Neural Discrete Representation Learning

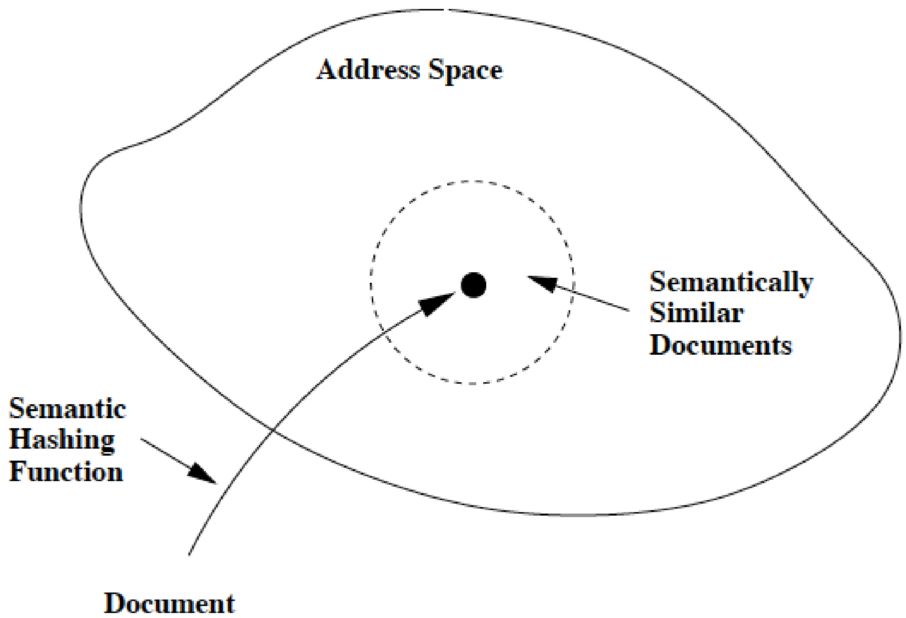
Van den Oord et al.

Park Jungsoo

Data Mining & Information Systems Lab.
Department of Computer Science and Engineering,
College of Informatics, Korea University

1 Introduction

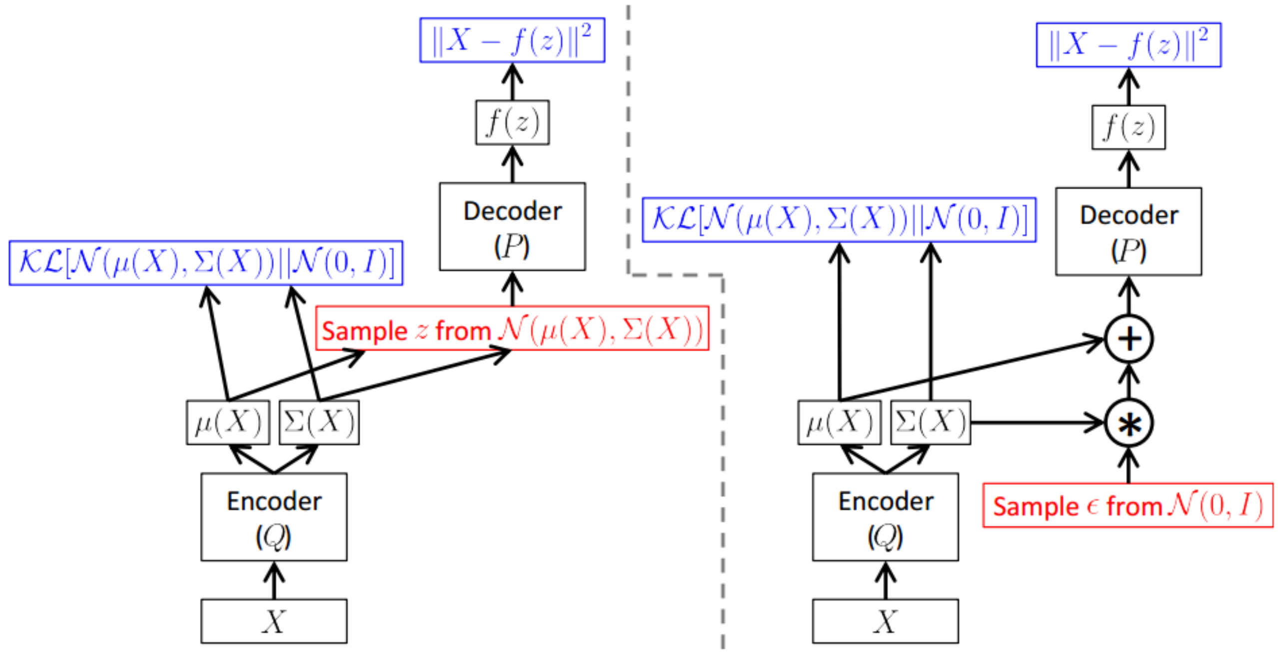
Why Discrete Latent Representation?



