

## Appendix

This section first illustrates the implementation of VFMTok and AR generation models, covering their learning rates, optimization approaches, and training demands. Subsequently, we present more ablation studies on VFMTok, investigating the effects of different architectural designs on image reconstruction and generation. Besides, all VFMToks in this study utilize a frozen, pre-trained DINOv2-L [2] as their encoder. Next, we show AR image generation with the other VFMs. Subsequently, we answer the question of what makes VFMs a good visual tokenizer. Finally, we demonstrate additional visualization samples of class-to-image generation using VFMTok.

### A VFMTok Implementation

**Tokenizer training.** VFMTok is trained on the ImageNet [3] training set at  $336 \times 336$  resolution with random crop augmentation. All models share identical training settings: a constant learning rate of  $10^{-4}$ , AdamW optimizer [6] ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.95$ , weight decay = 0.05), a batch size of 256, and 50 training epochs. For training losses, the commitment loss weight is 0.25 and the adversarial loss weight is 0.5, with the adversarial loss activated after 20,000 iterations. Besides, VFMTok requires 1.5 days of training on 16 Nvidia H800 GPUs.

**AR model optimization.** The AR model training configuration aligns with LlamaGen’s [8], except our training resolution is  $336 \times 336$  and the duration depends on model parameters. Other key settings include: a base learning rate of  $10^{-4}$  per 256 batch size; AdamW optimizer [6] ( $\beta_1 = 0.9$ ,  $\beta_2 = 0.95$ , weight decay = 0.05, gradient clipping of 1.0); a dropout rate of 0.1 for input token embeddings, attention, and FFN modules; and a 0.1 class condition embedding dropout for classifier-free guidance. Besides, A VFMTok-L requires 19.4 hours of training on 8 NVIDIA H800 GPUs.

### B Supplemental Ablation Study

In this subsection, we conduct more ablation studies on the design of our approach, including the AR image generation with resolution of  $256 \times 256$ , the effect of single-level v.s. multi-level features, the effect of shared ViT v.s. unshared ViT, the impact of different components, and the number of tokens to represent an image. Additionally, the effect of different VFMs on image reconstruction and generation is also presented.

#### B.1 AR Generation with Resolution of $256 \times 256$

In the main paper, given that the input resolution of the vision foundation model is  $336 \times 336$ , we adjust the resolution of reconstructed and generated images to  $336 \times 336$  by default, thus avoiding changing the number of tokens for image representation. Following the common practices [8, 9], we also train the image tokenizer and the AR generation model with the resolution of  $256 \times 256$ , respectively. We first initialize and train VFMTok tokenizer for 50 epochs, then integrate it with AR generative models. Considering computational costs, models with fewer parameters like VFMTok-B and VFMTok-L are trained for 300 epochs, while larger AR models for 50 epochs. This setting aligns with LlamaGen [8] for  $256 \times 256$  image generation. Furthermore, to ensure a fair comparison with the advanced autoregressive generation framework, RAR [11], we also incorporated VFMTok into RAR [11], maintaining the same setup during the training phase. As shown in Tab. 1, VFMTok not only achieves a decent reconstruction performance but also improves the generation quality compared to its counterparts [8, 11] by a large margin. It is worth noting that VFMTok also accelerates the convergence speed during AR model training and significantly improves synthesis quality.

#### B.2 Effect of Single-level v.s. Multi-level Features

In this paragraph, we investigate the separate impacts of each single-level or multi-level features on image reconstruction and generation. Specifically, we begin by conducting an ablation study to evaluate the impact of each single-layer feature on image reconstruction and generation. Subsequently, we cumulatively add each feature layer to observe the combined effect.

**Setup.** We initialize and train each tokenizer for 50 epochs. Once the tokenizer is optimized, it is integrated with an AR generation model, VFMTok-L, for a total of 100 training epochs. During evaluation, the image reconstruction and generation performance are reported.

