GigaTok: Scaling Visual Tokenizers to 3 Billion Parameters for Autoregressive Image Generation

Supplementary Material

A. Limitations and Future Work

This study primarily focuses on scaling tokenizers for class-conditional image generation. While we have demonstrated the effectiveness of GigaTok for downstream class-conditional generation, expanding the scope to include text-conditional image generation or video generation remains an open avenue for future work. Additionally, unlike CNN-based 2D tokenizers, 1D Transformer-based tokenizers are not directly applicable to multiple resolutions without additional training adjustments. This challenge presents an important direction for further exploration. Besides scaling the model sizes of tokenizers, the effect of scaling training data, codebook dimension and codebook size for downstream autoregressive generation are left for future research.

B. Configurations for AR models

Size	Params.	Blocks	Heads	Dim.
B	111M	12	12	768
L	343M	24	16	1024
XL	775M	36	20	1280
XXL	1.4B	48	24	1536

Table 1. **Architectures of the LLamaGen models in our experiments.**

AR model training. We scale up the training of downstream Llama-style [19, 21] AR models to compare generation performance with other models. For model training, we use WSD learning rate scheduler [6, 8] with 1×10^{-4} base learning rate, 0.2 decay ratio and 1 epoch warm-up. We do not use AdaLN [17, 20] as it is specific for class-conditional generation. We use a batch size of 256 for training the B, L and XL models and a 512 batch size for training the XXL model. Our AR models are trained for 300 epochs on the 256×256 ImageNet training set.

CFG for gFID. Since gFID of GPT models can be largely affected by classifier free guidance (CFG) [18, 19] and often has an optimal CFG [19], for fair comparison, we search the optimal CFG using zero-order search with a step of 0.25 and use the lowest gFID as the final value. For AR Probing, we use constant CFG scheduling for simplicity. For system-level comparison, we use a step function for CFG scheduling inspired by [11]. Specifically, the AR models predict the first 18% tokens without CFG, *i.e.*, CFG $= 1$ for better diversity, and use CFG for the remaining tokens for better visual quality. Interestingly, we find that the 1.4B LlamaGen model achieves the best gFID without CFG.

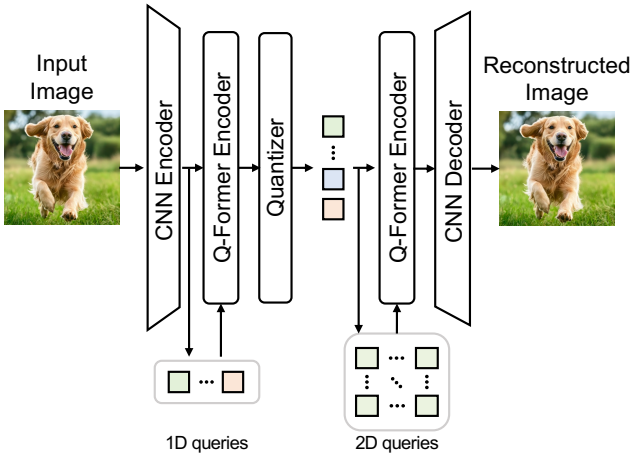

Figure 1. **The architecture of GigaTok with Q-Former.**

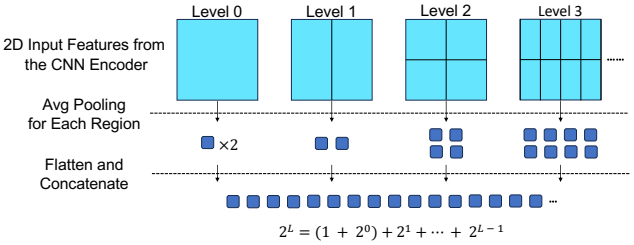

Figure 2. **Initialization of 1D queries in Q-Former modules.**

C. Detailed GigaTok Implementation

Please refer to Tab. 2 for training details.

