LEARNING ENERGY-BASED MODELS BY DIFFUSION RECOVERY LIKELIHOOD

Anonymous authors
Paper under double-blind review

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 with recovery likelihood, which maximizes the conditional probability of the data at a certain noise level given their noisy versions at a higher noise level. Optimizing recovery likelihood is more tractable than marginal likelihood, as sampling from the conditional distributions is much easier than sampling from the marginal distributions. After training, synthesized images can be generated by the sampling process that initializes from Gaussian white noise 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

EBMs (LeCun et al., 2006; Ngiam et al., 2011; Kim & Bengio, 2016; Zhao et al., 2016; Goyal et al., 2017; Xie et al., 2016b; Finn et al., 2016; Gao et al., 2018; Kumar et al., 2019; Nijkamp et al., 2019b; Du & Mordatch, 2019; Grathwohl et al., 2019; Desjardins et al., 2011; Gao et al., 2020; Che et al., 2020; Grathwohl et al., 2020; Qui et al., 2019; Rhodes et al., 2020) are an appealing class of probabilistic models, which can be viewed as generative versions of discriminators (Jin et al., 2017; Lazarow et al., 2017; Lee et al., 2018; Grathwohl et al., 2020), yet can be learned from unlabeled data. Despite a number of desirable properties, two challenges remain for training EBMs on high-dimensional datasets. First, learning EBMs by maximum likelihood requires Markov Chain Monte Carlo (MCMC) to generate samples from the model, which can be extremely expensive. Second, as pointed out in Nijkamp et al. (2019a), 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.

Another line of work, originating from Sohl-Dickstein et al. (2015), is to learn from a diffused version of the data, which are obtained from the original data via a diffusion process that sequentially adds Gaussian white noise. From such diffusion data, one can learn the conditional model of the data at a certain noise level given their noisy versions at the higher noise level. After learning the sequence of conditional models that invert the diffusion process, one can generate synthesized images from Gaussian white noise images by ancestral sampling. Building on Sohl-Dickstein et al. (2015), Ho et al. (2020) further developed the method, obtaining strong image synthesis results.

Inspired by Sohl-Dickstein et al. (2015) and Ho et al. (2020), we propose a diffusion recovery likelihood method to tackle the challenge of training EBMs directly on a dataset by instead learning a sequence of EBMs for the marginal distributions of the diffusion process. The sequence of marginal EBMs are learned with recovery likelihoods that are defined as the conditional distributions that invert the diffusion process. Compared to standard maximum likelihood estimation (MLE) of EBMs,
learning marginal EBMs by diffusion recovery likelihood only requires sampling from the conditional distributions, which is much easier than sampling from the marginal distributions. After learning the marginal EBMs, we can generate synthesized images by a sequence of conditional samples initialized from the Gaussian white noise distribution. Unlike [Ho et al. (2020)] that approximates the reverse process by normal distributions, in our case the conditional distributions are derived from the marginal EBMs, which are more flexible. The framework of recovery likelihood was originally proposed in [Bengio et al. (2013)]. In our work, we adapt it to learning the sequence of marginal EBMs from the diffusion data. [Rhodes et al. (2020)] proposes a related framework where they learn the sequence of EBMs based on noise contrastive estimation.

Our work is also related to the denoising score matching method of [Vincent (2011)], which was further developed by [Song & Ermon (2019, 2020)] for learning from diffusion data. The training objective used for diffusion probabilistic models is a weighted version of the denoising score matching objective, as revealed by [Ho et al. (2020)]. These methods learn the score functions (the gradients of the energy functions) directly, instead of using the gradients of learned energy functions as in EBMs. On the other hand, [Saremi et al. (2018)] parametrizes the score function as the gradient of a MLP energy function, and [Saremi & Hyvarinen (2019)] further unifies denoising score matching and neural empirical Bayes.

