TAMING THE ENTROPY CLIFF: VARIABLE CODEBOOK SIZE QUANTIZATION FOR AUTOREGRESSIVE VISUAL GENERATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Most discrete visual tokenizers rely on a default design: every position in the sequence shares the same codebook. Researchers try to scale the codebook size K to get better reconstruction performance. Such a constant-codebook design can consume the finite-data entropy budget very quickly. We observe that the per-position empirical conditional entropy decays so quickly along the sequence that, after a few positions, the conditional distribution becomes essentially deterministic. On ImageNet with $K = 16384$, this happens after only about 2 out of 256 positions, when most prefixes already identify individual images in the dataset. We call this phenomenon the **Entropy Cliff**. Under full conditional utilization, larger codebooks exhaust the finite-data entropy budget in fewer positions. Interestingly, the empirical entropy decreases more gradually in our language comparison, where the effective entropy per position stays below the codebook capacity. To address this, we propose **Variable Codebook Size Quantization (VCQ)**, where the codebook size K_t grows monotonically along the sequence from $K_{\min} = 2$ to $K_{\max}$, leaving the loss function, parameter count, and AR training procedure unchanged. With a vanilla autoregressive Transformer and standard next-token prediction, VCQ-Base achieves **gFID w/o CFG 14.80** on ImageNet 256×256, compared with 27.98 and 18.16 for constant codebooks of size 16384 and 8192, respectively. Scaled up, it reaches **gFID 1.71** with 684M autoregressive parameters, without any extra training techniques such as semantic regularization or causal alignment. The extreme information bottleneck at $K_{\min} = 2$ encourages semantic concentration in early encoder features: a linear probe on features from only the first 10 positions reaches **43.8% top-1 accuracy** on ImageNet, compared to 27.1% for uniform codebooks. Ultimately, these results show that what matters is not only the total capacity of the codebook, but also how that capacity is distributed and organized.

1 INTRODUCTION

Discrete tokenizers are the foundation of autoregressive visual generation. From VQ-VAE (van den Oord et al., 2017) and VQ-GAN (Esser et al., 2021) to more recent works such as LlamaGen (Sun et al., 2024), GigaTok (Xiong et al., 2025), a persistent trend has been to increase the codebook size K for better reconstruction quality; a size of 16k or more has become standard. These works differ in training objectives, network architectures, and inference strategies, yet they share one design choice with striking unanimity: every position in the sequence can potentially use all tokens in the same codebook size. We refer to this default codebook setting as the ***constant codebook size*** quantization or uniform codebooks.

However, the uniform codebook has significant flaws (Fig. 2(a)). We encode ImageNet images into 256 tokens with a uniform codebook of $K = 16384$ (the codebook from LlamaGen (Sun et al., 2024)) and calculate empirical conditional entropy from prefix frequencies. At $t = 0$, the conditional entropy is about 14 bits. By $t = 2$, it falls below 1 bit and approaches zero thereafter. Most observed prefixes then identify individual images in the dataset, each with a fixed empirical continuation. We call this phenomenon the ***Entropy Cliff***. On the full ImageNet training set, 96.6% of images already have unique two-token prefixes (Appendix E).

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.

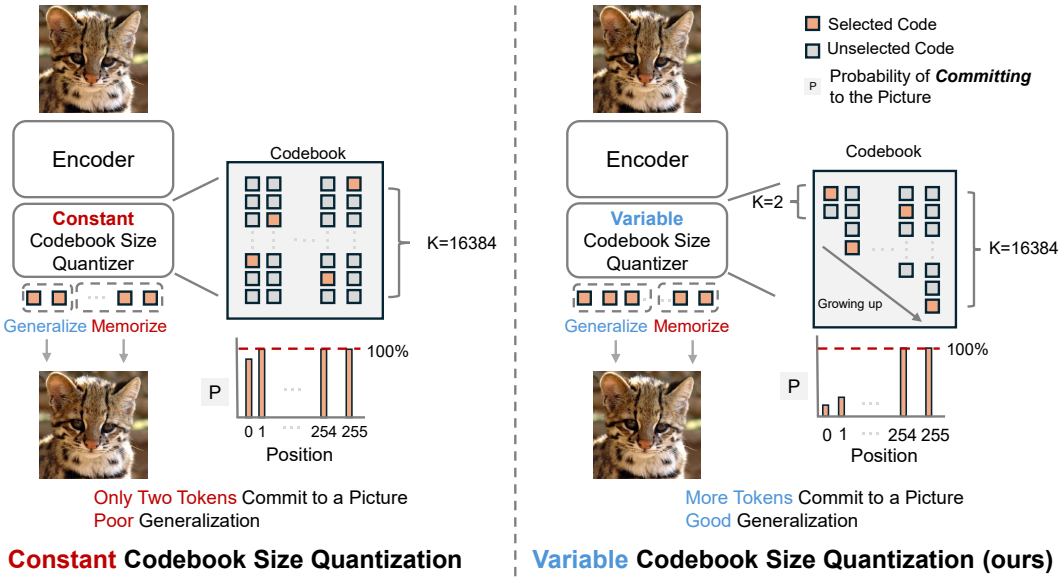

Figure 1: **VCQ architecture. Left**: Constant Codebook Size Quantization, where every position t shares the same codebook of size K, with a full-utilization capacity floor of $t^*_{\text{floor}} = \lceil \log_2 N / \log_2 K \rceil$ positions. **Right**: our Variable Codebook Size Quantization (VCQ), where the codebook size K_t grows monotonically from $K_{\min} = 2$ to $K_{\max}$ along the sequence. Early positions use small codebook size to impose a strong information bottleneck and delay the entropy cliff, while later positions use larger codebook size to preserve reconstruction fidelity. We implement VCQ with a single shared codebook of $K_{\max}$ entries, where position t selects the nearest neighbor from only the first K_t entries—introducing no extra parameters and leaving the loss function and AR training procedure unchanged.

Regularization such as weight decay acts on the AR model, while codebook size controls the tokenized representation. A naive remedy is to globally reduce the codebook size, which limits early-position capacity and can delay the empirical cliff. However, uniformly shrinking the codebook also restricts later positions, where richer local detail is required, which can reduce reconstruction fidelity. This motivates controlling how capacity is distributed across the autoregressive sequence. Early positions primarily determine high-level semantic structure and are particularly sensitive to excessive discrete freedom, while later positions benefit from larger codebooks to model fine-grained details.

This observation motivates **Variable Codebook Size Quantization (VCQ)**, which assigns smaller codebooks to early positions to stabilize global autoregressive generation, and progressively increases the codebook size at later positions to preserve reconstruction quality and local detail. Specifically, K_t increases monotonically from $K_{\min} = 2$ to $K_{\max}$ (Eq. 5). At early positions, smaller codebooks impose a stronger information bottleneck that delays the onset of the entropy cliff; at later positions, larger codebooks provide sufficient capacity to preserve reconstruction quality.

We observe advantages of VCQ along three axes:

1. **Delayed entropy cliff.** VCQ spreads the empirical entropy across more positions by limiting early-position capacity. The finite-data analysis in §2 explains how the codebook schedule changes the capacity floor, while the actual entropy curve also depends on conditional utilization.
2. **Improved generation quality.** Compared with the constant $K = 16384$ baseline, VCQ-Base reduces gFID w/o CFG from 27.98 to 14.80 (-47%), and, with CodebookSize-Aware CFG, gFID from 6.43 to 4.79 (-26%). Generation quality improves with scale: VCQ-Large achieves gFID=1.71 with a 684M-parameter AR model, without semantic alignment or causal regularization (§4.2).
3. **Semantic concentration.** $K_{\min} = 2$ encourages global semantic information to concentrate in early encoder features. Using features from only the first 10 positions (3.9% of the sequence), linear probing top-1 accuracy on ImageNet classification reaches 43.8%, compared to 27.1% for constant-codebook (§4.4).

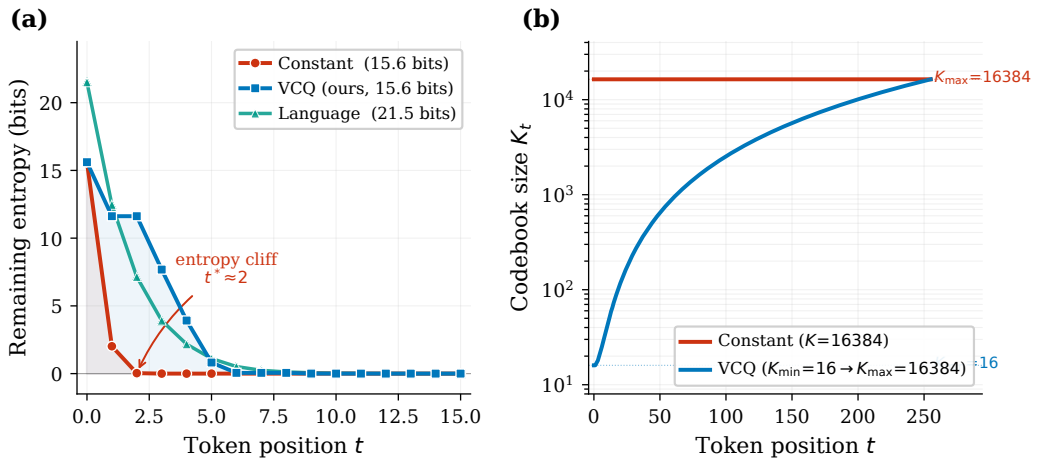

Figure 2: **Entropy cliff and codebook schedule.** (a) Empirical remaining budget estimated from token-prefix frequencies. The image curves are measured on ImageNet. Constant ($K = 16384$) approaches zero after about 2 positions, while VCQ spreads the decrease across more positions. The LAMBADA comparison decreases more gradually. For clarity, this plot uses a Power schedule with $K_{\min} = 16$. (b) Codebook size schedule: Constant is fixed at $K = 16384$; VCQ increases gradually from $K_{\min}$ to $K_{\max}$.

2 The Entropy Cliff of Constant Codebook Size Quantization

Notation. We consider the empirical distribution of N equally weighted tokenized samples with sequence length L. Positions are indexed from 0. Position t has at most K_t possible codes, and $H_{\text{emp}}(x_t \mid x_{<t})$ denotes its empirical conditional entropy.

2.1 A Finite-Data Entropy Budget

Define the remaining budget after a length-t prefix as

$$R(t) = \log_2 N - \sum_{i<t} H_{\text{emp}}(x_i \mid x_{<i}). \tag{1}$$

Proposition 1 (Finite-data entropy budget)**.** *Let* $I(t) = \sum_{i<t} \log_2 K_i$. *Then*

$$\max\big(0, \log_2 N - I(t)\big) \le R(t) \le \log_2 N, \tag{2}$$

and the entire remaining token sequence has empirical conditional entropy at most $R(t)$. *If all length-t prefixes are distinct,* $R(t) = 0$ *and all subsequent empirical continuations are deterministic.*

The proof and full derivation are in Appendix E.

