WETOK

Powerful Discrete Tokenization for High-fidelity Visual Reconstruction

Shaobin Zhuang et al. | Shanghai Jiao Tong University, WeChat Vision, Tencent Inc., and collaborators

ICLR 2026 | 5-minute report | Discrete tokenizer

Goal

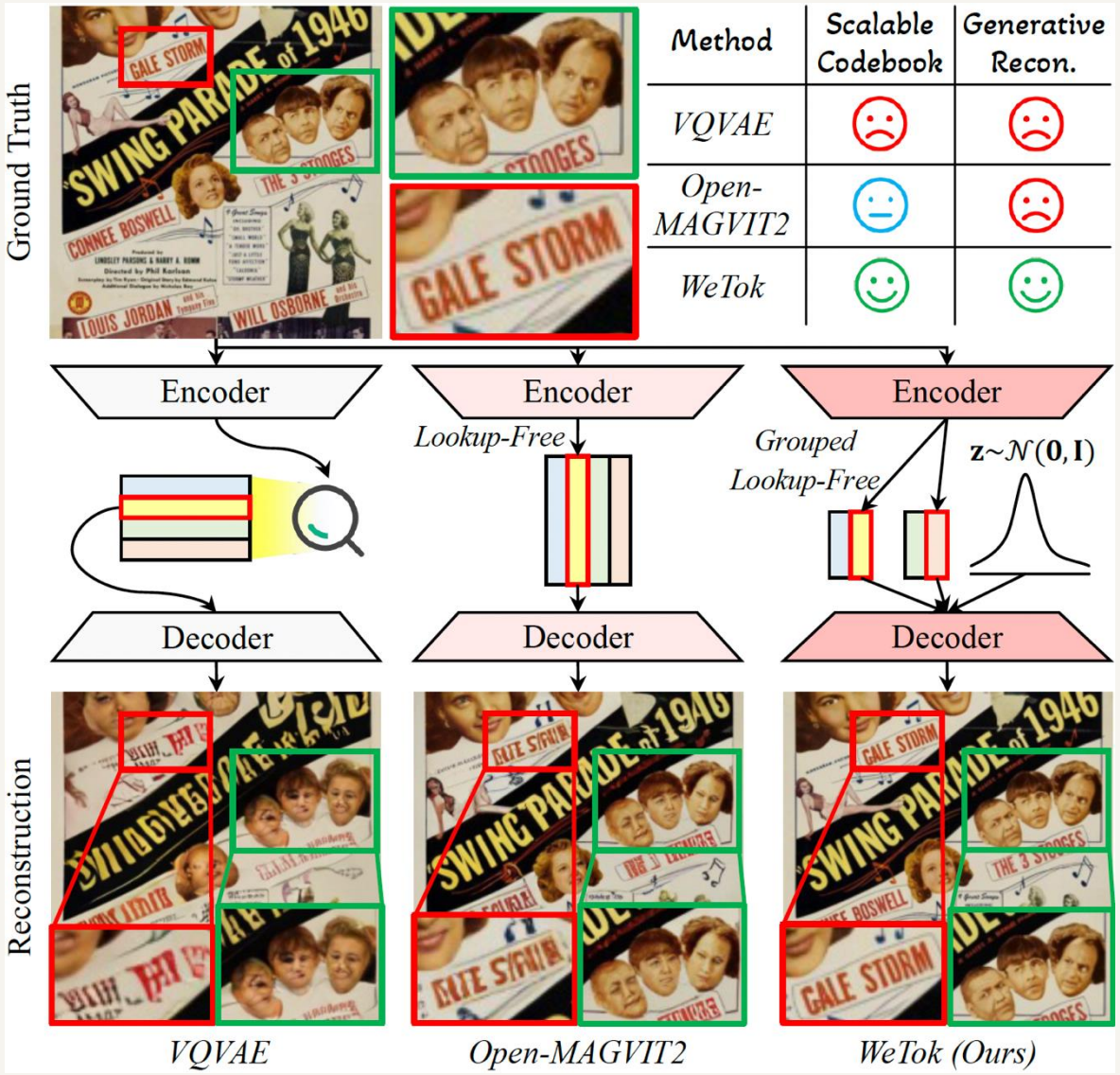
Keep the compactness of discrete tokens without sacrificing reconstruction fidelity.

GQ

Grouped lookup-free quantization for scalable codebooks.

GD

A generative decoder that restores plausible details.



Why is this problem hard?

Trade-off in existing visual tokenizers

- Discrete tokenizers are compact, but fidelity often drops.
- Continuous tokenizers reconstruct well, but are less efficient.
- The goal is a discrete tokenizer that is both compact and high-fidelity.

