Title

Author 1[*]
Department of YYY, University of XXX
and
Author 2
Department of ZZZ, University of WWW

August 21, 2025

Abstract

Abstrct.

Keywords: 7 or fewer keywords

1 Mixture of diffusion

We consider a generative model that combines both discrete and continuous latent variables. For each observation y_i, we assume the existence of:

- A discrete latent variable $x_i \in \{1, \ldots, K\}$, representing an unobserved class label;
- A continuous latent variable $z_i = (z_{i1}, z_{i2}, \cdots, z_{iT})$, representing the latent variables related to the diffusion process.
- The visible observation $y_i \in \mathcal{Y}$, such as an image.

The joint distribution over these variables is defined as:

$$p_\theta(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z}) = \prod_{i=1}^{n} p_\theta(x_i, z_i, y_i) = \prod_{i=1}^{n} \left[p_\theta(y_i \mid x_i, z_{i1}) \prod_{t=1}^{T-1} p_\theta(z_{i,t} \mid x_i, z_{i,t+1}) p(z_{i,T}) p_\theta(x_i) \right]$$

[*]The authors gratefully acknowledge

where $p_\theta(x_i)$ is a categorical prior, $p(z_{i,T})$ is typically a standard normal distribution, $p_\theta(z_{i,t} \mid x_i, z_{i,t+1})$ is the denoising probability conditioned on the label, and $p(y_i \mid z_{i1}, x_i)$ is the last decoder.

The semi-variational distribution is

$$q(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{Y}, \theta^{(l)}) = q(\boldsymbol{Z} \mid \boldsymbol{Y}) p(\boldsymbol{X} \mid \boldsymbol{Z}, \boldsymbol{Y}, \theta^{(l)}) = \prod_{i=1}^{n} q(z_i \mid y_i) p(x_i \mid z_i, y_i, \theta^{(l)})$$

where $p(x_i \mid z_i, y_i, \theta^{(l)})$ could also be replaced by a variational distribution (full VI). $q(z_i \mid y_i)$ is the forward process of the diffusion, we have

$$q(z_i \mid y_i) = q(z_{i1} \mid y) \prod_{t=1}^{T-1} q(z_{i,t+1} \mid z_{it}).$$

Then for variational EM algorithm, the objective function is the ELBO:

$$\begin{aligned}
\mathcal{L}(\theta) &= E_{q(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{Y},\theta^{(l)})} \left(\frac{p_\theta(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})}{q(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{Y}, \theta^{(l)})} \right) \\
&= E_{q(\boldsymbol{X},\boldsymbol{Z}|\boldsymbol{Y},\theta^{(l)})} \left(\frac{\prod_{i=1}^{n} \left[p_\theta(y_i \mid x_i, z_{i1}) \prod_{t=1}^{T-1} p_\theta(z_{i,t} \mid x_i, z_{i,t+1}) p(z_{i,T}) p_\theta(x_i) \right]}{\prod_{i=1}^{n} q(z_i \mid y_i) p(x_i \mid z_i, y_i, \theta^{(l)})} \right) \\
&= \sum_{i=1}^{n} \mathbb{E}_{\substack{z_i \sim q(z_i|y_i) \\ x_i \sim p(x_i|z_i,y_i,\theta^{(l)})}} \left(\frac{p_\theta(y_i \mid x_i, z_{i1}) \prod_{t=1}^{T-1} p_\theta(z_{i,t} \mid x_i, z_{i,t+1}) p(z_{i,T})}{q(z_i \mid y_i)} + \frac{p_\theta(x_i)}{p(x_i \mid z_i, y_i, \theta^{(l)})} \right) \\
&\propto \sum_{i=1}^{n} \mathbb{E}_{\substack{z_i \sim q(z_i|y_i) \\ x_i \sim p(x_i|z_i,y_i,\theta^{(l)})}} \left(\frac{p_\theta(y_i \mid x_i, z_{i1}) \prod_{t=1}^{T-1} p_\theta(z_{i,t} \mid x_i, z_{i,t+1}) p(z_{i,T})}{q(z_i \mid y_i)} + p_\theta(x_i) \right) \\
&= \mathcal{L}_d(\theta) + \mathcal{L}_s(\theta)
\end{aligned}$$

where $p(x_i \mid z_i, y_i, \theta^{(l)})$ is unrelated to the parameter θ. The former part is related to the diffusion, the later part is related to the label. If x_i is given, the former part is traditional ELBO for diffusion models with condition x_i. The problem is that x_i is also hidden, we need to sample x_i from the posterior distribution if using Monte Carlo to approximate the ELBO.

