Learning Energy-Based Models by Diffusion Recovery Likelihood

Ruiqi Gao  
UCLA  
ruiqigao@ucla.edu

Yang Song  
Stanford University  
yangsong@cs.stanford.edu

Ben Poole  
Google Brain  
pooleb@google.com

Ying Nian Wu  
UCLA  
ywu@stat.ucla.edu

Diederik P. Kingma  
Google Brain  
durk@google.com

Abstract

While energy-based models (EBMs) exhibit a number of desirable properties, training and sampling on high-dimensional datasets remains challenging. Inspired by recent progress on diffusion probabilistic models, we present a diffusion recovery likelihood method to tractably learn and sample from a sequence of EBMs trained on increasingly noisy versions of a dataset. Each EBM is trained by maximizing the recovery likelihood: the conditional probability of the data at a certain noise level given their noisy versions at a higher noise level. The recovery likelihood objective is more tractable than the marginal likelihood objective, since it only requires MCMC sampling from a relatively concentrated conditional distribution. Moreover, we show that this estimation method is theoretically consistent: it learns the correct conditional and marginal distributions at each noise level, given sufficient data. After training, synthesized images can be generated efficiently by a sampling process that initializes from a spherical Gaussian distribution and progressively samples the conditional distributions at decreasingly lower noise levels. Our method generates high fidelity samples on various image datasets. On unconditional CIFAR-10 our method achieves FID 9.60 and inception score 8.58, superior to the majority of GANs. Moreover, we demonstrate that unlike previous work on EBMs, our long-run MCMC samples from the conditional distributions do not diverge and still represent realistic images, allowing us to accurately estimate the normalized density of data even for high-dimensional datasets.

1 Introduction

Energy-based models \cite{lecun2006energy, ngiam2011multimodal} are an expressive family of probabilistic models that have attracted much attention in recent research of deep generative models \cite{kim2016variational, zhao2016energy, goyal2017focal, xie2016discriminative, finn2016energy, gao2018energy, kumar2019energy, nijkamp2019energy, du2019density, gradwohl2019energy, desjardins2011energy, gao2020energy, che2020energy, gradwohl2020energy, qiu2019energy, rhodes2020energy}. They can be easily parameterized with discriminative models \cite{jin2017learning, Lazarow2017, lee2018learning, gradwohl2020energy}, and trained without data labels. Despite having multiple advantages, two challenges remain for training EBMs on high-dimensional datasets. First, learning EBMs by maximum likelihood requires Markov chain Monte Carlo (MCMC) sampling from the model, which is typically computationally prohibitive. Second, as pointed out in \cite{nijkamp2019energy}, the energy potentials learned with non-convergent MCMC do not have a valid steady-state, in the sense that samples from long-run Markov chains can differ greatly from observed samples, making it difficult to evaluate the learned energy potentials.

To improve the performance of EBMs, we leverage insights from recent work on diffusion probabilistic models \cite{sohl2015deep, ho2020denoising} and score-based generative models \cite{song2019generative, song2020score}. This line of work diffuses data into noise with a sequence of...
Figure 1: Generated samples on LSUN 128\(^2\) church\_outdoor (left), LSUN 128\(^2\) bedroom (center) and CelebA 64\(^2\) (right).

Gaussian distributions at increasing scales, and then learns to reverse this perturbation process by training a sequence of models to reverse each step of the noise corruption. After training such a sequence of models, we can obtain samples from Gaussian white noise by sampling from each model sequentially with decreasing noise scales. These methods have demonstrated great success in applications such as image generation (Ho et al., 2020; Song & Ermon, 2020) and audio synthesis (Chen et al., 2020; Kong et al., 2020).

Inspired by Sohl-Dickstein et al. (2015) and Ho et al. (2020), we propose to train EBMs with diffusion recovery likelihood, a better method than training them directly on a dataset with the standard likelihood. Specifically, we perturb the dataset with a sequence of noise distributions, and learn a sequence of EBMs to model the marginal distributions of the perturbation process. The sequence of EBMs are learned by maximizing recovery likelihoods, which are the densities of conditional distributions that reverse each step of the perturbation process. Compared to standard maximum likelihood estimation (MLE) of EBMs, learning marginal EBMs with recovery likelihoods only require sampling from conditional distributions, which is arguably much easier than sampling from marginal distributions (Bengio et al., 2014). After learning all marginal EBMs, we can generate image samples by starting from the Gaussian white noise, and then produce samples from each conditional distribution in the descending order of noise scales.

Unlike Ho et al. (2020) where the reverse conditional models are parameterized with normal distributions, in our case the conditional models are derived from the marginal EBMs, which are much more flexible. Our method has similarities to Bengio et al. (2013) where the same recovery likelihood objective is used but with a single noise level and without EBMs, leading to different theoretical properties. Importantly, the model in Bengio et al. (2013) does not directly estimate a marginal distribution, while we learn a the sequence of EBMs to model the marginal distributions of the perturbation process. Rhodes et al. (2020) also propose to train EBMs based on a series of intermediate distributions, but their training approach is a variant of noise contrastive estimation—not a likelihood-based approach like ours.

We demonstrate the efficacy of diffusion recovery likelihood on CIFAR-10, CelebA and LSUN datasets. The generated samples are of high fidelity and comparable to GAN-based methods. On CIFAR-10, we achieve FID 9.60 and inception score 8.58, exceeding existing methods of learning explicit EBMs to a large extent. We also demonstrate that diffusion recovery likelihood outperforms denoising score matching from diffusion data if we naively take the gradients of explicit energy functions as the score functions. More interestingly, by using a thousand diffusion time steps, we demonstrate that even very long MCMC chains from the sequence of conditional distributions produce samples that represent realistic images. With the faithful long-run MCMC samples from the
conditional distributions, we can accurately estimate the marginal partition function at zero noise level by importance sampling, and thus evaluate the normalized density of the data under the EBM.

2 BACKGROUND

Let $x \sim p_{\text{data}}(x)$ denote a training example, and $p_\theta(x)$ denote a model’s probability density function that aims to approximates $p_{\text{data}}(x)$. An energy-based model (EBM) is defined as:

$$p_\theta(x) = \frac{1}{Z_\theta} \exp(f_\theta(x)), \quad (1)$$

where $Z_\theta = \int \exp(f_\theta(x)) dx$ is the partition function, which is typically intractable to compute for high-dimensional $x$. For images, we parameterize $f_\theta(x)$ with a convolutional neural network with a scalar output.