- Computational Efficiency
- Interpretability and Communication
- More Natural

1 Introduction

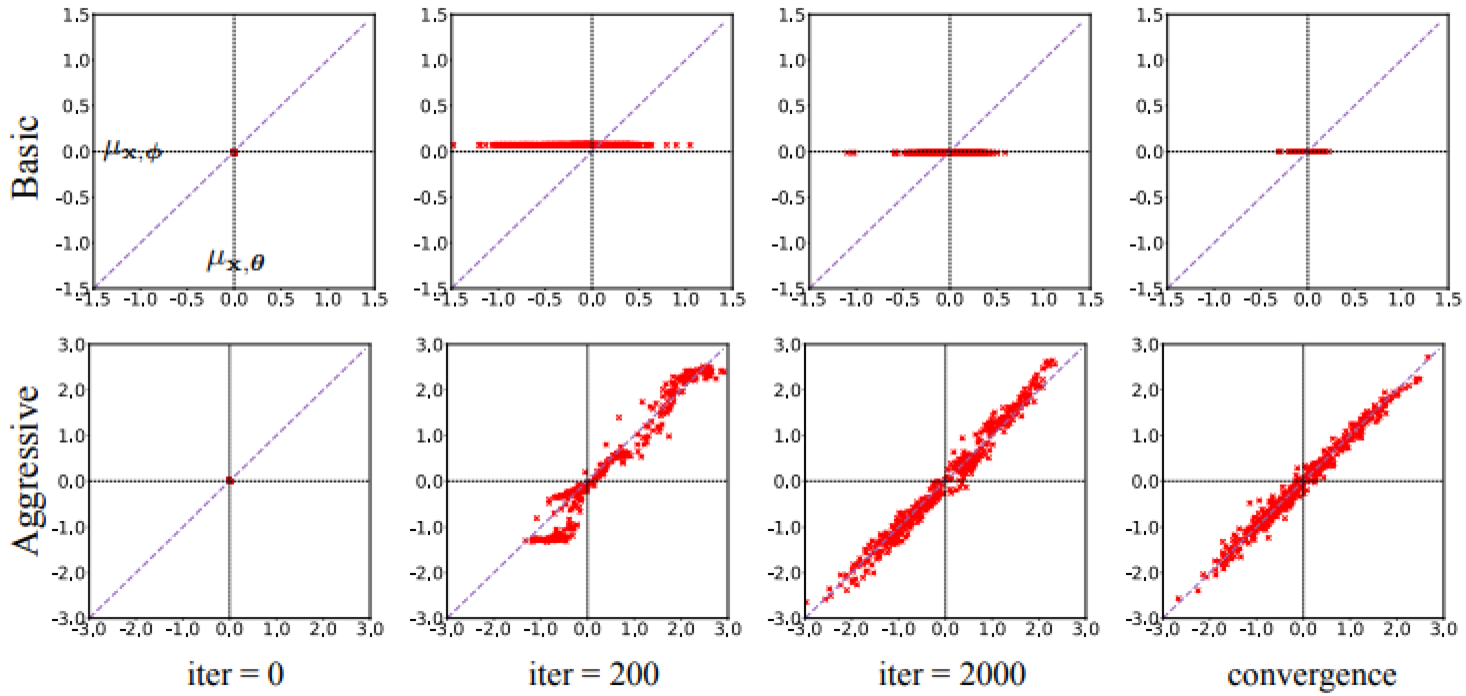
Auto-Encoding Variational Bayes



- Maximizing ELBO
- Ancestral sampling from Standard Normal D.

