Patchwise Generative ConvNet: Training Energy-Based Models from a Single Natural Image for Internal Learning

Zilong Zheng [1]*, Jianwen Xie [2], Ping Li [2]
[1] University of California, Los Angeles, CA
[2] Cognitive Computing Lab, Baidu Research, Bellevue, WA
z.zheng@ucla.edu, {jianwenxie,liping11}@baidu.com

Abstract

Exploiting internal statistics of a single natural image has long been recognized as a significant research paradigm where the goal is to learn the internal distribution of patches within the image without relying on external training data. Different from prior works that model such a distribution implicitly with a top-down latent variable model (e.g., generator), this paper proposes to explicitly represent the statistical distribution within a single natural image by using an energy-based generative framework, where a pyramid of energy functions, each parameterized by a bottom-up deep neural network, are used to capture the distributions of patches at different resolutions. Meanwhile, a coarse-to-fine sequential training and sampling strategy is presented to train the model efficiently. Besides learning to generate random samples from white noise, the model can learn in parallel with a self-supervised task (e.g., recover the input image from its corrupted version), which can further improve the descriptive power of the learned model. The proposed model is simple and natural in that it does not require an auxiliary model (e.g., discriminator) to assist the training. Besides, it also unifies internal statistics learning and image generation in a single framework. Experimental results presented on various image generation and manipulation tasks, including super-resolution, image editing, harmonization, style transfer, etc, have demonstrated the effectiveness of our model for internal learning.

1. Introduction

Learning internal statistics or modeling the internal distribution of patches within a single natural image can date back to learning statistical models for texture synthesis in computer vision. In 1926, a pioneer Julesz [16] initiated the research on texture perception in pre-attentive vision by raising the following fundamental question:

> What features and statistics are characteristics of a texture pattern, so that texture pairs that share the same features and statistics cannot be told apart by pre-attentive human visual perception?
>
> — Béla Julesz [16]

Julesz's question implies two challenging tasks: (1) What are the internal statistical properties that define a texture from the human perception perspective? (2) Given a set of statistical properties, how can we synthesize diverse realistic texture patterns with identical internal statistical properties? These two questions motivate various researchers on pursuing statistical representation and learning frameworks for texture synthesis. Representative pioneer works include k-gon statistics [45], primal sketch [22], and FRAME (Filters, Random field, And Maximum Entropy) [46] *etc*. The FRAME, in particular, models texture as an energy-based model (EBM) [19], seeking to represent stochastic textures by simultaneously learning statistics of textures based on Gabor filter responses and generating novel texture patterns that exhibit the same statistics as the learned texture image by Gibbs sampling [10].

Empowered with the recent development of deep learning techniques, the energy-based Generative ConvNet [36] (also known as DeepFRAME model [34]) has been proposed as a deep generalization of the FRAME model for modeling high dimensional signals. Remarkable successes of the generative ConvNets have been shown in modeling and synthesizing images [36, 6, 27, 5, 12], video sequences [41, 42], 3D voxels [38, 39], molecule [4], unordered point clouds [37], *etc*.

More recently, the computer vision community has shown a growing interest in the research topic of deep internal learning (DIL), with works [33, 30, 28] that train deep models on a single natural example. In this paper, we bring the powerful energy-based generative ConvNet framework into DIL by proposing an unconditional generative model

*This work was conducted when Zilong Zheng was a research intern at Baidu Research – 10900 NE 8th St. Bellevue, WA 98004, USA.

learned from a single natural image. Specifically, we show that the internal statistics of overlapping patches within an image can be learned by an energy-based generative ConvNet, in which the internal statistics are represented by an energy function parameterized by a deep convolutional neural network, and the generation is driven by the estimated energy function. To capture different scales of internal statistical properties, we sequentially learn a pyramid of EBMs with different resolutions in a coarse-to-fine manner. The EBM at each scale is a generative ConvNet and trained by the "analysis by synthesis" scheme, in which we generate samples from the EBM via Markov chain Monte Carlo (MCMC) [21, 1] and then use the samples to compute the gradient of the log-likelihood to update the model parameters. Taking advantage of the multiple resolution setting, the sampling of each EBM can be more efficient by using a sequential sampling strategy, where the lower resolution EBM uses its synthesized images to initialize the MCMC of the higher resolution EBM. Once the EBMs are trained from a single image, the pyramid of the learned statistics can be useful for different vision tasks, such as generation of images with complex structures and textures, super-resolution, image editing, style transfer, and harmonization. The proposed energy-based internal learning framework is appealing because of the following aspects:

- **Architecture efficiency:** Each EBM at a different resolution only contains one single bottom-up network as the energy function, and does not need any other assisting network architecture for joint training.
- **Training efficiency:** The EBM relies on maximum likelihood estimation (MLE), which in general does not encounter the mode collapse issue that would commonly occur in adversarial learning [11].
- **Representation efficiency:** The energy-based learning amounts to training a model that can synthesize images that match the observed statistics. It unifies the concepts of description and generation into one single framework.

The main contributions of this work are four-fold: (i) We are the first to study energy-based deep internal learning from a single image. (ii) We propose to sequentially train and sample from a pyramid of EBMs with different resolutions in a coarse-to-fine manner for efficient sampling, stable training and powerful representation. (iii) To enhance the training, we propose to train our energy-based framework in parallel with some self-supervised tasks. (iv) We provide strong results in our experiments to verify the effectiveness of the proposed framework in a wide range of image generation and manipulation tasks.

2. Related Work

Energy-based generative models (EBMs) [46, 19, 36] have been widely explored over recent years for representation learning in various domains. By bringing in the power of deep ConvNets, Xie *et al*. [36] propose the Generative ConvNet, which represents the energy function as a convolutional neural network and generates images via MCMC sampling process. Nijkamp *et al*. [27] propose to use a non-convergent short-run MCMC to learn the EBM. However, learning such EBM from high-dimensional data has long been considered as challenging. Thus, various approaches are proposed to assist the training process. For example, the CoopNets [35] trains the EBM jointly with a generator network as an amortized sampler via MCMC teaching; the Multigrid [6] proposes to learn the EBM with multi-grid sampling; Han *et al*. [13] propose triangle divergence that trains the EBM without MCMC by incorporating a Variational Auto-Encoder (VAE) [18]; Xie *et al*. [40] propose to train the EBM with a VAE as an amortized sampler. Recent advances also bring in flow-based models [7] and diffusion recovery likelihood [8]. Our work leverages the previous success on large-scale image datasets and focuses on learning EBMs to represent both global and local statistics of patches within a *single* natural image.

Deep internal learning (DIL) [30] aims at exploiting the internal recurrence of information within natural signals rather than relying on external training data. There are mainly two directions of work. One direction is to exploit the power of deep networks in modeling the internal statistics of the input image. For example, "zero-shot" super-resolution (ZSSR) [30] trains an image-specific CNN from a set of extracted image patches for the super-resolution task. The deep image prior (DIP) [33] shows that a randomly-initialized generator network can be used as a prior distribution for recovering noisy images by conditional generation.