**Observation.** The results, as presented in the Table 2 and Table 3, show that while a single feature layer offers no significant advantage, the quality of both image reconstruction and generation markedly improves as more layers are incorporated.

Table 1: VFMTok performs image reconstruction and AR generation with the size of  $256 \times 256$ .

Approach	Image recon.			code usage $\uparrow$		AR gen.			
	#Toks	rFID $\downarrow$	rIS $\uparrow$	$Q_C$	$Q_P$	#Epochs	Para.	gFID $\downarrow$	gIS $\uparrow$
LlamaGen-B	256	2.22	169.8	–	95.2%	300	111M	5.46	193.6
LlamaGen-L							343M	3.81	248.3
LlamaGen-XL						50	775M	3.39	227.1
LlamaGen-XXL							1.4B	3.09	253.6
LlamaGen-3B							3.1B	3.06	279.7
RAR-L [11]	256	2.12	171.4	–	100.0%	400	461M	1.70	299.5
RAR-XL [11]							955M	1.50	306.9
RAR-XXL [11]							1.5B	1.48	326.0
VFMTok-B	256	1.02	213.2	100.0%	–	100	111M	3.95	248.4
VFMTok-L							343M	3.02	271.6
VFMTok-B						300	111M	3.61	247.6
VFMTok-L							343M	2.79	276.0
VFMTok-XL						50	775M	2.79	277.1
VFMTok-XXL							1.4B	2.62	279.7
VFMTok-2B							2B	2.64	284.0
VFMTok(RAR-L)	256	0.88	216.2	–	100.0%	400	461M	1.47	<b>316.2</b>
VFMTok(RAR-XL)							955M	<u>1.38</u>	303.3
VFMTok(RAR-XXL)							1.5B	<b>1.30</b>	300.0

Table 2: Performance of each single-level feature.  $F_i$  represents the indexed feature level.

$F_i$	Image recon.				AR gen.	
	#Toks	rFID $\downarrow$	rIS $\uparrow$	$Q_C$	gFID $\downarrow$	gIS $\uparrow$
$F_1$	256	1.04	186.4	100.0%	3.84	257.9
$F_2$		0.95	200.6		3.79	272.7
$F_3$		1.03	208.5		3.69	274.9
$F_4$		1.23	214.8		3.64	277.7

Table 3: Performance of cumulatively added features.  $F_i$  represents the indexed feature level.

$F_i$	Image recon.				AR gen.	
	#Toks	rFID $\downarrow$	rIS $\uparrow$	$Q_C$	gFID $\downarrow$	gIS $\uparrow$
$F_1$	256	1.04	186.4	100.0%	3.84	257.9
$+F_2$		0.94	205.0		3.69	274.4
$+F_3$		0.94	210.8		3.27	272.5
$+F_4$		0.89	215.4		3.09	274.2

### B.3 Effect of Shared ViT v.s. Unshared ViT

In this work, VFMTok utilizes a shared ViT to generate latent features for pixel rendering and high-level VFM feature (specifically, from the last layer) reconstruction, respectively. However, it is uncertain if the sharing parameter is optimal. To this end, we experimented with another unshared ViT of the same architecture to generate the high-level VFM feature. Following the training setup, we train the tokenizer and AR model – VFMTok-L for 50 epochs. As shown in Tab. 4, the shared ViT ensures a better image reconstruction and synthesis quality with enriched semantics. Therefore, we utilize shared ViT in the VFMTok design by default.

### B.4 Effect of Different Components

In VFMTok, we introduce three key features: a frozen foundation (*e.g.*, DINOv2 [2]) model as the encoder, multi-level plain feature interaction, and reconstructing the pre-trained feature, to achieve a notable performance in image reconstruction and generation. To achieve this, we incorporate several learnable modules into VFMTok. Namely, the mask tokens  $M_I$  and  $M_f$  for image and its pre-trained feature reconstruction, the deformable attention layer adopted in the deformable transformer, and a set of learnable register tokens to address the potential artifacts in the latent feature. Thus, some questions naturally emerge: 1) Whether the mask token could be shared between  $M_I$  and  $M_f$ ; 2) Could the deformable attention layer be replaced with the vanilla cross-attention layer; 3) Can the register tokens be discarded from VFMTok?

