NSDE 1: Stochastic Dynamics

CS 598: Deep Generative and Dynamical Models

Instructor: Arindam Banerjee

November 9, 2021

Latent Variable Models Recap

- Joint distribution of a latent variable model (LVM)

$$p_\theta(x, z) = p_\theta(z)p_\theta(x|z) \ ,$$

  - x denotes the observed variable
  - z denotes the latent variable
  - θ denotes the parameters
- Problems of interest
  - Compute marginal or conditional distributions

$$p_\theta(x) = \int_z p_\theta(x, z)dz \qquad p_\theta(z|x) = \frac{p_\theta(x, z)}{p_\theta(x)}$$

  - Estimate θ by optimizing a function of $p_\theta(x)$
- Problems need to compute high-d integrals

Variational Inference

- Construct a distribution $q_\phi(z|x)$ with parameters ϕ
- Choose family q and parameters ϕ to approximate true posterior

$$q_\phi(z|x) \approx p_\theta(z|x)$$

- For any $q_\phi(z|x)$

$$\log p_\theta(x) \geq \mathbb{E}_{q_\phi(z|x)}[\log p_\theta(x,z) - \log q_\phi(z|x)] = \mathcal{L}$$

- Goal: Choosing a more flexible $q_\phi(z|x)$

MCMC and Auxiliary Variables

- MCMC uses a stochastic transition operator: $z_t \sim q(z_t|z_{t-1}, x)$
- Main idea: Variational distribution q based on T steps of MCMC

$$q(z_{0:T}|x) = q(z_0|x) \prod_{t=1}^{T} q(z_t|z_{t-1}, x)$$

- Intermediate auxiliary r.v.s: $\mathrm{y} = z_{0:T-1}$
- Variational lower bound

$$\begin{aligned}
\mathcal{L}_{\text{aux}} &= \mathbb{E}_{q(y,z_T|x)}[\log(p(x, z_T)r(y|x, z_T)) - \log q(y, z_T|x)] \\
&= \mathcal{L} - \mathbb{E}_{q(\mathrm{z}_T|\mathrm{x})}[KL[q(\mathrm{y}|\mathrm{z}_t, \mathrm{x})\|r(\mathrm{y}|\mathrm{z}_T, \mathrm{x})] \\
&\leq \mathcal{L} \leq \log p(\mathrm{x})
\end{aligned}$$

- Here $\mathcal{L}$ corresponds to ELBO for $q(\mathrm{z}_T|\mathrm{x})$

MCMC and Auxiliary Variables (Contd.)

- Optimal choice $r(y|x, z_T) = q(y|x, z_T)$, maybe intractable
- Approximate using distribution with Markov structure

$$r(z_0, \ldots, z_{T-1}|x, z_T) = \prod_{t=1}^{T} r_t(z_{t-1}|x, z_t)$$

- Variational lower bound

$$\log p(x) \geq \mathbb{E}_q[\log p(x, z_T) + \log r(z_0, \ldots, z_{T-1}) - \log q(z_0, \ldots, z_T|x)]$$

$$= \mathbb{E}_q \left[\log \frac{p(x, z_T)}{q(z_0|x)} + \sum_{t=1}^{T} \log \frac{r_t(z_{t-1}|x, z_t)}{q_t(z_t|x, z_{t-1})} \right]$$

- Index $t \Rightarrow$ transition q_t, inverse r_t may change with t

MCMC Lower Bound Estimate

Algorithm 1 MCMC lower bound estimate

Require: Model with joint distribution $p(x, z)$ and a desired but intractable posterior $p(z|x)$
Require: Number of iterations T
Require: Transition operator(s) $q_t(z_t|x, z_{t-1})$
Require: Inverse model(s) $r_t(z_{t-1}|x, z_t)$
  Draw an initial random variable $z_0 \sim q(z_0|x)$
  Initialize the lower bound estimate as
  $L = \log p(x, z_0) - \log q(z_0|x)$
  for $t = 1 : T$ **do**
    Perform random transition $z_t \sim q_t(z_t|x, z_{t-1})$
    Calculate the ratio $\alpha_t = \frac{p(x,z_t)r_t(z_{t-1}|x,z_t)}{p(x,z_{t-1})q_t(z_t|x,z_{t-1})}$
    Update the lower bound $L = L + \log[\alpha_t]$
  end for
  return the unbiased lower bound estimate L

Optimizing the Lower Bound

Algorithm 2 Markov Chain Variational Inference (MCVI)

Require: Forward Markov model $q_\theta(z)$ and backward Markov model $r_\theta(z_0, \ldots, z_{t-1}|z_T)$

Require: Parameters θ

Require: Stochastic estimate $L(\theta)$ of the variational lower bound $\mathcal{L}_{\text{aux}}(\theta)$ from Algorithm 1

while not converged **do**

Obtain unbiased stochastic estimate $\hat{g}$ with $E_q[\hat{g}] = \nabla_\theta \mathcal{L}_{\text{aux}}(\theta)$ by differentiating $L(\theta)$

Update the parameters θ using gradient $\hat{g}$ and a stochastic optimization algorithm

end while

return final optimized variational parameters θ

Example: Bi-variate Gaussians

- Bi-variate Gaussian model

$$p(z_1, z_2) \propto \exp\left[-\frac{(z_1 - z_2)^2}{2\sigma_1^2} - \frac{(z_1 + z_2)^2}{2\sigma_2^2}\right]$$

- Two approaches for MCMC
  - Gibbs sampling based on $q(z_i|z_{-i}) = \mathcal{N}(\mu_i, \sigma_i^2)$
  - Over-relaxation based on

$$q(z_{i,t}|z_{t-1}) = \mathcal{N}(\mu_i + \alpha(z_{i,t-1} - \mu_i), \sigma_i^2(1 - \alpha^2))$$

