ON GENERATIVE ENERGY-BASED MODELS

YVES ATCHADÉ, KEER JIANG, AND YI SUN

(May 2023)

Abstract. Energy-based models (EBM) are well-known density estimation models that are statistically attractive, but computationally difficult to fit. We connect the short-run MCMC method of Nijkamp et al. (2019) with the algorithm unrolling framework to make the case for a new class of density estimation models that we call `generative EBM` (GEBM). We show that the short-run MCMC method of Nijkamp et al. (2019) implicitly fits a GEBM by minimizing a maximum mean discrepancy (MMD) metric, where the MMD kernel is taken as the neural tangent kernel of the related deep neural network function. The idea can be applied more broadly, and as an illustration, we propose a new and fast estimation procedure for high-dimensional Gaussian graphical models under a ℓ^1-norm penalty.

1. Introduction

Energy-based models are generalizations of graphical models that are widely used in machine learning. These models first appeared in statistical physics with the seminal work of Ising (1925), and were later studied in various fields under sometimes different names: Gibbs measures Georgii (1988), Markov random fields Guyon (1995), graphical models Besag (1974), Boltzmann machine Hinton and Sejnowski (1983). The term energy-based model originates from the machine learning community and refers to specifications where the negative log-density is a deep neural network function (LeCun et al. (2006); Du and Mordatch (2019); Du et al. (2020); Ingraham et al. (2019)). However, in this work we will use the term energy-based model more broadly.

2010 *Mathematics Subject Classification.* 62F15, 62Jxx.

Key words and phrases. Energy-based models, density estimation, deep learning, graphical lasso, stochastic proximal gradient.

This work is partially supported by the NSF grant DMS 2015485 and 2210664.

Y. Atchadé: Boston University, 111 Cummington Mall, Boston 02215 MA, United States. *E-mail address:* atchade@bu.edu.

K. Jiang: Boston University, 111 Cummington Mall, Boston 02215 MA, United States. *E-mail address:* kejiang@bu.edu.

Y. Sun: Boston University, 111 Cummington Mall, Boston 02215 MA, United States. *E-mail address:* ysun4@bu.edu.

On the flip side of their great modeling flexibility is the fact that EBMs are difficult to fit due to their intractable normalizing constants. Classical maximum likelihood inference for EBMs leads to a representation of the score function as an integral with respect to the energy-based distribution. Evaluating these integrals becomes the computational bottleneck. Most of the literature on this issue has focused on developing efficient MCMC methods with various heuristics (Hinton (2002); Tieleman (2008); Du and Mordatch (2019)).

1.1. **Main contributions.** To address these computational challenges, Nijkamp et al. (2019) introduced the idea of running short, noise-initialized, and non-persistent Markov chains to approximate the score function of EBMs. Their approach greatly simplifies the implementation of EBMs, yet produced remarkably good results. The authors intuited that their approach amounts to fitting, by moment matching, a modified version of the EBM. The main contribution of this work is to further analyze Nijkamp et al. (2019). First, we decouple the modeling framework and the estimation framework of Nijkamp et al. (2019). On the modeling side, we show that their framework is an application of algorithm unrolling, and this yields a new model that we call `generative EBM`. We use the term *generative* here to connote that it is easy to generate samples these models. On the estimation side, we show that the short-run MCMC method of Nijkamp et al. (2019) is a minimum distance estimation of the generative EBM, using a maximum mean discrepancy (MMD) metric. Furthermore, the kernel of the MMD metric is precisely the neural tangent kernel of the deep neural network function.

The key point of this work is that the strategy of replacing an EBM by a generative EBM can be applied more widely. In particular, we show that the method can be used to tackle high-dimensional graphical models that are widely used in statistics. As an illustration, we propose a new and fast estimation procedure for high-dimensional Gaussian graphical models under a ℓ^1-norm penalty that is much faster than the deterministic proximal gradient algorithm, at the cost of a small loss of accuracy.

The remaining of the paper is organized as follows. We introduce the EBMs in Section 2. The approximate EBMs of Nijkamp et al. (2019) are defined and analyzed in Section 3, and an extension to Gaussian graphical models is proposed in Section 4. The numerical illustrations are collected in Section 4 and 5, including illustrations with Gaussian graphical models and image density estimations. Some technical proofs are collected in Section 7, and some concluding thoughts are proposed in Section 6.

2. Density estimation using energy-based models

Let $P_\star$ be a probability measure of interest on some domain $\mathsf{X} \subseteq \mathbb{R}^p$ equipped with its Lebesgue measure that we denote $\mathrm{d}x$ or μ_{Leb}. Suppose that we have random samples $X_1, \ldots, X_n \overset{i.i.d.}{\sim} P_\star$. Let P_n denote the corresponding empirical measure on X:

$$P_n(\cdot) \overset{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \delta_{X_i}(\cdot),$$

where δ_x denotes the Dirac measure with mass at x. We consider the problem of estimating $P_\star$ using an EBM $\{p_\theta,\ \theta \in \Theta\}$, where p_θ is of the form

$$p_\theta(x) = \frac{e^{-\mathcal{E}_\theta(x)}}{\int_{\mathsf{X}} e^{-\mathcal{E}_\theta(x)} \mathrm{d}x}, \quad x \in \mathsf{X}, \tag{1}$$

for some function $\mathcal{E}_\theta : \mathsf{X} \to \mathbb{R}$ that we will call the energy function. Throughout we assume that the parameter space Θ is a subset of $\mathbb{R}^d$ with non-empty interior, and equipped with its Euclidean structure, with inner product $\langle \cdot, \cdot \rangle$ and norm $\| \cdot \|_2$. We assume that $\int_{\mathsf{X}} e^{-\mathcal{E}_\theta(x)} \mathrm{d}x < \infty$ for all $\theta \in \Theta$. We also assume that the function $\theta \mapsto \mathcal{E}_\theta(x)$ is differentiable for each $x \in \mathsf{X}$, and the gradient of the function $\theta \mapsto \log \int_{\mathsf{X}} e^{-\mathcal{E}_\theta(x)} \mathrm{d}x$ is

$$-\int_{\mathsf{X}} \nabla_\theta \mathcal{E}_\theta(x) p_\theta(x) \mathrm{d}x,$$

where $\nabla_\theta \mathcal{E}_\theta(x)$ denotes the gradient of $\theta \mapsto \mathcal{E}_\theta(x)$.

One of the appeal of EBMs in machine learning is that they are universal density approximators. Indeed, assuming that X is bounded, if $p(x) = e^{-\mathcal{E}(x)}/Z$ is a density on X, then

$$\|p - p_\theta\|_{\mathrm{tv}} \overset{\text{def}}{=} \sup_{A \subseteq \mathsf{X},\ A \text{ meas.}} \left| \int_A p(x) \mathrm{d}x - \int_A p_\theta(x) \mathrm{d}x \right| \leq \frac{\mu_{\mathsf{Leb}}(\mathsf{X})}{2} \|\mathcal{E} - \mathcal{E}_\theta\|_\infty, \tag{2}$$

where $\|f\|_\infty \overset{\text{def}}{=} \sup_{x \in \mathsf{X}} |f(x)|$. The proof of the inequality in the last display can be found for instance in (Georgii (1988) Section 8.1). From this result, approximation properties of the energy function class $\{\mathcal{E}_\theta,\ \theta \in \Theta\}$ transfer to the EBM. For instance, if $\mathcal{E}_\theta$ is a deep feed-forward model, then recent results (see e.g. DeVore et al. (2021) and the references therein) imply that the resulting EBM is a universal density approximator.