**Setup.** To answer the above questions, we conducted 3 different experiments: 1)  $M_l$  and  $M_f$  share the same mask token  $M_{\text{share}}$ ; 2) Replacing deformable attention with cross-attention in deformable transformer. To address the memory, computational demands, and fairness considerations of cross-attention, we first concatenate multi-level features from a VFM along the channel dimension, then apply a single MLP for dimensionality reduction before these features are fed to the cross-attention transformer for interaction with queries. Besides, each query interacts with VFM features within a  $16 \times 16$  window to simulate region-adaptive behavior; 3) Removing the register tokens from VFMTok. Additionally, linear probing is carried out on the  $[\text{CLS}]$  token to estimate the semantic representation capability of VFMTok.

**Observation.** As shown in Tab. 5, unshared mask tokens seldom affect image reconstruction or generation quality but significantly degrade VFMTok’s overall semantic representation. This indicates that image reconstruction requires a certain amount of semantic information, but this semantic information is not as strong as that requested for VFM’s feature reconstruction. Besides, Tab. 5 also reveals that shared mask tokens can enhance the semantic representation of VFMTok, potentially benefiting downstream comprehension tasks. Hence, we use shared mask tokens by default in VFMTok. Introducing cross-attention for learnable queries and multi-level feature interaction shows an evident decline in image reconstruction, generation, and semantic level. Compared to deformable attention that focuses on local regions, using cross-attention may introduce redundant information, this redundancy weakens the overall semantic representation by complicating quantized visual token distribution, thereby hindering image generation. Additionally, a slight decrease in image reconstruction, generation, as well as the semantic representation is also observed when register tokens  $\text{Tok}_{\text{reg}}$  are removed from VFMTok. This suggests that introducing register tokens into VFMTok is reasonable since they can eliminate some artifacts in the features.

Table 5: Impact study on mask embeddings, attention, and registered tokens.

$M_{\text{share}}$	Attention		$\text{Tok}_{\text{reg}}$	<i>Image Recon.</i>			<i>AR gen.</i>		L.P.
	Cross-Attn	Deform-Attn		#Toks	rFID↓	rIS↑	gFID↓	gIS↑	
✓	✓		✓	256	1.00	211.5	3.89	271.5	34.1
✓		✓	✓		0.91	215.8	3.45	276.5	68.3
		✓	✓		0.89	216.1	3.50	276.0	64.7
✓		✓	✓		0.89	215.4	3.42	277.3	69.4

## B.5 Effect of the Number of Tokens to Represent an Image

In this work, we utilize a set of region-adaptive tokens to represent an image. The number of these tokens is empirically fixed at 256 throughout our experiments. However, the impact of tokens’ cardinality on image reconstruction and generation quality remains unexplored.

**Setup.** To investigate this, we designed a controlled setup to isolate the impact of query quantity while maintaining architectural consistency across experiments. Specifically, we parameterize the image tokenizer with varying numbers of learnable queries. Each tokenizer variant is trained on the ImageNet [3] dataset for 50 epochs. Subsequently, the trained tokenizer is integrated into an AR image generation model, LlamaGen-L, and trained for 50 epochs. Both image reconstruction and generation quality are estimated on ImageNet [3] dataset with FID and IS, respectively.