  - Equivalent for $\alpha = 0$
  - General α may lead to better mixing

Example: Bi-variate Gaussians

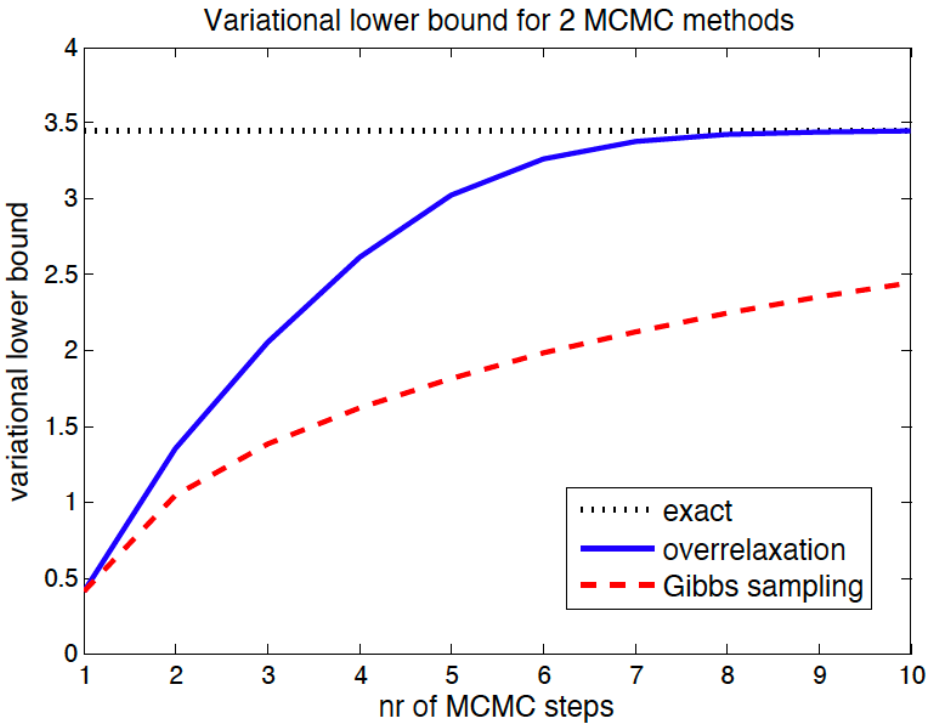

Figure 1. The log marginal likelihood lower bound for a bivariate Gaussian target and an MCMC variational approximaton, using Gibbs sampling or Adler's overrelaxation.

Hamiltonian Dynamics

- State of the system q, momentum p ($= mv$, mass $\times$ velocity)
  - Potential energy $U(q)$: height of surface, negative log-likelihood
  - Kinetic energy $K(p) = \frac{1}{2}mv^2 = \frac{p^2}{2m}$
- Dynamics at a high level (without friction, $K + U$ is conserved)
  - Flat surface, move with constant velocity
  - Upwards slope: K decreases, slows down, stop and slide back down
  - Downward slope: K increases, speeds up, reach valley, overshoot
- System is described by the Hamiltonian $H(q, p)$
- Hamilton's equations characterize change in $q_i, p_i, i = 1, \ldots, d$

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i} \qquad \frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}$$

- With $z = (q, p) \in \mathbb{R}^{2d}$, Hamilton's equation is

$$\frac{dz}{dt} = J\nabla H(z)\ , \qquad J = \begin{bmatrix} 0_{d\times d} & I_{d\times d} \\ -I_{d\times d} & 0_{d\times d} \end{bmatrix}$$

Hamiltonian Monte Carlo

- Hamiltonian as
$$H(q,p) = U(q) + K(p) \ , \qquad K(p) = p^T M^{-1} p/2$$
- M is typically diagonal, or scaled identity
- Reversible: Mapping $T_s : (q(t), p(t)) \mapsto (q(t+s), p(t+s))$ is one-one, onto
  - Has a well defined inverse T_{-s}
- Conservation: Hamiltonian is invariant over time, $\frac{dH}{dt} = 0$
- Volume preservation: Known as Liouville's Theorem
  - T_s applied to some region R of (q,p) space with volume V
  - The image of R from T_s will have the same volume

Hamiltonian Monte Carlo: Leapfrog Method

- Hamilton's equations characterize change in $q_i, p_i, i = 1, \ldots, d$

$$\frac{dq_i}{dt} = \frac{\partial H}{\partial p_i} \qquad \frac{dp_i}{dt} = -\frac{\partial H}{\partial q_i}$$

- Hamiltonian as

$$H(q,p) = U(q) + K(p)\ , \qquad K(p) = p^T M^{-1} p/2$$

- Leapfron updates

$$\begin{aligned} p_i(t+\varepsilon/2) &= p_i(t) - (\varepsilon/2)\frac{\partial U}{\partial q_i}(q(t)) \\ q_i(t+\varepsilon) &= q_i(t) + \varepsilon \frac{p_i(t+\varepsilon/2)}{m_i} \\ p_i(t+\varepsilon) &= p_i(t+\varepsilon/2) - (\varepsilon/2)\frac{\partial U}{\partial q_i}(q(t+\varepsilon)) \end{aligned}$$

Hamiltonian Variational Inference

- HMC approximates $p(\mathrm{z}|\mathrm{x})$ by expanding the space to include v
- Auxiliary variables v (momentum) with $v_t' \sim q(v_t'|\mathrm{x}, z_{t-1})$
- Simulate dynamics by iterative updates based on leapfrog on the Hamiltonian

$$H(v, z) = \frac{1}{2} v^T M^{-1} v - \log p(\mathrm{x}, \mathrm{z})$$

- Related respectively to the kinetic and potential energies
- Dynamics is guided by gradient of exact log-posterior
  - Approximation automatically adapts to local shape of true posterior
- Tradeoffs
  - Better quality approximation, lower variance in SGD estimates
  - Cost per iteration is higher: m MCMC steps, k leapfrog for each

Algorithm: Hamiltonian Variational Inference

Algorithm 3 Hamiltonian variational inference (HVI)

Require: Unnormalized log posterior $\log p(x, z)$
Require: Number of iterations T
Require: Momentum initialization distribution(s) $q_t(v'_t|z_{t-1}, x)$ and inverse model(s) $r_t(v_t|z_t, x)$
Require: HMC stepsize and mass matrix ϵ, M
  Draw an initial random variable $z_0 \sim q(z_0|x)$
  Init. lower bound $L = \log[p(x, z_0)] - \log[q(z_0|x)]$
  for $t = 1 : T$ **do**
    Draw initial momentum $v'_t \sim q_t(v'_t|x, z_{t-1})$
    Set $z_t, v_t = \text{Hamiltonian\_Dynamics}(z_{t-1}, v'_t)$
    Calculate the ratio $\alpha_t = \frac{p(x,z_t)r_t(v_t|x,z_t)}{p(x,z_{t-1})q_t(v'_t|x,z_{t-1})}$
    Update the lower bound $L = L + \log[\alpha_t]$
  end for
  return lower bound L, approx. posterior draw z_T