For a constant codebook, the capacity floor reaches zero at

$$t^*_{\text{floor}} = \left\lceil \frac{\log_2 N}{\log_2 K} \right\rceil. \tag{3}$$

To describe the actual decay, let $A_t = \min(\log_2 K_t, R(t))$ and define conditional utilization as $\rho_t = H_{\text{emp}}(x_t \mid x_{<t})/A_t \in [0, 1]$ for $A_t > 0$, with $\rho_t = 0$ otherwise. Then $R(t+1) = R(t) - \rho_t A_t$. Full conditional utilization ($\rho_t = 1$ whenever $A_t > 0$) makes the budget follow the capacity floor. Lower utilization can delay the measured cliff.

Information redistribution by VCQ. VCQ keeps K_t small at early positions and increases it toward $K_{\max}$. This slows the growth of $I(t)$ and raises the early-position floor on $R(t)$. Under full conditional utilization, the floor reaches zero at the first t for which $I(t) \ge \log_2 N$. VCQ therefore changes the data-side capacity constraint directly, without changing the AR training objective.

Empirical measurement. For each observed prefix $p = x_{<t}$, we count its occurrences $n(p)$ and the occurrences $n(p, c)$ followed by code c. We compute

$$H_{\text{emp}}(x_t \mid x_{<t}) = -\sum_{p,c} \frac{n(p,c)}{N} \log_2 \frac{n(p,c)}{n(p)}, \tag{4}$$

where terms with zero count are omitted. Figure 2(a) shows the remaining budget derived from these empirical counts. Its image curves are measured on ImageNet. The constant-codebook curve approaches the capacity floor, while VCQ delays the decrease.

Training-set evidence. On the full ImageNet training set of 1.28M images, 96.6% have unique two-token prefixes with $K = 16384$. After three tokens, the empirical next-token entropy falls below 0.002 bits. These counts directly support the entropy cliff in the training data: most prefixes have only one observed continuation (Appendix E, Table 11).

Remark (Language comparison)**.** *We also measure 3M sliding-window samples of length 256 from LAMBADA (Paperno et al., 2016), using the Llama (Touvron et al., 2023) tokenizer with a vocabulary of approximately 32K. At the first position, ρ_0 is 97.1% for the constant-codebook image tokenizer and 60.5% for language. Language consumes a smaller fraction of the available budget initially, and its measured budget decreases more gradually.*

2.2 The data-side problem and the role of regularization

Once an observed prefix uniquely identifies a sample, its empirical continuation is fixed. When this happens at most positions, repeated-prefix evidence about conditional variability is largely absent from the finite dataset. This is the data-side problem identified by the entropy cliff.

Regularization and shared parameters allow the model to learn a smoother solution across different prefixes. Consistent with this, the baseline's predictive entropy on ImageNet averages 13.5 bits over positions 3–16 and never falls below 9.1 bits. Regularization smooths the model distribution, while the empirical continuation counts describe the tokenized data.

VCQ addresses the data side by restricting early codebooks and spreading the empirical information across more positions. Under the same AR training configuration and weight decay of 0.03, VCQ-Cosine improves gFID w/o CFG from 27.98 to 14.80 and first-10-position probing accuracy from 27.1% to 43.8%. These results support changing the tokenizer's information allocation while retaining the regularization used by the baseline.

3 Variable codebook quantization

Fig. 1 illustrates the difference between Constant Codebook Size Quantization and our Variable Codebook Size Quantization. Our method is simple and intuitive: the only change is to make the codebook size vary by position rather than remain fixed. We elaborate on two design choices below.

Schedule family K_t increases monotonically from $K_{\min}$ to $K_{\max}$:

$$K_t = K_{\min} + (K_{\max} - K_{\min}) \cdot f\left(\frac{t}{L-1}\right), \qquad f(0) = 0,\ f(1) = 1 \tag{5}$$

f controls the convexity of the growth curve (Fig. 2(b)):

- **Linear**: $f(\tau) = \tau$. Uniform growth, no hyperparameters.
- **Cosine**: $f(\tau) = 1 - \cos(\pi\tau/2)$. Slow at the beginning, fast toward the end. *(default)*
- **Power**: $f(\tau) = \tau^{\alpha}$. α controls the degree of suppression at early positions.

Table 1 summarizes the concrete schedule configurations used in our VCQ experiments, including the minimum and maximum codebook sizes, the resulting average codebook size $\bar{K}$, and the nominal bits per pixel (BPP). Slower growth at early positions places a larger share of capacity at later positions, at the cost of a smaller average codebook size $\bar{K}$ and higher rFID. The uniform codebook is the special case $K_{\min} = K_{\max}$. We default to $K_{\min} = 2$, $K_{\max} = 16384$, and the Cosine schedule.

Table 1: Schedule parameters. $\bar{K}$ is the average codebook size, and BPP the nominal bits per pixel.

Schedule	$K_{\min}$	$K_{\max}$	$\bar{K}$	BPP
Constant (K=16384)	16384	16384	16384	0.055
Constant (K=8192)	8192	8192	8192	0.051
Linear	2	16384	8193	0.049
Cosine	2	16384	5964	0.044
Power (α=2.5)	2	16384	4696	0.041
Cosine-L	2	11264	4100	0.042

Shared codebook We implement VCQ with a shared codebook: a single global codebook of $K_{\max}$ entries is maintained, and position t selects the nearest neighbor from only the first K_t entries. No extra parameters are introduced and the gradient flow is unchanged. The tokenizer is trained with standard VQ losses (Esser et al., 2021) (L1+LPIPS (Zhang et al., 2018)+PatchGAN (Isola et al., 2017)) and the AR model with standard next-token prediction. VCQ changes only the size of the quantization candidate set at each position. Our GPU implementation computes distances to all $K_{\max}$ codes before masking invalid entries, retaining the full-codebook search cost.

CodebookSize-Aware CFG The non-uniform codebook of VCQ requires a corresponding adjustment to CFG: at early positions where the codebook is small and the distribution is concentrated, standard CFG over-conditions. CodebookSize-Aware CFG ties the guidance scale to the excess information capacity at position t:

$$s_t = \bar{s}_t \cdot \frac{\log_2 K_t - \log_2 K_{\min}}{\log_2 K_{\max} - \log_2 K_{\min}} \tag{6}$$

where $\bar{s}_t$ is the base CFG scale at position t, following the cosine-power schedule in Appendix C. When $K_t = K_{\min}$, $s_t = 0$ (no CFG); when $K_t = K_{\max}$, $s_t = \bar{s}_t$ (standard CFG). The output logits at position t are $(1 + s_t)\,\text{logits}_{\text{cond}} - s_t\,\text{logits}_{\text{uncond}}$. For constant codebooks, we use cosine CFG with the settings in Appendix C. Appendix D shows that lowering its initial scale from 1 to 0 brings no clear improvement.

4 Experiments

4.1 Setup

We conduct our main experiment on ImageNet 256×256 (Russakovsky et al., 2015). Our baseline tokenizer is a classical 1D tokenizer (Yu et al., 2024c) with sequence length=256. We train the tokenizer with standard L1, LPIPS and PatchGAN loss. The tokenizer is trained without class labels; class conditioning is introduced only during AR training. The AR model is then trained on the frozen tokenizer with a next-token prediction task. We use the same standard training setup across codebook configurations. All experiments use AdamW (Loshchilov & Hutter, 2019). Reconstruction quality is reported with rFID and PSNR; generation quality is reported with gFID (Heusel et al., 2017), Inception Score (Salimans et al., 2016), and the Precision/Recall metric (Kynkäänniemi et al., 2019). rFID is reported on the ImageNet validation set, and gFID follows the ADM evaluation protocol. Full configurations are listed in Appendix B; throughout the paper, configurations are named as "Tok-AR" (e.g., VCQ B-L = Base Tokenizer + Large AR).

VCQ configuration. $K_{\min} = 2$, default Cosine schedule. Base Tok: $K_{\max} = 16384$. Large Tok: $K_{\max} = 11264$.

4.2 Baseline comparison: schedule selection and ablations

Table 3 compares all codebook configurations under the same Base Tokenizer + Base AR. The training objective, optimizer, and number of steps are identical; the only difference is the codebook structure.

Table 2: **Model configurations.** Tokenizer and AR model variants used in experiments.

Tokenizer		
Base	ViT-B Enc + ViT-B Dec, $K_{\max} = 16384$	172M
Large	ViT-B Enc + ViT-L Dec, $K_{\max} = 11264$	496M
AR Model		
B		115M
L		343M
XL		684M

Table 3: **Baseline comparison**. All configurations share exactly the same training setup; the only difference is the codebook structure. gFID uses each method's best inference hyperparameters.

Schedule	$\bar{K}$	BPP	PSNR ↑	rFID ↓	gFID ↓	$\text{gFID}_{\text{noCFG}}$ ↓
Constant (K=16384)	16384	0.055	**20.67**	**2.26**	6.43	27.98
Constant (K=8192)	8192	0.051	20.48	2.36	5.87	18.16
VCQ-Linear	8193	0.049	20.35	2.42	5.32	**14.70**
VCQ-Cosine	5964	0.044	20.02	2.77	**4.79**	14.80
VCQ-Power (α=2.5)	4696	0.041	19.82	2.89	5.14	16.19

Variable structure vs. total capacity. Constant K=8192 and VCQ-Linear have similar reconstruction quality (rFID 2.36 vs. 2.42), while gFID w/o CFG improves from 18.16 to 14.70. Their total nominal capacities, $C = \sum_t \log_2 K_t$, are approximately 3328 and 3209 bits per image, respectively, a 3.6% difference. BPP is $C/(256 \times 256)$. Capacity alone also does not order the observed results: Power has lower BPP than Cosine (0.041 vs. 0.044) but worse gFID w/o CFG (16.19 vs. 14.80). Together, these results support VCQ's combination of a strong early information bottleneck and larger codebooks at later positions.

Schedule selection. From Constant (K=16384) to VCQ-Cosine, gFID w/o CFG drops from 27.98 to 14.80 (-47%) and gFID drops from 6.43 to 4.79 (-26%). VCQ-Linear matches VCQ-Cosine on gFID w/o CFG (14.70 vs. 14.80), but Cosine still wins under inference-time guidance (4.79 vs. 5.32 gFID). We therefore use Cosine as the default for the full pipeline.

Reconstruction cost vs. generation gain. VCQ-Cosine improves generation at a reconstruction cost: rFID increases from 2.26 to 2.77, while gFID w/o CFG decreases from 27.98 to 14.80. Linear retains better reconstruction quality, while Cosine gives the best guided generation in our comparison.

4.3 Comparison with existing methods

Table 4 jointly compares VCQ against existing methods on reconstruction (rFID/PSNR) and generation (gFID/IS/Prec/Rec). The comparison groups span continuous-token diffusion baselines, VAR/masked-transformer baselines, 2D discrete-token AR, and 1D discrete-token AR. On the reconstruction side, VCQ-L reaches an rFID of 0.96, competitive among 256-token methods. On the generation side, VCQ uses no auxiliary training techniques (no semantic regularization (Xiong et al., 2025; Zheng et al., 2025), no causal alignment (Wu et al., 2026), no hierarchical residuals (Zhang et al., 2026), no random-order training (Yu et al., 2024b; Pang et al., 2025)). Our controlled comparison in Table 3 shows that VCQ improves gFID w/o CFG under the same AR training configuration.

