BEYOND SCALAR ASSIGNMENTS: GEOMETRIC BINARY QUANTIZATION FOR VISUAL TOKENIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Lookup-free quantization (LFQ) constructs large visual vocabularies compositionally from binary variables. Compared with conventional vector quantization, which relies on an explicit learned vector codebook and vector-level nearest-neighbor search, LFQ replaces this machinery with independent sign assignments, offering a more scalable discrete representation. However, this simplicity fixes the quantizer-side binary support to an axis-aligned hypercube. Although the encoder can adapt its representation toward this support, the quantizer geometry itself remains prescribed. To address this limitation, we introduce Geometric Binary Quantization (GBQ), which learns linear transformations to adapt the binary code geometry. However, directly applying a single transformation to the full binary latent would make exact nearest-code search prohibitively expensive. GBQ therefore factorizes the latent into small blocks and learns a local transformation within each block, enabling vector-level assignment locally while preserving exact and tractable assignment globally. Using the same tokenizer architecture and reconstruction objective on ImageNet-1K, GBQ achieves better reconstruction than WeTok with substantially fewer training-image exposures. Under matched-budget controlled training, the reconstruction advantage emerges early and persists throughout optimization, while GBQ maintains high code utilization without explicit entropy regularization. GBQ also retains competitive zero-shot reconstruction on out-of-domain data.

1 INTRODUCTION

Visual tokenization maps images into compact sequences of discrete tokens that can be modeled by token-based generative models. Such discrete representations have become a common interface for high-fidelity reconstruction and token-based image generation across vector-quantized autoencoders, masked-token models, residual tokenizers, and autoregressive image generators (van den Oord et al., 2017; Esser et al., 2021; Yu et al., 2022; Chang et al., 2022; Lee et al., 2022; Yu et al., 2024; Sun et al., 2024). Conventional vector quantization (VQ) represents each encoder feature by its nearest entry in a learned vector codebook, making codebook capacity and optimization central to tokenizer quality. Scaling explicit VQ to large vocabularies remains challenging because the codebook must be stored and searched during quantization, while hard nearest-neighbor assignment directly updates only a small subset of code entries. Recent methods have substantially improved the scalability and utilization of explicit VQ through better codebook construction, parameterization, and optimization (Zhu et al., 2024; Shi et al., 2025; Zhu et al., 2025). Despite these advances, explicit VQ retains its defining structure: each discrete index selects an entire learned vector code from an explicit codebook.

Lookup-Free Quantization (LFQ) takes a fundamentally different approach, constructing an implicit combinatorial code space from binary variables without maintaining an explicit learned vector codebook (Yu et al., 2024). Quantization then reduces to independent per-dimension sign assignments, enabling exponentially large binary code spaces through simple discrete decisions without a global codebook lookup. This factorized binary formulation has since been adopted and extended by subsequent visual tokenizers, with later methods like BSQ and WeTok improving code utilization through entropy estimation (Luo et al., 2024; Zhao et al., 2025; Zhuang et al., 2026b) and scaling binary tokenization to larger generative and multimodal models (Zhuang et al., 2026a; Ai et al., 2026).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

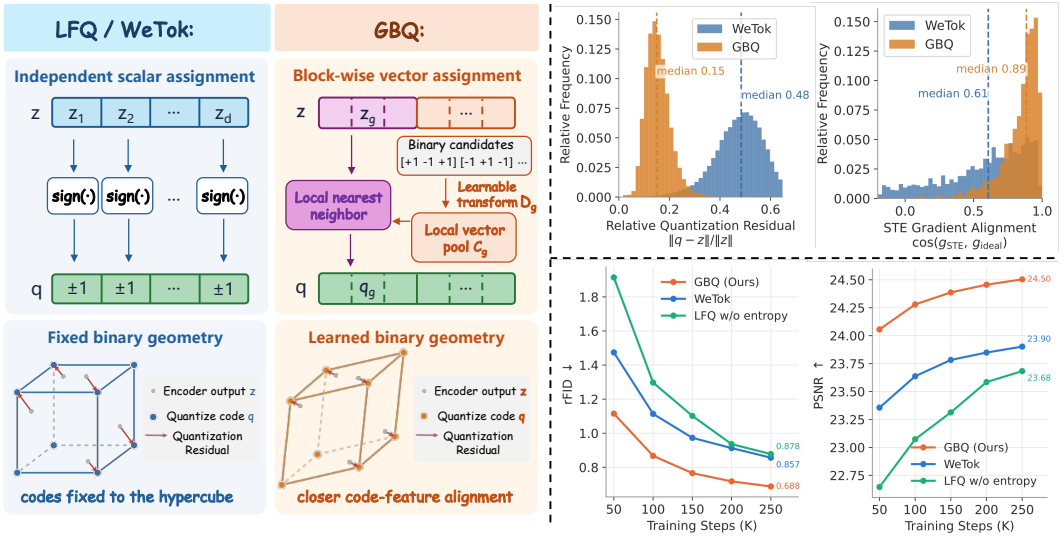

Figure 1: **Geometric view and empirical diagnostics of GBQ. Left**: LFQ and WeTok assign each dimension independently under a prescribed binary geometry, with codes fixed to the hypercube vertices, whereas GBQ performs joint block-wise assignment in a learned local geometry that aligns codes more closely with encoder features. **Top right**: Under a matched tokenizer setting on ImageNet validation features, GBQ reduces the median relative quantization residual from 0.48 to 0.15 and increases the median STE–reference gradient alignment from 0.61 to 0.89. **Bottom right**: Under a matched training budget, GBQ achieves better reconstruction quality from the early stages of training and maintains the advantage throughout training compared with WeTok and LFQ without entropy loss, as measured by rFID and PSNR. Measurement details are provided in Appendix C.1.

However, in the standard LFQ formulation, the quantizer geometry remains prescribed: the codewords lie on the vertices of an axis-aligned binary hypercube, and Euclidean nearest-code assignment decomposes exactly into independent per-dimension sign decisions. Entropy regularization can improve code utilization and reshape the encoder representation during training, but it leaves the prescribed binary support and factorized assignment rule unchanged. Under a controlled ImageNet-1K tokenizer setting, we analyze WeTok (Zhuang et al., 2026b) using two latent-space diagnostics: *relative quantization residual* for code–feature mismatch and *STE–reference gradient alignment* for gradient consistency. As shown in Fig. 1 (top right), WeTok exhibits a substantial relative quantization residual and limited STE–reference gradient alignment, characterizing code–feature mismatch and reduced gradient consistency under the prescribed binary geometry.

To relax this geometric constraint, we introduce **Geometric Binary Quantization (GBQ)**. As illustrated in Fig. 1 (left), GBQ departs from LFQ's lookup-free sign assignment by grouping the binary latent into blocks and learning a local linear transformation of the 2^b binary states within each block, yielding a structured vector codebook for exact nearest-code assignment. This enables coupled vector-level assignment within each block and allows the induced local code geometry to co-adapt with the encoder representation. Because the search is restricted to 2^b codes per block and remains factorized across blocks, the assignment is exact while its computational overhead is controlled by the block size. As shown in Fig. 1 (top right), GBQ reduces the median relative quantization residual from 0.48 to 0.15 and increases the median STE–reference gradient alignment from 0.61 to 0.89. The early-training curves in Fig. 1 (bottom right) further show lower rFID and higher PSNR for GBQ from the early stages of training under the matched training budget.

Our contributions are three-fold:

- We introduce GBQ, a block-wise binary quantizer that learns local geometry over each block's complete binary state space, enabling exact joint assignment while preserving compositional binary identities.
- We characterize when the induced binary geometry remains separable and when it permits non-separable interactions among variables within a block, and show that block-wise fac-

torization preserves exact global nearest-code assignment with computational complexity controlled by the block size.

- We show that GBQ improves reconstruction fidelity over previous LFQ methods with substantially fewer training-image exposures; under matched-budget training, this advantage emerges early and persists throughout optimization, while high code utilization is maintained without explicit entropy regularization. These gains come with little practical overhead at moderate block sizes and extend to competitive zero-shot reconstruction on out-of-domain data.

2 Related Work

2.1 Vector Quantization for Visual Tokenization

Vector quantization has long served as a basic mechanism for learning discrete visual representations. Early works established the encoder–codebook–decoder paradigm and progressively improved its reconstruction quality and representational capacity through hierarchical latents, stronger objectives, and improved architectures (van den Oord et al., 2017; Razavi et al., 2019; Esser et al., 2021; Yu et al., 2022; Zheng et al., 2022). Other works restructure the quantization unit through residual, product, or grouped vector quantization. RQ-VAE (Lee et al., 2022) performs sequential residual code selection, while PQGAN (Zavadski et al., 2026), UniTok (Ma et al., 2025), MGVQ (Jia et al., 2025), ReVQ (Zhang et al., 2025a), and DCVQ (Zhang et al., 2025b) partition the representation into multiple quantized groups or subspaces. GBQ differs from grouped or product VQ in how each local codebook is parameterized: rather than learning 2^b local code vectors independently, it applies a shared learnable linear transformation to the complete binary state set, inducing a structured local code geometry while preserving the combinatorial identity space across blocks.