Q-Fomrer in GigaTok. GigaTok utilizes Q-Former [2, 12] to build 1D tokenizers, as shown in Fig. 1. For Q-Former encoder in GigaTok, we initialize the 1D queries initialized from the 2D input features of the CNN encoder using a multi-level average pooling strategy, as shown in Fig. 2. Specifically, for the same 2D input features, we spatially divide them with different granularity at different levels, and perform average pooling for every divided region at each level. The pooled features are flattened and concatenated from level 0 to the last level. Therefore, a 1D token sequence with 2^L length can be initialized with L levels from 2D input features. At the decoding stage, the 2D queries are all initialized from the first 1D latent feature.

Entropy Loss for VQ Tokenizers. While entropy loss [23,

Configuration	S-S	S-B	S-L	B-L	XL-XXL
Q-Former Encoder depth	6	6	6	12	36
Q-Former Encoder heads	8	8	8	12	20
Q-Former Encoder dim.	512	512	512	768	1280
Q-Former Decoder depth	6	12	24	24	48
Q-Former Decoder heads.	8	12	16	16	24
Q-Former Decoder dim.	512	768	1024	1024	1536
Params (M)	136	232	533	622	2896
Codebook size	16384				
Codebook dimension	8				
#Tokens	256				
Training epochs	100	200	200	200	300
Batch size	128	128	256	256	256
Alignment Layer l	3				
Learning rate schedule	Cosine Decay				
Base learning rate	1×10^{-4}				
Minimum learning rate	1×10^{-5}				
LR warm-up iterations	0	0	0	0	5000
Optimizer	AdamW[14]				
Opt. momentum	$\beta_1 = 0.9, \beta_2 = 0.95$				
Entropy Loss weight	0	0	0	0	5×10^{-3}

Table 2. GigaTok configuration and default training details

24] is discussed for LFQ [24], its application to VQ tokenizers is less commonly explained. We provide a detailed derivation of the entropy loss specifically for VQ tokenizers. Mathematically, for quantization process from continuous vector $\mathbf{z} \in \mathbb{R}^D$ to quantized vector $\hat{\mathbf{z}} = \mathbf{c}_i \in \mathbb{R}^D$ where $\mathbf{c}_i$ is the i-th codebook vector from codebook $\mathbf{C} \in \mathbb{R}^{N \times D}$, we assume this process is statistical and follows the following distribution:

$$p(\hat{\mathbf{z}} = \mathbf{c}_i | \mathbf{z}) \triangleq \text{softmax}(-l_2(\mathbf{z}, \mathbf{C}))[i] \tag{1}$$

where $l_2(\mathbf{z}, \mathbf{C}) \in \mathbb{R}^N$ is the L_2 distance between $\mathbf{z}$ and all the codebook vectors. Then, minimization of the quantization error can be partially achieved by minimizing the expectation of entropy $\mathbb{E}_{\mathbf{z}}\left[H(\hat{\mathbf{z}}|\mathbf{z})\right]$, which can be understood as maximizing the prediction confidence for $p(\hat{\mathbf{z}}|\mathbf{z})$. To encourage higher codebook utilization, we aim to make the average appearance probability of codebook vectors more uniform. This is achieved by maximizing the entropy $H(\hat{\mathbf{z}})$, Therefore, the optimization of the two entropy terms leads to the final entropy loss equation:

$$\mathcal{L}_{\text{entropy}} = \mathbb{E}_{\mathbf{z}}\left[H(\hat{\mathbf{z}}|\mathbf{z})\right] - H(\hat{\mathbf{z}}) \tag{2}$$

In practice, to calculate $H(\hat{\mathbf{z}})$, we estimate $p(\hat{\mathbf{z}} = \mathbf{c}_i)$ by $p(\hat{\mathbf{z}} = \mathbf{c}_i) = \mathbb{E}_{\mathbf{z}}\left[p(\hat{\mathbf{z}} = \mathbf{c}_i|\mathbf{z})\right]$. Note that entropy loss is **not** our contribution. We only provide a detailed definition of entropy loss in VQ scenarios for better understanding.