To fit the EBM (1), the negative log-likelihood function of the dataset $(X_1, \ldots, X_n)$ is $n \times \ell_n(\theta)$, where

$$\ell_n(\theta) \overset{\text{def}}{=} \int_{\mathsf{X}} \mathcal{E}_\theta(x) P_n(\mathrm{d}x) + \log \int_{\mathsf{X}} e^{-\mathcal{E}_\theta(x)} \mathrm{d}x. \tag{3}$$

With $G_\theta(x) \stackrel{\text{def}}{=} \nabla_\theta \mathcal{E}_\theta(x)$, it follows that

$$\nabla \ell_n(\theta) = \int_{\mathsf{X}} G_\theta(x) P_n(\mathrm{d}x) - \int_{\mathsf{X}} G_\theta(x) p_\theta(x) \mathrm{d}x.$$

To improve estimation it is often useful to fit the model using a regularization term. Let $\mathcal{R} : \mathbb{R}^d \to (-\infty, +\infty]$ be a regularization function that we will assume convex with a non-empty domain, but not necessarily smooth. Let $\mathrm{Prox}_\gamma^{\mathcal{R}}$ denote its proximal operator. Specifically, given $\gamma > 0$,

$$\mathrm{Prox}_\gamma^{\mathcal{R}}(\theta) = \operatorname*{\mathsf{Argmin}}_{u \in \mathbb{R}^d} \left[\mathcal{R}(u) + \frac{1}{2\gamma} \|u - \theta\|_2^2 \right].$$

A penalized maximum likelihood estimation of θ is then obtained by minimizing the function

$$f_n(\theta) \stackrel{\text{def}}{=} \ell_n(\theta) + \mathcal{R}(\theta). \tag{4}$$

For instance, the initial constraint that θ belongs to the chosen parameter space Θ can be built into the regularization function by taking $\mathcal{R}(\theta) = \mathcal{R}_1(\theta) + \iota_\Theta(\theta)$, for some regularization function $\mathcal{R}_1$, and $\iota_\Theta(u) = 0$ if $u \in \Theta$, $\iota_\Theta(u) = +\infty$ otherwise. In that case, provided that Θ is compact, and $\mathcal{R}_1$ is continuous, The minimization problem (4) is often solved by finding the solutions of the fixed point equation

$$\theta = \mathrm{Prox}_\gamma^{\mathcal{R}} \left(\theta - \gamma \nabla \ell_n(\theta) \right), \tag{5}$$

for appropriate $\gamma > 0$ (see e.g. Combettes and Wajs (2005) Theorem 3.4. and Proposition 3.1.(iii)).

Starting from (5), the maximum penalized likelihood estimator can then be approximated using a well-known stochastic proximal gradient descent algorithm (Ackley et al. (1985); Younes (1988); Nitanda (2014); Atchadé et al. (2017)). A concise description is given in Algorithm 1. This entails drawing random samples from p_θ (typically using MCMC) to approximate the second integral in $\nabla \ell_n(\theta)$. However, because the mixing of the MCMC sampling of p_θ is typically poorly understood and can vary widely with θ, the algorithm often fails, particularly in high-dimensional problems. Various tricks, such as the contrastive divergence (CD) scheme of Hinton (2002), or the persistent contrastive divergence of Tieleman (2008) are often employed to stabilize and accelerate the convergence of the MCMC sampling. However these methods are of limited use in overparametrized EBM.

For instance, CD corresponds to taking $\hat{p}_\theta = P_n K_\theta^L$ in Algorithm 1, where K_θ is a Markov kernel with invariant distribution p_θ, and K_θ^L is the kernel K_θ iterated L times[1]. To see why this is a sensible choice, suppose that the true data distribution

[1]In other words, $\hat{p}_\theta$ is the distribution obtained by drawing a data point from the empirical measure P_n, and using that data point as initial value for a Markov chain with kernel K_θ run for L iterations.

is $P_\star = P_{\theta_\star}$ for some $\theta_\star \in \Theta$, and suppose for the sake of the argument that X is bounded, and K_θ is a contraction on the space of probability measures on X equipped with the BL-metric:

$$\|p - q\|_{\mathsf{BL}} \stackrel{\text{def}}{=} \sup_{f:\, \|f\|_{\mathsf{BL}} \leq 1} \left| \int_{\mathsf{X}} f(x) p(\mathrm{d}x) - \int_{\mathsf{X}} f(x) q(\mathrm{d}x) \right|,$$

where $\|f\|_{\mathsf{BL}}$ is the sum of the infinity norm and Lipschitz norm of f. In that case, if ρ_θ denotes the contraction constant of K_θ, we have from (2) that

$$\|P_n K_\theta^L - p_\theta\|_{\mathsf{BL}} \leq \rho_\theta^L \|P_n - p_\theta\|_{\mathsf{BL}} \leq \rho_\theta^L \left(\|P_n - P_\star\|_{\mathsf{BL}} + \frac{\mu_{\mathsf{Leb}}(\mathsf{X})}{2} \|\mathcal{E}_\theta - \mathcal{E}_{\theta_\star}\|_\infty \right).$$

Since $X_i \overset{i.i.d.}{\sim} P_\star$, it is generally the case that $\|P_n - P_\star\|_{\mathsf{BL}}$ is small. Therefore, if the energy functions $\mathcal{E}_\theta$ do not vary too much, CD would work well, even when L is small. However, CD is typically of little help when applied to EBMs from deep neural networks, since for these models the terms $\|\mathcal{E}_\theta - \mathcal{E}_{\theta_\star}\|_\infty$ are typically very large quantities.

Algorithm 1. Let $\theta^{(0)} \in \mathbb{R}^d$ be the initial solution, and $\{\gamma^{(k)},\ k \geq 1\}$ a sequence of step-size. At iteration $k \geq 1$, given $\theta^{(k-1)}$:

(1) Draw $X_1^+, \ldots, X_B^+ \overset{i.i.d.}{\sim} P_n$, and draw random variables $X_1^-, \ldots, X_B^- \overset{i.i.d.}{\sim} \hat{p}_{\theta^{(k-1)}}$, where $\hat{p}_{\theta^{(k-1)}}$ is some approximation of $p_{\theta^{(k-1)}}$, typically based on MCMC. Compute

$$\widehat{\nabla \ell_n}(\theta^{(k-1)}) \stackrel{\text{def}}{=} \frac{1}{B} \sum_{i=1}^B G_{\theta^{(k-1)}}(X_i^+) - \frac{1}{B} \sum_{i=1}^B G_{\theta^{(k-1)}}(X_i^-). \tag{6}$$

(2) Compute

$$\theta^{(k)} = \mathrm{Prox}_{\gamma^{(k)}}^{\mathcal{R}} \left(\theta^{(k-1)} - \gamma^{(k)} \widehat{\nabla \ell_n}(\theta^{(k-1)}) \right).$$

Persistent CD improves on CD by taking $\hat{p}_{\theta^{(k-1)}}$ as $\nu^{(k)} K_{\theta^{(k-1)}}^L$, where the initial distribution $\nu^{(k)}$ is built from negative samples drawn over the previous few iterations. A similar analysis as above shows that persistent CD will also typically fail unless the step-size $\{\gamma^{(k)},\ k \geq 1\}$ are appropriately tuned, and small enough to keep the variations $\|\mathcal{E}_{\theta^{(k)}} - \mathcal{E}_{\theta^{(k-1)}}\|_\infty$ small, which leads to a much more costly algorithm. In conclusion, it is the case that fitting EBMs in large scale problems is often a formidable computational undertaking.