Example: Beta-Binomial Model

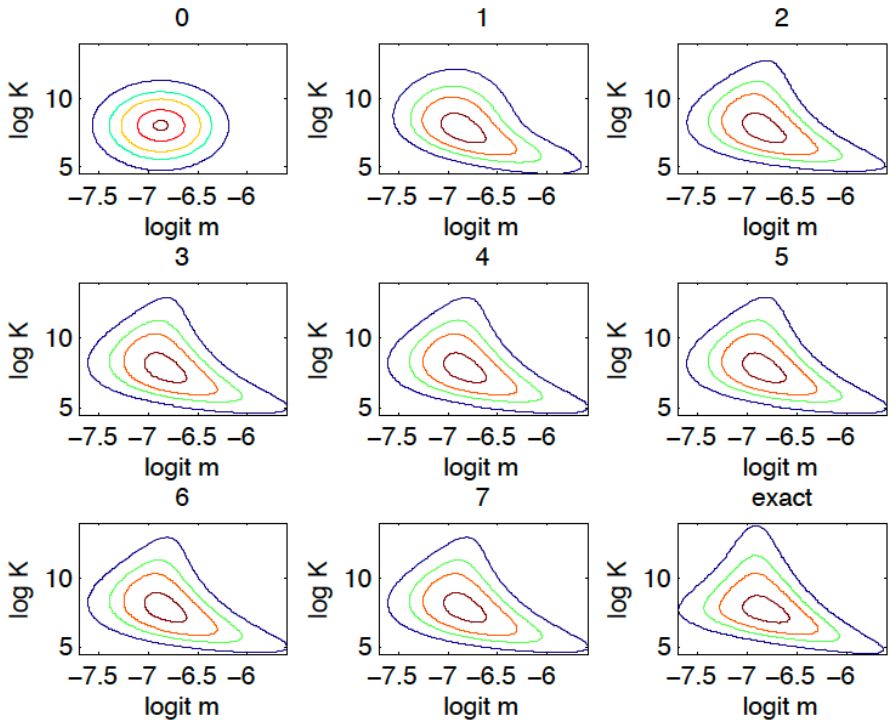

Figure 2. Approximate posteriors for a varying number of leapfrog steps. Exact posterior at bottom right.

Example: Beta-Binomial Model

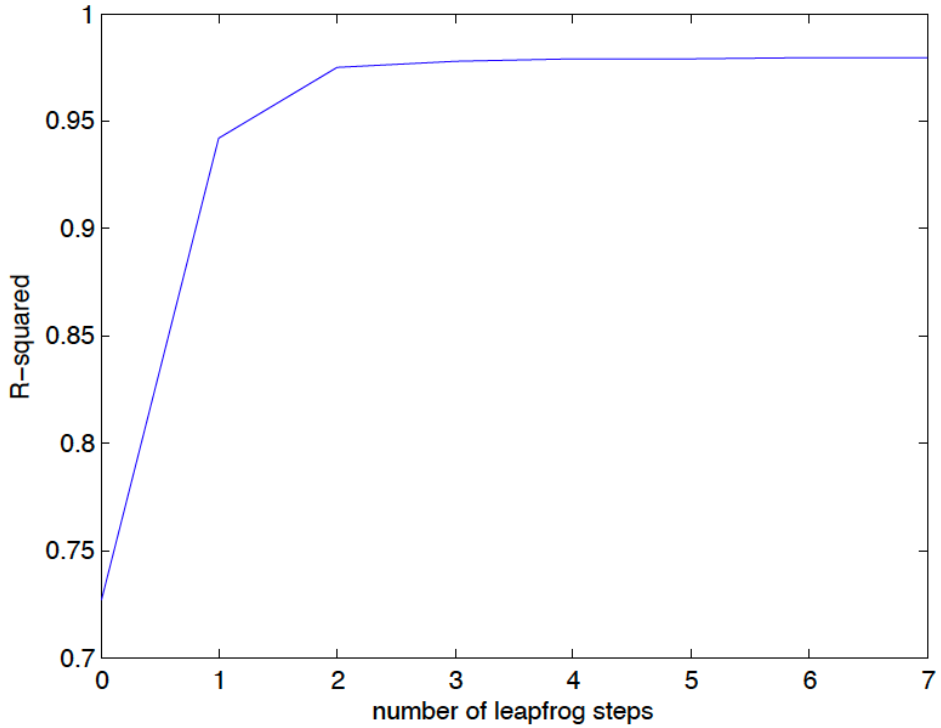

Figure 3. R-squared accuracy measure (Salimans & Knowles, 2013) for approximate posteriors using a varying number of leapfrog steps.

Example: Hand-written Digits

Table 1. Comparison of our approach to other recent methods in the literature. We compare the average marginal log-likelihood measured in nats of the digits in the MNIST test set. See section 3.2 for details.

Model	$\log p(x) \geq -$	$\log p(x) = -$
HVI + fully-connected VAE:		
Without inference network:		
5 leapfrog steps	90.86	87.16
10 leapfrog steps	87.60	85.56
With inference network:		
No leapfrog steps	94.18	88.95
1 leapfrog step	91.70	88.08
4 leapfrog steps	89.82	86.40
8 leapfrog steps	88.30	85.51
HVI + convolutional VAE:		
No leapfrog steps	86.66	83.20
1 leapfrog step	85.40	82.98
2 leapfrog steps	85.17	82.96
4 leapfrog steps	84.94	82.78
8 leapfrog steps	84.81	82.72
16 leapfrog steps	84.11	82.22
16 leapfrog steps, $n_h = 800$	83.49	81.94
From (Gregor et al., 2015):		
DBN 2hl		84.55
EoNADE		85.10
DARN 1hl	88.30	84.13
DARN 12hl	87.72	
DRAW	80.97	

