

使用非平衡热力学原理构造深度无监督学习。  
先对数据加噪，再用一系列去噪白噪声。

## Deep Unsupervised Learning using Nonequilibrium Thermodynamics

Jascha Sohl-Dickstein

Stanford University

JASCHA@STANFORD.EDU

Eric A. Weiss

University of California, Berkeley

EWEISS@BERKELEY.EDU

Niru Maheswaranathan

Stanford University

NIRUM@STANFORD.EDU

Surya Ganguli

Stanford University

SGANGULI@STANFORD.EDU

### Abstract

A central problem in machine learning involves modeling complex data-sets using highly flexible families of probability distributions in which learning, sampling, inference, and evaluation are still analytically or computationally tractable. Here, we develop an approach that simultaneously achieves both flexibility and tractability. The essential idea, inspired by non-equilibrium statistical physics, is to systematically and slowly destroy structure in a data distribution through an iterative forward diffusion process. We then learn a reverse diffusion process that restores structure in data, yielding a highly flexible and tractable generative model of the data. This approach allows us to rapidly learn, sample from, and evaluate probabilities in deep generative models with thousands of layers or time steps, as well as to compute conditional and posterior probabilities under the learned model. We additionally release an open source reference implementation of the algorithm.

### 1. Introduction

Historically, probabilistic models suffer from a tradeoff between two conflicting objectives: *tractability* and *flexibility*. Models that are *tractable* can be analytically evaluated and easily fit to data (e.g. a Gaussian or Laplace). However,

these models are unable to aptly describe structure in rich datasets. On the other hand, models that are *flexible* can be molded to fit structure in arbitrary data. For example, we can define models in terms of any (non-negative) function  $\phi(\mathbf{x})$  yielding the flexible distribution  $p(\mathbf{x}) = \frac{\phi(\mathbf{x})}{Z}$ , where  $Z$  is a normalization constant. However, computing this normalization constant is generally intractable. Evaluating, training, or drawing samples from such flexible models typically requires a very expensive Monte Carlo process.

A variety of analytic approximations exist which ameliorate, but do not remove, this tradeoff—for instance mean field theory and its expansions (T, 1982; Tanaka, 1998), variational Bayes (Jordan et al., 1999), contrastive divergence (Welling & Hinton, 2002; Hinton, 2002), minimum probability flow (Sohl-Dickstein et al., 2011b;a), minimum KL contraction (Lyu, 2011), proper scoring rules (Gneiting & Raftery, 2007; Parry et al., 2012), score matching (Hyvärinen, 2005), pseudolikelihood (Besag, 1975), loopy belief propagation (Murphy et al., 1999), and many, many more. Non-parametric methods (Gershman & Blei, 2012) can also be very effective<sup>1</sup>.

#### 1.1. Diffusion probabilistic models

We present a novel way to define probabilistic models that allows:

1. extreme flexibility in model structure,
2. exact sampling,

<sup>1</sup>Non-parametric methods can be seen as transitioning smoothly between tractable and flexible models. For instance, a non-parametric Gaussian mixture model will represent a small amount of data using a single Gaussian, but may represent infinite data as a mixture of an infinite number of Gaussians.

3. easy multiplication with other distributions, e.g. in order to compute a posterior, and
4. the model log likelihood, and the probability of individual states, to be cheaply evaluated.

Our method uses a Markov chain to gradually convert one distribution into another, an idea used in non-equilibrium statistical physics (Jarzynski, 1997) and sequential Monte Carlo (Neal, 2001). We build a generative Markov chain which converts a simple known distribution (e.g. a Gaussian) into a target (data) distribution using a diffusion process. Rather than use this Markov chain to approximately evaluate a model which has been otherwise defined, we explicitly define the probabilistic model as the endpoint of the Markov chain. Since each step in the diffusion chain has an analytically evaluable probability, the full chain can also be analytically evaluated.

Learning in this framework involves estimating small perturbations to a diffusion process. Estimating small perturbations is more tractable than explicitly describing the full distribution with a single, non-analytically-normalizable, potential function. Furthermore, since a diffusion process exists for any smooth target distribution, this method can capture data distributions of arbitrary form.

We demonstrate the utility of these *diffusion probabilistic models* by training high log likelihood models for a two-dimensional swiss roll, binary sequence, handwritten digit (MNIST), and several natural image (CIFAR-10, bark, and dead leaves) datasets.

## 1.2. Relationship to other work

The wake-sleep algorithm (Hinton, 1995; Dayan et al., 1995) introduced the idea of training inference and generative probabilistic models against each other. This approach remained largely unexplored for nearly two decades, though with some exceptions (Sminchisescu et al., 2006; Kavukcuoglu et al., 2010). There has been a recent explosion of work developing this idea. In (Kingma & Welling, 2013; Gregor et al., 2013; Rezende et al., 2014; Ozair & Bengio, 2014) variational learning and inference algorithms were developed which allow a flexible generative model and posterior distribution over latent variables to be directly trained against each other.

The variational bound in these papers is similar to the one used in our training objective and in the earlier work of (Sminchisescu et al., 2006). However, our motivation and model form are both quite different, and the present work retains the following differences and advantages relative to these techniques:

1. We develop our framework using ideas from physics, quasi-static processes, and annealed importance sampling rather than from variational Bayesian methods.

2. We show how to easily multiply the learned distribution with another probability distribution (eg with a conditional distribution in order to compute a posterior)
3. We address the difficulty that training the inference model can prove particularly challenging in variational inference methods, due to the asymmetry in the objective between the inference and generative models. We restrict the forward (inference) process to a simple functional form, in such a way that the reverse (generative) process will have the same functional form.
4. We train models with thousands of layers (or time steps), rather than only a handful of layers.
5. We provide upper and lower bounds on the entropy production in each layer (or time step)

There are a number of related techniques for training probabilistic models (summarized below) that develop highly flexible forms for generative models, train stochastic trajectories, or learn the reversal of a Bayesian network. Reweighted wake-sleep (Bornschein & Bengio, 2015) develops extensions and improved learning rules for the original wake-sleep algorithm. Generative stochastic networks (Bengio & Thibodeau-Laufer, 2013; Yao et al., 2014) train a Markov kernel to match its equilibrium distribution to the data distribution. Neural autoregressive distribution estimators (Larochelle & Murray, 2011) (and their recurrent (Uria et al., 2013a) and deep (Uria et al., 2013b) extensions) decompose a joint distribution into a sequence of tractable conditional distributions over each dimension. Adversarial networks (Goodfellow et al., 2014) train a generative model against a classifier which attempts to distinguish generated samples from true data. A similar objective in (Schmidhuber, 1992) learns a two-way mapping to a representation with marginally independent units. In (Rippel & Adams, 2013; Dinh et al., 2014) bijective deterministic maps are learned to a latent representation with a simple factorial density function. In (Stuhlmüller et al., 2013) stochastic inverses are learned for Bayesian networks. Mixtures of conditional Gaussian scale mixtures (MCGSMs) (Theis et al., 2012) describe a dataset using Gaussian scale mixtures, with parameters which depend on a sequence of causal neighborhoods. There is additionally significant work learning flexible generative mappings from simple latent distributions to data distributions – early examples including (MacKay, 1995) where neural networks are introduced as generative models, and (Bishop et al., 1998) where a stochastic manifold mapping is learned from a latent space to the data space. We will compare experimentally against adversarial networks and MCGSMs.

