A. Additional Theoretical and Empirical Analysis

A.1. KL-VAE Formulation

In this section, we provide a detailed description of KL-VAE [21, 25]. KL-VAE models both the prior and posterior distributions as Gaussians. Specifically, the prior $p(z)$ is defined as an isotropic unit Gaussian $\mathcal{N}(0, \mathbf{I})$. The posterior distribution $q_\phi(z|x)$ is parameterized by an encoder that predicts the mean $\mu_\phi(x)$ and variance $\sigma^2_\phi(x)$. Using the reparameterization trick, the latent variable z is obtained as

$$\begin{aligned} q_\phi(z|x) &= \mathcal{N}(z; \mu_\phi(x), \sigma^2_\phi(x)), \\ z &= \mu_\phi(x) + \sigma_\phi(x) \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, \mathbf{I}). \end{aligned} \tag{8}$$

The KL divergence between the posterior and the prior is given by

$$\begin{aligned} &\mathcal{L}_{\text{KL}}(q_\phi(z)\|p(z)) \\ &= \int q_\phi(z|x)\big(\log q_\phi(z|x) - \log p(z)\big)\, dz \\ &= -\frac{1}{2}\sum_{i=1}^{D}\big(1 + \log(\sigma_i^2) - \mu_i^2 - \sigma_i^2\big), \end{aligned} \tag{9}$$

where D denotes the dimensionality of the latent space.The KL term plays a crucial role in the overall training objective, i.e., the Evidence Lower Bound (ELBO). Specifically, it acts as a regularizer that enforces the learned posterior $q_\phi(z|x)$ to stay close to the prior $p(z)$, thereby encouraging smooth and continuous representations.

A.2. Mitigating Posterior Collapse via Masked Reconstruction

A.2.1. Corrupted Evidence Lower Bound (ELBO)

Standard VAE training optimizes the Evidence Lower Bound (ELBO):

$$\mathcal{L}_{\text{ELBO}} = \mathbb{E}_{q_\phi(Z|X)}[\log p_\theta(X|Z)] - \beta \cdot \text{KL}(q_\phi(Z|X)\|p(Z)), \tag{10}$$

which balances reconstruction (first term) against regularization of the posterior $q_\phi(Z|X)$ toward the prior $p(Z)$ (second term). Under strong compression and large β, this KL penalty can push $q_\phi(Z|X)$ too close to $p(Z)$, causing *posterior collapse*: $q_\phi(Z|X) \approx p(Z)$. At this point, Z carries no information about X, and the decoder effectively becomes an unconditional model $p_\theta(X)$, leading to poor reconstructions.

MacTok takes a different approach by training on *masked* images. Let $\tilde{X}$ be the masked image after applying a stochastic masking operation $C_m(\tilde{X}|X)$ with ratio m. The encoder sees only $\tilde{X}$, but the decoder must still reconstruct the full image X. This gives us the *corrupted ELBO*:

$$\begin{aligned} \mathcal{L}_{\text{corrupted}} = \mathbb{E}_{X,\tilde{X}\sim C_m(\cdot|X)}[\mathbb{E}_{q_\phi(Z|\tilde{X})}[-\log p_\theta(X|Z)] \\ + \beta \cdot \text{KL}(q_\phi(Z|\tilde{X})\|p(Z))]. \end{aligned} \tag{11}$$

The key difference is this information asymmetry: the encoder only gets partial information $\tilde{X}$, while the decoder has to predict everything, including what was masked. This forces Z to actually encode useful information from $\tilde{X}$—otherwise the decoder has no way to reconstruct the missing parts.

A.2.2. Why Collapsed Solutions Become Suboptimal

Consider what happens when the posterior collapses: $q_\phi(Z|\tilde{X}) = p(Z)$. Now Z is independent of both $\tilde{X}$ and X, so:

$$\begin{aligned} \mathbb{E}_{q_\phi(Z|\tilde{X})=p(Z)}[-\log p_\theta(X|Z)] &= \mathbb{E}_{Z\sim p(Z)}[-\log p_\theta(X|Z)] \\ &= \mathbb{E}_{Z\sim p(Z)}[-\log p_\theta(X)] \\ &= -\log p_\theta(X), \end{aligned} \tag{12}$$

where $p_\theta(X)$ is just the unconditional image distribution.

