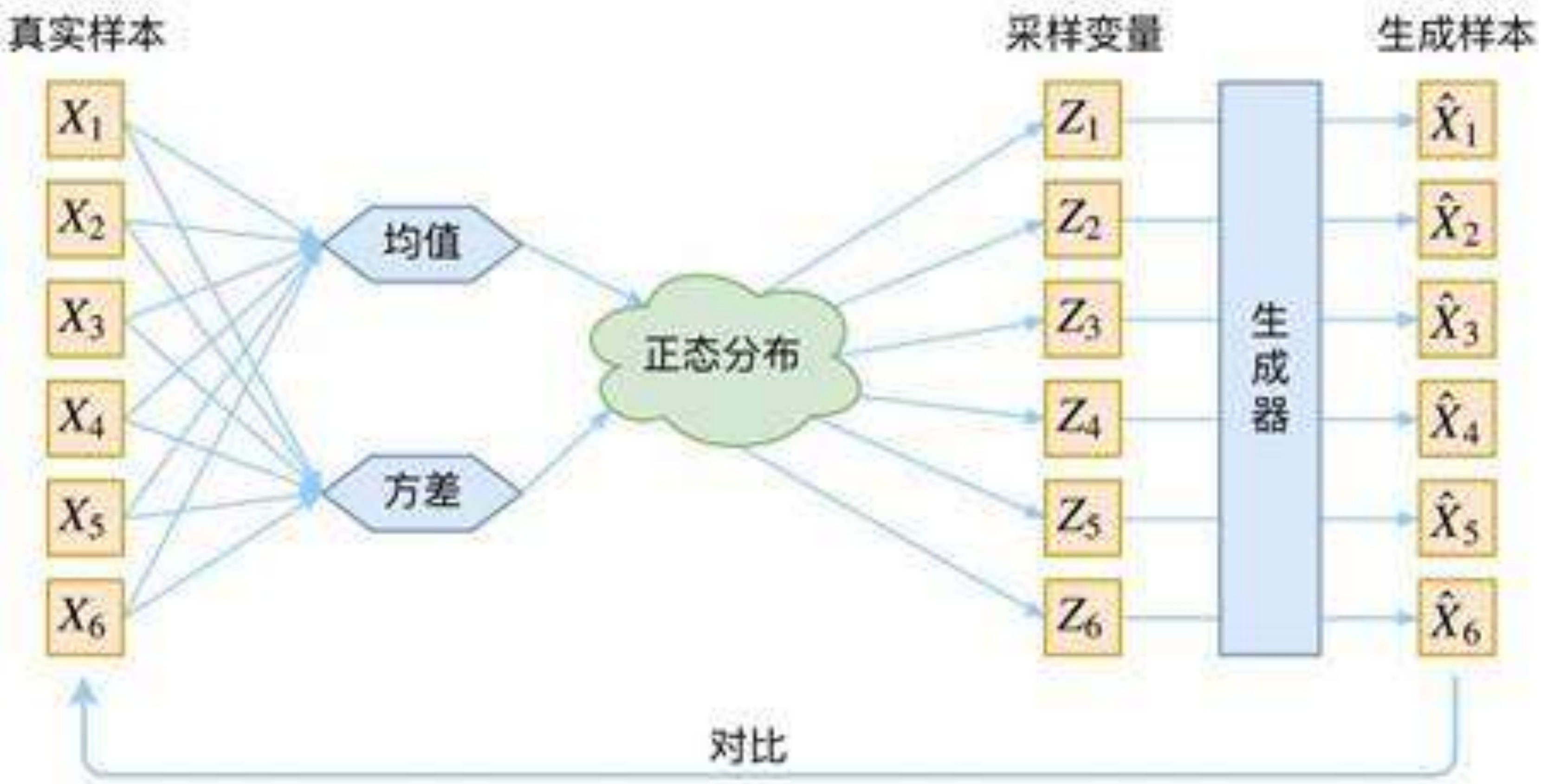
Neural Discrete Representation Learning (VQ-VAE)

