Differentiable Vector Quantization for Rate-Distortion Optimization of Generative Image Compression

Shiyin Jiang, Wei Long, Minghao Han,

Zhenghao Chen, Ce Zhu, Shuhang Gu

https://github.com/CVL-UESTC/RDVQ

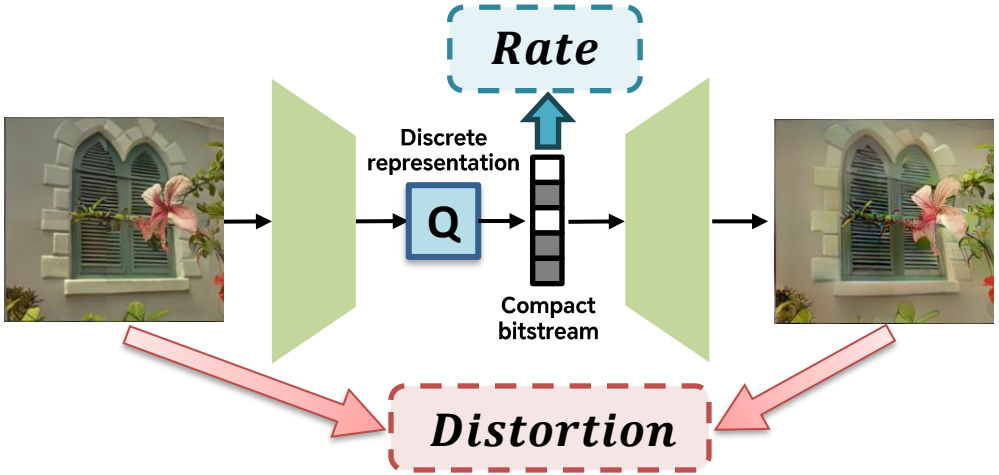
The University of Electronic Science and Technology of China (UESTC)

Computer Vison Lab (CVL-UESTC)

Image Compression: Fewer Bits, Less Distortion

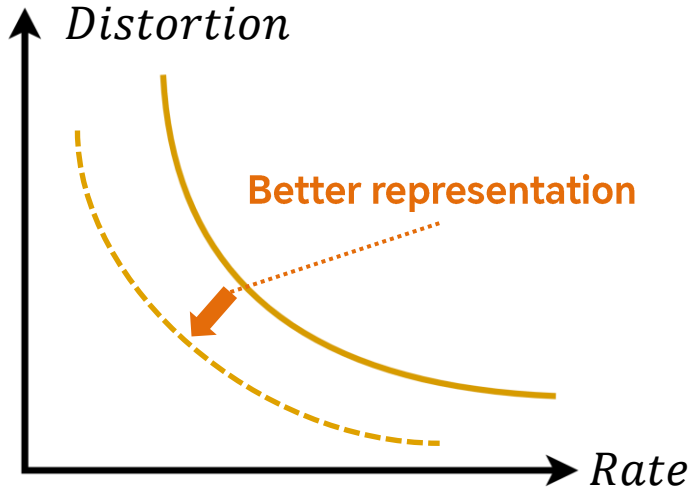
Goal: use fewer bits while preserving visual information.

Compression Pipeline



- **Rate:** Bit length of bitstream
- **Distortion:** Visual information loss

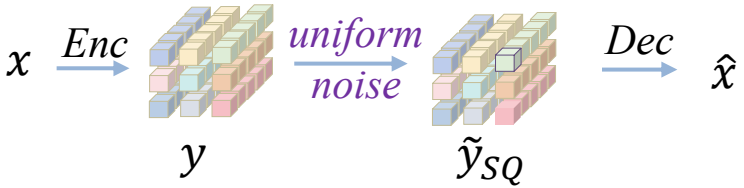
Better Representation Improves RD Trade-off



Key question:
Can we build a compression model with more expressive quantized representations?

