

Diffusion Language Models (DLM)

- ① J. Sohl-Dickstein, E. Weiss, N. Maheswaranathan, S. Ganguli, Deep unsupervised learning using non-equilibrium thermodynamics, arXiv 1503.03585, 2015.
- ② J. Austin, D. Johnson, J. Ho, D. Tarlow, R. van den Berg, Structured denoising diffusion models in discrete state-spaces, arXiv 2107.03006v3, 2023.

(above are general DLMs)

- ③ S. Schöe et. al., Simple and effective masked diffusion language models, arXiv 2406.07524v2, 2024
- ④ J. Shi, K. Han, Z. Wang, A. Doucet, M. Titsias, Simplified and generalized masked diffusion for discrete data, arXiv 2406.04329v4, 2025.

(above are masked DLMs)

- ⑤ S. Dieleman et. al., Continuous diffusion for categorical data, arXiv 2211.15089v3, 2022.

(above is continuous DLM)

- ⑥ S. Liu et. al., Think while you generate: Discrete diffusion with planned denoising, arXiv 2410.06264v2, 2025

(above is a uniform DLM)

⑦ Y. Lipman et. al., Flow matching guide and code, arXiv 2412.06246v1, 2024.

⑧ A. Lou, C. Meng, S. Ermon, Discrete diffusion modeling by estimating the ratios of data distribution, arXiv 2310.16834v3, 2024

(above are from a flow perspective).

⑨ L. Rout, C. Caramanis, S. Shakkottai, Anchored diffusion language model, NeurIPS 2025.

(masked DLM with supervised latent).

Note that there are many other DLMs that either propose a new model, and/or scale the above models; e.g., LLaDA, GGM, DVO, VDLM, GIDD, BD3-LM.

(masking)

(uniform noise)

(mixed noise)

(Diffusion + AR)

Please refer to [9] for references.

Thanks to Litu Rout for discussions on the notes and figures.

Overview of a Diffusion Language Model

Figure 1

Forward (noising) process.

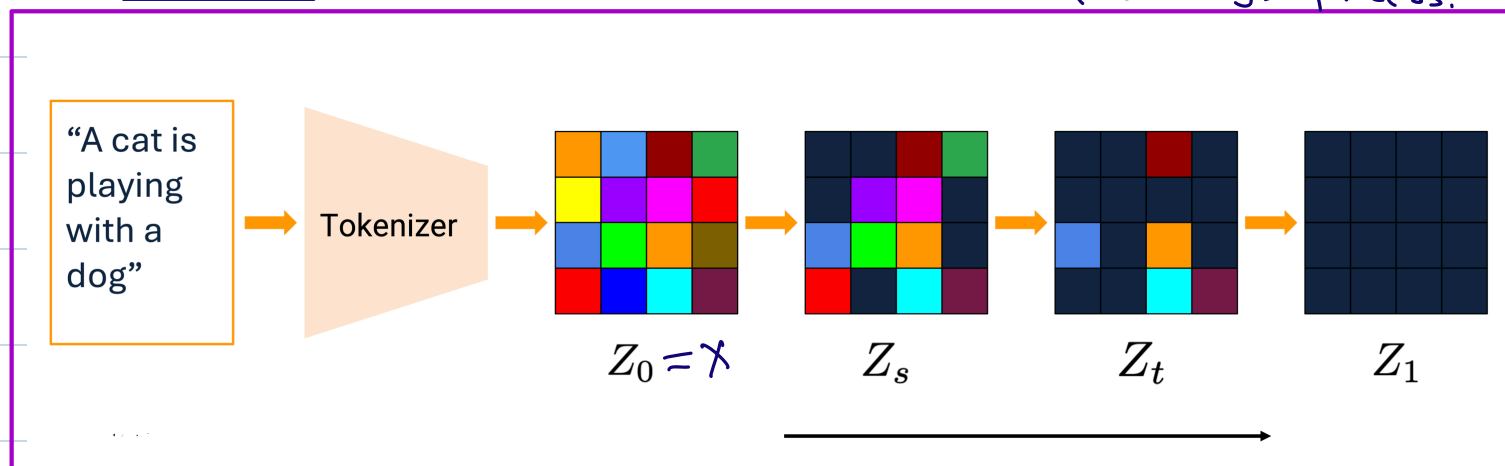


Image: Inspired by talks from [1,2,3,4].