VAE

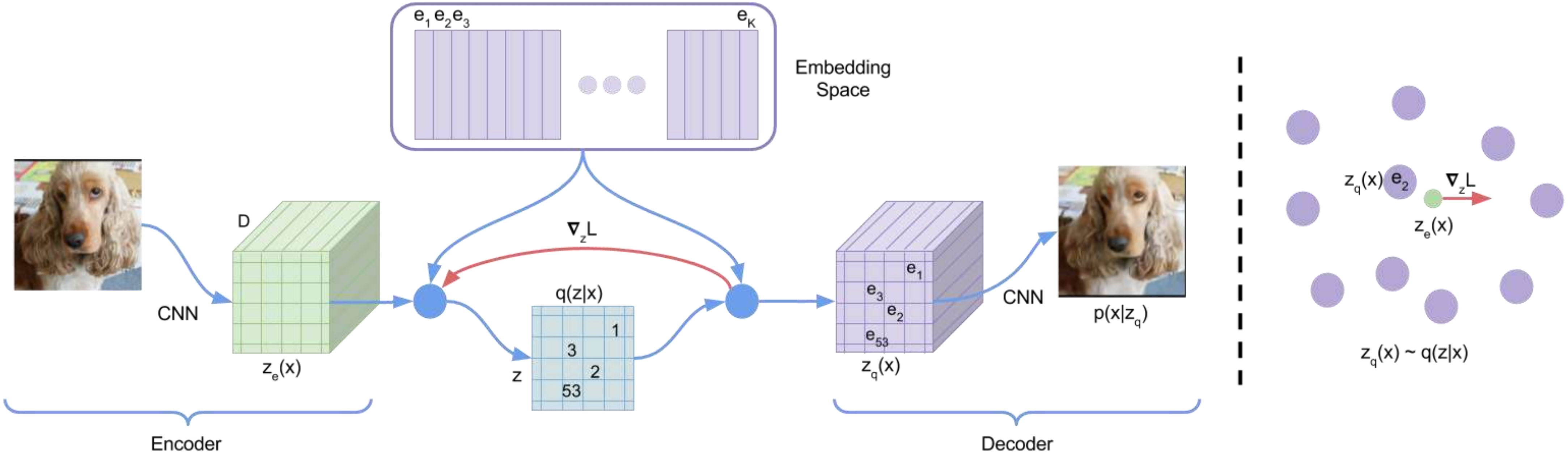


真实样本
X_1
X_2
X_3
X_4
X_5
X_6
均值
方差
正态分布
采样变量
Z_1
Z_2
Z_3
Z_4
Z_5
Z_6
生成器
生成样本
$\hat{X}_1$
$\hat{X}_2$
$\hat{X}_3$
$\hat{X}_4$
$\hat{X}_5$
$\hat{X}_6$
对比

Why do we quantize?

- Discrete representations are potentially a more natural fit for many of the modalities (Audio, Visions and Texts)
- Data compression
- Tokenization