Additional implementation details. To stabilize the training of our tokenizer with a hybrid architecture, we initially use a shortcut feature reconstruction trick at the first 15k iterations of the tokenizer training. But we later found that this trick can be replaced with a simple 1-epoch learning rate warmup combined with entropy loss [4, 24]. Specifically for this trick, we additionally give the output feature of the CNN encoder to the CNN decoder directly to be trained for reconstruction, and also align the output feature of the Transformer decoder to the output feature of the CNN encoder, besides the original training objectives. Note that this strategy is complex and can even hinder performance for XL-XXL tokenizers. We recommend using the learning rate warmup combined with entropy loss [4, 24] instead, for both XL-XXL tokenizer and the smaller ones. Additionally, we utilize the rotation trick [5] for all tokenizers, though we observe its effect on performance to be limited for our tokenizer. The implementation of the semantic regularization is partially inspired by REPA [26].

D. Full Evaluation Results and Analysis

Here we present the full evaluation results for the tokenizers and downstream AR models, as summarized in Tab. 3. We observe that scaling up visual tokenizers consistently improves the reconstruction quality across multiple metrics. Interestingly, for the 1.4B AR model, the lowest gFID is obtained without applying any CFG. This phenomenon is

Tokenizer	Param.	rFID↓	LPIPS↓	PSNR↑	SSIM↑	AR Model	Param.	gFID↓	Acc.↑	IS↑	Precision↑	Recall↑
LlamaGen-Tok. [19]	72M	2.19	-	20.79	0.675	LlamaGen-B [19]	111M	5.46	-	193.61	0.83	0.45
GigaTok-S-S	136M	1.01	0.2226	20.74	0.670	LlamaGen-B (1d) [19]	111M	4.05	62.6	240.61	0.81	0.51
GigaTok-S-B	232M	0.89	0.2121	20.93	0.677	LlamaGen-B (1d) [19]	111M	3.83	62.9	233.31	0.83	0.51
GigaTok-B-L	622M	0.81	0.2059	21.21	0.685	LlamaGen-B (1d) [19]	111M	3.26	67.6	221.02	0.81	0.56
						LlamaGen-XXL (1d) [19]	1.4B	2.03⋆	69.4	238.52	0.80	0.63
GigaTok-B-L	622M	0.51‡	0.206	21.32	0.691	LlamaGen-B (1d) [19]	111M	3.33	67.7	265.43	0.80	0.56
GigaTok-XL-XXL	2.9B	0.79	0.1947	21.65	0.699	LlamaGen-B (1d) [19]	111M	3.15	72.0	224.28	0.82	0.55
						LlamaGen-XXL (1d) [19]	1.4B	1.98⋆	74.0	256.76	0.81	0.62

Table 3. **Full results for our tokenizers and AR models on ImageNet 256×256.** For gFID, we present the lowest value between w/ or w/o CFG scenarios. ‡: Using frozen DINO [3] for discriminator, which largely improves rFID. ⋆: Without classifier-free-guidance.

also observed in the concurrent work FlexTok [1], despite significant differences between GigaTok and FlexTok. We hypothesize that semantic regularization might be the primary contributing factor for this phenomenon.

Discussion on Scaling and Enhancing the Discriminator. Recently, VAR [20], ImageFolder [13], and the concurrent work UniTok [15] have begun leveraging DINO-based discriminators [3, 16] to enhance tokenizer training, achieving impressive improvements in rFID scores. We have also experimented with the same DINO discriminator configuration as VAR. Our results indicate that although rFID scores improve, the downstream generation quality improvements are less significant, as detailed in Tab. 3. Furthermore, when applying the DINO discriminator to XL-XXL tokenizers, we observed that adversarial training frequently encounters instability. Specifically, a strong discriminator quickly learns to distinguish reconstructed samples, diminishing the benefits of adversarial training and leading to blurry artifacts. We leave further exploration of discriminator scaling and enhancement strategies for future work.

E. Training Tokenizers for More Iterations

While we largely resolve the reconstruction *vs*. generation dilemma regarding tokenizer **model scaling**, this challenge persists for tokenizer **training duration scaling**. To illustrate this phenomenon, we train five S-S tokenizers ranging from 40 to 120 epochs using a cosine learning rate scheduler, as detailed in Tab. 2. The results are presented in Fig. 3.

When extending tokenizer training iterations, reconstruction quality consistently improves. However, downstream generation quality initially improves but subsequently degrades with further increases in tokenizer training duration. Additionally, the validation loss of AR probing continuously rises with longer tokenizer training, regardless of semantic regularization. This trend suggests an increasing complexity in the tokenizer's latent space as the training duration extends.