Related ideas from physics include the Jarzynski equality (Jarzynski, 1997), known in machine learning as An-

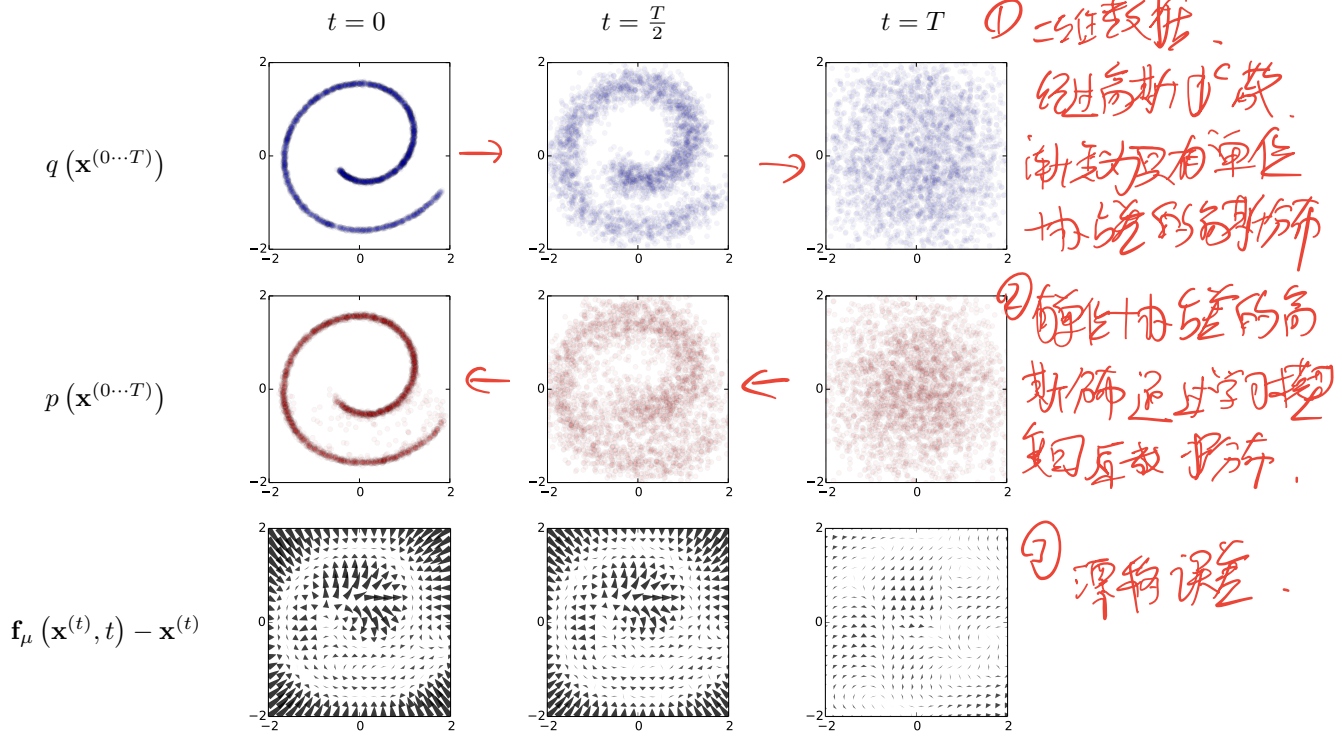


Figure 1. The proposed modeling framework trained on 2-d swiss roll data. The top row shows time slices from the forward trajectory  $q(\mathbf{x}^{(0\dots T)})$ . The data distribution (left) undergoes Gaussian diffusion, which gradually transforms it into an identity-covariance Gaussian (right). The middle row shows the corresponding time slices from the trained reverse trajectory  $p(\mathbf{x}^{(0\dots T)})$ . An identity-covariance Gaussian (right) undergoes a Gaussian diffusion process with learned mean and covariance functions, and is gradually transformed back into the data distribution (left). The bottom row shows the drift term,  $\mathbf{f}_\mu(\mathbf{x}^{(t)}, t) - \mathbf{x}^{(t)}$ , for the same reverse diffusion process.

nealed Importance Sampling (AIS) (Neal, 2001), which uses a Markov chain which slowly converts one distribution into another to compute a ratio of normalizing constants. In (Burda et al., 2014) it is shown that AIS can also be performed using the reverse rather than forward trajectory. Langevin dynamics (Langevin, 1908), which are the stochastic realization of the Fokker-Planck equation, show how to define a Gaussian diffusion process which has any target distribution as its equilibrium. In (Suykens & Vandewalle, 1995) the Fokker-Planck equation is used to perform stochastic optimization. Finally, the Kolmogorov forward and backward equations (Feller, 1949) show that forward and reverse diffusion processes can be described using the same functional form. The Kolmogorov forward equation corresponds to the Fokker-Planck equation, while the Kolmogorov backward equation describes the time-reversal of this diffusion process, but requires knowing gradients of the density function as a function of time.

## 2. Algorithm

Our goal is to define a forward (or inference) diffusion process which converts any complex data distribution into a

simple, tractable, distribution, and then learn a finite-time reversal of this diffusion process which defines our generative model distribution (See Figure 1). We first describe the forward, inference diffusion process. We then show how the reverse, generative diffusion process can be trained and used to evaluate probabilities. We also derive entropy bounds for the reverse process, and show how the learned distributions can be multiplied by any second distribution (e.g. as would be done to compute a posterior when inpainting or denoising an image).

### 2.1. Forward Trajectory

We label the data distribution  $q(\mathbf{x}^{(0)})$ . The data distribution is gradually converted into a well behaved (analytically tractable) distribution  $\pi(\mathbf{y})$  by repeated application of a Markov diffusion kernel  $T_\pi(\mathbf{y}|\mathbf{y}'; \beta)$  for  $\pi(\mathbf{y})$ , where  $\beta$  is the diffusion rate,

$$\pi(\mathbf{y}) = \int d\mathbf{y}' T_\pi(\mathbf{y}|\mathbf{y}'; \beta) \pi(\mathbf{y}') \quad (1)$$

$$q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) = T_\pi(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}; \beta_t). \quad (2)$$

3个扶过型: 状态转移核函数, 又叫做前核

# ① 前向扩散

$$\pi(\mathbf{y}) = \int d\mathbf{y}' T_{\pi}(\mathbf{y}|\mathbf{y}'; \beta) \pi(\mathbf{y}') \quad (1)$$

(1) → 当数据为平稳分布, 再应用扩散核不变分布

$$q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) = T_{\pi}(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}; \beta_t) \quad (2)$$

(2) → 通过反复应用扩散核, 原始数据即转变为平稳分布

扩散核 → 可以是高斯/二值函数

$$q(\mathbf{x}^{(0:T)}) = q(\mathbf{x}^{(0)}) \prod_{t=1}^T q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)}) \quad (3)$$