The energy-based model in equation (1) can, in principle, be learned through MLE. Specifically, suppose we observe samples $x_i \sim p_{\text{data}}(x)$ for $i = 1, 2, \ldots, n$. The log-likelihood function is

$$\mathcal{L}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i) \doteq \mathbb{E}_{x \sim p_{\text{data}}} [\log p_\theta(x)]. \quad (2)$$

In MLE, we seek to maximize the log-likelihood function, where the gradient approximately follows (Younes, 1999; Xie et al., 2016b)

$$\frac{\partial}{\partial \theta} \mathbb{E}_{p_{\text{data}}} [\log p_\theta(x)] = \mathbb{E}_{x \sim p_{\text{data}}} \left[ \frac{\partial}{\partial \theta} f_\theta(x) \right] - \mathbb{E}_{x \sim p_\theta} \left[ \frac{\partial}{\partial \theta} f_\theta(x) \right]. \quad (3)$$

The expectations can be approximated by averaging w.r.t. the observed samples and the synthesized samples drawn from the model distribution $p_\theta(x)$ respectively. Generating synthesized samples from $p_\theta(x)$ can be done with Markov chain Monte Carlo (MCMC) such as Langevin dynamics (or Hamiltonian Monte Carlo (Giroldi & Calderhead, 2011)), which iterates

$$x^{\tau+1} = x^{\tau} + \delta^2 \nabla_x f_\theta(x^{\tau}) + \delta \epsilon^{\tau}, \quad (4)$$

where $\tau$ indexes the time, $\delta$ is the step size, and $\epsilon^{\tau} \sim \mathcal{N}(0, I)$. However, for multi-model distributions on high-dimensional data, MCMC sampling can take a long time to converge, and the sampling chains may have difficulty traversing modes. We provide an example in Figure 3, where training EBMs with MCMC samples results in malformed energy landscapes (Nijkamp et al., 2019b), even if these samples look reasonable. Alleviating this problem requires advanced techniques, such as coupled MCMC (Qiu et al., 2019), to debias the estimated gradient over synthesized samples.
3 Recovery Likelihood

3.1 From Marginal to Conditional

In order to address the difficulty of sampling from \( p_\theta(x) \), we consider the recovery likelihood similar to Bengio et al. (2013), defined by the density of the sample conditioned on its noisy version perturbed by Gaussian noise. Specifically, let \( \tilde{x} = ax + \sigma \epsilon \) be the noisy sample of \( x \), where \( a \) is a positive coefficient, and \( \epsilon \sim \mathcal{N}(0, I) \). For ease of presentation we hereafter assume \( a = 1 \), but our discussion can be easily generalized to arbitrary \( a \). Suppose \( p_\theta(x) \) is defined by the EBM in equation (1) then the corresponding conditional probability of \( x \) given \( \tilde{x} \) can be derived as being

\[
p_\theta(x|\tilde{x}) = \frac{1}{Z_\theta(\tilde{x})} \exp \left( f_\theta(x) - \frac{1}{2\sigma^2} \| \tilde{x} - x \|^2 \right),
\]

(5)

where \( Z_\theta(\tilde{x}) = \int \exp \left( f_\theta(x) - \frac{1}{2\sigma^2} \| x - \tilde{x} \|^2 \right) dx \) is the partition function of this conditional EBM. See Appendix A.1 for the derivation. Compared to \( p_\theta(x) \) (equation 1), the extra quadratic term \( \frac{1}{2\sigma^2} \| \tilde{x} - x \|^2 \) in \( p_\theta(x|\tilde{x}) \) constrains the density to be localized around \( \tilde{x} \), making the density less multi-modal and easier to sample from. As we will show later, when \( \sigma \) is small, \( p_\theta(x|\tilde{x}) \) is approximately a single mode Gaussian distribution, which greatly reduces the burden of MCMC.

3.2 Maximizing Recovery Likelihood

With the conditional EBM, assume we have observed samples \( x_i \sim p_{data}(x) \) and the corresponding perturbed samples \( \tilde{x}_i = x_i + \sigma \epsilon_i \) for \( i = 1, ..., n \). We define the recovery log-likelihood function as

\[
\mathcal{J}(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i|\tilde{x}_i).
\]

(6)

The term recovery indicates that we attempt to recover the clean sample \( x_i \) from the noisy sample \( \tilde{x}_i \). Thus, instead of maximizing \( \mathcal{L}(\theta) \) in equation 2 we can maximize \( \mathcal{J}(\theta) \), whose corresponding distributions are easier to sample from. Specifically, we generate approximate samples from \( p_\theta(x|\tilde{x}) \) by \( K \) steps of Langevin dynamics that iterates according to

\[
x^{\tau+1} = x^{\tau} + \frac{\delta^2}{2} (\nabla_x f_\theta(x^{\tau}) + \frac{1}{\sigma^2} (\tilde{x} - x^{\tau})) + \delta \epsilon^{\tau}.
\]

(7)

The model is then updated following the same learning gradients as MLE (equation 3), because the quadratic term \( -\frac{1}{2\sigma^2} \| \tilde{x} - x \|^2 \) is not a function of \( \theta \). Following the classical analysis of MLE, we can show that the point estimate given by maximizing recovery likelihood is a consistent estimator of the true parameters, which means that given enough data, a rich enough model and exact sampling, maximizing the recovery likelihood learns \( \theta \) such that \( p_{data}(x) = p_\theta(x) \). See Appendix A.2 for a theoretical explanation.

3.3 Normal Approximation to Recovery Likelihood

When the variance of perturbed noise \( \sigma^2 \) is small, \( p_\theta(x|\tilde{x}) \) can be approximated by a normal distribution via a first order Taylor expansion of \( f_\theta \) at \( \tilde{x} \). Specifically, the negative conditional energy is

\[
-\mathcal{E}_\theta(x|\tilde{x}) = f_\theta(x) - \frac{1}{2\sigma^2} \| \tilde{x} - x \|^2
\]

\[
\approx f_\theta(\tilde{x}) + \langle \nabla_x f_\theta(\tilde{x}), x - \tilde{x} \rangle - \frac{1}{2\sigma^2} \| \tilde{x} - x \|^2
\]

\[
= -\frac{1}{2\sigma^2} \left( \| x - (\tilde{x} + \sigma^2 \nabla_x f_\theta(\tilde{x})) \|^2 \right) + c,
\]

where \( c \) is a constant with respect to \( x \) (see Appendix A.3 for a detailed derivation). In the above approximation, we do not perform second order Taylor expansion because \( \sigma^2 \) is small, and \( \| x - \tilde{x} \|^2 / 2\sigma^2 \) will dominate all the second order terms from Taylor expansion. Thus we can approximate \( p_\theta(x|\tilde{x}) \) by a Gaussian approximation \( \tilde{p}_\theta(x|\tilde{x}) \):

\[
\tilde{p}_\theta(x|\tilde{x}) = \mathcal{N} \left( x; \tilde{x} + \sigma^2 \nabla_x f_\theta(\tilde{x}), \sigma^2 I \right).
\]

(11)
We can sample from this distribution using:

$$x_{\text{gen}} = \tilde{x} + \sigma^2 \nabla_x f_\theta(\tilde{x}) + \sigma \epsilon,$$

(12)

where $\epsilon \sim \mathcal{N}(0, I)$. This resembles a single step of Langevin dynamics, except that $\sigma \epsilon$ is replaced by $\sqrt{2\sigma \epsilon}$ in Langevin dynamics. This normal approximation has two implications: (1) it verifies the fact that the conditional density $p_\theta(x|\hat{x})$ can be generally easier to sample from when $\sigma$ is small; (2) it provides hints for choosing the step size of Langevin dynamics, as discussed in section 3.5.