**Observation.** As show in Tab. 6, image reconstruction exhibits a positive correlation with the tokens’ cardinality. As the number of tokens increases, metrics for estimating image reconstruction, namely rFID and rIS, both show gradual improvement. However, this trend does not hold for generation: as the number of tokens increases to higher values, there is no significant improvement in generation quality. Therefore, to balance generation quality with computational cost, and maintain fairness with vanilla counterparts [4, 8, 14], we fix the number of tokens at 256 across our experiments. Actually, it’s observed that **144** visual tokens suffice for representing

Table 6: The effect of the number of tokens on image reconstructions and generation

#Tokens	<i>Image recon.</i>		<i>AR gen.</i>	
	rFID↓	rIS↑	gFID↓	gIS↑
36	2.61	175.4	3.93	222.4
64	2.09	188.3	3.59	250.5
100	1.44	204.4	3.54	270.8
121	1.36	202.5	3.59	267.1
144	1.20	204.6	3.46	274.9
169	1.11	212.4	3.42	275.5
196	1.08	213.7	3.32	271.2
225	0.96	215.5	3.45	279.2
256	0.89	215.4	3.42	277.3
289	0.85	217.2	3.41	272.3
361	0.80	218.8	3.64	277.9
400	0.79	221.3	3.36	272.3
441	0.73	221.8	3.61	275.5
576	0.60	222.8	3.57	278.8

images in ImageNet [3]. This finding indicates VFMTok can further eliminate redundancy within image representations, yielding more compact and more effective image compression.

## B.6 Effect of Different VFMs

Here, we explore employing different frozen VFMs [2, 10, 12] as the encoder of VFMTok, and incorporate them into different AR generative frameworks – LlamaGen [8] and RAR [11]. As shown in Tab. 7, VFMTok implemented with SigLIP [12] also consistently yields decent image reconstruction and generation quality based on AR models of different scales. Additionally, we also observe a potential correlation between the AR synthesis quality and different VFMs. VFM with stronger representation capabilities will achieve higher generation performance.

Table 7: Class-conditional image generation with different VFMs.

Type	Method	#Epoch	#Para.	#Tok.	Generation w/ CFG				Generation w/o CFG			
					gFID	sFID	gIS	Pre. Rec.	gFID	sFID	gIS	Pre. Rec.
AR.	VFMTok-B(SigLIP)	300	111M	256	3.53	5.76	254.4	0.85 0.51	3.75	5.80	156.3	0.79 0.58
	VFMTok-L(SigLIP)		343M		2.61	5.54	272.1	0.84 0.56	2.11	5.65	214.0	0.81 0.61
	VFMTok-XXL(SigLIP)	200	1.4B		2.09	5.75	272.6	0.82 0.60	1.85	5.78	251.2	0.81 0.61
	VFMTok-2B(SigLIP)		2.2B		2.05	5.77	271.4	0.82 0.61	1.92	5.78	260.5	0.82 0.61
	VFMTok-XL(DINOv2)	300	775M	256	2.41	5.53	276.8	0.83 0.59	2.10	5.54	258.1	0.82 0.60
	VFMTok-XL(SigLIP)				2.42	5.70	275.1	0.83 0.59	1.99	5.75	241.9	0.81 0.61
	VFMTok-XL(SigLIP2)				2.20	<b>5.38</b>	272.1	0.83 0.59	1.93	5.43	253.3	0.81 0.60
	RAR-L(SigLIP)	400	461M	256	1.46	6.18	<b>312.9</b>	0.78 0.64	2.26	5.78	204.3	0.79 0.62
	RAR-XL(SigLIP)		955M		1.40	6.39	311.7	0.78 0.65	1.87	5.77	226.7	0.79 0.63
	RAR-XXL(SigLIP)		1.5B		<b>1.38</b>	6.18	298.4	0.78 0.65	1.68	5.64	239.5	0.79 0.63
	RAR-XL(SigLIP2)	400	461M	256	1.50	6.37	292.7	0.77 0.66	2.05	5.50	217.7	0.79 0.62
	RAR-XL(SigLIP2)		955M		1.46	6.02	288.4	0.78 0.65	1.72	<b>5.41</b>	244.8	0.80 0.63
	RAR-XXL(SigLIP2)		955M		1.43	6.07	291.0	0.79 0.65	<b>1.65</b>	5.44	<b>266.5</b>	0.81 0.63