3. Generative EBMs

In this section we will assume for simplicity that the function $(x, \theta) \mapsto \mathcal{E}_\theta(x)$ is continuously differentiable. To address the computational challenges alluded to above, Nijkamp et al. (2019) introduced the idea of running a short, noise-initialized, and non-persistent Markov chain to approximate the score function. Specifically, take ν any probability distribution on $\mathbb{R}^p$, and $L \geq 1$ an integer. Typically, we take ν as the isotropic multivariate Gaussian distribution. Given θ, step-size sequences $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_L)$, $\boldsymbol{\sigma} = (\sigma_1, \ldots, \sigma_L)$ where $\rho_i > 0$, and $\sigma_i \geq 0$, generate a sequence $(X_0 \ldots, X_L)$ as follows. First we draw $X_0 \sim \nu$, and then draw

$$X_j = X_{j-1} - \rho_j \nabla_x \mathcal{E}_\theta(X_{j-1}) + \sigma_j Z_j, \quad j = 1, \ldots, L, \tag{7}$$

where $(Z_1, \ldots, Z_L)$ are iid random vectors with distribution $\mathbf{N}(0, I_p)$, and $\nabla_x \mathcal{E}_\theta(x)$ denotes the partial derivative with respect to x of the function $\mathcal{E}_\theta(x)$. Let π_θ denote the distribution of X_L where the randomness comes from the initial distribution ν, and from the noise $(Z_1, \ldots, Z_L)$ (when $\sigma_i > 0$). Clearly, π_θ depends also on the choice of ν, $\boldsymbol{\rho}, \boldsymbol{\sigma}$, and L. But we shall omit those dependencies in the notation. The short-run MCMC approximation of Nijkamp et al. (2019) consists in using Algorithm 1, with $\hat{p}_\theta$ chosen as π_θ. The resulting algorithm is presented in Algorithm 2. Because L is typically small and the initial distribution ν is taken as a noise-generating distribution, we note that Algorithm 2 is fundamentally different from Algorithm 1.

Our goal in this note is to shed some light on Algorithm 2. First, we find it important to decouple the modeling framework and the estimation framework of Nijkamp et al. (2019). On the modeling side, we argue that the short-run MCMC framework implicitly replaces the initial EBM $\{p_\theta,\ \theta \in \Theta\}$ by an approximation $\{\pi_\theta,\ \theta \in \Theta\}$ that we propose to call a generative EBM (GEBM). Furthermore, we offer the view that the GEBM is an instance of algorithm unrolling modeling. Algorithm unrolling is a general framework for constructing statistical models from optimization algorithms. We refer the reader to (Ongie et al. (2020); Shlezinger et al. (2021)) for thorough literature reviews. First we note that the EBM p_θ is the unique solution of the minimization problem

$$\min_\mu \left[\int \mathcal{E}_\theta(x) \mu(\mathrm{d}x) + \int \log\left(f_\mu(x)\right) f_\mu(x) \mathrm{d}x \right], \tag{8}$$

where f_μ is the density of μ, and the minimization is taken over all probability measure that are absolutely continuous with respect to $\mu_{\texttt{Leb}}$. Therefore we can construct more tractable densities that retain the features of p_θ by iterating an optimization algorithm for solving (8). With $\sigma_j = \sqrt{2\rho_j}$, the dynamics in (7) is precisely the discretization of the gradient flow for solving (8) (see e.g. Wibisono (2018)). Although not considered

here, the interest of this connection to algorithm unrolling is that more sophisticated optimization algorithms for (8) (for instance the underdamped Langevin dynamics) can be considered leading to GEBM with potentially different statistical properties.

Algorithm 2. Let $\theta^{(0)} \in \mathbb{R}^d$ be the initial solution, and $\{\gamma^{(k)},\ k \geq 1\}$ a sequence of step-size. At iteration $k \geq 1$, given $\theta^{(k-1)}$:

(1) Draw $X_1^+, \ldots, X_B^+ \overset{i.i.d.}{\sim} P_n$ if needed, and draw random variables $X_1^-, \ldots, X_B^- \overset{i.i.d.}{\sim} \pi_{\theta^{(k-1)}}$, where π_θ is as described in (7), and compute

$$\widehat{\Delta^{(k)}} = \frac{1}{B}\sum_{i=1}^{B} G_{\theta^{(k-1)}}(X_i^+) - \frac{1}{B}\sum_{i=1}^{B} G_{\theta^{(k-1)}}(X_i^-).$$

(2) Compute

$$\theta^{(k)} = \mathrm{Prox}^{\mathcal{R}}_{\gamma^{(k)}}\left(\theta^{(k-1)} - \gamma^{(k)}\widehat{\Delta^{(k)}}\right).$$

3.1. **Some basic properties of generative energy-based models.** In the last section, we have introduced the GEBM $\{\pi_\theta, \theta \in \Theta\}$ as a different density class that approximates $\{p_\theta,\ \theta \in \Theta\}$ but is easier to draw samples from. We argue in this section that Algorithm 2 consists in fitting the GEBM $\{\pi_\theta, \theta \in \Theta\}$ to data by maximum mean discrepancy minimization.

For two probability measures μ_1, μ_2 on X, and a family $\mathcal{F}$ of measurable real-valued functions on X, we define

$$\mathsf{d}_{\mathcal{F}}(\mu_1, \mu_2) \overset{\text{def}}{=} \sup_{f \in \mathcal{F}} \left| \int_{\mathsf{X}} f(x)\mu_1(\mathrm{d}x) - \int_{\mathsf{X}} f(x)\mu_2(\mathrm{d}x) \right|.$$

Given $\theta \in \Theta$, we recall that $G_\theta(x)$ denotes the gradient $\nabla_\theta \mathcal{E}_\theta(x)$. Let $G_{j,\theta}(x)$ denote the j-component of $G_\theta(x)$. In other words, $G_{j,\theta}(x) = \partial \mathcal{E}_\theta(x)/\partial\theta_j$. We define

$$\mathcal{G}_\theta \overset{\text{def}}{=} \left\{ \sum_{j=1}^{d} w_j G_{j,\theta},\ \ w \overset{\text{def}}{=} (w_1, \ldots, w_d) \in \mathbb{R}^d,\ \|w\|_2 \leq 1 \right\}.$$

And for $\theta \in \Theta$, we define

$$\mathsf{d}_\theta(\mu_1, \mu_2) \overset{\text{def}}{=} \mathsf{d}_{\mathcal{G}_\theta}(\mu_1, \mu_2).$$

Given $f(\cdot) = \sum_{j=1}^d \alpha_j G_{j,\theta}(\cdot)$, and $g(\cdot) = \sum_{j=1}^d \beta_j G_{j,\theta}(\cdot)$ two elements of $\mathcal{G}_\theta$, we define their inner product as

$$\langle f, g \rangle_\theta \overset{\text{def}}{=} \sum_{j=1}^{d} \alpha_j \beta_j.$$

$\mathcal{G}_\theta$ equipped with the inner product $\langle \cdot, \cdot \rangle_\theta$ is (the unit ball of) a reproducing kernel Hilbert space with reproducing kernel

$$\mathcal{K}_\theta(x, x') \stackrel{\text{def}}{=} \sum_{j=1}^{d} G_{j,\theta}(x) G_{j,\theta}(x').$$

We refer the reader to (Wainwright (2019)) for more details on reproducing kernel Hilbert spaces. In the deep learning literature, the kernel $\mathcal{K}_\theta$ is known as the neural tangent kernel of the model (Jacot et al. (2018)). It is easily seen that

$$\mathsf{d}_\theta(\mu_1, \mu_2)^2 = \int \mathcal{K}_\theta(x, y) \mu_1(\mathrm{d}x) \mu_1(\mathrm{d}y) = \|\mu_1(G_\theta) - \mu_2(G_\theta)\|_2^2.$$

These observations on reproducing kernel Hilbert spaces are well-known, and hold in broader generality (see for instance (Gretton et al. (2012))). In what follows we write $\|A\|_{\mathsf{op}}$ to denote the spectral norm of A. And for $\theta \in \Theta$, we set

$$H_\theta(x) \stackrel{\text{def}}{=} \nabla_\theta \log \pi_\theta(x),\ x \in \mathsf{X}.$$

We make the following standard boundedness and Lipschitz smoothness assumption.