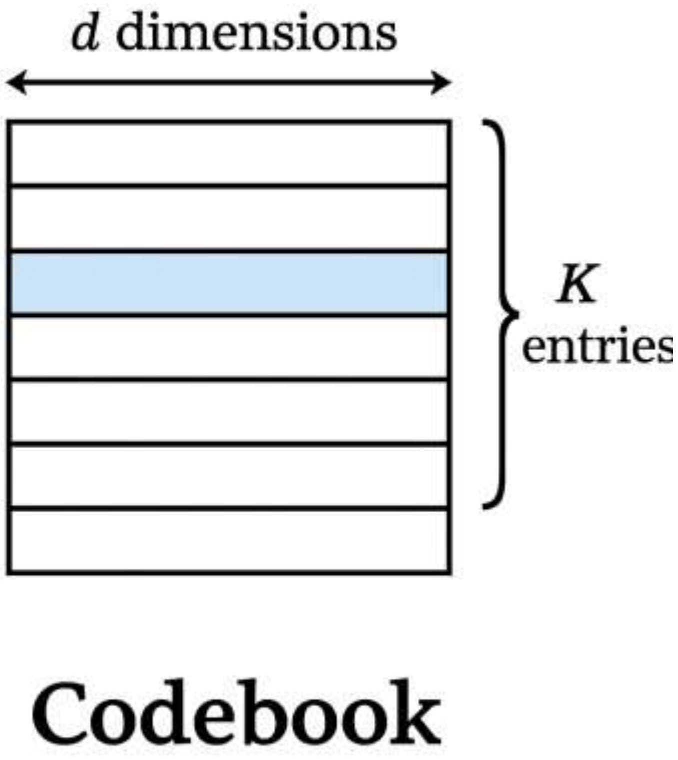
VQ-VAE



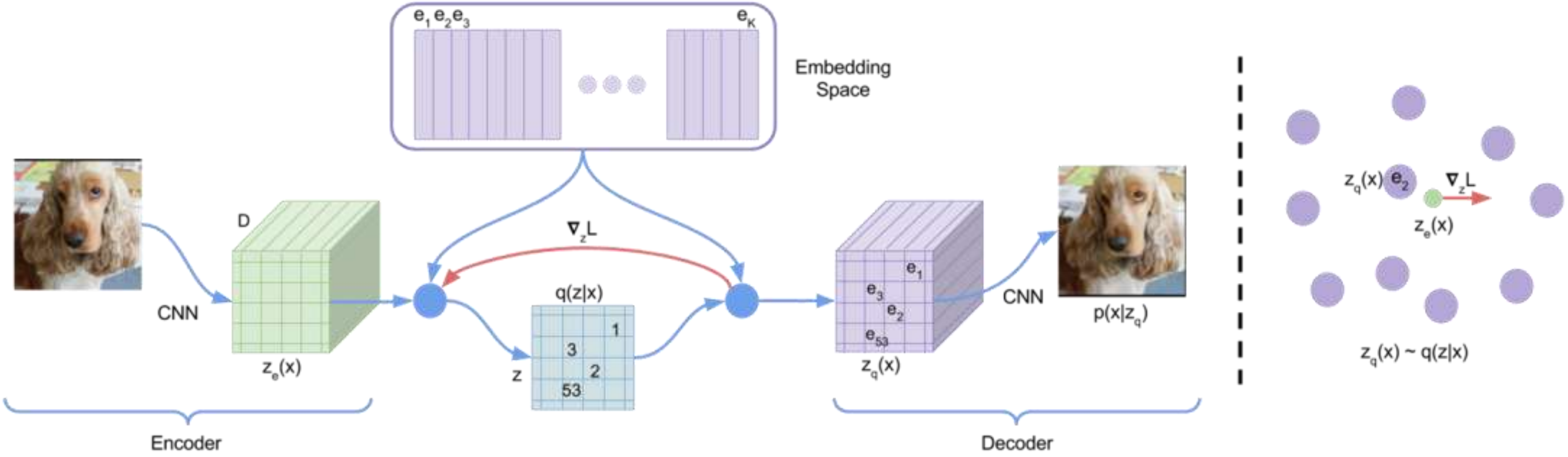
e1 e2 e3
eK
Embedding
Space
D
CNN
ze(x)
∇zL
q(z|x)
1
3
2
53
z
e1
e3
e2
e53
zq(x)
CNN
p(x|zq)
Encoder
Decoder
zq(x) e2
∇zL
ze(x)
zq(x) ~ q(z|x)

How do we quantize?

- Map the encoder output z_e into an entry e_i of the $K \times d$ codebook (similar to K-means)
  - Calculate the distance between z_e and e_i
  - Choose the nearest entry as the input for decoder



Straight Through Estimator



Loss Function

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

Reconstruction Loss

Codebook Loss

Commitment Loss

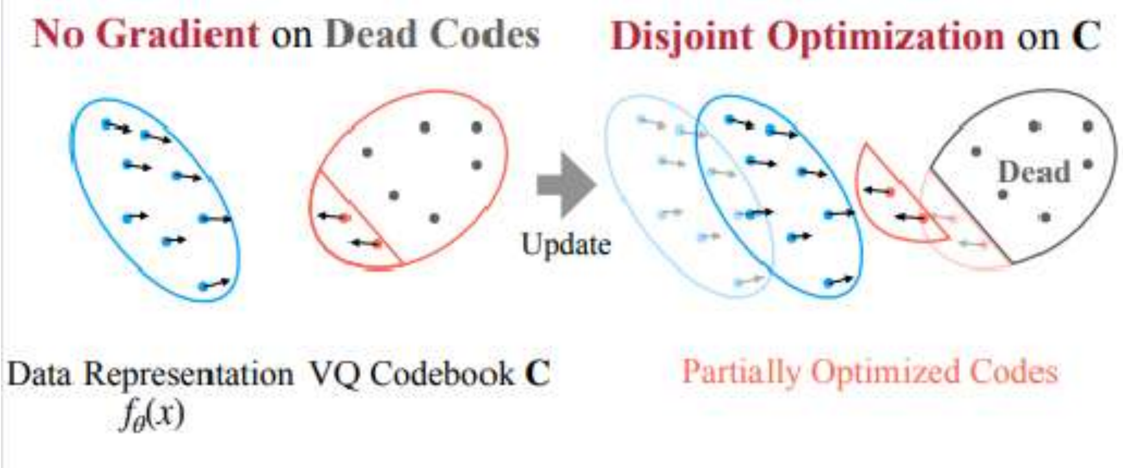
The resulting algorithm to be quite robust to β, as the results did not vary for values of β ranging from 0.1 to 2.0.