We can break this down by what's visible versus what's masked:

$$-\log p_\theta(X) = -\log p_\theta(X_{\text{visible}}) - \log p_\theta(X_{\text{masked}}). \tag{13}$$

The problem is the second term: $-\log p_\theta(X_{\text{masked}})$. Without any context, the decoder has to guess what's in the masked regions based purely on dataset statistics—maybe "skies are usually blue" or "grass is usually green." But this fails for any specific image. As we mask more pixels (higher m), this blind guessing gets worse and $-\log p_\theta(X)$ shoots up.

Compare this to when Z actually encodes information from $\tilde{X}$. Now the decoder can use contextual clues—if it sees grass and trees in the visible parts, it knows this is probably an outdoor scene; if the visible colors are warm, maybe it's sunset. This capability of recovering latent details from partial or degraded visual cues shares underlying principles with robust image processing pipelines designed for severely suboptimal conditions. This gives much better predictions:

$$-\log p_\theta(X|Z) = -\log p_\theta(X_{\text{visible}}|Z) - \log p_\theta(X_{\text{masked}}|Z), \tag{14}$$

where $-\log p_\theta(X_{\text{masked}}|Z)$ is now significantly smaller because the decoder can make informed guesses based on what Z encoded.

Let's define the benefit of having an informative Z as:

$$\Delta \triangleq -\log p_\theta(X) - \mathbb{E}_{q_\phi(Z|\tilde{X})}[-\log p_\theta(X|Z)]. \tag{15}$$

Larger Δ means Z is more useful. Now compare total losses:

$$\text{Loss}_{\text{collapse}} = -\log p_\theta(X), \tag{16}$$

$$\text{Loss}_{\text{informative}} = \mathbb{E}_{q_\phi(Z|\tilde{X})}[-\log p_\theta(X|Z)] + \beta \cdot \epsilon, \tag{17}$$

where $\epsilon = \text{KL}(q_\phi(Z|\tilde{X}) \| p(Z)) > 0$ is the KL cost of keeping Z informative. The informative solution wins when:

$$\Delta > \beta \cdot \epsilon. \tag{18}$$

So the collapsed solution is suboptimal whenever $\beta < \Delta/\epsilon$.

Here's where masking matters: it directly increases Δ. As we mask more:

- Without context (collapsed case), predicting more masked pixels becomes exponentially harder, pushing $-\log p_\theta(X)$ way up.
- With context from Z (informative case), we can still make reasonable predictions based on visible cues, so $\mathbb{E}[-\log p_\theta(X|Z)]$ stays relatively controlled.

Higher m widens the gap Δ, which means informative posteriors stay optimal for a broader range of β (Eq. 18).

Without masking, there's a loophole: the decoder can just copy local patterns from the input. Even if Z is mostly useless, reconstructions still look okay, so Δ stays small and collapse becomes competitive. Masking closes this loophole—the decoder *has to* use Z to fill in the missing parts, which keeps information flowing through the latent space even under strong regularization.

In conclusion, masking prevents collapse through a simple mechanism. First, it makes the reconstruction task harder, so Z needs to be informative. Second, if Z collapses and becomes useless, the decoder is forced to blindly guess large portions of the image, incurring huge losses. Third, by increasing Δ, masking ensures that keeping Z informative remains the better strategy across a wide range of β values. This is how MacTok maintains meaningful continuous tokens even with aggressive compression and regularization.

A.3. Visualization of KL Divergence Dynamics

As illustrated in Fig. 7, applying latent token masking postpones posterior collapse compared to the conventional KL-VAE baseline. Nevertheless, this improvement is transient, as the model ultimately converges to a degenerate solution over the course of training. In contrast, masking image tokens yields a markedly steadier optimization process and produces more resilient latent representations. We attribute this behavior to the fact that image masking encourages both the encoder and decoder to reason over incomplete visual inputs, thereby encouraging the latent space to encode more structural and semantic information.