1 Introduction

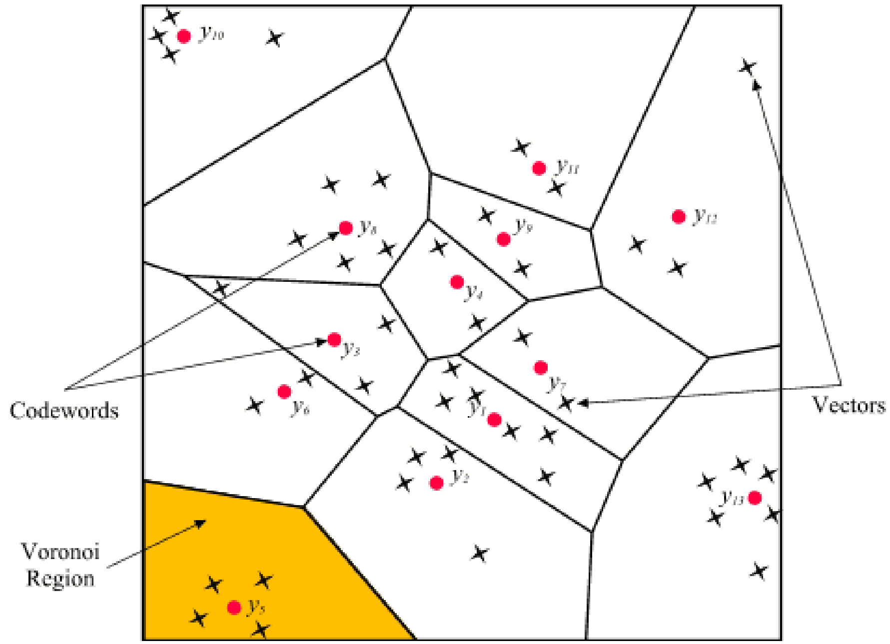
Posterior Collapse



- With strong autoregressive decoder, posterior collapse happens
- Not capturing meaningful representation