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 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 approximate $p_{\text{data}}(x)$. 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 analytically intractable 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 \( i = 1, 2, ..., n \). The log-likelihood function is

\[
L(\theta) = \frac{1}{n} \sum_{i=1}^{n} \log p_\theta(x_i) = E_{x \sim p_{\text{data}}} \left[ \log p_\theta(x) \right].
\]

(2)

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

\[
-\frac{\partial}{\partial \theta} D_{\text{KL}}(p_{\text{data}}||p_\theta) = E_{x \sim p_{\text{data}}} \left[ \frac{\partial}{\partial \theta} f_\theta(x) \right] - E_{x \sim p_\theta} \left[ \frac{\partial}{\partial \theta} f_\theta(x) \right].
\]

(3)

The expectations can be approximated by averaging over 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 (Girolami & Calderhead, 2011)), which iterates

\[
x^{t+1} = x^t + \frac{\delta^2}{2} \nabla_x f_\theta(x^t) + \delta \epsilon^t,
\]

(4)

where \( t \) indexes the time, \( \delta \) is the step size, and \( \epsilon^t \sim \mathcal{N}(0, I) \). The difficulty lies in the fact that for high-dimensional and multi-modal distributions, MCMC sampling can take a long time to converge, and the sampling chains may have difficulty traversing modes. As demonstrated in Figure 2 training EBMs with synthesized samples from non-convergent MCMC results in malformed energy landscapes (Nijkamp et al., 2019b), even if the samples from the model look reasonable. Qiu et al. (2019) proposes coupled MCMC to tackle this problem.

3 Recovery Likelihood

3.1 From Marginal to Conditional

Given the difficulty of sampling from the marginal density \( p_\theta(x) \), following Bengio et al. (2013), we use the recovery likelihood defined by the density of the observed sample conditional on a noisy sample perturbed by isotropic Gaussian noise. Specifically, let \( \tilde{x} = x + \sigma \epsilon \) be the noisy observation of \( x \), where \( \epsilon \sim \mathcal{N}(0, I) \). Suppose \( p_\theta(x) \) is defined by the EBM in equation 1 then the conditional EBM can be derived as

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

(5)
where \( \tilde{Z}_\theta(\tilde{x}) = \int \exp \left( f_\theta(x) - \frac{1}{2\sigma^2} \| \tilde{x} - x \|^2 \right) \, dx \) is the partition function of this conditional EBM. See Appendix \[A.1\] for the derivation. Compared to \( p_0(x) \) (equation \[1\]), the extra quadratic term \( \frac{1}{2\sigma^2} \| \tilde{x} - x \|^2 \) in \( p_0(x|\tilde{x}) \) constrains the energy landscape to be localized around \( \tilde{x} \), making the latter less multi-modal and easier to sample from. As we will show later, when \( \sigma \) is small, \( p_0(x|\tilde{x}) \) is approximately a single mode Gaussian distribution, which greatly reduces the burden of MCMC.

A more general formulation is \( \tilde{x} = ax + \sigma \epsilon \), where \( a \) is a positive constant. In that case, we can let \( y = ax \), and treat \( y \) as the observed sample. Assume \( p_0(y) = \frac{1}{Z_0} \exp(f_0(y)) \), then by change of variable, the density function of \( x \) can be derived as \( g_\theta(x) = a p_0(ax) \).

### 3.2 Maximizing Recovery Likelihood

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

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

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 \[3\] we can maximize \( J(\theta) \), whose distributions are easier to sample from. Specifically, we generate synthesized samples by \( K \) steps of Langevin dynamics that iterates

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

The model is then updated following the same learning gradients as MLE (equation \[5\]), because the quadratic term \(-\frac{1}{2\sigma^2} \| \tilde{x} - x \|^2 \) is not related to \( \theta \). Following the classical analysis of MLE, we can show that the point estimate given by maximizing recovery likelihood is an unbiased estimator of the true parameters, which means that given enough data, a rich enough model and exact synthesis, maximizing the recovery likelihood learns \( \theta \) such that \( p_{\text{data}}(x) = p_\theta(x) \). See Appendix \[A.2\] for a theoretical explanation.