Before we discuss general DLMs, consider the (currently) most popular noising model, the masked DLM. At a high level, language is tokenized, where each token is an integer representing a word in a (learned) dictionary. For example, if a GPT-2 Byte-Pair Encoding (BPE) is used, the basic dictionary consists of a vocabulary of 50,257 words in the dictionary, which is augmented with a few extra words such as 'mask', and any other special words as needed.

Unlike autoregressive (AR) language models, a DLM

has pre-decided a fixed number of tokens. (e.g. 4096 tokens), where each token is an integer from the vocabulary.

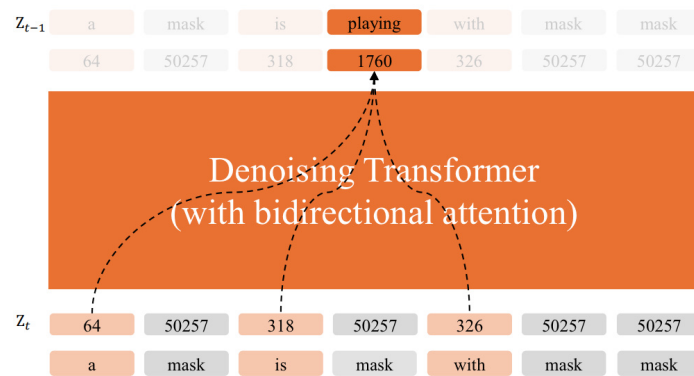
Training a DLM

In the case of a masked DLM, the forward noising process starts with a L -token sequence (in Figure 1, arranged as a grid of colors, with each color representing a distinct integer). The diffusion evolves over T steps, indexed by time $t \in \{ \frac{0}{T}, \frac{1}{T}, \dots, \frac{T}{T} \}$.

At a random time (we will specify the details later), each token is flipped to a 'mask' token, such that at $t=1$, all tokens are masked (see Figure 1).

The reverse process learns to unmask tokens, and the learned denoising transformer is trained using a cross entropy loss (details later in this note).

Figure 2: Inference with masked DLM



Denoising Transformer: Unmasks using already unmasked tokens

← L tokens →

D3PM: <https://arxiv.org/abs/2107.03006> MDLM: <https://s-sahoo.com/mdlm/> MD4: <https://arxiv.org/abs/2406.04329>

Inference with masked DLM

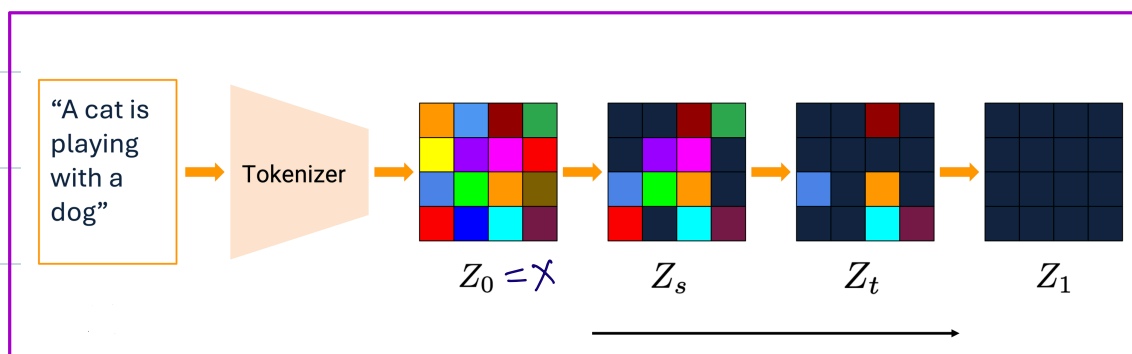
During inference, we start with all masks, and iteratively unmask as illustrated in Figure 2. The denoising transformer (with bidirectional attention) predicts logits for all currently masked tokens, based on already unmasked tokens. Then the next iterate of tokens are sampled from these predicted logits, (one or more than one token can be unmasked, depending on the specific unmasking strategy; thus parallel decoding is possible.)

For inference with a prompt, the prompt is tokenized and the initial token state is set to the state: $[\text{prompt}] [\text{prompt}] \dots [\text{prompt}] [\text{mask}] \dots [\text{mask}]$

L -length token sequence.

Note that we will need to do supervised fine tuning on the base model (which is trained to only start from all masks) to handle prompts as above (e.g., instruction fine tuning with question-answer pairs).

