily extend the results to the case of adaptive learning rates in the style of Adam, following the analysis of Guo et al. (2021).

B.4. Proof of Theorem 4

If $\mathcal{L}(\theta)$ is μ-PL, which is $2\mu(\mathcal{L}(\theta) - \mathcal{L}^*) \leq \|\nabla\mathcal{L}(\theta)\|^2$, we can further improve the rate.

Define that $\gamma_{t+1} = \beta_{t+1}, \Gamma_t = \mathcal{L}(\theta_t) - \mathcal{L}^* + A\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2 + B\|\mathbf{u}_t - g(\theta_t)\|^2$, and then we have:

$$\begin{aligned}
&\mathbb{E}\left[\Gamma_{t+1}\right] \\
\leq&(1-\mu\eta_t)\mathbb{E}\left[\mathcal{L}(\theta_t) - \mathcal{L}^*\right] + \frac{\eta_t}{2}\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2 - \frac{\eta_t}{4}\|\mathbf{v}_t\|^2 + A(1-\beta_{t+1})\|\mathbf{v}_t - \nabla\mathcal{L}(\theta_t)\|^2 \\
&+ \frac{2L_0^2 + 18C^4L^2}{\beta_{t+1}}A\eta_t^2\|\mathbf{v}_t\|^2 + 18\beta_{t+1}C^2L^2A\|\mathbf{u}_t - g(\theta_t)\|^2 + A(2\zeta_h^2 + 6C^2L^2\sigma_g^2 + 2C^2\zeta_g^2)\beta_{t+1}^2 \\
&+ B(1-\beta_{t+1})\|\mathbf{u}_t - g(\theta_t)\|^2 + \frac{BC^2\eta_t^2}{\beta_{t+1}}\|\mathbf{v}_t\|^2 + B\beta_{t+1}^2\sigma_g^2
\end{aligned}$$

Setting $B = 36C^2L^2A, \beta_{t+1} = \max\{2\mu, \frac{1}{A}, 8A(2L_0^2 + 36C^4L^2)\}\eta_t$, we have:

$$\mathbb{E}\left[\Gamma_{t+1}\right] \leq (1-\mu\eta_t)\mathbb{E}\left[\Gamma_t\right] + A\beta_{t+1}^2(2\zeta_h^2 + 42C^2L^2\sigma_g^2 + 2C^2\zeta_g^2)$$

Define $L' = 2\zeta_h^2 + 42C^2L^2\sigma_g^2 + 2C^2\zeta_g^2, L'' = \max\{8(2L_0^2 + 54C^4L^2), 2\}$, we have:

$$\mathbb{E}\left[\Gamma_{t+1}\right] \leq (1-\mu\eta_t)\mathbb{E}\left[\Gamma_t\right] + A\beta_{t+1}^2 L'.$$

If $\mu \leq 1$, set $A = 1, \beta_{t+1} = L''\eta_t, 1 - \mu\eta_t = \frac{\eta_t^2}{\eta_{t-1}^2}$, which is $\frac{1}{\eta_t} = \frac{\mu}{2} + \sqrt{\frac{\mu^2}{4} + \frac{1}{\eta_{t-1}^2}}$.

$$\begin{aligned}
\mathbb{E}\left[\Gamma_{T+1}\right] &\leq (1 - \mu\eta_T)\mathbb{E}\left[\Gamma_T\right] + L'(L'')^2\eta_T^2 \\
&\leq \frac{\eta_T}{\eta_{T-1}}\mathbb{E}\left[\Gamma_T\right] + L'(L'')^2\eta_T^2 \\
&\leq \frac{\eta_T}{\eta_{T-2}}\mathbb{E}\left[\Gamma_T\right] + L'(L'')^2\eta_T^2 * 2 \\
&\leq \frac{\eta_T}{\eta_0}\mathbb{E}\left[\Gamma_T\right] + L'(L'')^2\eta_T^2 * T
\end{aligned}$$

We have $\frac{1}{\eta_t} \geq \frac{\mu}{2} + \frac{1}{\eta_{t-1}}$, so we have $\frac{1}{\eta_T} \geq \frac{\mu T}{2} + \frac{1}{\eta_0}$, where $\eta_0 > 0$ Finally, we have $\eta_T \leq \frac{2}{\mu T}$.

So $\mathbb{E}\left[\Gamma_{T+1}\right] \leq \frac{4}{\eta_0^2\mu^2T^2}\mathbb{E}\left[\Gamma_1\right] + L'(L'')^2\frac{4}{\mu^2 T} \leq \epsilon$. The complexity is $T = \max\left\{\mathcal{O}(\frac{1}{\mu\eta_0\sqrt{\epsilon}}), \mathcal{O}(\frac{\zeta_h^2+\zeta_g^2+\sigma_g^2}{\mu^2\epsilon})\right\}$.