The other direction of DIL is the GAN-based generation, where the internal distribution is implicitly modeled by a generator and trained in an adversarial approach. One recent art that is closely related to us is the SinGAN [28], where a pyramid of multi-scale patch generators and discriminators are trained adversarially from the input image. A similar idea is applied to InGAN [29], which uses a conditional generator that contains a geometric transformation to determine the size/shape of the output. Different from previous approaches, our work seeks to *explicitly* model the internal distribution of a single image by an energy-based framework. Rather than using generators for ancestral sampling, our model generate examples by an iterative MCMC process. Like other works in DIL, our model is learned in a fully *unsupervised* manner.

3. Patchwise Generative ConvNet

3.1. Model Foundation

Let $\mathbf{I}$ denote a training image and p_θ denote a probability density function that approximates the internal statistics of

$\mathbf{I}$, then the patchwise generative ConvNet is defined as

$$p_\theta(\mathbf{I}) = \frac{1}{Z(\theta)} \exp(f_\theta(\mathbf{I})), \tag{1}$$

where $Z(\theta) = \int \exp(f_\theta(\mathbf{I}))d\mathbf{I}$ is the normalization constant, f_θ is a convolutional network denoting the negative energy of $\mathbf{I}$, *i.e.*, $\mathcal{E}(\mathbf{I}) = -f_\theta(\mathbf{I})$. The maximum likelihood estimation (MLE) seeks to find θ to maximize the log-likelihood function of the single image $\mathbf{I}$, *i.e.*,

$$\mathcal{L}(\theta) = \log p_\theta(\mathbf{I}). \tag{2}$$

The gradient of the $\mathcal{L}(\theta)$ with respect to θ is given by

$$\frac{\partial}{\partial \theta}\mathcal{L}(\theta) = \frac{\partial}{\partial \theta} f_\theta(\mathbf{I}) - \mathbb{E}_{\mathbf{I}\sim p_\theta}[\frac{\partial}{\partial \theta} f_\theta(\mathbf{I})], \tag{3}$$

in which the expectation term is analytically intractable and can be approximated by MCMC sampling such as Langevin dynamics [26], which iterates

$$\mathbf{I}_{t+1} = \mathbf{I}_t + \frac{\delta^2}{2}\frac{\partial}{\partial \mathbf{I}} f_\theta(\mathbf{I}_t) + \delta\epsilon_t, \tag{4}$$

where t indexes the time step and δ is the Langevin step size. $\epsilon_t \sim \mathcal{N}(0, I)$ is a Gaussian noise serving as a Brownian motion that is useful to explore different modes.

3.2. Multi-Scale Modeling

In this section, we will extend the model in Eq. (1) to a multi-scale version so that it can capture different scales of internal statistics from the image. Let $\{\mathbf{I}^{(s)}, s = 0, ..., S\}$ denote the multi-scale versions of a training image $\mathbf{I}$, with s indexing the scale, $\mathbf{I}^{(0)}$ representing the minimal scale version of $\mathbf{I}$, and $\mathbf{I}^{(S)}$ representing the original scale version of $\mathbf{I}$. Given a training image $\mathbf{I}$, we can easily create a pyramid of images with different scales of $\mathbf{I}$ by downsampling operations. Thus, $\mathbf{I}^{(s)}$ is a downsampled version of $\mathbf{I}$ by a scaling factor $1/r^{S-s}$, where $r > 1$, or $\mathbf{I}^{(s-1)}$ is a downsampled version of $\mathbf{I}^{(s)}$ by a scaling factor $1/r$.

Our multi-scale model consists of a pyramid of EBMs, which are generative ConvNets $\{p_{\theta_s}(\mathbf{I}^{(s)}), s = 0, ..., S\}$, trained against a pyramid of images $\{\mathbf{I}^{(s)}, s = 0, ..., S\}$. Each $p_{\theta_s}(\mathbf{I}^{(s)})$ is responsible for synthesizing realistic images based on the patch distribution learned from the image $\mathbf{I}^{(s)}$ at the corresponding scale s. This can be accomplished by "analysis by synthesis", in which synthesis examples are produced by Langevin dynamics in Eq. (4) and then the sample average is used to approximate the gradient of the log-likelihood in Eq. (3) for the purpose of updating the parameter θ_s. For $s = 0, ..., S$,

$$\frac{\partial}{\partial \theta_s}\mathcal{L}(\theta_s) = \frac{\partial}{\partial \theta_s} f_{\theta_s}(\mathbf{I}^{(s)}) - \frac{1}{n}\sum_{i=1}^{n}[\frac{\partial}{\partial \theta_s} f_{\theta_s}(\tilde{\mathbf{I}}_i^{(s)})], \tag{5}$$

where $\{\tilde{\mathbf{I}}_i^{(s)}, i = 1, ..., n\}$ are the synthesized images sampled from $p_{\theta_s}(\mathbf{I}^{(s)})$ via Langevin dynamics. The challenge to train our framework might lie in the Langevin sampling from the pyramid of EBMs.

3.3. Multi-Scale Sequential Sampling

As to the MCMC sampling strategy, instead of using a noise initialized long-run Langevin dynamics, we can take full advantage of the multi-scale modeling setting to efficiently produce a pyramid of synthesized images by using finite-step MCMC at each scale, which is initialized from the synthesized image generated at the previous coarser scale. To be specific, let $\tilde{\mathbf{I}}_t^{(s)}$ denote the synthesized image at Langevin time step t from the model at scale s, and $K^{(s)}$ denote the number of Langevin steps for model at scale s. We first initialize $\tilde{\mathbf{I}}_0^{(0)}$ by sampling from the uniform distribution $\mathcal{U}$, and then run $K^{(0)}$ Langevin steps to obtain $\tilde{\mathbf{I}}_{K^{(0)}}^{(0)}$. After that, for scale $s = 1, ..., S$, the up-scaled version of $\tilde{\mathbf{I}}_{K^{(s-1)}}^{(s-1)}$ sampled from the model $p_{\theta_{s-1}}(\mathbf{I}^{(s-1)})$ at the previous coarser scale is used to initialize the finite-step Langevin dynamics that samples from the model $p_{\theta_s}(\mathbf{I}^{(s)})$ at the subsequent finer scale.

Formally, the multi-scale sequential sampling can be presented as follows: for $s = 0, ..., S$,

$$\tilde{\mathbf{I}}_0^{(s)} = \begin{cases} Z \sim \mathcal{U}_d((-1, 1)^d) & s = 0 \\ \text{Upsample}(\tilde{\mathbf{I}}_{K^{(s-1)}}^{(s-1)}) & s > 0 \end{cases} \tag{6}$$

$$\tilde{\mathbf{I}}_{t+1}^{(s)} = \tilde{\mathbf{I}}_t^{(s)} + \frac{\delta^2}{2}\frac{\partial}{\partial \mathbf{I}^{(s)}} f_{\theta_s}(\tilde{\mathbf{I}}_t^{(s)}) + \delta\epsilon_t^{(s)}, \tag{7}$$

where $t = 0, .., K^{(s)} - 1$.