Masked Diffusion Language Model (we use, notation from D3PM [2], see also MDLM [3] MD4 [4] and ADLM [9]).



$X \sim q(x)$, ground-truth data joint dist. over L -length token sequence.

$K \rightarrow$ vocabulary size ; $\mathcal{V} = \{1, 2, \dots, K\}$.

(note that this includes the 'mask' token, encoded as the integer K).

token sequence $\mathcal{X} = (x^1 x^2 \dots x^L)$,

with $x^l \in \text{Simplex}(K-1)$, i.e.,

$$x^l = \begin{pmatrix} x^l(1) \\ \vdots \\ x^l(K) \end{pmatrix}, \quad x^l(j) \in \{0, 1\}, \quad \sum_{j=1}^K x^l(j) = 1$$

i.e., one-hot encoding representation of a word in the vocabulary, with $x^l(K)$ corresponding to 'mask'.

The forward process: The process is modeled as a time inhomogeneous Markov Chain over a discrete state space.

$$X \rightarrow Z_0 \rightarrow Z_{\frac{1}{T}} \rightarrow \dots \rightarrow Z_{s(i)} \rightarrow Z_{t(i)} \rightarrow \dots \rightarrow Z_1$$

typically, $Z_0 = X$, but they can differ; see pp20

$$s(i) = \frac{i-1}{T}$$

$$t(i) = \frac{i}{T}$$

all mask
↑ absorbing state

$$t(i) = \frac{i}{T}, \quad i \in \{0, 1, 2, \dots, T\}.$$

$$s(i) = t(i-1) \quad \left(\text{i.e., one step before } t(i) \right).$$

$$q(z_t | x) = \prod_{l=1}^L q(z_t^l | x^l)$$

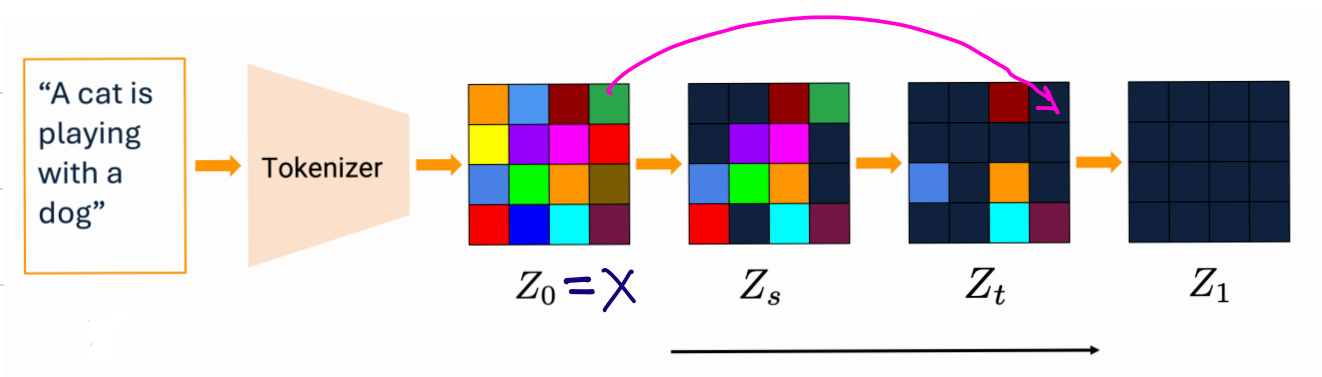
→ ①

$$\text{where } q(z_t^l | x^l) = \text{Cat}(z_t^l : \alpha_t x^l + (1 - \alpha_t) m),$$

$$\text{where } m = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{pmatrix}, \quad \alpha_t \in [0, 1],$$

$$\alpha_0 = 1, \quad \alpha_0 > \alpha_{1/T} > \dots > \alpha_1 = 0$$

token masked w.p. $(1 - \alpha_t)$



Notation: $r(y) = \text{Cat}(x; \alpha a + (1-\alpha)b)$

for $a, b \rightarrow$ one-hot vectors, $\alpha \in [0, 1]$

$$y \in \{a, b\}, \quad r(a) = P(Y=a) = \alpha$$
$$r(b) = P(Y=b) = (1-\alpha).$$

If $C \in \text{Simplex}(k-1)$ i.e., $C = \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_k \end{pmatrix}$,
 $\hookrightarrow$ prob. vector.