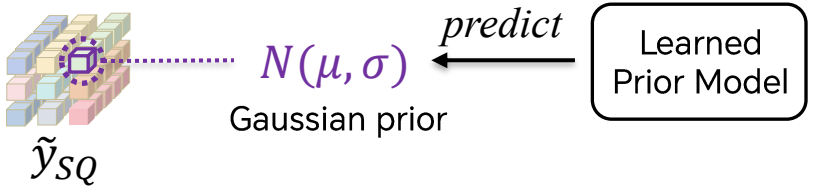
VAE-Based Compression Relies on Scalar Quantization

VAE Autoencoder



- Additive uniform noise approximates **element-wise SQ**
- $\boldsymbol{Distortion = D(x, \hat{x})}$

Learned Prior for Rate Estimation



- **Gaussian prior** models each latent element
- $\boldsymbol{Rate = -log(p(\tilde{y}_{SQ}))}$

Joint Optimization: $\boldsymbol{L = R + \lambda \cdot D}$

But SQ-based representations are limited:

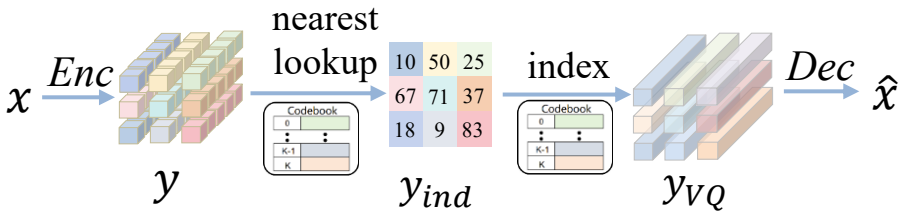
- Element-wise quantization limits **representation capacity**
- Gaussian prior over-constrains **latent distribution**
- Noise relaxation causes **train–test mismatch**

This motivates us to explore VQ-based compression.

Element-wise scalar quantization limits representation quality.

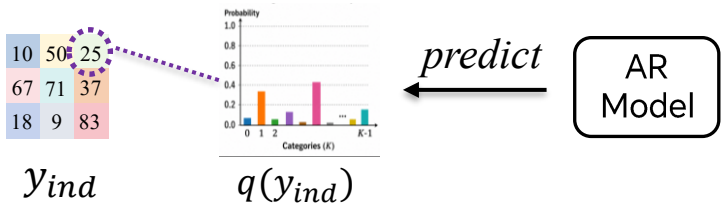
VQVAE: Compression-Aligned, But Not Jointly Optimized

VQVAE AutoEncoder



- **Vector quantization** with a learnable codebook
- **Reconstruction loss → distortion**: $\boldsymbol{Distortion} = \boldsymbol{D}(\mathrm{x}, \hat{\mathrm{x}})$

AR Prior for Rate Modeling



- AR prior predicts code probabilities
- **Cross entropy → coding rate**: $\boldsymbol{Rate} = \boldsymbol{H}\big(p(y_{ind}), q(y_{ind})\big)$

VQVAE is naturally aligned with compression,
but two-stage training decouples joint RD optimization.

Two-stage training:

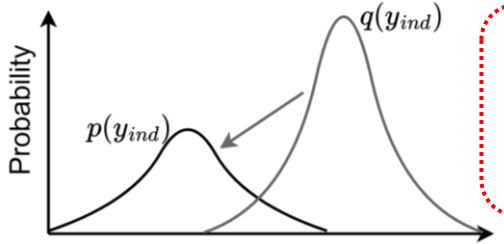
Stage1: Learn representation

$\boldsymbol{D}(\mathrm{x}, \hat{\mathrm{x}}) \rightarrow$ opt. $p(y_{ind})$

Stage2: Fit AR prior

$\boldsymbol{H}\big(p(y_{ind}), q(y_{ind})\big) \rightarrow$ opt. $q(y_{ind})$

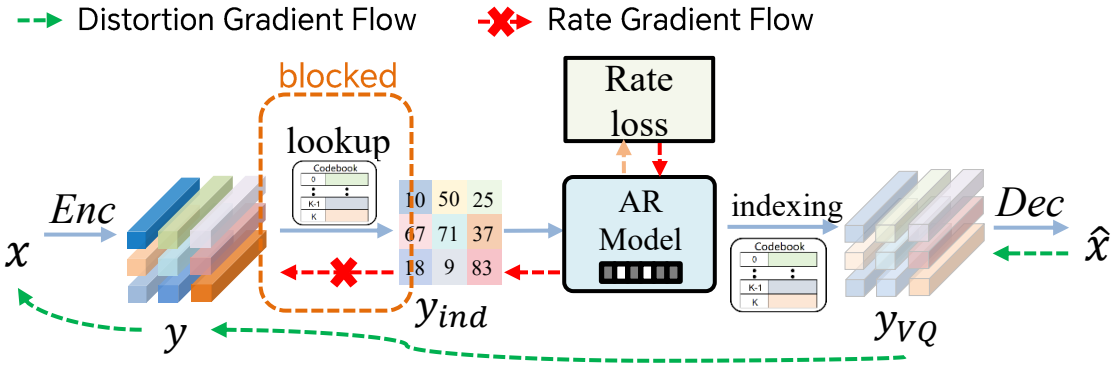
Fixed



- Rate optimization fits fixed codex distribution, but **cannot shape feature distribution**.