Scaling behavior. VCQ's generation quality improves as the tokenizer and AR model scale up. With the Base Tokenizer, VCQ B-B (115M) already cuts gFID w/o CFG from 27.98 to 14.80 at matched scale; scaling the AR alone, VCQ B-L drops it to 7.47, and VCQ B-XL to 6.82. Using the Large-tokenizer configuration in Appendix B, VCQ L-B reaches a gFID w/o CFG of 5.14 (already better than any Base-Tok configuration); L-L and L-XL further drive gFID w/o CFG down to 2.72 and **2.29**, on par with strong CFG-equipped 2D-token AR baselines such as LlamaGen-XXL (1.4B,

Table 4: **Reconstruction and generation comparison** (ImageNet 256^2 class-conditional). Naming format "Tok-AR" (B=Base, L=Large, XL=XLarge). rFID/PSNR are tokenizer reconstruction properties and are identical across rows that share the same tokenizer. "Diff."=diffusion, "M+D"=mask+diffusion, "Mask."=masked transformer, "HAR"=hierarchical AR, "PAR"=parallel AR. ^{S}Semantic/VFM: semantic regularization, semantic alignment, or frozen/pre-trained vision foundation model; ^{C}Clustering: periodic token/codebook clustering; ^{D}Diff./FM: diffusion or flow-matching generator, decoder, token-distribution model, or loss; ^{T}Dropout: tail dropout.

Method	Gen & Param	rFID ↓	PSNR ↑	gFID ↓	gFID w/o CFG ↓	IS ↑	Prec ↑	Rec ↑
Continuous tokens (diffusion / MAR)								
LDM-4-G (Rombach et al., 2022)^{D}	Diff. 400M	0.27	—	3.60	10.56	247.7	0.87	0.48
DiT-XL/2 (Peebles & Xie, 2023)^{D}	Diff. 675M	0.62	—	2.27	9.62	278.2	0.83	0.57
SiT-XL (Ma et al., 2024)^{D}	Diff. 675M	—	—	2.06	8.61	270.3	0.82	0.59
REPA (SiT-XL) (Yu et al., 2025)^{SD}	Diff. 675M	—	—	1.42	5.90	305.7	0.80	0.65
LightningDiT (Yao et al., 2025)^{SD}	Diff. 675M	0.28	—	1.35	2.17	295.3	0.79	0.65
MAR-B (Li et al., 2024)^{D}	M+D 208M	0.87	—	2.31	3.48	281.7	0.82	0.57
MAR-H (Li et al., 2024)^{D}	M+D 943M	0.87	—	1.55	2.35	303.7	0.81	0.62
FlowAR-B (Ren et al., 2024)^{D}	Flow+VAR 300M	0.87	—	2.90	—	272.5	0.84	0.54
Discrete tokens, next-scale and masked generation								
VAR-d30 (Tian et al., 2024)	VAR 2.0B	—	—	1.92	—	323.1	0.82	0.59
MaskGIT (Chang et al., 2022)	Mask. 227M	2.28	—	—	6.18	182.1	0.80	0.51
MAGVIT-v2 (Yu et al., 2024a)	Mask. 307M	—	—	1.78	3.65	319.4	—	—
2D discrete tokens, AR								
VQGAN (Esser et al., 2021)	AR 1.4B	4.98	—	—	15.78	74.3	—	—
RQ-Trans. (Lee et al., 2022)	AR 3.8B	3.20	—	—	7.55	134.0	—	—
LlamaGen-XL (Sun et al., 2024)	AR 775M	0.94	—	2.62	—	244.1	0.80	0.57
LlamaGen-XXL (Sun et al., 2024)	AR 1.4B	0.94	—	2.34	—	253.9	0.80	0.59
PAR-L-4× (Wang et al., 2025b)	PAR 343M	0.94	—	3.76	—	218.9	0.84	0.50
1D discrete tokens, non-standard AR (mask / random-order / hierarchical / parallel)								
TiTok-S-128 (Yu et al., 2024c)	Mask. 287M	1.71	—	1.97	4.44	281.8	—	—
TiTok-L-32 (Yu et al., 2024c)	Mask. 177M	2.21	—	2.77	3.15	199.8	—	—
ImageFolder (Li et al., 2025)ST	VAR 362M	0.80	—	2.60	—	295.0	0.75	0.63
ResTok (Zhang et al., 2026)ST	HAR 326M	1.28	—	2.34	—	257.8	0.79	0.60
SpectralAR (Huang et al., 2025)	AR 310M	4.03	—	3.02	—	282.2	0.81	0.55
DetailFlow-16 (Liu et al., 2025)	PAR 326M	1.22	19.05	2.96	—	221.4	0.82	0.57
RAR-XL (Yu et al., 2024b)	RandAR 955M	—	—	1.50	3.72	306.9	0.80	0.62
RAR-XXL (Yu et al., 2024b)	RandAR 1.5B	—	—	1.48	3.26	326.0	0.80	0.63
1D discrete tokens, standard AR								
GigaTok-B-L (Xiong et al., 2025)^{S}	AR 111M	0.81	21.21	3.26	—	221.0	0.81	0.56
VFMTok-L (Zheng et al., 2025)^{S}	AR 343M	0.89	—	2.75	2.11	278.8	0.84	0.57
AliTok-B (Wu et al., 2026)^{C}	AR 177M	0.86	—	1.44	2.40	319.5	0.77	0.65
AliTok-L (Wu et al., 2026)^{C}	AR 318M	0.86	—	1.38	1.98	326.2	0.78	0.65
AliTok-XL (Wu et al., 2026)^{C}	AR 662M	0.86	—	1.28	1.88	306.3	0.79	0.65
VCQ B-B (ours)	AR 115M	2.77	20.02	4.79	14.80	221.9	0.828	0.452
VCQ B-L (ours)	AR 343M	2.77	20.02	2.75	7.47	258.3	0.801	0.572
VCQ B-XL (ours)	AR 684M	2.77	20.02	2.38	6.82	306.9	0.802	0.587
VCQ L-B (ours)	AR 115M	0.96	19.71	2.73	5.14	218.7	0.777	0.590
VCQ L-L (ours)	AR 343M	0.96	19.71	1.90	2.72	243.6	0.752	0.658
VCQ L-XL (ours)	AR 684M	0.96	19.71	1.71	2.29	250.8	0.762	0.670

gFID 2.34). With CodebookSize-Aware CFG, VCQ L-L and VCQ L-XL reach gFID 1.90 and **1.71** with 343M and 684M AR parameters, respectively, and no auxiliary training techniques.

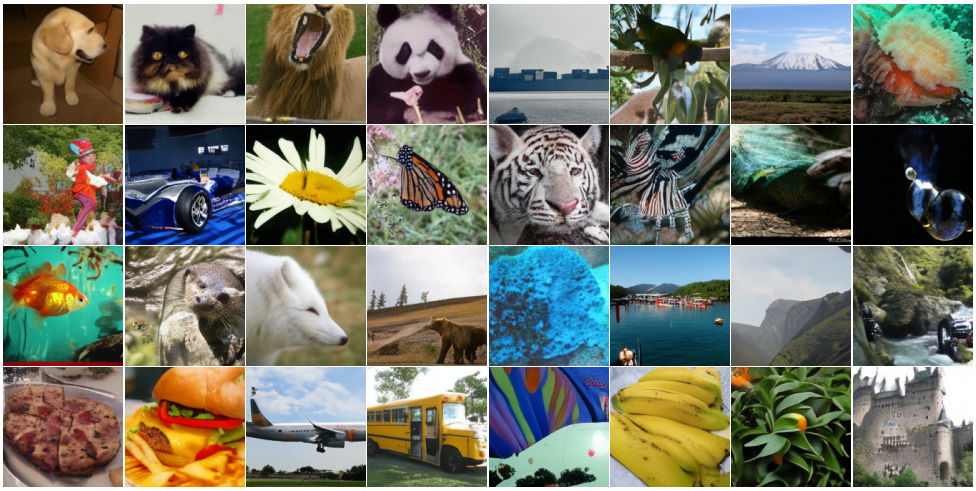

Figure 3: **Uncurated class-conditional samples from VCQ L-XL.** A 4×8 grid covering visually diverse ImageNet classes with no manual curation. Samples use the Large Tokenizer ($K_{\max}=11264$) and AR-XL (684M) with CodebookSize-Aware CFG at the best gFID configuration: cosine CFG with $s = 3$ and $p = 1.0$.

Table 5: **Semantic aggregation.** Linear probing uses mean-pooled pre-quantization features from the first k positions.

Tokenizer	k	Top-1 ↑	Top-5 ↑
Constant (K=16384)	10	27.12%	49.27%
Constant (K=16384)	256	48.02%	71.51%
VCQ-Linear	10	31.24%	54.59%
VCQ-Linear	256	46.60%	70.11%
VCQ-Cosine	10	43.77%	67.70%
VCQ-Cosine	256	**48.79%**	**72.70%**
VCQ-Power (α=2.5)	10	**49.42%**	**72.70%**
VCQ-Power (α=2.5)	256	46.93%	70.71%

4.4 Semantic aggregation and visualization

We average the pre-quantization features of the first k tokens and train a linear classifier on the resulting feature, following the linear probing protocol of MAE (He et al., 2022).

$K_{\min}=2$ limits the first discrete code to a capacity of only 1 bit. The probing results suggest that this bottleneck encourages semantic information to concentrate in early encoder features. Using features from only 10 positions (3.9% of the sequence), VCQ-Cosine achieves a linear probing accuracy of 43.77%, while Constant reaches only 27.12%. With all 256 positions, the linear-probe accuracies are comparable (48.79% vs. 48.02%), while VCQ performs better with only the first 10 positions.

The Cosine schedule grows more slowly at the start than Linear, imposing a stronger early information bottleneck; the first-10-position accuracy differs by 12.5 percentage points. This semantic concentration emerges without explicit semantic supervision.

5 Conclusion

We identify the Entropy Cliff: large uniform codebooks can consume the finite-data entropy budget within the first few positions, leaving most empirical continuations fixed. VCQ delays this decline by using small codebooks at early positions and progressively larger ones later. This simple structural

change redistributes capacity along the sequence without changing the loss function, parameter count, or AR training procedure.

On ImageNet, VCQ reduces gFID w/o CFG from 27.98 to 14.80 under the same AR training configuration. Combined with CodebookSize-Aware CFG, it improves gFID from 6.43 to 4.79. Scaling to the Large tokenizer and a 684M-parameter AR model further brings gFID to 1.71, without semantic regularization or causal alignment. The strong information bottleneck at early positions also encourages semantic concentration in encoder features: linear probing on the first 10 positions reaches 43.8% accuracy, compared with 27.1% for the uniform codebook.

Our experiments focus on ImageNet 256×256 class-conditional generation, and the empirical cliff depends on the tokenized data and codebook schedule. Evaluating VCQ on more datasets and combining it with existing training techniques are natural next steps. These results show that codebook design should consider both its total capacity and how that capacity is distributed across the sequence.