$$\text{with } c_i \geq 0, \quad \sum_{i=1}^k c_i = 1$$

Then $r(y) = \text{Cat}(x; \alpha C + (1-\alpha)b)$

$\alpha \in [0, 1]$

$\hookrightarrow$ prob. vector

$\hookrightarrow$ one-hot, with '1' at j^* component.

$$r(j) = P(Y=j) = \alpha c_j, \quad j \neq j^*$$

$$r(j^*) = P(Y=j^*) = \alpha c_{j^*} + (1-\alpha).$$

Equivalently, the forward process is a time-inhomogeneous Markov Chain:

→ ②

$$q(z_t^l | z_s^l) = \text{Cat}\left(z_t^l; \frac{\alpha_t}{\alpha_s} z_s + \left(1 - \frac{\alpha_t}{\alpha_s}\right) m\right)$$

$t = t(i)$

$s = t(i-1)$

(Note that notation is abused as earlier notes; the variable z_t encodes both the value of the state as well as the time).

Suppose $z_0^l = x^l = x^l$. Then, observe that the above time-inhomogeneous MC can either be in state x^l or state 'm', and once masked, remains 'm' for all future times (absorbing state).

(recall $\alpha_0 = 1$, $\alpha_{t(i)} < \alpha_{t(i-1)}$ by defn. of $\alpha_{t(i)}$)

Showing that the two characterizations are equivalent:

$$q(z_t^l | x^l) = \text{Cat}(z_t^l : \alpha_t x^l + (1 - \alpha_t) m) \quad \text{①}$$

$$q(z_t^l | x^l) = \sum_{z_s^l} q(z_t^l | z_s^l, x^l) q(z_s^l | x^l)$$

$$= \sum_{z_s^l} q(z_t^l | z_s^l) q(z_s^l | x^l)$$

$\hookrightarrow \in \{x^l, m\}$

Also, recall $z_t^l \in \{x^l, m\}$ by construction. Thus,

$$\text{We denote } P(z_t^l = m | x^l = x^l) \doteq q(z_t^l = m | x^l)$$

(bad notation, but consistent with literature)

$$\begin{aligned} \text{Then, } q(z_t^l = m | x^l) &= q(z_t^l = m | z_s^l = m) \cdot q(z_s^l = m | x^l) \\ &+ q(z_t^l = m | z_s^l = x^l) q(z_s^l = x^l | x^l) \end{aligned}$$

i.e.,

$$(1 - \alpha_t) = 1 \cdot (1 - \alpha_s) + q(z_t^l = x^l | z_s^l = x^l) \cdot \alpha_s.$$

$$\Rightarrow q(z_t^l = x^l | z_s^l = x^l) = 1 - \frac{\alpha_t}{\alpha_s}$$

$$\Rightarrow q(z_t^l = m | z_s^l = x^l) = \left(\frac{\alpha_t}{\alpha_s} \right) \quad \square$$

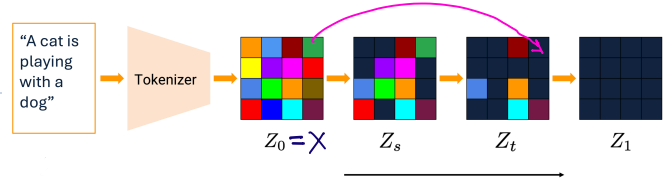
Next, the reverse posterior distribution is given by:

$$q(z_s^l | z_t^l, x^l) = \frac{q(z_s^l, z_t^l, x^l)}{q(z_t^l, x^l)}$$

$$= \frac{q(z_t^l | z_s^l, x^l) \cdot q(z_s^l | x^l) \cancel{q(x^l)}}{q(z_t^l | x^l) \cancel{q(x^l)}}$$

$$= \frac{q(z_t^l | z_s^l) \cdot q(z_s^l | x^l)}{q(z_t^l | x^l)} \rightarrow \textcircled{3}$$

Consider two cases:



(i) $z_t^l \neq m$: Here, $z_s^l \neq m$ and $z_s^l = z_t^l$

$$\therefore q(z_s^l | z_t^l, x^l) = \text{Cat}(z_s^l; z_t^l) \rightarrow \textcircled{4a}$$