In Eq.(6), $\mathcal{U}_d((-1, 1)^d)$ is the uniform distribution with a closed interval from -1 to 1, and d is the number of dimensions of $\tilde{\mathbf{I}}^{(0)}$. We use Upsample$(\cdot)$ to denote an upsampling operation with a scaling factor r, where $r > 1$, which expands the synthesized images from the coarser scale to the finer scale. The upsampling operation is a pseudo-inverse of the downsampling operation used in creating the image pyramid, given the fact that the up-scaled version of $\mathbf{I}^{(s-1)}$ is not comparable with the original $\mathbf{I}^{(s)}$ due to the loss of high resolution details. The short-run Langevin dynamics at scale s samples $\tilde{\mathbf{I}}^{(s)}$ by creating more high resolution details for the up-scaled $\tilde{\mathbf{I}}^{(s-1)}$, which is much easier than sampling from scratch, especially when s is large.

3.4. Multi-Scale MCMC as a Flow Generator

We can simplify the Eq. (6) and Eq. (7) by rewriting them into the following compact form:

$$Z \sim p_0(Z); \tilde{\mathbf{I}}^{(s)} = M_{\Theta_s}^{(s)}(Z, \boldsymbol{\epsilon}), \tag{8}$$

where p_0 is the prior distribution to initialize the short-run MCMC for the model at the smallest scale, which is set to be a uniform distribution in Eq. (6). We use $\Theta_s = (\theta_0, \theta_1, ..., \theta_s)$ to denote the models from the minimum scale up to scale s, and the synthesized image $\tilde{\mathbf{I}}^{(s)}$ at scale s is only affected by Θ_s. $\boldsymbol{\epsilon}$ denotes all the randomness in the multi-scale short-run MCMC due to the Langevin noise term in Eq. (7). $M^{(s)}$ contains all steps of Langevin updates in synthesizing the image $\tilde{\mathbf{I}}^{(s)}$ at scale s. Thus, $M^{(s)}$ can be viewed as a noise-injected residual network with $\sum_{j=0}^{s} K^{(j)}$ layers, then Z as the latent variables and p_0 as the prior distribution of Z. In general, the model in Eq. (8) depicts an energy-based dynamics to generate a pyramid of synthesized images $\{\tilde{\mathbf{I}}^{(s)}, s = 0, ..., S\}$ from a noise Z. At the convergence of our learning algorithm, for $s = 0, ..., S$, we have $\mathcal{L}'(\theta_s) = 0$, which is

$$\frac{1}{n}\sum_{i=1}^{n}[\frac{\partial}{\partial \theta_s} f_{\theta_s}(\tilde{\mathbf{I}}_i^{(s)})] = \frac{\partial}{\partial \theta_s} f_{\theta_s}(\mathbf{I}^{(s)}). \tag{9}$$

That means the learned parameters $\{\theta_s, s = 0, ..., S\}$ can generate realistic image patterns $\{\tilde{\mathbf{I}}_i^{(s)}, i = 1, ..., n\}$ that match the observed training image in terms of internal statistics $\phi_{\theta_s}(\mathbf{I}^{(s)}) = \frac{\partial}{\partial \theta_s} f_{\theta_s}(\mathbf{I}^{(s)})$, which is defined by the leaned negative energy function f_{θ_s}.

3.5. Self-Supervised Parallel Training

The above framework includes two stages: (1) learning the internal statistics $\{\phi_{\theta_s}(\mathbf{I}^{(s)}), s = 0, ..., S\}$ from a single image, and (2) generating new images based on the learned internal statistics. In other words, the internal statistics is learned in the task of image generation.

Eq. (8) defines an unconditional distribution $p_{\Theta_S}(\mathbf{I}^{(S)})$, which corresponds to an unconditional generation. We can derive the conditional distribution $p_{\Theta_S}(\mathbf{I}^{(S)}|C)$ from $p_{\Theta_S}(\mathbf{I}^{(S)})$. This conditional form of the model can be used for different tasks. For example, (i) if the input condition C is the low resolution version of $\mathbf{I}$, the learned M_{Θ_S} targets the task of super-resolution; (ii) If the condition information C is the noisy version of $\mathbf{I}$, the task will be denoising. These tasks are self-supervised since the low resolution or noisy version of $\mathbf{I}$ can be created by the model itself.

We can learn the internal statistics in the context of these self-supervised tasks by maximizing the conditional log-likelihood of the image given the input condition, *i.e.*,

$$\mathcal{L}_{\text{cond}}(\Theta_S) = \log p_{\Theta_S}(\mathbf{I}^{(S)}|C = c), \tag{10}$$

where c is the observed value of the condition C. The learning and sampling algorithm is essentially the same as maximizing the unconditional log-likelihood in Eq. (5), except that in the sampling step, we need to sample from the conditional distribution, which amounts to using c to initialize the Z in the generation process in Eq. (8).

In this paper, we find that learning our internal statistics with extra self-supervised tasks can not only stabilize the training process but also improve the overall synthesis quality. For example, we can add an auxiliary image super-resolution task, and learn our model simultaneously for random image generation and super-resolution, which means that in addition to starting from uniform white noise, our sequential sampling also starts from the low-resolution image (a downsampled version of the training image) and outputs a super-resolved image that seeks to match the original one. Specifically, for scale 0, we use $\mathbf{I}_{\text{LR}}^{(0)} = \text{Upsample}(\text{Downsample}(\mathbf{I}^{(0)}))$ as the low-resolution (LR) version of $\mathbf{I}^{(0)}$, where $\text{Upsample}(\cdot)$ and $\text{Downsample}(\cdot)$ are upsampling and downsampling operations that use scaling factors r and $1/r$, respectively. Then we treat $c = \mathbf{I}_{\text{LR}}^{(0)}$ as the initial condition in Eq. (10) and the objective maximizes the total log-likelihood:

$$\mathcal{L}_{\text{tot}}(\Theta^S) = \mathcal{L}(\Theta^S) + \lambda \mathcal{L}_{\text{cond}}(\Theta^S), \tag{11}$$

where λ is a hyperparameter that controls the importance of the self-supervised task in the training process. In this experiment, we set $\lambda = 0.1$ and use 8 scales.

4. Experiments

In this section, we first qualitatively present our results and compare our approach against the prior art on DIL. We then study the effectiveness of different modules in the proposed multi-scale training paradigm. Lastly, we demonstrate the capability of our model on various image generation and manipulation tasks. For brevity, we refer to our proposed method as `PatchGenCN`.

4.1. Implementation

Image Preprocessing Given an input image, if the length of its longer edge exceeds 250 pixels, we will first proportionally resize it such that its longer edge fits to 250 pixels. The (resized) input image is denoted by $\mathbf{I}^{(S)}$. We then create $\{\mathbf{I}^{(s)}, s = 0, ..., S-1\}$ by sequentially downsampling the image $\mathbf{I}^{(S)}$ with a properly chosen scaling factor $1/r$ until the length of the shorter edge becomes 25 pixels. For all experiments, we use the Lanczos filter for downsampling and the BiCubic interpolation for upsampling.

Model Architecture Our model contains a single neural network that plays the role of energy function at each scale. We follow [28, 15, 20] to use the Patch ConvNets to capture the internal statistics of overlapping image patches within the entire image. Specifically, the EBM at each scale is parameterized by a ConvNet that consists of five convolutional layers with kernel size 3×3 and stride 1. To stabilize the training process, we use ELU[3] as the activation function and spectral normalization [25] to regularize the parameters in convolutional layers.

