# Learning Generative ConvNets via Multi-grid Modeling and Sampling

Ruiqi Gao[1]*, Yang Lu[2]*, Junpei Zhou[3], Song-Chun Zhu[1], Ying Nian Wu[1]
[1] University of California, Los Angeles, USA, [2] Amazon, [3] Zhejiang University, China
ruiqigao@ucla.edu, ylumzn@amazon.com
jpzhou1996@gmail.com, {sczhu, ywu}@stat.ucla.edu

## Abstract

*This paper proposes a multi-grid method for learning energy-based generative ConvNet models of images. For each grid, we learn an energy-based probabilistic model where the energy function is defined by a bottom-up convolutional neural network (ConvNet or CNN). Learning such a model requires generating synthesized examples from the model. Within each iteration of our learning algorithm, for each observed training image, we generate synthesized images at multiple grids by initializing the finite-step MCMC sampling from a minimal $1 \times 1$ version of the training image. The synthesized image at each subsequent grid is obtained by a finite-step MCMC initialized from the synthesized image generated at the previous coarser grid. After obtaining the synthesized examples, the parameters of the models at multiple grids are updated separately and simultaneously based on the differences between synthesized and observed examples. We show that this multi-grid method can learn realistic energy-based generative ConvNet models, and it outperforms the original contrastive divergence (CD) and persistent CD.*

## 1. Introduction

This paper studies the problem of learning energy-based generative ConvNet models [24, 10, 12, 11, 39, 25, 35, 28, 45, 46, 15] of images. The model is in the form of a Gibbs distribution where the energy function is defined by a bottom-up convolutional neural network (ConvNet or CNN). It can be derived from the commonly used discriminative ConvNet [23, 21] as a direct consequence of the Bayes rule [4], but unlike the discriminative ConvNet, the generative ConvNet is endowed with the gift of imagination in that it can generate images by sampling from the probability distribution of the model. As a result, the generative ConvNet can be learned in an unsupervised setting without requiring class labels. The learned model can be used as a prior model for image processing. It can also be turned into a discriminative ConvNet for classification.

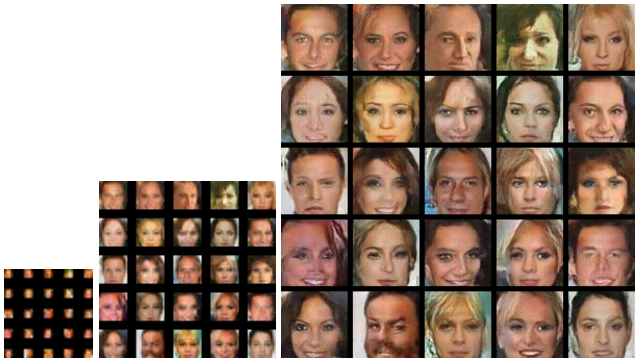

Figure 1. Synthesized images at multi-grids. From left to right: $4 \times 4$ grid, $16 \times 16$ grid and $64 \times 64$ grid. Synthesized image at each grid is obtained by 30 step Langevin sampling initialized from the synthesized image at the previous coarser grid, beginning with the $1 \times 1$ grid.

The maximum likelihood learning of the energy-based generative ConvNet model follows an "analysis by synthesis" scheme: we sample the synthesized examples from the current model, usually by Markov chain Monte Carlo (MCMC), and then update the model parameters based on the difference between the observed training examples and the synthesized examples. The probability distribution or the energy function of the learned model is likely to be multi-modal if the training data are highly varied. The MCMC may have difficulty traversing different modes and may take a long time to converge. A simple and popular modification of the maximum likelihood learning is the contrastive divergence (CD) learning [10], where for each observed training example, we obtain a corresponding synthesized example by initializing a finite-step MCMC from the observed example. Such a method can be scaled up to large training datasets using mini-batch training. However, the synthesized examples may be far from fair samples of the current model, thus resulting in bias of the learned model parameters. A modification of CD is persistent CD [42],

*Equal contributions.

where the MCMC is still initialized from the observed example at the initial learning epoch. However, in each subsequent learning epoch, the finite-step MCMC is initialized from the synthesized example of the previous epoch. Running persistent chains may make the synthesized examples less biased by the observed examples, although the persistent chains may still have difficulty traversing different modes of the learned model.

To address the above challenges under the constraint of finite budget MCMC, we propose a multi-grid method to learn the energy-based generative ConvNet models at multiple scales or grids. Specifically, for each training image, we obtain its multi-grid versions by repeated down-scaling. Our method learns a separate generative ConvNet model at each grid. Within each iteration of our learning algorithm, for each observed training image, we generate the corresponding synthesized images at multiple grids. Specifically, we initialize the finite-step MCMC sampling from the minimal $1 \times 1$ version of the training image, and the synthesized image at each grid serves to initialize the finite-step MCMC that samples from the model of the subsequent finer grid. See Fig. 1 for an illustration, where we sample images sequentially at 3 grids, with 30 steps of the Langevin dynamics at each grid. After obtaining the synthesized images at the multiple grids, the models at the multiple grids are updated separately and simultaneously based on the differences between the synthesized images and the observed training images at different grids.

The advantages of the proposed method are as follows.

(1) The finite-step MCMC is initialized from the $1 \times 1$ version of the observed image, instead of the original observed image. Thus the synthesized image is much less biased by the observed image compared to the original CD.

(2) The learned models at coarser grids are expected to be smoother than the models at finer grids. Sampling the models at increasingly finer grids sequentially is like a simulated annealing process [19] that helps the MCMC to mix.

(3) Unlike the original CD or persistent CD, the learned models are equipped with a fixed budget MCMC to generate new synthesized images from scratch, because we only need to initialize the MCMC by sampling from the one-dimensional histogram of the $1 \times 1$ version of the training images.