(ii) $z_t^l = m$: Here, $z_s^l = m$ or $z_s^l = x^l$

$$q(z_s^l = m | z_t^l = m, x^l) \stackrel{\textcircled{3}}{=} \frac{q(z_t^l = m | z_s^l = m) q(z_s^l = m | x^l)}{q(z_t^l = m | x^l)}$$

$$= \frac{1 \cdot (1 - \alpha_s)}{1 - \alpha_t}$$

$$q(z_t^l | x^l) = \text{Cat}(z_t^l; \alpha_t x^l + (1 - \alpha_t)m) \rightarrow \textcircled{1}$$

$$\therefore q(z_s^l = x^l | z_t^l = m, x^l) = 1 - \frac{(1-\alpha_s)}{1-\alpha_t} = \frac{\alpha_s - \alpha_t}{1-\alpha_t}$$

i.e.,

→ (4b)

$$q(z_s^l | z_t^l, x^l) = \text{Cat}\left(z_s^l; \left(\frac{\alpha_s - \alpha_t}{1-\alpha_t}\right) x^l + \left(\frac{1-\alpha_s}{1-\alpha_t}\right) m\right)$$

Learned Reverse Process for Unmasking

Similar to the ELBO derivation in Note 10, we now parameterize a learned reverse process for unmasking. Specifically, we learn a reverse transition probability $p_\theta(z_s | z_t)$ s.t.

↳ parameterization

$$p_\theta(z_s | z_t) \doteq \prod_{l=1}^L p_\theta(z_s^l | z_t) \rightarrow (\otimes)$$

(parameterized as conditional product form)

(Note: We use $\doteq$ to indicate design choice / parameterization)

and $p_{\theta}(z_s^l | z_t)$ is trained to approximate the reverse posterior distribution, by learning to predict x^l from z_t , i.e.,

from z_t , predict a categorical distribution over x^l . Unlike the continuous state-space setting where we used Tweedie's formula to predict the conditional expectation of x_0 given x_t , here, the prediction is a categorical distribution, i.e.,

$x_{\theta}^l(z_t) \in \text{Simplex}(K-1)$ is a prob. vector, with

$$x_{\theta}^l(z_t)(j) \geq 0, \quad \sum_{j=1}^K x_{\theta}^l(z_t)(j) = 1.$$

Next, parameterize the learned reverse transition probability $p_{\theta}(z_s^l | z_t)$ using the reverse posterior $q(z_s^l | z_t, x^l)$ (see ④) as follows.

Recall: $q(z_s^l | z_t, x^l) = \begin{cases} m & \text{w.p. } \frac{1-\alpha_s}{1-\alpha_t} \\ x^l & \text{w.p. } \frac{\alpha_s - \alpha_t}{1-\alpha_t} \end{cases}$

Also, the **learned** categorical predictor $\hat{x}^l$ is:

$$\hat{x}^l = j \quad \text{with prob. } x_{\theta}^l(z_t)(j)$$

↳ j^{th} component of the learned predictor

Thus, we parameterize:

$$p_{\theta}(z_s^l | z_t)(j) \doteq q(z_s^l | z_t, \hat{x}^l = j) \underbrace{x_{\theta}^l(z_t)(j)}_{\substack{\text{prob. that} \\ \hat{x}^l = j}}$$

learn to approximate

Interpretation: $p_{\theta}(z_s^l | z_t) \approx q(z_s^l | z_t, x^l)$

Since we do not know x^l , predict a distribution over x^l and use the weighted reverse posterior, with weights corresponding to the estimated probabilities.

Recall that from 4a and 4b, we have:

$$q(z_s^l | z_t^l, x^l) = \begin{cases} \text{Cat}(z_s^l; z_t^l), & z_t^l \neq m \\ \text{Cat}\left(z_s^l; \left(\frac{\alpha_s - \alpha_t}{1 - \alpha_t}\right) x^l + \left(\frac{1 - \alpha_s}{1 - \alpha_t}\right) m\right), & z_t^l = m \end{cases}$$