$$\begin{aligned}
p(x_i \mid z_i, y_i, \theta^{(l)}) &\propto p(z_i, y_i \mid x_i, \theta^{(l)}) p(x_i \mid \theta^{(l)}) \\
p(x_i, z_i, y_i \mid \theta^{(l)}) &= p_{\theta^{(l)}}(y_i \mid x_i, z_{i1}) \prod_{t=1}^{T-1} p_{\theta^{(l)}}(z_{i,t} \mid x_i, z_{i,t+1}) p(z_{i,T}) := \xi^{(l)}(x_i) \\
p(x_i = k \mid z_i, y_i, \theta^{(l)}) &= \frac{\xi^{(l)}(k) p(x_i = k \mid \theta^{(l)})}{\sum_{k=1}^{K} \xi^{(l)}(k) p(x_i = k \mid \theta^{(l)})}
\end{aligned}$$

If we exactly sample x_i from the posterior distribution, we need to go over from 1 to T. It is computational intensive. Instead, we follow the training tricks of DDPM. We sample a time t from the discrete uniform distribution from 1 to T and calculate the denoising probability to approximate $\xi^{(l)}(x_i)$:

$$\tilde{\xi}^{(l)}(x_i)$$

$$:= \begin{cases} e^T p(z_{i,T}) p_{\theta^{(l)}}(y_i \mid x_i, z_{i1}) = e^T p(z_{i,T}) N\left(y_i; \frac{1}{\sqrt{\alpha_1}}\left(z_{i1} - \frac{\beta_1}{\sqrt{1-\bar{\alpha}_1}}\epsilon_{\theta^{(l)}}\left(z_{i1}, 1, x_i\right)\right), \sigma_1^2\right) & t = 1 \\ e^T p(z_{i,T}) p_{\theta^{(l)}}(z_{i,t-1} \mid x_i, z_{i,t}) = e^T p(z_{i,T}) N\left(z_{i,t-1}; \frac{1}{\sqrt{\alpha_1}}\left(z_{it} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_{\theta^{(l)}}\left(z_{it}, t, x_i\right)\right), \sigma_t^2\right) & t = 2, 3, \cdots, T \end{cases}$$

where if $t > 1$, $z_{i,t-1}$ and $z_{i,t}$ are drawn from the distribution $q(z_i \mid y_i)$, i.e., the forward process of diffusion process. It $t = 1$, we only need sample z_{i1}. (Actually, we can always use the last decoder probability? need ablation study.) Here $e^T p(z_{i,T})$ are constant with respect to the x_i. not affecting the sampling process.

Then the posterior distribution could be

$$\tilde{p}(x_i = k \mid z_i, y_i, \theta^{(l)}) = \frac{\tilde{\xi}^{(l)}(k) p(x_i = k \mid \theta^{(l)})}{\sum_{k=1}^K \tilde{\xi}^{(l)}(k) p(x_i = k \mid \theta^{(l)})}$$

Then we have the algorithm

Algorithm 1 Training

repeat

For i from 1 to n

$t \sim \text{Uniform}(\{1, \ldots, T\})$

Sample ϵ and ϵ^* both from $\mathcal{N}(0, I)$

If $t = 1$, $z_{i,1}^{(l)} = \sqrt{1-\beta_t} y_i + \sqrt{\beta_t}\epsilon^*$

Else $z_{i,t-1}^{(l)} = \sqrt{\bar{\alpha}_{t-1}} y_i + \sqrt{1-\bar{\alpha}_{t-1}}\epsilon, \quad z_{i,t}^{(l)} = \sqrt{1-\beta_t} z_{i,t-1}^{(l)} + \sqrt{\beta_t}\epsilon^*$