3.4 Connection to variational inference and score matching

The normal approximation to the conditional distribution leads to a natural connection to diffusion probabilistic models (Sohl-Dickstein et al., 2015; Ho et al., 2020) and denoising score matching with Langevin dynamics (Song & Ermon, 2019; 2020). Specifically, instead of modeling $p_\theta(x)$ as an energy-based model, previous work on diffusion probabilistic models employs variational probabilistic models (Sohl-Dickstein et al., 2015; Ho et al., 2020) and denoising score matching with Langevin dynamics when $\sigma$ is sufficiently small (see Appendix A.5 for details). We can further show that the learning gradient of maximizing log-likelihood of the normal approximation is approximately the same as the learning gradient of maximizing the original recovery log-likelihood with one step of Langevin dynamics (see Appendix A.5). As a result, the training process of maximizing recovery likelihood agrees with diffusion probabilistic models and denoising score matching with Langevin dynamics when $\sigma$ is small.

As the normal approximation is accurate only when $\sigma$ is small, it requires many time steps in the diffusion process for this approximation to work well, same as observed in Ho et al. (2020) and Song & Ermon (2020). In contrast, our diffusion recovery likelihood framework can be more flexible in choosing the number of time steps and the magnitude of $\sigma$.

3.5 Diffusion recovery likelihood

Sampling from $p_\theta(x|\hat{x})$ with MCMC becomes simpler when $\sigma$ is smaller. In the limit of $\sigma \rightarrow \infty$, $p_\theta(x|\hat{x})$ becomes the marginal distribution $p_\theta(x)$, which as we discussed before is highly multimodal and thus hard to sample from. To keep $\sigma$ small but still enable efficient sample generation from white noise, we propose to maximize a sequence of recovery likelihoods by chaining together a sequence of perturbation distributions with increasing intensity. This idea is in spirit similar to methods in Sohl-Dickstein et al. (2015); Ho et al. (2020) and Song & Ermon (2019; 2020). Specifically, assume a sequence of perturbed observations $x_0, x_1, \ldots, x_T$ such that

$$x_0 \sim p_{\text{data}}(X); \quad x_{t+1} = \sqrt{1 - \sigma_{t+1}^2} x_t + \sigma_{t+1} \epsilon_{t+1}, \quad t = 0, 1, \ldots, T - 1.$$

(15)

The scaling factor $\sqrt{1 - \sigma_{t+1}^2}$ ensures that the sequence is a spherical interpolation between the observed sample and Gaussian white noise. Let $y_t = \sqrt{1 - \sigma_{t+1}^2} x_t$, and we assume a sequence of conditional EBM

$$p_\theta(y_t|x_{t+1}) = \frac{1}{Z_{\theta,t}(x_{t+1})} \exp \left( f_\theta(y_t, t) - \frac{1}{2\sigma_{t+1}^2} \| x_{t+1} - y_t \|^2 \right), \quad t = 0, 1, \ldots, T - 1.$$

(16)

where $f_\theta(y_t, t)$ is defined by a neural network conditioned on $t$. 

Preprint. Work in progress.
We follow the learning algorithm in section 3.2. Inspired by the sampling procedure of the normal approximation (equation (12)), we set the step size \( \delta_t = b \sigma_t \) for Langevin dynamics, where \( b < 1 \) is a tunable hyperparameter. This schedule turns out to work well in practice, and thereby our Langevin sampling chain iterates according to

\[
y^+_{t} = y_{t} + \frac{b^2 \sigma^2 t}{2} (\nabla_y f_\theta(y^+_{t}, t) + \frac{1}{\sigma^2 t}(x_{t+1} - y^+_{t})) + b \sigma_t \epsilon_t.
\] (17)

Algorithm 1 summarizes the training procedure. For sample generation, we start from Gaussian noise, and initialize the MCMC for the model of the previous time step with the synthesized sample obtained at the current time step. We provide a detailed sampling algorithm in algorithm 2, where \( K \) denotes the number of Langevin sampling steps per noise scale. To show the efficacy of our method, we give several 2D toy examples learned by our diffusion recovery likelihood in Figures 2 and 3.

Algorithm 1

```
repeat
    Sample \( t \sim \text{Unif}(\{0, \ldots, T - 1\}) \).
    Sample pairs \((y_t, x_{t+1})\).
    Set synthesized sample \( y_t^- = x_{t+1} \).
    for \( \tau \leftarrow 1 \) to \( K \) do
        Update \( y_t^- \) according to equation (17)
    end for
    Update \( \theta \) following the gradients
    \[ \frac{\partial}{\partial \theta} f_\theta(y_t, t) - \frac{\partial}{\partial \theta} f_\theta(y_t^-, t). \]
until converged.
```

Algorithm 2

```
Sample \( x_T \sim \mathcal{N}(0, I) \).
for \( t \leftarrow T - 1 \) to \( 0 \) do
    \( y_t = x_{t+1} \).
    for \( \tau \leftarrow 1 \) to \( K \) do
        Update \( y_t \) according to equation (17)
    end for
    \( x_t = y_t / \sqrt{1 - \sigma^2_{t+1}} \).
end for
return \( x_0 \).
```

4 EXPERIMENTS

To show that diffusion recovery likelihood works well for various numbers of noise scales, we test the method under two settings: (1) \( T = 6 \), with \( K = 30 \) steps of Langevin sampling per noise scale and \( b = 0.0002 \); (2) \( T = 1000 \), with sampling from the normal approximation. Here (1) is close to the noise schedule of Song & Ermon (2019); while (2) resembles the noise schedule of Ho et al. (2020) where the magnitude of noise added at each time step is much smaller compared to (1). In both settings, we let \( \sigma^2_t \) increase linearly over \( t \). The network structure of \( f_\theta(x, t) \) is based on Wide ResNet (Zagoruyko & Komodakis, 2016) without weight normalization. As in Ho et al. (2020), we encode \( t \) with sinusoidal positional embeddings. Architectures and training details are in Appendix B. From now on we refer to the two settings as \( T6 \) and \( T1k \) respectively.

4.1 IMAGE GENERATION

In Figures 1 and 2, we show uncurated samples from our models on CIFAR-10 32 \( \times \) 32, CelebA 64 \( \times \) 64, and LSUN 64 \( \times \) 64/128 \( \times \) 128 datasets under the setting \( T6 \) (see Appendix C.4 for more samples). Our generated images have comparable visual quality to GAN-based methods. Quantitatively, we provide Fréchet Inception Distance (FID) (Heusel et al., 2017) and inception scores (Salimans et al., 2016) for both CIFAR-10 and CelebA 64 \( \times \) 64 datasets in Tables 1 and 3. On CIFAR-10, our model achieves an FID of 9.60 and inception score of 8.58, outperforming existing methods of learning explicit energy-based models by a large margin, as well as a majority of GAN-based methods. On CelebA 64 \( \times \) 64, our model obtains results comparable with the state-of-the-art GAN-based methods, and outperforms existing score-based methods (Song & Ermon, 2019, 2020). Note that both score-based methods (Song & Ermon, 2019, 2020) and diffusion probabilistic models (Ho et al., 2020) parametrize and learn score functions whereas we directly learn explicit energy-based models.
Table 1: FID and inception scores on CIFAR-10.