→ ④

Thus, when $z_t^l = m$, we have:

$$p_\theta(z_s^l | z_t) = \begin{cases} m & \text{w.p. } \left(\frac{1 - \alpha_s}{1 - \alpha_t}\right) \\ \vdots & \\ j & \text{w.p. } \left(\frac{\alpha_s - \alpha_t}{1 - \alpha_t}\right) \cdot x_\theta^l(z_t)(j) \\ \vdots & \end{cases}$$

$$= \text{Cat}\left(z_s^l; \left(\frac{\alpha_s - \alpha_t}{1 - \alpha_t}\right) x_\theta^l(z_t) + \left(\frac{1 - \alpha_s}{1 - \alpha_t}\right) m\right)$$

and when $z_t^l \neq m$, $z_s^l = z_t^l$, i.e.,

$$p_\theta(z_s^l | z_t) = \text{Cat}(z_s^l; z_t^l)$$

i.e.,

→ (5)

$$p_\theta(z_s^l | z_t) \doteq \begin{cases} \text{Cat}(z_s^l; z_t^l) & , z_t^l \neq m \\ \text{Cat}\left(z_s^l; \left(\frac{\alpha_s - \alpha_t}{1 - \alpha_t}\right) x_\theta^l(z_t) + \left(\frac{1 - \alpha_s}{1 - \alpha_t}\right) m\right) & , \\ & z_t^l = m \end{cases}$$

In literature, this is usually written as:

$$p_\theta(z_s^l | z_t) \doteq q(z_s^l | z_t, \underbrace{x_\theta^l(z_t)}_{\text{this is now a prob vector, and need not be one-hot.}})$$

this is now
a prob vector, and
need not be one-hot.

The learned categorical distribution $x_\theta^l(z_t)$

has two properties ([3, 4]):

① Zero-masking: $\langle \chi_{\theta}^l(z_t), m \rangle = 0$

(i.e., do not predict mask)

② Carry-over unmasking: For any $z_t^l \neq m$,

$\langle \chi_{\theta}^l(z_t), z_t^l \rangle = 1$, i.e., for an

unmasked token z_t^l , predict $\hat{x}^l = z_t^l$ w.p. 1.

Deriving the Training Loss:

Similar to Note 10, the training loss is derived using the ELBO lower bound (also referred to as NELBO $\rightarrow$ Negative ELBO in literature).

So far, we have denoted $Z_0 = X$ (i.e., the first step of the discrete diffusion process is the ground truth). However, sometimes it might be useful to have an extra post-processing step (e.g. extract answer, or permute tokens, etc.). In this case, we would have:

$$X \rightarrow Z_0 \rightarrow \dots \rightarrow Z_s \rightarrow Z_t \rightarrow \dots \rightarrow Z_1$$

↳ to enable
learned post processing
in the reverse process.

Training Loss:

reconstruction / post diffusion loss = 0 if $x = Z_0$

$$\mathcal{L}_\theta(x) = E_{Z_0 \sim q(\cdot|x)} \left[-\ln p_\theta(x|Z_0) \right]$$

$$+ \sum_{i=1}^T E_{Z_{t(i)} \sim q(\cdot|x)} \left[\left(\frac{\alpha_{t(i)} - \alpha_{s(i)}}{1 - \alpha_{t(i)}} \right) \sum_{l=1}^L \ln \langle x_\theta^l(Z_{t(i)}), x^l \rangle \right]$$

↙
Cross-entropy loss between
predicted categorical distribution and
one-hot encoded ground truth token

Claim: $-\ln p_\theta(x) \leq \mathcal{L}_\theta(x)$

PF: $x = (x^1, x^2, \dots, x^L)$. In this proof, we will write the proof for $L=1$ (the general case is available in [1] and follow ups)

Recall the ELBO lower bound (Note 8, pp. 9-10):

Note 8

$$\begin{aligned}
 \ln p_\theta(x) &= \ln \int_z p_\theta(x, z) dz \\
 &\quad \text{log-likelihood fn} \quad \quad \quad \text{latent.} \\
 &= \ln \int_z p_\theta(x, z) \cdot \frac{q_\phi(z|x)}{q_\phi(z|x)} dz = \ln E_{z \sim q_\phi(\cdot|x)} \left[\frac{p_\theta(x, z)}{q_\phi(z|x)} \right] \\
 &\quad \quad \quad \text{Jensen's Ineq.} \quad \quad \quad \text{phi parameterizes forward process } q_\phi(z|x) \\
 &\geq E_{z \sim q_\phi(\cdot|x)} \left[\ln \left(\frac{p_\theta(x, z)}{q_\phi(z|x)} \right) \right] \rightarrow \text{ELBO} \\
 &\quad \quad \quad \text{ELBO}(x).
 \end{aligned}$$

Now, as in Note 10 (DDPM ELBO derivation, pp 5 and Decomp #2, pp. 9), we have $\phi \rightarrow$ forward process parameterized by $\{\alpha_t\}$, and