扩散结果, 向前转移

表示从  $t=0$  到  $t=T$  的整个轨迹的联合分布

# ② 逆向扩散

扩散核使用高斯/二值函数  
正向与逆向函数几乎一致

$$p(\mathbf{x}^{(T)}) = \pi(\mathbf{x}^{(T)}) \quad (4)$$

(4) → 末端分布为平稳分布

$$p(\mathbf{x}^{(0:T)}) = p(\mathbf{x}^{(T)}) \prod_{t=1}^T p(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}) \quad (5)$$

还原结果

$$p(\mathbf{x}^{(0)}) = \int d\mathbf{x}^{(1:T)} p(\mathbf{x}^{(0:T)}) \quad (6)$$

此积分无法计算, 但用正向扩散核估计

$$p(\mathbf{x}^{(0)}) = \int d\mathbf{x}^{(1:T)} p(\mathbf{x}^{(0:T)}) \frac{q(\mathbf{x}^{(1:T)}|\mathbf{x}^{(0)})}{q(\mathbf{x}^{(1:T)}|\mathbf{x}^{(0)})} \quad (7)$$

$$= \int d\mathbf{x}^{(1:T)} q(\mathbf{x}^{(1:T)}|\mathbf{x}^{(0)}) \frac{p(\mathbf{x}^{(0:T)})}{q(\mathbf{x}^{(1:T)}|\mathbf{x}^{(0)})} \quad (8)$$

$\beta$  无限大时, 正向核和反向核几乎一致

$$= \int d\mathbf{x}^{(1:T)} q(\mathbf{x}^{(1:T)}|\mathbf{x}^{(0)}) p(\mathbf{x}^{(0:T)}) \quad (9)$$

此时可利用  $q(\mathbf{x}^{(1:T)}|\mathbf{x}^{(0)})$  来估计  $p(\mathbf{x}^{(0)})$

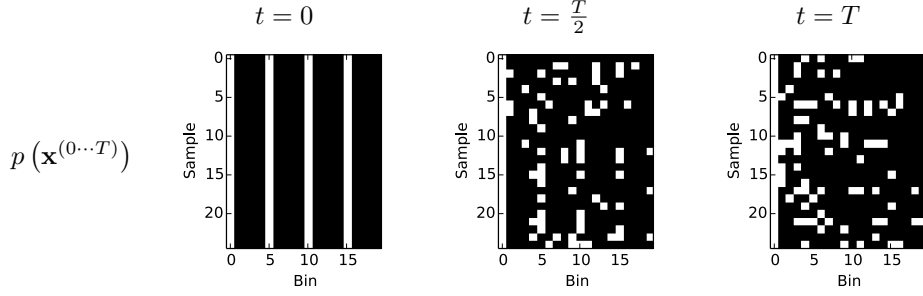


Figure 2. Binary sequence learning via binomial diffusion. A binomial diffusion model was trained on binary ‘heartbeat’ data, where a pulse occurs every 5th bin. Generated samples (left) are identical to the training data. The sampling procedure consists of initialization at independent binomial noise (right), which is then transformed into the data distribution by a binomial diffusion process, with trained bit flip probabilities. Each row contains an independent sample. For ease of visualization, all samples have been shifted so that a pulse occurs in the first column. In the raw sequence data, the first pulse is uniformly distributed over the first five bins.



Figure 3. The proposed framework trained on the CIFAR-10 (Krizhevsky & Hinton, 2009) dataset. (a) Example training data. (b) Random samples generated by the diffusion model.

The forward trajectory, corresponding to starting at the data distribution and performing  $T$  steps of diffusion, is thus

$$q(\mathbf{x}^{(0 \dots T)}) = q(\mathbf{x}^{(0)}) \prod_{t=1}^T q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)}) \quad (3)$$

For the experiments shown below,  $q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)})$  corresponds to either Gaussian diffusion into a Gaussian distribution with identity-covariance, or binomial diffusion into an independent binomial distribution. Table C.1 gives the diffusion kernels for both Gaussian and binomial distributions.

## 2.2. Reverse Trajectory

The generative distribution will be trained to describe the same trajectory, but in reverse,

$$p(\mathbf{x}^{(T)}) = \pi(\mathbf{x}^{(T)}) \quad (4)$$

$$p(\mathbf{x}^{(0 \dots T)}) = p(\mathbf{x}^{(T)}) \prod_{t=1}^T p(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)}) \quad (5)$$

For both Gaussian and binomial diffusion, for continuous diffusion (limit of small step size  $\beta$ ) the reversal of the diffusion process has the identical functional form as the

forward process (Feller, 1949). Since  $q(\mathbf{x}^{(t)} | \mathbf{x}^{(t-1)})$  is a Gaussian (binomial) distribution, and if  $\beta_t$  is small, then  $q(\mathbf{x}^{(t-1)} | \mathbf{x}^{(t)})$  will also be a Gaussian (binomial) distribution. The longer the trajectory the smaller the diffusion rate  $\beta$  can be made.

During learning only the mean and covariance for a Gaussian diffusion kernel, or the bit flip probability for a binomial kernel, need to be estimated. As shown in Table C.1,  $\mathbf{f}_\mu(\mathbf{x}^{(t)}, t)$  and  $\mathbf{f}_\Sigma(\mathbf{x}^{(t)}, t)$  are functions defining the mean and covariance of the reverse Markov transitions for a Gaussian, and  $\mathbf{f}_b(\mathbf{x}^{(t)}, t)$  is a function providing the bit flip probability for a binomial distribution. The computational cost of running this algorithm is the cost of these functions, times the number of time-steps. For all results in this paper, multi-layer perceptrons are used to define these functions. A wide range of regression or function fitting techniques would be applicable however, including nonparametric methods.

## 2.3. Model Probability

The probability the generative model assigns to the data is

$$p(\mathbf{x}^{(0)}) = \int d\mathbf{x}^{(1 \dots T)} p(\mathbf{x}^{(0 \dots T)}) \quad (6)$$

与数据的小, 数据分布在模型  
改变  
如行色各异  
板, 需位  
汁均值与协  
方差  
★又打= 很厉害, 需估计位高转  
板

Naively this integral is intractable – but taking a cue from annealed importance sampling and the Jarzynski equality, we instead evaluate the relative probability of the forward and reverse trajectories, averaged over forward trajectories,

$$p(\mathbf{x}^{(0)}) = \int d\mathbf{x}^{(1\cdots T)} p(\mathbf{x}^{(0\cdots T)}) \frac{q(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)})}{q(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)})} \quad (7)$$

$$= \int d\mathbf{x}^{(1\cdots T)} q(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)}) \frac{p(\mathbf{x}^{(0\cdots T)})}{q(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)})} \quad (8)$$

$$= \int d\mathbf{x}^{(1\cdots T)} q(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)}) \cdot p(\mathbf{x}^{(T)}) \prod_{t=1}^T \frac{p(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})}{q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)})} \quad (9)$$

β无限小时，正向分布和反向分布几乎一致。  
 如时可利用 q(x^{(1...T)}|x^{(0)}) 来估计 p(x^{(0)})

This can be evaluated rapidly by averaging over samples from the forward trajectory  $q(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)})$ . For infinitesimal  $\beta$  the forward and reverse distribution over trajectories can be made identical (see Section 2.2). If they are identical then only a *single* sample from  $q(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)})$  is required to exactly evaluate the above integral, as can be seen by substitution. This corresponds to the case of a quasi-static process in statistical physics (Spinney & Ford, 2013; Jarzynski, 2011).