Representation Collapse

- For an entry e_i of the codebook,

$$\frac{\partial L}{\partial e_i} = \begin{cases} \text{not } 0, & if\ e_i\ is\ chosen \\ 0, & if\ e_i\ is\ not\ chosen \end{cases}$$

- With e_i and z_e closer and closer, only a small subset of the codebook entries will be updated.



ADDRESSING REPRESENTATION COLLAPSE IN VECTOR QUANTIZED MODELS WITH ONE LINEAR LAYER (simVQ)

Representation Collapse

- Representation Collapse:
  - the contradiction between codebook expansion and low codebook utilization in VQ models where increasing the codebook size fails to improve the performance.
  - the disjoint optimization process that updates only a subset of codebook vectors

SimVQ

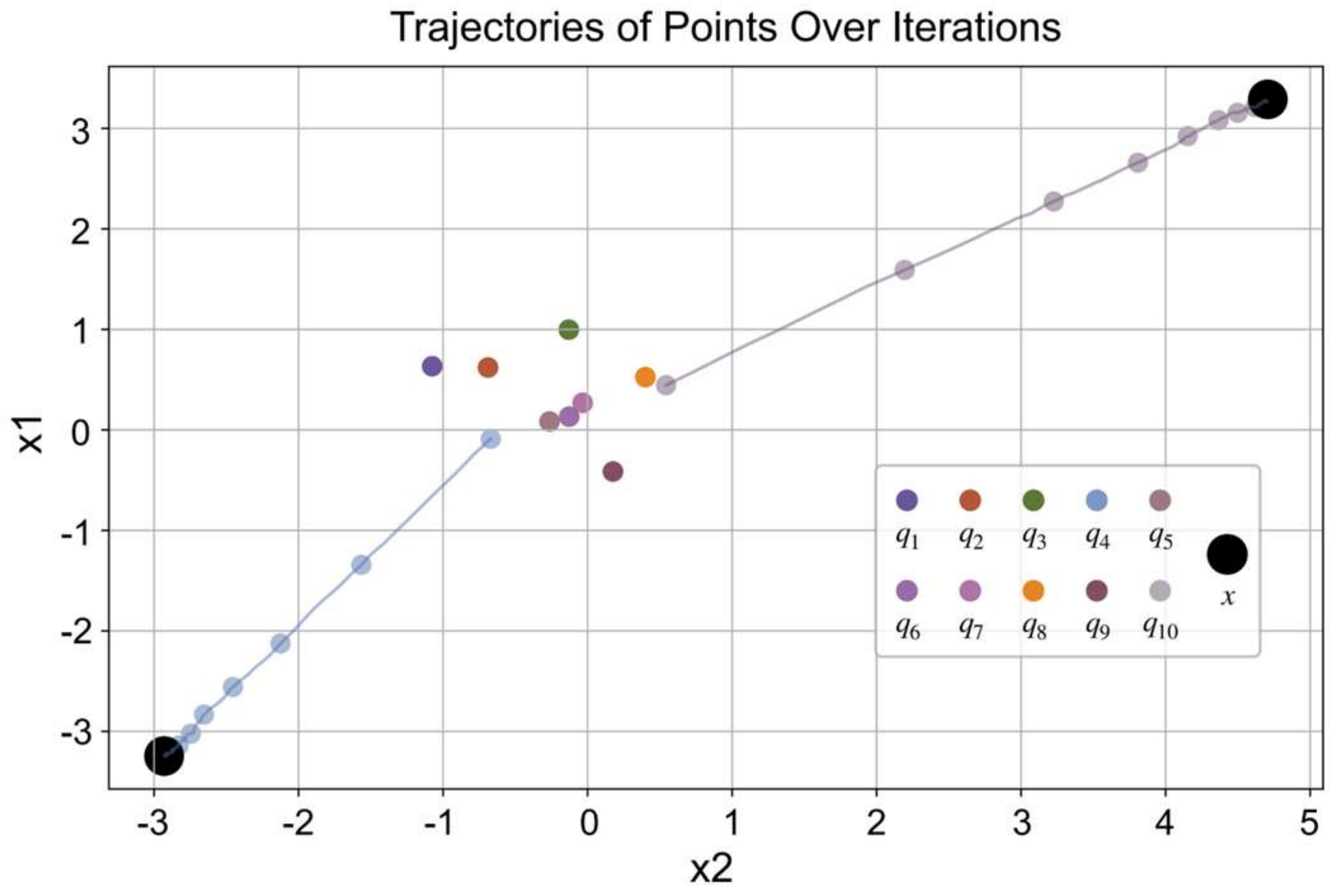
- Reparameterized codebook C with $W\pi$, with $W = (w_1, w_2, \ldots, w_d)$ and $\pi = (\pi_1, \pi_2, \ldots, \pi_K)$
  - the optimization of the reparameterized codebook can be divided into three scenarios:
    1. **Updating $\boldsymbol{\pi}$ with $\boldsymbol{W}$ frozen**: The vanilla VQ is a special case of this scenario with $W = I$.
    2. **Updating $\boldsymbol{W}$ with $\boldsymbol{\pi}$ frozen:** The entire codebook $W\pi$ adjusts to the latent distribution of z_e. The basis matrix W rotates and stretches the codebook space.
    3. **Updating both $\boldsymbol{W}$ and $\boldsymbol{\pi}$:** The selected subset of codes moves towards z_e while the space spanned by $\boldsymbol{W}$ undergoes simultaneous rotation and stretching.