We show that the proposed method can learn realistic models of images. The learned models can be used for image processing such as image inpainting. The learned feature maps can be used for subsequent tasks such as classification.

The contributions of our paper are as follows. We propose a multi-grid method for learning energy-based generative ConvNet models. We show empirically that the proposed method outperforms the original CD, persistent CD, as well as the single-grid learning. More importantly, we show that a small budget MCMC is capable of generating diverse and realistic patterns. The deep energy-based models have not received the attention they deserve in the recent literature because of the reliance on MCMC sampling. It is our hope that this paper will stimulate further research on designing efficient MCMC algorithms for learning deep energy-based models.

## 2. Related work

Our method is related to CD [10] for training energy-based models. In general, both the data distribution of the observed training examples and the learned model distribution can be multi-modal, and the data distribution can be even more multi-modal than the model distribution. The finite-step MCMC of CD initialized from the data distribution may only explore local modes around the training examples, thus the finite-step MCMC may not get close to the model distribution. This can also be the case with persistent CD [42]. In contrast, our method initializes the finite-step MCMC from the minimal $1 \times 1$ version of the original image, and the sampling of the model at each grid is initialized from the image sampled from the model at the previous coarser grid. The model distribution at the coarser grid is expected to be smoother than the model distribution at the finer grid, and the coarse to fine MCMC is likely to generate varied samples from the learned models. As a result, the learned models obtained by our method can be closer to maximum likelihood estimate than the original CD.

The multi-grid Monte Carlo method originated from statistical physics [9]. The motivation for multi-grid Monte Carlo is that reducing the scale or resolution leads to a smoother or less multi-modal distribution. Our work is perhaps the first to apply the multi-grid sampling to the learning of deep energy-based models. The difference between our method and the multi-grid MCMC in statistical physics is that in the latter, the distribution of the lower resolution is obtained from the distribution of the higher resolution. In our work, the models at different grids are learned from training images at different resolutions directly and separately.

Besides energy-based generative ConvNet model, another popular deep generative model is the generator network or implicit generative model which maps the latent vector that follows a simple prior distribution to the image via a top-down ConvNet. The model is usually trained together with an assisting model such as an inferential model as in the variational auto-encoder (VAE) [18, 38, 31], or a discriminative model as in the generative adversarial networks (GAN) [8, 6, 37]. The focus of this paper is on training deep energy-based models, without resorting to a different class of models, so that we do not need to be concerned with the mismatch between the two different classes of models.

Our learning method is based on maximum likelihood. Recently, building on the early work of [43], [15, 22] have developed an introspective learning method to learn the energy-based model, where the energy function is discriminatively learned. It is possible to apply multi-grid learning and sampling to their method.

We would like to emphasize that this paper is not another paper on GAN. This paper seeks to answer the following question: Whether it is possible to learn the deep energy-based probabilistic models from big datasets by maximum likelihood type of algorithms, without relying on an extra network such as an implicit generative network? We believe this is a fundamental question, especially because an energy-based model corresponds directly to a discriminative classifier (see subsection 3.2). Our paper answers this question in affirmative.

# 3. Generative ConvNet

## 3.1. The model

Let $Y$ be the image defined on a squared (or rectangle) grid. We use $p_\theta(Y)$ to denote the probability distribution of $Y$ with parameter $\theta$. The energy-based generative ConvNet model is as follows [45]:

$$p_\theta(Y) = \frac{1}{Z(\theta)} \exp\left[f_\theta(Y)\right] p_0(Y), \tag{1}$$

where $p_0(Y)$ is the reference distribution such as Gaussian white noise $p_0(Y) \propto \exp\left(-\|Y\|^2/2\sigma^2\right)$ (or a uniform distribution within a bounded range). $f_\theta(Y)$ is defined by a bottom-up ConvNet whose parameters are denoted by $\theta$. The normalizing constant $Z(\theta) = \int \exp\left[f_\theta(Y)\right] p_0(Y) dY$ is analytically intractable. $p_\theta$ can be written in the form of an energy-based model: $p_\theta(Y) = \frac{1}{Z(\theta)} \exp[-\mathscr{E}_\theta(Y)]$. The energy function is

$$\mathscr{E}_\theta(Y) = \frac{1}{2\sigma^2}\|Y\|^2 - f_\theta(Y). \tag{2}$$

The local energy minima [13] satisfy an auto-encoder [45] $\frac{Y}{\sigma^2} = \frac{\partial}{\partial Y} f_\theta(Y)$. The learned model is likely to be multi-modal if the training data are highly varied.

## 3.2. Correspondence to discriminative ConvNet

Model (1) corresponds to a classifier in the following sense [4, 45, 43, 22, 15]. Suppose there are $K$ categories, $p_{\theta_k}(Y)$, for $k = 1, ..., K$, in addition to the background category $p_0(Y)$. The ConvNets $f_{\theta_k}(Y)$ for $k = 1, ..., K$ may share common lower layers. Let $\rho_k$ be the prior probability of category $k$, $k = 0, ..., K$. Then the posterior probability for classifying an example $Y$ to the category $k$ is a softmax multi-class classifier

$$\Pr(k|Y) = \frac{\exp(f_{\theta_k}(Y) + b_k)}{\sum_{k=0}^{K} \exp(f_{\theta_k}(Y) + b_k)}, \tag{3}$$

where $b_k = \log(\rho_k/\rho_0) - \log Z(\theta_k)$, and for $k = 0$, $f_{\theta_0}(Y) = 0$, $b_0 = 0$. Conversely, if we have the softmax classifier (3), then the distribution of each category is $p_{\theta_k}(Y)$ of the form (1). Thus the energy-based generative ConvNet directly corresponds to the commonly used discriminative ConvNet.