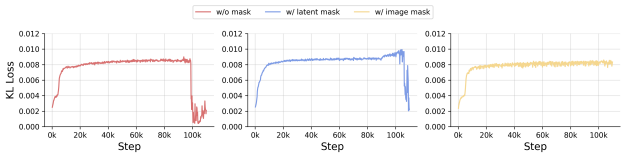


Figure 7. Comparison of different masking strategies of the KL loss curve.

B. Additional Implementation Details

In this section, we present additional implementation details for tokenizer training and downstream generative model training.

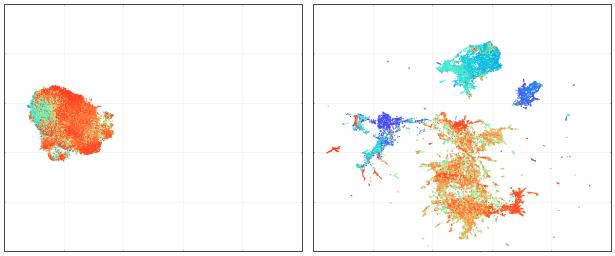
B.1. Implementation Details of MacTok

We train the MacTok tokenizers on ImageNet at resolution of 256×256 for 250K iterations with a batch size of 256 and at 512×512 for 500K iterations with a batch size of 128. Data augmentation includes horizontal flipping and center cropping. We use AdamW optimizer with $\beta_1 = 0.9, \beta_2 = 0.95$, a weight decay of 1×10^{-4}. The learning rate follows a cosine annealing schedule, peaking at 1×10^{-4} and preceded by a linear warm-up of 5K and 10K steps for the 256 and 512 resolutions. To improve the stability of adversarial learning, we employ a frozen DINO-S [2, 37] network as the discriminator as in [5, 46] and incorporate the adaptive weighting scheme. Moreover, we enhance discriminator training by introducing DiffAug [67], consistency regularization [64], and LeCAM regularization [47], as used in [5]. The regularization weights for the consistency and LeCAM terms are set to 4.0 and 0.001, respectively. The overall training objective follows common practice with loss weights $\lambda_1 = 1.0$, $\lambda_2 = 0.2$, $\lambda_3 = 1 \times 10^{-6}$, and $\lambda_4 = 0.1$.

B.2. Implementation Details of Generative Models

LightningDiT [57] The training configuration of our LightningDiT models closely follows the original setup. As our model operates on 1D latent tokens, we set the patch size to 1. LightningDiT-XL is trained with a constant learning rate of 2×10^{-4} and a global batch size of 1024. We adopt a cosine noise scheduler and rotary positional embeddings, consistent with the original implementation. In the main paper, we report results of LightningDiT-XL trained for 400K iterations. For conditional generation with classifier-free guidance (CFG), we use a guidance scale of 2.5 for LightningDiT models trained on MacTok with 128 tokens and 2.7 for those trained with 64 tokens. These values are selected via grid search based on gFID and IS metrics computed over 10K generated samples.

SiT [36] We follow the original training configuration of SiT, using a constant learning rate of 1×10^{-4} and a



(a) MacTok-64 w/o RA. (b) MacTok-64

Figure 8. Visualization of laten space from (a) MacTok-64 trained without Representation alignment; (b) MacTok-64

global batch size of 256. A linear learning rate scheduler is adopted, as it demonstrates better empirical performance in our setting. The main results are reported after 4M training iterations. For conditional generation with CFG, we set the guidance scale to 2.3 for SiT models trained on MacTok with 128 tokens and 2.4 for those trained with 64 tokens. Following REPA [61], the guidance interval is set to $[0, 0.7]$ for CFG-based results. The optimal values are determined through grid search by evaluating gFID and IS over 10K generated samples.

