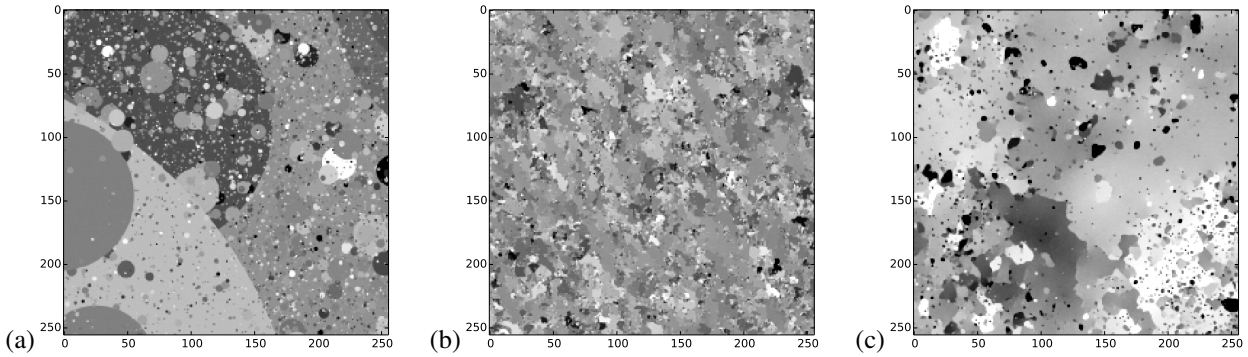

Figure 4. The proposed framework trained on dead leaf images (Jeulin, 1997; Lee et al., 2001). *(a)* Example training image. *(b)* A sample from the previous state of the art natural image model (Theis et al., 2012) trained on identical data, reproduced here with permission. *(c)* A sample generated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

where $\tilde{Z}_t\left(\mathbf{x}^{(t+1)}\right)$ is the normalization constant.

For a Gaussian, each diffusion step is typically very sharply peaked relative to $r\left(\mathbf{x}^{(t)}\right)$, due to its small variance. This means that $\frac{r\left(\mathbf{x}^{(t)}\right)}{r\left(\mathbf{x}^{(t+1)}\right)}$ can be treated as a small perturbation to $p\left(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)}\right)$. A small perturbation to a Gaussian effects the mean, but not the normalization constant, so in this case Equations 21 and 22 are equivalent (see Appendix C).

2.5.3. Applying $r\left(\mathbf{x}^{(t)}\right)$

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

If $r\left(\mathbf{x}^{(t)}\right)$ can be multiplied with a Gaussian (or binomial) distribution in closed form, then it can be directly multiplied with the reverse diffusion kernel $p\left(\mathbf{x}^{(t)}|\mathbf{x}^{(t+1)}\right)$ in closed form. This applies in the case where $r\left(\mathbf{x}^{(t)}\right)$ consists of a delta function for some subset of coordinates, as in the inpainting example in Figure 5.

2.5.4. Choosing $r\left(\mathbf{x}^{(t)}\right)$

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

$$r\left(\mathbf{x}^{(t)}\right) = r\left(\mathbf{x}^{(0)}\right). \tag{23}$$

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

2.6. Entropy of Reverse Process

Since the forward process is known, we can derive upper and lower bounds on the conditional entropy of each step in the reverse trajectory, and thus on the log likelihood,

$$H_q\left(\mathbf{X}^{(t)}|\mathbf{X}^{(t-1)}\right) + H_q\left(\mathbf{X}^{(t-1)}|\mathbf{X}^{(0)}\right) - H_q\left(\mathbf{X}^{(t)}|\mathbf{X}^{(0)}\right) \leq H_q\left(\mathbf{X}^{(t-1)}|\mathbf{X}^{(t)}\right) \leq H_q\left(\mathbf{X}^{(t)}|\mathbf{X}^{(t-1)}\right), \tag{24}$$

where both the upper and lower bounds depend only on $q\left(\mathbf{x}^{(1\cdots T)}|\mathbf{x}^{(0)}\right)$, 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 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). Model training was with SFO (Sohl-Dickstein et al., 2014), except for CIFAR-10. CIFAR-10 results used the

[3] An earlier version of this paper reported higher log likelihood bounds on CIFAR-10. These were the result of the model learning the 8-bit quantization of pixel values in the CIFAR-10 dataset. The log likelihood bounds reported here are instead for data that has been pre-processed by adding uniform noise to remove pixel quantization, as recommended in (Theis et al., 2015).