The correspondence to discriminative ConvNet classifier justifies the importance and naturalness of the energy-based generative ConvNet model.

# 4. Maximum likelihood

While the discriminative ConvNet must be learned in a supervised setting, the generative ConvNet model $p_\theta(Y)$ in (1) can be learned from unlabeled data by maximum likelihood.

## 4.1. Learning and sampling

Suppose we observe training examples $\{Y_i, i = 1, ..., n\}$ from an unknown data distribution $P_{\text{data}}(Y)$. The maximum likelihood learning seeks to maximize the log-likelihood function

$$L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(Y_i). \tag{4}$$

If the sample size $n$ is large, the maximum likelihood estimator minimizes the Kullback-Leibler divergence $\text{KL}(P_{\text{data}} \| p_\theta)$ from the data distribution $P_{\text{data}}$ to the model distribution $p_\theta$. The gradient of $L(\theta)$ is

$$L'(\theta) = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial \theta} f_\theta(Y_i) - \text{E}_\theta \left[ \frac{\partial}{\partial \theta} f_\theta(Y) \right], \tag{5}$$

where $\text{E}_\theta$ denotes the expectation with respect to $p_\theta(Y)$. The key to the above identity is that $\frac{\partial}{\partial \theta} \log Z(\theta) = \text{E}_\theta[\frac{\partial}{\partial \theta} f_\theta(Y)]$.

The expectation in equation (5) is analytically intractable and has to be approximated by MCMC, such as the Langevin dynamics [48, 7], which iterates the following step:

$$\begin{aligned} Y_{\tau+\Delta\tau} &= Y_\tau - \frac{\Delta\tau}{2} \frac{\partial}{\partial Y} \mathscr{E}_\theta(Y_\tau) + \sqrt{\Delta\tau} Z_\tau \\ &= Y_\tau - \frac{\Delta\tau}{2} \left[ \frac{Y_\tau}{\sigma^2} - \frac{\partial}{\partial Y} f_\theta(Y_\tau) \right] + \sqrt{\Delta\tau} Z_\tau, \end{aligned} \tag{6}$$

where $\tau$ indexes the time of the Langevin dynamics, $\Delta\tau$ is the step size, and $Z_\tau \sim \text{N}(0, I)$ is Gaussian white noise. Let the distribution of $Y_\tau$ be $p_\tau$, then $\text{KL}(p_\tau || p_\theta) \to 0$ monotonically as $\tau \to \infty$ according to the second law of thermodynamics [3]. $\text{KL}(p_\tau || p_\theta)$ can be decomposed into energy and entropy. The gradient descent part of the Langevin dynamics reduces the energy, while the Brownian motion part increases the entropy. A Metropolis-Hastings step may be added to correct for the finite step size $\Delta\tau$. We have also implemented Hamiltonian Monte Carlo (HMC) for sampling the generative ConvNet [33, 4].

We can run $\tilde{n}$ parallel chains of the Langevin dynamics according to (6) to obtain the synthesized examples $\{\tilde{Y}_i, i = 1, ..., \tilde{n}\}$. The Monte Carlo approximation to $L'(\theta)$ is

$$\begin{aligned} L'(\theta) &\approx \frac{1}{n}\sum_{i=1}^{n} \frac{\partial}{\partial \theta} f_\theta(Y_i) - \frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \frac{\partial}{\partial \theta} f_\theta(\tilde{Y}_i) \quad (7) \\ &= \frac{\partial}{\partial \theta}\left[\frac{1}{\tilde{n}}\sum_{i=1}^{\tilde{n}} \mathscr{E}_\theta(\tilde{Y}_i) - \frac{1}{n}\sum_{i=1}^{n} \mathscr{E}_\theta(Y_i)\right], \end{aligned}$$

which is used to update $\theta$.

### 4.2. Contrastive divergence

The MCMC sampling of $p_\theta$ may take a long time to converge, especially if the learned $p_\theta$ is multi-modal, which is often the case because $P_{\text{data}}$ is usually multi-modal. In order to learn from large datasets, we can only afford small budget MCMC, i.e., within each learning iteration, we can only run MCMC for a small number of steps. To meet such a challenge, [10] proposed the contrastive divergence (CD) method, where within each learning iteration, we initialize the finite-step MCMC from each $Y_i$ in the current training batch to obtain a synthesized example $\tilde{Y}_i$. The parameters are then updated according to the learning gradient (7).

Let $M_\theta$ be the transition kernel of the finite-step MCMC that samples from $p_\theta(Y)$. For any probability distribution $p(Y)$ and any Markov transition kernel $M$, let $Mp(Y') = \int p(Y)M(Y, Y')dY$ denote the marginal distribution obtained after running $M$ starting from $p$. The learning gradient of CD approximately follows the gradient of the difference between two Kullback-Leibler (KL) divergences:

$$\text{KL}(P_{\text{data}} \| p_\theta) - \text{KL}(M_\theta P_{\text{data}} \| p_\theta), \quad (8)$$

thus the name "contrastive divergence". If $M_\theta P_{\text{data}}$ is close to $p_\theta$, then the second divergence is small, and the CD estimate is close to maximum likelihood which minimizes the first divergence. However, it is likely that $P_{\text{data}}$ and the learned $p_\theta$ are multi-modal. It is expected that $p_\theta$ is smoother than $P_{\text{data}}$, i.e., $P_{\text{data}}$ is "colder" than $p_\theta$ in the language of simulated annealing [19]. If $P_{\text{data}}$ is different from $p_\theta$, it is unlikely that $M_\theta P_{\text{data}}$ becomes much closer to $p_\theta$ due to the trapping of local modes. This may lead to bias in the CD estimate.