Scenario 1: Updating $\boldsymbol{\pi}$ with $\boldsymbol{W}$ frozen

- For one of the entries $e_i = W\pi_i$,

$$\frac{\partial L}{\partial \pi_i} = \frac{\partial L}{\partial e_i}\frac{\partial e_i}{\partial \pi_i} = \frac{\partial L}{\partial e_i} W^T$$

Scenario 1: Updating $\boldsymbol{\pi}$ with $\boldsymbol{W}$ frozen



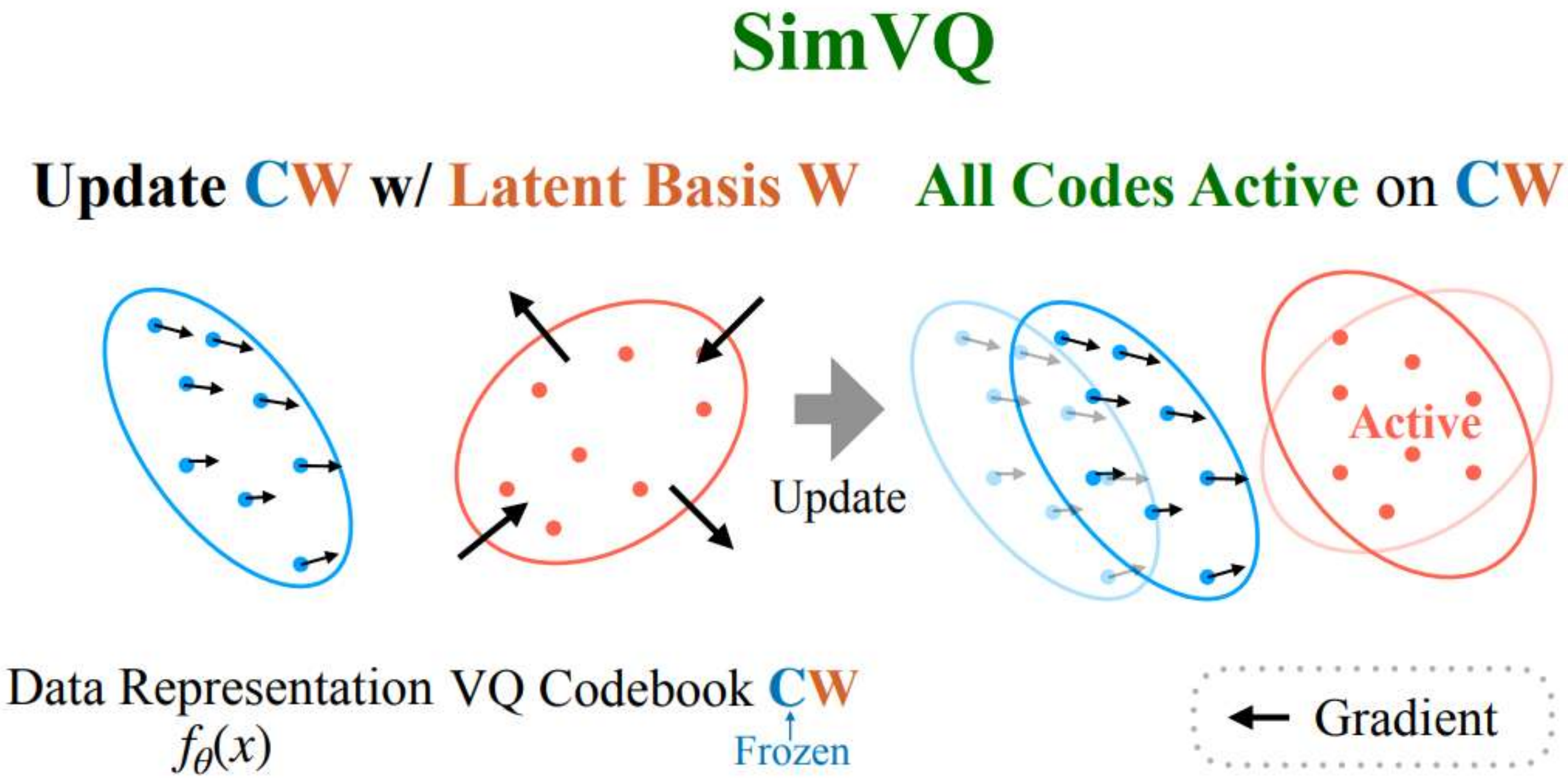
Scenario 2: Updating $\boldsymbol{W}$ with $\boldsymbol{\pi}$ frozen