Why Naive Joint RD Optimization Fails

Naive Joint Optimization



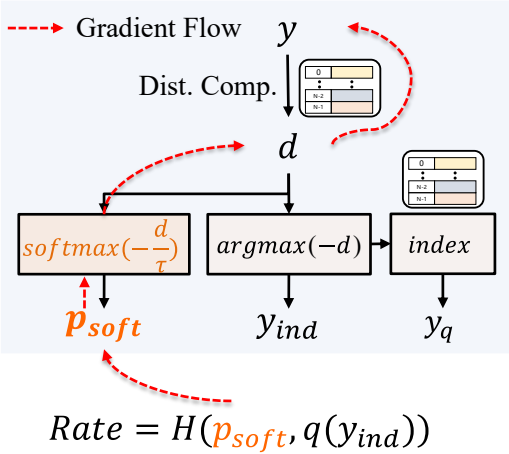
Problem

- Distortion path is differentiable
- Rate path is blocked by hard lookup
- Rate cannot update the VQ representation

Takeaway: Rate gradients are blocked before reaching the VQ representation.

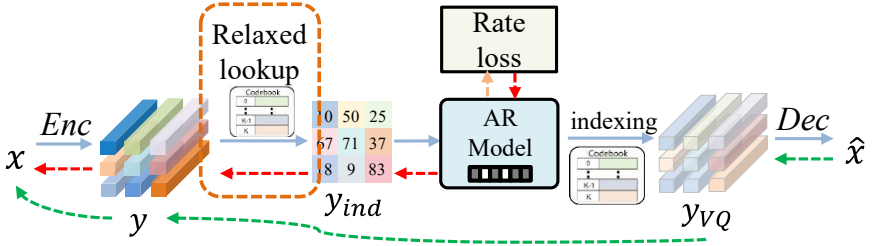
Relaxed Lookup Makes VQ-VAE Jointly RD-Optimizable

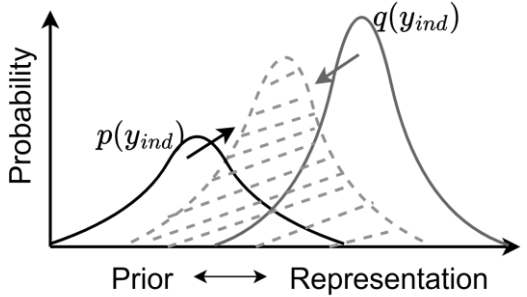
Key Idea: Relaxed Lookup



- Compute distances to codebook entries
- Use **softmax** to get $\mathbf{p_{soft}}$, which is used for **rate estimation** and **gradient flow**