We hypothesize that data scaling may alleviate this issue, and leave it for future exploration. In practice, allocating computational resources toward model scaling rather than extended training duration may yield better tokenizer performance.

F. Linear Probing Accuracy of Tokenizers

We show that the linear probing accuracy of the tokenizer encoders may not necessarily indicate the performance of downstream AR models. We utilize the intermediate checkpoints during the training of B-L and XL-XXL tokenizers for evaluation. As shown in Fig. 4, the XL-XXL tokenizer encoder presents an overfitting trend in terms of tokenizer encoder linear probing accuracy. However, this overfitting trend is not reflected in AR Probing linear probing accuracy or gFID. Therefore, the linear probing accuracy of the tokenizer encoders may not be a good indicator of downstream model performance. Similarly, a concurrent work UniTok [15], also points out that the performance of the tokenizer encoder in terms of zero-shot ImageNet classification accuracy may not necessarily reflect the visual understanding ability of downstream LLMs trained on the tokenizer.

The abnormality for large tokenizers reveals that the linear probing accuracy of the tokenizer is not necessarily a good indicator for downstream generation models. Since we care more about the representation learning for downstream models than for the tokenizers, using AR Probing as a direct evaluating method is better than indirect tokenizer linear probing accuracy.

G. More Discussions About Related Work

TiTok [25] explores the use of 1D Transformer-based tokenizers under a high compression rate setting. TiTok seminally explores the model scaling of visual tokenizers and uses larger tokenizers for higher compression rate. However, the reconstruction *vs*. generation dilemma for scaling tokenizers is not solved in TiTok. As a result, the best generation model in TiTok is still trained on its smallest tokenizer variant.

ViTok [7] is a concurrent work which has explored the ef-

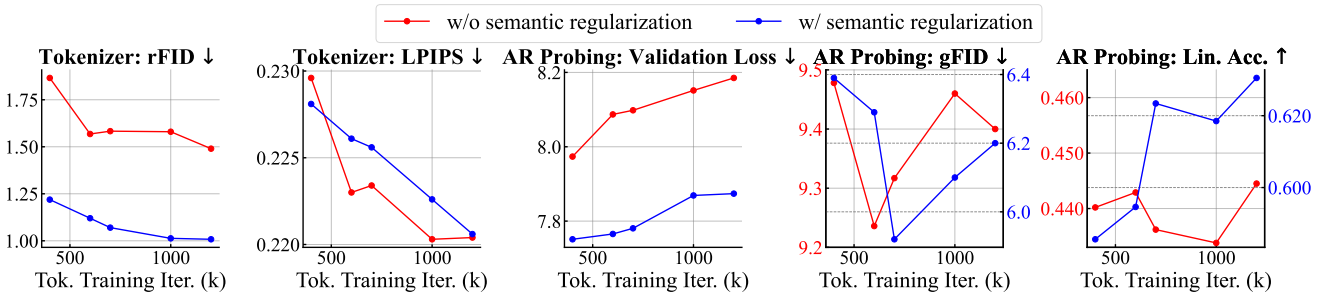

Figure 3. **Training duration scaling trends of tokenizers for reconstruction, downstream generation and representation quality with and without semantic regularization.** Note that in the last two figures, the red and blue curves correspond to different scales on the y-axis.

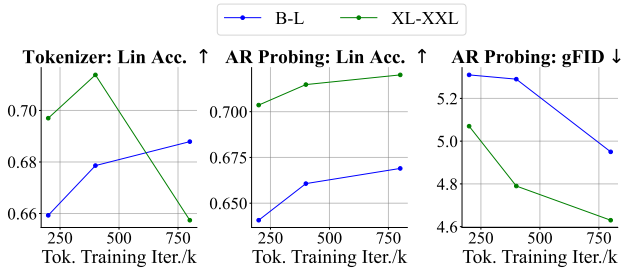

Figure 4. **The linear probing accuracy of tokenizer encoders does not necessarily reflect downstream model performance.** As the training proceeds, the XL-XXL tokenizer encoder presents an overfitting trend measured by linear probing accuracy, but downstream model performances consistently improve.

fect of model scaling for VAE [9]. ViTok evaluates its VAE models in terms of both reconstruction and downstream diffusion generation performance. While having a very different setting from GigaTok, ViTok similarly finds that asymmetric design is better for VAEs. While ViTok suggests that small encoders are optimal, we point out that in our setting scaling encoders is also beneficial. Notably, the reconstruction *vs*. generation dilemma for scaling visual tokenizers is not solved in ViTok. We hypothesize that adding semantic regularization may similarly help solve the tokenizer scaling dilemma for VAEs, but leave it for future study.