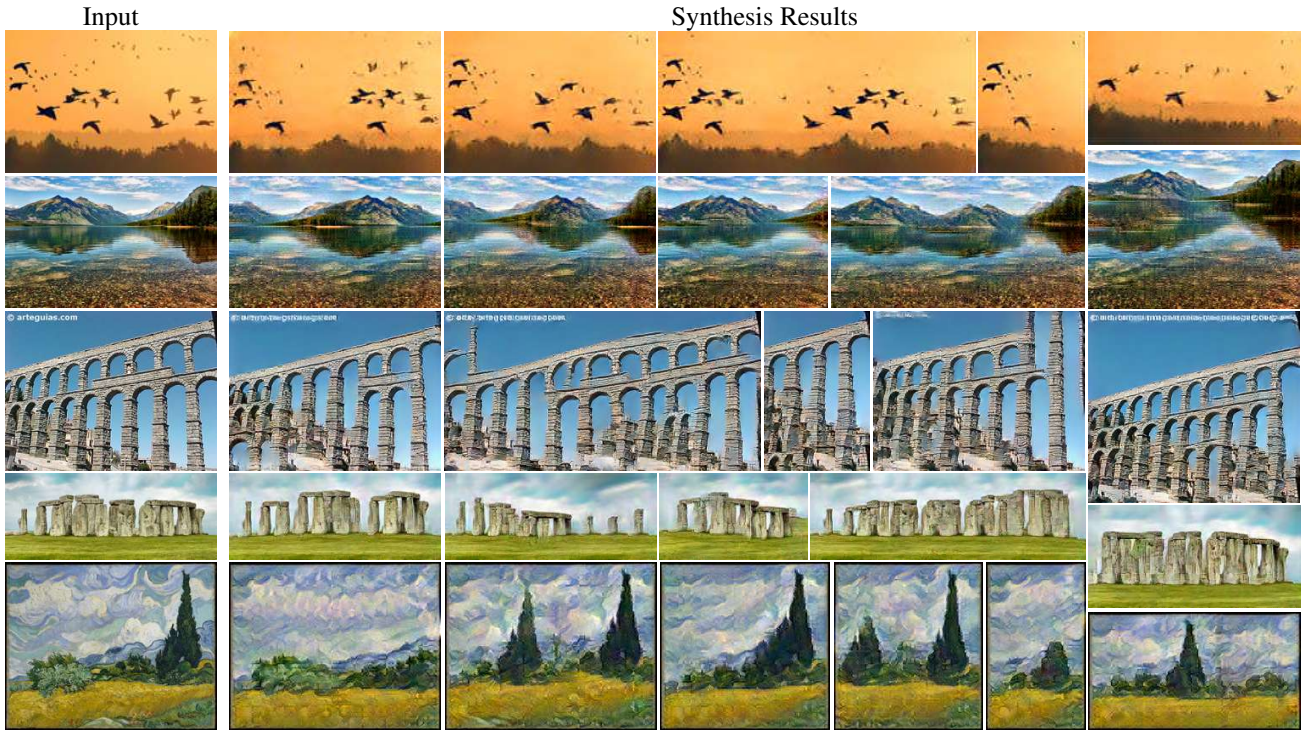

Figure 1: Random Image Synthesis. Each row demonstrates a single training example and multiple synthesis results of various aspect ratios. Our framework is able to generate realistic images with arbitrary sizes and aspect ratios by sampling from the learned distribution that captures different scales of internal statistical properties of patches within the input image.

Training Details We use 60 Langevin steps with step size 0.1 for the EBM at the first scale and use 30 steps for each of the other higher scale EBMs. For all scales of θ_s, we use Adam Optimizer [17] and linearly decay the learning rate from 4×10^{-4} to 5×10^{-5}. Each scale is trained for 4000 epochs or until an early stop criteria is met, *e.g.*, the mean squared error in the self-supervised task is less than 0.001.

4.2. Unconditional Image Generation

4.2.1 Evaluation

We evaluate our proposed method using scene images selected from the Places [44] and LSUN [43] datasets, as well as art images downloaded from WikiArt and Web. Different from image retargeting as in [29], the goal of this task is to generate random samples that match the internal statistical properties of the training image.

Figure 1 qualitatively shows the synthesis results by learning from a single input image. The compelling performance demonstrates that our model is able to capture patchwise statistics and generate realistic images of arbitrary sizes and various aspect ratios. Our observations can be summarized as follows: (i) each sampled result not only contains local repetitive patterns existing in the texture information but also preserves the global spatial layout as shown in the training example; (ii) the results may contain objects that have different sizes or shapes as in the training input, *e.g.*, stones and trees in Figure 1; (iii) results of different sizes are generated by more than resizing the image size, but also matching the statistics within the image patches.

We quantitatively evaluate the realism of our synthesized results using the following metrics:

- **Human study: "Real vs Fake" test** We run "real vs fake" perceptual studies on the generated samples to assess the realism of our results. We follow the same perceptual study protocol from Shaham *et al*. [28] to run both paired studies, where users were asked to find the fake image from a pair of real image and generated sample, and unpaired studies, where users were asked to judge whether a presented image is real or fake. In both cases, we presented the images for 1 second. We gathered data from 25 participants per algorithm we tested. Each participant performed a sequence of 30 trials for paired tests and 60 for unpaired tests with 30 training images and 30 corresponding generated samples.

- **Single Image Fréchet Inception Distance (SIFID)** We follow [28] to adopt SIFID metric, an extension of the Fréchet Inception Distance (FID) [14], to automatically assess the patchwise similarity of inception features be-

tween a generated sample and a single real image. Specifically, rather than resizing images to the size of 299×299 as in computing FID, we feed in an image of its original resolution to the InceptionNet [31] and take the output of layer `Conv2d_2b_3x3`, the last layer of the first convolutional block, to retrieve its patchwise features. Then the distance is computed using the same formula in [14].

- **Naturalness Image Quality Evaluator (NIQE)** [24] Even though SIFID can partially show the realism of the generated samples, its value could suffer from a high variance for different generated results. Therefore, we additionally use NIQE, a no-reference image quality score, to evaluate the overall naturalness of the generated samples. The NIQE score is measured by comparing the statistical features of input images to a corpus of natural, undistorted scene images using a natural scene statistic (NSS) model. Lower NIQE score indicates better image quality with less artifacts.

4.2.2 Comparison Against Baselines

One important baseline method closest to ours is SinGAN [28]. We compare our model with SinGAN in Table 1 using the above metrics. As can be seen, our synthesized results are on par with or better than the generation outputs from SinGAN over all metrics. The lower NIQE score indicates better perceptual quality compared to SinGAN. The numbers reported here are not perfectly aligned with those in [28] because of the difference in the testing images.

In Figure 2, we present an example of comparison of our generation process with that of SinGAN [28]. Our model can generate meaningful results at all scales, while SinGAN may fail at the top few scales. This observation matches the behavior of the multi-scale sampling strategy presented in Section 3.3, *i.e.*, the EBM at the first scale learns to capture the global layout of the training image, while each of the EBMs at the subsequent finer scales learns to enrich the output of the EBM at the previous coarser scale with details.