A complementary line of work focuses on scaling and optimizing explicit codebooks. Improved VQGAN (Yu et al., 2022), VQGAN-LC (Zhu et al., 2024), and SimVQ (Zhu et al., 2025) improve large-codebook training through lower-dimensional lookup, pretrained code construction, or shared code reparameterization. IBQ (Shi et al., 2025) and LGQ (Altun et al., 2026) improve codebook optimization by propagating training signals through assignments over the full codebook, while retaining explicit vector codebooks and vector-level discrete selection. GBQ instead parameterizes each local codebook as a shared linear transform of the complete binary state set, preserving compositional binary identities.

2.2 Lookup-Free and Binary Quantization

Lookup-free quantizers avoid an explicit learned vector codebook by constructing the discrete space compositionally from scalar variables. FSQ (Mentzer et al., 2024) forms an implicit Cartesian-product vocabulary from independently quantized scalar levels, while LFQ (Yu et al., 2024) uses binary scalar variables, yielding a hypercube whose Euclidean nearest-code assignment reduces to element-wise sign decisions. Open-MAGVIT2 (Luo et al., 2024) further studies the scalability of LFQ-based tokenization for autoregressive generation. More recently, Spherical Leech Quantization (Zhao et al., 2026) explores fixed lattice geometries beyond the binary hypercube, highlighting code geometry as an alternative design axis.

A separate challenge for binary tokenizers is effective utilization of their combinatorial vocabulary. LFQ introduces entropy regularization over the binary-code distribution. As dimensionality grows, later methods use increasingly factorized statistics: BSQ (Zhao et al., 2025) factorizes its entropy approximation across bits, while WeTok (Zhuang et al., 2026b) estimates entropy over groups of binary variables to balance statistical fidelity and computational cost. Subsequent work scales binary token spaces to larger generative and multimodal models (Zhuang et al., 2026a; Ai et al., 2026). These approaches retain prescribed code geometries, whereas GBQ learns local linear transformations of the binary state space so that the geometry governing discrete assignment can co-adapt with the representation. Empirically, GBQ maintains high code utilization without explicit entropy regularization.

3 METHOD

Section 3.1 places vector quantization and LFQ under a common nearest-code formulation and shows how the structure of the code space determines the scope of each discrete assignment. Section 3.2 introduces GBQ, which learns local binary geometries and performs exact block-wise vector assignment while preserving an exact factorization of the global problem. Section 3.3 describes the straight-through optimization and the complete tokenizer objective.

3.1 A UNIFIED NEAREST-CODE VIEW

Given a latent feature $z \in \mathbb{R}^d$ and a discrete code set $\mathcal{C} \subset \mathbb{R}^d$, we write nearest-code quantization as

$$Q_{\mathcal{C}}(z) = \arg\min_{c \in \mathcal{C}} \|z - c\|_2^2. \tag{1}$$

Both conventional VQ and LFQ are instances of this formulation; the structure of $\mathcal{C}$ determines whether assignment is a joint vector decision or decomposes into independent scalar decisions.

Vector-level assignment. Conventional VQ uses an explicit learned codebook $\mathcal{C}_{\mathrm{VQ}} = \{e_1, \ldots, e_K\}$, yielding

$$k^\star = \arg\min_k \|z - e_k\|_2^2, \qquad Q_{\mathrm{VQ}}(z) = e_{k^\star}. \tag{2}$$

Each discrete index selects an entire learned vector, so nearest-code assignment remains a joint vector-level decision.

Separable binary assignment. LFQ uses the prescribed binary code space $\mathcal{C}_{\mathrm{LFQ}} = \{-1, +1\}^d$. Under Euclidean distance,

$$Q_{\mathrm{LFQ}}(z) = \arg\min_{c \in \{-1,+1\}^d} \|z - c\|_2^2 = \mathrm{sign}(z), \tag{3}$$

because the objective decomposes coordinate-wise. Thus, the same nearest-code problem that remains vector-level in VQ becomes d independent scalar decisions under the prescribed binary hypercube.

In practice, LFQ-style tokenizers commonly regularize the occupancy of the combinatorial code space through an entropy objective over a soft assignment distribution $p(c \mid z)$,

$$\mathcal{L}_{\mathrm{ent}} = \mathbb{E}_z\big[H(p(c \mid z))\big] - \lambda_{\mathrm{ent}}\, H\big(\mathbb{E}_z[p(c \mid z)]\big). \tag{4}$$

The first term encourages confident per-token assignments, while the second promotes diversity in the aggregate code distribution. LFQ, BSQ, and WeTok instantiate this objective using full-code, bit-wise, and group-wise statistics, respectively (Yu et al., 2024; Zhao et al., 2025; Zhuang et al., 2026b). These objectives affect code occupancy and the learned encoder distribution, but do not change the prescribed binary code locations or the sign-based assignment in Eq. (3).

3.2 BLOCK-WISE GEOMETRIC BINARY QUANTIZATION

GBQ retains the combinatorial binary identity space while making the local code geometry learnable through linear transformations. To keep exact nearest-code assignment tractable, we restrict each transformation to a small block of binary variables.

Structured local codebook. Given a latent vector $z \in \mathbb{R}^d$, we partition its channels into $G = d/b$ non-overlapping blocks $z^{(g)} \in \mathbb{R}^b$. Let $\mathcal{B}_b = \{-1, +1\}^b$ denote the 2^b binary states, and let $S_b = [s_1, \ldots, s_{2^b}] \in \{-1, +1\}^{b \times 2^b}$ enumerate all states in $\mathcal{B}_b$. Intuitively, a learnable linear transformation maps each binary sign pattern to a local code vector. Formally, for each block g, we introduce $D_g \in \mathbb{R}^{b \times b}$ and define

$$C_g = D_g S_b, \qquad c_{g,j} = D_g s_j, \tag{5}$$

where $C_g = [c_{g,1}, \ldots, c_{g,2^b}]$ is the resulting structured local codebook. The shared transformation jointly determines the locations of all 2^b local codes and thereby induces a learnable local code geometry.

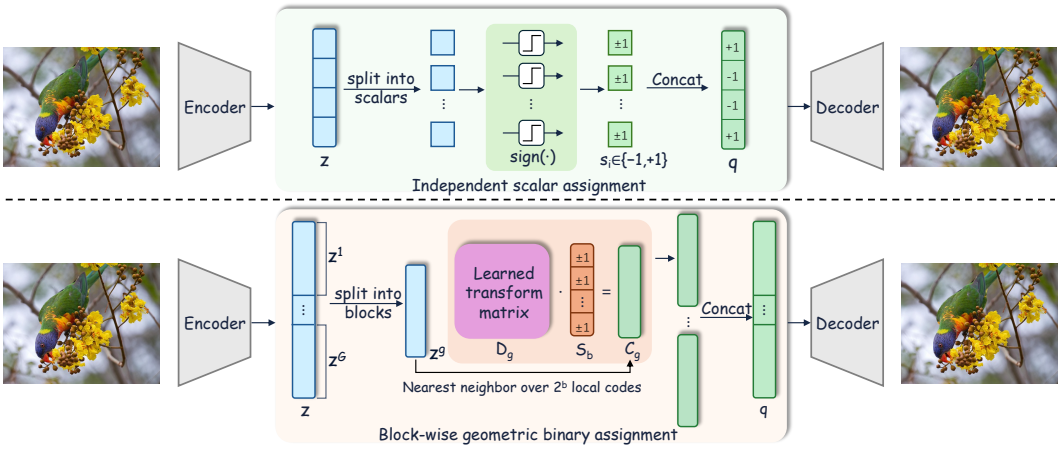

Figure 2: From coordinate-wise binary assignment to block-wise geometric binary quantization. **Top:** LFQ independently assigns each latent coordinate to a prescribed binary value. **Bottom:** GBQ groups latent dimensions into blocks and applies a learnable linear transformation to the complete binary state set, inducing a structured local code geometry for exact nearest-code assignment. The learned geometry can enable coupled vector-level decisions within each block, while block-wise factorization preserves exact global assignment.

Given $z^{(g)}$, GBQ performs exact nearest-code assignment within the local codebook,

$$j_g^\star = \operatorname*{arg\,min}_{j\in\{1,\dots,2^b\}} \left\| z^{(g)} - D_g s_j \right\|_2^2, \qquad q^{(g)} = D_g s_{j_g^\star}. \tag{6}$$

The quantized latent is $q = [q^{(1)}, \dots, q^{(G)}]$. The selected state $s_{j_g^\star}$ specifies the discrete binary identity, whereas $q^{(g)}$ is the corresponding transformed code vector passed to the decoder. Across all blocks, GBQ introduces $Gb^2 = db$ learnable transformation parameters.