C. Additional Results

In this appendix, we provide supplementary evidence to support the effectiveness of our approach. Specifically, we include further visualizations of the latent token space, more ablation studies, extended quantitative evaluations of generative models trained on MacTok, and additional qualitative examples of reconstructed and generated images. These results complement the main paper by highlighting the structural organization of the latent space, the generative fidelity across different resolutions and token settings.

C.1. Latent Space Visualization

Fig. 8 illustrates the UMAP projection of the latent representations obtained with 64 tokens. We compare the latent space learned by MacTok-64 with and without Representation alignment (RA). As shown, MacTok-64 with Representation alignment generates more structured and separable embeddings compared to the model trained without alignment. This visualization confirms that MacTok effectively organizes the latent space with fewer tokens, supporting downstream tasks such as linear probing and generative modeling, and showing great promise for broader spatial perception applications that require dense structural consistency.

C.2. Ablation Study

Decoder Fine-tuning. Tab. 6a reports MacTok's performance when freezing the encoder and fine-tuning only the decoder without masking. Specifically, the encoder is frozen and the decoder is trained for 10 epochs without mask. This strategy notably improves rFID and slightly enhances gFID, indicating that decoder fine-tuning effectively restores reconstruction quality degraded by high mask ratios while preserving the latent space.

Table 6. Ablation studies of decoder fine-tuning and model size, showing their effects on MacTok's performance

Tokenizer	rFID↓	gFID↓
MacTok-64	0.93	3.28
+FT	0.75	3.22
MacTok-128	0.57	3.19
+FT	0.43	3.15

(a) Decoder fine-tuning.

Model Size	#Params	rFID
MacTok-S	45M	0.78
MacTok-B	176M	0.57
MacTok-BL	391M	0.57

(b) MacTok model size.

Model Size. Tab. 6b evaluates MacTok model size on ImageNet at 256×256. MacTok-B significantly outperforms MacTok-S, whereas further scaling does not yield additional gains. Consequently, MacTok-B is adopted as the default. For 512×512 generation with 64 tokens, we use MacTok-BL to ensure fair comparison with SoftVQ-VAE and mitigate reconstruction degradation at higher resolution.

C.3. Main Results

We present the complete quantitative results, including both precision and recall, for the ImageNet 256×256 and 512×512 benchmarks in Tab. 7 and Tab. 8, respectively. All evaluations are conducted on SiT-XL models trained for 4M steps and LightningDiT-XL models trained for 400K steps. Notably, our models achieve state-of-the-art generative performance at 512×512 resolution and deliver results comparable to leading approaches at 256×256 resolution. Moreover, our models exhibit superior conditional gFID scores even without applying classifier-free guidance (CFG), outperforming SoftVQ-VAE [5] and other vanilla generative baselines [1, 36, 38, 41, 61] that utilize at least 256 or 1024 tokens. We further include results measured across different training durations, as summarized in Tab. 9.

C.4. Reconstruction Visualization

We present the reconstruction results of MacTok using 64 and 128 latent tokens in Fig. 9 and Fig. 10, respectively. As shown, increasing the number of tokens leads to finer spatial details and improved texture fidelity, demonstrating the scalability of MacTok's latent representation. In contrast, reconstructions from collapsed baselines (see Fig. 11) fail to recover meaningful visual content, indicating that posterior collapse severely limits the model's representational capacity. MacTok's semantically structured latent space effectively preserves both global layout and local semantics, resulting in faithful and perceptually consistent reconstruc-

tions even under limited token budgets. These visualizations complement the quantitative evaluation in the main paper and further verify the robustness of our latent modeling strategy.

C.5. Generation Visualization

More visualizations of LightningDiT-X and SiT-XL trained on MacTok with 64 and 128 tokens are provided here.

Table 7. **System-level comparison** on ImageNet 256×256 conditional generation. We report both Precision and Recall under classifier-free guidance (CFG) and non-CFG settings. "# Params (G)" denotes generator parameters; "Tok. Model" refers to the tokenizer model type; "Token Type" indicates 1D or 2D tokenization; "# Params (T)" denotes tokenizer parameters; and "# Tokens" represents the number of latent tokens.