$Z = (z_0, \dots, z_1)$ as the latent variables, instead of $(x_1, x_2, \dots, x_T)$ as in Note 10, inset

to the right $\rightarrow$

$$\begin{aligned} \textcircled{\text{El 20}} &= E_{(x_1, \dots, x_T) \sim q(\cdot | x_0)} \left[-\ln \frac{p_\theta(x_0, x_1, x_2, \dots, x_T)}{q(x_1, x_2, \dots, x_T | x_0)} \right] \\ &= E_{(x_1, \dots, x_T) \sim q(\cdot | x_0)} \left[-\ln \frac{p_\theta(x_T)}{q(x_T | x_0)} - \sum_{t=2}^T \ln \frac{p_\theta(x_{t-1} | x_t)}{q(x_{t-1} | x_t, x_0)} \right. \\ &\quad \left. - \ln p_\theta(x_0 | x_1) \right] \end{aligned}$$

Thus we have:

$$-\ln p_\theta(x) \leq E_{q(z_0, \dots, z_1 | x)} \left[\underbrace{-\ln p_\theta(x | z_0)}_{=0 \text{ if } z_0=x} \right]$$

$$+ \underbrace{\ln \frac{q(z_1 | x)}{p_\theta(z_1)}}_{=0 \text{ if } z_1 \text{ is all masks}} + \underbrace{\sum_{i=1}^T \ln \frac{q(z_{s(i)} | z_{t(i)}, x)}{p_\theta(z_{s(i)} | z_{t(i)})}}_{\textcircled{\text{b iii}}}]$$

$\rightarrow \textcircled{\text{b}}$

We now simplify the third term ($\textcircled{\text{b iii}}$) as follows:

$$\textcircled{\text{b iii}} = E_{q(z_0, \dots, z_1 | x)} \left[\sum_{i=1}^T \ln \frac{q(z_{s(i)} | z_{t(i)}, x)}{p_\theta(z_{s(i)} | z_{t(i)})} \right]$$

Consider the i^{th} term in this expression:

$$b^{(iii)}, i^{\text{th}} \text{ term} = E_{\substack{(z_{s(i)}, z_{t(i)}) \\ \sim q(\cdot|x)}} \left[\ln \frac{q(z_{s(i)} | z_{t(i)}, x)}{p_{\theta}(z_{s(i)} | z_{t(i)})} \right]$$

$$= E_{z_{t(i)} \sim q(\cdot|x)} \left[E_{z_{s(i)} \sim q(z_{s(i)} | z_{t(i)}, x)} \left[\ln \frac{q(z_{s(i)} | z_{t(i)}, x)}{p_{\theta}(z_{s(i)} | z_{t(i)})} \right] \right] \quad \textcircled{a}$$

Case (i): $z_{t(i)} \neq m$ (i.e., $z_{t(i)} = x$).

Here, $\textcircled{a} = 0$. To see this,

$$\begin{aligned} \textcircled{a} &= \underbrace{q(z_{s(i)} = m | z_{t(i)} = x, x)}_{=0} \ln \frac{q(z_{s(i)} = m | z_{t(i)} = x, x)}{p_{\theta}(z_{s(i)} = m | z_{t(i)} = x)} \\ &\quad + \underbrace{q(z_{s(i)} = x | z_{t(i)} = x, x)}_{=1} \ln \frac{\overbrace{q(z_{s(i)} = x | z_{t(i)} = x, x)}^{=1}}{\underbrace{p_{\theta}(z_{s(i)} = x | z_{t(i)} = x)}_{=1}} \end{aligned}$$

(because of carry-over unmasking)

Case(ii) $z_{t(i)} = m$

$$a = E_{z_{s(i)} \sim q(z_{s(i)} | z_{t(i)}, x)} \left[\ln \frac{q(z_{s(i)} | z_{t(i)}, x)}{p_{\theta}(z_{s(i)} | z_{t(i)})} \right]$$

$$= q(z_{s(i)} = m | z_{t(i)} = m, x) \ln \frac{q(z_{s(i)} = m | z_{t(i)} = m, x)}{p_{\theta}(z_{s(i)} = m | z_{t(i)} = m)}$$

$$+ q(z_{s(i)} = x | z_{t(i)} = m, x) \ln \frac{q(z_{s(i)} = x | z_{t(i)} = m, x)}{p_{\theta}(z_{s(i)} = x | z_{t(i)} = m)}$$