MAGVIT-v2 [24] introduces LFQ to enhance discrete tokenizers. It also introduces the entropy penalty for tokenizer training, which is shown to be important for training large-scale tokenizers in our work. Instead of tokenizer model scaling, MAGVIT-v2 focuses more on scaling the codebook size of tokenizers. While codebook dimension and codebook size are important bottlenecks for visual tokenizers, we point out that model size scaling is also an important way for improving visual tokenizers.

ImageFolder [13] utilizes two branches for image encoding to handle high-level semantic information and low-level visual details respectively. It seminally utilizes semantic alignment to enhance the learned representation of tokenizers.

VA-VAE [22] tames the reconstruction *vs*. generation dilemma in increasing latent dimensions for continuous VAE [9, 10]. VA-VAE improves the reconstruction-generation Pareto Frontier by introducing vision foundation model alignment loss. In contrast, we seek continuous improvements in both reconstruction and generation by scaling tokenizers. Semantic regularization serves different purposes in the two works.

References

[1] 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. *arXiv preprint arXiv:2502.13967*, 2025. 3

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 1

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 3

[4] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12873–12883, 2021. 2

[5] Christopher Fifty, Ronald G Junkins, Dennis Duan, Aniketh Iger, Jerry W Liu, Ehsan Amid, Sebastian Thrun, and Christopher Ré. Restructuring vector quantization with the rotation trick. *arXiv preprint arXiv:2410.06424*, 2024. 2

[6] Alexander Hägele, Elie Bakouch, Atli Kosson, Loubna Ben Allal, Leandro Von Werra, and Martin Jaggi. Scaling laws and compute-optimal training beyond fixed training durations. *arXiv preprint arXiv:2405.18392*, 2024. 1

[7] Philippe Hansen-Estruch, David Yan, Ching-Yao Chung, Orr Zohar, Jialiang Wang, Tingbo Hou, Tao Xu, Sriram Vishwanath, Peter Vajda, and Xinlei Chen. Learnings from scaling visual tokenizers for reconstruction and generation. *arXiv preprint arXiv:2501.09755*, 2025. 3

[8] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024. 1

[9] Diederik P Kingma. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 4

[10] Diederik P Kingma, Max Welling, et al. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. 4

[11] Tuomas Kynkäänniemi, Miika Aittala, Tero Karras, Samuli Laine, Timo Aila, and Jaakko Lehtinen. Applying guidance in a limited interval improves sample and distribution quality in diffusion models. *arXiv preprint arXiv:2404.07724*, 2024. 1

[12] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 1

[13] Xiang Li, Kai Qiu, Hao Chen, Jason Kuen, Jiuxiang Gu, Bhiksha Raj, and Zhe Lin. Imagefolder: Autoregressive image generation with folded tokens. *arXiv preprint arXiv:2410.01756*, 2024. 3, 4

[14] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. 2

[15] 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. *arXiv preprint arXiv:2502.20321*, 2025. 3

[16] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023. 3

[17] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023. 1

[18] 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*, pages 10684–10695, 2022. 1

[19] 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. 1, 3

[20] Keyu Tian, Yi Jiang, Zehuan Yuan, BINGYUE PENG, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. 1, 3

[21] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[22] Jingfeng Yao and Xinggang Wang. Reconstruction vs. generation: Taming optimization dilemma in latent diffusion models. *arXiv preprint arXiv:2501.01423*, 2025. 4

[23] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023. 1

[24] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion–tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023. 2, 4

[25] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *arXiv preprint arXiv:2406.07550*, 2024. 3

[26] 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. *arXiv preprint arXiv:2410.06940*, 2024. 2