Coupled vector assignment. The structure of the discrete assignment is governed by the local geometry induced by the transformation D_g. For a candidate state $s \in \mathcal{B}_b$, let $M_g = D_g^\top D_g$. Using $s_i^2 = 1$, the assignment energy becomes

$$\left\| z^{(g)} - D_g s \right\|_2^2 = \text{const} - 2s^\top D_g^\top z^{(g)} + 2\sum_{i<k} (M_g)_{ik} s_i s_k. \tag{7}$$

The off-diagonal entries of M_g therefore determine interactions among the binary decisions within a block. When M_g is diagonal, the interaction terms vanish and the assignment remains separable. The special case $D_g = I_b$ recovers LFQ. More generally, a diagonal $D_g^\top D_g$ allows the local codebook to change its orientation and per-direction scale while keeping the binary decisions independent. A non-diagonal Gram matrix introduces cross-variable interaction terms, enabling coupled vector-level assignment within the block. Appendix B.1 formalizes this characterization.

Exact global factorization. Despite the possible within-block coupling, the block structure preserves exact global tractability. Let $D = \text{BlockDiag}(D_1, \dots, D_G)$ and $s = [s^{(1)}, \dots, s^{(G)}]$. The global assignment energy decomposes as

$$\|z - Ds\|_2^2 = \sum_{g=1}^{G} \left\| z^{(g)} - D_g s^{(g)} \right\|_2^2. \tag{8}$$

Hence the exact minimizer over the binary identity space is obtained by concatenating the block-wise minimizers in Eq. (6). Direct evaluation considers 2^b candidates per block, giving $G2^b = (d/b)2^b$ local candidate evaluations and $\mathcal{O}(d\,2^b)$ arithmetic per latent vector. The block size b therefore controls both the scope of within-block interaction and the search cost: $b = d$ removes the block restriction and permits fully coupled binary assignment at global exponential cost, whereas moderate

block sizes provide local vector-level interaction while keeping exact search tractable. Appendix B.3 formalizes the exact block factorization and the relation among quantizer families induced by different block sizes, while Appendix B.4 gives the corresponding complexity analysis.

3.3 Training and Optimization

The block-wise nearest-code assignment is discrete, so we train GBQ with the identity straight-through estimator (STE) (Bengio et al., 2013). Writing the quantized latent from Eq. (6) as q, the decoder receives

$$q_{\mathrm{st}} = z + \mathrm{sg}(q - z), \tag{9}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient. The forward value is q, while the backward derivative with respect to z is the identity. Reconstruction gradients therefore reach the encoder through the straight-through path, while the transformation matrices D_g are optimized through the quantization objective.

We use the standard codebook and commitment terms (van den Oord et al., 2017),

$$\mathcal{L}_{\mathrm{quant}} = \|\mathrm{sg}(z) - q\|_2^2 + \beta \, \|z - \mathrm{sg}(q)\|_2^2 , \tag{10}$$

where the squared errors are averaged over spatial positions and channels, and β is the commitment coefficient. The first term updates the selected codes through the transformation matrices D_g, while the second updates the encoder and encourages its features to remain close to the selected codes. The encoder representation and the induced code geometry are therefore optimized jointly, allowing adaptation on both the representation and quantizer sides. Appendix B.5 discusses the scope of this co-adaptation view and gives a simple characterization of the STE gradient discrepancy.

Following the base tokenizer (Zhuang et al., 2026b), the decoder reconstructs $\hat{x} = \mathrm{Dec}(q_{\mathrm{st}})$ and is trained with pixel, perceptual, and adversarial objectives (Zhang et al., 2018),

$$\mathcal{L}_{\mathrm{AE}} = \|x - \hat{x}\|_1 + \lambda_p \, \mathrm{LPIPS}(x, \hat{x}) + \lambda_{\mathrm{adv}} \, \mathcal{L}_G(\hat{x}). \tag{11}$$

Following the baseline implementation, we use $\lambda_p = 1.0$ and $\lambda_{\mathrm{adv}} = 0.1$, and adversarial training is enabled from the beginning. The complete tokenizer objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{AE}} + \lambda_{\mathrm{quant}} \mathcal{L}_{\mathrm{quant}}. \tag{12}$$

All remaining architecture and loss settings follow the tokenizer backbone. The default GBQ model uses no explicit entropy regularizer; we examine its effect separately in the ablation.

4 Experiments

In this section, we evaluate GBQ on image reconstruction and autoregressive generation, followed by controlled ablations of its key design choices and computational efficiency.

4.1 Experimental Setup

We conduct tokenizer experiments on ImageNet-1K at 256×256 resolution and evaluate reconstruction on the 50K validation images (Deng et al., 2009). For the controlled comparisons in Figure 1 and Table 4, all variants use the same tokenizer architecture, a global batch size of 128, and a matched 250K-step training budget, with differences confined to the quantizer and explicitly stated regularization choices. Table 1 provides a broader comparison with prior methods under their respective reported training settings. The published WeTok result uses 400K steps with a global batch size of 1024, whereas our headline GBQ runs use a global batch size of 128. GBQ-400K and GBQ-800K therefore process 1/8 and 1/4 as many training images as the reported WeTok run, respectively. Unless otherwise specified, ablations use a matched 250K-step budget. GBQ uses latent dimensionality $d = 32$ with $G = 4$ blocks of size $b = 8$. Full architecture and optimization details are provided in Appendix A. We report rFID, LPIPS, PSNR, and SSIM for reconstruction and FID, IS, Precision, and Recall for generation. Metric definitions and evaluation details are provided in Appendix A.1.

Table 1: Reconstruction on the ImageNet-1K 256×256 validation set, grouped by spatial downsampling ratio. Baseline results follow the respective papers. The published WeTok result uses 400K steps with global batch size 1024, whereas GBQ uses batch size 128; GBQ-400K and GBQ-800K therefore correspond to only $1/8$ and $1/4$ of WeTok's training-image exposure, respectively.

Method	Type	Tok. Res.	Ratio	Train Res.	Codebook	rFID↓	PSNR↑	Usage↑
VQGAN (Esser et al., 2021)	2D	16×16	16	256^2	1024	8.30	19.51	–
VQGAN (Esser et al., 2021)	2D	16×16	16	256^2	16384	4.99	20.00	–
SD-VQGAN (Rombach et al., 2022)	2D	16×16	16	256^2	16384	5.15	–	–
MaskGIT (Chang et al., 2022)	2D	16×16	16	256^2	1024	2.28	–	–
ReVQ (Zhang et al., 2025a)	2D	16×16	16	256^2	65536	2.57	21.69	–
LlamaGen (Sun et al., 2024)	2D	16×16	16	256^2	32768	2.26	20.59	85%
LlamaGen (Sun et al., 2024)	2D	16×16	16	256^2	16384	2.19	20.79	97%
ReVQ (Zhang et al., 2025a)	2D	16×16	16	256^2	2^{18}	2.05	21.96	–
SweetTok (Tan et al., 2025)	1D	256	16	256^2	10481	0.73	–	–
FlexTok (Bachmann et al., 2025)	1D	256	–	256^2	64000	1.45	18.53	–
VAR (Tian et al., 2024)	2D	16×16	16	256^2	4096	–	21.30	97%
IBQ (Shi et al., 2025)	2D	16×16	16	256^2	16384	1.37	22.35	96%
Open-MAGVIT2 (Luo et al., 2024)	2D	16×16	16	256^2	2^{18}	1.17	22.64	100%
IBQ (Shi et al., 2025)	2D	16×16	16	256^2	2^{18}	1.00	20.30	84%
FlowMo-Lo (Sargent et al., 2025)	1D	256	–	256^2	2^{18}	0.95	22.07	–
VFMTok (Zheng et al., 2025)	1D	256	–	256^2	16384	0.89	–	100%
GigaTok (Xiong et al., 2025)	1D	256	–	256^2	16384	0.79	21.65	–
AliTok (Wu et al., 2026)	1D	273	–	256^2	4096	0.84	–	–
MGVQ (Jia et al., 2025)	2D	16×16	16	256^2	2^{52}	0.64	23.71	100%
WeTok (Zhuang et al., 2026b)	2D	16×16	16	256^2	2^{32}	0.61	24.50	100%
GBQ 400K steps (Ours)	2D	16×16	16	256^2	2^{32}	0.62	24.60	100%
GBQ 800K steps (Ours)	2D	16×16	16	256^2	2^{32}	**0.59**	**24.72**	100%
ViT-VQGAN (Yu et al., 2022)	2D	32×32	8	256^2	8192	1.28	–	–
OmniTokenizer-VQ (Wang et al., 2024)	2D	32×32	8	256^2	8192	1.11	–	–
LlamaGen (Sun et al., 2024)	2D	32×32	8	256^2	16384	0.59	24.45	–
Open-MAGVIT2 (Luo et al., 2024)	2D	32×32	8	128^2	2^{18}	0.34	27.02	100%
BSQ (Zhao et al., 2025)	1D	1024	–	256^2	2^{18}	1.14	25.36	100%
FlowMo-Hi (Sargent et al., 2025)	1D	1024	–	256^2	2^{18}	0.56	24.93	–
Selftok (Wang et al., 2025)	1D	1024	–	256^2	2^{15}	0.54	26.30	–
BSQ (Zhao et al., 2025)	1D	1024	–	256^2	2^{36}	0.45	28.14	100%
SweetTok (Tan et al., 2025)	1D	1024	16	256^2	10481	0.37	–	–
MGVQ (Jia et al., 2025)	2D	32×32	8	256^2	2^{52}	0.31	28.42	100%
WeTok (Zhuang et al., 2026b)	2D	32×32	8	256^2	2^{32}	0.19	29.69	100%
GBQ 400K steps (Ours)	2D	32×32	8	256^2	2^{32}	0.19	29.85	100%
GBQ 800K steps (Ours)	2D	32×32	8	256^2	2^{32}	**0.18**	**29.92**	100%