We compare the model complexity of our method with two GAN-based models in Table 2. For fair comparison, we use 8 scales for all multi-scale architectures. The training time is measured as the sum of average computation time per epoch over all scales, while the inference time is measured as the average duration of generating one sample of the original resolution. Similar to SinGAN, we use light-weight ConvNets for all EBMs, which have much fewer parameters than InGAN. Besides, we show comparable training time to that of SinGAN. The key factors resulting in slowness of our training are the MCMC sampling and spectral normalization [25], while SinGAN takes a slightly longer time because of the iterative computation of gradient

Methods	SIFID	real vs fake paired	real vs fake unpaired	NIQE
SinGAN [28]	0.11	35.73%	39.33%	5.22
PatchGenCN (ours)	0.09	33.60%	40.13%	5.10

Table 1: Quantitative evaluation on images from the Places [44] dataset. The values for the "real vs fake" indicate the percentage of participants who label the generated samples as *real* ones.

Methods	# Params	Train Time (sec./epoch)	Inf. Time (sec.)
SinGAN [28]	1.15 M	1.58	0.04
InGAN [29]	6.81 M	0.20	0.05
PatchGenCN (ours)	0.99 M	1.50	0.72

Table 2: Model complexity comparisons of our method with GAN-based methods, measured as an average of 10 images of size 250×166 pixels on a single RTX 2080 GPU.

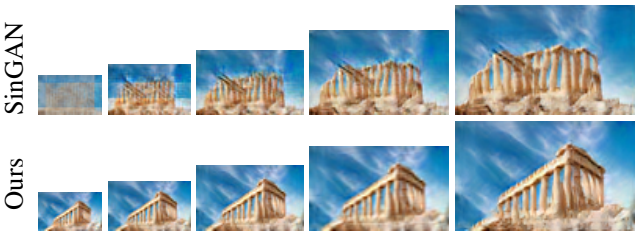

Figure 2: Comparison of the coarse-to-fine sequential generation between our model and SinGAN [28].

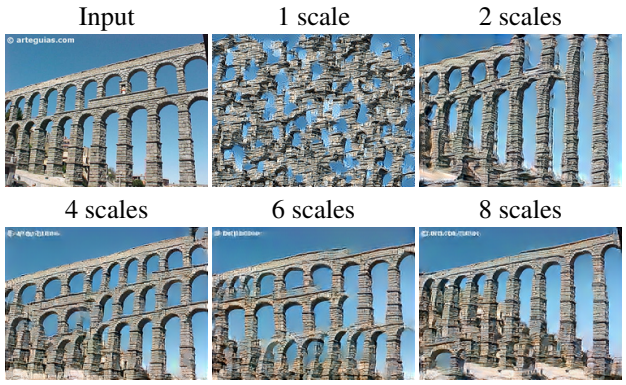

Figure 3: Synthesis results with different numbers of scales.

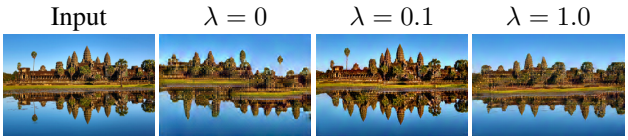

Figure 4: Synthesis results using different values of the importance factor λ of the self-supervised task.

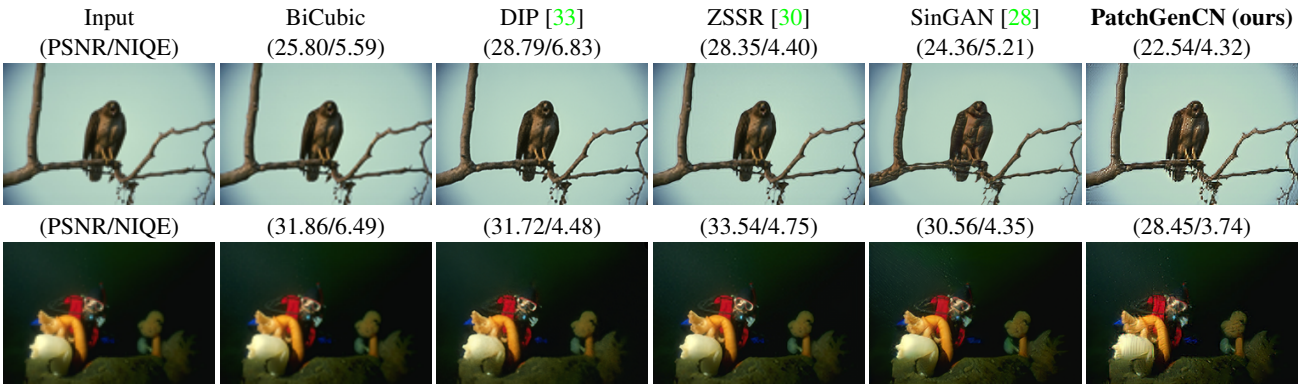

Figure 5: Comparison on super-resolution task with baseline models. The first column shows the low resolution training images and the rest columns display the $4\times$ super-resolution results. We use NIQE [24] to measure the visual quality of the super-resolved results. PSNR between the generated result and the real high resolution image is also reported for reference.

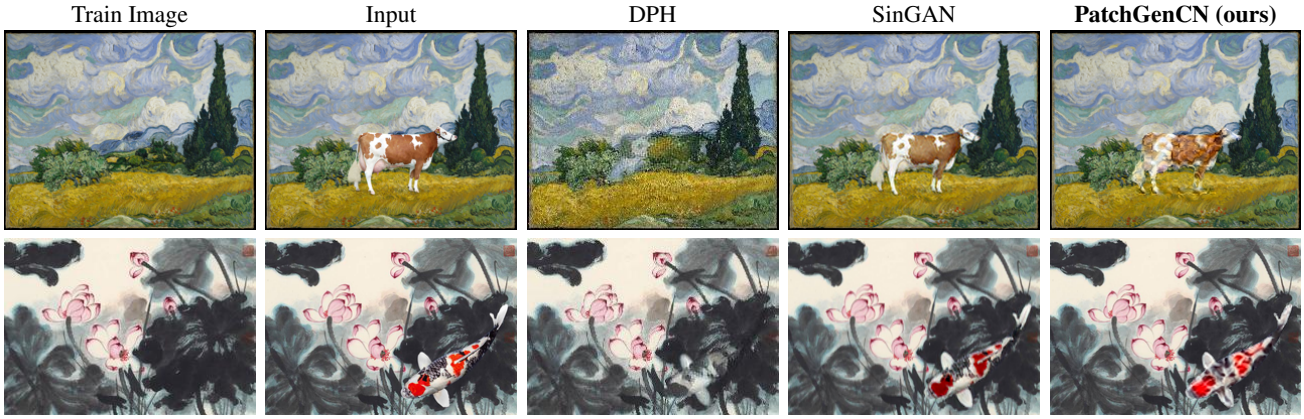

Figure 6: We qualitatively compare our methods with Deep Painting Harmonization (DPH) [32] and SinGAN [28] on image harmonization task. Our model is able to blend the original image with an external object by preserving the object's identity and applying the learned texture and color information to the object.

penalty. As to inference, our model takes a bit longer than GAN-based methods due to the usage of MCMC.