AI USE STATEMENT

LLMs were used for language polishing, coding assistance, reviewing mathematical formulas, and searching for related literature. The core method and research were developed without AI assistance. The authors take full responsibility for the final content of this paper.

REFERENCES

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, 2025. URL https://arxiv.org/abs/2502.13967.

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 (CVPR)*, pp. 11315–11325, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Chang_MaskGIT_Masked_Generative_Image_Transformer_CVPR_2022_paper.html.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat gans on image synthesis. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 8780–8794, 2021. URL https://proceedings.neurips.cc/paper/2021/hash/49ad23d1ec9fa4bd8d77d02681df5cfa-Abstract.html.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021*, pp. 12873–12883. Computer Vision Foundation / IEEE, 2021. doi: 10.1109/CVPR46437.2021.01268. URL https://openaccess.thecvf.com/content/CVPR2021/html/Esser_Taming_Transformers_for_High-Resolution_Image_Synthesis_CVPR_2021_paper.html.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/He_Masked_Autoencoders_Are_Scalable_Vision_Learners_CVPR_2022_paper.html.

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 Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/4c5bcfec8584af0d967f1ab10179ca4b-Abstract.html.

Yuanhui Huang, Weiliang Chen, Wenzhao Zheng, Yueqi Duan, Jie Zhou, and Jiwen Lu. Spectralar: Spectral autoregressive visual generation, 2025. URL https://arxiv.org/abs/2506.10962.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. URL https://openaccess.thecvf.com/content_cvpr_2017/html/Isola_Image-To-Image_Translation_With_CVPR_2017_paper.html.

Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/0234c510bc6d908b28c70ff313743079-Abstract.html.

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 (CVPR)*, pp. 11523–11532, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Lee_Autoregressive_Image_Generation_Using_Residual_Quantization_CVPR_2022_paper.html.

Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 56424–56445. Curran Associates, Inc., 2024. doi: 10.52202/079017-1797. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/66e226469f20625aaebddbe47f0ca997-Paper-Conference.pdf.

Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. Imagefolder: Autoregressive image generation with folded tokens. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=QE1LFzXQPL.

Yiheng Liu, Liao Qu, Huichao Zhang, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Xian Li, Shuai Wang, Daniel K. Du, Fangmin Chen, Zehuan Yuan, and Xinglong Wu. Detailflow: 1d coarse-to-fine autoregressive image generation via next-detail prediction, 2025. URL https://arxiv.org/abs/2505.21473.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

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, 2024. URL https://arxiv.org/abs/2409.04410.

Nanye Ma, Mark Goldstein, Michael S. Albergo, Nicholas M. Boffi, Eric Vanden-Eijnden, and Saining Xie. Sit: Exploring flow and diffusion-based generative models with scalable interpolant transformers. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol (eds.), *Computer Vision – ECCV 2024*, pp. 23–40, Cham, 2024. Springer Nature Switzerland. ISBN 978-3-031-72980-5. URL https://arxiv.org/abs/2401.08740.

Keita Miwa, Kento Sasaki, Hidehisa Arai, Tsubasa Takahashi, and Yu Yamaguchi. One-d-piece: Image tokenizer meets quality-controllable compression, 2025. URL https://arxiv.org/abs/2501.10064.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=a68SUt6zFt. Featured Certification.

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 (CVPR)*, pp. 45–55, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Pang_RandAR_Decoder-only_Autoregressive_Visual_Generation_in_Random_Orders_CVPR_2025_paper.html.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The LAMBADA dataset: Word prediction requiring a broad discourse context. In Katrin Erk and Noah A. Smith (eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, Berlin, Germany, aug 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1144. URL https://aclanthology.org/P16-1144/.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pp. 4172–4182. IEEE, 2023. doi: 10.1109/ICCV51070.2023.00387. URL https://doi.org/10.1109/ICCV51070.2023.00387.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pp. 8748–8763. PMLR, 2021. URL http://proceedings.mlr.press/v139/radford21a.html.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8821–8831. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/ramesh21a.html.

Sucheng Ren, Qihang Yu, Ju He, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Flowar: Scale-wise autoregressive image generation meets flow matching, 2024. URL https://arxiv.org/abs/2412.15205.

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 (CVPR)*, pp. 10684–10695, June 2022. URL https://openaccess.thecvf.com/content/CVPR2022/html/Rombach_

High-Resolution_Image_Synthesis_With_Latent_Diffusion_Models_CVPR_2022_paper.html.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.*, 115(3):211–252, 2015. doi: 10.1007/S11263-015-0816-Y. URL https://doi.org/10.1007/s11263-015-0816-y.

Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, 2016. URL https://proceedings.neurips.cc/paper/2016/hash/8a3363abe792db2d8761d6403605aeb7-Abstract.html.

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 (ICCV)*, pp. 19471–19481, October 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Sargent_Flow_to_the_Mode_Mode-Seeking_Diffusion_Autoencoders_for_State-of-the-Art_Image_ICCV_2025_paper.html.

Fengyuan Shi, Zhuoyan Luo, Yixiao Ge, Yujiu Yang, Ying Shan, and Limin Wang. Scalable image tokenization with index backpropagation quantization, 2025. URL https://arxiv.org/abs/2412.02692.

Peize Sun, Yi Jiang, Shoufa Chen, Shilong Zhang, Bingyue Peng, Ping Luo, and Zehuan Yuan. Autoregressive model beats diffusion: Llama for scalable image generation, 2024. URL https://arxiv.org/abs/2406.06525.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 84839–84865. Curran Associates, Inc., 2024. doi: 10.52202/079017-2694. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/9a24e284b187f662681440ba15c416fb-Paper-Conference.pdf.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

Aäron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 6306–6315, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/7a98af17e63a0ac09ce2e96d03992fbc-Abstract.html.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 2017. URL https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

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. Selftok: Discrete visual tokens of autoregression, by diffusion, and for reasoning, 2025a. URL https://arxiv.org/abs/2505.07538.

Yuqing Wang, Shuhuai Ren, Zhijie Lin, Yujin Han, Haoyuan Guo, Zhenheng Yang, Difan Zou, Jiashi Feng, and Xihui Liu. Parallelized autoregressive visual generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12955–12965, June 2025b. URL https://openaccess.thecvf.com/content/CVPR2025/html/Wang_Parallelized_Autoregressive_Visual_Generation_CVPR_2025_paper.html.

Mark Weber, Lijun Yu, Qihang Yu, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. Maskbit: Embedding-free image generation via bit tokens. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=NYe2JuN3v3. Featured Certification, Reproducibility Certification.

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 *The Fourteenth 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, 2025. URL https://arxiv.org/abs/2504.08736.

Jingfeng Yao, Bin Yang, and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15703–15712, June 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Yao_Reconstruction_vs._Generation_Taming_Optimization_Dilemma_in_Latent_Diffusion_Models_CVPR_2025_paper.html.

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*, 2022a. URL https://openreview.net/forum?id=pfNyExj7z2.

Jiahui Yu, Yuanzhong Xu, Jing Yu Koh, Thang Luong, Gunjan Baid, Zirui Wang, Vijay Vasudevan, Alexander Ku, Yinfei Yang, Burcu Karagol Ayan, Ben Hutchinson, Wei Han, Zarana Parekh, Xin Li, Han Zhang, Jason Baldridge, and Yonghui Wu. Scaling autoregressive models for content-rich text-to-image generation. *Transactions on Machine Learning Research*, 2022b. ISSN 2835-8856. URL https://openreview.net/forum?id=AFDcYJKhND. Featured Certification.

Lijun Yu, Jose 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 *The Twelfth International Conference on Learning Representations*, 2024a. URL https://openreview.net/forum?id=gzqrANCF4g.

Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation, 2024b. URL https://arxiv.org/abs/2411.00776.

Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024c. URL https://openreview.net/forum?id=tOXoQPRzPL.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=DJSZGGZYVi.

Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pp. 586–595. Computer Vision Foundation / IEEE Computer Society, 2018. doi: 10.1109/CVPR.

2018.00068. URL http://openaccess.thecvf.com/content_cvpr_2018/html/Zhang_The_Unreasonable_Effectiveness_CVPR_2018_paper.html.

Xu Zhang, Cheng Da, Huan Yang, Kun Gai, Ming Lu, and Zhan Ma. Restok: Learning hierarchical residuals in 1d visual tokenizers for autoregressive image generation, 2026. URL https://arxiv.org/abs/2601.03955.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=PESrAH82Zh.

A Related work

Discrete visual tokenizers. VQ-VAE (van den Oord et al., 2017) and VQ-GAN (Esser et al., 2021) established the basic paradigm of representing images as sequences of discrete codes from a finite codebook. Subsequent works have improved from various angles: better optimization and larger codebooks (ViT-VQGAN (Yu et al., 2022a); LlamaGen (Sun et al., 2024) with $K = 16384$; Open-MAGVIT2 (Luo et al., 2024) and MAGVIT-v2 (Yu et al., 2024a) with LFQ; IBQ (Shi et al., 2025) with index back-propagation); 1D / variable-length tokenizers (TiTok (Yu et al., 2024c); FlexTok (Bachmann et al., 2025); One-D-Piece (Miwa et al., 2025); SelfTok (Wang et al., 2025a)); training techniques (GigaTok (Xiong et al., 2025) with semantic regularization, AliTok (Wu et al., 2026) with causal alignment, ResTok (Zhang et al., 2026) with hierarchical residuals, ImageFolder (Li et al., 2025) with folded tokens, MaskBit (Weber et al., 2024) with bit-level codes, FlowMo (Sargent et al., 2025) with mode-seeking diffusion autoencoders); and the use of pre-trained vision foundation models such as DINOv2 (Oquab et al., 2024) and CLIP (Radford et al., 2021) as encoders or alignment targets (LightningDiT/VA-VAE (Yao et al., 2025), VFMTok (Zheng et al., 2025)). To the best of our knowledge, VCQ is the first work to systematically explore position-dependent codebook cardinality K_t as a primary design variable.

AR visual generation. Early AR image models such as DALL-E (Ramesh et al., 2021) and Parti (Yu et al., 2022b) treat tokenized images as sequences and generate them with large Transformers (Vaswani et al., 2017); VQ-GAN (Esser et al., 2021) brought GPT-style class-conditional AR to ImageNet. Subsequent developments include scaling (LlamaGen (Sun et al., 2024)), residual quantization (RQ-Transformer (Lee et al., 2022)), next-scale prediction (VAR (Tian et al., 2024)), masked / non-causal generation (MaskGIT (Chang et al., 2022)), diffusion-loss heads on continuous tokens (MAR (Li et al., 2024), FlowAR (Ren et al., 2024)), parallel decoding (PAR (Wang et al., 2025b)), random or randomized generation orders (RandAR (Pang et al., 2025), RAR (Yu et al., 2024b)), and coarse-to-fine 1D ordering (DetailFlow (Liu et al., 2025), SpectralAR (Huang et al., 2025)). On the diffusion side, ADM (Dhariwal & Nichol, 2021) and DDPM (Ho et al., 2020) together with classifier-free guidance (Ho & Salimans, 2022) laid the foundation for latent diffusion (LDM (Rombach et al., 2022)) and Diffusion/Interpolant Transformers (DiT (Peebles & Xie, 2023), SiT (Ma et al., 2024), REPA (Yu et al., 2025), LightningDiT (Yao et al., 2025)). We study codebook structure with 1D token sequences and standard next-token prediction.