| Model                  | FID ↓ | Inception ↑ |
|------------------------|--------|-------------|
| **GAN-based**          |        |             |
| WGAN-GP (Gulrajani et al. 2017) | 36.4   | 7.86 ± .07  |
| SNGAN (Miyato et al. 2018) | 21.7   | 8.22 ± .05  |
| SNGAN-DDLS (Che et al. 2020) | 15.42  | 9.09 ± .10  |
| StyleGAN2-ADA (Karras et al. 2020) | 3.26   | 9.74 ± .05  |
| **Score-based**        |        |             |
| NCSN (Song & Ermon 2019) | 25.32  | 8.87 ± .12  |
| NCSN-v2 (Song & Ermon 2020) | 10.87  | 8.40 ± .07  |
| DDPM (Hö et al. 2020)   | **3.17** | 9.46 ± .11  |
| **Explicit EBM-conditional** |        |             |
| CoopNets (Xie et al. 2019) | -      | 7.30        |
| EBM-IG (Du & Mordatch 2019) | 37.9   | 8.30        |
| JEM (Grathwohl et al. 2019) | 38.4   | 8.76        |
| **Explicit EBM**       |        |             |
| Multi-grid (Gao et al. 2018) | 40.01  | 6.56        |
| CoopNets (Xie et al. 2018a) | 33.61  | 6.55        |
| EBM-SR (Nijkamp et al. 2019b) | -      | 6.21        |
| EBM-IG (Du & Mordatch 2019) | 38.2   | 6.78        |
| **Ours (T6)**          | **9.60** | **8.58 ± .12** |

Table 2: Ablation of training objectives, time steps $T$ and sampling steps $K$ on CIFAR-10. $K = 0$ indicates that we sample from the normal approximation.

| Setting / Objective | FID ↓ | Inception ↑ |
|---------------------|--------|-------------|
| $T = 1, K = 180$    | 32.12  | 6.89 ± .08  |
| $T = 1000, K = 0$   | 22.58  | 7.86 ± .11  |
| $T = 1000, K = 0$ (DSM) | 21.76  | 7.80 ± .07  |
| $T = 6, K = 30$     | 9.60   | 8.58 ± .12  |
| $T = 6, K = 50$     | **9.36** | **8.68 ± .11** |

Table 3: FID scores on CelebA 64^2.

| Model                  | FID ↓ |
|------------------------|-------|
| QA-GAN (Parimala & Channappayya 2019) | 6.42  |
| COCO-GAN (Lin et al. 2019) | **4.0** |
| NVAE (Vahdat & Kautz 2020) | 14.74 |
| NCSN (Song & Ermon 2019) | 25.30 |
| NCSN-v2 (Song & Ermon 2020) | 10.23 |
| EBM-SR (Nijkamp et al. 2019b) | 23.02 |
| EBM-Triangle (Han et al. 2020) | 24.70 |
| **Ours (T6)**          | **5.98** |

Figure 4: Generated samples on unconditional CIFAR-10 (left) and LSUN 64^2 church_outdoor (center) and LSUN 64^2 bedroom (right).

Figure 5: Interpolation results between the leftmost and rightmost generated samples. For top to bottom: LSUN church_outdoor 128^2, LSUN bedroom 128^2 and CelebA 64^2.
Table 4: Test bits/dim on CIFAR-10. † indicates that we estimate the bits/dim with the approximated log partition function instead of analytically computing it, and thus it is not directly comparable (see section 4.3).

| Model                  | BPD↓ |
|------------------------|------|
| DDPM (Ho et al., 2020) | 3.70 |
| Glow (Kingma & Dhariwal, 2018) | 3.35 |
| Flow++ (Ho et al., 2019) | 3.08 |
| PixelCNN (Van den Oord et al., 2016) | 3.03 |
| Sparse Transformer (Child et al., 2019) | 2.80 |
| DistAug (Jun et al., 2020) | 2.56 |
| **Ours† (T1k)**        | 3.18 |

Figure 6: Image inpainting on LSUN church_outdoor 128² (left) and CelebA 64² (right). With each block, the top row are mask images while the bottom row are inpainted images.

Interpolation. As shown in Figure 5, our model is capable of smooth interpolation between two generated samples, using a similar method to Song & Ermon (2020). Specifically, for two samples \( x_0 \) and \( x_1 \), we do a sphere interpolation between the initial white noise images \( x_0^{(0)} \) and \( x_1^{(1)} \) and the noise terms of Langevin dynamics \( \epsilon_{t,0} \) and \( \epsilon_{t,1} \) in all sampling steps. More interpolation results can be found in Appendix C.2.

Image inpainting. One promising application of energy-based models is using the learned model as a prior for image processing, such as image inpainting, denoising and super-resolution. Such applications have been explored in Gao et al. (2018); Du & Mordatch (2019); Song & Ermon (2019), to name a few. In Figure 6, we demonstrate that models learned by maximizing recovery likelihoods are capable of realistic and semantically meaningful image inpainting with a method similar to Song & Ermon (2019). Specifically, given a masked image and the corresponding mask, we first obtain a sequence of perturbed masked images at different noise levels. The inpainting can be easily achieved by running Langevin dynamics progressively on the masked pixels while keeping the observed pixels fixed at decreasingly lower noise levels. Additional image inpainting results can be found in Appendix C.3.

4.2 ABLATION STUDY

We investigate the effect of choosing different number of time steps \( T \) and number of sampling steps \( K \) on CIFAR-10, and report the results of ablation study in Table 2. First, to show that it is beneficial to learn by diffusion recovery likelihood, we compare against a baseline approach \((T = 1, K = 180)\) where we use only one time step so that the recovery likelihood becomes equivalent to marginal likelihood. This baseline amounts to the method adopted by Nijkamp et al. (2019b) and Du & Mordatch (2019). For fair comparison, we use the same number of MCMC steps for both our \( T6 \) setting and the baseline method (i.e., 180 sampling steps). As shown by Table 2, our method outperforms this baseline by a large margin. Moreover, our models can be trained more efficiently as the number of sampling steps per iteration is reduced and amortized across time steps.

In addition, we report the sample quality of setting \( T1k \). We test two training objectives for this setting: (1) maximizing recovery likelihoods \((T = 1000, K = 0)\) and (2) maximizing likelihoods of the approximated normal distributions \((T = 1000, K = 0)\) (DSM). As mentioned in section 3.4, (2) is equivalent to the training objective of denoising score matching with Langevin dynamics (Song & Ermon, 2019, 2020) and denoising diffusion probabilistic model (Ho et al., 2020), except that the score functions are taken as the gradients of explicit energy functions. In practice, for a direct comparison, (2) follows the same implementation as in Ho et al. (2020), except that we parameterize the score function as the gradient of an energy-based model. We observe from Table 2 that (1) and (2)
achieve similar sample quality in terms of quantitative metrics, where (2) results in a slightly better FID score yet a slightly worse inception score. This corroborates that training objectives of (1) and (2) are consistent. Both (1) and (2) perform worse than setting $T6$. A possible explanation is that the sampling error may accumulate over time steps, so that a more flexible schedule of time steps accompanied with certain amount of sampling steps is preferred.

Finally, we examine the influence of varying the number of sampling steps while fixing the number of time steps. Figure 7 demonstrates FID scores computed on 2,500 samples every 15,000 iterations. We observe that training becomes unstable when the number of sampling steps are too small ($T = 6, K = 10$), and that more sampling steps lead to better sample quality. However, since $K = 50$ does not lead to significant improvement over $K = 30$ but has much higher computational cost, we choose $K = 30$ for image generation on all datasets.