1 Introduction

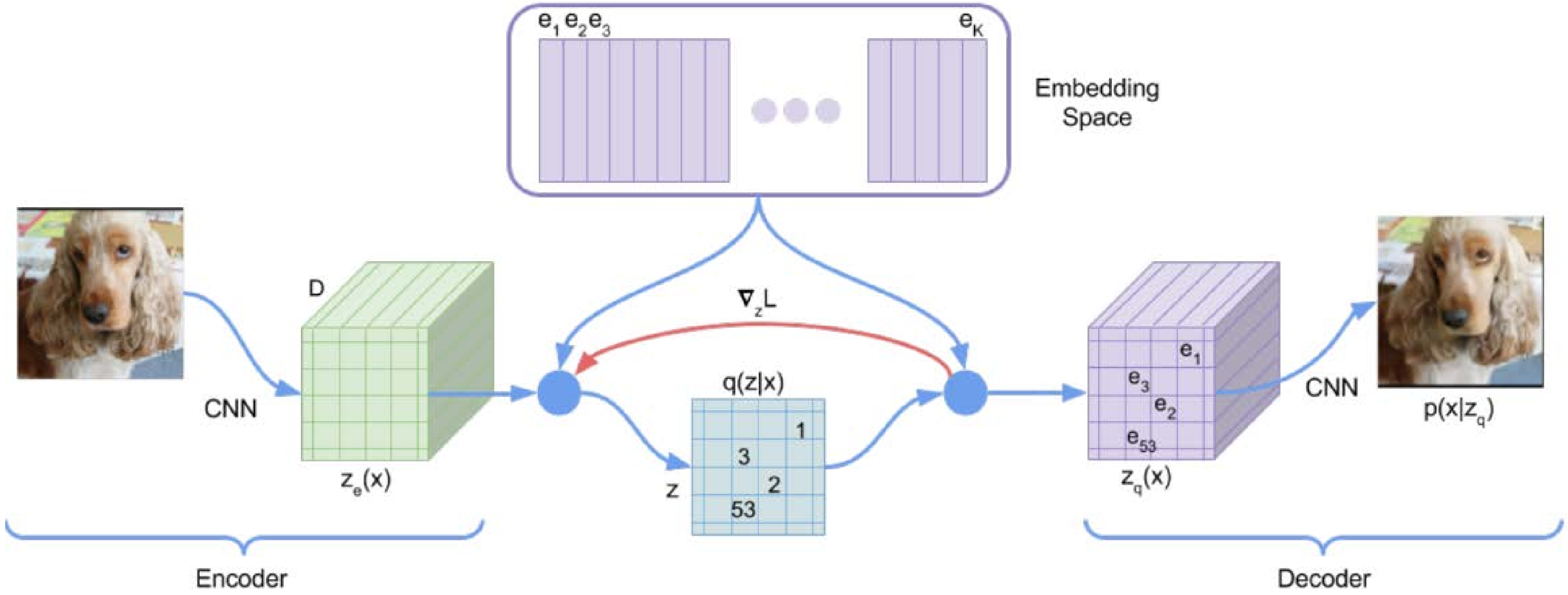
Vector Quantization



- Quantization technique for modeling probability density function by distribution of prototype vectors
- Originally used in lossy data compression

2 Method

Overview



Categorical Distribution

- Posterior and Prior are categorical distributions.

$$L = \log p(x|z_q(x)) + \|\text{sg}[z_e(x)] - e\|_2^2 + \beta\|z_e(x) - \text{sg}[e]\|_2^2,$$

$$q(z = k|x) = \begin{cases} 1 & \text{for } \mathrm{k} = \text{argmin}_j \|z_e(x) - e_j\|_2, \\ 0 & \text{otherwise} \end{cases},$$

$$z_q(x) = e_k, \quad \text{where} \quad k = \text{argmin}_j \|z_e(x) - e_j\|_2$$

Resolving Posterior Collapse

- As for vanilla VAE,

$$D_{KL}((q_{\phi}(\mathbf{z})||p_{\boldsymbol{\theta}}(\mathbf{z})) = \int q_{\boldsymbol{\theta}}(\mathbf{z}) \left(\log p_{\boldsymbol{\theta}}(\mathbf{z}) - \log q_{\boldsymbol{\theta}}(\mathbf{z})\right) \, d\mathbf{z}$$
$$= \frac{1}{2} \sum_{j=1}^{J} \left(1 + \log((\sigma_j)^2) - (\mu_j)^2 - (\sigma_j)^2\right)$$

- As for VQ VAE,

$$D_{\mathrm{KL}}(P \parallel Q) = \sum_{x \in \mathcal{X}} P(x) \log\left(\frac{P(x)}{Q(x)}\right).$$

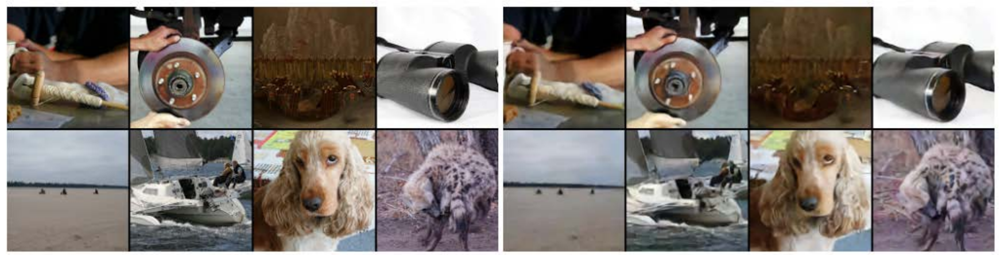
For specific k, p(x), posterior is 1 and 0 otherwise, therefore yielding

$$\log K$$

Thus, posterior collapse(KL term -> 0) problem doesn't happen

Autoregressive Prior

- After training with uniform prior, autoregressive model is fit for learning prior distribution, thus generating more realistic images