H 1. *For all $\theta_0 \in \Theta$, the function $\theta \mapsto \int_{\mathsf{X}} G_{\theta_0}(x) \pi_\theta(x) \mathrm{d}x$ is twice continuously differentiable under the integral and there exists $L < \infty$ that may depend on θ_0 such that for all $\theta \in \Theta$,*

$$\int_{\mathsf{X}} \left[\|G_{\theta_0}(x)\|_2^2 + \|H_\theta(x)\|_2^2 \right] \pi_\theta(\mathrm{d}x) \le L,$$

and for all $\theta, \phi \in \Theta$,

$$\left\| \int_{\mathsf{X}} G_{\theta_0}(x) \left[H_\theta(x)^T \pi_\theta(x) - H_\phi(x)^T \pi_\phi(x) \right] \mathrm{d}x \right\|_{\mathsf{op}} \le L \|\theta - \phi\|_2.$$

We will also impose the following standard condition on the step-size sequence.

H 2. *There exists $c_0 > 0$ such that the sequence $\{\gamma^{(k)},\ k \ge 1\}$ satisfies*

$$0 < \gamma^{(k)} \le c_0, \quad \sum_{k \ge 1} \gamma^{(k)} = \infty, \quad \text{and} \quad \sum_{k \ge 1} \left(\gamma^{(k)} \right)^2 < \infty.$$

The following result is a statement on the limiting behavior of the sequence $\{\theta^{(k)},\ k \ge 1\}$ produced by Algorithm 2. For simplicity, we focus on the case $\mathcal{R} \equiv 0$.

Theorem 1. *Assume that H1 and H2 hold for some constant c_0 small enough. Let $\{\theta^{(k)},\ k \ge 0\}$ denotes the sequence generated by Algorithm 2, with $\mathcal{R} \equiv 0$. Let $G_0 \stackrel{\text{def}}{=} G_{\theta^{(0)}}$. Suppose that there exists $\mu > 0$ such that for all $v \in \mathbb{R}^d$, and for all $k \ge 1$,*

$$v^T \left(\int_{\mathsf{X}} H_{\theta^{(k)}}(x) \left\{ G_0(x) \right\}^T \pi_{\theta^{(k)}}(x) \mathrm{d}x \right) v \ge \mu \|v\|_2^2, \tag{9}$$

and

$$\mathcal{K}_{\theta^{(k)}} = \mathcal{K}_{\theta^{(0)}}. \tag{10}$$

Then

$$\lim_{k\to\infty} \mathbb{E}\left[\mathsf{d}_{\theta^{(0)}}(\pi_{\theta^{(k)}}, P_n) \mid \theta^{(0)}\right] = 0.$$

Proof. See Section 7.1. □

To motivate assumption (9), consider the ideal case where $\pi_\theta(x) = p_\theta(x) = e^{-\langle\theta, G(x)\rangle}/Z_\theta$. In that case $G_0 = G$, $H_\theta = G - \pi_\theta(G)$, and

$$\int_{\mathsf{X}} H_\theta(x) G_0(x)^{\intercal} \pi_\theta(x) \mathrm{d}x = \int_{\mathsf{X}} \left(G(x) - \pi_\theta(G)\right) \left(G(x) - \pi_\theta(G)\right)^{\intercal} \pi_\theta(\mathrm{d}x).$$

Hence in this particular case, (9) corresponds to the assumption that the covariance matrices in the last display remains positive definite during the algorithm, which often holds for graphical models. In the general case, (9) is a measure of covariation between G_0 and H_θ. Its positive definiteness is admittedly a very difficult assumption to check when it comes to deep learning models. Nevertheless, although much remains to be learned about deep learning models, the currently emerging understanding supports (9). Indeed, in these highly over-parameterized deep learning models, due to the wealth of local solutions, the parameter θ needs not vary much during training. Therefore, one can argue that (9) is equivalent to the positive definiteness of

$$\left\{\nabla_\theta \log \pi_{\theta^{(0)}}(x)\right\} \left\{\nabla_\theta \mathcal{E}_{\theta^{(0)}}(x)\right\}^{\intercal},$$

which is a more plausible assumption.

We assume in (10) that the neural tangent kernels remain stable during the algorithm. Clearly, that assumption holds if G_θ does not depend on θ, as for instance with many graphical models. It has been observed recently that for deep and wide neural network functions, the neural tangent kernel is indeed remarkably stable during training (Du et al. (2019); Bietti and Mairal (2019)). Hence for these types of deep neural networks, assumption (10) also seems reasonable.

Assuming the initial solution $\theta^{(0)}$ is non-random, the main implication of the theorem is that the short-run MCMC algorithm converges to the minimum distance estimator

$$\hat{\theta} \stackrel{\text{def}}{=} \operatorname*{Argmin}_{\theta\in\Theta} \mathsf{d}_{\theta^{(0)}}(\pi_\theta, P_n), \tag{11}$$

as anticipated by Nijkamp et al. (2019). This estimator is closely related to the maximum mean discrepancy GAN (MMD-GAN) estimators Li et al. (2017); Binkowski et al. (2018). More specifically, let $\mathcal{K}_\phi : \mathsf{X} \times \mathsf{X} \to \mathbb{R}$, $\phi \in \Phi$, be a family of kernels,

and let $\mathcal{F}_\phi$ denote the unit ball of the reproducing kernel Hilbert space associated to $\mathcal{K}_\phi$. The MMD-GAN estimator of θ is define as

$$\underset{\theta\in\Theta}{\mathsf{Argmin}}\ \max_{\phi\in\Theta} \mathsf{d}_{\mathcal{F}_\phi}(\pi_\theta, P_n).$$

Hence the difference between the short run MCMC and MMD-GAN is that the former aims at a specific MMD kernel, whereas the latter search for the best kernel in a given set. Although conceptually more flexible, the MMD-GAN leads to challenging min-max optimization problems that are challenging to solve.

We now turn to the statistical properties of $\hat{\theta}$. We recall that the sub-exponential norm of a random variable Z is defined as

$$\|Z\|_{\psi_1} \overset{\text{def}}{=} \inf\left\{t>0,\ \mathbb{E}\left(e^{\frac{|X|}{t}}\right)\le 2\right\}.$$

Proposition 2. *Assume H1, and suppose that the initial value $\theta^{(0)}$ in Algorithm 2 is non-random, and*

$$\int_{\mathsf{X}} \|G_{\theta^{(0)}}(x)\|_\infty P_\star(\mathrm{d}x) < \infty,$$

$$\textit{and}\quad \max_{1\le j\le p} \|G_{j,\theta^{(0)}}(X) - \mathbb{E}(G_{j,\theta^{(0)}}(X))\|_{\psi_1} \le K, \quad (12)$$

for some constant $K<\infty$. Then with probability at least $1-2/d$, the estimator $\hat{\theta}$ as defined in (11) satisfies

$$\mathsf{d}_{\theta^{(0)}}(\pi_{\hat{\theta}}, P_\star) \le \min_{\theta\in\Theta}\ \mathsf{d}_{\theta^{(0)}}(\pi_\theta, P_\star) + c_1 K\sqrt{\frac{d\log(d)}{n}}.$$

Proof. See Section 7.2. □

The term $(\min_{\theta\in\Theta}\ \mathsf{d}_{\theta^{(0)}}(\pi_\theta, P_\star))$ is the model approximation error, whereas the second term $(K\sqrt{\frac{d\log(d)}{n}})$ is the statistical error. We note however that the sub-exponential norm K may depend unfavorably on the dimension. Bounding the model error $\min_{\theta\in\Theta}\ \mathsf{d}_{\theta^{(0)}}(\pi_\theta, P_\star)$ is a more challenging problem that we leave for possible future research.