4.3 LONG-RUN CHAIN ANALYSIS

Aside from achieving high quality generation, our models also capture a faithful energy potential. One common approach to check the learned potential is to perform long-run MCMC sampling and examine whether samples still remain realistic. As pointed out in Nijkamp et al. (2019a), almost all existing methods for EBM training fail in getting realistic samples from long-run MCMC. In contrast, we demonstrate below that by composing a thousand diffusion time steps ($T1k$ setting), we can use MCMC to form steady long-run chains for conditional distributions.

![Figure 8: Left: Adjusted step size of HMC over time step. Center: Acceptance rate over time step. Right: Estimated log partition function over number of samples with different number of sampling steps per time step. The x-axis is plotted in log scale.](image)

After training the model by maximizing the diffusion recovery likelihood under setting $T1k$, we first sample from the normal approximation as the first sampling step, and then use Hamiltonian Monte Carlo (HMC) (Neal et al., 2011) with 2 leapfrog steps to perform the remaining sampling steps. We adaptively adjust the step size of HMC to make the average acceptance rate in range $[0.6, 0.9]$, which is computed over 1000 chains for 100 steps. Figure 8 displays the adjusted step size (left) and acceptance rate (center) over time step. The adjusted step size increases in log scale. With this step size schedule, we generate long-run chains from the learned sequence of conditional distributions. As shown in Figure 9, images remain realistic for even $100k$ sampling steps in total (i.e., 100 sampling steps per time step), resulting in an FID of 24.89. This score is close to the one computed on samples generated by $1k$ steps (i.e., sampled from normal approximation), which is 25.12. As a further check, we recruit a No-U-Turn Sampler (Hoffman & Gelman, 2014) with the same step size schedule as HMC to perform long-run sampling, where the samples also remain realistic (see Appendix C.1 for more details).

Given the long-run MCMC samples from the conditional distributions, we can estimate the log ratio of the partition functions of the marginal distributions, and further estimate the partition function of $p_\theta(y_0)$. The strategy is based on annealed importance sampling (Neal, 2001). See Appendix A.6 for the implementation details. The right subfigure of Figure 8 depicts the estimated log partition function of $p_\theta(y_0)$ over the number of MCMC samples used. To verify the estimation strategy and again check the long-run chain samples, we conduct multiple runs using samples generated...
with different numbers of HMC steps and display the estimation curves. All the curves saturate to values close to each other at the end, indicating the stability of long-run chain samples and the effectiveness of the estimation strategy. With the estimated partition function, by change of variable, we can estimate the normalized density of data as \( \sqrt{1 - \sigma_1^2} p_0(\sqrt{1 - \sigma_1^2} x_0) \). We report test bits per dimension on CIFAR-10 in Table 4. Note that the result should be taken with a grain of salt, because the partition function is estimated by samples and as shown in Appendix A.6, it is a stochastic lower bound of the true value, which will match the true value only in the limit of infinite samples.

5 CONCLUSION

We propose to learn EBMs by diffusion recovery likelihood, a variant of MLE applied to multiple scales of noise perturbations. We achieve high quality image synthesis, and with a thousand noise levels, we obtain faithful long-run MCMC samples that indicate the validity of the learned energy potentials. One direction for future work is to combine the high quality sample generation of model-T6 and the faithful long-run MCMC sampling of model-T1k for the best of both worlds. Since this method can learn EBMs efficiently with a smaller budget of MCMC, we are also interested in scaling it up to higher resolution images and investigating this method in other data modalities.

ACKNOWLEDGEMENT

The work was done while Ruiqi Gao and Yang Song were interns at Google Brain. The work of Ying Nian Wu is supported by NSF DMS-2015577. We thank Alexander A. Alemi, Jonathan Ho, Tim Salimans and Kevin Murphy for their insightful discussions during the course of this project.

REFERENCES

Yoshua Bengio, Li Yao, Guillaume Alain, and Pascal Vincent. Generalized denoising auto-encoders as generative models. In Advances in neural information processing systems, pp. 899–907, 2013.

Yoshua Bengio, Eric Laufer, Guillaume Alain, and Jason Yosinski. Deep generative stochastic networks trainable by backprop. In International Conference on Machine Learning, pp. 226–234, 2014.

Tong Che, Ruixiang Zhang, Jascha Sohl-Dickstein, Hugo Larochelle, Liam Paull, Yuan Cao, and Yoshua Bengio. Your gan is secretly an energy-based model and you should use discriminator driven latent sampling. arXiv preprint arXiv:2003.06060, 2020.

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

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. arXiv preprint arXiv:1904.10509, 2019.

Guillaume Desjardins, Yoshua Bengio, and Aaron C Courville. On tracking the partition function. In Advances in neural information processing systems, pp. 2501–2509, 2011.

Yilun Du and Igor Mordatch. Implicit generation and generalization in energy-based models. arXiv preprint arXiv:1903.08689, 2019.

Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A connection between generative adversarial networks, inverse reinforcement learning, and energy-based models. arXiv preprint arXiv:1611.03852, 2016.

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, pp. 9155–9164, 2018.

Ruiqi Gao, Erik Nijkamp, Diederik P Kingma, Zhen Xu, Andrew M Dai, and Ying Nian Wu. Flow contrastive estimation of energy-based models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7518–7528, 2020.
Mark Girolami and Ben Calderhead. Riemann manifold langevin and hamiltonian monte carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(2):123–214, 2011.

Anirudh Goyal Alias Parth Goyal, Nan Rosemary Ke, Surya Ganguli, and Yoshua Bengio. Variational walkback: Learning a transition operator as a stochastic recurrent net. In *Advances in Neural Information Processing Systems*, pp. 4392–4402, 2017.

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. *arXiv preprint arXiv:1912.03263*, 2019.

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

Roger B Grosse, Siddharth Ancha, and Daniel M Roy. Measuring the reliability of mcmc inference with bidirectional monte carlo. In *Advances in Neural Information Processing Systems*, pp. 2451–2459, 2016.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pp. 5767–5777, 2017.

Tian Han, Erik Nijkamp, Linqi Zhou, Bo Pang, Song-Chun Zhu, and Ying Nian Wu. Joint training of variational auto-encoder and latent energy-based model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7978–7987, 2020.

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*, pp. 6626–6637, 2017.

Jonathan Ho, Xi Chen, Aravind Srinivas, Yan Duan, and Pieter Abbeel. Flow++: Improving flow-based generative models with variational dequantization and architecture design. *arXiv preprint arXiv:1902.00275*, 2019.

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

Matthew D Hoffman and Andrew Gelman. The no-u-turn sampler: adaptively setting path lengths in hamiltonian monte carlo. *J. Mach. Learn. Res.*, 15(1):1593–1623, 2014.

Long Jin, Justin Lazarow, and Zhuowen Tu. Introspective classification with convolutional nets. In *Advances in Neural Information Processing Systems*, pp. 823–833, 2017.

Heewoo Jun, Rewon Child, Mark Chen, John Schulman, Aditya Ramesh, Alec Radford, and Ilya Sutskever. Distribution augmentation for generative modeling. In *Proceedings of Machine Learning and Systems 2020*, pp. 10563–10576. 2020.

Tero Karras, Miika Aittala, Janne Hellsten, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Training generative adversarial networks with limited data. *arXiv preprint arXiv:2006.06676*, 2020.