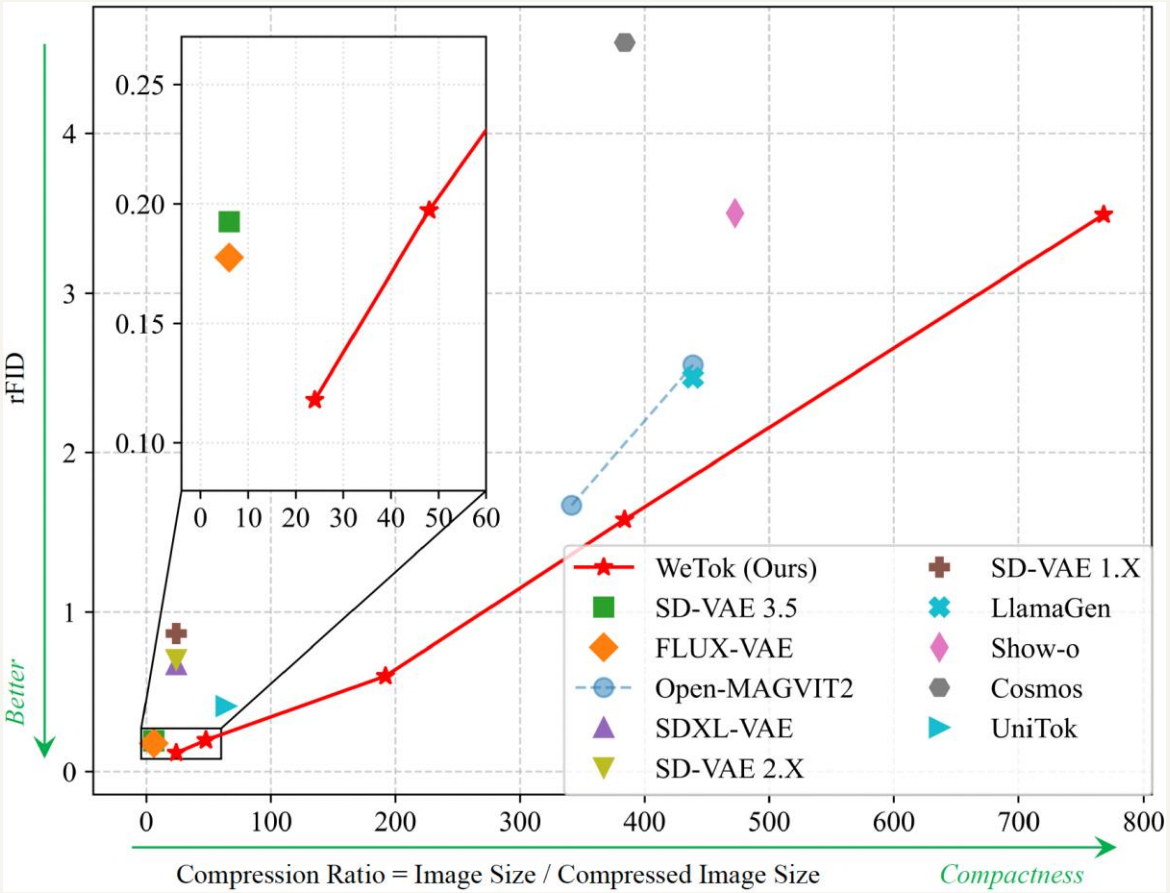
Two bottlenecks

1. Scalable codebook

LFQ improves reconstruction by enlarging the implicit codebook, but entropy loss grows memory and compute with code size.

2. Generative modeling

A deterministic decoder learns an average image, so fine details become blurry at high compression.