Reconstruction-generation trade-off. GigaTok (Xiong et al., 2025) attributes the harm of increasing K for generation to increased latent space complexity, and proposes semantic regularization as a remedy. We provide a complementary information-theoretic explanation: uniform large codebooks can leave almost no empirical conditional entropy at later positions, while regularization allows the AR model to learn a smoother distribution. VCQ changes this data-side allocation of information.

Evaluation protocol. Throughout the paper we follow standard protocols: FID (Heusel et al., 2017) and Inception Score (Salimans et al., 2016) as fidelity/quality summaries, and the improved Precision/Recall (Kynkäänniemi et al., 2019) as a separate measure of fidelity vs. coverage. Reconstruction quality is reported with rFID, PSNR. We use the standard ADM (Dhariwal & Nichol, 2021) evaluation implementation for FID, IS, and Precision/Recall. Architectural and optimization choices follow widely-used recipes (ViT (Dosovitskiy et al., 2021), Transformer (Vaswani et al., 2017), AdamW (Loshchilov & Hutter, 2019), PatchGAN-style adversarial loss (Isola et al., 2017), Llama-style decoder-only AR (Touvron et al., 2023)).

B Training details

Dataset and compute accounting. All tokenizer and autoregressive (AR) experiments are trained on the ImageNet-1K training set. Experiments were run on a heterogeneous NVIDIA GPU cluster. For comparability, we report training duration as A100-equivalent wall-clock time.

Tokenizer training. Both tokenizers use a 1D sequence length of 256 and a shared cosine VCQ codebook with $K_{\min} = 2$. The Base tokenizer uses $K_{\max} = 16384$, while the Large tokenizer uses $K_{\max} = 11264$. The Base tokenizer follows a ViT-B encoder and ViT-B decoder design with 172M parameters in total. The Large tokenizer keeps the same ViT-B encoder but uses a larger

ViT-L decoder, resulting in 496M parameters. For the Large tokenizer, LPIPS is computed using a ConvNeXt-S backbone following (Wu et al., 2026).

Both tokenizers are trained for 130 ImageNet epochs with a global batch size of 256. The 1.3M optimization-step count includes alternating generator and discriminator updates, corresponding to approximately 650k tokenizer (generator) updates. The Base tokenizer is trained on 8 GPUs and takes 13.0 A100-equivalent hours. The Large tokenizer is trained on 8 GPUs and takes 18.2 A100-equivalent hours.

Tokenizer optimization uses AdamW. After a 13k-step warmup, the learning rate stays at 10^{-4} until step 65k, then decays linearly to 3×10^{-5}. The reconstruction objective combines L1 reconstruction, LPIPS perceptual loss, adversarial generator loss, adversarial discriminator loss, and the VQ commitment loss. The L1, LPIPS, discriminator, and commitment losses use weight 1.0, while the adversarial generator loss uses weight 0.1. We use random cropping and horizontal flipping for image augmentation, and adversarial augmentation is always enabled. Training uses bf16 mixed precision.

Training budgets across methods. Table 6 summarizes the training epochs for VCQ, GigaTok (Xiong et al., 2025), and AliTok (Wu et al., 2026). VCQ uses 130 tokenizer epochs, compared with 200 for GigaTok and 180 for AliTok. The AR budgets depend on the configuration; our Base-tokenizer and Large-tokenizer groups use 400 and 800 epochs, respectively. Model sizes, batch sizes, and training recipes vary across methods.

Table 6: Training budgets in epochs. VCQ tokenizer epochs count generator updates. AliTok AR budgets are listed for B/L/XL; VCQ AR budgets apply to all three AR sizes within each tokenizer group.

Method	Tokenizer epochs	AR epochs
GigaTok	200	300
AliTok	180	800 / 800 / 400
VCQ Base tokenizer	130	400
VCQ Large tokenizer	130	800

Autoregressive training. AR models are trained on frozen tokenizers with standard next-token prediction. All AR runs use ten-crop augmentation, horizontal flipping, class-label dropout of 0.1, AR dropout of 0.1, and token-embedding dropout of 0.1.

For the Base-tokenizer group, BB, BL, and BXL are trained for 1.0M optimization steps with a global batch size of 512, corresponding to 400 ImageNet epochs. BB uses the 115M AR-B model with 12 layers, 12 heads, and hidden dimension 768. It is trained on 8 GPUs and takes 7.6 A100-equivalent hours. BL uses the 343M AR-L model with 24 layers, 16 heads, and hidden dimension 1024. It is trained on 8 GPUs and takes 15.7 A100-equivalent hours. BXL uses the 684M AR-XL model with 32 layers, 20 heads, and hidden dimension 1280. It is trained on 16 GPUs and takes 20.0 A100-equivalent hours.

For this Base-tokenizer group, AR optimization uses AdamW with $\beta = (0.9, 0.96)$. After a 10k-step warmup, the learning rate stays at 2×10^{-4} until step 50k, then decays linearly to 4×10^{-5}. The AR weight decay is 3×10^{-2}, while bias, normalization, and adaptive normalization parameters are excluded from weight decay. Gradient clipping is set to 1.0.

For the Large-tokenizer group, LB, LL, and LXL are trained for 500k optimization steps with a global batch size of 2048, corresponding to 800 ImageNet epochs. LB uses the 115M AR-B model and is trained on 16 GPUs for 5.3 A100-equivalent hours. LL uses the 343M AR-L model and is trained on 16 GPUs for 12.4 A100-equivalent hours. LXL uses the 684M AR-XL model and is trained on 32 GPUs for 15.1 A100-equivalent hours.

For this Large-tokenizer group, AR optimization also uses AdamW with $\beta = (0.9, 0.96)$ and the same weight-decay rule as above. The learning rate is linearly decayed from 4×10^{-4} to 10^{-5}, with a 62.5k-step warmup. Gradient clipping is set to 1.0.

Table 7: CFG and temperature configurations corresponding to the reported gFID and gFID w/o CFG values.

Model	Comment	CFG	Temperature
Constant (K=16384)	gFID=6.43	cosine, scale=20, p=1.5	1.0
Constant (K=16384)	gFID w/o CFG=27.98	cfg=0	0.85
Constant (K=8192)	gFID=5.87	cosine, scale=20, p=2.0	1.0
Constant (K=8192)	gFID w/o CFG=18.16	cfg=0	0.85
VCQ-Linear	gFID=5.32	cosine, scale=14, p=1.75	1.0
VCQ-Linear	gFID w/o CFG=14.70	cfg=0	0.85
VCQ-Cosine / VCQ B-B	gFID=4.79	cosine, scale=10, p=1.5	1.0
VCQ-Cosine / VCQ B-B	gFID w/o CFG=14.80	cfg=0	0.85
VCQ-Power (α=2.5)	gFID=5.14	cosine, scale=10, p=1.75	1.0
VCQ-Power (α=2.5)	gFID w/o CFG=16.19	cfg=0	0.85
VCQ B-L	gFID=2.75	cosine, scale=4, p=0.75	1.0
VCQ B-L	gFID w/o CFG=7.47	cfg=0	0.85
VCQ B-XL	gFID=2.38	cosine, scale=5, p=0.75	1.0
VCQ B-XL	gFID w/o CFG=6.82	cfg=0	0.85
VCQ L-B	gFID=2.73	cosine, scale=4, p=1.25	1.0
VCQ L-B	gFID w/o CFG=5.14	cfg=0	0.90
VCQ L-L	gFID=1.90	cosine, scale=4, p=1.25	1.0
VCQ L-L	gFID w/o CFG=2.72	cfg=0	0.90
VCQ L-XL	gFID=1.71	cosine, scale=3, p=1.0	1.0
VCQ L-XL	gFID w/o CFG=2.29	cfg=0	0.95

C Inference details

Table 7 lists the CFG and temperature settings used for the main gFID and gFID w/o CFG results.

Cosine-power CFG. We use the power-cosine profile to schedule the base guidance scale:

$$c_p(\tau) = \frac{1 - \cos(\pi \tau^p)}{2}, \qquad \bar{s}_t = a + (s - a) c_p(\tau_t), \tag{7}$$

where $\tau_t \in [0, 1]$ is normalized generation progress. The parameters s and $p > 0$ correspond to scale and p in Table 7; larger p delays the rise in guidance. Constant codebooks use $s_t = \bar{s}_t$, while VCQ applies the codebook weight in Eq. (6). All scales follow the logits convention $(1 + s_t)\,\text{logits}_{\text{cond}} - s_t\,\text{logits}_{\text{uncond}}$. Runs without CFG set $s_t = 0$ throughout.

D Disentangling CodebookSize-Aware CFG and VCQ contributions

The gFID improvement of VCQ involves two factors: the codebook structure (VCQ) and the inference strategy (CodebookSize-Aware CFG from Eq. (6)). This section disentangles them through two controlled experiments.

Experiment 1: Vanilla CFG vs. CodebookSize-Aware CFG. Table 8 switches inference strategies on the same VCQ tokenizer and AR model. With other sampling settings fixed, gFID w/o CFG is unaffected by the choice of CFG schedule and reflects the AR model's modeling quality.

VCQ brings gFID w/o CFG from 27.98 to 14.80 (-47%), which is the contribution of the codebook structure. CodebookSize-Aware CFG further reduces gFID from 7.61 to 4.79 (-2.82), which is the contribution of the inference strategy.

The gFID of VCQ plus Vanilla Cosine CFG (7.61) is worse than Constant (6.43) because Vanilla Cosine CFG does not match VCQ's variable structure: at early positions where the codebook

Table 8: **Disentangling VCQ and CodebookSize-Aware CFG contributions.** The VCQ rows use the same tokenizer and AR model with different inference strategies.

Tokenizer	Inference strategy	gFID ↓	$\text{gFID}_{\text{noCFG}}$ ↓	ΔgFID
Constant (K=16384)	Cosine CFG	6.43	27.98	—
Constant (K=8192)	Cosine CFG ($a = 0$)	5.87	18.16	—
VCQ	Cosine CFG	7.61	14.80	—
VCQ	CodebookSize-Aware CFG	**4.79**	14.80	−2.82

is small and the distribution is concentrated, the standard cosine CFG schedule over-conditions. CodebookSize-Aware CFG ties the scale to excess information capacity (Eq. (6)); when $K_t = K_{\min}$, $s_t = 0$, resolving this mismatch.

Experiment 2: Lower initial CFG on uniform codebooks. With a constant codebook, codebook size provides no position-dependent signal for guidance. We therefore compare standard cosine CFG, which starts at 1, with a schedule starting at 0. This allows CFG scales below 1 at early positions, as in CodebookSize-Aware CFG.

Table 9: **Lower initial CFG on uniform codebooks.**

Tokenizer	CFG strategy	Best gFID ↓	Δ
Constant K=16K	Standard cosine CFG	6.43	—
Constant K=16K	CodebookSize-Aware CFG	6.43	0.00
Constant K=8K	Standard cosine CFG	5.92	—
Constant K=8K	CodebookSize-Aware CFG	5.87	−0.05

Both CFG strategies give similar results on uniform codebooks, with no clear benefit from the lower initial scale. VCQ improves unguided generation, while CodebookSize-Aware CFG improves guided sampling with variable codebooks.