uniform

autoregressive

Generating Diverse High-Fidelity Images with VQ-VAE-2

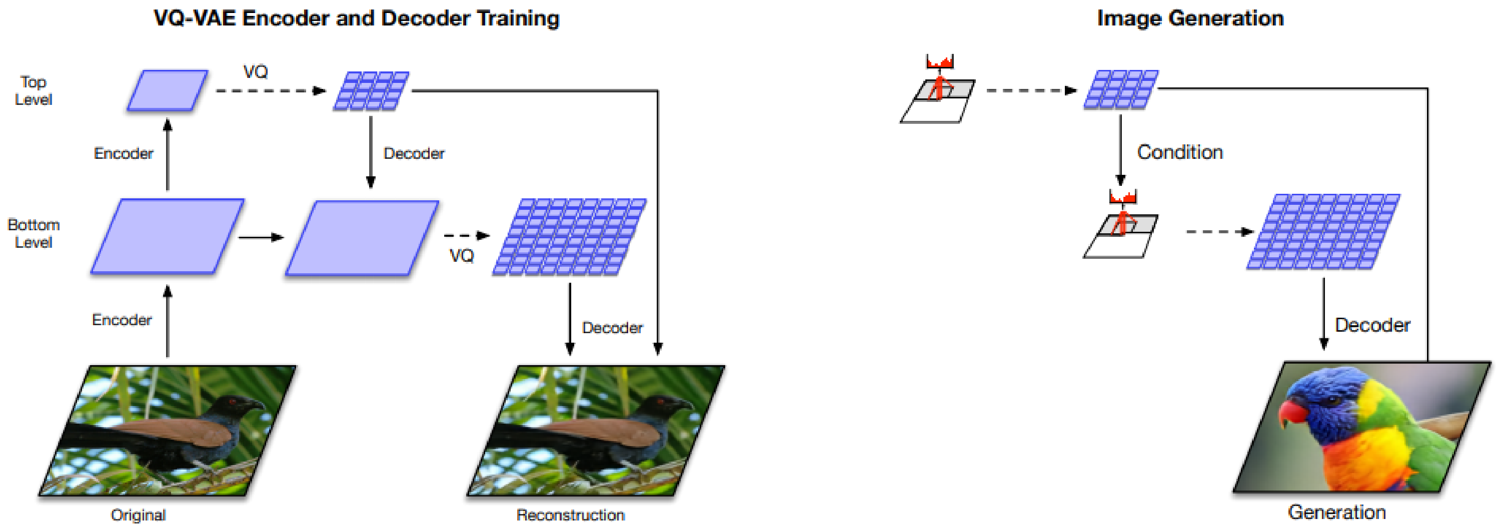
Ali Razavi et al.

Park Jungsoo

Data Mining & Information Systems Lab.
Department of Computer Science and Engineering,
College of Informatics, Korea University

2 Method

Overview



- Top CB for global structure, Bottom CB for the details

Algorithm

Algorithm 1 VQ-VAE training (stage 1)

Require: Functions E_{top} , E_{bottom}, D, $\mathbf{x}$ (batch of training images)

$\mathbf{h}_{top} \leftarrow E_{top}(\mathbf{x})$

▷ quantize with top codebook eq 1

$\mathbf{e}_{top} \leftarrow Quantize(\mathbf{h}_{top})$

$\mathbf{h}_{bottom} \leftarrow E_{bottom}(\mathbf{x}, \mathbf{e}_{top})$

▷ quantize with bottom codebook eq 1

$\mathbf{e}_{bottom} \leftarrow Quantize(\mathbf{h}_{bottom})$

$\hat{\mathbf{x}} \leftarrow D(\mathbf{e}_{top}, \mathbf{e}_{bottom})$

▷ Loss according to eq 2

$\theta \leftarrow Update(\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}))$

Algorithm 2 Prior training (stage 2)

$\mathbf{T}_{top}, \mathbf{T}_{bottom} \leftarrow \emptyset$ ▷ training set

for $\mathbf{x} \in$ training set **do**

  $\mathbf{e}_{top} \leftarrow Quantize(E_{top}(\mathbf{x}))$

  $\mathbf{e}_{bottom} \leftarrow Quantize(E_{bottom}(\mathbf{x}, \mathbf{e}_{top}))$

  $\mathbf{T}_{top} \leftarrow \mathbf{T}_{top} \cup \mathbf{e}_{top}$

  $\mathbf{T}_{bottom} \leftarrow \mathbf{T}_{bottom} \cup \mathbf{e}_{bottom}$

end for

$p_{top} = \texttt{TrainPixelCNN}(\mathbf{T}_{top})$

$p_{bottom} = \texttt{TrainCondPixelCNN}(\mathbf{T}_{bottom}, \mathbf{T}_{top})$

▷ Sampling procedure

while true **do**

  $\mathbf{e}_{top} \sim p_{top}$

  $\mathbf{e}_{bottom} \sim p_{bottom}(\mathbf{e}_{top})$

  $\mathbf{x} \leftarrow D(\mathbf{e}_{top}, \mathbf{e}_{bottom})$

end while

3 Experiment

Effect of Hierarchical Latent Representation

h_{top} $h_{\text{top}}, h_{\text{middle}}$ $h_{\text{top}}, h_{\text{middle}}, h_{\text{bottom}}$ Original

Figure 3: Reconstructions from a hierarchical VQ-VAE with three latent maps (top, middle, bottom). The rightmost image is the original. Each latent map adds extra detail to the reconstruction. These latent maps are approximately 3072x, 768x, 192x times smaller than the original image (respectively).