### 2.4. Training

Training amounts to maximizing the model log likelihood,

$$L = \int d\mathbf{x}^{(0)} q(\mathbf{x}^{(0)}) \log p(\mathbf{x}^{(0)}) \quad (10)$$

$$= \int d\mathbf{x}^{(0)} q(\mathbf{x}^{(0)}) \cdot \log \left[ \frac{\int d\mathbf{x}^{(1\cdots T)} q(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)}) \cdot p(\mathbf{x}^{(T)}) \prod_{t=1}^T \frac{p(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})}{q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)})}}{p(\mathbf{x}^{(T)}) \prod_{t=1}^T \frac{p(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})}{q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)})}} \right], \quad (11)$$

which has a lower bound provided by Jensen's inequality,

$$L \geq \int d\mathbf{x}^{(0\cdots T)} q(\mathbf{x}^{(0\cdots T)}) \cdot \log \left[ p(\mathbf{x}^{(T)}) \prod_{t=1}^T \frac{p(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})}{q(\mathbf{x}^{(t)}|\mathbf{x}^{(t-1)})} \right]. \quad (12)$$

As described in Appendix B, for our diffusion trajectories this reduces to,

$$L \geq K \quad (13)$$

$$K = - \sum_{t=2}^T \int d\mathbf{x}^{(0)} d\mathbf{x}^{(t)} q(\mathbf{x}^{(0)}, \mathbf{x}^{(t)}) \cdot$$

$$D_{KL} \left( q(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}, \mathbf{x}^{(0)}) \parallel p(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}) \right) + H_q(\mathbf{X}^{(T)}|\mathbf{X}^{(0)}) - H_q(\mathbf{X}^{(1)}|\mathbf{X}^{(0)}) - H_p(\mathbf{X}^{(T)}). \quad (14)$$

当取与正向和逆向扩散完全相同时取等。

where the entropies and KL divergences can be analytically computed. The derivation of this bound parallels the derivation of the log likelihood bound in variational Bayesian methods.

As in Section 2.3 if the forward and reverse trajectories are identical, corresponding to a quasi-static process, then the inequality in Equation 13 becomes an equality.

Training consists of finding the reverse Markov transitions which maximize this lower bound on the log likelihood,

$$\max_{\hat{p}} \hat{p}(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)}) = \operatorname{argmax}_{p(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})} K. \quad (15)$$

The specific targets of estimation for Gaussian and binomial diffusion are given in Table C.1.

Thus, the task of estimating a probability distribution has been reduced to the task of performing regression on the functions which set the mean and covariance of a sequence of Gaussians (or set the state flip probability for a sequence of Bernoulli trials).

#### 2.4.1. SETTING THE DIFFUSION RATE $\beta_t$

The choice of  $\beta_t$  in the forward trajectory is important for the performance of the trained model. In AIS, the right schedule of intermediate distributions can greatly improve the accuracy of the log partition function estimate (Grosse et al., 2013). In thermodynamics the schedule taken when moving between equilibrium distributions determines how much free energy is lost (Spinney & Ford, 2013; Jarzynski, 2011).

In the case of Gaussian diffusion, we learn<sup>2</sup> the forward diffusion schedule  $\beta_{2\cdots T}$  by gradient ascent on  $K$ . The variance  $\beta_1$  of the first step is fixed to a small constant to prevent overfitting. The dependence of samples from  $q(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)})$  on  $\beta_{1\cdots T}$  is made explicit by using ‘frozen noise’ – as in (Kingma & Welling, 2013) the noise is treated as an additional auxiliary variable, and held constant while computing partial derivatives of  $K$  with respect to the parameters.

For binomial diffusion, the discrete state space makes gradient ascent with frozen noise impossible. We instead choose the forward diffusion schedule  $\beta_{1\cdots T}$  to erase a constant fraction  $\frac{1}{T}$  of the original signal per diffusion step, yielding a diffusion rate of  $\beta_t = (T - t + 1)^{-1}$ .

### 2.5. Multiplying Distributions, and Computing Posteriors

Tasks such as computing a posterior in order to do signal denoising or inference of missing values requires multipli-

<sup>2</sup>Recent experiments suggest that it is just as effective to instead use the same fixed  $\beta_t$  schedule as for binomial diffusion.

对有一些任务如降噪去噪来使用这种计算后训练更实时需可能的分布与模型分布约束。

估计: 均值 协方差 均值 协方差

防止过拟合。β\_1 是小。

噪声视为辅助变量，非扩散参数。

1. **公式 (10) 和 (11) - 对数似然:** 这些公式表示训练的目标是最大化模型的对数似然  $L$ 。对数似然是数据在给定模型下出现的概率的对数。最大化  $L$  意味着我们想找到参数使得观测数据出现的概率最大。
2. **公式 (12) - Jensen不等式的应用:** 利用Jensen不等式提供了一个  $L$  的下限。这意味着我们可以通过最大化这个下限来间接地最大化对数似然。这是变分推断方法中常用的策略, 因为直接最大化  $L$  可能是困难的或不可解的。
3. **公式 (13) 和 (14) - 对数似然的下限  $K$ :** 公式 (14) 进一步定义了  $K$  的具体形式。这包含了对每个时间步  $t$  的积分, 以及两个重要的概念: KL散度  $D_{KL}$  和熵  $H_q$ 。KL散度是衡量两个概率分布差异的指标, 而熵度量了分布的不确定性。公式中的  $H_q$  项表示在模型分布与真实分布间的熵差, 而  $D_{KL}$  表示扩散过程中的前后状态间的分布差异。
4. **公式 (15) - 最优化:** 训练的目标是找到最大化  $K$  的马尔可夫转移  $\hat{p}(x^{(t-1)}|x^{(t)})$ 。这相当于找到一个在每个时间步  $t$  上最好地解释数据的模型。
5. **训练多层感知器:** 为了最大化  $K$ , 使用多层感知器来逼近和优化反向马尔可夫转移的参数 (例如高斯分布的均值和协方差或二项式分布的翻转概率)。多层感知器是一种神经网络, 通过非线性变换能够逼近各种复杂的函数。在这种上下文中, 它被训练以最小化  $K$  的负数或者最大化  $K$  本身。
6. **模型训练的任务简化:** 最终, 训练模型的任务被简化为回归问题——在高斯扩散的情况下, 是对均值和协方差的回归, 在二项式扩散的情况下, 是对状态翻转概率的回归。