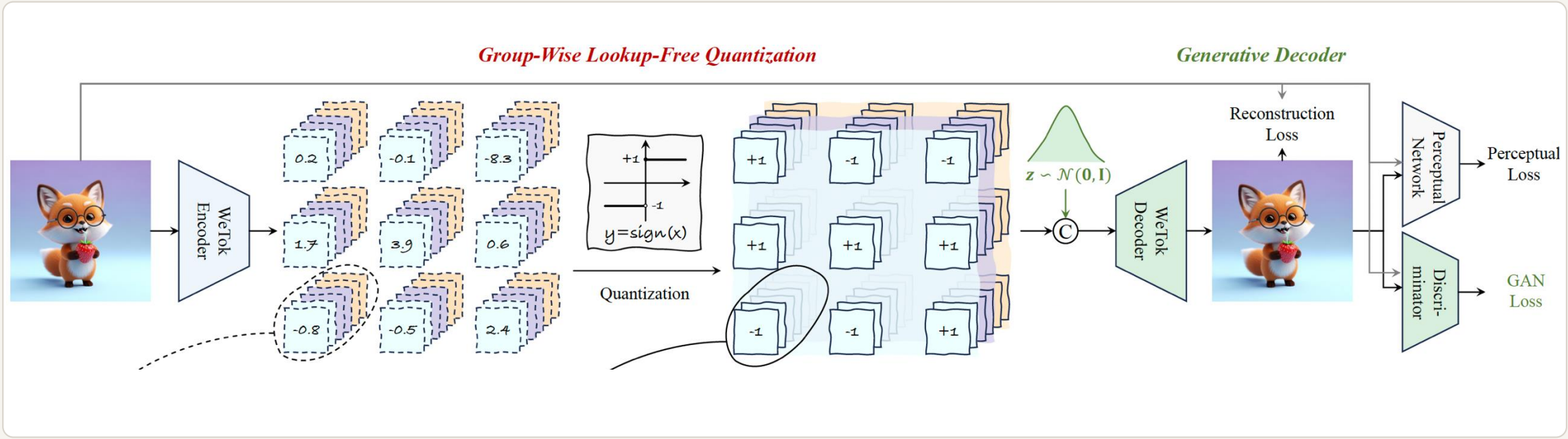


WeTok shifts the Pareto frontier toward both better fidelity and stronger compactness.

WeTok in one slide

1. Encode image to latent features
2. Group-wise lookup-free quantization
3. Decode from tokens + sampled z
4. Reconstruction + perceptual + GAN losses

Design principle: keep tokenization concise, but make decoding expressive enough to recover missing high-frequency details.



Tokenization side

GQ scales the implicit codebook without LFQ's memory bottleneck.

Decoder side

The decoder uses both the quantized token and sampled Gaussian noise z.