4.2.3 Ablation Studies

We run ablation studies to evaluate the effectiveness of different modules in our framework.

Number of total scales We show the generation results using different numbers of scales in Figure 3. When the model is trained with a single scale, the generated result is basically a texture image, where image patches are randomly distributes. When using 2 scales, our model can create a coarse structure, however, the details are still missing because internal statistics at other scales are not learned. We can see more details as the number of total scales increases.

Effectiveness of parallel self-supervised training Figure 4 shows the results using different values of λ in Eq. (11), where $\lambda = 0$ indicates that the model is trained without extra self-supervised tasks. We can see that the quality of the synthesized images improves by adding a self-supervised task. Experiments show that using a λ either too large or too small will lead to an unstable training process.

4.3. Super-Resolution

Our model can increase the resolution of the input image by a factor r^k, $k \in \mathbb{N}$, without relying on any external training data. We first train the model on the input image with a scaling factor r. We only need to use the trained EBM f_{θ_S} at the original scale S for the task of super-resolution. We start from the up-scaled input image $\text{Upsample}(\mathbf{I}^{(S)})$ and then run the multi-scale sequential sampling for $s = S+1, ..., S+k$ by following the same process introduced in Eqs.(6) and (7), except that we only use f_{θ_S} for all $s > S$. Figure 5 shows some $4\times$ super-resolution

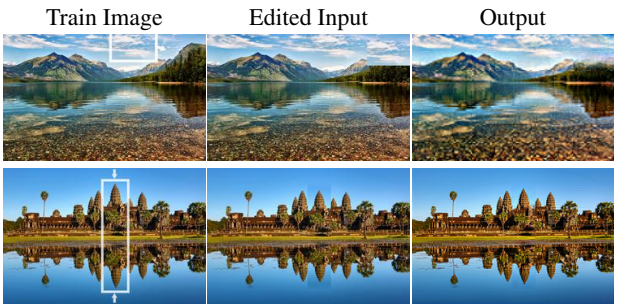

Figure 7: Image Editing.

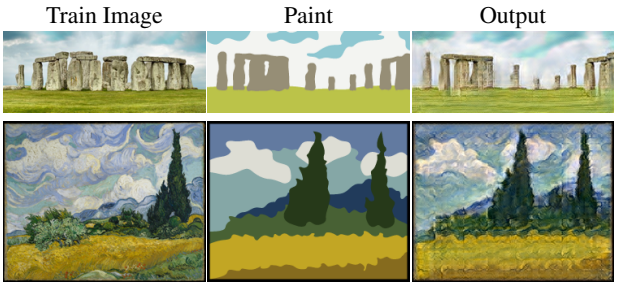

Figure 8: Paint to Image

(a) Genearated results with different style-content pairs.

(b) Comparison with neural style transfer [9].

Figure 9: Style Transfer.

results using images from the BSD100 [23] dataset. We follow [28, 2] to use NIQE [24] as the major quantitative metric for evaluating the visual perception quality. Our model outperforms the prior art of deep internal learning in terms of NIQE, since it is able to produce more details.

4.4. Image Manipulation

Given a background image $\mathbf{I}^{(S)}$, we can manipulate it by either copying and moving some region or pasting an external object in it. The resulting edited image is denoted by $\mathbf{I}'^{(S)}$. Our model can blend the pasted object with the original background image or smooth the artifacts due to editing. We first train our model on $\mathbf{I}^{(S)}$ and obtain the pyramid of EBMs. We then create the down-scaled version of the edited image $\mathbf{I}'^{(\hat{s})}$, where $0 < \hat{s} < S$ is an intermediate scale such that $\mathbf{I}'^{(\hat{s})}$ will not loss so many details. We run the multi-scale sequential sampling with $\{f_{\theta_s}, s = \hat{s}, ..., S\}$, starting from $\tilde{\mathbf{I}}_0^{(\hat{s})} \leftarrow \mathbf{I}'^{(\hat{s})}$. The synthesized output $\tilde{\mathbf{I}}^{(S)}$ is the harmonized result of $\mathbf{I}'^{(S)}$. Typically, we set $\hat{s} = S - 4$ or $\hat{s} = S - 5$. In Figure 6, we compare our results with baselines on the image harmonization task. Our model can apply more texture information from the background image to the objects than [28], while preserving better object identity than [32]. Similar qualitative results can also be seen in Figure 7 for image editing. We find that more advanced editing is also applicable. If the edited image is a painting clipart that specifies the layout of semantic objects, then the resulting synthesis will be an image, where the global structure of the painting is preserved, while the texture matching the background image. See Figure 8 for some examples.

4.5. Style Transfer

Given a style image $\mathbf{I}^{(S)}$ and a content image $\mathbf{I}_c^{(S)}$, we can learn our model to stylize $\mathbf{I}_c^{(S)}$ with the style in $\mathbf{I}^{(S)}$, while preserving the content's identity. During the training of an unconditional model $p_{\Theta_S}(\mathbf{I}^{(S)})$ on $\mathbf{I}^{(S)}$, we add an extra conditional generation task to simultaneously learn $p_{\Theta_S}(\mathbf{I}^{(S)}|\mathbf{I}_c^{(S)})$ for style transfer, which can be done by downsampling $\mathbf{I}_c^{(S)}$ and using it to initialize the Langevin at one of the coarser scales. Specifically, when training EBM at scale s, we chose the coarser scale $\max(0, s - n)$, where $0 < n < S$, as the starting point to generate the stylized image. Figure 9 shows that our model can convert images to artistic styles while preserving the content better than [9].

5. Conclusion

We propose the `PatchGenCN`, a novel multi-scale patchwise energy-based framework with a bottom-up ConvNet serving as the energy function at each scale, for learning the internal distribution within a single natural image. Compelling performance demonstrates the powerful capability of our model on capturing internal patchwise statistics within a single image and generating realistic images on various image generation and manipulation tasks. Our model is appealing because it integrates the representation and generation into one single framework. Our paper pushes the boundaries of deep internal learning.

References

[1] Adrian Barbu and Song-Chun Zhu. *Monte Carlo Methods*. Springer, 2020. 2

[2] Yochai Blau and Tomer Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6228–6237, 2018. 8

[3] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *International Conference on Learning Representations (ICLR)*, 2016. 4

[4] Yilun Du, Joshua Meier, Jerry Ma, Rob Fergus, and Alexander Rives. Energy-based models for atomic-resolution protein conformations. In *International Conference on Learning Representations (ICLR)*, 2019. 1

[5] Yilun Du and Igor Mordatch. Implicit generation and modeling with energy based models. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3608–3618, 2019. 1

[6] Ruiqi Gao, Yang Lu, Junpei Zhou, Song-Chun Zhu, and Ying Nian Wu. Learning generative convnets via multi-grid modeling and sampling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9155–9164, 2018. 1, 2

[7] Ruiqi Gao, Erik Nijkamp, Diederik P Kingma, Zhen Xu, Andrew M Dai, and Ying Nian Wu. Flow contrastive estimation of energy-based models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7518–7528, 2020. 2