A persistent version of CD [42] is to initialize the MCMC from the observed $Y_i$ in the beginning, and then in each learning epoch, the MCMC is initialized from the synthesized $\tilde{Y}_i$ obtained in the previous epoch. The persistent CD may still face the challenge of traversing and exploring different local energy minima.

### 4.3. Modified and adversarial CDs

This subsection explains modifications of CD, including methods based on an additional generator network. It can be skipped in the first reading.

The original CD initializes MCMC sampling from the data distribution $P_{\text{data}}$. We may modify it by initializing MCMC sampling from a given distribution $P_0$, in the hope that $M_\theta P_0$ is closer to $p_\theta$ than $M_\theta P_{\text{data}}$. The learning gradient approximately follows the gradient of

$$\text{KL}(P_{\text{data}} \| p_\theta) - \text{KL}(M_\theta P_0 \| p_\theta). \quad (9)$$

That is, we run a finite-step MCMC from a given initial distribution $P_0$, and use the resulting samples as synthesized examples to approximate the expectation in (5). The approximation can be made more accurate using annealed importance sampling [32]. Following the idea of simulated annealing, $P_0$ should be a "smoother" distribution than $p_\theta$ (the extreme case is to start from white noise $P_0$). Unlike persistent CD, here the finite-step MCMC is non-persistent, sometimes also referred to as "cold start", where the MCMC is initialized from a given $P_0$ within each learning iteration, instead of from the examples synthesized by the previous learning epoch. The cold start version is easier to implement for mini-batch learning.

With the multi-grid method (to be introduced in the next section), at each grid, $P_0$ is the distribution of the images generated by the previous coarser grid. At the smallest grid, $P_0$ is the one-dimensional histogram of the $1 \times 1$ versions of the training images.

Another possibility is to recruit a generator network $q_\alpha(Y)$ as an approximated direct sampler [16, 5], so that $p_\theta$ and $q_\alpha$ can be jointly learned by the adversarial CD:

$$\min_{p_\theta} \max_{q_\alpha} \left[\text{KL}(P_{\text{data}} \| p_\theta) - \text{KL}(q_\alpha \| p_\theta)\right]. \quad (10)$$

That is, the learning of $p_\theta$ is modified CD with $q_\alpha$ supplying synthesized examples, and the learning of $q_\alpha$ is based on $\min_{q_\alpha} \text{KL}(q_\alpha \| p_\theta)$, which is a variational approximation. The adversarial CD is related to Wasserstein GAN [1], except that the former regularizes the entropy of the generator, while the latter regularizes the critic.

[44] also studied the problem of joint learning of the energy-based model and the generator model. The learning of the energy-based model is based on the modified CD:

$$\text{KL}(P_{\text{data}} \| p_\theta) - \text{KL}(M_\theta q_\alpha \| p_\theta), \quad (11)$$

with $q_\alpha$ taking the role of $P_0$, whereas the learning of the generator is based on how $M_\theta q_\alpha$ modifies $q_\alpha$, and is accomplished by $a_{t+1} = \arg\min_\alpha \text{KL}(M_\theta q_{\alpha_t} \| q_\alpha)$, i.e., $q_\alpha$ accumulates MCMC transitions to be close to the stationary distribution of $M_\theta$, which is $p_\theta$.

In this paper, we shall not consider recruiting a generator network, so that we do not need to worry about the mismatch between the generator model and the energy-based model. In other words, instead of relying on a learned approximate direct sampler, we endeavor to develop small budget MCMC for sampling.

## 5. Multi-grid modeling and sampling

We propose a multi-grid method for learning and sampling generative ConvNet models. For an image $Y$, let $(Y^{(s)}, s = 0, ..., S)$ be the multi-grid versions of $Y$, with $Y^{(0)}$ being the minimal $1 \times 1$ version of $Y$, and $Y^{(S)} = Y$. For each $Y^{(s)}$, we can divide the image grid into squared blocks of $d \times d$ pixels. We can reduce each $d \times d$ block into a single pixel by averaging the intensity values of the $d \times d$ pixels. Such a down-scaling operation maps $Y^{(s)}$ to $Y^{(s-1)}$. Conversely, we can also define an up-scaling operation, by expanding each pixel of $Y^{(s-1)}$ into a $d \times d$ block of constant intensity to obtain an up-scaled version $\hat{Y}^{(s)}$ of $Y^{(s-1)}$. The up-scaled $\hat{Y}^{(s)}$ is not identical to the original $Y^{(s)}$ because the high resolution details are lost. The mapping from $Y^{(s)}$ to $Y^{(s-1)}$ is a linear projection onto a set of orthogonal basis vectors, each of which corresponds to a $d \times d$ block. The up-scaling operation is a pseudo-inverse of this linear mapping. In general, $d$ does not even need to be an integer (e.g., $d = 1.5$) for the existence of the linear mapping and its pseudo-inverse.

Let $p^{(s)}_{\theta^{(s)}}(Y^{(s)})$ be the energy-based generative ConvNet model at grid $s$. $p^{(0)}$ can be simply modeled by a one-dimensional histogram of $Y^{(0)}$ pooled from the $1 \times 1$ versions of the training images.

Within each learning iteration, for each training image $Y_i$ in the current learning batch, we initialize the finite-step MCMC from the $1 \times 1$ image $Y^{(0)}_i$. For $s = 1, ..., S$, we sample from the current $p^{(s)}_{\theta^{(s)}}(Y^{(s)})$ by running $l$ steps of the Langevin dynamics from the up-scaled version of $\tilde{Y}^{(s-1)}_i$ sampled at the previous coaser grid. After that, for $s = 1, ..., S$, we update the model parameters $\theta^{(s)}$ based on the difference between the synthesized $\{\tilde{Y}^{(s)}_i\}$ and the observed $\{Y^{(s)}_i\}$ according to equation (7).