If $\mu \geq 1$, set $A = \frac{1}{\mu}, \beta_{t+1} = L''\mu\eta_t$, and the complexity is $T = \max\left\{\mathcal{O}(\frac{1}{\mu\eta_0\sqrt{\epsilon}}), \mathcal{O}(\frac{\zeta_h^2+\zeta_g^2+\sigma_g^2}{\mu\epsilon})\right\}$.

Note that existing methods usually employ two-loop stage-wise designed algorithms to analyze for PL condition (Wang & Yang, 2022; Jiang et al., 2022a).

B.5. Proof of Lemma 1

We analyze the variance σ_g^2 of our estimator $g(\theta)$. By setting $\mu = g(\theta) = \int p_0(\widetilde{\mathbf{z}};\theta)d\widetilde{\mathbf{z}} = \mathbb{E}_{\widetilde{\mathbf{z}}\sim q(\widetilde{\mathbf{z}})}\left[\frac{p_0(\widetilde{\mathbf{z}};\theta)}{q(\widetilde{\mathbf{z}})}\right]$, the variance of our estimator is

$$\mathbb{E}_{\widetilde{\mathbf{z}}\sim q(\widetilde{\mathbf{z}})}\left\|\frac{p_0(\widetilde{\mathbf{z}};\theta)}{q(\widetilde{\mathbf{z}})} - \mu\right\|^2 = \int\left(\frac{p_0(\widetilde{\mathbf{z}};\theta)}{q(\widetilde{\mathbf{z}})} - \mu\right)^2 q(\widetilde{\mathbf{z}})d\widetilde{\mathbf{z}} = \int\frac{(p_0(\widetilde{\mathbf{z}};\theta) - \mu q(\widetilde{\mathbf{z}}))^2}{q(\widetilde{\mathbf{z}})}d\widetilde{\mathbf{z}}$$

To get the minimum variance, we should choose $q(\widetilde{\mathbf{z}}) = \frac{p_0(\widetilde{\mathbf{z}};\theta)}{\mu} = \frac{p_0(\widetilde{\mathbf{z}};\theta)}{\int p_0(\widetilde{\mathbf{z}};\theta)d\widetilde{\mathbf{z}}} = p_\theta(\widetilde{\mathbf{z}})$ and the variance would be zero.

B.6. Proof of Proposition 2

First, we write the MLE objective function as:

$$\begin{aligned}
\mathcal{L}(\theta) &= \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\left[-\theta x + \frac{1}{2}x^2\right] + \log\int e^{\theta x - \frac{1}{2}x^2}d\mathbf{x} \\
&= \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}\left[-\theta x + \frac{1}{2}x^2\right] + \log\sqrt{2\pi} + \frac{\theta^2}{2}.
\end{aligned}$$

So, we have:

$$L(\theta) - L(\theta^*) = \mathbb{E}_{\mathbf{x}\sim p_{\text{data}}}[(\theta^* - \theta)\mathbf{x}] + \frac{\theta^2 - \theta^{*2}}{2} = (\theta^* - \theta)\theta^* + \frac{\theta^2 - (\theta^*)^2}{2} = (\theta - \theta^*)^2.$$

B.7. Proof of Proposition 3

We prove that the objective is 1-strongly convex, which is a stronger condition than 1-PL condition.

Assume that $\mu = \int x p_\theta(x) d\mathbf{x}$ and we have:

$$
\begin{aligned}
&\nabla^2 L(\theta) \\
&= \frac{\left(\int x^2 e^{\theta x - \frac{1}{2}x^2} d\mathbf{x}\right)\left(\int e^{\theta x - \frac{1}{2}x^2} d\mathbf{x}\right) - \left(\int x e^{\theta x - \frac{1}{2}x^2} d\mathbf{x}\right)^2}{\left[\int e^{\theta x - \frac{1}{2}x^2} d\mathbf{x}\right]^2} \\
&= \frac{\int x^2 e^{\theta x - \frac{1}{2}x^2} d\mathbf{x}}{\int e^{\theta x - \frac{1}{2}x^2} d\mathbf{x}} - \left[\frac{\int x e^{\theta x - \frac{1}{2}x^2} d\mathbf{x}}{\int e^{\theta x - \frac{1}{2}x^2} d\mathbf{x}}\right]^2 \\
&= \int x^2 p_\theta(x) d\mathbf{x} - \left[\int x p_\theta(x) d\mathbf{x}\right]^2 \\
&= \int x^2 p_\theta(x) d\mathbf{x} - 2\mu \int x p_\theta(x) d\mathbf{x} + \mu^2 \\
&= \int x^2 p_\theta(x) - 2\mu x p_\theta(x) + \mu^2 p_\theta(x) d\mathbf{x} \\
&= \int (x-\mu)^2 p_\theta(x) d\mathbf{x} \\
&= \mathbb{E}_{p(\theta)}\left[(x-\mu)^2\right] \\
&= 1
\end{aligned}
$$

The last equation is due to the fact that the variance is fixed as 1 for the model. So, the objective is 1-strongly convex, and according to our analysis for PL condition, our method enjoys a complexity of $\mathcal{O}(\epsilon^{-1})$ due to Theorem 4.