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

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Diederik P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. In *Advances in Neural Information Processing Systems*, pp. 10215–10224, 2018.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.
Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Rithesh Kumar, Anirudh Goyal, Aaron Courville, and Yoshua Bengio. Maximum entropy generators for energy-based models. \textit{arXiv preprint arXiv:1901.08508}, 2019.

Justin Lazarow, Long Jin, and Zhuowen Tu. Introspective neural networks for generative modeling. In \textit{Proceedings of the IEEE International Conference on Computer Vision}, pp. 2774–2783, 2017.

Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. \textit{Predicting structured data}, 1(0), 2006.

Kwonjoon Lee, Weijian Xu, Fan Fan, and Zhuowen Tu. Wasserstein introspective neural networks. In \textit{Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition}, pp. 3702–3711, 2018.

Chieh Hubert Lin, Chia-Che Chang, Yu-Sheng Chen, Da-Cheng Juan, Wei Wei, and Hwann-Tzong Chen. Coco-gan: generation by parts via conditional coordinating. In \textit{Proceedings of the IEEE International Conference on Computer Vision}, pp. 4512–4521, 2019.

Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Large-scale celebfaces attributes (celeba) dataset. \textit{Retrieved August}, 15:2018, 2018.

Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. \textit{arXiv preprint arXiv:1802.05957}, 2018.

Radford M Neal. Annealed importance sampling. \textit{Statistics and computing}, 11(2):125–139, 2001.

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

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

Erik Nijkamp, Mitch Hill, Tian Han, Song-Chun Zhu, and Ying Nian Wu. On the anatomy of mcmc-based maximum likelihood learning of energy-based models. \textit{arXiv preprint arXiv:1903.12370}, 2019a.

Erik Nijkamp, Mitch Hill, Song-Chun Zhu, and Ying Nian Wu. On learning non-convergent short-run mcmc toward energy-based model. \textit{arXiv preprint arXiv:1904.09770}, 2019b.

KANCHARLA Parimala and Sumohana and Channappayya. Quality aware generative adversarial networks. In \textit{Advances in Neural Information Processing Systems}, pp. 2948–2958, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. \textit{arXiv preprint arXiv:2006.09011}, 2020.
Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in Neural Information Processing Systems*, 33, 2020.

Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. In *Advances in neural information processing systems*, pp. 4790–4798, 2016.

Jianwen Xie, Yang Lu, Ruiqi Gao, Song-Chun Zhu, and Ying Nian Wu. Cooperative training of descriptor and generator networks. *arXiv preprint arXiv:1609.09408*, 2016a.

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

Jianwen Xie, Zilong Zheng, Xiaolin Fang, Song-Chun Zhu, and Ying Nian Wu. Cooperative training of fast thinking initializer and slow thinking solver for multi-modal conditional learning. *arXiv preprint arXiv:1902.02812*, 2019.

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

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.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.
A  EXTENDED DERIVATIONS

A.1  DERIVATION OF EQUATION 5

Let \( \tilde{x} = x + \sigma \epsilon \), where \( \epsilon \sim \mathcal{N}(0, I) \). Given the marginal distribution of

\[
p_\theta(x) = \frac{1}{Z_\theta} \exp(f_\theta(x)),
\]

(18)

We can derive the conditional distribution of \( x \) given \( \tilde{x} \) as

\[
p_\theta(x|\tilde{x}) = p_\theta(x)p(\tilde{x}|x)/p(\tilde{x})
\]

\[= \frac{1}{Z_\theta} \exp(f_\theta(x)) \frac{1}{(2\pi\sigma^2)^{\frac{n}{2}}} \exp(-\frac{1}{2\sigma^2}||\tilde{x} - x||^2)/p(\tilde{x})
\]

(19)

\[
= \frac{1}{Z_\theta(\tilde{x})} \exp \left( f_\theta(x) - \frac{1}{2\sigma^2}||\tilde{x} - x||^2 \right),
\]

(20)

where we absorb all the terms that are irrelevant of \( x \) as \( \tilde{Z}_\theta(\tilde{x}) \).

A.2  THEORETICAL UNDERSTANDING

In this subsection, we analyze the asymptotic behavior of maximizing the recovery log-likelihood.

For model class \( \{p_\theta(x), \forall \theta\} \), suppose there exists \( \theta^* \) such that \( p_{data} = p_{\theta^*} \). According to the classical theory of MLE, let \( \hat{\theta}_0 \) be the point estimate by MLE. Then we have \( \hat{\theta} \) is an unbiased estimator of \( \theta^* \) with asymptotic normality:

\[
\sqrt{n}(\hat{\theta}_0 - \theta^*) \to \mathcal{N}(0, \mathcal{I}_0(\theta^*)^{-1}),
\]

(22)

where \( \mathcal{I}_0(\theta) = \mathbb{E}_{x \sim p_\theta}[ -\nabla_\theta^2 \log p_\theta(x)] \) is the Fisher information, and \( n \) is the number of observed samples.

Let \( \hat{\theta} \) be the point estimate given by maximizing recovery log-likelihood, we can derive a result in parallel to that of MLE:

\[
\sqrt{n}(\hat{\theta} - \theta^*) \to \mathcal{N}(0, \mathcal{I}(\theta^*)^{-1}),
\]

(23)

where \( \mathcal{I}(\theta) = \mathbb{E}_{p_\theta(x, \tilde{x})}[ -\nabla_\theta^2 \log p_\theta(x|\tilde{x})] \). The relationship between \( I_0(\theta) \) and \( I(\theta) \) is that

\[
I_0(\theta) = I(\theta) + \mathbb{E}_{p_\theta(x, \tilde{x})}[ -\nabla_\theta^2 \log p_\theta(\tilde{x})].
\]

(24)

Thus there is loss of information, but \( \hat{\theta} \) is still an unbiased estimator of \( \theta^* \) with asymptotic normality.

A.3  DETAILED DERIVATION OF NORMAL APPROXIMATION

\[
-\mathcal{E}_0(x|\tilde{x}) = f_\theta(x) - \frac{1}{2\sigma^2}||\tilde{x} - x||^2
\]

(25)

\[\equiv f_\theta(\tilde{x}) + \langle \nabla_x f_\theta(\tilde{x}), x - \tilde{x} \rangle - \frac{1}{2\sigma^2}||\tilde{x} - x||^2 \]

(26)

\[= -\frac{1}{2\sigma^2} \left[ ||x||^2 - 2\langle \tilde{x}, x \rangle + ||\tilde{x}||^2 \right] + \langle \nabla_x f_\theta(\tilde{x}), x \rangle - \langle \nabla_x f_\theta(\tilde{x}), \tilde{x} \rangle + f_\theta(\tilde{x}) \]

(27)

\[= -\frac{1}{2\sigma^2} \left[ ||x||^2 - 2\langle \tilde{x} + \sigma^2 \nabla_x f_\theta(\tilde{x}), x \rangle \right] - \frac{1}{2\sigma^2}||\tilde{x}||^2 - \langle \nabla_x f_\theta(\tilde{x}), \tilde{x} \rangle + f_\theta(\tilde{x}) \]

(28)

\[= -\frac{1}{2\sigma^2} \left[ ||x||^2 - (\tilde{x} + \sigma^2 \nabla_x f_\theta(\tilde{x}))||^2 \right] + c, \]

(29)