## B.7 What makes a VFM a good visual tokenizer?

To answer this problem, we first leverage existing vision foundation models, which were supervised with different learning objectives, *e.g.* masked image modeling in pixel or latent space (Pixel-MIM and Latent-MIM) and contrastive learning (C.L) – to construct VFMTok for image reconstruction and generation, and subsequently provide more insights on this question.

**Setup.** The encoder of vanilla VQGAN [8] is substituted with different vision foundation models (VFMs), which are supervised with distinct learning objectives. All of these tokenizers are trained on the ImageNet [3] training set for 50 epochs. Subsequently, these tokenizers are incorporated into an AR generative model – LlamaGen-L [8] and trained for 100 epochs. Both image reconstruction and generation quality are evaluated on the ImageNet validation set with FID and IS, respectively. Additionally, the top-1 accuracy of linear probing is also reported.

**Observation** As shown in the Table. 8, Pixel-MIM (MAE [5]) aligns best to reconstruction objectives, thus, VQGAN(MAE) achieves the optimal image reconstruction quality. However, it does not provide a feature space that is friendly for generation tasks. Its semantics is also inferior to the other VFMs. Contrastive learning-only models, VQGAN(CLIP) and VQGAN(SigLIP) achieve the best semantics (top-1 accuracy in linear probing), but modest image reconstruction and generation performance. Without a mask image modeling objective, they lack sufficient capability to model the local structures. Latent-MIM and contrastive learning co-optimized approaches, VQGAN(SigLIP2), and VQGAN(DINOv2) achieve the best overall performance.

**Discussion** The training objectives of vision foundation models(VFMs) affect the image reconstruction and generation of VFMTok. CLIP [7] and SigLIP [12] apply contrastive learning to image-text pairs, whereas DINO [1] applies a contrastive learning-like clustering objective to images. MAE [5] performs mask image modeling on image patches in pixel space, and iBOT [13] predicts DINO-like cluster assignments of image patches in latent space. DINOv2 [2] basically is the combination of DINO [1] and iBOT [13], and SigLIP2 [10] is also essentially the integration of SigLIP [12] and iBOT [13] (termed TIPS loss in their paper), omitting other regularizing losses. We hereby provide a holistic comparison of the training objectives of different VFMs in Table 9.

Table 8: Image reconstruction and generation with VFM of a distinct learning objective.

Tokenizer	Pixel-MIM	Latent-MIM	C.L.	#tok.	rFID↓	rIS↑	gFID↓	gIS↑	L.P.(%)↑
VQGAN(Clip [7])	✗	✗	✓	576	1.47	182.0	3.45	221.2	59.5
VQGAN(SigLIP [12])	✗	✗	✓		1.26	190.8	3.50	246.1	60.3
VQGAN(SigLIP2 [10])	✗	✓	✓		0.96	198.4	3.39	267.8	55.5
VQGAN(DINOv2 [2])	✗	✓	✓		0.99	206.3	3.34	268.6	56.4
VGQGAN(MAE [5])	✓	✗	✗		0.67	207.6	3.40	265.5	39.0

Notably, iBOT [13], DINOv2 [2], and SigLIP2 [10] are performing codebook learning resemble that in VQGAN on image patches, despite codes being soft instead of one-hot, and their codebook vectors being high-dimensional (*e.g.*, 256). Following DINO [1], they learn a set of 8,192 or 65,536 prototypes, and require two augmented versions of the same patch (or image) to have identical prototype assignments (soft codes). They also predict the soft codes of masked patches, thus performing mask image modeling(Latent-MIM) in latent space. This formulation is termed self-distillation in DINO [1], and then extended to patches with Laent-MIM by iBOT [13], which was subsequently inherited by DINOv2 [2] and SigLIP2 [10].