4.2 Main Results

We first evaluate the representational quality of GBQ through image reconstruction. Table 1 compares GBQ with representative vector-quantized and binary visual tokenizers on ImageNet, with WeTok serving as our primary controlled baseline under the same tokenizer architecture and reconstruction objective. The 400K-step GBQ model matches WeTok's optimization-step budget while using only $1/8$ of its training-image exposure, and achieves comparable reconstruction across the two token resolutions. With 800K steps, corresponding to only $1/4$ of WeTok's exposure, GBQ further improves the reconstruction metrics and attains the strongest rFID and PSNR among the directly comparable settings. Table 1 also includes MGVQ (Jia et al., 2025), a multi-group VQ method. GBQ achieves lower rFID and higher PSNR at both spatial downsampling ratios while using a substantially smaller equivalent codebook size. Matched controlled comparisons are provided in Figure 1 and Table 4.

We further evaluate the ImageNet-trained GBQ tokenizer on the MS-COCO 2017 (Lin et al., 2014) validation set without fine-tuning, together with SD-VAE 1.x (Rombach et al., 2022) and WeTok. This setting probes whether the learned quantizer geometry transfers beyond the ImageNet training distribution. Despite being trained only on ImageNet-1K, GBQ remains competitive on resized

Table 2: Zero-shot reconstruction on the MS-COCO 2017 validation set. SD-VAE 1.x (Rombach et al., 2022) and WeTok use large-scale general-domain pretraining, whereas GBQ is trained only on ImageNet-1K. The reported training-data scale of SD-VAE 1.x follows WeTok.

			MS-COCO 256×256			MS-COCO original res.		
Method	Train Data	Comp. Ratio	rFID↓	PSNR↑	SSIM↑	rFID↓	PSNR↑	SSIM↑
SD-VAE 1.x (Discrete)	General (1B)	110	5.75	24.17	0.70	6.07	22.54	0.65
WeTok	General (400M)	192	**4.41**	**24.44**	**0.74**	3.80	**23.70**	**0.67**
GBQ	ImageNet-1K	192	4.46	24.37	0.72	**3.72**	23.49	0.66

Table 3: Class-conditional generation on ImageNet 256×256. Baseline numbers are as reported; * denotes generated images at 384×384 resized to 256×256 for evaluation. GBQ-B and GBQ-H use corresponding BitDance generator architectures with our GBQ tokenizer.

Model	Type	Order	#Params	FID↓	IS↑	Pre.↑	Rec.↑
LlamaGen-L* (Sun et al., 2024)	AR	raster	343M	3.07	256.1	0.83	0.52
LlamaGen-XXL* (Sun et al., 2024)	AR	raster	1.4B	2.34	253.9	0.80	0.59
RandAR-L (Pang et al., 2025)	AR	random	343M	2.55	288.8	0.81	0.58
RandAR-XXL (Pang et al., 2025)	AR	random	1.4B	2.15	322.0	0.79	0.62
RAR-L (Yu et al., 2025)	AR	hybrid	461M	1.70	299.5	0.81	0.60
RAR-XXL (Yu et al., 2025)	AR	hybrid	1.5B	1.48	326.0	0.80	0.63
Open-MAGVIT2-L (Luo et al., 2024)	AR	raster	804M	2.51	271.7	0.84	0.54
MAGVIT-v2 (Yu et al., 2024)	Mask.	random	307M	1.78	319.4	–	–
VAR-d20 (Tian et al., 2024)	VAR	–	600M	2.57	302.6	0.83	0.56
VAR-d30 (Tian et al., 2024)	VAR	–	2B	1.92	323.1	0.82	0.59
BitDance-B (Ai et al., 2026)	AR	raster	242M	1.68	297.1	0.80	0.62
BitDance-H (Ai et al., 2026)	AR	raster	1.0B	1.24	304.4	0.81	0.64
GBQ-B	AR	raster	242M	1.63	298.9	0.78	0.63
GBQ-H	AR	raster	1.0B	1.22	305.6	0.81	0.64

256×256 images and achieves the lowest rFID at the original image resolution, showing competitive transfer beyond its training distribution.

We further evaluate GBQ for class-conditional autoregressive generation on ImageNet using the more recent BitDance framework (Ai et al., 2026), which advances the same line of binary-token generative modeling as WeTok. Like the binary tokenizers used in WeTok/BitDance, GBQ also organizes the 32-bit representation into groups, although the grouping serves a different role in quantization; in both cases, the autoregressive generator models the complete 32-bit token jointly. We use our 800K-step GBQ tokenizer with the corresponding B/H generator architectures and adopt a generation setup based on the BitDance ImageNet configuration. As shown in Table 3, GBQ remains compatible with the BitDance generator and yields broadly comparable generation metrics at both model scales.

4.3 Ablation Studies

Geometry, coupling, and utilization. We first examine how learnable geometry and entropy regularization affect reconstruction and code utilization. To isolate orientation adaptation, we construct a global-rotation control that fixes $D_g = I$ and learns only an orthogonal transformation W with $W^\top W = I$, so the assignment remains separable. We report Util_8 following WeTok. Since this 8-bit metric saturates at 100% for all compared variants under our evaluation set, we additionally report Util_{16} over adjacent group pairs to provide greater discrimination in the larger joint state space.

As shown in Table 4, GBQ substantially improves reconstruction over both LFQ without entropy regularization and WeTok. In terms of utilization, WeTok reaches high Util_{16} with entropy regularization, whereas GBQ reaches 100% Util_{16} without it and changes little when the same regularizer is added. The global-rotation control remains close to LFQ, suggesting that orientation adaptation

Table 4: Geometry and utilization ablations under the matched 250K-step budget. The global-rotation control changes code orientation while preserving an identity Gram matrix, whereas GBQ permits non-diagonal interactions. Util_8 follows WeTok's 8-bit utilization metric; Util_{16} applies the same definition to adjacent 16-bit group pairs for greater discrimination when Util_8 saturates.

Method	Transform	Gram structure	rFID↓	PSNR↑	Util_8 ↑	Util_{16} ↑
LFQ w/o entropy	Identity	I	0.88	23.68	100%	66.7%
WeTok	Identity	I	0.86	23.90	100%	98.3%
LFQ + global rotation	Global orthogonal W	I	0.90	23.66	100%	–
GBQ (Ours)	Block-wise D_g	General	**0.69**	**24.50**	100%	**100%**
GBQ + entropy	Block-wise D_g	General	0.69	24.48	100%	100%

Table 5: Effect of block size b at fixed $d = 32$, increasing the scope of within-block vector assignment while preserving the total binary budget.

Block size b	rFID↓	PSNR↑	LPIPS↓
1	0.87	23.77	0.17
2	0.82	24.20	0.17
4	0.71	24.47	0.16
8	**0.69**	**24.50**	**0.16**

Table 6: Joint capacity scaling with G at fixed $b = 8$ ($d = 8G$), jointly increasing latent dimensionality and the global binary identity space.

Blocks G	Total bits d	rFID↓	PSNR↑
2	16	1.57	22.47
4	32	0.69	24.50
8	64	0.40	26.33
16	**128**	**0.29**	**28.09**

alone does not explain the performance of full GBQ. Additional details and training dynamics for the rotation control are provided in Appendix C.2.

Scaling behavior. We study two complementary scaling axes. First, we vary the block size b at fixed $d = 32$, increasing the scope of within-block interaction while keeping the total binary dimensionality unchanged. At $b = 1$, each D_g reduces to a scalar; consequently, the global transform D is diagonal and each $D_g^\top D_g$ is trivially diagonal. Reconstruction improves as b increases, consistent with the nested-family characterization in Appendix B.3. We use $b = 8$ by default. Second, we fix $b = 8$ and increase the number of blocks G, so that $d = 8G$. This jointly scales the latent dimensionality, global binary identity space, decoder input width, and transformation parameters, and reconstruction improves consistently along this joint capacity axis.