RDVQ: Joint RD Optimization with VQ

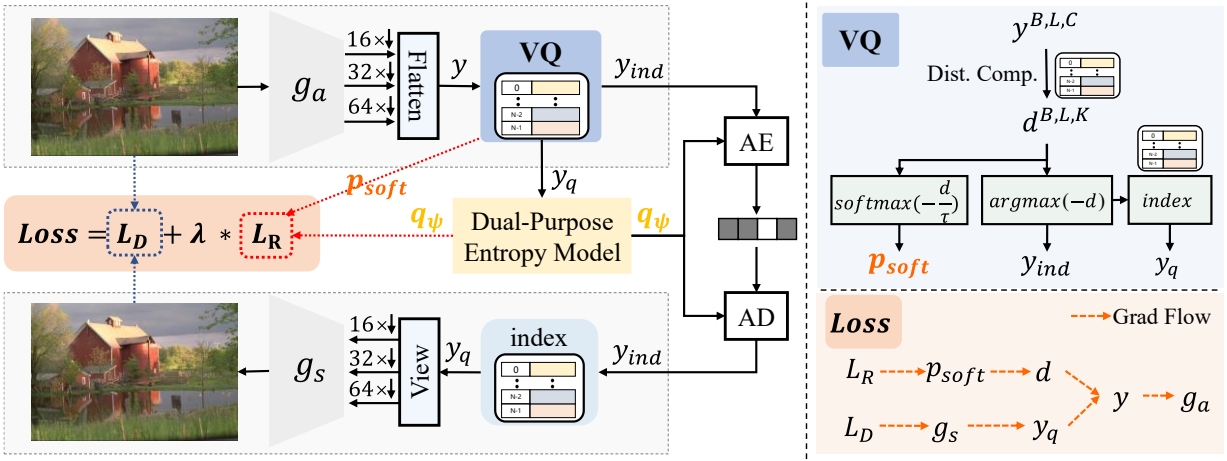




A simple relaxation lets both rate and distortion shape the VQ representation.

A Simple Framework for VQ-Based Compression

A simple codec framework with a VQ tokenizer and an AR probability model.



VQ Tokenizer ($g_a + VQ + g_s$)

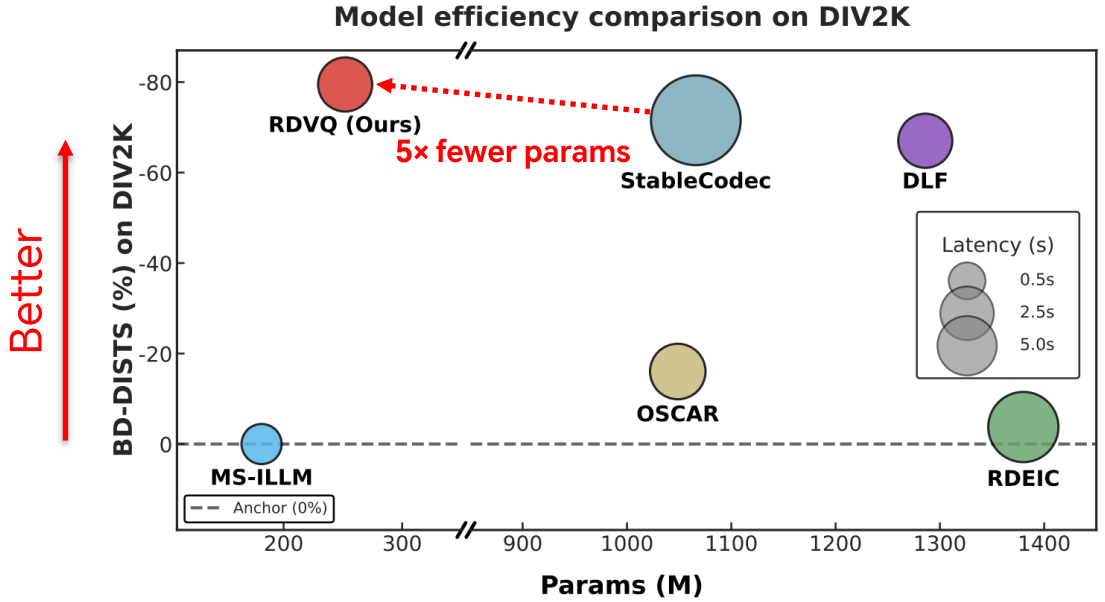
- Encode images into multi-scale VQ tokens
- Decode VQ tokens back to reconstructed images

AR Probability Model (Dual-Purpose Entropy Model)

- **Entropy modeling:** predict code probabilities
- **Autoregressive generation for test-time rate control:** complete remaining indices from prefixes