### 3.3 Normal Approximation to Conditional

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 at \( \tilde{x} \). Specifically, the negative conditional energy is

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

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

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

where \( c \) include terms irrelevant of \( x \) (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 \( \| \tilde{x} - x \|^2 / 2\sigma^2 \) will dominate all the second order terms from Taylor expansion. Thus we can approximate \( p_\theta(x|\tilde{x}) \) by a Gaussian approximation \( \bar{p}_\theta(x|\tilde{x}) \):

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

We can sample from this distribution using:

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

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 traits: (1) it verifies the fact that the conditional density \( p_\theta(x|\tilde{x}) \) can be generally easier to sample from when \( \sigma \) is small; (2) it provides hints of 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 (Vincent, 2011; Song & Ermon, 2019, 2020; Saremi et al., 2018; Saremi & Hyvarinen, 2019). Specifically, in diffusion probabilistic models, instead of modeling $p_0(x)$ as an energy-based model, it recruits variational inference and directly models the conditional density as

$$ p_0(x|\tilde{x}) \sim \mathcal{N}(\tilde{x} + \sigma^2 s_\theta(\tilde{x}), \sigma^2), $$

which is in agreement with the normal approximation (equation [1]), with $s_\theta(x) = \nabla_x f_\theta(x)$. On the other hand, the training objective of denoising score matching is to minimize

$$ \frac{1}{2 \sigma^2} \mathbb{E}_{p(x, \xi)}[\|x - (\tilde{x} + \sigma^2 s_\theta(\tilde{x}))\|^2], $$

where $s_\theta(x)$ is the score of the density of $\tilde{x}$. This objective is in agreement with the objective of maximizing log-likelihood of the normal approximation (equation [1]), except that for normal approximation, $\nabla_x f_\theta(\cdot)$ is the score of density of $x$, instead of $\tilde{x}$. However, the difference between the scores of density of $x$ and $\tilde{x}$ is of $O(\sigma^2)$, which is negligible when $\sigma$ is sufficiently small (see Appendix [A.4] 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.3]). It indicates that the training process of maximizing recovery likelihood agrees with the one of diffusion probabilistic models and denoising score matching 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, which is also reported in Ho et al. (2020) and Song & Ermon (2020). In contrast, the 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

As we discuss, sampling from $p_0(x|\tilde{x})$ becomes simple only when $\sigma$ is small. In the extreme case when $\sigma \to \infty$, $p_0(x|\tilde{x})$ converges to the marginal distribution $p_0(x)$, which is again highly multimodal and difficult to sample from. To keep $\sigma$ small and meanwhile equip the model with the ability to generate new samples initialized from white noise, inspired by Sohl-Dickstein et al. (2015) and Ho et al. (2020), we propose to learn a sequence of recovery likelihoods, on gradually perturbed observed data based on a diffusion process. Specifically, assume a sequence of perturbed observations $x_0, x_1, \ldots, x_T$ such that

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

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

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

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

We follow the learning algorithm in section [3.2]. A question is how to determine the step size schedule $\delta_t$ of Langevin dynamics. Inspired by the sampling procedure of the normal approximation (equation [12]), we set the step size $\delta_t = b \sigma_t$, where $b < 1$ is a tuned hyperparameter. This schedule turns out to work well in practice. Thus the $K$ steps of Langevin dynamics iterates

$$ y_{t+1}^* = y_t^* + \frac{b^2 \sigma_t^2}{2} (\nabla y f_\theta(y_t^*, t) + \frac{1}{\sigma_t^2} (x_{t+1} - y_t^*)) + b \sigma_t \epsilon^*.$$

Algorithm [1] summarizes the training procedure. After training, we initialize the MCMC sampling from Gaussian white noise, and the synthesized sample at each time step serves to initialize the MCMC that samples from the model of the previous time step. See algorithm [2] to show the efficacy of our method, Figures [3] and [2] display several 2D toy examples learned by diffusion recovery likelihood.
Algorithm 1 Training