Efficiency. GBQ introduces exact local search over 2^b candidates, so we measure its end-to-end inference cost as the block size increases. At the default $b = 8$, peak memory is unchanged from the separable baseline and latency increases only marginally, indicating that the additional local search contributes little to the end-to-end tokenizer cost at the block sizes used here. This practical behavior does not remove the exponential dependence of exact local enumeration on b; Appendix B.4 gives the corresponding complexity analysis.

Table 7: End-to-end inference cost at 256×256 with batch size 16 under bf16 evaluation.

Method	b	Memory (GB)	Latency (ms/batch)
LFQ	1	11.12	219.97
GBQ	2	11.12	220.36
GBQ	4	11.12	220.53
GBQ (default)	8	11.12	221.00

5 Conclusion

We introduced Geometric Binary Quantization (GBQ), which replaces independent binary sign decisions with exact block-wise vector assignment over structured codebooks induced by learnable local linear transformations. By enabling non-separable within-block interactions while preserving exact factorization across blocks, GBQ improves reconstruction and maintains high code utilization without explicit entropy regularization, with lower training-image exposure than the reported WeTok setting and little practical overhead. More broadly, GBQ identifies assignment granularity as a useful design axis between scalar binary quantization and fully vector-level quantization.

AI USE STATEMENT

Generative AI tools were used to assist with language editing, sentence restructuring, presentation clarity, and checks of LaTeX formatting and manuscript organization. All AI-assisted content was reviewed by the authors. The authors verified the scientific claims, mathematical formulations, experimental results, and conclusions and take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

The formulation and optimization procedure of GBQ are described in Section 3, and the experimental setup and evaluation protocol are described in Section 4. The appendix provides additional architecture and training details, theoretical analyses, diagnostic definitions, ablation settings, and implementation details supporting reproducibility.

REFERENCES

Yuang Ai, Jiaming Han, Shaobin Zhuang, Weijia Mao, Xuefeng Hu, Ziyan Yang, Zhenheng Yang, Yali Wang, Huaibo Huang, Xiangyu Yue, and Hao Chen. BitDance: Scaling autoregressive generative models with binary tokens. *arXiv preprint arXiv:2602.14041*, 2026.

Idil Bilge Altun, Mert Onur Cakiroglu, Elham Buxton, Mehmet Dalkilic, and Hasan Kurban. LGQ: Learnable geometric quantization for image tokenization. *arXiv preprint arXiv:2602.16086*, 2026. doi: 10.48550/arXiv.2602.16086.

Roman Bachmann, Jesse Allardice, David Mizrahi, Enrico Fini, Oğuzhan Fatih Kar, Elmira Amirloo, Alaaeldin El-Nouby, Amir Zamir, and Afshin Dehghan. FlexTok: Resampling images into 1d token sequences of flexible length. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 2241–2292, 2025.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. MaskGIT: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11315–11325, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12873–12883, 2021.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Mingkai Jia, Wei Yin, Xiaotao Hu, Jiaxin Guo, Xiaoyang Guo, Qian Zhang, Xiao-Xiao Long, and Ping Tan. Mgvq: Could vq-vae beat vae? a generalizable tokenizer with multi-group quantization, 2025. URL https://arxiv.org/abs/2507.07997.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11523–11532, 2022.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Computer Vision – ECCV 2014*, pp. 740–755. Springer, 2014.

Zhuoyan Luo, Fengyuan Shi, Yixiao Ge, Yujiu Yang, Limin Wang, and Ying Shan. Open-MAGVIT2: An open-source project toward democratizing auto-regressive visual generation. *arXiv preprint arXiv:2409.04410*, 2024.

Chuofan Ma, Yi Jiang, Junfeng Wu, Jihan Yang, Xin Yu, Zehuan Yuan, Bingyue Peng, and Xiaojuan Qi. Unitok: A unified tokenizer for visual generation and understanding. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4306.

Fabian Mentzer, David Minnen, Eirikur Agustsson, and Michael Tschannen. Finite scalar quantization: VQ-VAE made simple. In *International Conference on Learning Representations*, 2024.

Ziqi Pang, Tianyuan Zhang, Fujun Luan, Yunze Man, Hao Tan, Kai Zhang, William T. Freeman, and Yu-Xiong Wang. RandAR: Decoder-only autoregressive visual generation in random orders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 45–55, 2025. doi: 10.1109/CVPR52734.2025.00014.

Ali Razavi, Aaron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10684–10695, 2022.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Advances in Neural Information Processing Systems*, volume 29, 2016.

Kyle Sargent, Kyle Hsu, Justin Johnson, Li Fei-Fei, and Jiajun Wu. Flow to the mode: Mode-seeking diffusion autoencoders for state-of-the-art image tokenization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 19471–19481, 2025.

Fengyuan Shi, Zhuoyan Luo, Yixiao Ge, Yujiu Yang, Ying Shan, and Limin Wang. Scalable image tokenization with index backpropagation quantization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16037–16046, 2025.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation. *arXiv preprint arXiv:2406.06525*, 2024.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2818–2826, 2016.

Zhentao Tan, Ben Xue, Jian Jia, Junhao Wang, Wencai Ye, Shaoyun Shi, Mingjie Sun, Wenjin Wu, Quan Chen, and Peng Jiang. SweetTok: Semantic-aware spatial-temporal tokenizer for compact video discretization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 23541–23550, 2025.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-2694.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

Bohan Wang, Zhongqi Yue, Fengda Zhang, Shuo Chen, Li'an Bi, Junzhe Zhang, Xue Song, Kennard Yanting Chan, Jiachun Pan, Weijia Wu, Mingze Zhou, Wang Lin, Kaihang Pan, Saining Zhang, Liyu Jia, Wentao Hu, Wei Zhao, and Hanwang Zhang. Discrete visual tokens of autoregression, by diffusion, and for reasoning, 2025.

Junke Wang, Yi Jiang, Zehuan Yuan, Bingyue Peng, Zuxuan Wu, and Yu-Gang Jiang. OmniTokenizer: A joint image-video tokenizer for visual generation. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0887.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4): 600–612, 2004. doi: 10.1109/TIP.2003.819861.

Pingyu Wu, Kai Zhu, Yu Liu, Longxiang Tang, Jian Yang, Yansong Peng, Wei Zhai, Yang Cao, and Zheng-Jun Zha. Towards sequence modeling alignment between tokenizer and autoregressive model. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=GT3obnZ5Fk.

Tianwei Xiong, Jun Hao Liew, Zilong Huang, Jiashi Feng, and Xihui Liu. GigaTok: Scaling visual tokenizers to 3 billion parameters for autoregressive image generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18770–18780, 2025.

Jiahui Yu, Xin Li, Jing Yu Koh, Han Zhang, Ruoming Pang, James Qin, Alexander Ku, Yuanzhong Xu, Jason Baldridge, and Yonghui Wu. Vector-quantized image modeling with improved VQGAN. In *International Conference on Learning Representations*, 2022.

Lijun Yu, José Lezama, Nitesh Bharadwaj Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Agrim Gupta, Xiuye Gu, Alexander G. Hauptmann, Boqing Gong, Ming-Hsuan Yang, Irfan Essa, David A. Ross, and Lu Jiang. Language model beats diffusion – tokenizer is key to visual generation. In *International Conference on Learning Representations*, 2024.

Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 18431–18441, 2025. doi: 10.1109/ICCV51701.2025.01713.

Denis Zavadski, Nikita Philip Tatsch, and Carsten Rother. Pqgan: Product-quantised image representation for high-quality image synthesis. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=D8oqcochgq.

Borui Zhang, Qihang Rao, Wenzhao Zheng, Jie Zhou, and Jiwen Lu. Quantize-then-rectify: Efficient vq-vae training, 2025a. URL https://arxiv.org/abs/2507.10547.

Jiayou Zhang, Yifan Shen, Guangyi Chen, Le Song, and Eric P. Xing. Dimensional collapse in VQVAEs: Evidence and remedies. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. doi: 10.52202/085713-5693.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 586–595, 2018.

Yue Zhao, Yuanjun Xiong, and Philipp Krähenbühl. Image and video tokenization with binary spherical quantization. In *International Conference on Learning Representations*, 2025.

Yue Zhao, Hanwen Jiang, Zhenlin Xu, Chutong Yang, Ehsan Adeli, and Philipp Krähenbühl. Spherical leech quantization for visual tokenization and generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Anlin Zheng, Xin Wen, Xuanyang Zhang, Chuofan Ma, Tiancai Wang, Gang Yu, Xiangyu Zhang, and Xiaojuan Qi. Vision foundation models as effective visual tokenizers for autoregressive generation. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-2099.

Chuanxia Zheng, Tung-Long Vuong, Jianfei Cai, and Dinh Phung. MoVQ: Modulating quantized vectors for high-fidelity image generation. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Lei Zhu, Fangyun Wei, Yanye Lu, and Dong Chen. Scaling the codebook size of VQ-GAN to 100,000 with a utilization rate of 99%. In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-0401.