E Finite-data entropy analysis

Empirical distribution. Let S be a uniformly sampled index among N observations. Each observation is deterministically encoded as $X = (X_0, \ldots, X_{L-1})$, where X_i takes at most K_i values. All entropies in this section refer to this empirical distribution and use base-2 logarithms. Duplicate complete token sequences are allowed.

Remaining budget and proof of Proposition 1. Since the prefix $X_{<t}$ is a deterministic function of S, the chain rule gives

$$R(t) = H(S \mid X_{<t}) = \log_2 N - H(X_{<t}) \tag{8}$$

$$= \log_2 N - \sum_{i<t} H(X_i \mid X_{<i}). \tag{9}$$

Each conditional entropy satisfies $0 \le H(X_i \mid X_{<i}) \le \log_2 K_i$, while $H(X_{<t}) \le H(S) = \log_2 N$. Hence

$$\max\left(0, \log_2 N - \sum_{i<t} \log_2 K_i\right) \le R(t) \le \log_2 N. \tag{10}$$

The remaining token sequence is also determined by S, so

$$H(X_{t:L} \mid X_{<t}) = H(X) - H(X_{<t}) \tag{11}$$

$$= R(t) - H(S \mid X) \le R(t), \tag{12}$$

where $X_{t:L}$ denotes $(X_t, \ldots, X_{L-1})$. Thus $R(t)$ equals the remaining token entropy when all complete token sequences are distinct; otherwise it is an upper bound. Also, $R(t) = 0$ if and only if

all N length-t prefixes are distinct. In that case the entire tail is deterministic given the prefix. More generally, $R(t) \le \epsilon$ leaves at most ϵ bits for the whole tail.

Conditional utilization. Define

$$A_i = \min\big(\log_2 K_i, R(i)\big), \qquad \rho_i = \begin{cases} H(X_i \mid X_{<i})/A_i, & A_i > 0, \\ 0, & A_i = 0. \end{cases} \tag{13}$$

The tail bound above implies $0 \le H(X_i \mid X_{<i}) \le A_i$, so $\rho_i \in [0, 1]$ and

$$R(i+1) = R(i) - \rho_i A_i. \tag{14}$$

If $\rho_i = 1$ at each step with positive available budget, then

$$R(t) = \max\left(0, \log_2 N - \sum_{i<t} \log_2 K_i\right). \tag{15}$$

This expression gives the full-utilization floor.

Constant codebooks and the capacity floor. For $K_i = K$, the floor reaches zero at $t^*_{\text{floor}} = \lceil \log_2 N / \log_2 K \rceil$. Equivalently, $K^t \ge N$ is necessary for the prefix space to distinguish all N observations. It is not sufficient for the actual tokenizer to assign distinct prefixes. Under full conditional utilization the per-position entropy is

$$H(X_t \mid X_{<t}) = \min\left(\log_2 K, \max\big(0, \log_2 N - t \log_2 K\big)\right). \tag{16}$$

Without that condition, earlier positions may consume less entropy and leave more for later positions. For $N = K^2$ sequences of the form (a, a, b), with a and b independently ranging over all K codes, every position uses every code. Yet $H(X_1 \mid X_0) = 0$ and $H(X_2 \mid X_0, X_1) = \log_2 K$. Marginal usage is therefore insufficient to determine the cliff.

Dataset size. Table 10 reports capacity floors computed from dataset size with $K = 16384$. Multiplying N by K increases this computed floor by one position.

Table 10: Capacity floors for constant $K = 16384$. Actual empirical curves depend on the tokenized samples and conditional utilization.

Dataset	N	$\log_2 N$	t^*_{floor}
CIFAR-10/100	50K	15.6	2
COCO	118K	16.9	2
ImageNet-1K	1.28M	20.3	2
CC12M	12M	23.5	2
LAION-400M	400M	28.6	3
LAION-5B	5B	32.2	3

Empirical estimation and model regularization. Equation (4) measures conditional entropy from observed prefix counts. For Figure 2(a), image prefixes are counted on the ImageNet validation set. A unique prefix has a single empirical continuation and contributes zero conditional entropy. The remaining-budget curve follows from these empirical entropies, whereas the capacity floor uses N and the codebook schedule.

A regularized AR model can share information across prefixes and retain high predictive entropy. In our baseline, validation predictive entropy averages 13.5 bits over positions 3–16, and its minimum over positions is 9.1 bits. VCQ changes the empirical prefix structure, while regularization shapes the model distribution.

Training-set prefix counts. We also compute prefix counts on all 1,281,167 ImageNet training images encoded with the constant codebook of $K = 16384$. Table 11 reports the number of distinct length-t prefixes and the number of images whose prefix is shared with another image.

By prefix length 2, 96.6% of images have unique prefixes; by length 3, this fraction reaches 98.9%. The empirical next-token entropy falls below 0.002 bits at length 3. Thus most training prefixes have only one observed continuation, consistent with the data-side entropy cliff.

Table 11: Prefix statistics for the constant codebook on the ImageNet training set. t is the prefix length; mean count is the number of images per distinct prefix. The last column is the empirical entropy of the next token, in bits.

t	Distinct prefixes	Images with shared prefixes	Mean count	$H_{\mathrm{emp}}(x_t \mid x_{<t})$
1	16,384	1,281,167	78.20	6.330
2	1,251,312	42,963	1.024	0.0995
3	1,274,113	13,495	1.006	0.00175
4	1,274,808	12,463	1.005	0.00045

F VARIABLE CODEBOOKS WITH A SPATIAL 2D TOKENIZER

We also evaluate reconstruction with a spatial 2D tokenizer under constant and Cosine codebook schedules. Both configurations use the same training budget within this 2D setting.

Table 12: Reconstruction with a spatial 2D tokenizer.

Codebook schedule	rFID ↓
Constant	2.16
VCQ-Cosine	3.92

The rFID increase is 1.76, compared with 0.51 for the main 1D comparison (2.26 to 2.77). In a spatial tokenizer, restricting early positions reduces the capacity available to particular spatial regions. The larger reconstruction cost suggests that the same schedule may be less suitable for this setting. Each schedule is compared with the constant-codebook baseline for the same tokenizer architecture.

G ADDITIONAL GENERATION SAMPLES

We provide the full uncurated sample gallery for **VCQ L-XL** (Large Tokenizer with $K_{\max} = 11264$, AR-XL with 684M parameters). Sampling uses CodebookSize-Aware CFG at the configuration that minimizes gFID under ADM protocol: cosine CFG with $s = 3$ and $p = 1.0$. Each page is an 8×8 grid of 64 randomly selected ImageNet classes, one uncurated sample per class. Across the eight appendix pages, the class sets are disjoint, yielding 512 sampled classes in total.

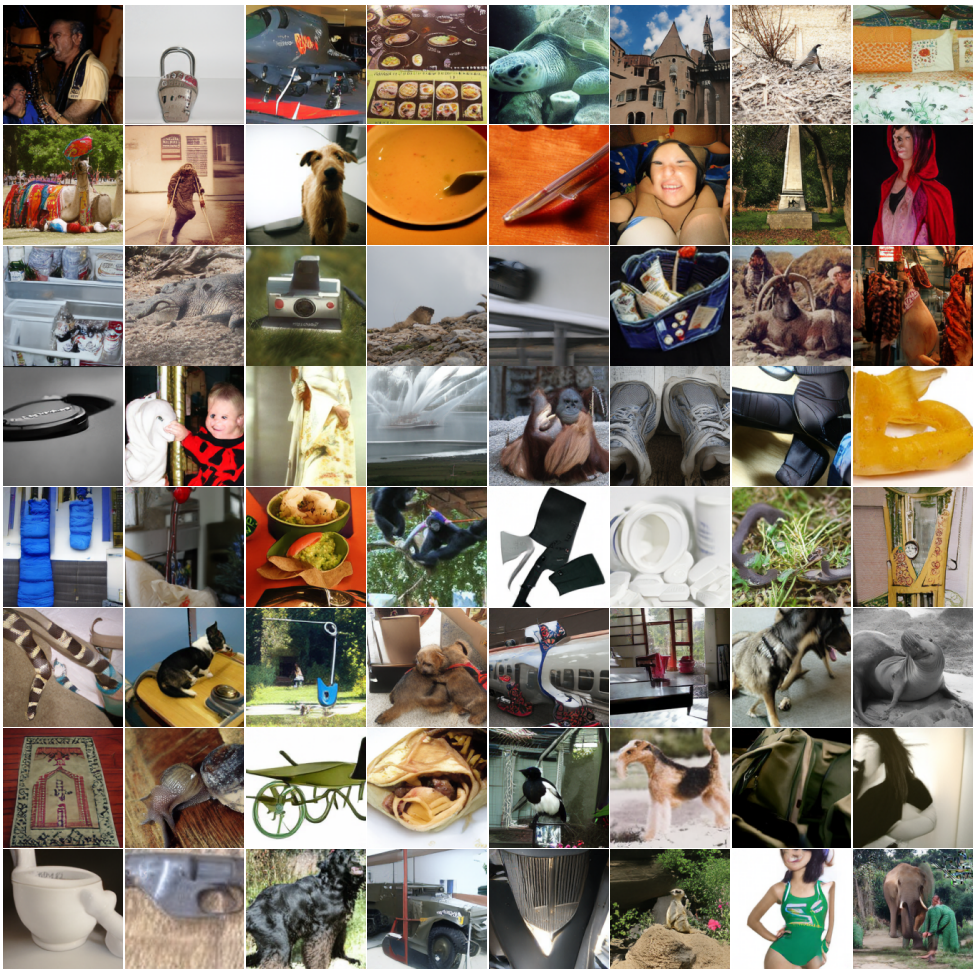

Figure 4: **Uncurated VCQ L-XL samples (1/8).** An 8×8 grid of 64 randomly selected distinct ImageNet classes. Sampling uses LXL with CodebookSize-Aware CFG, cosine CFG with $s = 3$ and $p = 1.0$.