3.2. **Implementation using stochastic proximal gradient ADAM.** In some problems, selecting the correct step-size sequence $\{\gamma^{(k)},\ k\ge 1\}$ in Algorithm 2 can be challenging. In many of these cases, the use of an adaptive momentum (ADAM) yields better behaving solvers. Algorithm 3 below gives a synoptic view of the stochastic proximal gradient algorithm with ADAM. The algorithm requires adaptation parameters β_1, β_2 which are typically set to $\beta_1 = 0.9$, and $\beta_2 = 0.9999$, and a tolerance parameter ϵ typically taken as 10^{-8}. In the algorithm, $\oplus, \otimes$ and $\oslash$ denote component-wise addition, multiplication and division, respectively.

We should add that although ADAM has become the de facto standard method for deep learning optimization, its theoretical properties remain poorly understood. We refer to Bock and Weib (2019) for some local convergence results.

Algorithm 3. Let $\theta^{(0)} \in \mathbb{R}^d$ be the initial solution, and $\{\gamma^{(k)},\ k \geq 1\}$ a sequence of step-size. Set $m^{(0)} = v^{(0)} = \mathbf{0}_d$. At iteration $k \geq 1$, given $\theta^{(k-1)}$, $m^{(k-1)}$, $v^{(k-1)}$:

(1) Draw $X_1^+, \ldots, X_B^+ \overset{i.i.d.}{\sim} P_n$ if needed, and draw random variables $X_1^-, \ldots, X_B^- \overset{i.i.d.}{\sim} \pi_{\theta^{(k-1)}}$, where π_θ is as described in (7), and compute

$$\widehat{\Delta^{(k)}} = \frac{1}{B}\sum_{i=1}^{B} G_{\theta^{(k-1)}}(X_i^+) - \frac{1}{B}\sum_{i=1}^{B} G_{\theta^{(k-1)}}(X_i^-).$$

(2) Compute

$$m^{(k)} = \beta_1 m^{(k-1)} + (1-\beta_1)\widehat{\Delta^{(k)}},$$
$$v^{(k)} = \beta_2 v^{(k-1)} + (1-\beta_2)\widehat{\Delta^{(k)}} \otimes \widehat{\Delta^{(k)}},$$
$$\bar{\theta} = \theta^{(k-1)} - \gamma^{(k)} \frac{\sqrt{1-\beta_2^k}}{(1-\beta_1^k)} m^{(k)} \oslash \sqrt{v^{(k)} \oplus \epsilon}.$$
$$\theta^{(k)} = \mathrm{Prox}^{\mathcal{R}}_{\gamma^{(k)}}(\bar{\theta}).$$

4. An illustration with Gaussian graphical models

Since graphical models are examples of energy-based models, we show here that the generative energy-based modeling developed above leads to a novel method for fitting graphical models. We focus on Gaussian graphical models, but extensions beyond is straightforward. We consider the problem of estimating a precision matrix $\theta \in \mathbb{R}^{p\times p}$ from a data set $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^p$, where $\mathbf{x}_i \overset{\text{i.i.d.}}{\sim} \mathbf{N}(0, \theta^{-1})$ for $i = 1, \ldots, n$. One of the most popular such regularization method is the graphical Lasso (GLASSO) Yuan and Lin (2007); Friedman et al. (2007), where we estimate θ under the assumption that it is sparse. This leads to:

$$\underset{\theta\in\mathcal{M}_+}{\text{minimize}} \quad -\log\det\theta + \mathsf{Tr}(\theta S) + \lambda\sum_{i,j}|\theta_{ij}|, \tag{13}$$

where, $S = \frac{1}{n}\sum_{i=1}^n \mathbf{x}_i\mathbf{x}_i'$ is the sample covariance matrix, $\mathcal{M}_+$ denotes the set of positive definite matrices and $\lambda > 0$ is a tuning parameter that controls the degree of regularization. So long as $\lambda > 0$, Problem (13) admits a unique minimizer. The problem of computing (13) has generated an impressive literature (Mazumder and Hastie (2012); Rolfs et al. (2012); Yuan (2012); Li and Toh (2010); Hsieh et al. (2014). All these existing algorithms have a cost per-iteration of at least $O(p^3)$, possibly larger

for some of these algorithms. Hence when p is large, fitting a Gaussian graphical model using GLASSO is typically costly.

The density of the multivariate distribution $\mathbf{N}(0, \theta^{-1})$ can be written as an energy-based model of the form

$$p_\theta(\mathbf{x}) = \frac{e^{-\mathcal{E}_\theta(\mathbf{x})}}{\int_{\mathbb{R}^p} e^{-\mathcal{E}_\theta(\mathbf{x})} \mathrm{d}\mathbf{x}}, \quad \text{where} \quad \mathcal{E}_\theta(\mathbf{x}) = \frac{1}{2}\mathbf{x}'\theta\mathbf{x}$$

The corresponding generative EBM is then as follows. Given θ, and a step-size $\rho > 0$, we first draw $X_0 \sim \mathbf{N}(0, \mathbf{I}_p)$, and define the sequence $(X_1 \ldots, X_L)$ generated as follows.

$$X_j = (\mathbf{I}_p - \rho\theta)\, X_{j-1} + \sqrt{\rho} Z_j, \quad j = 1, \ldots, L, \tag{14}$$

where $Z_1, \ldots, Z_L \overset{i.i.d.}{\sim} \mathbf{N}(0, \mathbf{I}_p)$. We take π_θ as the distribution of X_L. It is easy seen that in this case $\pi_\theta = \mathbf{N}(0, \Sigma_L(\theta))$, where

$$\Sigma_L(\theta) = \theta^{-1} + (\mathbf{I}_p - \rho\theta)^{2L} \left(\mathbf{I}_p - \theta^{-1}\right).$$

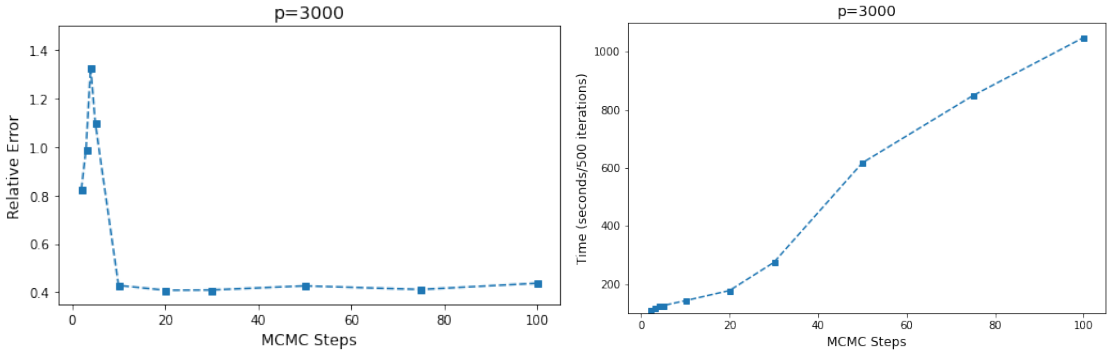
Algorithm 2 thus readily applies. Here the proximal operator is the usual lasso soft-thresholding operator (with threshold $\gamma\lambda$) applied componentwise. Note that iterating equation (14) L times to generate each X_i^- is done at the computational cost of $O(Lp^2)$. Hence for this example, each iteration of Algorithm 2 has a computational cost of $O(B(L+1)p^2)$. When $B \times L$ is small compared to p, each iteration of Algorithm 2 is cheaper than the typical $O(p^3)$ needed by classical GLASSO solvers.