1. **扩散率  $\beta_t$  的选择:** 在前向轨迹中, 扩散率  $\beta_t$  对于模型性能至关重要。在高斯扩散过程中, 扩散率  $\beta_t$  通过梯度上升来学习, 以便找到最大化某个目标函数  $K$  的最佳值。
2. **高斯扩散的  $\beta_t$ :** 为了避免过拟合, 并确保模型学习到的不仅仅是噪声, 第一步的扩散率  $\beta_1$  被固定为一个小常数。这是因为第一步对学习过程的影响特别大。对于后续的步骤, 扩散率  $\beta_{2...T}$  是通过学习来适应的。
3. **冻结噪声 (Frozen Noise):** 冻结噪声是一种技术, 其中训练过程中的随机噪声被保持不变。在一些方法中, 例如 Kingma & Welling (2013) 提出的变分自编码器, 噪声是作为一个额外的参数加入到模型中, 而不是在每次采样时随机生成。这样可以在计算梯度时保持噪声的一致性, 从而使梯度计算变得可靠。这对于模型学习稳定的模式和结构是有帮助的。
4. **二项式扩散的  $\beta_t$ :** 由于二项式扩散涉及离散状态空间, 使得不能用同样的方式对噪声进行冻结。因此, 作者采用了一个不同的扩散率调度  $\beta_{1...T}$ 。具体来说, 每步的扩散率是前一步信号强度的一部分, 这导致了一个  $\beta_t$  的递增系列, 它与每个时间步骤呈倒数关系。

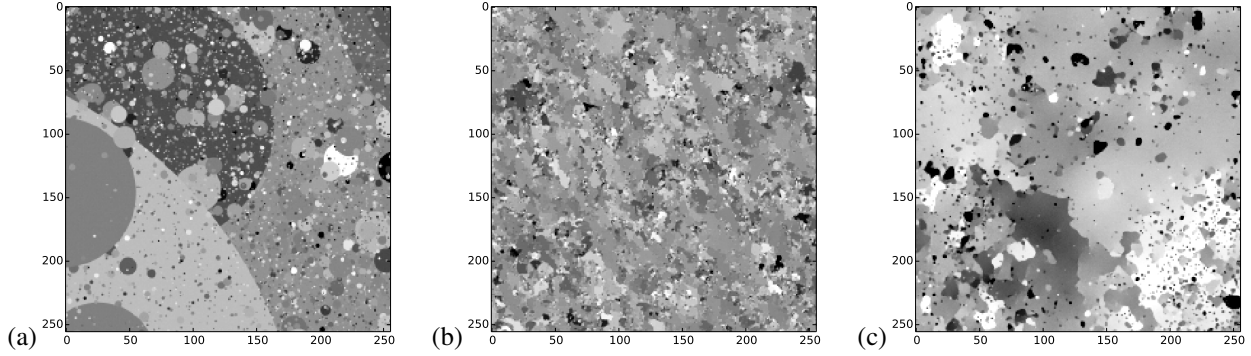


Figure 4. The proposed framework trained on dead leaf images (Jeulin, 1997; Lee et al., 2001). (a) Example training image. (b) A sample from the previous state of the art natural image model (Theis et al., 2012) trained on identical data, reproduced here with permission. (c) A sample generated by the diffusion model. Note that it demonstrates fairly consistent occlusion relationships, displays a multiscale distribution over object sizes, and produces circle-like objects, especially at smaller scales. As shown in Table 2, the diffusion model has the highest log likelihood on the test set.

对贝叶斯估计后验分布与先验分布和似然函数相乘  
 对贝叶斯估计  
 cation of the model distribution  $p(\mathbf{x}^{(0)})$  with a second distribution, or bounded positive function,  $r(\mathbf{x}^{(0)})$ , producing a new distribution  $\tilde{p}(\mathbf{x}^{(0)}) \propto p(\mathbf{x}^{(0)}) r(\mathbf{x}^{(0)})$ .

Multiplying distributions is costly and difficult for many techniques, including variational autoencoders, GSNs, NADEs, and most graphical models. However, under a diffusion model it is straightforward, since the second distribution can be treated either as a small perturbation to each step in the diffusion process, or often exactly multiplied into each diffusion step. Figure 5 demonstrates the use of a diffusion model to perform inpainting of a natural image. The following sections describe how to multiply distributions in the context of diffusion probabilistic models.

### 2.5.1. MODIFIED MARGINAL DISTRIBUTIONS

First, in order to compute  $\tilde{p}(\mathbf{x}^{(0)})$ , we multiply each of the intermediate distributions by a corresponding function  $r(\mathbf{x}^{(t)})$ . We use a tilde above a distribution or Markov transition to denote that it belongs to a trajectory that has been modified in this way.  $\tilde{q}(\mathbf{x}^{(0 \dots T)})$  is the modified forward trajectory, which starts at the distribution  $\tilde{q}(\mathbf{x}^{(0)}) = \frac{1}{\tilde{Z}_0} q(\mathbf{x}^{(0)}) r(\mathbf{x}^{(0)})$  and proceeds through the sequence of intermediate distributions

$$\tilde{q}(\mathbf{x}^{(t)}) = \frac{1}{\tilde{Z}_t} q(\mathbf{x}^{(t)}) r(\mathbf{x}^{(t)}), \quad (16)$$

where  $\tilde{Z}_t$  is the normalizing constant for the  $t$ th intermediate distribution.

### 2.5.2. MODIFIED CONDITIONAL DISTRIBUTIONS

Next, writing the relationship between the forward and reverse conditional distributions demonstrates how multiplying each intermediate distribution by  $r(\mathbf{x}^{(t)})$  changes the Markov diffusion chain. By Bayes' rule the forward chain

presented in Section 2.1 satisfies

$$q(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}) q(\mathbf{x}^{(t)}) = q(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)}) q(\mathbf{x}^{(t+1)}). \quad (17)$$

The new chain must instead satisfy

$$\tilde{q}(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}) \tilde{q}(\mathbf{x}^{(t)}) = \tilde{q}(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)}) \tilde{q}(\mathbf{x}^{(t+1)}). \quad (18)$$

As derived in Appendix C, one way to choose a new Markov chain which satisfies Equation 18 is to set

$$\tilde{q}(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}) \propto q(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)}) r(\mathbf{x}^{(t+1)}), \quad (19)$$

$$\tilde{q}(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)}) \propto q(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)}) r(\mathbf{x}^{(t)}). \quad (20)$$

So that  $\tilde{p}(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)})$  corresponds to  $\tilde{q}(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)})$ ,  $p(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)})$  is modified in the corresponding fashion,

$$\tilde{p}(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)}) \propto p(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)}) r(\mathbf{x}^{(t)}). \quad (21)$$

### 2.5.3. APPLYING $r(\mathbf{x}^{(t)})$

If  $r(\mathbf{x}^{(t)})$  is sufficiently smooth, then it can be treated as a small perturbation to the reverse diffusion kernel  $p(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)})$ . In this case  $\tilde{p}(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)})$  will have an identical functional form to  $p(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)})$ , but with perturbed mean and covariance for the Gaussian kernel, or with perturbed flip rate for the binomial kernel. The perturbed diffusion kernels are given in Table C.1.