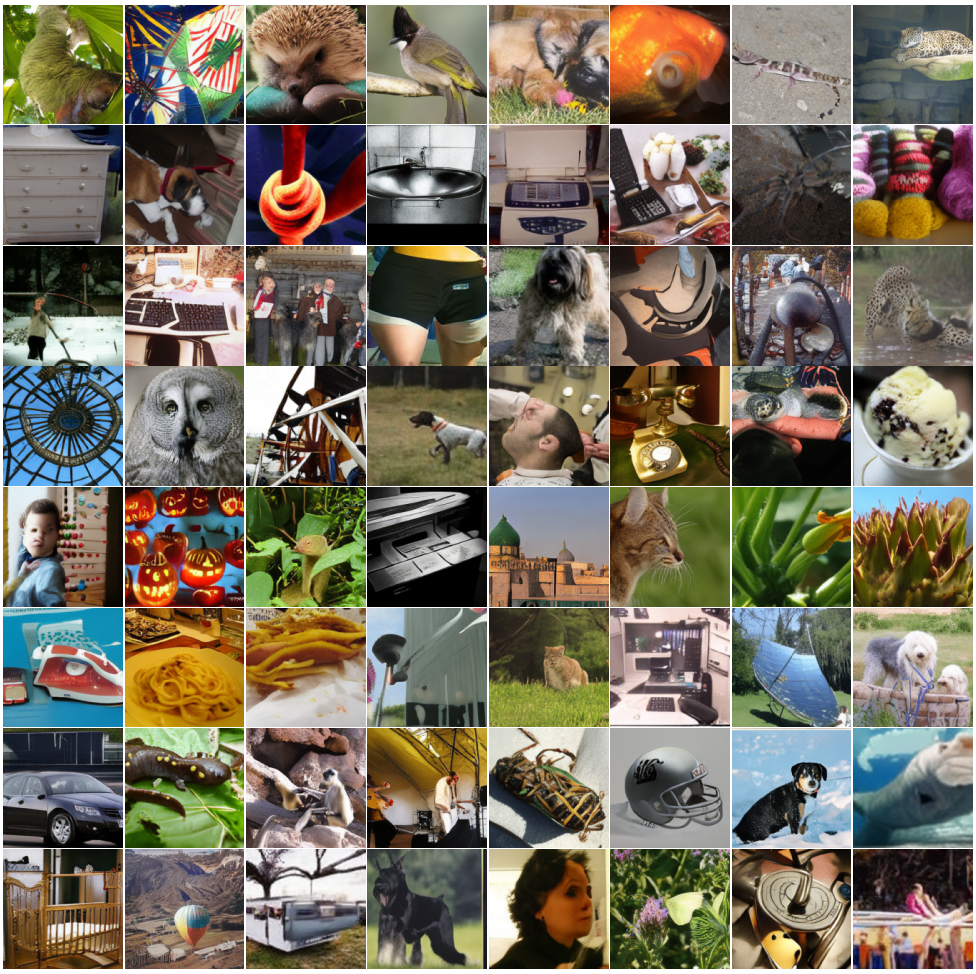

Figure 5: **Uncurated VCQ L-XL samples (2/8).** An 8×8 grid of 64 randomly selected distinct ImageNet classes, disjoint from the other gallery pages. Sampling uses LXL with CodebookSize-Aware CFG, cosine CFG with $s = 3$ and $p = 1.0$.

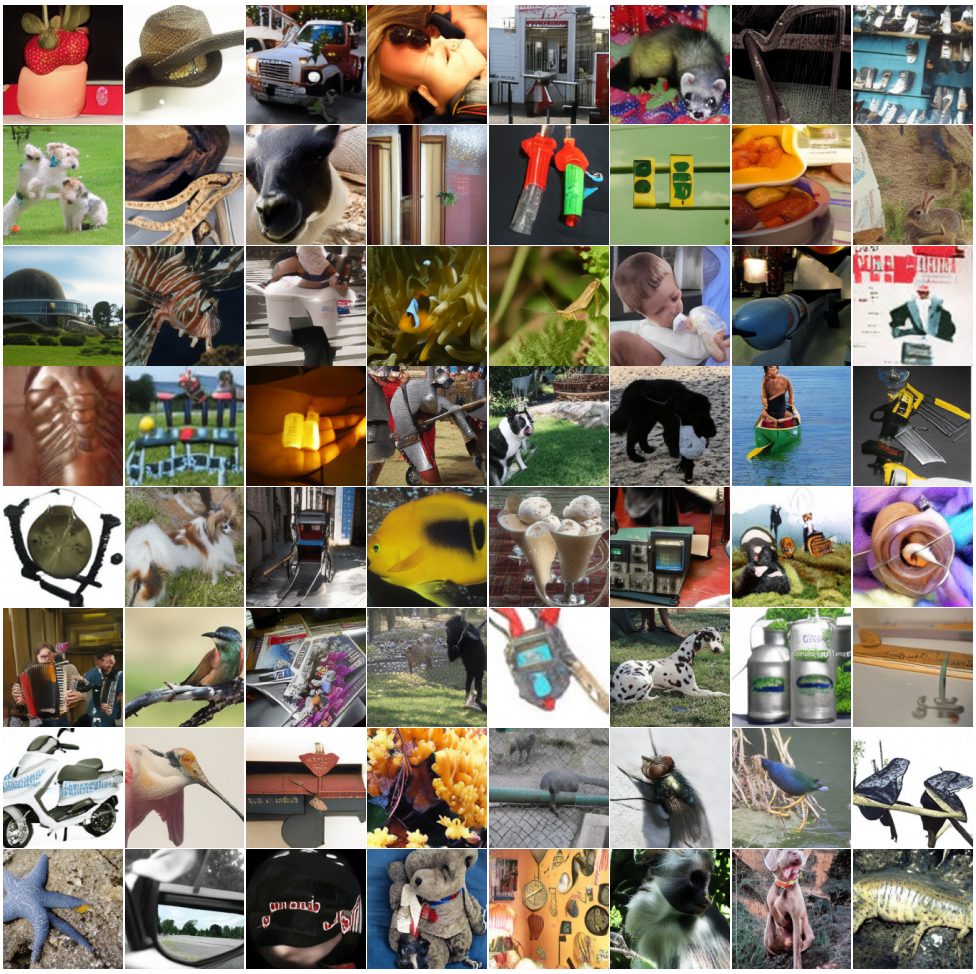

Figure 6: **Uncurated VCQ L-XL samples (3/8).** An 8×8 grid of 64 randomly selected distinct ImageNet classes, disjoint from the other gallery pages. Sampling uses LXL with CodebookSize-Aware CFG, cosine CFG with $s = 3$ and $p = 1.0$.

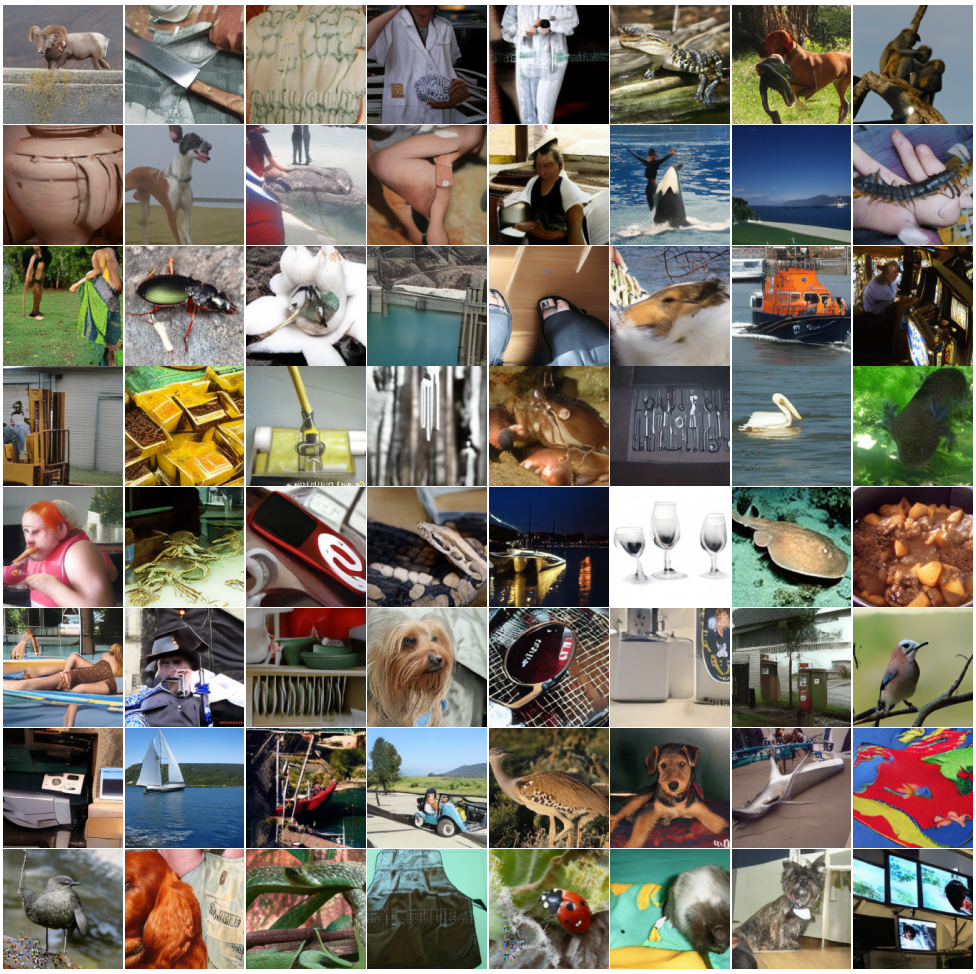

Figure 7: **Uncurated VCQ L-XL samples (4/8).** An 8×8 grid of 64 randomly selected distinct ImageNet classes, disjoint from the other gallery pages. Sampling uses LXL with CodebookSize-Aware CFG, cosine CFG with $s = 3$ and $p = 1.0$.

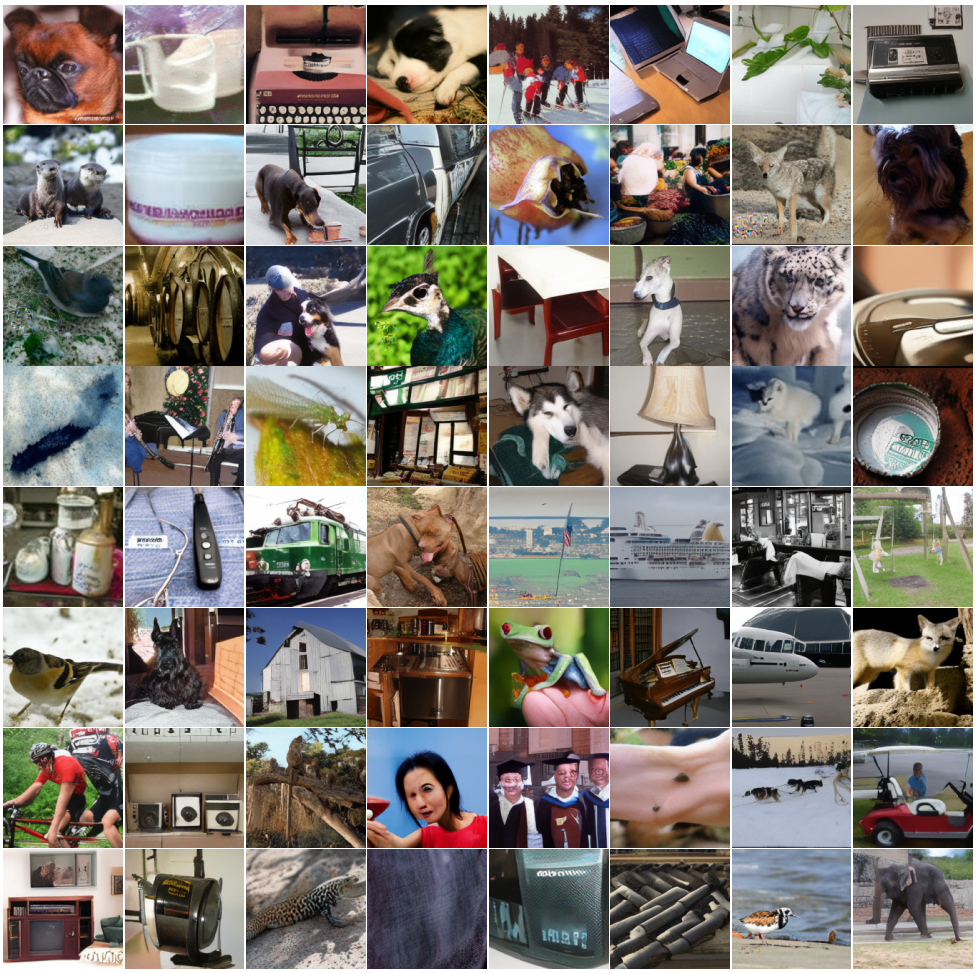

Figure 8: **Uncurated VCQ L-XL samples (5/8).** An 8×8 grid of 64 randomly selected distinct ImageNet classes, disjoint from the other gallery pages. Sampling uses LXL with CodebookSize-Aware CFG, cosine CFG with $s = 3$ and $p = 1.0$.