Algorithm 1 provides the details of the multi-grid method.

In the above sampling scheme, $p^{(0)}$ can be sampled directly because it is a one-dimensional histogram. Each $p^{(s)}$ is expected to be smoother than $p^{(s+1)}$. Thus the sampling scheme is similar to simulated annealing, where we run finite-step MCMC through a sequence of probability distributions that are increasingly multi-modal (or cold), in the hope of reaching and exploring major modes of the model distributions. The learning process then shifts these major modes toward the observed examples, while sharpening these modes along the way, in order to memorize the observed examples with these major modes of the model distributions.

Let $P^{(s)}_{\text{data}}$ be the data distribution of $\{Y^{(s)}_i\}$. Let $p^{(s)}_{\theta^{(s)}}$ be the model at grid $s$. Let $P^{(s)}_{\theta^{(s-1)}}$ be the up-scaled version of the model $p^{(s-1)}_{\theta^{(s-1)}}$. Specifically, let $Y^{(s-1)} \sim p^{(s-1)}_{\theta^{(s-1)}}$ be a random example at grid $s-1$, and let $\hat{Y}^{(s)}$ be the up-scaled version of $Y^{(s-1)}$, then $P^{(s)}_{\theta^{(s-1)}}$ is the distribution of $\hat{Y}^{(s)}$. Let $M^{(s)}_{\theta^{(s)}}$ be the Markov transition kernel of $l$-step Langevin dynamics that samples $p^{(s)}_{\theta^{(s)}}$. The learning gradient of the multi-grid method at grid $s$ approximately follows the gradient of the difference between two KL divergences:

$$\text{KL}\left(P^{(s)}_{\text{data}} \| p^{(s)}_{\theta^{(s)}}\right) - \text{KL}\left(M^{(s)}_{\theta^{(s)}} P^{(s)}_{\theta^{(s-1)}} \| p^{(s)}_{\theta^{(s)}}\right). \tag{12}$$

$P^{(s)}_{\theta^{(s-1)}}$ is smoother than $p^{(s)}_{\theta^{(s)}}$, and $M^{(s)}_{\theta^{(s)}}$ will evolve $P^{(s)}_{\theta^{(s-1)}}$ to a distribution close to $p^{(s)}_{\theta^{(s)}}$ by creating details at the current resolution. If we use the original CD by initializing MCMC from $P^{(s)}_{\text{data}}$, then we are sampling a multi-modal (cold) distribution $p^{(s)}_{\theta^{(s)}}$ by initializing from a presumably even more multi-modal (or colder) distribution $P^{(s)}_{\text{data}}$, and we may not expect the resulting distribution to be close to the target $p^{(s)}_{\theta^{(s)}}$.

**Algorithm 1** Multi-grid sampling and learning

**Input:**

(1) training examples $\{Y^{(s)}_i, s = 1, ..., S, i = 1, ..., n\}$,
(2) number of Langevin steps $l$,
(3) number of learning iterations $T$.

**Output:**

(1) estimated parameters $(\theta^{(s)}, s = 1, ..., S)$,
(2) synthesized examples $\{\tilde{Y}^{(s)}_i, s = 1, ..., S, i = 1, ..., n\}$.

1: Let $t \leftarrow 0$, initialize $\theta^{(s)}, s = 1, ..., S$.
2: **repeat**
3: For $i = 1, ..., n$, initialize $\tilde{Y}^{(0)}_i = Y^{(0)}_i$.
4: For $s = 1, ..., S$, initialize $\tilde{Y}^{(s)}_i$ as the up-scaled version of $\tilde{Y}^{(s-1)}_i$, and run $l$ steps of the Langevin dynamics to evolve $\tilde{Y}^{(s)}_i$, each step following equation (6).
5: For $s = 1, ..., S$, update $\theta^{(s)}_{t+1} = \theta^{(s)}_t + \gamma_t L'(\theta^{(s)}_t)$, with step size $\gamma_t$, where $L'(\theta^{(s)}_t)$ is computed according to equation (7).
6: Let $t \leftarrow t + 1$.
7: **until** $t = T$

## 6. Experiments

**Project page**: The code and more results can be found at http://www.stat.ucla.edu/~ruiqigao/multigrid/main.html.

We learn the models at 3 grids: $4 \times 4$, $16 \times 16$ and $64 \times 64$, which we refer to as grid1, grid2 and grid3, respectively. That is, we set $S = 3$ (number of grids), $d = 4$ (reducing each $4 \times 4$ block to a pixel in the down-scaling operation).

We conduct qualitative and quantitative experiments to evaluate our method with respect to several baseline methods. The first baseline is the single-grid method: starting from a $1 \times 1$ image, we directly up-scale it to $64 \times 64$ and sample a $64 \times 64$ image using a single generative ConvNet. The other two baselines are CD1 (running 1 step Langevin dynamics from the observed images) and persistent CD. Both CD baselines initialize the MCMC sampling from the observed images.