Sample $x_i^{(l)}$ from $\tilde{p}(x_i = k \mid z_i, y_i, \theta^{(l)})$

Optimization

$\theta_d^{(l+1)} = \theta_d^{(l)} - \eta \nabla_{\theta_d} \left\| \epsilon - \boldsymbol{\epsilon}_{\theta_d}^{(l)}\left(\sqrt{\bar{\alpha}_t} y_i + \sqrt{1-\bar{\alpha}_t}\epsilon, t, x_i^{(l)}\right)\right\|^2$

End

If we go through all data, instead of a single y_i, we have $\pi_k^{(l+1)} = \dfrac{\sum_{i=1}^n I(x_i^{(l)} = k)}{\sum_{k=1}^K \sum_{i=1}^n I(x_i^{(l)} = k)}$

until converged

Here actually we should update π_k by all samples $y_1, y_2, \cdots, y_n$ or a batch, instead of a single y_i. In real applications, we could fix $p(x_i)$, i.e., uniform prior, then we can use update θ_d based on a randomly selected y_i similar to DDPM.

The sampling algorithm given x is:

Algorithm 2 Sampling (Given x)

$z_T \sim \mathcal{N}(0, I)$
for $t = T, \ldots, 1$ **do**
  $z \sim \mathcal{N}(0, I)$ if $t > 1$, else $z = 0$
  $z_{t-1} = \frac{1}{\sqrt{\alpha_t}} \left(z_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_\theta(z_t, t, x) \right) + \sigma_t z$
end for
return x_0

If x is not given, we can sample it from the prior $p_\theta(x)$.

dataset: mnist, CIFAR10

ablation: always use the last decoder probability to estimate the posterior (or randomly select one t); begin training from the already trained conditional diffusion (fine tuning)? or train from the random initial?

metric: FID, NLL, randomly mask some label and report the accuracy? And some nice reconstructed figures should be reported in the manuscript

2 HMDM (hidden Markov diffusion model)

Suppose we have K states, $S = \{s_1, s_2, \cdots, s_K\}$. The hidden states are $\boldsymbol{X} = (x_1, x_2, \cdots, x_n)$ and the observations are $\boldsymbol{Y} = (y_1, y_2, \cdots, y_n)$. To enhance the model fitting power, we introduce latent variables $\boldsymbol{Z}$ related to the diffusion process. The initial distribution is $\pi_k = p(x_1 = s_k)$ and the transition distribution $A = \{a_{ij}\}_{1 \le i \le K, 1 \le j \le K}$, where $a_{ij} = p(x_l = s_j \mid x_{l-1} = s_i)$, for any l. We define $b_k(y_i, z_i) = p_\theta(y_i, z_i \mid x_i = s_k)$.

The conditional distribution is

$$p_\theta(y_i, z_i \mid x_i) = \left[p_\theta(y_i \mid x_i, z_{i1}) \prod_{t=1}^{T-1} p_\theta(z_{i,t} \mid x_i, z_{i,t+1}) p(z_{i,T}) p_\theta(x_i) \right]$$

The semi-variational distribution is:

$$q(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{Y}, \theta_{(l)}) = p(\boldsymbol{X} \mid \boldsymbol{Z}, \boldsymbol{Y}, \theta^{(l)}) \prod_{i=1}^{n} q(z_i \mid y_i)$$

The joint distribution is:

$$
\begin{aligned}
p_\theta(\boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{Y}) &= p_\theta(x_1) p_\theta(y_1, z_1 \mid x_1) p_\theta(x_2 \mid x_1) p_\theta(y_2, z_2 \mid x_2) \cdots \\
&= p_\theta(x_1) p_\theta(y_1, z_1 \mid x_1) \prod_{i=1}^{n-1} p_\theta(x_{i+1} \mid x_i) p_\theta(y_{i+1}, z_{i+1} \mid x_{i+1})
\end{aligned}
$$