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 Progressive sampling

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 is flexible for diffusion process of various magnitudes of noise, we test the method under two settings: (1) $T = 6$ with $K = 30$ steps of Langevin dynamic per time step, $b = 0.0002$; (2) $T = 1000$, with sampling from normal approximation. (2) resembles the noise schedule of Ho et al. (2020) and the magnitude of noise added at each time step is much smaller compared to (1). For both settings, we set $\sigma^2_t$ to increase linearly. The network structure of $f_\theta(x, t)$ is based on Wide ResNet (Zagoruyko & Komodakis, 2016) and we remove weight normalization. $t$ is encoded by Transformer sinusoidal position embedding as in (Ho et al., 2020). Architecture and training details are in Appendix B. Henceforth we simply refer the two settings as $T6$ and $T1k$.

4.1 IMAGE GENERATION

Figures 1 and 4 display uncurated samples generated from learned models on CIFAR-10, CelebA $64 \times 64$, LSUN $64 \times 64$ and $128 \times 128$ datasets under $T6$ setting. The samples are of high fidelity and comparable to GAN-based methods. Appendix C.4 provides more generated samples. Tables 1 and 3 summarize the quantitative evaluations on CIFAR-10 and CelebA datasets, in terms of Frechet Inception Distance (FID) (Heusel et al., 2017) and inception scores (Salimans et al., 2016). On CIFAR-10, our model achieves FID 9.60 and inception score 8.58, which outperforms existing methods of learning explicit energy-based models to a large extent, and is superior to a majority of GAN-based methods. On CelebA, our model obtains results comparable with the state-of-the-art GAN-based methods, and outperforms score-based methods (Song & Ermon, 2019, 2020). Note that the score-based methods (Song & Ermon, 2019, 2020) and diffusion probabilistic models (Ho et al., 2020) directly parametrize and learn the score of data distribution, whereas our goal is to learn explicit energy-based models.

Figure 4: Generated samples on unconditional CIFAR-10 (left) and LSUN $64^2$ church_outdoor (center) and LSUN $64^2$ bedroom (right).
Table 1: FID and inception scores on CIFAR-10.

<table>
<thead>
<tr>
<th>Model</th>
<th>FID↓</th>
<th>Inception↑</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>GAN-based</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WGAN-GP (Gulrajani et al., 2017)</td>
<td>36.4</td>
<td>7.86 ± 0.07</td>
</tr>
<tr>
<td>SNGAN (Miyato et al., 2018)</td>
<td>21.7</td>
<td>8.22 ± 0.05</td>
</tr>
<tr>
<td>SNGAN-DDLS (Che et al., 2020)</td>
<td>15.42</td>
<td>9.09 ± 0.10</td>
</tr>
<tr>
<td>StyleGAN2-ADA (Karras et al., 2020)</td>
<td>3.26</td>
<td><strong>9.74</strong> ± 0.05</td>
</tr>
<tr>
<td><strong>Score-based</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NCSN (Song &amp; Ermon, 2019)</td>
<td>25.32</td>
<td>8.87 ± 0.12</td>
</tr>
<tr>
<td>NCSN-v2 (Song &amp; Ermon, 2020)</td>
<td>10.87</td>
<td>8.40 ± 0.07</td>
</tr>
<tr>
<td>DDPM (Ho et al., 2020)</td>
<td><strong>3.17</strong></td>
<td>9.46 ± 0.11</td>
</tr>
<tr>
<td><strong>Explicit EBM-conditional</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CoopNets (Xie et al., 2019)</td>
<td>-</td>
<td>7.30</td>
</tr>
<tr>
<td>EBM-IG (Du &amp; Mordatch, 2019)</td>
<td>37.9</td>
<td>8.30</td>
</tr>
<tr>
<td>JEM (Grathwohl et al., 2019)</td>
<td>38.4</td>
<td>8.76</td>
</tr>
<tr>
<td><strong>Explicit EBM</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Multi-grid (Gao et al., 2018)</td>
<td>40.01</td>
<td>6.56</td>
</tr>
<tr>
<td>CoopNets (Xie et al., 2016a)</td>
<td>33.61</td>
<td>6.55</td>
</tr>
<tr>
<td>EBM-SR (Nijkamp et al., 2019b)</td>
<td>-</td>
<td>6.21</td>
</tr>
<tr>
<td>EBM-IG (Du &amp; Mordatch, 2019)</td>
<td>38.2</td>
<td>6.78</td>
</tr>
<tr>
<td><strong>Ours (T6)</strong></td>
<td><strong>9.60</strong></td>
<td><strong>8.58</strong> ± 0.12</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Setting / Objective</th>
<th>FID↓</th>
<th>Inception↑</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T = 1, K = 180$</td>
<td>32.12</td>
<td>6.89 ± 0.08</td>
</tr>
<tr>
<td>$T = 1000, K = 0$</td>
<td>22.58</td>
<td>7.86 ± 0.11</td>
</tr>
<tr>
<td>$T = 1000, K = 0$ (DSM)</td>
<td>21.76</td>
<td>7.80 ± 0.07</td>
</tr>
<tr>
<td>$T = 6, K = 10$</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>$T = 6, K = 30$</td>
<td>9.60</td>
<td>8.58 ± 0.12</td>
</tr>
<tr>
<td>$T = 6, K = 50$</td>
<td><strong>9.36</strong></td>
<td><strong>8.68</strong> ± 0.11</td>
</tr>
</tbody>
</table>