### 6.1. Implementation details

The training images are resized to $64 \times 64$. Since the models of the three grids act on images of different scales, we design a specific ConvNet structure per grid: grid1 has a 3-layer network with $5 \times 5$ stride 2 filters at the first layer and $3 \times 3$ stride 1 filters at the next two layers; grid2 has a 4-layer network with $5 \times 5$ stride 2 filters at the first layer and $3 \times 3$ stride 1 filters at the next three layers; grid3 has a 3-layer network with $5 \times 5$ stride 2 filters at the first layer, $3 \times 3$ stride 2 filters at the second layer, and $3 \times 3$ stride 1 filters at the third layer. Numbers of channels are $96 - 128 - 256$ at grid1 and grid3, and $96 - 128 - 256 - 512$ at grid2. A fully-connected layer with 1 channel output is added on top of every grid to get the value of $f_\theta(Y)$. Batch normalization [14] and leaky ReLU activations are applied after every convolution. At each iteration, we run $l = 30$ steps of the Langevin dynamics for each grid with $\sqrt{\Delta\tau} = 0.3$. All networks are trained simultaneously with mini-batches of size 100 and an initial learning rate of 0.3. Learning rate is decayed logarithmically every 10 iterations.

For CD1, persistent CD and the single-grid method, we follow the same setting as the multi-grid method except that for persistent CD and the single-grid method, we set the Langevin steps to 90 to maintain the same MCMC budget as the multi-grid method. We use the same network structure of grid3 for these baseline methods.

### 6.2. Synthesis

We learn multi-grid models from five datasets: CelebA [27], Large-scale Scene Understanding (LSUN) [41], CIFAR-10 [20], Street View Housing Numbers (SVHN) [34] and MIT places205 [47]. In the CelebA dataset, we randomly sample 10,000 images for training. Fig. 2 show synthesized images generated by models learned from CelebA dataset. We also show synthesized images generated by models learned by DCGAN [37] and the single-grid method. CD1 and persistent CD cannot synthesize realistic images, thus we do not bother to show their synthesis results. Compared with the single-grid method, images generated by the multi-grid method are more realistic. The results from multi-grid models are comparable to the results from DCGAN. Fig. 3 shows synthesized images from models learned from the LSUN bedrooms dataset, which contains more than 3 million training images. The SVHN dataset consists of color images of house numbers collected by Google Street View. The training set consists of 73,257 images and the testing set has 26,032 images. We learn the models in the unsupervised manner. MIT places205 contains images of 205 scene categories. We learn from a sin-


Figure 2. Synthesized images from models learned from the CelebA dataset. From left to right: observed images, images synthesized by DCGAN [37], single-grid method and multi-grid method. CD1 and persistent CD cannot synthesize realistic images and their results are not shown.

(a) Original images

(b) Synthesized images

Figure 3. Synthesized images generated by the multi-grid models learned from the LSUN bedrooms dataset.

gle category. Please refer to the supplementary materials for synthesized results by models learned from SVHN dataset and several categories of MIT places205 dataset.

Figure 4. Synthesized images generated by the multi-grid models learned from the CIFAR-10 dataset. Each row illustrates a category, and the multi-grid models are learned conditional on the category. From top to bottom: *airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck*.

Table 1. Inception scores on CIFAR-10.

| | Real images | DCGAN | multi-grid method |
|---|---|---|---|
| Inception score | 11.237 | 6.581 | 6.565 |

CIFAR-10 includes various object categories and has 50,000 training examples. Fig. 4 shows the synthesized images generated by models learned by the multi-grid method conditional on each category. In this experiment, we run 40 steps of the Langevin dynamics for each grid, and in the final synthesis after learning, we disable the noise term in the Langevin dynamics, which slightly improves the synthesis quality. We evaluate the quality of synthesized images quantitatively using the average inception score [40] in Table 1. The multi-grid method gets comparable inception score as DCGAN as reported in [26].

To check the diversity of Langevin dynamics sampling, we synthesize images by initializing the Langevin dynamics from the same $1 \times 1$ image. As shown in Fig. 5, after 90 steps of Langevin dynamics, the sampled images from the same $1 \times 1$ image are different from each other.

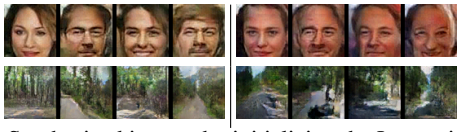

Figure 5. Synthesized images by initializing the Langevin dynamics sampling from the same $1 \times 1$ image. Each block of 4 images are generated from the same $1 \times 1$ image.

Table 2. Classification error of L2-SVM trained on the features learned from SVHN.

| Test error rate with # of labeled images | 1,000 | 2,000 | 4,000 |
|---|---|---|---|
| Persistent CD [42] | 45.74 | 39.47 | 34.18 |
| One-step CD [10] | 44.38 | 35.87 | 30.45 |
| Wasserstein GAN [1] | 43.15 | 38.00 | 32.56 |
| Deep directed generative models [16] | 44.99 | 34.26 | 27.44 |
| DCGAN[37] | 38.59 | 32.51 | 29.37 |
| single-grid method | 36.69 | 30.87 | 25.60 |
| multi-grid method | **30.23** | **26.54** | **22.83** |

Table 3. Classification error of CNN classifier trained on the features of three grids learned from SVHN.

| Test error rate with # of labeled images | 1,000 | 2,000 | 4,000 |
|---|---|---|---|
| DGN [17] | 36.02 | - | - |
| Virtual adversarial [30] | 24.63 | - | - |
| Auxiliary deep generative model [29] | 22.86 | - | - |
| Supervised CNN with the same structure | 39.04 | 22.26 | 15.24 |
| multi-grid method + CNN classifier | **19.73** | **15.86** | **12.71** |

### 6.3. Unsupervised feature learning for classfication

To evaluate the features learned by the multi-grid method, we perform a semi-supervised classification experiment by following the same procedure outlined in [37]. That is, we use the multi-grid method as a feature extractor. We first train a multi-grid model on the combination of SVHN training and testing sets in an unsupervised way. Then we train a regularized L2-SVM on the learned representations of grid 3. For fair comparison, we adopt the discriminator structure of [37] for grid 3, which has 4 convolutional layers of $5 \times 5$ filters with 64, 128, 256 and 512 channels respectively. The features from all the convolutional layers are max pooled and concatenated to form a 15,360-dimensional vector. We randomly sample 1000, 2000 and 4000 labeled examples from the training dataset to train the SVM and test on the testing dataset. Within the same setting, we compare the learned features of the multi-grid method with the single-grid method, persistent CD [42], one-step CD [10], Wasserstein GAN [1], deep directed generative models [16] and DCGAN [37]. Table 2 shows the classification results, indicating that the multi-grid method learns strong features.