- For one of the entries $e_i = W\pi_i$,

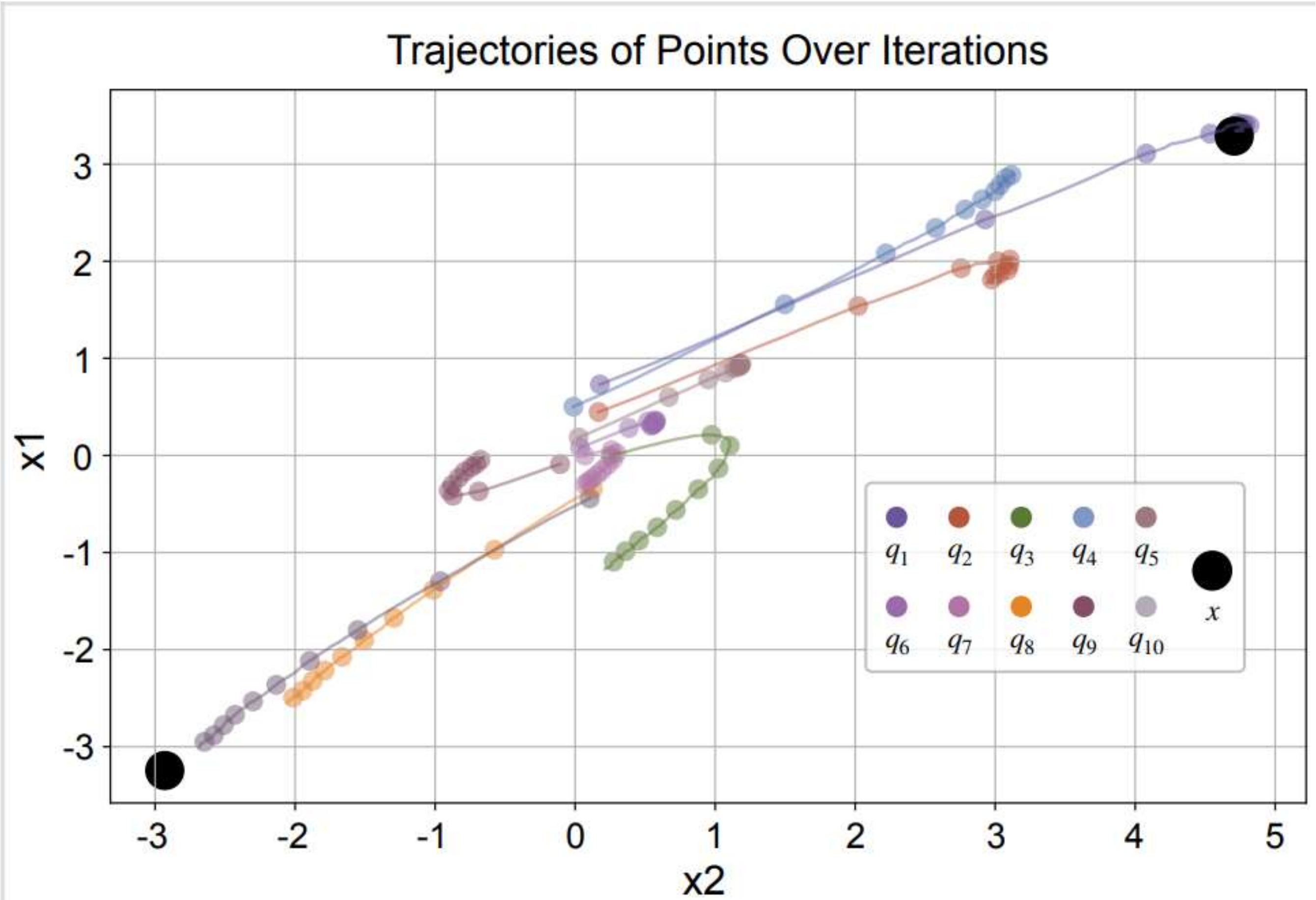
$$\frac{\partial L}{\partial W} = \sum_j \frac{\partial L}{\partial e_j}\frac{\partial e_j}{\partial W} = \sum_j \frac{\partial L}{\partial e_j}\pi_j^T$$