Table 3: FID scores on CelebA 64$^2$.

<table>
<thead>
<tr>
<th>Model</th>
<th>FID↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>QA-GAN (Parimala &amp; Channappaya, 2019)</td>
<td>6.42</td>
</tr>
<tr>
<td>COCO-GAN (Lin et al., 2019)</td>
<td><strong>4.0</strong></td>
</tr>
<tr>
<td>NVAE (Vahdat &amp; Kautz, 2020)</td>
<td>14.74</td>
</tr>
<tr>
<td>NCSN (Song &amp; Ermon, 2019)</td>
<td>25.30</td>
</tr>
<tr>
<td>NCSN-v2 (Song &amp; Ermon, 2020)</td>
<td>10.23</td>
</tr>
<tr>
<td>EBM-SR (Nijkamp et al., 2019b)</td>
<td>23.02</td>
</tr>
<tr>
<td>EBM-Triangle (Han et al., 2020)</td>
<td>24.70</td>
</tr>
<tr>
<td><strong>Ours (T6)</strong></td>
<td>5.98</td>
</tr>
</tbody>
</table>

Table 4: Test bits per dimension on CIFAR-10. † indicates that we estimate the bit per dimension with the approximated log partition function instead of analytically computing it, and thus should be take with a grain of salt. See section 4.3

<table>
<thead>
<tr>
<th>Model</th>
<th>BPD↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>DDPM (Ho et al., 2020)</td>
<td>3.70</td>
</tr>
<tr>
<td>Glow (Kingma &amp; Dhariwal, 2018)</td>
<td>3.35</td>
</tr>
<tr>
<td>Flow++ (Ho et al., 2019)</td>
<td>3.08</td>
</tr>
<tr>
<td>GPixelCNN (Van den Oord et al., 2016)</td>
<td>3.03</td>
</tr>
<tr>
<td>Sparse Transformer (Child et al., 2019)</td>
<td>2.80</td>
</tr>
<tr>
<td>DistAug (Jun et al., 2020)</td>
<td><strong>2.56</strong></td>
</tr>
<tr>
<td><strong>Ours† (T1k)</strong></td>
<td>3.18</td>
</tr>
</tbody>
</table>

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$^2$.