Method	# Params(G)	Tok. Model	Token Type	# Params(T)	#Tokens↓	Tok. rFID↓	w/o CFG				w/ CFG			
							gFID↓	IS↑	Prec↑	Recall↑	gFID↓	IS↑	Prec↑	Recall↑
Auto-regressive														
ViT-VQGAN [58]	1.7B	VQ	2D	64M	1024	1.28	4.17	175.1	-	-	-	-	-	-
RQ-Trans. [28]	3.8B	RQ	2D	66M	256	3.20	-	-	-	-	3.80	323.7	-	-
MaskGIT [3]	227M	VQ	2D	66M	256	2.28	6.18	182.1	0.80	0.51	-	-	-	-
LlamaGen-3B [45]	3.1B	VQ	2D	72M	576	2.19	-	-	-	-	2.18	263.3	0.80	0.58
WeTok [69]	1.5B	VQ	2D	400M	256	0.60	-	-	-	-	2.31	276.6	0.84	0.55
VAR [46]	2B	MSRQ	2D	109M	680	0.90	-	-	-	-	1.92	323.1	0.75	0.63
MaskBit [51]	305M	LFQ	2D	54M	256	1.61	-	-	-	-	1.52	328.6	-	-
MAR-H [34]	943M	KL	2D	66M	256	1.22	2.35	227.8	0.79	0.62	1.55	303.7	0.81	0.62
l-DeTok [56]	479M	KL	2D	172M	256	0.62	1.86	238.6	0.82	0.61	1.35	304.1	0.81	0.62
TiTok-S-128 [60]	287M	VQ	1D	72M	128	1.61	-	-	-	-	1.97	281.8	-	-
GigaTok [53]	111M	VQ	1D	622M	256	0.51	-	-	-	-	3.15	224.3	0.82	0.55
ImageFolder [35]	362M	MSRQ	1D	176M	286	0.80	-	-	-	-	2.60	295.0	0.75	0.63
Diffusion-based														
LDM-4 [41]	400M		2D				10.56	103.5	0.71	0.62	3.60	247.7	0.87	0.48
U-ViT-H/2 [1]	501M	KL	2D	55M	4096	0.27	-	-	-	-	2.29	263.9	0.82	0.57
MDTv2-XL/2 [15]	676M		2D				5.06	155.6	0.72	0.66	1.58	314.7	0.79	0.65
DiT-XL/2 [38]	675M		2D				9.62	121.5	0.67	0.67	2.27	278.2	0.79	0.65
SiT-XL/2 [36]	675M	KL	2D	84M	1024	0.62	8.30	131.7	0.68	0.67	2.06	270.3	0.83	0.53
+REPA [61]			2D				5.90	157.8	0.70	0.69	1.42	305.7	0.82	0.59
LightningDiT [57]	675M	KL	2D	70M	256	0.28	2.17	205.6	-	-	1.35	295.3	-	-
TexTok-256 [62]	675M	KL	1D	176M	256	0.73	-	-	-	-	1.46	303.1	0.79	0.64
MAETok [4]	675M	AE	1D	176M	128	0.48	2.31	216.5	0.78	0.62	1.67	311.2	0.81	0.63
SoftVQ-VAE [5]	675M	SoftVQ	1D	176M	64	0.88	5.98	138.0	0.74	0.64	1.78	279.0	0.80	0.63
Ours														
MacTok+LightningDiT	675M	KL	1D	176M	64	0.75	4.15	167.8	0.75	0.65	1.68	307.3	0.77	0.66
					128	0.43	3.12	186.2	0.75	0.66	1.50	299.8	0.78	0.67
MacTok+SiT-XL	675M				64	0.75	3.77	181.6	0.77	0.63	1.58	310.4	0.78	0.66
					128	0.43	2.82	189.2	0.77	0.64	1.44	302.5	0.79	0.66

Table 8. **System-level comparison** on ImageNet 512×512 conditional generation. We report both Precision and Recall under classifier-free guidance (CFG) and non-CFG settings.