Probabilistic Models based on Diffusions

- Markov chain to convert one distribution to another
  - Used in non-equilibrium statistical physics
  - Related to sequential Monte Carlo, annealed importance sampling
  - Related to Langevin dynamics with target distribution
- Forward trajectory: Data distribution to Gaussians, by Diffusions
- Generative model is the reverse trajectory
- Forward (inference) process is restricted to simple form
  - Ensures the reverse process will have the same functional form
- Models with thousands of layers (time steps)

Gaussian Diffusion: Forward and Reverse Trajectories

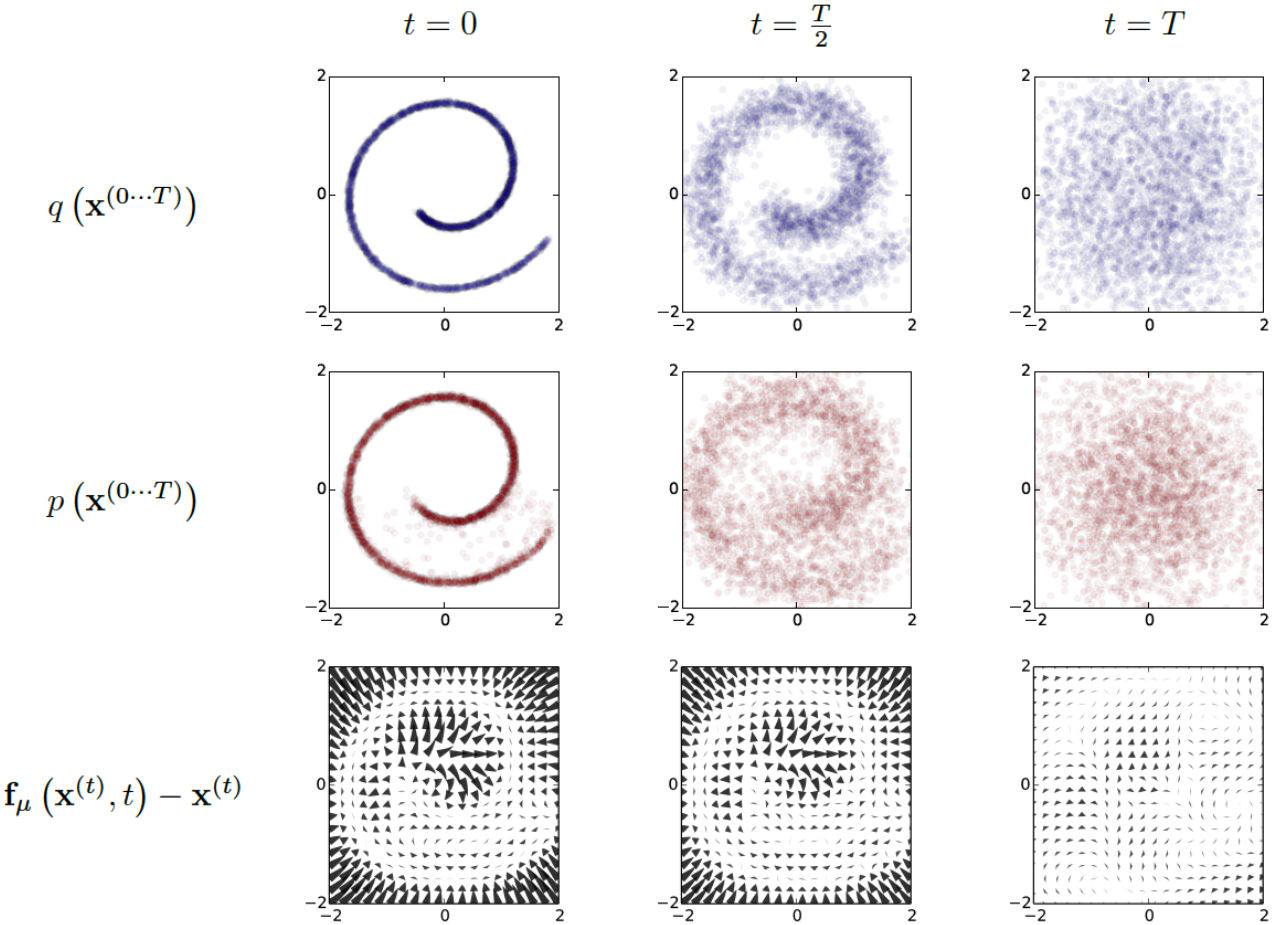

Figure 1. The proposed modeling framework trained on 2-d swiss roll data. The top row shows time slices from the forward trajectory $q\left(\mathbf{x}^{(0\cdots T)}\right)$. The data distribution (left) undergoes Gaussian diffusion, which gradually transforms it into an identity-covariance Gaussian (right).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

Binomial Diffusion

$p\left(\mathbf{x}^{(0\cdots T)}\right)$

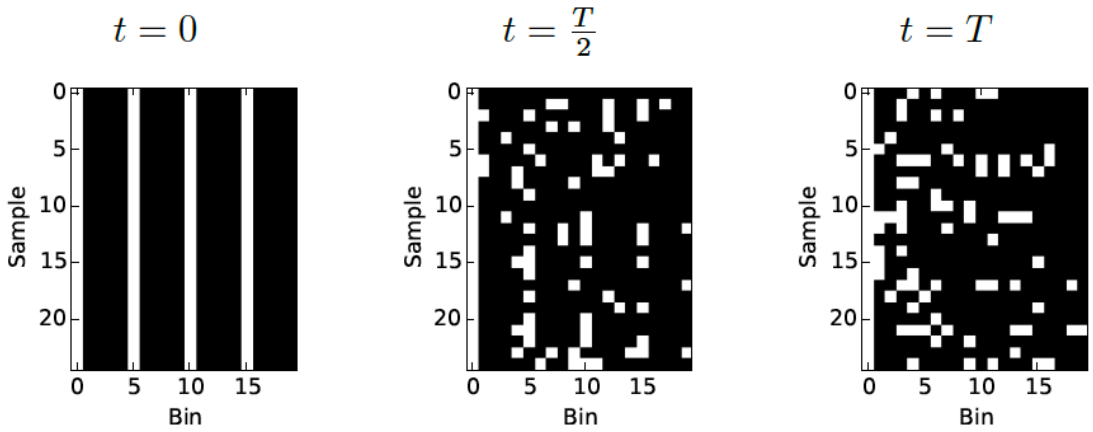

Figure 2. Binary sequence learning via binomial diffusion. A binomial diffusion model was trained on binary 'heartbeat' data, where a pulse occurs every 5th bin. Generated samples (left) are identical to the training data. The sampling procedure consists of initialization at independent binomial noise (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.