4.1. **Numerical Experiments.** We illustrate the practical merit of Algorithm 2 on some synthesis dataset. We test the algorithm with $p \in \{3000, 5000\}$, and $n = p/2$. We then solve the graphical lasso problem (13) with $\lambda = 0.03$. The true precision matrix $\theta_\star$ is generated as follows. First we generate a symmetric sparse matrix M such that the proportion of non-zeros entries is $5/p$. We magnified the signal by adding 2 to all the non-zeros entries of M, and subtracting 2 for negative non-zero entries. Then we set $\theta_\star = M + (1 - \lambda_{\mathsf{min}}(M))\, I_p$, where $\lambda_{\mathsf{min}}(A)$ is the smallest eigenvalue of A.

As initial solution we use the diagonal matrix obtained by taking the inverse sample variances. And we run Algorithm 2 with a constant step-size $\gamma = 8$, and a mini-batch size $B = 30$. The step-size for generating the negative samples using (7) was set to $\rho = 0.008$. We run Algorithm 2 for 500 iterations, and compute:

$$\mathrm{error}(\theta) = \left\|\theta - \hat{\theta}\right\|_F / \|\hat{\theta}\|_F, \qquad \text{and} \quad \mathrm{sp}(\theta) = \frac{\|\theta\|_0}{p}.$$

Figure 1 shows the relative error, and the computational time as the number of MCMC steps L increases, when $p = 3000$. The result shows excellent recovery of $\hat{\theta}$ for L as small as 10, with little improvement for $L > 10$.



(A) (B)

FIGURE 1. Relative error, and computational time versus L, the number of MCMC steps.

We also compare the performance of the short run MCMC with the deterministic proximal gradient (PG) algorithm of Rolfs et al. (2012). In this comparison we set $L = 10$, and $B = 30$. We repeat the experiments 30 times and take average values for the relative error, sparsity, and computational time. The relative error, computational time, and sparsity are presented in table 1. The results show that the short run MCMC is roughly $p/(3 * B * L) \approx 5$ faster than the deterministic proximal gradient algorithm at the cost of a small loss of accuracy.

	error	**Running times** (500 iterations)	**sp**
Algo 2	0.401	350 seconds	5.58
PG	0.387	1728 seconds	7.20

TABLE 1. Comparison of Algo 2 with the deterministic proximal gradient algorithm (PG) of Rolfs et al. (2012). $p = 5000$.

5. Illustration with density estimation for image data

Statistical models from machine learning have produced several breakthrough in high-dimensional density estimation over the last decade, with modeling framework such as generative adversarial networks (Goodfellow et al., 2014). In this section we reproduce the results of Nijkamp et al. (2019) showing that the generative EBMs yield results with comparable performance.

(a) (b)

FIGURE 2. Generated MNIST and CelebA images with Uniform Noise Initial Distribution

Given a dataset $X_1, \ldots, X_n$, where each X_i is an image vectorized into a d-dimensional vector, we view the data as i.i.d. realizations of a generative EBM built as in (7), where the energy function $\mathcal{E}_\theta$ is a deep neural network with the same architecture as in Nijkamp et al. (2019). Specifically, we use a 5-layer CNN architecture where each convolutional layer is followed by a LeakyReLU activation function. We refer the reader to Section 8 for a detailed description of the architecture. We aim to estimate the parameter θ of the model, and evaluate the performance of the model to generate natural-looking images. We illustrate the model with two standard machine learning datasets: CIFAR10 and CelebA. The CIFAR-10 dataset is a large dataset consists of 60,000 32×32 color images in 10 classes, with 6000 per class. There are 50,000 training images and 10,000 test images. CelebFaces Attributes (CelebA) dataset is a large-scale attributes dataset with more than 200,000 celebrity images. The images in this dataset cover large pose variation and background.

We fit the model both with and without a lasso regularization, where the regularization parameter is set to $\lambda = 0.1$. In both models, we generate negative samples from the generative EBM using $\rho = 1$, and $\sigma = 0.0$, and L taken between 25 and 100. To fit the models, in both cases we use Algorithm 3, with batch-size $B = 64$, and an initial step-size $\gamma = 10^{-4}$. with step decay 0.8 every 100 iterations. A random selection of generated images using the estimated densities from CIFAR-10 and CelebA are presented in Figure 2, respectively.

We demonstrate the fidelity of generated images by using Inception Score (IS; higher values are preferred) (Barratt and Sharma, 2018) and Frechet Inception Distance (FID; lower values are preferred) (Heusel et al., 2017). Table 2 present the FID and IS with different number of MCMC steps. The results also include a comparison with DCGAN (Goodfellow et al. (2014)). These empirical results show that the generative EBM, together with the moment matching estimation procedure of Nijkamp et al. (2019) produce results that match the performance of DCGAN. In this example, and unlike the Gaussian graphical example, we did not find any clear advantage to adding a ℓ^1-regularization term. The penalization produced somewhat sparse models, but with notably degraded performances for larger values of L.

TABLE 2. Influence of short-run steps on performance for CIFAR10 and CelebA

Data	L	Without penalty		With Penalty			DCGAN	
		FID	IS	FID	IS	Sparsity	FID	IS
CIFAR-10	25	222	1.49	238	1.45	0.703	203	2.77
	50	187	3.13	204	2.05	0.585		
	75	187	3.34	198	1.95	0.541		
	100	173	2.76	193	1.90	0.581		
CelebA	25	320	1.64	259	2.42	0.640	117	2.53
	50	319	1.58	235	2.51	0.656		
	75	292	2.23	184	2.13	0.645		
	100	158	1.96	174	2.06	0.591		

6. FURTHER DISCUSSION

We have shown in this work that the short run MCMC methodology of Nijkamp et al. (2019) consists in replacing the initial EBM by a generative EBM that is then estimated by minimum MMD estimation, where the MMD kernel is taken as the neural tangent kernel of the deep neural network function. Importantly, the model is an example of algorithm unrolling modeling, and can be applied more broadly. Our numerical illustration shows that the method can substantially reduce the estimation time of high-dimensional Gaussian graphical models. And the numerical illustration

with the image density estimation problem reproduces the results of Nijkamp et al. (2019), and show that the method compares favorably with DCGAN.

7. Proofs

7.1. **Proof of Theorem 1.**

Proof. Throughout the proof $c_1, c_2, \cdots$ denote some constants that depend on the dimension d, and the sample size n, but not on the iteration index k. Let us set

$$G_0 \stackrel{\text{def}}{=} G_{\theta^{(0)}} = \nabla_\theta \mathcal{E}_\theta, \quad \mathsf{d}_0 \stackrel{\text{def}}{=} \mathsf{d}_{\theta^{(0)}},$$

$$\Delta_n(\theta) \stackrel{\text{def}}{=} \int_{\mathsf{X}} G_\theta(x) P_n(\mathrm{d}x) - \int_{\mathsf{X}} G_\theta(x) \pi_\theta(\mathrm{d}x), \ \ \theta \in \Theta, \tag{15}$$

and

$$D^{(k)} \stackrel{\text{def}}{=} \mathsf{d}_0(\pi_{\theta^{(k)}}, P_n)^2 = \|\pi_{\theta^{(k)}}(G_0) - P_n(G_0)\|_2^2.$$

We can write

$$D^{(k)} = D^{(k-1)} + \|\pi_{\theta^{(k)}}(G_0) - \pi_{\theta^{(k-1)}}(G_0)\|_2^2 + 2\left\langle \pi_{\theta^{(k)}}(G_0) - \pi_{\theta^{(k-1)}}(G_0), \pi_{\theta^{(k-1)}}(G_0) - P_n(G_0) \right\rangle. \tag{16}$$

We first work on the term $\|\pi_{\theta^{(k)}}(G_0) - \pi_{\theta^{(k-1)}}(G_0)\|_2^2$. Under Assumption 2, we can interchange derivation and integral, so that

$$M_\theta \stackrel{\text{def}}{=} \nabla_\theta \int_{\mathsf{X}} G_0(x) \pi_\theta(x) \mathrm{d}x = \int_{\mathsf{X}} G_0(x) \left\{\nabla_\theta \log \pi_\theta(x)\right\}^{\intercal} \pi_\theta(x) \mathrm{d}x.$$