Interpolation. As shown in Figure 5, our model is capable of smooth interpolation between two generated samples. Specifically, for two samples $x_{0}(0)$ and $x_{0}(1)$, we do a sphere interpolation between...
the initial white noise images $x_t^{(0)}$ and $x_t^{(1)}$, and the noise terms of Langevin dynamics $\epsilon_t^{(0)}$ and $\epsilon_t^{(1)}$ for every sampling step at every time step. More interpolation results can be found in Appendix C.2.

**Image inpainting.** A promising application of energy-based models is to use the learned model as a prior model 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)]. In Figure 6 we demonstrate that the learned models by maximizing recovery likelihoods are capable of realistic and semantically meaningful image inpainting. 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

Table 2 summarizes the results of ablation study on CIFAR-10. We investigate the effect of choosing different number of time steps $T$ and number of sampling steps $K$. 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 marginal likelihood. The approach is adopted by [Nijkamp et al. (2019b); Du & Mordatch (2019)]. For fair comparison, we equip the baseline method the same budget of MCMC sampling as our $T6$ setting (i.e., 180 sampling steps). Our method outperforms this baseline method by a large margin. Also the models can be trained more efficiently as the number of sampling steps per iteration is reduced and amortized by time steps.

Next, 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 the approximated normal distributions ($T=1000, K=0$ (DSM)). As mentioned in section 3.4, (2) is equivalent to the training objectives of denoising score matching (Song & Ermon, 2019; 2020) and 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 the score function is parametrized as the gradients of the explicit energy function used in our method. (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 verifies the fact that the 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 with many time steps, so that a more flexible schedule of time steps accompanied with certain amount of sampling steps is preferred.

Last, 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. The training becomes unstable when the number of sampling steps are not enough ($T = 6, K = 10$), and more sampling steps lead to better sample quality. However, since $K = 50$ does not gain significant improvement versus $K = 30$, yet of much higher computational cost, we keep $K = 30$ for image generation on all datasets.

### 4.3 Long-run Chain Analysis

Besides achieving high quality generation, a perhaps equally important aspect of learning EBMs is to obtain a faithful energy potential. A principle way to check the validity of the learned potential is to perform long-run sampling chains and see if the samples still remain realistic. However, as pointed out in [Nijkamp et al. (2019a)], almost all existing methods of learning EBMs fail in getting realistic long-run chain samples. In this subsection, we demonstrate that by composing a thousand diffusion time steps ($Tlk$ setting), we can use MCMC to form steady long-run chains for the conditional distributions.
First we prepare a faithful sampler for conducting long-run sampling. Specifically, after training the model under $T1k$ setting by maximizing diffusion recovery likelihood, for each time step, we first sample from the normal approximation and count it as one sampling step, and then use Hamiltonian Monte Carlo (HMC) (Neal et al., 2011) with 2 leapfrog steps to perform the consecutive sampling steps. To obtain a reasonable schedule of sampling step size, for each time step we adaptively adjust the step size of HMC to make the average acceptance rate range in $[0.6, 0.9]$, 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 logarithmically. 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 FID 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 details.

More interestingly, given the faithful 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 $g_\theta(x_0) = \sqrt{1 - \sigma^2_1}p_\theta(\sqrt{1 - \sigma^2_1}x_0)$. 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, that will converge to the true value when the number of samples grows large.

5 Conclusion

We propose to learn EBMs by diffusion recovery likelihood, a variant of MLE applied to diffusion processes. 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. In future work, we look forward to investigating settings to combine the high quality synthesis of model-$T6$ and faithful long-run MCMC samples of model-$T1k$, to achieve the best of two worlds. Since this method can learn EBMs efficiently with small budget of MCMC, we are also interested in scaling it up to higher resolution images and investigating this method in other data modalities.
REFERENCES


A Extended derivations

A.1 Derivation of equation

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

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

(19)

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

(20)

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

(21)

where we absorb all the terms that are irrelevant of \( x \) as \( Z_\theta(x) \).