If  $r(\mathbf{x}^{(t)})$  can be multiplied with a Gaussian (or binomial) distribution in closed form, then it can be directly multiplied with the reverse diffusion kernel  $p(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)})$  in closed form, and need not be treated as a perturbation. This



### 2.5.1. 修改边际分布

1. **计算  $\tilde{p}(x^{(0)})$ :** 在计算修改后的边际分布时, 原始的每个中间分布  $q(x^{(t)})$  都乘以一个函数  $r(x^{(t)})$ 。这个函数和转移概率相关, 反映了属于特定扩散轨迹或马尔可夫链的一个分布。新的分布  $\tilde{q}(x^{(0:T)})$  是通过对这些修改的中间分布的乘积来计算的。
2. **修改后的单步分布  $\tilde{q}(x^{(t)})$ :** 新的中间分布是原始分布乘以函数  $r$  再归一化的结果。

### 2.5.2. 修改条件分布

1. **条件分布的关系:** 公式(17)展示了未修改的条件分布关系, 而公式(18)展示了修改后的条件分布应满足的新关系。
2. **选择符合条件的马尔可夫链:** 公式(19)和(20)展示了如何选择一个符合公式(18)条件的新马尔可夫链, 其中  $\tilde{q}(x^{(t+1)}|x^{(t)})$  和  $\tilde{q}(x^{(t)}|x^{(t+1)})$  都与函数  $r$  相关。
3. **修改后的条件分布  $\tilde{p}(x^{(t)}|x^{(t+1)})$ :** 要使  $\tilde{p}(x^{(t+1)}|x^{(t)})$  对应于  $\tilde{q}(x^{(t+1)}|x^{(t)})$ , 公式(21)说明了如何相应地修改  $p(x^{(t+1)}|x^{(t)})$ 。

### 2.5.3. 应用 $r(x^{(t)})$

1. **如果  $r(x^{(t)})$  足够平滑,** 它可以看作是对逆扩散核  $p(x^{(t)}|x^{(t+1)})$  的小扰动, 这种情况下修改后的条件分布可以有与原始条件分布相同的功能形式, 但带有扰动项。
2. **应用闭合形式:** 如果  $r(x^{(t)})$  可以与高斯或二项式分布在闭合形式中相乘, 那么它可以直接乘以分布, 不需要被视为一个扰动。

### 2.5.4. 选择 $r(x^{(t)})$

1. **变化缓慢的  $r(x^{(t)})$ :**  $r(x^{(t)})$  通常应该在轨迹过程中缓慢变化。文本中的实验选择它为常数。
2. **其他选择:** 公式(22)提出了一个替代的选择, 即  $r(x^{(t)})$  是  $r(x^{(0)})$  乘以一个随时间降低的系数, 这保证了从  $\tilde{p}(x^{(T)})$  抽取逆轨迹的初始样本依然是直接的。

Dataset	$K$	$K - L_{null}$
Swiss Roll	2.35 bits	6.45 bits
Binary Heartbeat	-2.414 bits/seq.	12.024 bits/seq.
Bark	-0.55 bits/pixel	1.5 bits/pixel
Dead Leaves	1.489 bits/pixel	3.536 bits/pixel
CIFAR-10	11.895 bits/pixel	18.037 bits/pixel
MNIST	See table 2	

Table 1. The lower bound  $K$  on the log likelihood, computed on a holdout set, for each of the trained models. See Equation 12. The right column is the improvement relative to an isotropic Gaussian or independent binomial distribution.  $L_{null}$  is the log likelihood of  $\pi(\mathbf{x}^{(0)})$ .

applies in the case where  $r(\mathbf{x}^{(t)})$  consists of a delta function for some subset of coordinates, as in the inpainting example in Figure 5.

#### 2.5.4. CHOOSING $r(\mathbf{x}^{(t)})$

Typically,  $r(\mathbf{x}^{(t)})$  should be chosen to change slowly over the course of the trajectory. For the experiments in this paper we chose it to be constant.

$r(\mathbf{x}^{(t)})$  可保持常数

$$r(\mathbf{x}^{(t)}) = r(\mathbf{x}^{(0)}) \quad (22)$$

Another convenient choice is  $r(\mathbf{x}^{(t)}) = r(\mathbf{x}^{(0)})^{\frac{T-t}{T}}$ . Under this second choice  $r(\mathbf{x}^{(t)})$  makes no contribution to the starting distribution for the reverse trajectory. This guarantees that drawing the initial sample from  $\tilde{p}(\mathbf{x}^{(T)})$  for the reverse trajectory remains straightforward.

#### 2.6. Entropy of Reverse Process

Since the forward process is known, it is possible to place upper and lower bounds on the entropy of each step in the reverse trajectory. These bounds can be used to constrain the learned reverse transitions  $p(\mathbf{x}^{(t-1)}|\mathbf{x}^{(t)})$ . The bounds on the conditional entropy of a step in the reverse trajectory are

条件熵: 衡量给定一个状态时, 关于另一个状态. 熵的性质

$$H_q(\mathbf{X}^{(t)}|\mathbf{X}^{(t-1)}) + H_q(\mathbf{X}^{(t-1)}|\mathbf{X}^{(0)}) - H_q(\mathbf{X}^{(t)}|\mathbf{X}^{(0)}) \leq H_q(\mathbf{X}^{(t-1)}|\mathbf{X}^{(t)}) \leq H_q(\mathbf{X}^{(t)}|\mathbf{X}^{(t-1)}), \quad (23)$$

下界

where both the upper and lower bounds depend only on the conditional forward trajectory  $q(\mathbf{x}^{(1..T)}|\mathbf{x}^{(0)})$ , and can be analytically computed. The derivation is provided in Appendix A.

### 3. Experiments

We train diffusion probabilistic models on a variety of continuous datasets, and a binary dataset. We then demonstrate

Model	Log Likelihood
<b>Dead Leaves</b>	
MCGSM	1.244 bits/pixel
<b>Diffusion</b>	<b>1.489 bits/pixel</b>
<b>MNIST</b>	
Stacked CAE	121 ± 1.6 bits
DBN	138 ± 2 bits
Deep GSN	214 ± 1.1 bits
<b>Diffusion</b>	<b>220 ± 1.9 bits</b>
Adversarial net	225 ± 2 bits

Table 2. Log likelihood comparisons to other algorithms. Dead leaves images were evaluated using identical training and test data as in (Theis et al., 2012). MNIST log likelihoods were estimated using the Parzen-window code from (Goodfellow et al., 2014), and show that our performance is comparable to other recent techniques.

sampling from the trained model and inpainting of missing data, and compare model performance against other techniques. In all cases the objective function and gradient were computed using Theano (Bergstra & Breuleux, 2010), and model training was with SFO (Sohl-Dickstein et al., 2014). The lower bound on the log likelihood provided by our model is reported for all datasets in Table 1. A reference implementation of the algorithm utilizing Blocks (van Merriënboer et al., 2015) is available at <https://github.com/Sohl-Dickstein/Diffusion-Probabilistic-Models>.

### 3.1. Toy Problems

#### 3.1.1. SWISS ROLL

A diffusion probabilistic model was built of a two dimensional swiss roll distribution, using a radial basis function network to generate  $\mathbf{f}_\mu(\mathbf{x}^{(t)}, t)$  and  $\mathbf{f}_\Sigma(\mathbf{x}^{(t)}, t)$ . As illustrated in Figure 1, the swiss roll distribution was successfully learned. See Appendix Section D.1.1 for more details.

#### 3.1.2. BINARY HEARTBEAT DISTRIBUTION

A diffusion probabilistic model was trained on simple binary sequences of length 20, where a 1 occurs every 5th time bin, and the remainder of the bins are 0, using a multi-layer perceptron to generate the Bernoulli rates  $\mathbf{f}_b(\mathbf{x}^{(t)}, t)$  of the reverse trajectory. The log likelihood under the true distribution is  $\log_2(\frac{1}{5}) = -2.322$  bits per sequence. As can be seen in Figure 2 and Table 1 learning was nearly perfect. See Appendix Section D.1.2 for more details.