A first order Taylor expansion then implies that we can find $\bar{\theta}^{(k)} \in \Theta$ such that

$$\pi_{\theta^{(k)}}(G_0) = \pi_{\theta^{(k-1)}}(G_0) + M_{\bar{\theta}^{(k)}}(\theta^{(k)} - \theta^{(k-1)}) = \pi_{\theta^{(k-1)}}(G_0) - \gamma^{(k)} M_{\bar{\theta}^{(k)}} \left(\Delta_n(\theta^{(k-1)}) + \eta^{(k)}\right),$$

where

$$\eta^{(k)} = \left[\frac{1}{B}\sum_{i=1}^{B} G_{\theta^{(k-1)}}(X_i^+) - \int_{\mathsf{X}} G_{\theta^{(k-1)}}(x) P_n(\mathrm{d}x)\right] - \left[\frac{1}{B}\sum_{i=1}^{B} G_{\theta^{(k-1)}}(X_i^-) - \int_{\mathsf{X}} G_{\theta^{(k-1)}}(x) \pi_{\theta^{(k-1)}}(\mathrm{d}x)\right].$$

Hence, using (9) in Assumption 2, we can conclude that

$$\|\pi_{\theta^{(k)}}(G_0) - \pi_{\theta^{(k-1)}}(G_0)\|_2^2 \le L_k^2 \|\Delta_n(\theta^{(k-1)}) + \eta^{(k)}\|_2^2 \le 2L_k^2 \|\Delta_n(\theta^{(k-1)})\|_2^2 + 2L_k^2 \|\eta^{(k)}\|_2^2,$$

where $L_k = L\gamma^{(k)}$. Taking the conditional expectation given $\theta^{(k-1)}$, we have

$$\mathbb{E}\left(\|\eta^{(k)}\|_2^2 \mid \theta^{(k-1)}\right) \leq \frac{1}{B}\int_{\mathsf{X}} \|G_{\theta^{(k-1)}}(x)\|_2^2 P_n(\mathrm{d}x) + \frac{1}{B}\int_{\mathsf{X}} \|G_{\theta^{(k-1)}}(x)\|_2^2 \pi_{\theta^{(k-1)}}(\mathrm{d}x) \leq \frac{c_1}{B},$$

for some constant c_1. Using Assumption 1,

$$D^{(k-1)} = \mathsf{d}_{\theta^{(k-1)}}(\pi_{\theta^{(k-1)}}, P_n)^2 = \|\Delta_n(\theta^{(k-1)})\|_2^2.$$

The last term of on right-hand side of (16) can be written as

$$\begin{aligned}\langle \pi_{\theta^{(k)}}(G_0) - \pi_{\theta^{(k-1)}}(G_0), \pi_{\theta^{(k-1)}}(G_0) - P_n(G_0)\rangle \\ = \left(\theta^{(k)} - \theta^{(k-1)}\right)^{\intercal} M_{\theta^{(k-1)}}^{\intercal} \left(\pi_{\theta^{(k-1)}}(G_0) - P_n(G_0)\right) \\ + \left(\theta^{(k)} - \theta^{(k-1)}\right)^{\intercal} [M_{\bar{\theta}^{(k)}} - M_{\theta^{(k-1)}}]^{\intercal} \\ \times \left(\pi_{\theta^{(k-1)}}(G_0) - P_n(G_0)\right).\end{aligned}$$

The conditional expectation given $\theta^{(k-1)}$ of the first term on the right-hand side of the last display is

$$\begin{aligned}-\gamma^{(k)} \left(\pi_{\theta^{(k-1)}}(G_0) - P_n(G_0)\right)^{\intercal} M_{\theta^{(k-1)}}^{\intercal} \\ \times \left(\pi_{\theta^{(k-1)}}(G_0) - P_n(G_0)\right) \leq -\mu\gamma^{(k)} D^{(k-1)},\end{aligned}$$

where we use (9). By Assumption 2,

$$\begin{aligned}\left|\left(\theta^{(k)} - \theta^{(k-1)}\right)^{\intercal} [M_{\bar{\theta}^{(k)}} - M_{\theta^{(k-1)}}]^{\intercal} \times \left(\pi_{\theta^{(k-1)}}(G_0) - P_n(G_0)\right)\right| \\ \leq c_2 L \|\theta^{(k)} - \theta^{(k-1)}\|_2^2 \leq 2c_2 L\gamma_k^2 \left\|\Delta_n(\theta^{(k-1)}) + \eta^{(k)}\right\|_2^2.\end{aligned}$$

Therefore, taking the conditional expectation given $\theta^{(k-1)}$ on both sides of (16) yields

$$\mathbb{E}\left(D^{(k)} \mid \theta^{(k-1)}\right) \leq \left(1 - 2\mu\gamma^{(k)} + c_3 L_k^2\right) D^{(k-1)} + \frac{c_4 L_k^2}{B}, \tag{17}$$

and taking the expectation on both sides of the last display yields

$$\mathbb{E}\left(D^{(k)}\right) \leq \left(1 - 2\mu\gamma^{(k)} + c_3 L_k^2\right) \mathbb{E}\left(D^{(k-1)}\right) + \frac{c_4 L_k^2}{B}. \tag{18}$$

We conclude that if $\{\gamma^{(k)},\ k \geq 1\}$ is such that for all $k \geq 1$ $\mu\gamma^{(k)} \geq c_3 L_k^2$, and $\sum_k (\gamma^{(k)})^2 < \infty$, then

$$\mathbb{E}\left(D^{(k)}\right) \leq \left(1 - \mu\gamma^{(k)}\right) \mathbb{E}\left(D^{(k-1)}\right) + \frac{c_4 L_k^2}{B},$$

which implies that

$$\mu \sum_{k\geq 1} \gamma^{(k)} \mathbb{E}\left(D^{(k-1)}\right) \leq \mathbb{E}(D^{(0)}) + \sum_{k\geq 1} \frac{c_4 L_k^2}{B} < \infty.$$

Consequently, since $\sum_k \gamma^{(k)} = +\infty$, we must have $\lim_k \mathbb{E}\left(D^{(k-1)}\right) = 0$. Hence the theorem. □

7.2. **Proof of Proposition 2.** For all $\theta \in \Theta$, we have

$$\mathsf{d}_{\theta^{(0)}}(\pi_\theta, P_\star) \leq \mathsf{d}_{\theta^{(0)}}(\pi_\theta, P_n) + \mathsf{d}_{\theta^{(0)}}(P_n, P_\star).$$

From the last display, we have

$$\mathsf{d}_{\theta^{(0)}}(\pi_{\hat{\theta}}, P_\star) \leq \mathsf{d}_{\theta^{(0)}}(\pi_{\hat{\theta}}, P_n) + \mathsf{d}_{\theta^{(0)}}(P_n, P_\star).$$

By the definition of $\hat{\theta}$, for any $\theta \in \Theta$, we have

$$\mathsf{d}_{\theta^{(0)}}(\pi_{\hat{\theta}}, P_n) \leq \mathsf{d}_{\theta^{(0)}}(\pi_\theta, P_n) \leq \mathsf{d}_{\theta^{(0)}}(\pi_\theta, P_\star) + \mathsf{d}_{\theta^{(0)}}(P_n, P_\star).$$

Combining these two inequalities yields

$$\mathsf{d}_{\theta^{(0)}}(\pi_{\hat{\theta}}, P_\star) \leq \min_{\theta\in\Theta} \mathsf{d}_{\theta^{(0)}}(\pi_\theta, P_\star) + 2\mathsf{d}_{\theta^{(0)}}(P_n, P_\star)$$