A.4  DIFFERENCE BETWEEN THE SCORES OF \( p(x) \) AND \( p(\tilde{x}) \)

For notation clarity, with \( \tilde{x} = x + \epsilon \), we let \( \tilde{p} \) be the distribution of \( \tilde{x} \), and \( p \) be the distribution of \( x \). Then for a smooth testing function with vanishing tails,

\[
\mathbb{E}[h(\tilde{x})] = \mathbb{E}[h(x + \epsilon)]
\]

(30)

\[\equiv \mathbb{E}[h(x) + h'(x)\epsilon + h''(x)\epsilon^2/2]
\]

(31)

\[= \mathbb{E}[h(x)] + \mathbb{E}[h''(x)]\sigma^2/2. \]

(32)
Integral by part,
\[
\mathbb{E}[h''(x)] = \int h''(x)p(x)dx = -\int h'(x)p'(x)dx = \int p''(x)h(x)dx. \tag{33}
\]
Thus we have the heat equation
\[
\tilde{p}(x) = p(x) + p''(x)\sigma^2/2. \tag{34}
\]
The score
\[
\nabla_x \log \tilde{p}(x) = \nabla_x \log(p(x)) + \nabla_x \log(1 + p''(x)/p(x))\sigma^2/2
\]
\[
= \nabla_x \log(p(x)) + \nabla_x [p''(x)/p(x)]\sigma^2/2. \tag{36}
\]
Thus the difference between the score of \( p \) and \( \tilde{p} \) is of the order \( \sigma^2 \), which is negligible when \( \sigma^2 \) is small.

### A.5 Learning gradients of normal approximation and original recovery likelihood

In this subsection we demonstrate that the learning gradient of maximizing likelihood of the normal approximation is approximately the same as the gradient of maximizing the original recovery likelihood with one step of Langevin sampling. Specifically, the gradient of the normal approximation of recovery log-likelihood for an observed \( x_{\text{obs}} \) is
\[
\nabla_{\theta} \left( \frac{1}{2\sigma^2} \left[ \|x_{\text{obs}} - (\bar{x} + \sigma^2 f'_{\theta}(\bar{x}))\|^2 \right] \right) = \nabla_{\theta} f'_{\theta}(\bar{x})(x_{\text{obs}} - (\bar{x} + \sigma^2 f'_{\theta}(\bar{x})). \tag{37}
\]

On the other hand, to maximize the original recovery likelihood, suppose we sample \( x_{\text{syn}} \sim p_\theta(x|\bar{x}) \), then the gradient ascent of the original recovery log-likelihood is
\[
\nabla_{\theta} f_\theta(x_{\text{obs}}) - \mathbb{E}[\nabla_{\theta} f_\theta(x_{\text{syn}})] = h_\theta(x_{\text{obs}}) - \mathbb{E}[h_\theta(x_{\text{syn}})], \tag{38}
\]
where \( h_\theta(x) = \nabla_{\theta} f_\theta(x) \). Approximately, if we perform one step of Langevin dynamics from \( \bar{x} \) to obtain \( x_{\text{syn}} \), i.e., \( x_{\text{syn}} = \bar{x} + \sigma^2 f'_{\theta}(\bar{x}) + \sqrt{2}\sigma e \), and assume \( f_\theta(x) \) is locally linear in \( x \), then
\[
\nabla_{\theta} f_\theta(x_{\text{obs}}) - \mathbb{E}[\nabla_{\theta} f_\theta(x_{\text{init}})] = h_\theta(x_{\text{obs}}) - \mathbb{E}[h_\theta(x_{\text{obs}})]
\]
\[
= h_\theta(\bar{x}) + h'_{\theta}(\bar{x})(x_{\text{obs}} - \bar{x}) - \mathbb{E}[h_\theta(\bar{x}) + h'_{\theta}(\bar{x})(\sigma^2 f'_{\theta}(\bar{x}) + \sigma e)]
\]
\[
= h'_{\theta}(\bar{x})(x_{\text{obs}} - (\bar{x} + \sigma^2 f'_{\theta}(\bar{x}))
\]
\[
= \nabla_{\theta} f'_{\theta}(\bar{x})(x_{\text{obs}} - (\bar{x} + \sigma^2 f'_{\theta}(\bar{x})). \tag{43}
\]
Comparing equations (37) and (43) we see that the two gradients agree with each other.

### A.6 Estimating the partition function

We can utilize the sequence of learned distributions of \( y_t = (1 - \sigma_{t+1}^2)x_t \) to estimate the partition function. Specifically, the marginal distribution of \( y_t \) is
\[
p_\theta(y_t) = \frac{1}{Z_{\theta,t}} \exp (f_\theta(y_t, t)) \tag{44}
\]
We can estimate the ratio of the partition functions at two consecutive time steps using importance sampling
\[
\frac{Z_{\theta,t}}{Z_{\theta,t+1}} = \mathbb{E}_{p_\theta(y_{t+1})} [\exp(f_\theta(y, t) - f_\theta(y, t + 1))]
\]
\[
= \frac{1}{M} \sum_{i=1}^{M} [\exp(f_\theta(y_{t+1,i}, t) - f_\theta(y_{t+1,i}, t + 1))], \tag{46}
\]
where \( y_{t+1, i} \) are samples generated by progressive sampling. Starting from \( t = T \), where \( p_T(x) \) follows Gaussian distribution, we can compute \( \log Z_t \) along the reverse path of the diffusion process, until we reach \( t = 0 \):

\[
Z_{\theta, 0} = Z_{\theta, T} \prod_{t=0}^{T-1} \frac{Z_{\theta, t}}{Z_{\theta, t+1}}.
\]

(47)

In practice, since the ratio given by MCMC samples can vary across many orders of magnitude, it is more meaningful to estimate

\[
\log Z_{\theta, 0} = \log Z_{\theta, T} + \sum_{t=0}^{T-1} \log \frac{Z_{\theta, t}}{Z_{\theta, t+1}}.
\]

(48)

Unfortunately, although equation (46) is an unbiased estimator of \( Z_{\theta, T}/Z_{\theta, T+1} \), the logarithm of this estimator is generally a stochastic lower bound of \( \log(Z_{\theta, T}/Z_{\theta, T+1}) \) (Grosse et al. 2016). However, as we show below, this bound will gradually converge to an unbiased estimator of \( \log(Z_{\theta, T}/Z_{\theta, T+1}) \), as the number of samples becomes large. Specifically, let \( A \) be the estimator in equation (46), \( \mu \) be the true value of \( Z_{\theta, T}/Z_{\theta, T+1} \). We have \( E[A] = \mu \), then by second order Taylor expansion,

\[
E[\log A] = E \left[ \log \mu + \frac{1}{\mu} (A - \mu) - \frac{1}{2\mu^2} (A - \mu)^2 \right]
\]

(49)

\[
= \log \mu - \frac{1}{2\mu^2} Var(A).
\]

(50)

By law of large number, \( Var(A) \rightarrow 0 \) as \( M \rightarrow \infty \), and thus \( E[\log A] \rightarrow \log \mu \). This is also consistent with the estimation curves in the right subfigure of Figure 8, since \( Var(A) \geq 0 \), the estimation curve increases from below as the number of samples becomes larger. When the curve becomes stable, it indicates the convergence.