[8] Ruiqi Gao, Yang Song, Ben Poole, Ying Nian Wu, and Diederik P Kingma. Learning energy-based models by diffusion recovery likelihood. In *International Conference on Learning Representations (ICLR)*, 2021. 2

[9] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015. 8

[10] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, (6):721–741, 1984. 1

[11] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2672–2680, 2014. 2

[12] Will Grathwohl, Kuan-Chieh Wang, Jörn-Henrik Jacobsen, David Duvenaud, Mohammad Norouzi, and Kevin Swersky. Your classifier is secretly an energy based model and you should treat it like one. *International Conference on Learning Representations (ICLR)*, 2020. 1

[13] Tian Han, Erik Nijkamp, Xiaolin Fang, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Divergence triangle for joint training of generator model, energy-based model, and inferential model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8670–8679, 2019. 2

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6626–6637, 2017. 5, 6

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1125–1134, 2017. 4

[16] Bela Julesz. Visual pattern discrimination. *IRE transactions on Information Theory*, 8(2):84–92, 1962. 1

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2015. 5

[18] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *International Conference on Learning Representations (ICLR)*, 2014. 2

[19] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1(0), 2006. 1, 2

[20] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision (ECCV)*, pages 702–716. Springer, 2016. 4

[21] Jun S Liu. *Monte Carlo Strategies in Scientific Computing*. Springer Science & Business Media, 2008. 2

[22] David Marr and Tomaso Poggio. A computational theory of human stereo vision. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 204(1156):301–328, 1979. 1

[23] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 416–423. IEEE, 2001. 8

[24] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012. 6, 7, 8

[25] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *International Conference on Learning Representations (ICLR)*, 2018. 4, 6

[26] Radford M Neal et al. Mcmc using hamiltonian dynamics. *Handbook of Markov Chain Monte Carlo*, 2(11):2, 2011. 3

[27] Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. Learning non-convergent non-persistent short-run mcmc toward energy-based model. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5232–5242, 2019. 1, 2

[28] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4570–4580, 2019. 1, 2, 4, 5, 6, 7, 8

[29] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. Ingan: Capturing and retargeting the "dna" of a natural im-

age. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 4492–4501, 2019. 2, 5, 6

[30] Assaf Shocher, Nadav Cohen, and Michal Irani. "zero-shot" super-resolution using deep internal learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3118–3126, 2018. 1, 2, 7

[31] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. 6

[32] Yi-Hsuan Tsai, Xiaohui Shen, Zhe Lin, Kalyan Sunkavalli, Xin Lu, and Ming-Hsuan Yang. Deep image harmonization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3789–3797, 2017. 7, 8

[33] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9446–9454, 2018. 1, 2, 7

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

[35] Jianwen Xie, Yang Lu, Ruiqi Gao, and Ying Nian Wu. Cooperative learning of energy-based model and latent variable model via mcmc teaching. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 32, 2018. 2

[36] Jianwen Xie, Yang Lu, Song-Chun Zhu, and Yingnian Wu. A theory of generative convnet. In *International Conference on Machine Learning (ICML)*, pages 2635–2644, 2016. 1, 2

[37] Jianwen Xie, Yifei Xu, Zilong Zheng, Song-Chun Zhu, and Ying Nian Wu. Generative pointnet: Energy-based learning on unordered point sets for 3d generation, reconstruction and classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 1

[38] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, and Ying Nian Wu. Learning descriptor networks for 3d shape synthesis and analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 1

[39] Jianwen Xie, Zilong Zheng, Ruiqi Gao, Wenguan Wang, Song-Chun Zhu, and Ying Nian Wu. Generative VoxelNet: learning energy-based models for 3D shape synthesis and analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020. 1

[40] Jianwen Xie, Zilong Zheng, and Ping Li. Learning energy-based model with variational auto-encoder as amortized sampler. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2021. 2

[41] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Synthesizing dynamic patterns by spatial-temporal generative convnet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7093–7101, 2017. 1

[42] Jianwen Xie, Song-Chun Zhu, and Ying Nian Wu. Learning energy-based spatial-temporal generative convnets for dynamic patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 43(2):516–531, 2019. 1

[43] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, 2015. 5

[44] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in Neural Information Processing Systems (NIPS)*, pages 487–495, 2014. 5, 6

[45] Song Chun Zhu, Xiu Wen Liu, and Ying Nian Wu. Exploring texture ensembles by efficient markov chain monte carlo-toward a 'trichromacy' theory of texture. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(6):554–569, 2000. 1

[46] Song Chun Zhu, Yingnian Wu, and David Mumford. Filters, random fields and maximum entropy (frame): Towards a unified theory for texture modeling. *International Journal of Computer Vision (IJCV)*, 27(2):107–126, 1998. 1, 2

Appendix

We will provide full descriptions of the training and sampling algorithms and details about architecture design of the energy function to support the paper.

A. Algorithm Description

We provide the descriptions of the proposed learning and sampling algorithms in Algorithm 1 (illustrated by Figure 10) and Algorithm 2 (illustrated by Figure 11), respectively. Algorithm 1 presents the multi-scale sequential training of the pyramid of energy-based models, where the multi-scale sequential sampling presented in Algorithm 2 is used for efficient MCMC generation to compute the update gradients.

Algorithm 1 Multi-scale sequential training

Input:
(1) A single training image $\mathbf{I}$
(2) Numbers of Langevin steps at different scales $\{K^{(s)}, s = 0, ..., S\}$
Output:
(1) Model parameters $\{\theta^{(s)}, s = 0, ..., S\}$
(2) Different scales of synthesized images $\{\tilde{\mathbf{I}}^{(s)}, s = 0, ..., S\}$

Create multi-scale versions of the training image $\{\mathbf{I}^{(s)}, s = 0, ..., S\}$ by downsampling operation.
for $s = 0$ to S **do**
  repeat
    Sample $\{\tilde{\mathbf{I}}_i^{(s)}, i = 1, ..., n\}$ from the model at scale s by Algorithm 2
    Update θ_s according to Eq.(5) using Adam optimizer.
  until converged.
end for

Algorithm 2 Multi-scale sequential sampling

Input:
(1) The scale s' of the model that need to be sampled
(2) Numbers of Langvein steps $\{K^{(s)}, s = 0, ..., s'\}$
(3) Learned model parameters $\{\theta^{(s)}, s = 0, ..., s'\}$
Output:
(1) Synthesized image $\tilde{\mathbf{I}}^{(s')}$ at scale s'

for $s = 0$ to s' **do**
  if $s = 0$ **then**
    Initialize $\tilde{\mathbf{I}}_0^{(s)}$ with $\mathcal{U}_d((-1, 1)^d)$
  else
    Initialize $\tilde{\mathbf{I}}_0^{(s)}$ with Upsample$(\tilde{\mathbf{I}}_{K^{(s-1)}}^{(s-1)})$
  end if
  for $t = 0$ to $K^{(s)} - 1$ **do**
    Update $\tilde{\mathbf{I}}_{t+1}^{(s)}$ according to Eq.(7).
  end for
end for

B. Model Architecture

Table 3 shows the network structures of EBMs at different scales. Each model consists of five Conv2D layers with 3×3 kernel size. We add spatial zero paddings to the input and use padding size 0 for all convolutional layers. We use the Spectral Normalization to regularize the Conv2D parameters and ELU as the activation function. Parameters are initialized from a Gaussian distribution $\mathcal{N}(0, 0.005)$.