Samples generated on CIFAR-10

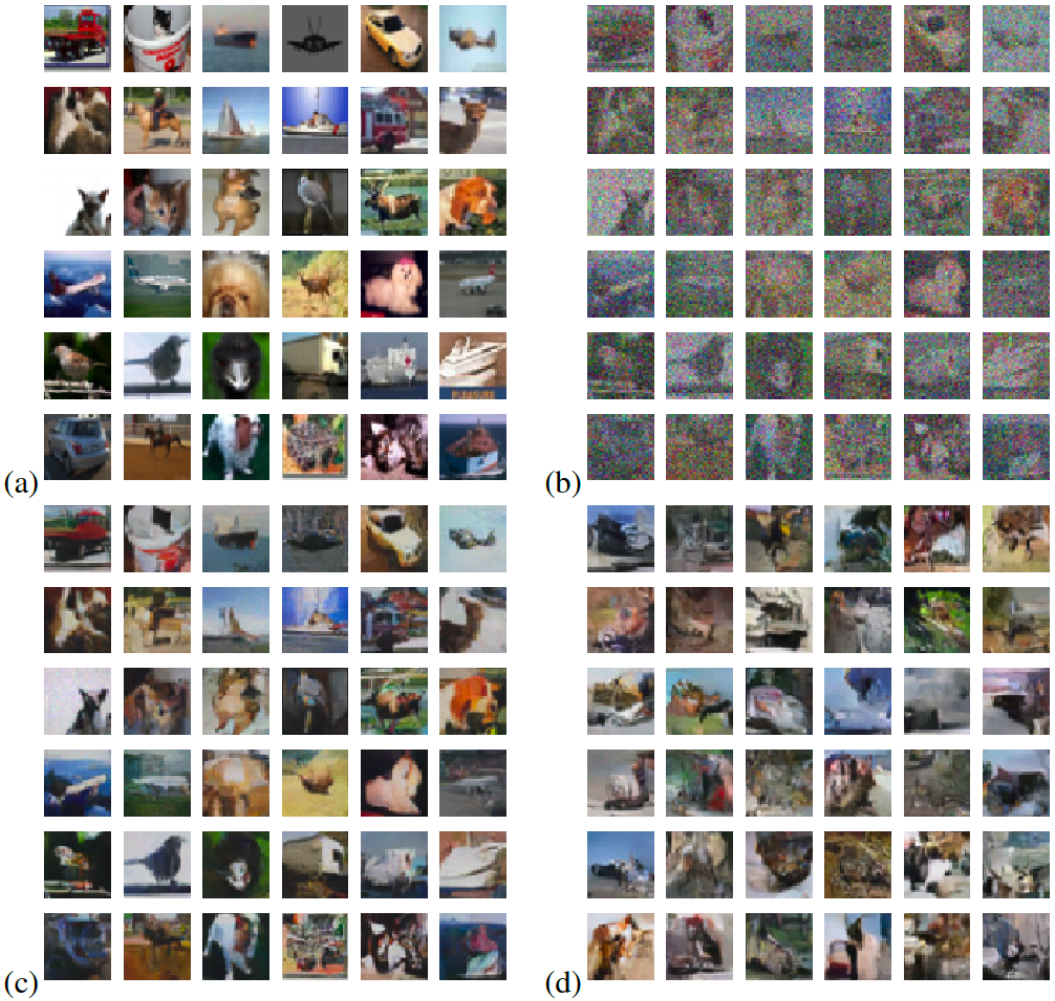

Figure 3. The proposed framework trained on the CIFAR-10 (Krizhevsky & Hinton, 2009) dataset. *(a)* Example holdout data (similar to training data). *(b)* Holdout data corrupted with Gaussian noise of variance 1 (SNR = 1). *(c)* Denoised images, generated by sampling from the posterior distribution over denoised images conditioned on the images in (b). *(d)* Samples generated by the diffusion model.

Forward Trajectory

- Data distribution is $q(\mathrm{x}^{(0)})$
- Target distribution is some well behaved distribution $\pi(\mathrm{y})$
- Forward trajectory goes from $q(\mathrm{x}^{(0)})$ to $\pi(\mathrm{y})$
  - Repeatedly apply Markov diffusion kernel $T_\pi(\mathrm{y}|\mathrm{y}';\beta)$
  - β is the diffusion rate
- Forward trajectory

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

$$q(\mathrm{x}^{(0\cdots T)}) = q(\mathrm{x}^{(0)}) \prod_{t=1}^{T} q(\mathrm{x}^{(t)}|\mathrm{x}^{(t-1)})$$

- Target distribution

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

Reverse Trajectory

- Generative model goes in reverse to data distribution

$$p(\mathrm{x}^{(T)}) = \pi(\mathrm{x}^{(T)})$$

$$p(\mathrm{x}^{(0\cdots T)}) = p(\mathrm{x}^{(T)}) \prod_{t=1}^{T} p(\mathrm{x}^{(t-1)}|\mathrm{x}^{(t)})$$

- For continuous diffusion, reverse process has the functional form
  - $q(\mathrm{x}^{(t)}|\mathrm{x}^{(t-1)})$ and $q(\mathrm{x}^{(t-1)}|\mathrm{x}^{(t)})$ are both Gaussians (or binomial)
  - Work with small diffusion rate β_t
- Need to learn mean and covariance of Gaussian diffusion kernel
  - $f_\mu(\mathrm{x}^{(t)}, t), f_\Sigma(\mathrm{x}^{(t)}, t)$ are modeled as deep nets