Table 9: The category of learning objectives adopted in VFMs.

VFM	P-MIM	L-MIM	C.L.
CLP [7]	✗	✗	✓
DINO [1]	✗	✗	✓
SigLIP [12]	✗	✗	✓
MAE [5]	✓	✗	✗
iBOT [13]	✗	✓	✗
SigLIP2 [10]	✗	✓	✓
DINOv2 [2]	✗	✓	✓

The connection between Latent-MIM and VQGAN makes it natural to convert these VFMs to tokenizers and expect decent image reconstruction and generation performance. Additionally, the global (image-level) contrastive learning objectives of DINOv2 [2] and SigLIP2 [10] also promise better high-level semantics (better performance on understanding tasks).

To summarize, the conclusion can be drawn as: masked image modeling(MIM) objectives primarily help reconstruction and generation. MIM in pixel space helps reconstruction more, but is less beneficial for generation than MIM in latent space. Contrastive learning(C.L.) objectives are less helpful for reconstruction and generation, but are important for understanding abilities (*e.g.*, top-1 accuracy on ImageNet [3]). Best VFMs for visual tokenization are trained with both mask image modeling in latent space and contrastive objectives, namely DINOv2 [2] and SigLIP2 [2].

## C Limitations

Beyond the decent improvement achieved by VFMTok the optimal architectural design of VFMTok and its scalability on large-scale datasets are still under exploration. Currently, VFMTok’s image reconstruction is not yet optimal, primarily due to limitations in its codebook design and model size. For fair comparison and maintaining a concise design, we adopt a vanilla codebook and keep VFMTok’s size comparable to VQGAN’s [8]. While advanced codebook designs and increased model size could further enhance reconstruction quality. Beyond architecture design, exploring the scalability of VFMTok on large-scale datasets is also crucial for developing a superior tokenizer and advancing towards unified generation and understanding tasks. These aspects, however, outline promising directions for our future research.

## D Broader Impacts

The advancements in image tokenizers and autoregressive (AR) image generation present significant broader impacts. Positively, these technologies can democratize content creation across art, design, and media, accelerate scientific research through enhanced data augmentation and visualization, and enable novel forms of personalized digital experiences. However, this transformative potential is accompanied by substantial ethical challenges. Key concerns include the proliferation of realistic misinformation (deepfakes), the potential for misuse in creating non-consensual or harmful content, the amplification of societal biases embedded in training data, and complex issues surrounding intellectual property and copyright. Furthermore, the computational resources required for training state-of-the-art models raise environmental considerations. Therefore, the responsible development and deployment of these powerful tools necessitate robust frameworks for ethical guidelines, bias detection and mitigation, provenance tracking, and public awareness to harness their benefits while minimizing societal risks.





Figure 1: Class-conditional image generation with CFG.





Figure 2: Class-conditional image generation without CFG.

## References

- [1] 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 International Conference on Computer Vision (ICCV)*, 2021.
- [2] Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. *arXiv preprint arXiv:2309.16588*, 2023.
- [3] 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*, pages 248–255, 2009.
- [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.
- [5] 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*, pages 16000–16009, 2022.
- [6] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [7] 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 *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021.
- [8] 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.
- [9] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *arXiv preprint arXiv:2404.02905*, 2024.
- [10] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.
- [11] Qihang Yu, Ju He, Xueqing Deng, Xiaohui Shen, and Liang-Chieh Chen. Randomized autoregressive visual generation. *arxiv*, 2024.
- [12] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.
- [13] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer. *International Conference on Learning Representations (ICLR)*, 2022.
- [14] Lei Zhu, Fangyun Wei, Yanye Lu, and Dong Chen. Scaling the codebook size of vqgan to 100,000 with a utilization rate of 99%. *arXiv preprint arXiv:2406.11837*, 2024.