$$e_i^{(t+1)} = W^{(t+1)}\pi_i = \left(W^{(t)} - \eta\frac{\partial L}{\partial W^{(t)}}\right)\pi_i = e_i^{(t)} - \eta\frac{\partial L}{\partial W^{(t)}}\pi_i$$

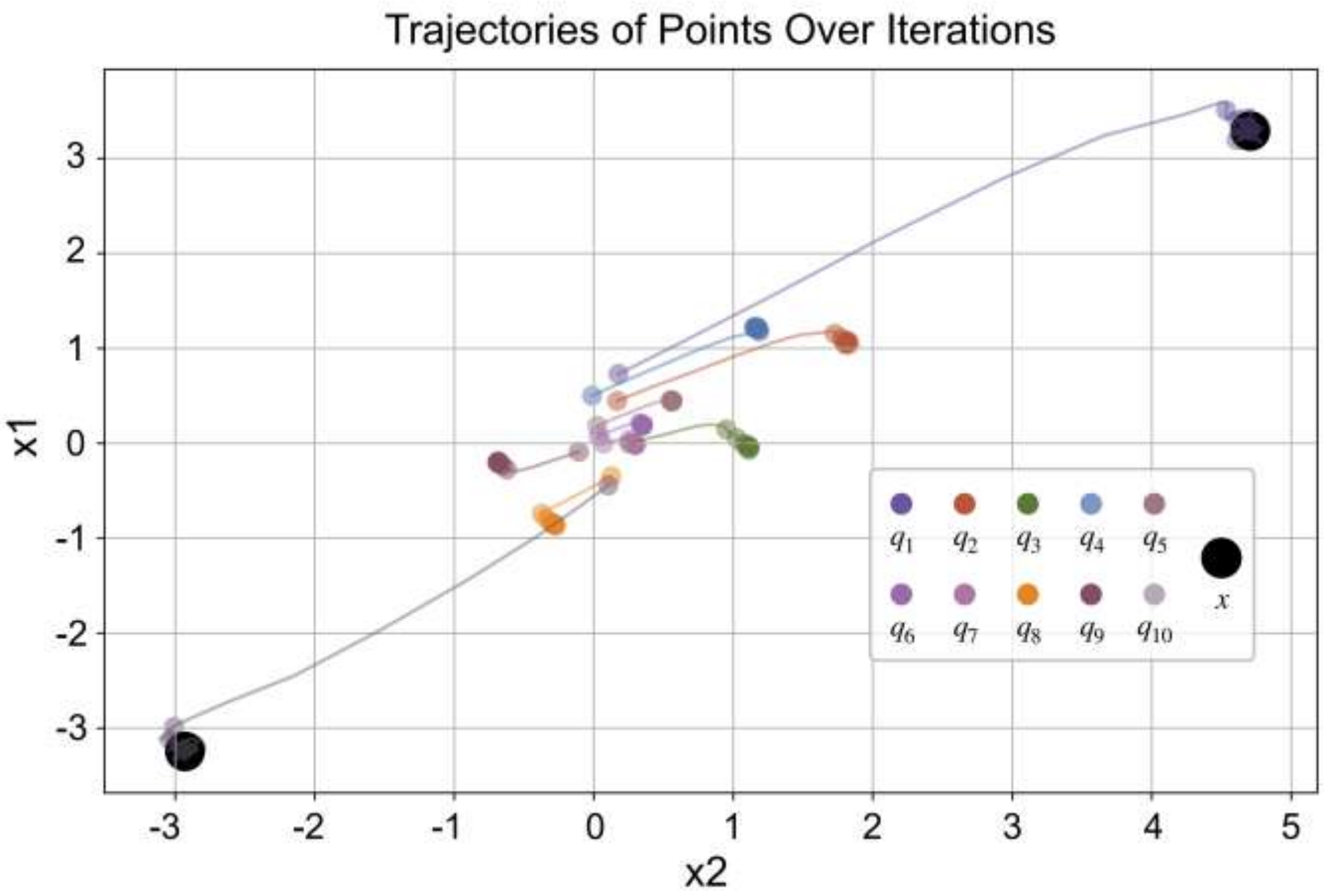
Scenario 2: Updating W with π frozen



Scenario 2: Updating $\boldsymbol{W}$ with $\boldsymbol{\pi}$ frozen



Scenario 3: Updating both W and π



Loss Curve

Reconstruction MSE Loss over iteration
0.105
0.1
0.095
0.09
0.085
0.08
0.075
0
20k
40k
60k
80k
100k
120k
140k
160k
180k
200k
220k
240k
VQGAN
LFQ
FSQ
VQGAN-FC
VQGAN-LC
SimVQ w/ C unfrozen
SimVQ-65536
SimVQ-262144
SimVQ-262144

Experiments

Table 1: Reconstruction performance on ImageNet-1k with a resolution of 128×128. All models are trained using images downsampled into 16×16 tokens. † Results are reproduced using the codebook size of $[8, 8, 8, 5, 5, 5]$ to approximately match $65, 536$. + Following VQGAN-LC, we extract CLIP features with the codebook frozen.

Method	Latent dim	Codebook size	Util↑	rFID↓	LPIPS↓	PSNR↑	SSIM↑
VQGAN (Esser et al., 2021)	128	65,536	1.4%	3.74	0.17	22.20	70.6
VQGAN-EMA (Razavi et al., 2019)	128	65,536	4.5%	3.23	0.15	22.89	72.3
VQGAN-FC (Yu et al., 2022a)	128	65,536	1.4%	5.33	0.18	21.45	68.8
VQGAN-FC (Yu et al., 2022a)	8	65,536	100.0%	2.63	0.13	23.79	77.5