Gaussian and Binomial Diffusions

		Gaussian	*Binomial*
Well behaved (analytically tractable) distribution	$\pi\left(\mathbf{x}^{(T)}\right) =$	$\mathcal{N}\left(\mathbf{x}^{(T)}; \mathbf{0}, \mathbf{I}\right)$	$\mathcal{B}\left(\mathbf{x}^{(T)}; 0.5\right)$
Forward diffusion kernel	$q\left(\mathbf{x}^{(t)}\vert\mathbf{x}^{(t-1)}\right) =$	$\mathcal{N}\left(\mathbf{x}^{(t)}; \mathbf{x}^{(t-1)}\sqrt{1-\beta_t}, \mathbf{I}\beta_t\right)$	$\mathcal{B}\left(\mathbf{x}^{(t)}; \mathbf{x}^{(t-1)}\left(1-\beta_t\right) + 0.5\beta_t\right)$
Reverse diffusion kernel	$p\left(\mathbf{x}^{(t-1)}\vert\mathbf{x}^{(t)}\right) =$	$\mathcal{N}\left(\mathbf{x}^{(t-1)}; \mathbf{f}_\mu\left(\mathbf{x}^{(t)}, t\right), \mathbf{f}_\Sigma\left(\mathbf{x}^{(t)}, t\right)\right)$	$\mathcal{B}\left(\mathbf{x}^{(t-1)}; \mathbf{f}_b\left(\mathbf{x}^{(t)}, t\right)\right)$
Training targets		$\mathbf{f}_\mu\left(\mathbf{x}^{(t)}, t\right), \mathbf{f}_\Sigma\left(\mathbf{x}^{(t)}, t\right), \beta_{1\cdots T}$	$\mathbf{f}_b\left(\mathbf{x}^{(t)}, t\right)$
Forward distribution	$q\left(\mathbf{x}^{(0\cdots T)}\right) =$	$q\left(\mathbf{x}^{(0)}\right)\prod_{t=1}^{T} q\left(\mathbf{x}^{(t)}\vert\mathbf{x}^{(t-1)}\right)$	
Reverse distribution	$p\left(\mathbf{x}^{(0\cdots T)}\right) =$	$\pi\left(\mathbf{x}^{(T)}\right)\prod_{t=1}^{T} p\left(\mathbf{x}^{(t-1)}\vert\mathbf{x}^{(t)}\right)$	
Log likelihood	$L =$	$\int d\mathbf{x}^{(0)} q\left(\mathbf{x}^{(0)}\right)\log p\left(\mathbf{x}^{(0)}\right)$	
Lower bound on log likelihood	$K =$	$-\sum_{t=2}^{T}\mathbb{E}_{q\left(\mathbf{x}^{(0)},\mathbf{x}^{(t)}\right)}\left[D_{KL}\left(q\left(\mathbf{x}^{(t-1)}\vert\mathbf{x}^{(t)},\mathbf{x}^{(0)}\right)\Vert p\left(\mathbf{x}^{(t-1)}\vert\mathbf{x}^{(t)}\right)\right)\right] + H_q\left(\mathbf{X}^{(T)}\vert\mathbf{X}^{(0)}\right) - H_q\left(\mathbf{X}^{(1)}\vert\mathbf{X}^{(0)}\right) - H_p\left(\mathbf{X}^{(T)}\right)$	
Perturbed reverse diffusion kernel	$\tilde{p}\left(\mathbf{x}^{(t-1)}\vert\mathbf{x}^{(t)}\right) =$	$\mathcal{N}\left(x^{(t-1)}; \mathbf{f}_\mu\left(\mathbf{x}^{(t)}, t\right) + \mathbf{f}_\Sigma\left(\mathbf{x}^{(t)}, t\right)\frac{\partial \log r\left(\mathbf{x}^{(t-1)'}\right)}{\partial \mathbf{x}^{(t-1)'}}\Big\vert_{\mathbf{x}^{(t-1)'}=f_\mu\left(\mathbf{x}^{(t)},t\right)}, \mathbf{f}_\Sigma\left(\mathbf{x}^{(t)}, t\right)\right)$	$\mathcal{B}\left(x_i^{(t-1)}; \frac{c_i^{t-1}d_i^{t-1}}{x_i^{t-1}d_i^{t-1}+(1-c_i^{t-1})(1-d_i^{t-1})}\right)$

Table App.1. The key equations in this paper for the specific cases of Gaussian and binomial diffusion processes. $\mathcal{N}\left(u; \mu, \Sigma\right)$ is a Gaussian distribution with mean μ and covariance Σ. $\mathcal{B}\left(u; r\right)$ is the distribution for a single Bernoulli trial, with $u = 1$ occurring with probability r, and $u = 0$ occurring with probability $1 - r$. Finally, for the perturbed Bernoulli trials $b_i^t = \mathbf{x}^{(t-1)}\left(1-\beta_t\right) + 0.5\beta_t$, $c_i^t = \left[\mathbf{f}_b\left(\mathbf{x}^{(t+1)}, t\right)\right]_i$, and $d_i^t = r\left(x_i^{(t)} = 1\right)$, and the distribution is given for a single bit i.

Model Probability

- Exact computation of probability can be intractable

$$p(\mathrm{x}^{(0)}) = \int p(x^{(0\ldots T)}) d\mathrm{x}^{(1\ldots T)}$$

- Instead can be computed based on the forward trajectory

$$p(\mathrm{x}^{(0)}) = \int p(\mathrm{x}^{(T)}) \left(\prod_{t=1}^{T} \frac{p(\mathrm{x}^{(t-1)}|\mathrm{x}^{(t)})}{q(\mathrm{x}^{(t)}|\mathrm{x}^{(t-1)})} \right) q(x^{(1\ldots T)}) d\mathrm{x}^{(1\ldots T)}$$

- If forward and backward distributions are the same
  - Need one sample, because of cancellations
  - Diffusion rate $\beta \to 0$

Training

- Training is based on maximizing log-likelihood w.r.t. $p(\mathrm{x}^{(t-1)}|\mathrm{x}^{(t)})$

$$
\begin{aligned}
L &= \int \log p(x^{(0)}) q(x^{(0)}) dx^{(0)} \\
&= \int \log \left[p(\mathrm{x}^{(T)}) \left(\prod_{t=1}^{T} \frac{p(\mathrm{x}^{(t-1)}|\mathrm{x}^{(t)})}{q(\mathrm{x}^{(t)}|\mathrm{x}^{(t-1)})} \right) q(x^{(1\dots T)}) \right] q(x^{(0)}) dx^{(0)} \\
&\geq \int \log \left[p(\mathrm{x}^{(T)}) \left(\prod_{t=1}^{T} \frac{p(\mathrm{x}^{(t-1)}|\mathrm{x}^{(t)})}{q(\mathrm{x}^{(t)}|\mathrm{x}^{(t-1)})} \right) \right] q(x^{(0\dots T)}) dx^{(0\cdots T)}
\end{aligned}
$$