Then the objective function (ELBO) is:

$$
\begin{aligned}
\mathcal{L}(\theta) &= E_{q(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{Y}, \theta^{(l)})} \left(\frac{p_\theta(\boldsymbol{X}, \boldsymbol{Y}, \boldsymbol{Z})}{q(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{Y}, \theta^{(l)})} \right) \\
&= E_{q(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{Y}, \theta^{(l)})} \left(\frac{p_\theta(x_1) p_\theta(y_1, z_1 \mid x_1) \prod_{i=1}^{n-1} p_\theta(x_{i+1} \mid x_i) p_\theta(y_{i+1}, z_{i+1} \mid x_{i+1})}{p(\boldsymbol{X} \mid \boldsymbol{Y}, \boldsymbol{Z}, \theta^{(l)}) \prod_{i=1}^{n} q(z_i \mid y_i)} \right) \\
&= E_{q(\boldsymbol{X}, \boldsymbol{Z} \mid \boldsymbol{Y}, \theta^{(l)})} \left(\sum_{i=1}^{n} \log \frac{p_\theta(y_i, z_i \mid x_i)}{q(z_i \mid y_i)} + \frac{p_\theta(x_1) \prod_{i=1}^{n-1} p_\theta(x_{i+1} \mid x_i)}{p(\boldsymbol{X} \mid \boldsymbol{Y}, \boldsymbol{Z}, \theta^{(l)})} \right) \\
&\propto E_{\substack{\boldsymbol{Z} \sim q(\boldsymbol{Z} \mid \boldsymbol{Y}) \\ \boldsymbol{X} \sim p(\boldsymbol{X} \mid \boldsymbol{Z}, \boldsymbol{Y}, \theta^{(l)})}} \left(\sum_{i=1}^{n} \log \frac{p_\theta(y_i, z_i \mid x_i)}{q(z_i \mid y_i)} + p_\theta(x_1) \prod_{i=1}^{n-1} p_\theta(x_{i+1} \mid x_i) \right) \\
&= \mathcal{L}_d(\theta) + \mathcal{L}_s(\theta)
\end{aligned}
$$

Unlike the mixture of diffusion, HMDM require more complex sampling strategy due to the Markov structure for the latent variables $\boldsymbol{X}$. In order to sample $\boldsymbol{X}$ from the distribution $p(\boldsymbol{X} \mid \boldsymbol{Z}, \boldsymbol{Y} \mid \theta^{(l)})$. Similar to the traditional HMM, we define the forward process:

$$
\alpha_k^{(l)}(i) = p \begin{pmatrix} y_1, & ..., & y_i, & \\ & & & x_i = s_k \mid \theta_{(l)} \\ z_1, & ..., & z_i, & \end{pmatrix}
$$

Then the initial is:

$$
\begin{aligned}
\alpha_k^{(l)}(1) &= p(y_1, z_1, x_1 = s_k \mid \theta^{(l)}) = p_{\theta^{(l)}}(x_1 = s_k) p_{\theta^{(l)}}(y_1, z_1 \mid x_1 = s_k) \\
&= \pi_k^{(l)} b_k^{(l)}(y_1, z_1)
\end{aligned}
$$

The iterative formula is:

$$
\begin{aligned}
\alpha_k^{(l)} &= p\left(\begin{matrix} y_1, & ..., & y_i, \\ z_1, & ..., & z_i, \end{matrix} x_i = s_k \;\middle|\; \theta_{(l)}\right) \\
&= p\left(z_i, y_i \;\middle|\; \begin{matrix} y_1, & ..., & y_{i-1}, \\ z_1, & ..., & z_{i-1}, \end{matrix} x_i = s_k, \theta_{(l)}\right) p\left(\begin{matrix} y_1, & ..., & y_{i-1}, \\ z_1, & ..., & z_{i-1}, \end{matrix} x_i = s_k \;\middle|\; \theta_{(l)}\right) \\
&= b_k^{(l)}(y_i, z_i) \sum_{j=1}^{K} p\left(\begin{matrix} y_1, & ..., & y_{i-1}, \\ z_1, & ..., & z_{i-1}, \end{matrix} x_{i-1} = s_j \;\middle|\; \theta_{(l)}\right) p(x_i = s_k \mid x_{i-1} = s_j, \theta_{(l)}) \\
&= b_k^{(l)}(y_i, z_i) \sum_{j=1}^{K} \alpha_j^{(l)}(i-1) a_{jk}^{(l)}
\end{aligned}
$$