Fast Decoding in Sequence Models Using Discrete Latent Variables

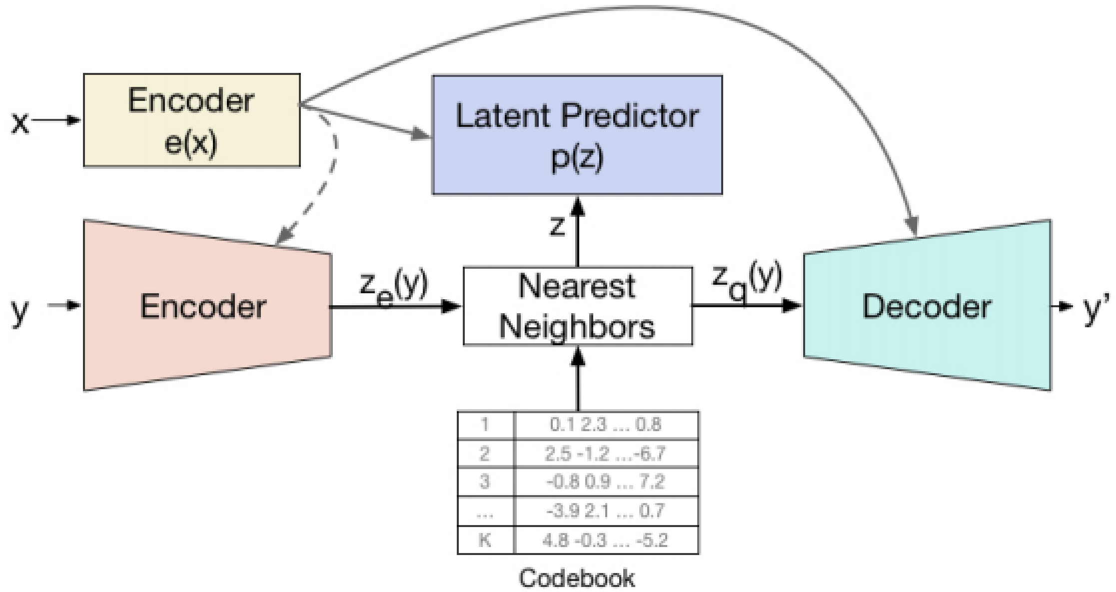
Lukasz Kaiser et al.

Park Jungsoo

Data Mining & Information Systems Lab.
Department of Computer Science and Engineering,
College of Informatics, Korea University

1 Method

Machine Translation



- Encoder function is a series of convolutional layers with residual connections
- Source sentence is encoded in to sequence of hidden states through multiple causal self-attention layers
- Decoder consists of transpose convolutional layers whose output is fed to a transformer decoder with causal attention.

Machine Translation

Model	n_c	n_s	BLEU	Latency	Speedup
Autoregressive Model (beam size=4)	-	-	28.1	331 ms	1×
Autoregressive Baseline (no beam-search)	-	-	27.0	265 ms	1.25×
NAT + distillation	-	-	17.7	39 ms	15.6× *
NAT + distillation + NPD=10	-	-	18.7	79 ms	7.68× *
NAT + distillation + NPD=100	-	-	19.2	257 ms	2.36× *
LT + Semhash	-	-	19.8	105 ms	3.15×
Our Results					
VQ-VAE	3	-	21.4	81 ms	4.08×
VQ-VAE with EM	3	5	22.4	81 ms	4.08×
VQ-VAE + distillation	3	-	26.4	81 ms	4.08×
VQ-VAE with EM + distillation	3	10	**26.7**	81 ms	4.08×
VQ-VAE with EM + distillation	4	10	25.4	58 ms	5.71×