- The lower bound (say, K) can be written as

$$
\begin{aligned}
K = -\sum_{t=2}^{T} \int KL \left(q(\mathrm{x}^{(t-1)}|\mathrm{x}^{(t)}, \mathrm{x}^{(0)}) \| p(\mathrm{x}^{(t-1)}|\mathrm{x}^{(t)}) \right) q(\mathrm{x}^{(0)}, \mathrm{x}^{(t)}) dx^{(0)} dx^{(t)} \\
+ H_q(X^{(T)}|X^{(0)}) - H_q(X^{(1)}|X^{(0)}) - H_p(X^{(T)})
\end{aligned}
$$

Computing Posteriors

- Involves multiplying $p(\mathrm{x}^{(0)})$ with positive $r(\mathrm{x}^{(0)})$
- Modified trajectory: $\tilde{p}(\mathrm{x}^{(t)}) = \frac{1}{\tilde{Z}_t} p(\mathrm{x}^{(t)}) r(\mathrm{x}^{(t)})$
- Want the perturbed Markov kernel to satisfy

$$\tilde{p}(\mathrm{x}^{(t)}) = \int \tilde{p}(\mathrm{x}^{(t)}|\mathrm{x}^{(t+1)}) \tilde{p}(\mathrm{x}^{(t+1)}) d\mathrm{x}^{(t+1)}$$

- Can be satisfied with the reverse transition

$$\tilde{p}(\mathrm{x}^{(t)}|\mathrm{x}^{(t+1)}) = \frac{1}{\tilde{Z}_t(\mathrm{x}^{(t+1)})} p(\mathrm{x}^{(t)}) r(\mathrm{x}^{(t)})$$

- For smooth $r(\mathrm{x}^{(t)})$, small perturbation to the original $p(\mathrm{x}^{(t)}|\mathrm{x}^{(t+1)})$
- Over time t, $r(\mathrm{x}^{(t)})$ should be slowly varying

Results: Log-likelihood

Model	*Log Likelihood*
Dead Leaves	
MCGSM	1.244 bits/pixel
Diffusion	**1.489 bits/pixel**
MNIST	
Stacked CAE	174 ± 2.3 bits
DBN	199 ± 2.9 bits
Deep GSN	309 ± 1.6 bits
Diffusion	$\mathbf{317 \pm 2.7}$ **bits**
Adversarial net	325 ± 2.9 bits
Perfect model	349 ± 3.3 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), with values given in bits, and show that our performance is comparable to other recent techniques. The perfect model entry was computed by applying the Parzen code to samples from the training data.

Results: Samples

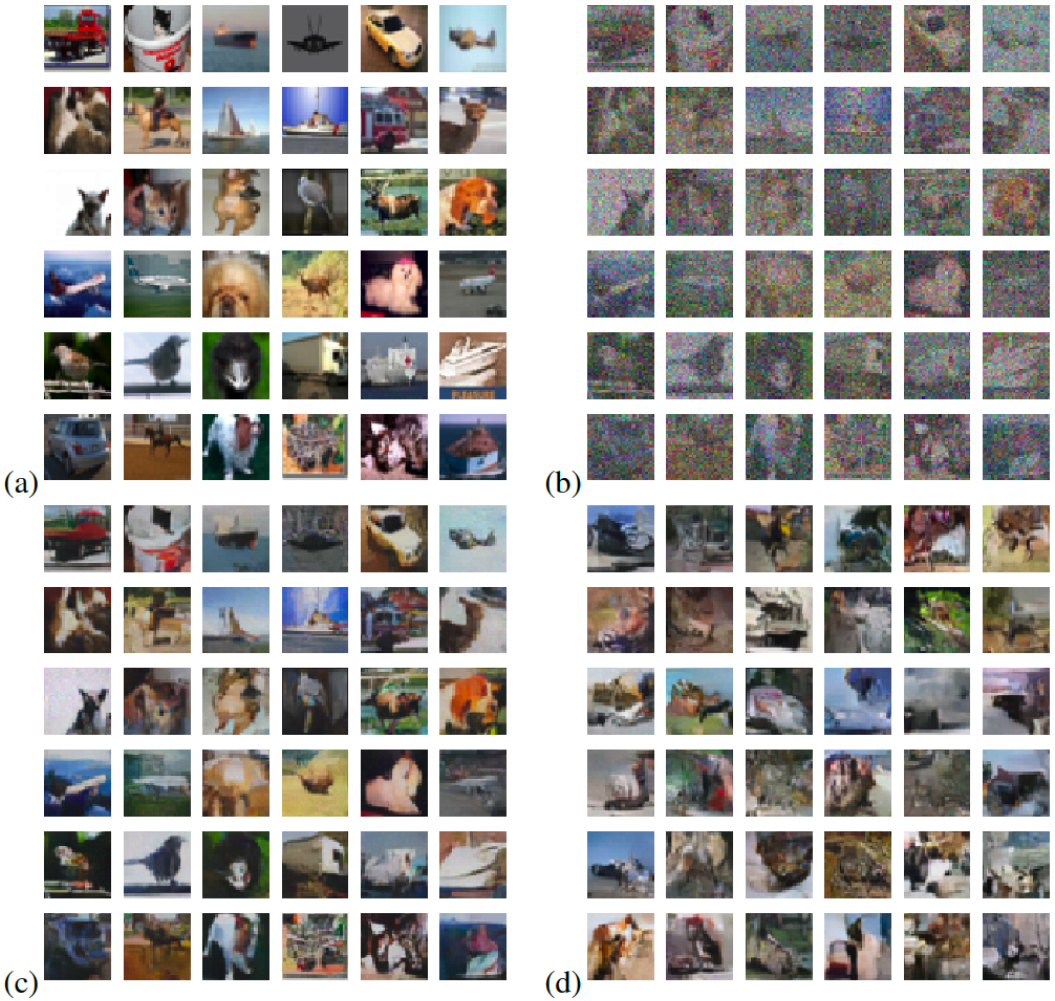

Figure 3. The proposed framework trained on the CIFAR-10 (Krizhevsky & Hinton, 2009) dataset. *(a)* Example holdout data (similar to training data). *(b)* Holdout data corrupted with Gaussian noise of variance 1 (SNR = 1). *(c)* Denoised images, generated by sampling from the posterior distribution over denoised images conditioned on the images in (b). *(d)* Samples generated by the diffusion model.