The calculation process follows:

$$\{\alpha_k^{(l)}(1)\}_k \to \{\alpha_k^{(l)}(2)\}_k \to \cdots \to \{\alpha_k^{(l)}(n)\}_k, \quad k = 1, 2, \ldots, K.$$

After finishing the forward process, we do the backward sampling:

$$x_n = s_k \mid \boldsymbol{Z}, \boldsymbol{Y}, \theta^{(l)} \sim p(x_n = s_k \mid \boldsymbol{Z}, \boldsymbol{Y}, \theta^{(l)}) \propto p(x_n = s_k, \boldsymbol{Z}, \boldsymbol{Y} \mid \theta^{(l)}) = \alpha_k^{(l)}(n)$$
$$p(x_n = s_k \mid \boldsymbol{Z}, \boldsymbol{Y}, \theta^{(l)}) = \frac{\alpha_k^{(l)}(n)}{\sum_{k=1}^{K} \alpha_k^{(l)}(n)}$$

Then sample $i = n-1, n-2, \cdots, 1$:

$$
\begin{aligned}
&p(x_i = s_k \mid x_{i+1} = s_j, \boldsymbol{Z}, \boldsymbol{Y}, \theta^{(l)}) \\
&\propto p_{\theta^{(l)}}(y_{1:i}, z_{1:i}, x_i = k) p_{\theta^{(l)}}(x_{i+1} = j \mid x_i = k) p_{\theta^{(l)}}(y_{i+1:n} \mid x_{i+1} = j) \\
&\propto p_{\theta^{(l)}}(y_{1:i}, z_{1:i}, x_i = k) p_{\theta^{(l)}}(x_{i+1} = j \mid x_i = k) \\
&= \alpha_k^{(l)}(i) a_{kj}^{(l)}
\end{aligned}
$$

which is

$$p(x_i = s_k \mid x_{i+1} = s_j, \boldsymbol{Z}, \boldsymbol{Y}, \theta^{(l)}) = \frac{\alpha_k^{(l)}(i) a_{kj}^{(l)}}{\sum_{k=1}^{K} \alpha_k^{(l)}(i) a_{kj}^{(l)}}$$

Then the backward sampling process is

$$x_n \to x_{n-1} \to \cdots \to x_1$$

However, it is complex to calculate $b_k^{(l)}(y_i, z_i)$:

$$\begin{aligned} b_k^{(l)}(y_i, z_i) &= p_{\theta^{(l)}}(y_i, z_i \mid x_i = s_k) \\ &= p_{\theta^{(l)}}(y_i \mid x_i, z_i) \prod_{t=1}^{T-1} p_{\theta^{(l)}}(z_{i,t} \mid x_i, z_{i,t+1}) p(z_{i,T}) \end{aligned}$$

It is time-consuming to go over from 1 to T. We follow the training tricks of DDPM. We sample a time t from the discrete uniform distribution from 1 to T and calculate the denoising probability to approximate $\xi^{(l)}(x_i)$:

$$\tilde{b}_k^{(l)}(y_i, z_i)$$

$$:= \begin{cases} e^T p(z_{i,T}) p_{\theta^{(l)}}(y_i \mid x_i, z_{i1}) = e^T p(z_{i,T}) N\left(y_i; \frac{1}{\sqrt{\alpha_1}}\left(z_{i1} - \frac{\beta_1}{\sqrt{1-\bar{\alpha}_1}} \epsilon_{\theta^{(l)}}\left(z_{i1}, 1, x_i\right)\right), \sigma_1^2\right) & t = 1 \\ e^T p(z_{i,T}) p_{\theta^{(l)}}(z_{i,t-1} \mid x_i, z_{i,t}) = e^T p(z_{i,T}) N\left(z_{i,t-1}; \frac{1}{\sqrt{\alpha_1}}\left(z_{it} - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}} \epsilon_{\theta^{(l)}}\left(z_{it}, t, x_i\right)\right), \sigma_t^2\right) & t = 2, 3, \cdots, T \end{cases}$$

