We start with initial images $x_0 \sim q(x_0)$, where q is a distribution that we want to approximate. Then, we assume Gaussian noise in the forward process in the form $q(x_t|x_{t-1}) = \mathcal{N}(x_t; \sqrt{\alpha_t}\, x_{t-1}, (1-\alpha_t)I)$.

Due to the "nice" properties of Gaussians we can get a closed form solution of $q(x_t|x_0)$. We can look inductively:

$\forall t\ x_t \sim q(x_t|x_{t-1})$

$$x_t = \sqrt{\alpha_t}\, x_{t-1} + \sqrt{1-\alpha_t}\, \varepsilon_{t-1}$$

$$= \sqrt{\alpha_t}\left(\sqrt{\alpha_{t-1}}\, x_{t-2} + \sqrt{1-\alpha_{t-1}}\, \varepsilon'_{t-2}\right) + \sqrt{1-\alpha_t}\, \varepsilon_{t-1}$$

$$= \sqrt{\alpha_t \alpha_{t-1}}\, x_{t-2} + \sqrt{\alpha_t - \alpha_t \alpha_{t-1}}\, \varepsilon'_{t-2} + \sqrt{1-\alpha_t}\, \varepsilon_{t-1}$$

$$= \sqrt{\alpha_t \alpha_{t-1}}\, x_{t-2} + \sqrt{\alpha_t - \alpha_t \alpha_{t-1} + 1 - \alpha_t}\, \varepsilon_{t-2}$$

$$= \sqrt{\alpha_t \alpha_{t-1}} X_{t-2} + \sqrt{1-\alpha_t \alpha_{t-1}}\ \varepsilon_{t-2}$$

Where I used the fact that the sum of two Gaussian random variables is still a Gaussian random variable with mean and variance as a sum of the means and variances of the two rand. variables.

We see that $q(X_t|X_0) = \mathcal{N}\left(X_t; \sqrt{\prod_{i=1}^{t} \alpha_i}\, X_0,\ 1 - \prod_{i=1}^{t} \alpha_i\ I\right)$ We will use the notation from the DDPM paper, $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_t$

From there we can calculate the posterior

$$q(X_{t-1}|X_t, X_0) = \frac{q(X_t|X_{t-1}, X_0)\, q(X_{t-1}|X_0)}{q(X_t|X_0)}$$

$$= \frac{\mathcal{N}\left(X_t; \sqrt{\alpha_t} X_{t-1}, (1-\alpha_t) I\right) \mathcal{N}\left(X_{t-1}; \sqrt{\bar{\alpha}_{t-1}}, (1-\bar{\alpha}_{t-1}) I\right)}{\mathcal{N}\left(X_t; \sqrt{\bar{\alpha}_t}, (1-\bar{\alpha}_t) I\right)}$$

which is a lengthy calculation by writing the exponentials. See "Understanding Diffusion

models: A unified perspective" (or a full derivation.

$$\propto \mathcal{N}\left(x_{t-1}; \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)x_0}{1-\bar{\alpha}_t}, \right.$$

$$\left. \frac{(1-\alpha_t)(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} \mathbb{I}\right).$$

The ELBO in this case is $\log p_\theta(x) \geq \mathbb{E}_{q(x_{1:T}|x_0)}$

$$\left[\log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)}\right], \text{ where } p_\theta \text{ is the reverse process dependent on the neural network's parameters}$$

Where $\log p(x_0) = \log \int p(x_{0:T})\, dx_{1:T}$.

T being the final step in the diffusion process.

So, $\mathbb{E}_{q(x_{1:T}|x_0)}\left[\log \frac{p(x_{0:T})}{q(x_{1:T}|x_0)}\right]$

In the rest of the calculation I will stop writing the expectation

$$\Rightarrow \log \frac{p_\theta(x_{0:T})}{q(x_{1:T}|x_0)} = \log \frac{p(x_T)\prod_{t=1}^{T} p_\theta(x_{t-1}|x_t)}{\prod_{t=1}^{T} q(x_t|x_{t-1})}$$

$$\prod_{t=1} q(x_t|x_{t-1})$$

$$= \log \frac{p_\theta(x_T)\, p_\theta(x_0|x_1) \prod_{t=2}^{T} p_\theta(x_{t-1}|x_t)}{q(x_1|x_0) \prod_{t=2}^{T} q(x_t|x_{t-1})}$$

$$= \log \frac{p_\theta(x_t)\, p_\theta(x_0|x_1)}{q(x_1|x_0)}$$

$$+ \log \frac{\prod_{t=2}^{T} p_\theta(x_{t-1}|x_t)}{\prod_{t=2}^{T} q(x_t|x_{t-1})}$$

$$= \log \frac{p_\theta(x_T)\, p_\theta(x_0|x_1)}{q(x_1|x_0)} + \log \prod_{t=2}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1})}$$

$$= \log \frac{p_\theta(x_T)\, p_\theta(x_0|x_1)}{q(x_1|x_0)}$$

$$+ \log \prod_{t=2}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1}, x_0)\, q(x_t|x_0)}$$