Using the sub-exponential assumption in (12), by Bernstein's inequality (Vershynin (2018)), there exists an absolute constant $c_0 > 0$ such that for $t = c_1 K\sqrt{\frac{2\log(d)}{n}}$, for some absolute constant c_1,

$$\mathbb{P}\left(\left\|\int_{\mathbb{R}^p} G_{\theta^{(0)}}(x) P_\star(\mathrm{d}x) - \int_{\mathbb{R}^p} G_{\theta^{(0)}}(x) P_n(\mathrm{d}x)\right\|_\infty > t\right) \leq 2d\exp\left(-c_0 \min\left(\frac{n^2t^2}{nK^2}, \frac{nt}{K}\right)\right) \leq \frac{2}{d}.$$

But since

$$\mathsf{d}_{\theta^{(0)}}(P_n, P_\star) \leq \sqrt{d}\left\|\int_{\mathbb{R}^p} G_{\theta^{(0)}}(x) P_\star(\mathrm{d}x) - \int_{\mathbb{R}^p} G_{\theta^{(0)}}(x) P_n(\mathrm{d}x)\right\|_\infty,$$

we conclude that with probability at least $1 - 2/d$,

$$\mathsf{d}_{\theta^{(0)}}(P_n, P_\star) \leq c_1 K\sqrt{\frac{d\log(d)}{n}}.$$

Hence the result.

8. Appendix

Table 3. CNN Model Architecture

Layers	In-Out Size	Stride
Input	$32 \times 32 \times 3$	
3×3 conv(64), LeakyReLU	$32 \times 32 \times 64$	1
4×4 conv(128), LeakyReLU	$16 \times 16 \times 128$	2
4×4 conv(256), LeakyReLU	$8 \times 8 \times 256$	2
4×4 conv(512), LeakyReLU	$4 \times 4 \times 512$	2
4×4 conv(1)	$1 \times 1 \times 1$	1

References

Ackley, D. H., Hinton, G. E. and Sejnowski, T. J. (1985). *Cognitive Science* **9** 147–169.

Atchadé, Y. F., Fort, G. and Moulines, E. (2017). On perturbed proximal gradient algorithms. *The Journal of Machine Learning Research* **18** 310–342.

Barratt, S. T. and Sharma, R. (2018). A note on the inception score. *ArXiv* **abs/1801.01973**.

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *J. Roy. Statist. Soc. Ser. B* **36** 192–236. With discussion by D. R. Cox, A. G. Hawkes, P. Clifford, P. Whittle, K. Ord, R. Mead, J. M. Hammersley, and M. S. Bartlett and with a reply by the author.

Bietti, A. and Mairal, J. (2019). On the inductive bias of neural tangent kernels. In *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox and R. Garnett, eds.), vol. 32. Curran Associates, Inc.

Binkowski, M., Sutherland, D. J., Arbel, M. and Gretton, A. (2018). Demystifying MMD GANs. In *International Conference on Learning Representations*.

Bock, S. and Weib, M. (2019). A proof of local convergence for the adam optimizer. In *IJCNN*. IEEE.

Combettes, P. and Wajs, V. (2005). Signal recovery by proximal forward-backward splitting. *Multiscale Modeling and Simulation* **4** 1168–1200.

DeVore, R., Hanin, B. and Petrova, G. (2021). Neural network approximation. *Acta Numerica* **30** 327–444.

Du, S. S., Zhai, X., Póczos, B. and Singh, A. (2019). Gradient descent provably optimizes over-parameterized neural networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

Du, Y., Meier, J., Ma, J., Fergus, R. and Rives, A. (2020). Energy-based models for atomic-resolution protein conformations. In *International Conference on Learning Representations*.

Du, Y. and Mordatch, I. (2019). Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems* (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alche Buc, E. Fox and R. Garnett, eds.), vol. 32. Curran Associates, Inc.

Friedman, J., Hastie, T. and Tibshirani, R. (2007). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics* **9** 432–441.

Georgii, H.-O. (1988). *Gibbs measures and phase transitions*, vol. 9 of *de Gruyter Studies in Mathematics*. Walter de Gruyter & Co., Berlin.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014). Generative adversarial networks.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B. and Smola, A. (2012). A kernel two-sample test. *J. Mach. Learn. Res.* **13** 723–773.

Guyon, X. (1995). *Random fields on a network*. Probability and its Applications (New York), Springer-Verlag, New York. Modeling, statistics, and applications, Translated from the 1992 French original by Carenne Ludeña.

Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B. and Hochreiter, S. (2017). Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *NIPS*.

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

Hinton, G. E. and Sejnowski, T. J. (1983). Optimal perceptual inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Hsieh, C.-J., Sustik, M. A., Dhillon, I. S. and Ravikumar, P. (2014). Quic: Quadratic approximation for sparse inverse covariance estimation. *Journal of Machine Learning Research* **15** 2911–2947.

Ingraham, J., Riesselman, A. J., Sander, C. and Marks, D. S. (2019). Learning protein structure with a differentiable simulator. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*.

OpenReview.net.

Ising, E. (1925). Beitrag zur Theorie des Ferromagnetismus. *Zeitschrift fur Physik* **31** 253–258.

Jacot, A., Gabriel, F. and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett, eds.), vol. 31. Curran Associates, Inc.

LeCun, Y., Chopra, S., Hadsell, R., Huang, F. J. and et al. (2006). A tutorial on energy-based learning. In *PREDICTING STRUCTURED DATA*. MIT Press.

Li, C.-L., Chang, W.-C., Cheng, Y., Yang, Y. and Póczos, B. (2017). Mmd gan: Towards deeper understanding of moment matching network. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS'17.

Li, L. and Toh, K.-C. (2010). An inexact interior point method for l1-regularized sparse covariance selection. *Mathematical Programming Computation* **31** 2000–2016.

Mazumder, R. and Hastie, T. (2012). The graphical lasso: New insights and alternatives. *Electronic Journal of Statistics* **6** 2125–2149.

Nijkamp, E., Hill, M., Zhu, S.-C. and Wu, Y. N. (2019). *Learning Non-Convergent Non-Persistent Short-Run MCMC toward Energy-Based Model*. Curran Associates Inc., Red Hook, NY, USA.

Nitanda, A. (2014). Stochastic proximal gradient descent with acceleration techniques. In *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence and K. Weinberger, eds.), vol. 27. Curran Associates, Inc.

Ongie, G., Jalal, A., Baraniuk, C., Dimakis, A. and Willett, R. (2020). Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory* **PP** 1–1.

Rolfs, B., Rajaratnam, B., Guillot, D., Wong, I. and Maleki, A. (2012). Iterative thresholding algorithm for sparse inverse covariance estimation. In *Advances in Neural Information Processing Systems*.

Shlezinger, N., Whang, J., Eldar, Y. C. and Dimakis, A. G. (2021). Model-based deep learning: Key approaches and design guidelines. In *2021 IEEE Data Science and Learning Workshop (DSLW)*.

Tieleman, T. (2008). Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th International Conference on Machine Learning*. ICML '08, Association for Computing Machinery, New York,

NY, USA.

Vershynin, R. (2018). *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press.

Wainwright, M. J. (2019). *High-Dimensional Statistics: A Non-Asymptotic Viewpoint*. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press.

Wibisono, A. (2018). Sampling as optimization in the space of measures: The langevin dynamics as a composite optimization problem .

Younes, L. (1988). Estimation and annealing for gibbsian fields. *Annales de l'Institut Henri Poincaré. Probabilité et Statistiques* **24** 269–294.

Yuan, M. and Lin, Y. (2007). Model selection and estimation in the gaussian graphical model. *Biometrika* **94** 19–35.

Yuan, X. (2012). Alternating direction method for covariance selection models. *Journal of Scientific Computing* **51** 261–273.