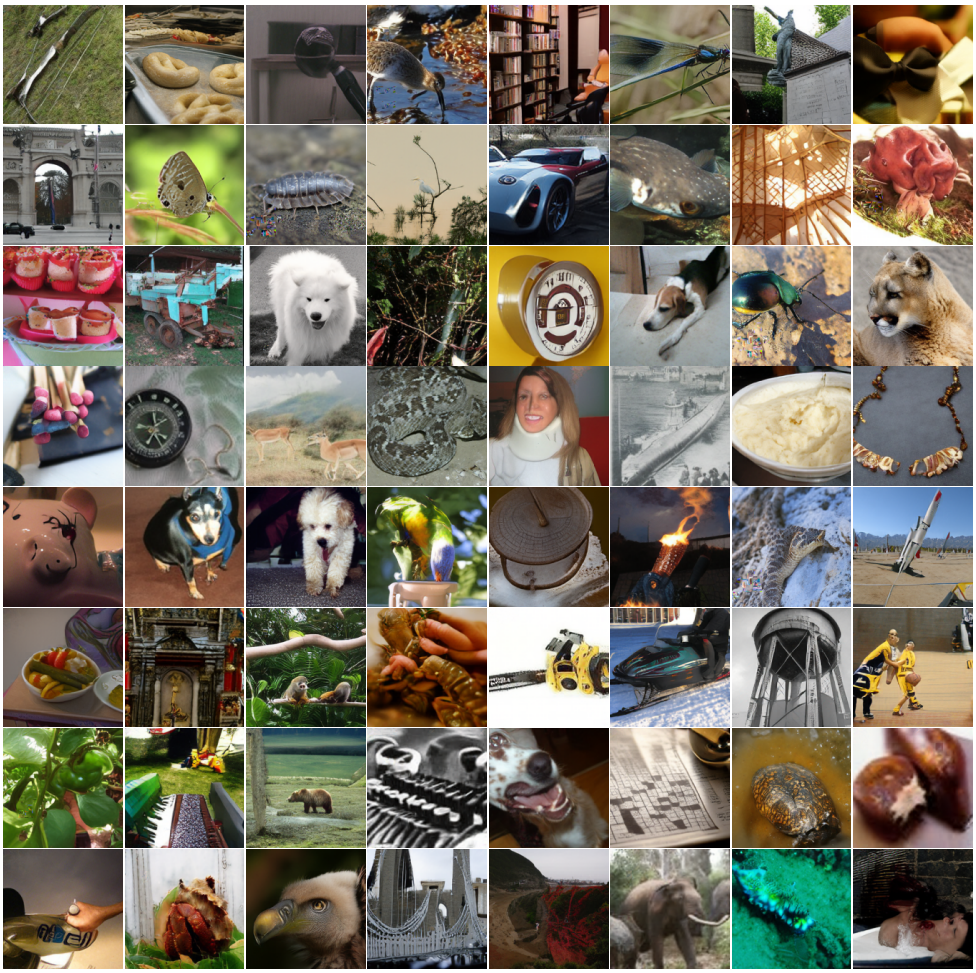

Figure 9: **Uncurated VCQ L-XL samples (6/8).** An 8×8 grid of 64 randomly selected distinct ImageNet classes, disjoint from the other gallery pages. Sampling uses LXL with CodebookSize-Aware CFG, cosine CFG with $s = 3$ and $p = 1.0$.

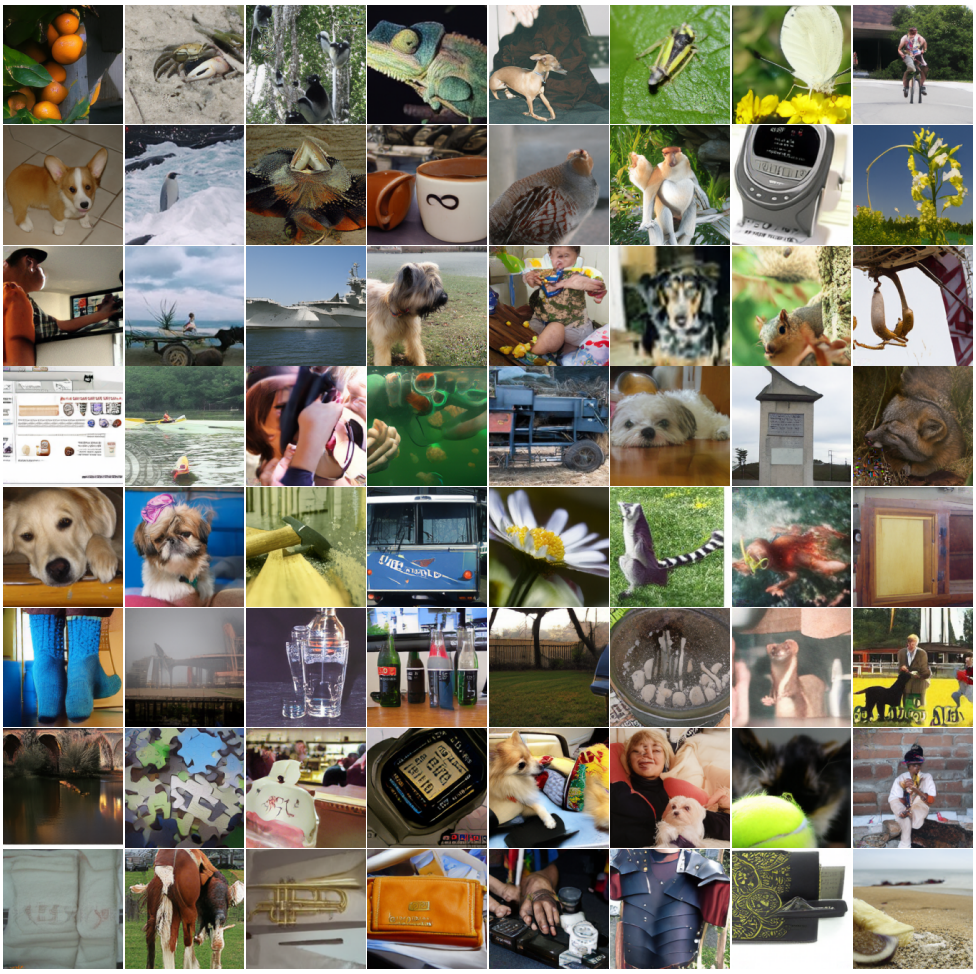

Figure 10: **Uncurated VCQ L-XL samples (7/8).** An 8×8 grid of 64 randomly selected distinct ImageNet classes, disjoint from the other gallery pages. Sampling uses LXL with CodebookSize-Aware CFG, cosine CFG with $s = 3$ and $p = 1.0$.

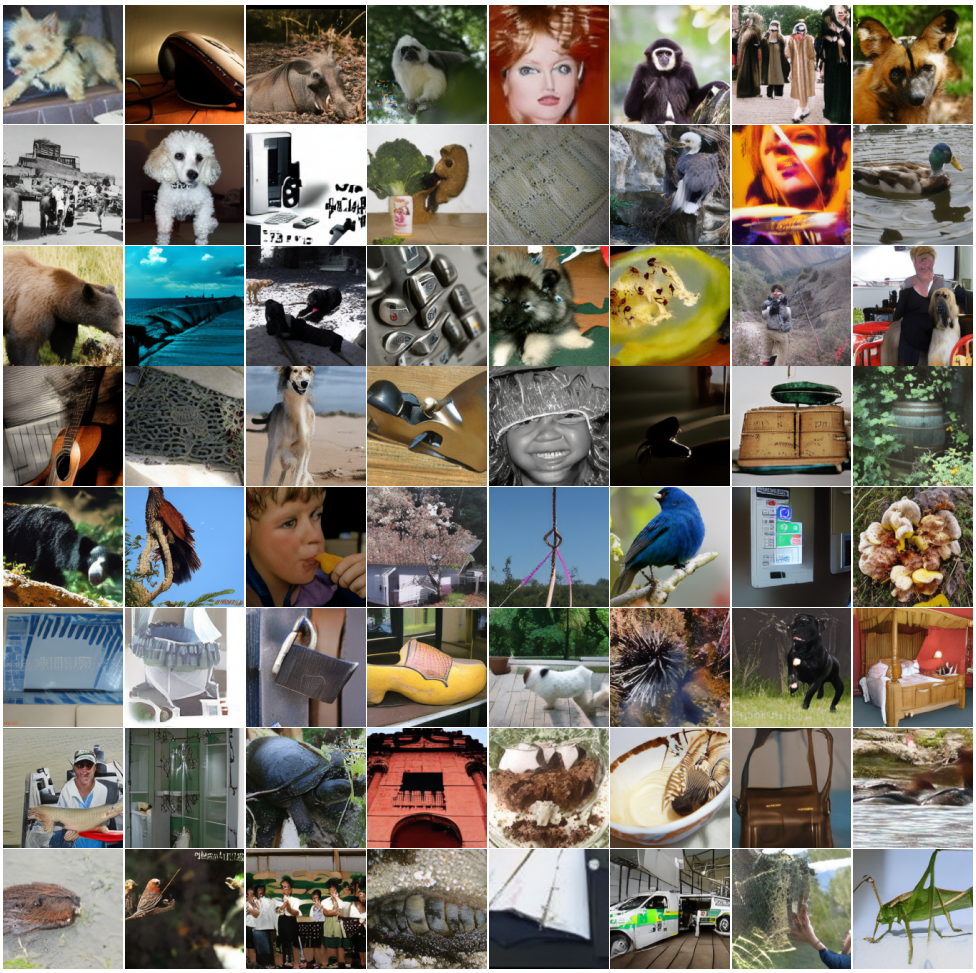

Figure 11: **Uncurated VCQ L-XL samples (8/8).** An 8×8 grid of 64 randomly selected distinct ImageNet classes, disjoint from the other gallery pages. Sampling uses LXL with CodebookSize-Aware CFG, cosine CFG with $s = 3$ and $p = 1.0$.