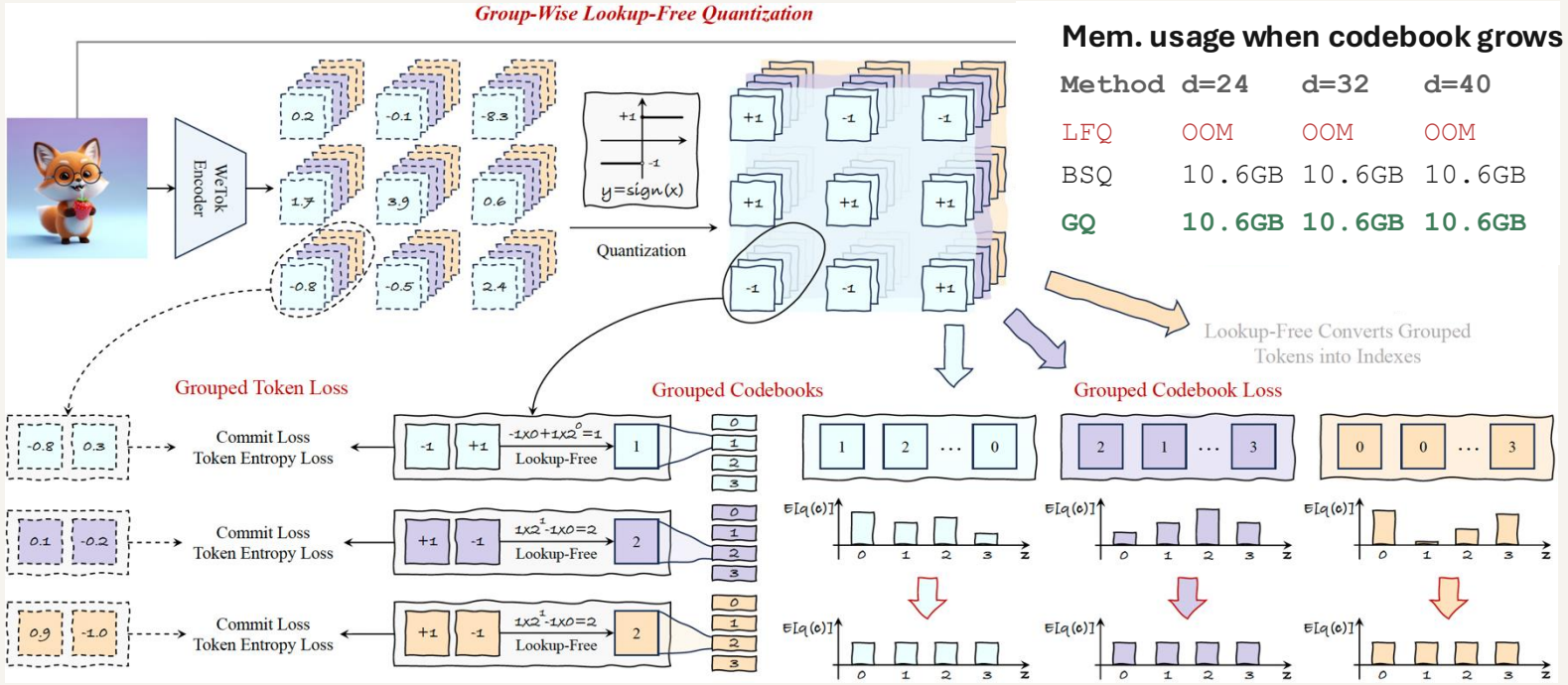
Training strategy

Stage 1 reconstructs; Stage 2 adapts the decoder with zero-init new channels.

Core idea #1: Group-Wise Lookup-Free Quantization

What GQ changes

- Split latent channels into g groups and apply lookup-free quantization per group.
- Rewrite token entropy exactly in grouped form and approximate codebook entropy group-wise.
- This removes LFQ's memory bottleneck while keeping a much better approximation than BSQ.
- The paper proves GQ has smaller codebook-entropy approximation error than BSQ.



Mem. usage when codebook grows

Method	d=24	d=32	d=40
LFQ	OOM	OOM	OOM
BSQ	10.6GB	10.6GB	10.6GB
GQ	10.6GB	10.6GB	10.6GB

Ablation summary

Under the same compression ratio, GQ uses BSQ-level memory, performs better than LFQ, and far exceeds BSQ.

Key message: GQ gives a tunable trade-off between approximation accuracy and memory cost, so the codebook can scale much further.

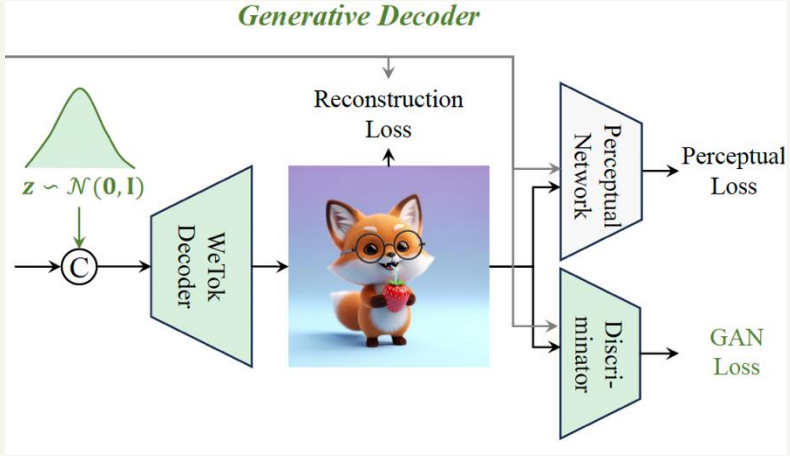
Core idea #2: Generative Decoder

Why a generative decoder?

- At high compression, one token can correspond to multiple valid textures and local details.
- A deterministic decoder predicts the conditional average, which appears blurry.
- We decode from G(z, U_Q): the discrete token provides structure, and sampled z provides detail.
- Two-stage training keeps the transition stable by zero-initializing the new input channels.

Stage 1: reconstruction training

Stage 2: generative adaptation



Generative decoder ablation

Setting	rFID	LPIPS	SSIM
Stage1 only	5.37	0.17	0.54
Stage1+Stage2	**3.90**	**0.16**	**0.55**

rFID improves by 27.4%

Results and takeaways

0.12 rFID @ 24x

Zero-shot high fidelity

On ImageNet, WeTok beats FLUX-VAE (0.18) and SD-VAE 3.5 (0.19) while keeping 24x compression.

3.49 rFID @ 768x

High-compression regime

WeTok outperforms Cosmos at 4.57, even though Cosmos uses only half the compression ratio.

2.31 gFID

Class-conditional generation

WeTok-AR-XL reaches state-of-the-art ImageNet generation quality among compared AR tokenizers.

More SOTA numbers

ImageNet 256x256 reconstruction reaches rFID 0.61 with 16x16 tokens and 0.19 with 32x32 tokens. WeTok is also stable under repeated compression-decompression.

Main message: discrete tokenizers can remain highly compact without giving up reconstruction quality.