Yongxin Zhu, Bocheng Li, Yifei Xin, Zhihua Xia, and Linli Xu. Addressing representation collapse in vector quantized models with one linear layer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 22968–22977, 2025.

Shaobin Zhuang, Yuang Ai, Jiaming Han, Weijia Mao, Xiaohui Li, Fangyikang Wang, Xiao Wang, Yan Li, Shanchuan Lin, Kun Xu, Zhenheng Yang, Huaibo Huang, Xiangyu Yue, Hao Chen, and Yali Wang. UniWeTok: An unified binary tokenizer with codebook size 2^{128} for unified multimodal large language model. *arXiv preprint arXiv:2602.14178*, 2026a.

Shaobin Zhuang, Yiwei Guo, Fangyikang Wang, Canmiao Fu, Zhipeng Huang, Zeyue Tian, Xiaohui Li, Ying Zhang, Chen Li, and Yali Wang. WeTok: Powerful discrete tokenization for high-fidelity visual reconstruction. In *International Conference on Learning Representations*, 2026b.

A Implementation and Evaluation Details

We build GBQ on the WeTok tokenizer backbone and change only the quantization module in the controlled tokenizer comparisons. The encoder, decoder, discriminator, latent resolution, and reconstruction objectives otherwise follow the baseline configuration. Unless otherwise specified, the default GBQ tokenizer uses latent dimensionality $d = 32$ and block size $b = 8$, yielding $G = d/b = 4$ blocks. Each block contains $2^b = 256$ binary states and uses a learnable transformation matrix $D_g \in \mathbb{R}^{8\times8}$. Across all blocks, GBQ therefore introduces only $Gb^2 = db = 256$ transformation parameters. The tokenizer uses a base channel width of 128, channel multipliers $[1, 1, 2, 2, 4]$, and four residual blocks, following the baseline architecture.

Optimization. All tokenizer models are optimized with Adam using $(\beta_1, \beta_2) = (0.5, 0.9)$, zero weight decay, and a constant learning rate of 10^{-4} (Kingma & Ba, 2015). We use an exponential moving average of model parameters. Training images are randomly cropped to 256×256. The default global batch size is 128, with a per-GPU batch size of 16 on 8 GPUs. Ablations are trained for 250K steps unless otherwise specified, while the headline GBQ models are trained for 400K or 800K steps. No learning-rate warmup is used.

Quantization objective. For GBQ, the quantization loss is

$$\mathcal{L}_{\text{quant}} = \|\text{sg}(z) - q\|_2^2 + \beta \, \|z - \text{sg}(q)\|_2^2 , \tag{13}$$

where the squared errors are averaged over spatial positions and channels. The default commitment coefficient is $\beta = 0.05$, and the quantization loss is weighted by $\lambda_{\text{quant}} = 0.25$ in the complete tokenizer objective. Unless otherwise specified, GBQ uses no explicit entropy regularizer.

Reconstruction objective. The decoder is trained with the same pixel, perceptual, and adversarial objectives as the baseline. Specifically,

$$\mathcal{L}_{\text{AE}} = \|x - \hat{x}\|_1 + \lambda_p \, \text{LPIPS}(x, \hat{x}) + \lambda_{\text{adv}} \, \mathcal{L}_G(\hat{x}), \tag{14}$$

where the perceptual term uses LPIPS with a VGG backbone. Following the released baseline implementation, the effective loss coefficients are

$$\lambda_p = 1.0, \qquad \lambda_{\text{adv}} = 0.1, \qquad \lambda_{\text{quant}} = 0.25.$$

The adversarial objective is enabled from the beginning of training. The complete tokenizer objective is

$$\mathcal{L} = \mathcal{L}_{\text{AE}} + \lambda_{\text{quant}} \mathcal{L}_{\text{quant}}. \tag{15}$$

Autoregressive generation. For class-conditional ImageNet generation, we adopt the BitDance autoregressive framework (Ai et al., 2026) and use its ImageNet configuration as the basis for our generation setup. Like WeTok/BitDance, GBQ also uses a grouped binary representation, although the grouping serves a different role in the quantizer; in both cases, the autoregressive generator jointly models the complete 32-bit token. We use the 800K-step GBQ tokenizer with $16\times$ spatial downsampling and latent dimensionality $d = 32$, together with the BitDance-B and BitDance-H generator architectures. Models are trained on ImageNet-1K at 256×256 resolution for 800 epochs with a learning rate of 6×10^{-4}, a global batch size of 1024, and an EMA decay of 0.9999. Sampling uses the BitDance ImageNet evaluation setup. Generation quality is evaluated on 50K class-conditional samples using FID, IS, Precision, and Recall.

A.1 Evaluation Protocol

Datasets and preprocessing. Tokenizer training is performed on the ImageNet-1K training set, and reconstruction is evaluated on all 50K ImageNet validation images. Unless otherwise specified, images are evaluated at 256×256 resolution following the baseline protocol. For zero-shot reconstruction, the ImageNet-trained tokenizer is evaluated without fine-tuning on the 5,000 images of MS-COCO 2017 validation, both after resizing to 256×256 and at their original resolution.

Reconstruction metrics. We report reconstruction FID (rFID), LPIPS-VGG, PSNR, and SSIM. rFID measures the distributional discrepancy between reconstructed and real images using Inception-V3 features (Heusel et al., 2017; Szegedy et al., 2016). LPIPS-VGG (Zhang et al., 2018) measures perceptual similarity using VGG features, while PSNR and SSIM (Wang et al., 2004) measure pixel-level fidelity and structural similarity, respectively. Lower rFID and LPIPS and higher PSNR and SSIM indicate better reconstruction.

Generation metrics. For class-conditional generation, we report FID (Heusel et al., 2017), Inception Score (IS) (Salimans et al., 2016), Precision, and Recall (Kynkäänniemi et al., 2019) over 50K generated images using the same evaluation protocol for all controlled comparisons.

Code utilization. Following WeTok, we measure code utilization over 8-bit groups. For group g, let $\mathcal{A}_g^{(8)}$ denote the set of 8-bit states activated at least once on the evaluation set. We define

$$\mathrm{Util}_8 = \frac{1}{G}\sum_{g=1}^{G}\frac{|\mathcal{A}_g^{(8)}|}{2^8}. \tag{16}$$

Each 8-bit group contains only $2^8 = 256$ states, so this statistic saturates at 100% for the compared variants under our evaluation set. To obtain a more discriminative measure of joint occupancy, we additionally concatenate adjacent pairs of 8-bit groups. Let $\mathcal{A}_p^{(16)}$ denote the set of observed 16-bit states for the p-th adjacent pair. For the default $d = 32$ setting, we define

$$\mathrm{Util}_{16} = \frac{1}{G/2}\sum_{p=1}^{G/2}\frac{|\mathcal{A}_p^{(16)}|}{2^{16}}. \tag{17}$$

Util_{16} therefore applies the same utilization definition to a larger joint state space and provides greater discrimination when the WeTok-style Util_8 metric is saturated. For explicit VQ methods, utilization denotes the conventional fraction of codebook entries activated at least once.

B Geometry, Expressivity, and Tractability of GBQ

This section characterizes the geometry, expressivity, and computational structure of GBQ. We first show that the Gram matrix of the learnable transformation jointly determines the interaction structure of nearest-code assignment and the geometry of the induced binary codes. In particular, the same diagonal-Gram condition characterizes both additive separability of the nearest-code energy and equal-radius binary realizations, yielding a concrete expressivity restriction of separable binary geometry and a strict separation example for coupled geometry. We then formalize the block-wise quantizer family and its exact-search complexity, and finally relate the fixed-representation analysis to end-to-end co-adaptation and the straight-through estimator.

For the geometric analysis, we consider a single block with $z \in \mathbb{R}^b$, $s \in \mathcal{B}_b = \{-1, +1\}^b$, and code $c(s) = Ds$, where $D \in \mathbb{R}^{b\times b}$. We write

$$M = D^\top D, \qquad h = D^\top z.$$

B.1 Transformation-Induced Geometry and Assignment

The 2^b binary states are realized as the structured code set

$$\mathcal{C}(D) = \{Ds : s \in \mathcal{B}_b\}.$$

Unlike an unconstrained VQ codebook, these vectors are jointly determined by a single transformation D, and they retain central symmetry since $c(-s) = -c(s)$.

Geometry among binary states. For any $s, t \in \mathcal{B}_b$,

$$\|c(s) - c(t)\|_2^2 = (s-t)^\top M(s-t). \tag{18}$$

Thus $M = D^\top D$ determines the pairwise geometry of the induced binary codes. For LFQ, $M = I$, so

$$\|s - t\|_2^2 = 4H(s,t),$$

where $H(s,t)$ is Hamming distance. A learned M relaxes this prescribed relationship. The same Gram matrix also determines whether the nearest-code energy decomposes across binary variables.

Proposition 1 (Separable nearest-code energy). The nearest-code energy over $\mathcal{C}(D)$ is additively separable in the binary variables for every z if and only if $M = D^\top D$ is diagonal.