In this case,

$$p_{\theta}(z_{s(i)} | z_{t(i)} = m) = \text{Cat}\left(z_s; \left(\frac{\alpha_s - \alpha_t}{1 - \alpha_t}\right) x_{\theta}(z_t) + \left(\frac{1 - \alpha_s}{1 - \alpha_t}\right) m\right)$$

$\rightarrow$ ($L=1$ case here; in general, we have $\alpha_{\theta}^L(\cdot)$)

$$q(z_{s(i)} | z_{t(i)} = m, x) = \text{Cat}\left(z_s; \left(\frac{\alpha_s - \alpha_t}{1 - \alpha_t}\right) x + \left(\frac{1 - \alpha_s}{1 - \alpha_t}\right) m\right)$$

$$\therefore \textcircled{a} = \left(\frac{1-\alpha_s}{1-\alpha_t} \right) \ln \frac{\left(\frac{1-\alpha_s}{1-\alpha_t} \right)}{\left(\frac{1-\alpha_s}{1-\alpha_t} \right)}$$

$$+ \left(\frac{\alpha_s - \alpha_t}{1-\alpha_t} \right) \ln \frac{\left(\frac{\alpha_s - \alpha_t}{1-\alpha_t} \right)}{\left(\frac{\alpha_s - \alpha_t}{1-\alpha_t} \right) \underbrace{x_{\theta}(z_t)(j)}}_{\text{select the } j^{\text{th}} \text{ index, where } j \text{ corresponds to } x}$$

i.e., $x = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ 0 \\ \vdots \end{pmatrix} \rightarrow j^{\text{th}} \text{ position}$

$x_{\theta}(z_t) = \begin{pmatrix} \vdots \\ x_{\theta}(z_t)(j) \\ \vdots \end{pmatrix} \rightarrow j^{\text{th}} \text{ position of the predicted prob. vector}$

But $x_{\theta}(z_t)(j) = \langle x_{\theta}(z_t), x \rangle$.

$$\therefore \textcircled{a} = - \left(\frac{\alpha_s - \alpha_t}{1-\alpha_t} \right) \ln \langle x_{\theta}(z_t), x \rangle$$

Putting both cases together,

$$\therefore \textcircled{a} = 0 \cdot \chi_{\{z_{t(i)} \neq m\}}$$

$$- \left(\frac{\alpha_s - \alpha_t}{1 - \alpha_t} \right) \ln \langle \pi_\theta(z_t), x \rangle \chi_{\{z_{t(i)} = m\}}.$$

This can be dropped; see below.

Finally, observe that if $z_t \neq m$, and given x ,

$\pi_\theta(z_t) = x$, because of carry-over unmasking.

$$\Rightarrow \langle \pi_\theta(z_t), x \rangle = 1$$

$$\Rightarrow \ln \langle \pi_\theta(z_t), x \rangle = 0.$$

$\therefore$

$$\textcircled{a} = - \frac{(\alpha_s - \alpha_t)}{1 - \alpha_t} \cdot \ln \langle \pi_\theta(z_t), x \rangle$$

(no need for indicators)

$$= \left(\frac{\alpha_t - \alpha_s}{1 - \alpha_t} \right) \ln \langle \pi_\theta(z_t), x \rangle$$

∴

$$b(iii), i^{th} \text{ term} = E_{\substack{(z_{s(i)}, z_{t(i)}) \\ \sim q(\cdot|x)}} \left[\ln \frac{q(z_{s(i)}|z_{t(i)}, \alpha)}{p_{\theta}(z_{s(i)}|z_{t(i)})} \right]$$

$$= E_{z_{t(i)} \sim q(\cdot|x)} \left[E_{z_{s(i)} \sim q(z_{s(i)}|z_{t(i)}, \alpha)} \left[\ln \frac{q(z_{s(i)}|z_{t(i)}, \alpha)}{p_{\theta}(z_{s(i)}|z_{t(i)})} \right] \right] \quad \textcircled{a}$$

$$= E_{z_{t(i)} \sim q(\cdot|x)} \left[\left(\frac{\alpha_t - \alpha_s}{1 - \alpha_t} \right) \ln \langle \alpha_{\theta}(z_t), \alpha \rangle \right]$$

Now put together all terms to get the ELBO loss.