$$q(x_{t-1}|x_0)$$

$$= \log \frac{p_\theta(x_T)\, p_\theta(x_0|x_1)}{q(x_1|x_0)}$$

$$+ \log \frac{q(x_1|x_0)}{q(x_T|x_0)} + \log \prod_{t=2}^{T} \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1},x_0)}$$

$$= \log \frac{p_\theta(x_T)\, p_\theta(x_0|x_1)}{q(x_T|x_0)} + \sum_{t=2}^{T} \log \frac{p_\theta(x_{t-1}|x_t)}{q(x_t|x_{t-1},x_0)}$$

$$= \mathbb{E}_{q(x_1|x_0)}[\log p_\theta(x_0|x_1)] - D_{KL}(q(x_T|x_0)\|p(x_T))$$

$$- \sum_{t=2}^{T} \mathbb{E}_{q(x_t|x_0)}\left[D_{KL}(q(x_{t-1}|x_t,x_0)\| p_\theta(x_{t-1}|x_t)\right]$$

After plugging back the expectations. All the terms in the above expression will receive a minus sign because of minimizing the negative log-likelihood, so the sign in the ELBO changes. The KL-di-

vergence terms can be calculated exactly since $q(x_{t-1}|x_t, x_0)$ and $p_\theta(x_{t-1}|x_t)$ are Gaussians. $D_{KL}(q(x_T|x_0)\,||\,p(x_T))$ is zero because of assuming $q(x_T|x_0)$ to be a standard Gaussian as well as $p(x_T)$. In the calculation above, we also used the Markovian assumption $p_\theta(x_t|x_{t-1}, x_{t-2}, \dots) = p_\theta(x_t|x_{t-1})$.

For the KL'n, $D_{KL}(q(x_{t-1}|x_t, x_0)\,||\,p_\theta(x_{t-1}|x_t))$

$$= \frac{1}{2}\,\frac{1}{\sigma_q^2}\,\left\|\mu_\theta - \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)x_0}{1-\bar{\alpha}_t}\right\|^2$$

the variance of $q(x_{t-1}|x_t, x_0)$

Where $p_\theta(x_{t-1}|x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t), \sigma_q^2 I)$

By a reparametrization of μ_θ as

$$\mu_\theta = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)\hat{x}_\theta(x_t, t)}{1-\bar{\alpha}_t}$$

we get : $\frac{1}{2}\frac{1}{\sigma_q^2}\cdot\frac{\bar{\alpha}_{t-1}(1-\alpha_t)^2}{(1-\bar{\alpha}_t)^2}\left\|\hat{x}_\theta(x_t,t)-x_0\right\|_2^2$

Furthermore, to improve training efficiency we can uniformly sample from $[0,T]$ instead of taking every $t\in[0,T]$. Indeed, this provides an approximation for the actual value. There is a deeper question regarding the meaning of using a NN to predict x_0 given the noisy image. Does it imply that when we sample from $\mathcal{N}(0,I)$, the resulting image generated is the prediction of the NN? Well, in theory yes. If the network is trained so that it captures the data characteristics, then it will produce a high quality image. In almost all scenarios, the dataset is very complex,

so the NN might [illegible] for less noisy steps, but for steps close to T the result might still be very noisy. The key word here is sample quality. You might want to think that we want to minimize ELBO and model this reverse process, and to do that we need the prediction at step t be close to the initial image. Afterwards, we use the reverse process to sample. So, we have this construction in order to model the data, otherwise try using the NN always and compare it to see if there are major differences (there are). Even so, it was tested and predicting ϵ the noise instead of $\hat{x}_\theta(x_t, t)$ achieves a better sampling quality.

Let's start again: $x_t = \sqrt{\bar{\alpha}_t} x_0 + \sqrt{1-\bar{\alpha}_t}\,\epsilon$

$$\Rightarrow x_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\,\epsilon}{}$$

$\sqrt{\alpha_t}$

$$\Rightarrow \mu_q(x_t, x_0) = \frac{\sqrt{\alpha_t}(1-\bar{\alpha}_{t-1})x_t + \sqrt{\bar{\alpha}_{t-1}}(1-\alpha_t)x_0}{1-\bar{\alpha}_t}$$

After substituting x_0 in μ_q we get

$$\mu_q = \frac{1}{\sqrt{\alpha_t}} x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \epsilon$$

Thus, we can reparametrize $\mu_\theta(x_t, t) = \frac{1}{\sqrt{\alpha_t}} x_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}\sqrt{\alpha_t}} \hat{\epsilon}_\theta(x_t, t)$, predicting the noise.

Empirically there is an improved sampling quality. The question about the fact we know x_0 and ϵ during training, then why we use neural networks for them is a very good one. This is something usually omitted, but of crucial significance. During training, even though we know x_0 and ϵ because we use them to make the training dataset, the NN doesn't know about them (and that's

why is learning). The NN at step t has as input (x_t, t), where $x_t = x_0 + \text{noise}_t$. Therefore, it needs to decouple the original image and the noise from x_t.

Furthermore, the term L_0 can be ignored because the reconstructed image will be very close to the original at the first step where the added noise is substantially smaller. Also, the gradient from L_0 is very small after the model trained for big steps.