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:

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

(22)

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

\[
I_0(\theta) = I(\theta) + E_{p_\theta(x, \tilde{x})}[-\nabla^2 \log p_\theta(\tilde{x})].
\]

(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)

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

(26)

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

(27)

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

(28)

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

(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,

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

(30)

\[
\quad = E[h(x)] + h'(x)\epsilon + h''(x)\epsilon^2/2
\]

(31)

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

(32)
Integral by part,
\[
\mathbb{E}[h''(x)] = \int h''(x)p(x)dx = - \int h'(x)p'(x)dx = \int p'(x)h(x)dx.
\]
Thus we have the heat equation
\[
\tilde{p}(x) = p(x) + p''(x)\sigma^2/2.
\] (34)
The score
\[
\nabla_x \log \tilde{p}(x) = \nabla_x \log p(x) + \nabla_x \log (1 + \sigma^2/\sigma^2/2)
\]
\[
\approx \nabla_x \log p(x) + \nabla_x [\sigma^2/\sigma^2/2].
\] (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 \( x_{obs} \) is
\[
\nabla_\theta \left( \frac{1}{2\sigma^2} \| x_{obs} - (\bar{x} + \sigma^2 f_\theta(\bar{x})) \|^2 \right) = \nabla_\theta f_\theta(\bar{x})(x_{obs} - (\bar{x} + \sigma^2 f_\theta(\bar{x}))).
\] (37)
On the other hand, to maximize the original recovery likelihood, suppose we sample \( x_{syn} \sim p_\theta(x|\bar{x}) \), then the gradient ascent of the original recovery log-likelihood is
\[
\nabla_\theta f_\theta(x_{obs}) - \mathbb{E}[\nabla_\theta f_\theta(x_{syn})] = h_\theta(x_{obs}) - \mathbb{E}[h_\theta(x_{syn})],
\] (38)
where \( h_\theta(x) = \nabla_\theta f_\theta(x) \). Approximately, if we perform one step of Langevin dynamics from \( \bar{x} \) to obtain \( x_{syn} \), i.e., \( x_{syn} = \bar{x} + \sigma^2 f_\theta(\bar{x}) + \sqrt{2}e \), and assume \( f_\theta(x) \) is locally linear in \( x \), then
\[
\nabla_\theta f_\theta(x_{obs}) - \mathbb{E}[\nabla_\theta f_\theta(x_{init})]
\]
\[
= h_\theta(x_{obs}) - \mathbb{E}[h_\theta(\bar{x} + \sigma^2 f_\theta(\bar{x}) + \sigma e)]
\] (40)
\[
\approx h_\theta(\bar{x}) + h_\theta(\bar{x})(x_{obs} - \bar{x}) - \mathbb{E}[h_\theta(\bar{x}) + h_\theta(\bar{x})(\sigma^2 f_\theta(\bar{x}) + \sigma e)]
\] (41)
\[
= h_\theta(\bar{x})(x_{obs} - (\bar{x} + \sigma^2 f_\theta(\bar{x})))
\] (42)
\[
= \nabla_\theta f_\theta(\bar{x})(x_{obs} - (\bar{x} + \sigma^2 f_\theta(\bar{x}))).
\] (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 \( y_t = \sqrt{1 - \sigma^2_{t+1} x_t} \) to estimate the partition function. Specifically, the marginal distribution of \( y_t \) is
\[
p_\theta(y_t) = \frac{1}{Z_{\theta,t}} \exp(f_\theta(y_t, t))
\] (44)
We can estimate the ratio of the partition functions at two consecutive time steps using importance sampling
\[
\frac{Z_{\theta,t}}{Z_{\theta,t+1}} = \mathbb{E}_{p_\theta(y_{t+1})} [\exp(f_\theta(y, t) - f_\theta(y, t + 1))]
\] (45)
\[
\approx \frac{1}{M} \sum_{i=1}^{M} [\exp(f_\theta(y_{t+1,i}, t) - f_\theta(y_{t+1,i}, t + 1))],
\] (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, then by second order Taylor expansion,

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

(49)

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

(50)

By law of large number, \( \text{Var}(A) \rightarrow 0 \) as \( M \rightarrow \infty \), and thus \( \mathbb{E}[\log A] \rightarrow \log \mu \). This is also consistent with the estimation curves in the right subfigure of Figure 8, since \( \text{Var}(A) \geq 0 \), 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 is 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.