Then we have the algorithm

If the initial distribution is fixed, we don't need to go over all data to estimate the initial distribution at each iteration. Similar to DDPM, we can randomly select one sequence of data and do the optimization and update $\boldsymbol{A}$ based on the predicted states of this sequence.

After training, we can predict the hidden states by the Viterbi algorithm. It is similar to the backward sampling, but a little different. The backward sampling is sampling, the viterbi algorithm is finding the maximum.

$$x_n^* = \arg\max_k p(x_n = s_k \mid \boldsymbol{Z}, \boldsymbol{Y}, \theta^{(l)}) = \arg\max_k \alpha_k(n)$$

Then sample $i = n-1, n-2, \cdots, 1$:

$$x_i^* = \arg\max_k p(x_i = s_k \mid x_{i+1} = s_j, \boldsymbol{Z}, \boldsymbol{Y}, \theta^{(l)}) = \arg\max_k \alpha_k(i) a_{kj}$$

where $\alpha_k(i)$ is approximated by $\tilde{\alpha}_k(i)$.

Experiment:

dataset: mnist, CIFAR10

ablation: always use the last decoder probability to estimate the posterior (or randomly select one t); begin training from the already trained conditional diffusion (fine tuning)? or train from the random initial?

metric: FID, NLL, randomly mask some label and report the accuracy? And some nice reconstructed figures should be reported in the manuscript

Algorithm 3 Diffusion-HMM Training with Posterior Sampling

repeat
  for $j = 1$ to N **do** ▷ Or iterate over a batch
    Select the j-th sequence $(y_1, y_2, \ldots, y_n)$
    Sample $t \sim \text{Uniform}(\{1, \ldots, T\})$
    Sample $\epsilon, \epsilon^* \sim \mathcal{N}(0, I)$
    if $t = 1$ **then**
      for $i = 1$ to n **do**
        $z_{i,1}^{(l)} = \sqrt{1 - \beta_t} \cdot y_i + \sqrt{\beta_t} \cdot \epsilon^*$
      end for
    else
      for $i = 1$ to n **do**
        $z_{i,t-1}^{(l)} = \sqrt{\bar{\alpha}_{t-1}} \cdot y_i + \sqrt{1 - \bar{\alpha}_{t-1}} \cdot \epsilon$
        $z_{i,t}^{(l)} = \sqrt{1 - \beta_t} \cdot z_{i,t-1}^{(l)} + \sqrt{\beta_t} \cdot \epsilon^*$
      end for
    end if
    for $i = 1$ to n **do**
      Compute emission probability: $\tilde{b}_k^{(l)}(y_i, z_i)$
      Compute forward message:

$$\tilde{\alpha}_k^{(l)}(i) = \tilde{b}_k^{(l)}(y_i, z_i) \cdot \sum_{j=1}^{K} \tilde{\alpha}_j^{(l)}(i-1) \cdot a_{jk}^{(l)}$$

    end for
    for $i = n$ to 1 **do**
      Sample hidden state $x_i^{(l)}$ via backward sampling
    end for
    Update decoder parameters:

$$\theta_d^{(l+1)} = \theta_d^{(l)} - \eta \cdot \nabla_{\theta_d} \left\| \epsilon - \boldsymbol{\epsilon}_{\theta_d}^{(l)} \left(\sqrt{\bar{\alpha}_t} y_i + \sqrt{1 - \bar{\alpha}_t} \epsilon, t, x_i^{(l)} \right) \right\|^2$$

  end for
  Update HMM parameters (initial π_k and transition a_{jk}) using sampled $\{x_i^{(l)}\}$ from all sequences
until converged

3 Discussion

enhance the tradition statistical modeling by introducing diffusion latent variable.