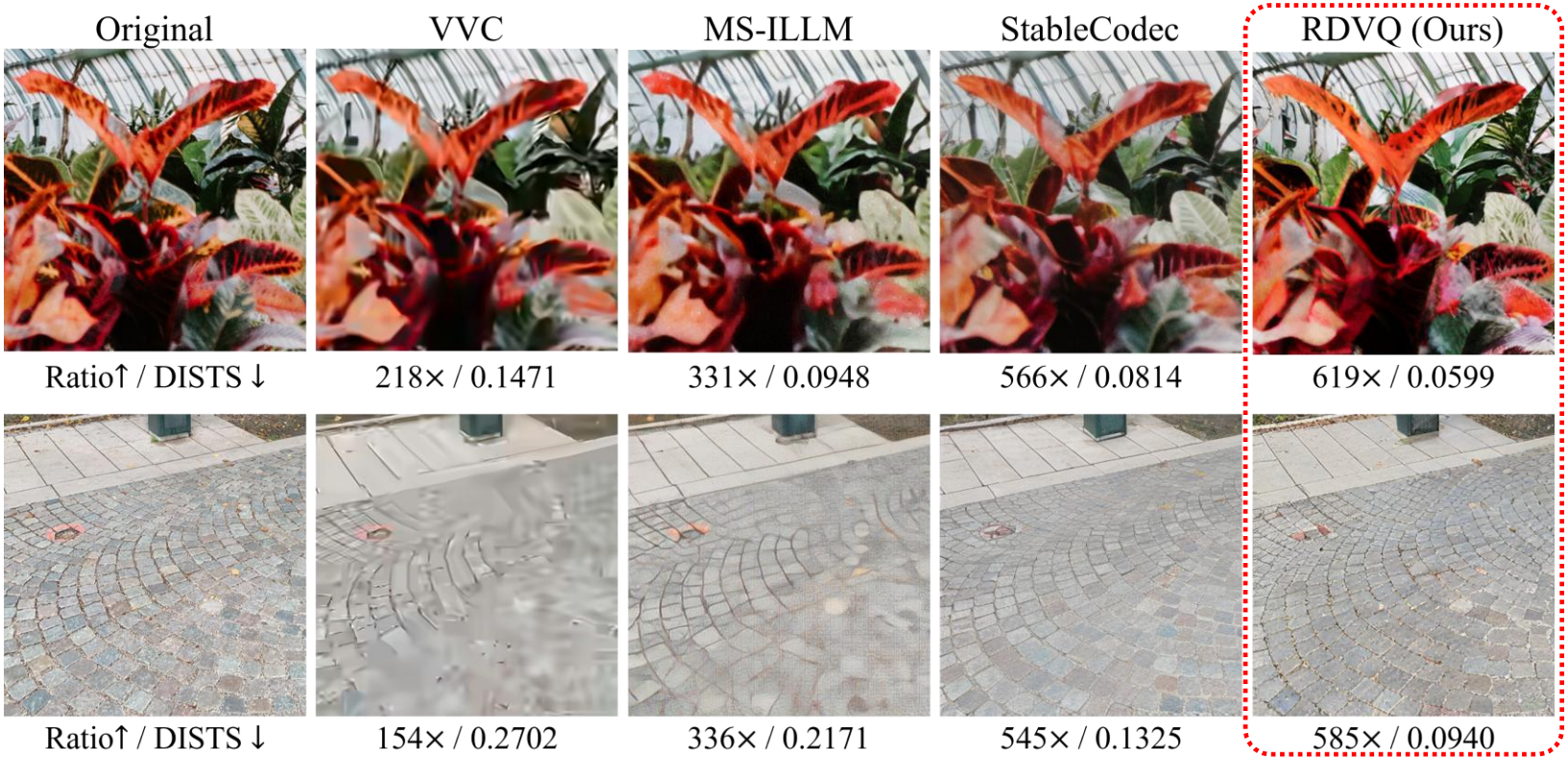
RDVQ Achieves SOTA with Fewer Parameters

Achieving SOTA results while requiring only 20% of model parameters



Outperforms prior SQ-based RD methods and previous VQ-based non-RD-optimized codecs