**B Experimental details**

**Model architecture.** Our network structure is based on Wide ResNet (Zagoruyko & Komodakis, 2016). Table 5 lists the detailed network structures of various resolutions. The number of ResBlocks at every level \( N \) is a hyperparameter that we sweep over. The values of \( N \) for various datasets are listed in Table 6. Each ResBlock consists of two Conv2D layers. For the second Conv2D layer, we use zero initialization for the weights, and add a trainable channel-wise scaling parameter to the output. We remove the weight normalization, and use leaky ReLU (slope = 0.2) as the activation function in ResBlocks. Spectral normalization (Miyato et al., 2018) is used to regularize parameters in Conv2D layer, ResBlocks and Dense layer. For encoding time step \( t \), we follow the scheme in (Ho et al., 2020). Specifically, the time step \( t \) is first transformed into sinusoidal embedding, and then two Dense layers are added. The time embedding is added after the first Conv2D layer of each ResBlock.

**Training.** We use Adam (Kingma & Ba, 2014) optimizer for all the experiments. We find that for high resolution images, using a smaller \( \beta_1 \) in Adam help stabilize training. We use learning rate 0.0001 for all the experiments. For the values of \( \beta_1 \), batch sizes and the number of training iterations for various datasets, see Table 6.

**Datasets.** We use the following datasets in our experiments: CIFAR-10 (Krizhevsky et al., 2009), CelebA (Liu et al., 2018) and LSUN (Yu et al., 2015). CIFAR-10 is of resolution \( 32 \times 32 \), and contains 50,000 training images and 10,000 test images. CelebA contains 202,599 face images, of which 162,770 are training images and 19,962 are test images. For processing, we first clip each image to \( 178 \times 178 \) and then resize it to \( 64 \times 64 \). For LSUN, we use church_outdoor and bedroom categories, which contains 126,227 and 3,033,042 training images respectively. Both categories contain 300 test images. For processing, we first crop each image to a square image of the smaller size among the height and weight, and then we resize it to \( 64 \times 64 \) or \( 128 \times 128 \). For resizing, we set antialias to True. We apply horizontal random flip as data augmentation for all datasets during training.
Evaluation metrics. We use FID and inception scores as quantitative evaluation metrics of sample quality. On all the datasets, we calculate FID and inception scores on 50,000 samples using the original code from [Salimans et al. (2016)] and [Heusel et al. (2017)].

Table 5: Model architectures of various solutions. \(N\) is a hyperparameter that we sweep over.

| Resolution | (a) \(32 \times 32\) | (b) \(64 \times 64\) | (c) \(128 \times 128\) |
|------------|-------------------|-------------------|-------------------|
| \(3 \times 3\) Conv2D, 128 | \(N\) ResBlocks, 128 | \(N\) ResBlocks, 128 | \(N\) ResBlocks, 128 |
| Downsample \(2 \times 2\) | Downsample \(2 \times 2\) | Downsample \(2 \times 2\) | Downsample \(2 \times 2\) |
| \(N\) ResBlocks, 256 | \(N\) ResBlocks, 256 | \(N\) ResBlocks, 256 | \(N\) ResBlocks, 256 |
| Downsample \(2 \times 2\) | Downsample \(2 \times 2\) | Downsample \(2 \times 2\) | Downsample \(2 \times 2\) |
| \(N\) ResBlocks, 256 | \(N\) ResBlocks, 256 | \(N\) ResBlocks, 256 | \(N\) ResBlocks, 256 |
| Downsample \(2 \times 2\) | Downsample \(2 \times 2\) | Downsample \(2 \times 2\) | Downsample \(2 \times 2\) |
| \(N\) ResBlocks, 256 | \(N\) ResBlocks, 256 | \(N\) ResBlocks, 512 | \(N\) ResBlocks, 512 |
| ReLU, global sum | ReLU, global sum | ReLU, global sum | ReLU, global sum |
| Dense 1 | Dense 1 | Dense 1 | Dense 1 |

(d) Time embedding (temb) | (e) ResBlock
---|---
sinusoidal embedding | leakyReLU, \(3 \times 3\) Conv2D
Dense, leakyReLU | + Dense(leakyReLU(temb))
 Dense | leakyReLU, \(3 \times 3\) Conv2D
 + input

Table 6: Hyperparameters of various datasets.

| Dataset                  | \(N\) | \(\beta_1\) in Adam | Batch size | Training iterations |
|--------------------------|-------|---------------------|------------|--------------------|
| CIFAR-10                 | 5     | 0.9                 | 256        | 50k                |
| CelebA                   | 2     | 0.9                 | 128        | 100k               |
| LSUN church_outdoor 64\(^2\) | 2     | 0.9                 | 128        | 100k               |
| LSUN bedroom 64\(^2\)    | 2     | 0.9                 | 128        | 100k               |
| LSUN church_outdoor 128\(^2\) | 2     | 0.5                 | 64         | 100k               |
| LSUN bedroom 128\(^2\)   | 5     | 0.5                 | 64         | 56k                |

C ADDITIONAL EXPERIMENTAL RESULTS

C.1 LONG-RUN CHAIN SAMPLING WITH NUTS

As a further check, we use a No-U-Turn Sampler [Hoffman & Gelman, 2014] to perform the long-run chain sampling, with the same step size schedule obtained for HMC sampler. Figure 10 displays samples with different number of sampling steps. The samples remain realistic after \(100k\) sampling steps in total and the FID score remains stable.

C.2 ADDITIONAL INTERPOLATION RESULTS

Figures [11], [12] and [13] display more examples of interpolation between two generated samples on CelebA \(64\(^2\), LSUN church_outdoor \(128\(^2\) and LSUN bedroom \(128\(^2\).
Figure 10: Long run chain samples with different total number of NUTS steps.

Figure 11: Interpolation results between the leftmost and rightmost generated samples on CelebA $64 \times 64$.

C.3 Additional Image Inpainting Results

Figures 14 and 15 show additional examples of image inpainting on CelebA $64^2$ and LSUN church_outdoor $128^2$.

C.4 Additional Uncurated Samples

Figures 16, 17, 18, 19, 20 and 21 show uncurated samples from the learned models under T6 setting on CIFAR-10, CelebA $64^2$, LSUN church_outdoor $128^2$, LSUN bedroom $128^2$, LSUN church_outdoor $64^2$ and LSUN bedroom $64^2$ datasets.
Figure 12: Interpolation results between the leftmost and rightmost generated samples on LSUN church_outdoor $128 \times 128$.  

Figure 13: Interpolation results between the leftmost and rightmost generated samples on LSUN bedroom $128 \times 128$. 
Figure 14: Image inpainting results on CelebA $64 \times 64$. Top: masked images, bottom: inpainted images.

Figure 15: Image inpainting results on LSUN church_outdoor $128 \times 128$. Top: masked images, bottom: inpainted images.
Figure 16: Generated samples on CIFAR-10.
Figure 17: Generated samples on CelebA $64 \times 64$. 
Figure 18: Generated samples on LSUN church_outdoor $128 \times 128$. FID=9.76
Figure 19: Generated samples on LSUN bedroom 128 × 128. FID=11.27
Figure 20: Generated samples on LSUN church_outdoor $64 \times 64$. FID=7.02
Figure 21: Generated samples on LSUN bedroom $64 \times 64$. FID=8.98