### 3.2. Images

We trained Gaussian diffusion probabilistic models on several image datasets. The multi-scale convolutional archi-

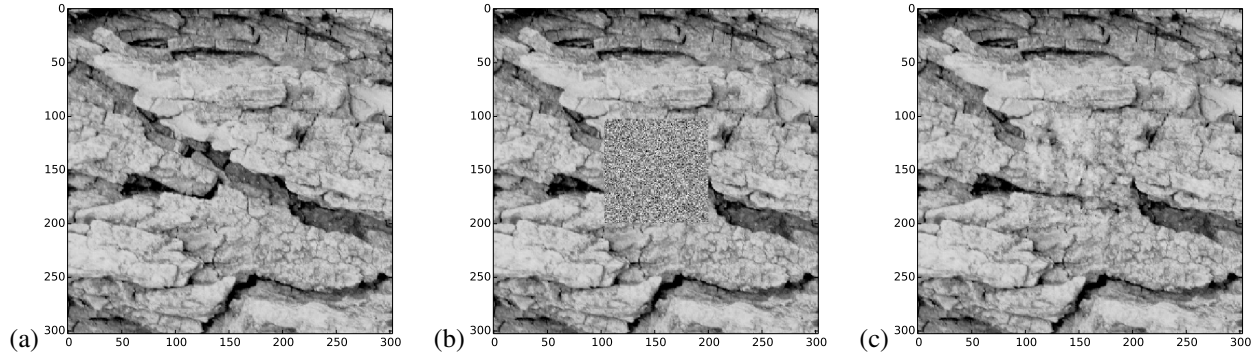


Figure 5. Inpainting. (a) A bark image from (Lazebnik et al., 2005). (b) The same image with the central  $100 \times 100$  pixel region replaced with isotropic Gaussian noise. This is the initialization  $\tilde{p}(\mathbf{x}^{(T)})$  for the reverse trajectory. (c) The central  $100 \times 100$  region has been inpainted using a diffusion probabilistic model trained on images of bark, by sampling from the posterior distribution over the missing region conditioned on the rest of the image. Note the long-range spatial structure, for instance in the crack entering on the left side of the inpainted region. The sample from the posterior was generated as described in Section 2.5, where  $r(\mathbf{x}^{(0)})$  was set to a delta function for known data, and a constant for missing data.

texture shared by these experiments is described in Appendix Section D.2.1, and illustrated in Figure D.1.

### 3.2.1. DATASETS

**MNIST** In order to allow a direct comparison against previous work on a simple dataset, we trained on MNIST digits (LeCun & Cortes, 1998). The relative log likelihoods are given in Table 2 to a variety of techniques (Bengio et al., 2012; Bengio & Thibodeau-Laufer, 2013; Goodfellow et al., 2014). Samples from the MNIST model are given in Figure App.1 in the Appendix. Our training algorithm provides an asymptotically exact lower bound on the log likelihood. However, most previous reported results on MNIST log likelihood rely on Parzen-window based estimates computed from model samples. For this comparison we therefore estimate MNIST log likelihood using the Parzen-window code released with (Goodfellow et al., 2014).

**CIFAR-10** A probabilistic model was fit to the training images for the CIFAR-10 challenge dataset (Krizhevsky & Hinton, 2009). Samples from the trained model are provided in Figure 3.

**Dead Leaf Images** Dead leaf images (Jeulin, 1997; Lee et al., 2001) consist of layered occluding circles, drawn from a power law distribution over scales. They have an analytically tractable structure, but capture many of the statistical complexities of natural images, and therefore provide a compelling test case for natural image models. As illustrated in Table 2 and Figure 4, we achieve state of the art performance on the dead leaves dataset.

**Bark Texture Images** A probabilistic model was trained on bark texture images (T01-T04) from (Lazebnik et al., 2005). For this dataset we demonstrate that it is straightforward to evaluate or generate from a posterior distribution, by inpainting a large region of missing data using a sample from the model posterior in Figure 5.

## 4. Conclusion

We have introduced a novel algorithm for modeling probability distributions that enables exact sampling and evaluation of probabilities and demonstrated its effectiveness on a variety of toy and real datasets, including challenging natural image datasets. For each of these tests we used a similar basic algorithm, showing that our method can accurately model a wide variety of distributions. Most existing density estimation techniques must sacrifice modeling power in order to stay tractable and efficient, and sampling or evaluation are often extremely expensive. The core of our algorithm consists of estimating the reversal of a Markov diffusion chain which maps data to a noise distribution; as the number of steps is made large, the reversal distribution of each diffusion step becomes simple and easy to estimate. The result is an algorithm that can learn a fit to any data distribution, but which remains tractable to train, *exactly* sample from, and evaluate, and under which it is straightforward to manipulate conditional and posterior distributions.

## Acknowledgements

We thank Lucas Theis, Subhaneil Lahiri, Ben Poole, Diederik P. Kingma, Taco Cohen, and Philip Bachman for extremely helpful discussion, and Ian Goodfellow for sharing Parzen-window code. We thank Khan Academy and the Office of Naval Research for funding Jascha Sohl-Dickstein. We further thank the

Office of Naval Research, the Burroughs-Wellcome foundation, Sloan foundation, and James S. McDonnell foundation for funding Surya Ganguli.