FSQ† (Mentzer et al., 2024)	16	64,000	100.0%	2.80	0.13	23.63	75.8
LFQ (Yu et al., 2024)	6	65,536	100.0%	2.88	0.13	23.60	77.2
VQGAN-LC-CLIP+ (Zhu et al., 2024a)	768	65,536	100.0%	2.40	0.13	23.98	77.3
SimVQ (ours)	128	65,536	100.0%	**2.24**	**0.12**	**24.15**	**78.4**
SimVQ (ours)	128	262,144	100.0%	**1.99**	**0.11**	**24.68**	**80.3**

Ablation Study on the codebook sizes

Table 2: Ablation study on the effect of various codebook sizes on ImageNet at a resolution of 128×128. † We directly copy the reported results of VQGAN-LC from the original paper on ImageNet 256×256 resolution.

Method	Codebook Size	Util↑	rFID↓	LPIPS↓	PSNR↑	SSIM↑
VQGAN-LC-CLIP†	50,000	99.9%	2.75	0.13	23.8	58.4
VQGAN-LC-CLIP†	100,000	99.9%	<u>2.62</u>	0.12	23.8	58.9
VQGAN-LC-CLIP†	200,000	99.8%	<u>2.66</u>	0.12	23.9	59.2
SimVQ	1,024	100.0%	3.67	0.16	22.34	70.8
SimVQ	8,192	100.0%	2.98	0.14	23.23	74.7
SimVQ	65,536	100.0%	2.24	0.12	24.15	78.4
SimVQ	262,144	100.0%	**1.99**	**0.11**	**24.68**	**80.3**

Ablation Study on the codebook optimization

Table 3: Ablation study of codebook optimization.

Initialization	Trainable	Util↑	rFID↓	LPIPS↓	PSNR↑	SSIM↑
Gaussian	Yes	100.0%	2.31	0.12	24.04	77.2
Uniform	No	100.0%	2.31	0.12	24.15	78.4
Gaussian	No	100.0%	**2.24**	**0.12**	**24.15**	**78.4**