Table 3: Model architectures of various image scales. w and h correspond to the width and the height of the scaled training image, respectively.

(a) $\max(w, h) < 64$.	(b) $\max(w, h) \geq 64$.
ZeroPadding2D((5, 5))	ZeroPadding2D((5, 5))
3×3 Conv2D, 64, ELU	3×3 Conv2D, 128, ELU
3×3 Conv2D, 32, ELU	3×3 Conv2D, 64, ELU
3×3 Conv2D, 32, ELU	3×3 Conv2D, 64, ELU
3×3 Conv2D, 32, ELU	3×3 Conv2D, 64, ELU
3×3 Conv2D, 1	3×3 Conv2D, 1

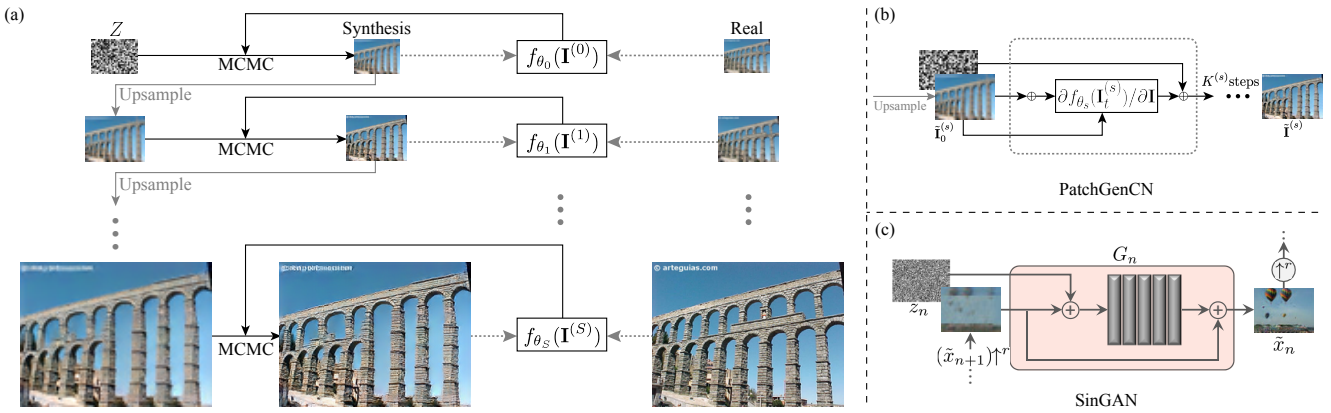

Figure 10: Learning framework of the multi-scale Patchwise Generative ConvNet (PatchGenCN). (a) Illustration of coarse-to-fine multi-scale learning and sampling procedure. Our model parameterizes the energy function by a convolutional network f_{θ_s} at each scale s. Z indicates an image initialized from the uniform white noise. The solid arrows in black indicate the multi-scale MCMC sampling paradigm; the dashed arrows in grey indicate the parameter updates; and the solid arrows in grey indicate the image upsampling operations. (b) Illustration of $K^{(s)}$-step Langevin sampling at scale s. $\oplus$ indicates the elementwise addition operation. (c) Illustration of single-scale generation of SinGAN, where the image synthesis is performed by the top-down generator G. Compared with (c), the sampling process in (b) is derived from the bottom-up energy function f_{θ_s}, and performed in an iterative way. Such a sampling process can be interpreted as a noise-injected $K^{(s)}$-layer residual generator network.

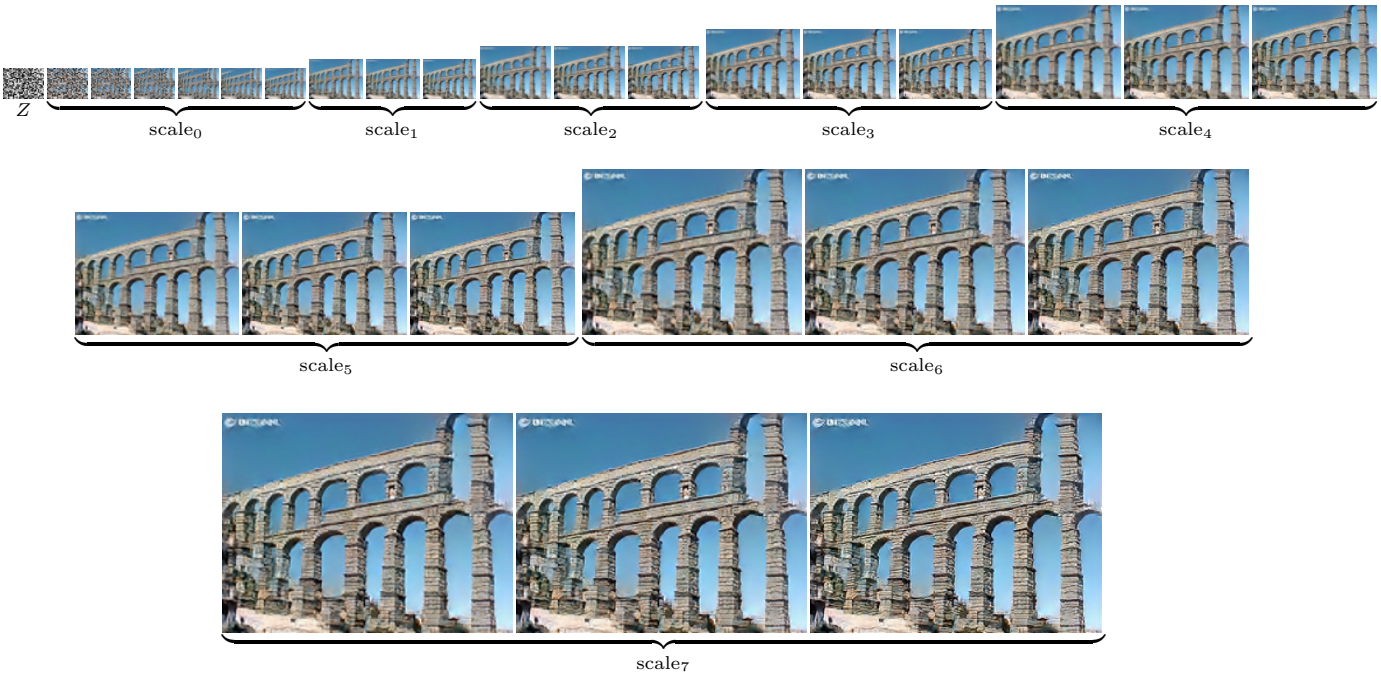

Figure 11: Multi-scale sequential sampling process starting from a randomly initialized noise image Z with the minimum scale. For each scale s, the initial synthesis is updated by $K^{(s)}$ steps of Langevin revision. We visualize a sampled image every 10 Langevin steps for each scale. Except that the initial synthesis at scale 0 is from uniform distribution, the Langevin dynamics of any other scale is initialized from the upsampled version of the Langevin output at its previous scale.