Results: Dead Leaf Images

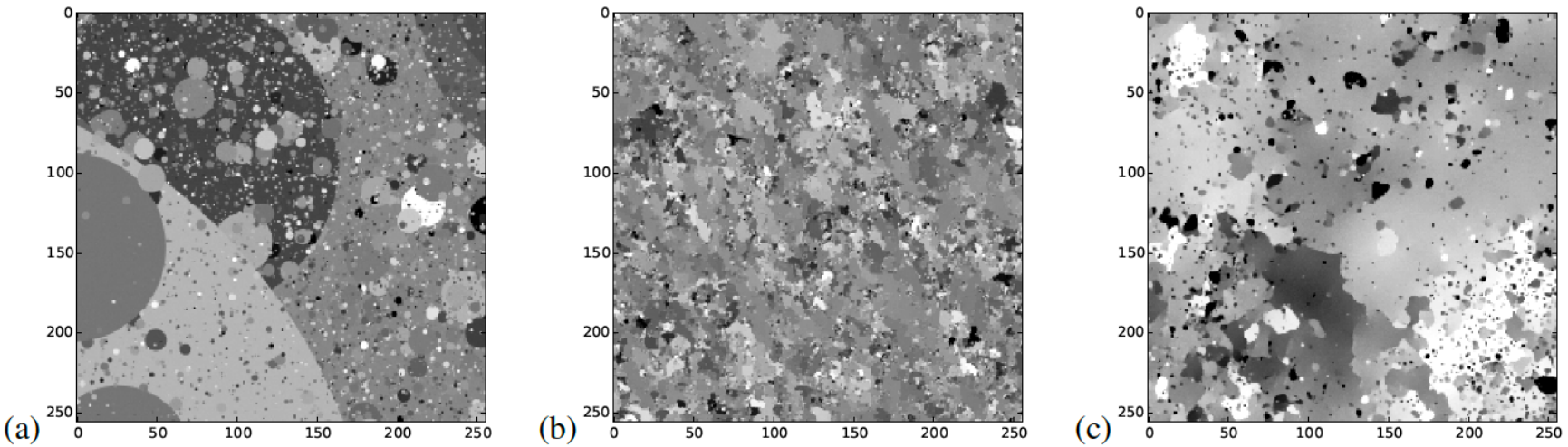

Figure 4. The proposed framework trained on dead leaf images (Jeulin, 1997; Lee et al., 2001). *(a)* Example training image. *(b)* A sample from the previous state of the art natural image model (Theis et al., 2012) trained on identical data, reproduced here with permission. *(c)* A sample generated 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.

Results: Inpainting

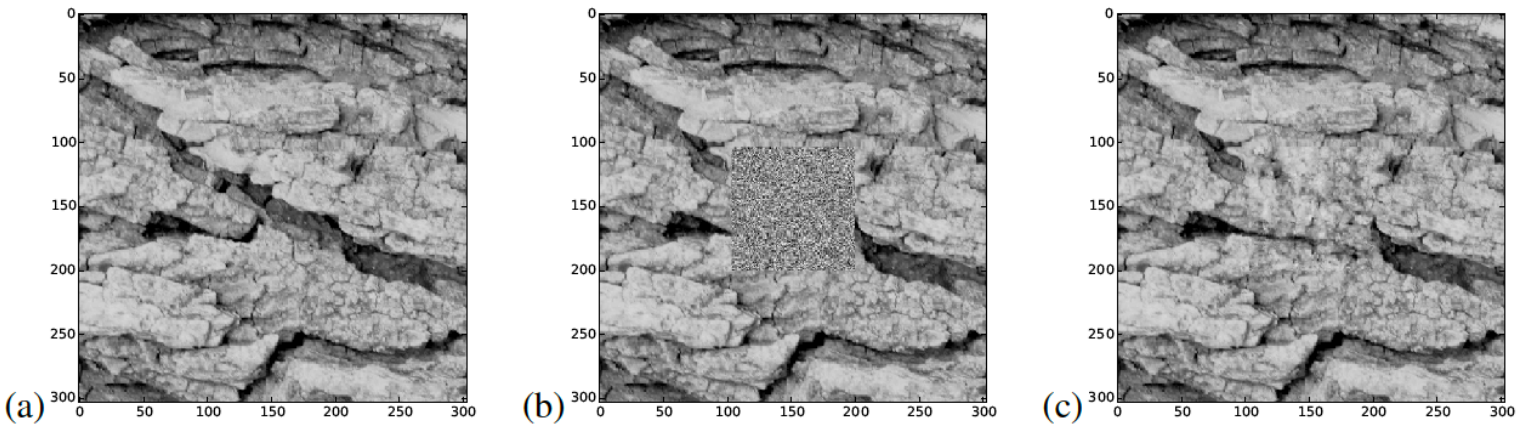

Figure 5. Inpainting. *(a)* A bark image from (Lazebnik et al., 2005). *(b)* The same image with the central 100×100 pixel region replaced with isotropic Gaussian noise. This is the initialization $\tilde{p}\left(\mathbf{x}^{(T)}\right)$ for the reverse trajectory. *(c)* The central 100×100 region has 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 $r\left(\mathbf{x}^{(0)}\right)$ was set to a delta function for known data, and a constant for missing data.

References

- T. Salimans, A. Diederik, D. P. Kingma, M. Welling. Markov Chain Monte Carlo and Variational Inference: Bridging the Gap, ICML, 2015.
- J. Sohl-Dickstein, E. A. Weiss, N. Maheswaranathan, and S. Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics, ICML, 2015.