## References

- Barron, J. T., Biggin, M. D., Arbelaez, P., Knowles, D. W., Keranen, S. V., and Malik, J. Volumetric Semantic Segmentation Using Pyramid Context Features. In *2013 IEEE International Conference on Computer Vision*, pp. 3448–3455. IEEE, December 2013. ISBN 978-1-4799-2840-8. doi: 10.1109/ICCV.2013.428.
- Bengio, Y. and Thibodeau-Laufer, E. Deep generative stochastic networks trainable by backprop. *arXiv preprint arXiv:1306.1091*, 2013.
- Bengio, Y., Mesnil, G., Dauphin, Y., and Rifai, S. Better Mixing via Deep Representations. *arXiv preprint arXiv:1207.4404*, July 2012.
- Bergstra, J. and Breuleux, O. Theano: a CPU and GPU math expression compiler. *Proceedings of the Python for Scientific Computing Conference (SciPy)*, 2010.
- Besag, J. Statistical Analysis of Non-Lattice Data. *The Statistician*, 24(3), 179-195, 1975.
- Bishop, C., Svensén, M., and Williams, C. GTM: The generative topographic mapping. *Neural computation*, 1998.
- Bornschein, J. and Bengio, Y. Reweighted Wake-Sleep. *International Conference on Learning Representations*, June 2015.
- Burda, Y., Grosse, R. B., and Salakhutdinov, R. Accurate and Conservative Estimates of MRF Log-likelihood using Reverse Annealing. *arXiv:1412.8566*, December 2014.
- Dayan, P., Hinton, G. E., Neal, R. M., and Zemel, R. S. The helmholtz machine. *Neural computation*, 7(5):889–904, 1995.
- Dinh, L., Krueger, D., and Bengio, Y. NICE: Non-linear Independent Components Estimation. *arXiv:1410.8516*, pp. 11, October 2014.
- Feller, W. On the theory of stochastic processes, with particular reference to applications. In *Proceedings of the [First] Berkeley Symposium on Mathematical Statistics and Probability*. The Regents of the University of California, 1949.
- Gershman, S. J. and Blei, D. M. A tutorial on Bayesian nonparametric models. *Journal of Mathematical Psychology*, 56(1): 1–12, 2012.
- Gneiting, T. and Raftery, A. E. Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association*, 102(477):359–378, 2007.
- Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative Adversarial Nets. *Advances in Neural Information Processing Systems*, 2014.
- Gregor, K., Danihelka, I., Mnih, A., Blundell, C., and Wierstra, D. Deep AutoRegressive Networks. *arXiv preprint arXiv:1310.8499*, October 2013.
- Grosse, R. B., Maddison, C. J., and Salakhutdinov, R. Annealing between distributions by averaging moments. In *Advances in Neural Information Processing Systems*, pp. 2769–2777, 2013.
- Hinton, G. E. Training products of experts by minimizing contrastive divergence. *Neural Computation*, 14(8):1771–1800, 2002.
- Hinton, G. E. The wake-sleep algorithm for unsupervised neural networks ). *Science*, 1995.
- Hyvärinen, A. Estimation of non-normalized statistical models using score matching. *Journal of Machine Learning Research*, 6:695–709, 2005.
- Jarzynski, C. Equilibrium free-energy differences from nonequilibrium measurements: A master-equation approach. *Physical Review E*, January 1997.
- Jarzynski, C. Equalities and inequalities: irreversibility and the second law of thermodynamics at the nanoscale. In *Annu. Rev. Condens. Matter Phys.* Springer, 2011.
- Jeulin, D. Dead leaves models: from space tessellation to random functions. *Proc. of the Symposium on the Advances in the Theory and Applications of Random Sets*, 1997.
- Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.
- Kavukcuoglu, K., Ranzato, M., and LeCun, Y. Fast inference in sparse coding algorithms with applications to object recognition. *arXiv preprint arXiv:1010.3467*, 2010.
- Kingma, D. P. and Welling, M. Auto-Encoding Variational Bayes. *International Conference on Learning Representations*, December 2013.
- Krizhevsky, A. and Hinton, G. Learning multiple layers of features from tiny images. *Computer Science Department University of Toronto Tech. Rep.*, 2009.
- Langevin, P. Sur la théorie du mouvement brownien. *CR Acad. Sci. Paris*, 146(530-533), 1908.
- Larochelle, H. and Murray, I. The neural autoregressive distribution estimator. *Journal of Machine Learning Research*, 2011.
- Lazebnik, S., Schmid, C., and Ponce, J. A sparse texture representation using local affine regions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(8):1265–1278, 2005.
- LeCun, Y. and Cortes, C. The MNIST database of handwritten digits. 1998.
- Lee, A., Mumford, D., and Huang, J. Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model. *International Journal of Computer Vision*, 2001.
- Lyu, S. Unifying Non-Maximum Likelihood Learning Objectives with Minimum KL Contraction. In Shawe-Taylor, J., Zemel, R. S., Bartlett, P., Pereira, F. C. N., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 24*, pp. 64–72. 2011.
- MacKay, D. Bayesian neural networks and density networks. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 1995.

- Murphy, K. P., Weiss, Y., and Jordan, M. I. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 467–475. Morgan Kaufmann Publishers Inc., 1999.
- Neal, R. Annealed importance sampling. *Statistics and Computing*, January 2001.
- Ozair, S. and Bengio, Y. Deep Directed Generative Autoencoders. *arXiv:1410.0630*, October 2014.
- Parry, M., Dawid, A. P., Lauritzen, S., and Others. Proper local scoring rules. *The Annals of Statistics*, 40(1):561–592, 2012.
- Rezende, D. J., Mohamed, S., and Wierstra, D. Stochastic Back-propagation and Approximate Inference in Deep Generative Models. *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, January 2014.
- Rippel, O. and Adams, R. P. High-Dimensional Probability Estimation with Deep Density Models. *arXiv:1410.8516*, pp. 12, February 2013.
- Schmidhuber, J. Learning factorial codes by predictability minimization. *Neural Computation*, 1992.
- Sminchisescu, C., Kanaujia, A., and Metaxas, D. Learning joint top-down and bottom-up processes for 3D visual inference. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pp. 1743–1752. IEEE, 2006.
- Sohl-Dickstein, J., Battaglino, P., and DeWeese, M. New Method for Parameter Estimation in Probabilistic Models: Minimum Probability Flow. *Physical Review Letters*, 107(22): 11–14, November 2011a. ISSN 0031-9007. doi: 10.1103/PhysRevLett.107.220601.
- Sohl-Dickstein, J., Battaglino, P. B., and DeWeese, M. R. Minimum Probability Flow Learning. *International Conference on Machine Learning*, 107(22):11–14, November 2011b. ISSN 0031-9007. doi: 10.1103/PhysRevLett.107.220601.
- Sohl-Dickstein, J., Poole, B., and Ganguli, S. Fast large-scale optimization by unifying stochastic gradient and quasi-Newton methods. In *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 604–612, 2014.
- Spinney, R. and Ford, I. Fluctuation Relations : A Pedagogical Overview. *arXiv preprint arXiv:1201.6381*, pp. 3–56, 2013.
- Stuhlmüller, A., Taylor, J., and Goodman, N. Learning stochastic inverses. *Advances in Neural Information Processing Systems*, 2013.
- Suykens, J. and Vandewalle, J. Nonconvex optimization using a Fokker-Planck learning machine. In *12th European Conference on Circuit Theory and Design*, 1995.
- T, P. Convergence condition of the TAP equation for the infinite-ranged Ising spin glass model. *J. Phys. A: Math. Gen.* 15 1971, 1982.
- Tanaka, T. Mean-field theory of Boltzmann machine learning. *Physical Review Letters E*, January 1998.
- Theis, L., Hosseini, R., and Bethge, M. Mixtures of conditional Gaussian scale mixtures applied to multiscale image representations. *PLoS one*, 7(7):e39857, 2012.
- Uria, B., Murray, I., and Larochelle, H. RNADE: The real-valued neural autoregressive density-estimator. *Advances in Neural Information Processing Systems*, 2013a.
- Uria, B., Murray, I., and Larochelle, H. A Deep and Tractable Density Estimator. *arXiv:1310.1757*, pp. 9, October 2013b.
- van Merriënboer, B., Chorowski, J., Serdyuk, D., Bengio, Y., Bogdanov, D., Dumoulin, V., and Warde-Farley, D. Blocks and Fuel. *Zenodo*, May 2015. doi: 10.5281/zenodo.17721.
- Welling, M. and Hinton, G. A new learning algorithm for mean field Boltzmann machines. *Lecture Notes in Computer Science*, January 2002.
- Yao, L., Ozair, S., Cho, K., and Bengio, Y. On the Equivalence Between Deep NADE and Generative Stochastic Networks. In *Machine Learning and Knowledge Discovery in Databases*, pp. 322–336. Springer, 2014.