Next we try to combine the learned features of three grids together. Specifically, we build a two-layer classification CNN on top of the top layer feature maps of three grids. The first layer is a $3 \times 3$ stride 1 convolutional layer with 64 channels operated separately on the feature maps of the three grids. Then the outputs from the three grids are concatenated to form a 34,624-dimensional vector. A fully-connected layer is added on top of the vector. We train this classifier using 1000, 2000 and 4000 labeled examples that are randomly sampled from the training set. As shown in Table 3, our method achieves a test error rate of 19.73% for

1,000 labeled images. For comparison, we train a classification network from scratch with the same structure (three networks as used in the multi-grid method plus two layers for classification) on the same labeled training data. It has a significantly higher error rate of 39.04% for 1,000 labeled training images. Our method also outperforms some methods that are specifically designed for semi-supervised learning, such as DGN [17], virtual adversarial [30] and auxiliary deep generative model [29].

### 6.4. Image inpainting

We further test our method on image inpainting. In this task, we try to learn the conditional distribution $p_\theta(Y_M|Y_{\bar{M}})$ by our models, where $M$ consists of pixels to be masked, and $\bar{M}$ consists of pixels not to be masked. In the training stage, we randomly place the mask on each training image, but we assume $Y_M$ is observed in training. We follow the same learning and sampling algorithm as in Algorithm 1, except that in the sampling step (i.e., step 4 in Algorithm 1), in each Langevin step, only the masked part of the image is updated, and the unmasked part remains fixed as observed. This is a generalization of the pseudo-likelihood estimation [2], which corresponds to the case where $M$ consists of one pixel. It can also be considered a form of associative memory [13]. After learning $p_\theta(Y_M|Y_{\bar{M}})$ from the fully observed training images, we then use it to inpaint the masked testing images, where the masked parts are not observed.

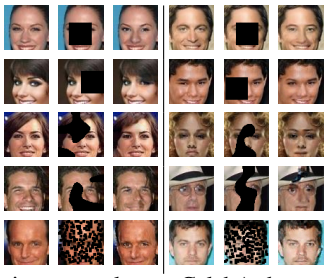

Figure 6. Inpainting examples on CelebA dataset. In each block from left to right: the original image, masked input, inpainted image by the multi-grid method.

Table 4. Quantitative evaluations for three types of masks. Lower values of error are better. Higher values of PSNR are better. PCD, CD1, SG, CE and MG indicate persistent CD, one-step CD, single-grid method, ContextEncoder and multi-grid method, respectively.

| | Mask | PCD | CD1 | SG | CE | MG |
|---|---|---|---|---|---|---|
| Error | Mask | 0.056 | 0.081 | 0.066 | 0.045 | **0.042** |
| | Doodle | 0.055 | 0.078 | 0.055 | 0.050 | **0.045** |
| | Pepper | 0.069 | 0.084 | 0.054 | 0.060 | **0.036** |
| PSNR | Mask | 12.81 | 12.66 | 15.97 | **17.37** | 16.42 |
| | Doodle | 12.92 | 12.68 | 14.79 | 15.40 | **16.98** |
| | Pepper | 14.93 | 15.00 | 15.36 | 17.04 | **19.34** |

We use 10,000 face images randomly sampled from CelebA dataset to train the model. We set the mask size at $32 \times 32$ for training. During training, the size of the mask is fixed but the position is randomly selected for each training image. Another 1,000 face images are randomly selected from CelebA dataset for testing. We find that during the testing, the mask does not need to be restricted to $32 \times 32$ square mask. So we test three different shapes of masks: 1) $32 \times 32$ square mask, 2) doodle mask with approximately 25% missing pixels, and 3) pepper and salt mask with approximately 60% missing pixels. Fig. 6 shows some inpainting examples.

We perform quantitative evaluations using two metrics: 1) reconstruction error measured by the per pixel difference and 2) peak signal-to-noise ratio (PSNR). Metrics are computed between the inpainting results obtained by different methods and the original face images on the masked pixels. We compare with persistent CD, CD1 and the single-grid method. We also compare with the ContextEncoder [36] (CE). We re-train the CE model on 10,000 training face images for fair comparison. As our tested masks are not in the image center, we use the "inpaintRandom" version of the CE code and randomly place a $32 \times 32$ mask in each image during training. The results are shown in Table 4. It shows that the multi-grid method works well for the inpainting task.

## 7. Conclusion

This paper seeks to address the fundamental question of whether we can learn energy-based generative ConvNet models purely by themselves without recruiting extra networks such as generator networks. This question is important both conceptually and practically because the energy-based generative ConvNet models correspond directly to discriminative ConvNet classifiers. Being able to learn and sample from such models also provides us a valuable alternative to GAN methods, by relieving us from the concerns with issues such as mismatch between two different classes of models, as well as instability in learning.

To answer the above question, we propose a multi-grid method for learning energy-based generative ConvNet models. Our work seeks to facilitate the learning of such models by developing small budget MCMC initialized from a simple distribution for sampling from the learned models. We show that our method can learn realistic models of images and the learned models can be useful for tasks such as image processing and classification.