<table>
<thead>
<tr>
<th>Resolution</th>
<th>(a) $32 \times 32$</th>
<th>(b) $64 \times 64$</th>
<th>(c) $128 \times 128$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$3 \times 3$ Conv2D, 128</td>
<td>$3 \times 3$ Conv2D, 128</td>
<td>$3 \times 3$ Conv2D, 128</td>
<td></td>
</tr>
<tr>
<td>$N$ ResBlocks, 128 Downsample $2 \times 2$</td>
<td>$N$ ResBlocks, 128 Downsample $2 \times 2$</td>
<td>$N$ ResBlocks, 128 Downsample $2 \times 2$</td>
<td></td>
</tr>
<tr>
<td>$N$ ResBlocks, 256 Downsample $2 \times 2$</td>
<td>$N$ ResBlocks, 256 Downsample $2 \times 2$</td>
<td>$N$ ResBlocks, 256 Downsample $2 \times 2$</td>
<td></td>
</tr>
<tr>
<td>$N$ ResBlocks, 256 Downsample $2 \times 2$</td>
<td>$N$ ResBlocks, 256 Downsample $2 \times 2$</td>
<td>$N$ ResBlocks, 256 Downsample $2 \times 2$</td>
<td></td>
</tr>
<tr>
<td>$N$ ResBlocks, 256 Downsample $2 \times 2$</td>
<td>$N$ ResBlocks, 512 Downsample $2 \times 2$</td>
<td>$N$ ResBlocks, 512 Downsample $2 \times 2$</td>
<td></td>
</tr>
<tr>
<td>ReLU, global sum Dense 1</td>
<td>ReLU, global sum Dense 1</td>
<td>ReLU, global sum Dense 1</td>
<td></td>
</tr>
<tr>
<td>(d) Time embedding (temb)</td>
<td>(e) ResBlock</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sinusoidal embedding Dense, leakyReLU</td>
<td>leakyReLU, $3 \times 3$ Conv2D + Dense(leakyReLU(temb))</td>
<td></td>
<td></td>
</tr>
<tr>
<td>+ input</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: Hyperparameters of various datasets.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>$N$</th>
<th>$\beta_1$ in Adam</th>
<th>Batch size</th>
<th>Training iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>5</td>
<td>0.9</td>
<td>256</td>
<td>50k</td>
</tr>
<tr>
<td>CelebA</td>
<td>2</td>
<td>0.9</td>
<td>128</td>
<td>100k</td>
</tr>
<tr>
<td>LSUN church_outdoor $64^2$</td>
<td>2</td>
<td>0.9</td>
<td>128</td>
<td>100k</td>
</tr>
<tr>
<td>LSUN bedroom $64^2$</td>
<td>2</td>
<td>0.9</td>
<td>128</td>
<td>100k</td>
</tr>
<tr>
<td>LSUN church_outdoor $128^2$</td>
<td>2</td>
<td>0.5</td>
<td>64</td>
<td>100k</td>
</tr>
<tr>
<td>LSUN bedroom $128^2$</td>
<td>5</td>
<td>0.5</td>
<td>64</td>
<td>56k</td>
</tr>
</tbody>
</table>

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 × 64.

C.3 Additional Image Inpainting Results

Figures 14 and 15 show additional examples of image inpainting on CelebA 64² and LSUN church_outdoor 128².

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 64², LSUN church_outdoor 128², LSUN bedroom 128², LSUN church_outdoor 64² and LSUN bedroom 64² 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 $128 \times 128$. 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