Proof. Using $s_i^2 = 1$,

$$\|z - Ds\|_2^2 = \|z\|_2^2 - 2h^\top s + \mathrm{tr}(M) + 2\sum_{i<j} M_{ij} s_i s_j. \tag{19}$$

If M is diagonal, the interaction terms vanish and the minimizer decomposes coordinate-wise as

$$s_i^\star = \mathrm{sign}(h_i), \qquad h = D^\top z, \tag{20}$$

up to arbitrary tie breaking when $h_i = 0$.

Conversely, if $M_{ij} \neq 0$ for some $i \neq j$, then, holding the remaining variables fixed,

$$E(+,+) - E(+,-) - E(-,+) + E(-,-) = 8M_{ij}.$$

Any function additive in the individual binary variables has zero mixed difference, so a nonzero off-diagonal entry prevents additive separability. □

Proposition 1 distinguishes geometric adaptation from coupled assignment. A diagonal Gram matrix permits changes in orientation and per-direction scale while preserving independent binary decisions; an orthogonal transformation is the special case $D^\top D = I$. A nonzero off-diagonal entry instead introduces an interaction between binary variables. For example, with two variables,

$$s_1^\star = \mathrm{sign}(h_1 - M_{12} s_2),$$

making the dependence on the other binary state explicit.

B.2 Expressivity of Separable and Coupled Binary Geometry

Separability imposes a specific geometric constraint: all binary realizations must lie at the same Euclidean radius from the origin.

Lemma 1 (Equal-radius characterization). All binary codes Ds, $s \in \mathcal{B}_b$, have the same Euclidean norm if and only if $D^\top D$ is diagonal.

Proof. Using $s_i^2 = 1$,

$$\|Ds\|_2^2 = \mathrm{tr}(M) + 2\sum_{i<j} M_{ij} s_i s_j. \tag{21}$$

If M is diagonal, the second term vanishes and $\|Ds\|_2^2 = \mathrm{tr}(M)$ for every binary state.

Conversely, suppose $\|Ds\|_2^2 = r^2$ for every s. For uniformly distributed $s \in \mathcal{B}_b$ and any $i \neq j$,

$$\mathbb{E}_s\left[\|Ds\|_2^2 s_i s_j\right] = 2M_{ij},$$

whereas the constant-norm assumption gives

$$\mathbb{E}_s[r^2 s_i s_j] = 0.$$

Hence $M_{ij} = 0$ for every $i \neq j$. □

Combining Proposition 1 and Lemma 1 gives

nearest-code energy is additively separable for every z
$\iff$ $D^\top D$ is diagonal
$\iff$ $\|Ds\|_2$ is constant over $s \in \mathcal{B}_b$.

Thus a separable learnable geometry may rotate and rescale its principal directions, but all 2^b binary realizations remain on a common sphere centered at the origin. Non-diagonal Gram structure simultaneously introduces assignment interactions and allows state-dependent code radii.

Corollary 1 (Radial distortion bound). For any separable binary geometry $c(s) = Ds$,

$$\mathbb{E} \min_{s \in \mathcal{B}_b} \|z - Ds\|_2^2 \geq \mathrm{Var}(\|z\|_2). \tag{22}$$

For a block-wise quantizer whose blocks are all separable, this extends to

$$\mathbb{E}\left[\sum_{g=1}^{G} \min_{s^{(g)} \in \mathcal{B}_b} \left\| z^{(g)} - D_g s^{(g)} \right\|_2^2\right] \geq \sum_{g=1}^{G} \mathrm{Var}\left(\|z^{(g)}\|_2\right). \tag{23}$$

Proof. By Lemma 1, all codes in a separable block have a common radius r. The reverse triangle inequality gives

$$\min_s \|z - Ds\|_2^2 \geq (\|z\|_2 - r)^2.$$

Therefore,

$$\mathbb{E} \min_s \|z - Ds\|_2^2 \geq \min_{r'} \mathbb{E}(\|z\|_2 - r')^2 = \mathrm{Var}(\|z\|_2).$$

Applying the same argument to each block and summing yields Eq. (23); no statistical independence among blocks is required. □

By contrast, Eq. (21) shows that nonzero off-diagonal entries allow the radius to depend on the binary state. This yields a strict expressivity gap even when binary-identity utilization is already maximal.

Theorem 1 (Strict separation from the separable family). Let

$$D_0 = \begin{pmatrix} a & \rho \\ \rho & a \end{pmatrix}, \qquad a > \rho > 0,$$

and let s be uniform over $\{-1, +1\}^2$, with latent features $z = D_0 s$. Then fixed LFQ uses all four binary identities and has expected distortion

$$2\big[(a-1)^2 + \rho^2\big];$$

every separable binary geometry has expected distortion at least

$$2\rho^2 > 0;$$

and the coupled geometry $D = D_0$ uses the same four identities and achieves zero distortion.

Proof. The four feature vectors are

$$D_0 s \in \{\pm(a+\rho)(1,1),\ \pm(a-\rho)(1,-1)\},$$

whose norms are $\sqrt{2}(a+\rho)$ and $\sqrt{2}(a-\rho)$, each with probability $1/2$. Hence

$$\mathrm{Var}(\|z\|_2) = 2\rho^2,$$

and Corollary 1 lower-bounds the distortion of every separable geometry by $2\rho^2$.

Since $a > \rho > 0$,

$$\mathrm{sign}(D_0 s) = s,$$

so fixed LFQ selects all four identities. Moreover,

$$\mathbb{E}_s \|(D_0 - I)s\|_2^2 = 2\big[(a-1)^2 + \rho^2\big].$$

Finally, choosing $D = D_0$ makes each feature exactly equal to its corresponding code, yielding zero distortion with the same binary identity set. □

Theorem 1 is an existence result for the constructed feature distribution rather than an end-to-end performance guarantee. Within this scope, it isolates an expressivity gap independent of identity utilization: full use of the binary states does not remove the geometric restriction of separable assignment.

B.3 Block Factorization and Assignment Granularity

Writing $D = \mathrm{BlockDiag}(D_1, \ldots, D_G)$ for the block-wise transformation in Section 3.2, we formalize two consequences below.

Proposition 2 (Exact block factorization). The set of global nearest-code minimizers factorizes exactly:

$$\operatorname*{argmin}_{s\in\{-1,+1\}^d} \|z - Ds\|_2^2 = \prod_{g=1}^{G} \operatorname*{argmin}_{s^{(g)}\in\{-1,+1\}^b} \left\| z^{(g)} - D_g s^{(g)} \right\|_2^2, \tag{24}$$

where the product denotes the Cartesian product of the local minimizer sets.

Proof. By Eq. (8), the objective is a sum of block-local terms, while the feasible state space factorizes as

$$\{-1,+1\}^d = \prod_{g=1}^{G} \{-1,+1\}^b.$$

Each term therefore attains its minimum independently, and concatenating any collection of local minimizers yields a global minimizer. □

Corollary 2 (Nested block families). Consider two block sizes $b_1 < b_2$ at fixed d, where b_2 is a multiple of b_1 and the b_2-partition is obtained by merging blocks of the b_1-partition. If $\mathcal{F}_b$ denotes the quantizer family realizable with block size b, then

$$\mathcal{F}_{b_1} \subseteq \mathcal{F}_{b_2}. \tag{25}$$

Proof. Within each merged b_2-dimensional block, choose the transformation to be block diagonal with the constituent b_1-dimensional transformations on its diagonal. This exactly reproduces the smaller-block geometry and assignment, so every member of $\mathcal{F}_{b_1}$ is realizable in $\mathcal{F}_{b_2}$. □

This nesting concerns quantizer expressivity only; it does not imply monotonic optimization behavior as b increases. Larger blocks preserve compatible smaller-block constructions while additionally allowing interactions across their former boundaries.

B.4 Exact-Search Complexity

From Eq. (6), GBQ evaluates 2^b candidates in each of $G = d/b$ blocks, for a total of

$$\frac{d}{b} 2^b$$

candidate evaluations per latent vector. Since each distance is b-dimensional, direct evaluation requires

$$\mathcal{O}(d\, 2^b) \tag{26}$$

arithmetic, compared with $\mathcal{O}(d\, 2^d)$ for exact search under a fully coupled d-dimensional binary geometry.

Constructing all structured local codebooks $C_g = D_g S_b$ costs $\mathcal{O}(db\, 2^b)$ arithmetic whenever the transformations change, and this cost is shared across all spatial latent positions using those codebooks. At inference, the learned local codebooks can be precomputed. For the default $d = 32$, $b = 8$ configuration, the search comprises

$$4 \times 2^8 = 1024$$

local candidate evaluations per latent vector. Thus the exponential dependence is confined to the local block size b; the end-to-end measurements in Table 7 show that this overhead remains small at the block sizes used in our experiments.