## Acknowledgment

The work is supported by DARPA SIMPLEX N66001-15-C-4035, ONR MURI N00014-16-1-2007, DARPA ARO W911NF-16-1-0579, and DARPA N66001-17-2-4029. We thank Erik Nijkamp for his help with writing and coding.

## References


[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017. 4, 7

[2] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 192–236, 1974. 8

[3] T. M. Cover and J. A. Thomas. *Elements of information theory*. John Wiley & Sons, 2012. 3

[4] J. Dai, Y. Lu, and Y.-N. Wu. Generative modeling of convolutional neural networks. *arXiv preprint arXiv:1412.6296*, 2014. 1, 3

[5] Z. Dai, A. Almahairi, P. Bachman, E. Hovy, and A. Courville. Calibrating energy-based generative adversarial networks. *arXiv preprint arXiv:1702.01691*, 2017. 4

[6] E. L. Denton, S. Chintala, R. Fergus, et al. Deep generative image models using a laplacian pyramid of adversarial networks. In *Advances in neural information processing systems*, pages 1486–1494, 2015. 2

[7] M. Girolami and B. Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011. 3

[8] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 2672–2680, 2014. 2

[9] J. Goodman and A. D. Sokal. Multigrid monte carlo method. conceptual foundations. *Physical Review D*, 40(6):2035, 1989. 2

[10] G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002. 1, 2, 4, 7

[11] G. E. Hinton, S. Osindero, and Y.-W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18:1527–1554, 2006. 1

[12] G. E. Hinton, S. Osindero, M. Welling, and Y.-W. Teh. Unsupervised discovery of nonlinear structure using contrastive backpropagation. *Cognitive Science*, 30(4):725–731, 2006. 1

[13] J. J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982. 3, 8

[14] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 6

[15] L. Jin, J. Lazarow, and Z. Tu. Introspective classification with convolutional nets. In *Advances in Neural Information Processing Systems*, pages 823–833, 2017. 1, 3

[16] T. Kim and Y. Bengio. Deep directed generative models with energy-based probability estimation. *arXiv preprint arXiv:1606.03439*, 2016. 4, 7

[17] D. P. Kingma, S. Mohamed, D. J. Rezende, and M. Welling. Semi-supervised learning with deep generative models. In *Advances in Neural Information Processing Systems*, pages 3581–3589, 2014. 7, 8

[18] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[19] S. Kirkpatrick, C. D. Gelatt, M. P. Vecchi, et al. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983. 2, 4

[20] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. 2009. 6

[21] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1

[22] J. Lazarow, L. Jin, and Z. Tu. Introspective neural networks for generative modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2774–2783, 2017. 3

[23] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1

[24] Y. LeCun, S. Chopra, R. Hadsell, M. Ranzato, and F. J. Huang. A tutorial on energy-based learning. 2006. 1

[25] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In *International Conference on Machine Learning*, pages 609–616, 2009. 1

[26] Q. Liu and D. Wang. Learning deep energy models: Contrastive divergence vs. amortized mle. *arXiv preprint arXiv:1707.00797*, 2017. 7

[27] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3730–3738, 2015. 6

[28] Y. Lu, S.-C. Zhu, and Y. N. Wu. Learning FRAME models using CNN filters. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016. 1

[29] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther. Auxiliary deep generative models. *arXiv preprint arXiv:1602.05473*, 2016. 7, 8

[30] T. Miyato, S.-i. Maeda, M. Koyama, K. Nakae, and S. Ishii. Distributional smoothing by virtual adversarial examples. *stat*, 1050:2, 2015. 7, 8

[31] A. Mnih and K. Gregor. Neural variational inference and learning in belief networks. In *International Conference on Machine Learning*, pages 1791–1799, 2014. 2

[32] R. M. Neal. Annealed importance sampling. *Statistics and Computing*, 11, 2001. 4

[33] R. M. Neal. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2, 2011. 3

[34] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, page 5, 2011. 6

[35] J. Ngiam, Z. Chen, P. W. Koh, and A. Y. Ng. Learning deep energy models. In *International Conference on Machine Learning*, pages 1105–1112, 2011. 1

[36] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 8

[37] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. 2, 6, 7

[38] D. J. Rezende, S. Mohamed, and D. Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International Conference on Machine Learning*, pages 1278–1286, 2014. 2

[39] R. Salakhutdinov and G. E. Hinton. Deep boltzmann machines. In *AISTATS*, 2009. 1

[40] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016. 7

[41] F. Y. Y. Z. S. Song and A. S. J. Xiao. Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 6

[42] T. Tieleman. Training restricted boltzmann machines using approximations to the likelihood gradient. In *Proceedings of the 25th international conference on Machine learning*, pages 1064–1071. ACM, 2008. 1, 2, 4, 7

[43] Z. Tu. Learning generative models via discriminative approaches. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007. 3

[44] J. Xie, Y. Lu, R. Gao, S.-C. Zhu, and Y. N. Wu. Cooperative training of descriptor and generator networks. *arXiv preprint arXiv:1609.09408*, 2017. 4

[45] J. Xie, Y. Lu, S.-C. Zhu, and Y. N. Wu. A theory of generative convnet. In *International Conference on Machine Learning*, pages 2635–2644, 2016. 1, 3

[46] J. Xie, S.-C. Zhu, and Y. N. Wu. Synthesizing dynamic patterns by spatial-temporal generative convnet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7093–7101, 2017. 1

[47] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems*, pages 487–495, 2014. 6

[48] S. C. Zhu and D. Mumford. Grade: Gibbs reaction and diffusion equitions. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 847–854, 1998. 3