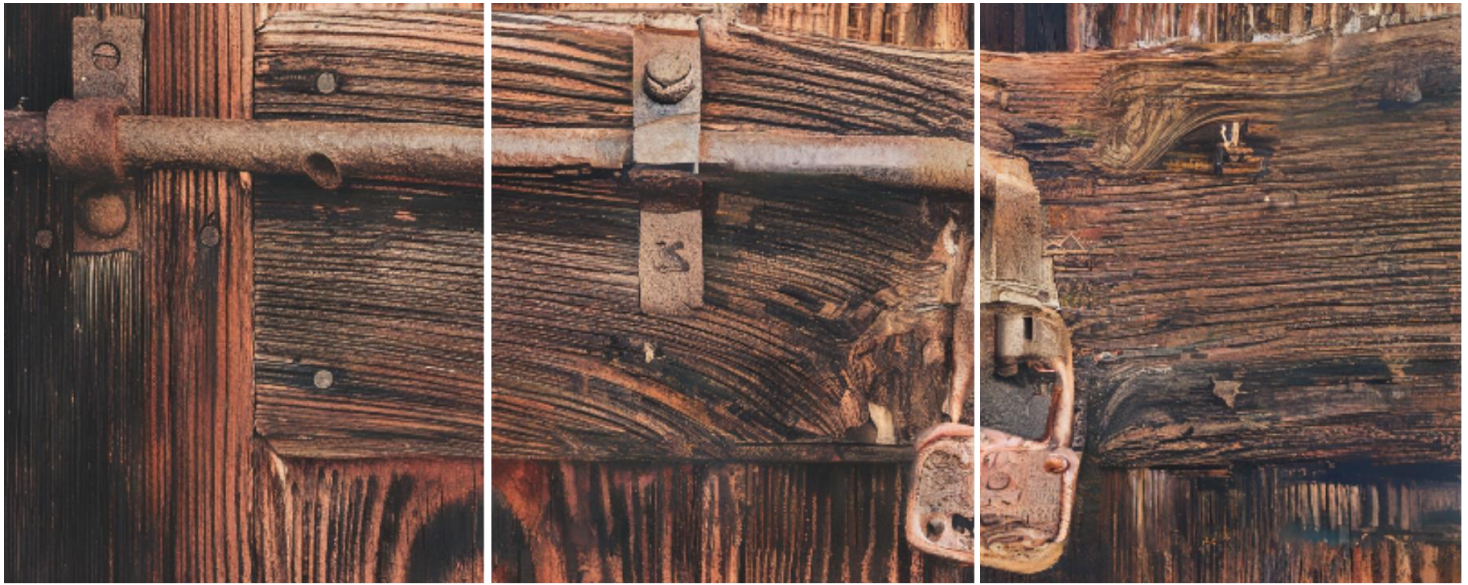
Higher Compression, Better Detail Preservation



AR Completion Enables Test-Time Rate Control

Transmit fewer indices and let the AR model complete the rest.

– More AR-completed tokens → Higher compression ratio



Ratio: ~500×　　Ratio: ~1000×　　Ratio: ~1500×

- Compression ratios from **~500×** to **~1500×** with only minor quality degradation
- **Joint RD optimization preserves AR generative ability.**

Relaxed Lookup Is Essential for Joint RD Optimization

Ablation on DIV2K-val

Method	bpp ↓	DISTS ↓	LPIPS ↓	FID ↓
RDVQ (full)	**0.0247**	**0.1005**	**0.2321**	**19.96**
w/o Relaxation	0.0464	0.2147	0.5031	86.93
K-means VQ	0.0247	0.1253	0.2831	28.08

Without relaxation:

- Rate gradients cannot effectively shape the representation, causing severe RD degradation.

Compared with K-means VQ:

- RDVQ learns more compressible VQ representations.

VQ alone is not enough → relaxed joint RD optimization is the key.

Joint RD Optimization Makes Latents More Predictable

Encoder output features become simpler at higher compression ratios

Less predictable → More predictable

Input / Ratio | Latent / ~500 × | Latent / ~700 × | Latent / ~1000 ×

Joint RD optimization encourages the encoder to generate more predictable features under stronger rate constraints, leading to a lower bitrate.

Conclusion

1. Revisit VQVAE as a Natural VQ Codec

VQ indices serve as compact symbols, reconstruction loss as distortion, and AR cross entropy as rate modeling.

2. Enable End-to-End VQ-Based Joint RD Optimization

Relaxed lookup provides effective rate feedback, making VQ representations jointly RD-optimizable.

3. Achieve SOTA Compression with a Simple Framework

RDVQ reaches SOTA-level performance with far fewer parameters, outperforming SQ-based RD methods and previous VQ-based codecs.

4. Outlook: RD-Optimized Tokens Beyond Compression

Prior-constrained token learning may provide compact and predictable token spaces for future generation and token representation tasks.

CVL-UESTC

Shuhang Gu

Leheng Zhang Xingyu Zhou Xiaorui Zhao Jinhua Zhang Jingbo Lu Weiyi You Shiyin Jiang Kai Chen

Wei Long Junyu Lou Chunyu Meng Jiajie Luo Jiale Zou JiaFeng Li Zhaoxun Wang Zihong Zhang

RedNote

Thanks!

Github