B.5 Relation to End-to-End Optimization

The preceding results characterize quantizer-side expressivity for a fixed latent representation. Under end-to-end training, the encoder distribution also adapts, as discussed in Section 3.3; GBQ retains this representation-side adaptation while additionally adapting the quantizer-side geometry.

For a fixed feature distribution $p(z)$ and the same block partition as in Section 3.2, define

$$\mathcal{D}_{\mathrm{GBQ}}(p) = \min_{\{D_g\}} \mathbb{E}_{z\sim p(z)} \left[\sum_{g=1}^{G} \min_{s^{(g)}\in\mathcal{B}_b} \left\| z^{(g)} - D_g s^{(g)} \right\|_2^2 \right], \tag{27}$$

$$\mathcal{D}_{\mathrm{LFQ}}(p) = \mathbb{E}_{z\sim p(z)} \left[\sum_{g=1}^{G} \min_{s^{(g)}\in\mathcal{B}_b} \left\| z^{(g)} - s^{(g)} \right\|_2^2 \right]. \tag{28}$$

Because $D_g = I_b$ is feasible for GBQ,

$$\mathcal{D}_{\mathrm{GBQ}}(p) \le \mathcal{D}_{\mathrm{LFQ}}(p). \tag{29}$$

Thus LFQ is a special case of GBQ, and for any fixed feature distribution $p(z)$, the optimal GBQ distortion cannot exceed that of LFQ. This statement concerns quantizer-side expressivity; during joint training, both the feature distribution and, for GBQ, the induced code geometry co-adapt.

STE gradient discrepancy. Using the STE in Eq. (9), let $\phi(u)$ denote the downstream reconstruction objective as a function of the decoder input. The gradient propagated through the STE and its continuous-latent reference are

$$g_{\mathrm{ST}} = \nabla_u \phi(u)\big|_{u=q}, \qquad g_{\mathrm{ref}} = \nabla_u \phi(u)\big|_{u=z}.$$

If the downstream gradient field is L_∇-Lipschitz, then

$$\|g_{\mathrm{ST}} - g_{\mathrm{ref}}\|_2 \le L_\nabla \|q - z\|_2. \tag{30}$$

Hence a smaller forward quantization residual tightens a local upper bound on the Euclidean discrepancy between the STE gradient and its continuous-latent reference. This bound does not by itself imply guaranteed convergence, improved final reconstruction, or monotonic improvement in gradient cosine similarity. The diagnostics in Section C.1 measure the corresponding residual and gradient alignment empirically.

C Additional Diagnostics and Analysis

This section provides additional measurements supporting the mechanism and controlled comparisons in the main paper. We first report the diagnostics and matched-budget training trajectories underlying Fig. 1, followed by implementation details and training behavior for the global-rotation control.

C.1 Mechanism Diagnostics and Training Dynamics

We provide the measurements underlying the diagnostic and early-training comparisons in Fig. 1. We first examine the forward quantization residual and the alignment between the latent-space gradient propagated through the straight-through estimator (STE) and its continuous-latent reference. We then report the matched-budget reconstruction trajectories used in the early-training comparison.

Relative quantization residual. For each latent token, we measure

$$r_{\mathrm{quant}} = \frac{\|q - z\|_2}{\|z\|_2}, \tag{31}$$

where z is the encoder feature before quantization and q is the continuous realization of the selected discrete code. We report the median over the evaluated latent tokens. Under the matched evaluation used in Fig. 1, the median relative residual decreases from 0.48 for WeTok to 0.15 for GBQ, indicating that the selected GBQ codes more closely match the encoder features in continuous latent space.

Table 8: Matched-budget reconstruction dynamics on the ImageNet-1K 256×256 validation set. All models use the same controlled tokenizer architecture and are evaluated on the 50K validation images.

Method	Metric	50K	100K	150K	200K	250K
GBQ (Ours)	rFID ↓	**1.115**	**0.867**	**0.766**	**0.717**	**0.688**
	PSNR ↑	**24.06**	**24.28**	**24.39**	**24.46**	**24.50**
WeTok	rFID ↓	1.474	1.114	0.973	0.913	0.857
	PSNR ↑	23.36	23.64	23.78	23.85	23.90
LFQ w/o entropy	rFID ↓	1.914	1.297	1.102	0.936	0.878
	PSNR ↑	22.65	23.07	23.31	23.59	23.68

STE–reference gradient alignment. Let g_{ST} denote the gradient with respect to the latent input propagated through the identity STE, and let g_{ref} denote the corresponding gradient obtained when the unquantized latent z is passed directly to the decoder. We measure their cosine similarity,

$$A_{\mathrm{grad}} = \frac{\langle g_{\mathrm{ST}}, g_{\mathrm{ref}} \rangle}{\|g_{\mathrm{ST}}\|_2 \|g_{\mathrm{ref}}\|_2}. \tag{32}$$

The median alignment increases from 0.61 for WeTok to 0.89 for GBQ. Together with the smaller forward residual, this observation is consistent with the local relationship in Eq. (30): reducing $\|q - z\|_2$ tightens an upper bound on the Euclidean discrepancy between the STE gradient and its continuous-latent reference. The bound does not imply a guarantee on cosine similarity, so the alignment is reported as an empirical diagnostic.

Matched-budget training dynamics. Table 8 reports the rFID and PSNR values underlying the early-training curves in Fig. 1. GBQ achieves better rFID and PSNR than the matched WeTok and LFQ baselines at every reported checkpoint over the 250K-step controlled training budget.

C.2 Global-Rotation Control

The global-rotation control isolates whether adapting the orientation of the binary code space is sufficient to obtain the improvement of GBQ. We fix the block transformations to $D_g = I_b$ and learn only a global orthogonal transformation W, initialized as the identity. We parameterize W using `torch.nn.utils.parametrizations.orthogonal`, which maps unconstrained underlying parameters continuously to an orthogonal matrix and maintains

$$W^\top W = I$$

throughout optimization. All remaining settings match the default GBQ configuration ($b = 8$, $\beta = 0.05$, and no entropy regularization), and the model is trained from scratch for 250K steps.

Because $W^\top W = I$, this control can rotate the binary codebook relative to the encoder features while preserving an identity Gram matrix. The pairwise interaction terms analyzed in Section B.1 therefore remain absent.

Table 9 reports the validation reconstruction-loss trajectories. The global-rotation control remains close to the factorized LFQ baseline throughout training, whereas GBQ consistently reaches a substantially lower reconstruction loss. This behavior is consistent with the reconstruction metrics in Table 4: adapting code orientation alone is insufficient to explain the improvement of the full block-wise geometry.

D Additional Qualitative Results

Figure 3 compares reconstructions from the controlled WeTok baseline and GBQ on ImageNet validation images. Each triplet contains the ground-truth image, the baseline reconstruction, and the GBQ reconstruction. The examples illustrate differences in fine structures such as small text and markings, thin edges, facial details, and local textures.

Table 9: Validation reconstruction loss (ℓ_1, lower is better) for the global-rotation control and matched tokenizer baselines. Values are reported every 50K training steps using EMA weights.

Method	50K	100K	150K	200K	250K
LFQ w/o entropy	0.0990	0.0941	0.0917	0.0889	0.0880
LFQ + global rotation	0.0974	0.0934	0.0918	0.0900	0.0888
WeTok	0.0914	0.0883	0.0870	0.0864	0.0859
GBQ (Ours)	**0.0848**	**0.0826**	**0.0816**	**0.0811**	**0.0805**

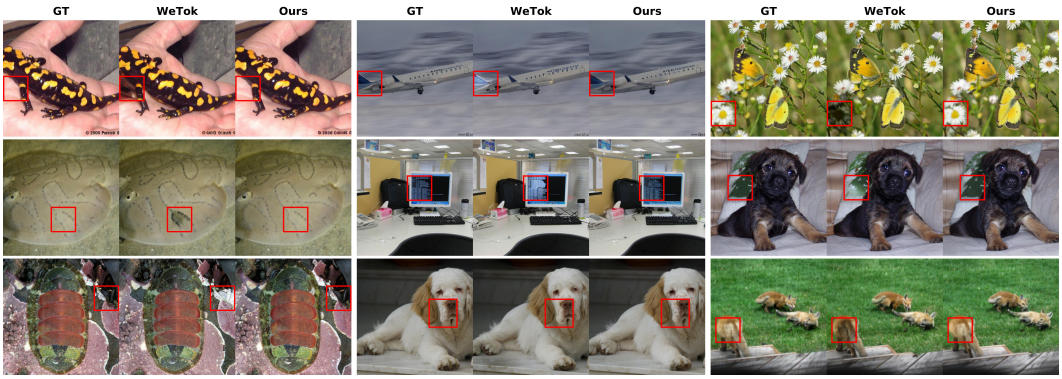

Figure 3: Qualitative reconstruction comparison on ImageNet validation images. Each triplet shows the ground truth, the controlled WeTok reconstruction, and the GBQ reconstruction. Highlighted regions indicate representative local differences in reconstruction quality.