Method	# Params(G)	Tok. Model	Token Type	# Params(T)	#Tokens↓	Tok. rFID↓	w/o CFG				w/ CFG			
							gFID↓	IS↑	Prec↑	Recall↑	gFID↓	IS↑	Prec↑	Recall↑
GAN														
BigGAN [3]	-	-	-	-	-	-	-	-	-	-	8.43	177.9	-	-
StyleGAN-XL [24]	168M	-	-	-	-	-	-	-	-	-	2.41	267.7	-	-
Auto-regressive														
MaskGIT [3]	227M	VQ	2D	66M	1024	1.97	7.32	156.0	-	-	-	-	-	-
MAGVIT-v2 [59]	307M	LFQ	2D	116M	1024	-	-	-	-	-	1.91	324.3	-	-
MAR-H [34]	943M	KL	2D	66M	1024	-	2.74	205.2	0.69	0.59	1.73	279.9	0.77	0.61
TiTok-B-128 [60]	177M	VQ	1D	202M	128	1.52	-	-	-	-	2.13	261.2	-	-
TiTok-L-64 [60]	177M	VQ	1D	644M	64	1.77	-	-	-	-	2.74	221.1	-	-
Diffusion-based														
ADM [10]	-	-	-	-	-	-	23.24	58.1	-	-	3.85	221.7	0.84	0.53
U-ViT-H/4 [1]	501M		2D				-	-	-	-	4.05	263.8	0.84	0.48
DiT-XL/2 [38]	675M		2D				9.62	121.5	-	-	3.04	240.8	0.84	0.54
SiT-XL/2 [36]	675M	KL	2D	84M	4096	0.62	-	-	-	-	2.62	252.2	0.84	0.57
DiT-XL [38]	675M		2D				9.56	-	-	-	2.84	-	-	-
UViT-H [1]	501M		2D				9.83	-	-	-	2.53	-	-	-
UViT-H	501M	AE	2D	323M	256	0.22	12.26	-	-	-	2.66	-	-	-
UViT-2B [1]	2B		2D				6.50	-	-	-	2.25	-	-	-
TexTok-128 [62]	675M	KL	1D	176M	128	0.97	-	-	-	-	1.80	305.4	0.81	0.63
MAETok [4]	675M	AE	1D	176M	128	0.62	2.79	204.3	0.81	0.62	1.69	304.2	0.82	0.62
SoftVQ-VAE [5]	675M	SoftVQ	1D	391M	64	0.71	7.96	133.9	0.73	0.63	2.21	290.5	0.85	0.59
Ours														
MacTok+SiT-XL	675M	KL	1D	391M	64	0.89	4.63	163.7	0.80	0.61	**1.52**	306.0	0.80	0.63
				176M	128	0.79	5.12	156.3	0.79	0.61	**1.52**	**316.0**	0.80	0.63

Table 9. Generation performance over training of SiT-XL trained on MacTok with 64 and 128 tokens.

Method	Training Iter.	w/o CFG				w/ CFG			
		FID	IS	Prec.	Recall	FID	IS	Prec.	Recall
MacTok-64	400K	7.60	121.4	0.72	0.63	2.15	268.2	0.77	0.63
	1M	5.34	147.7	0.74	0.63	1.73	290.4	0.77	0.65
	2M	4.58	159.9	0.75	0.63	1.60	303.0	0.78	0.65
	3M	3.98	174.7	0.76	0.63	1.60	308.2	0.78	0.66
	4M	3.77	181.6	0.77	0.63	1.58	310.4	0.78	0.66
MacTok-128	400K	6.45	127.2	0.73	0.63	1.97	253.2	0.77	0.64
	1M	4.31	153.6	0.75	0.64	1.60	271.7	0.77	0.65
	2M	3.69	168.5	0.75	0.65	1.48	287.0	0.78	0.66
	3M	3.28	176.2	0.76	0.65	1.45	293.1	0.78	0.66
	4M	2.82	189.2	0.77	0.64	1.44	302.5	0.79	0.66

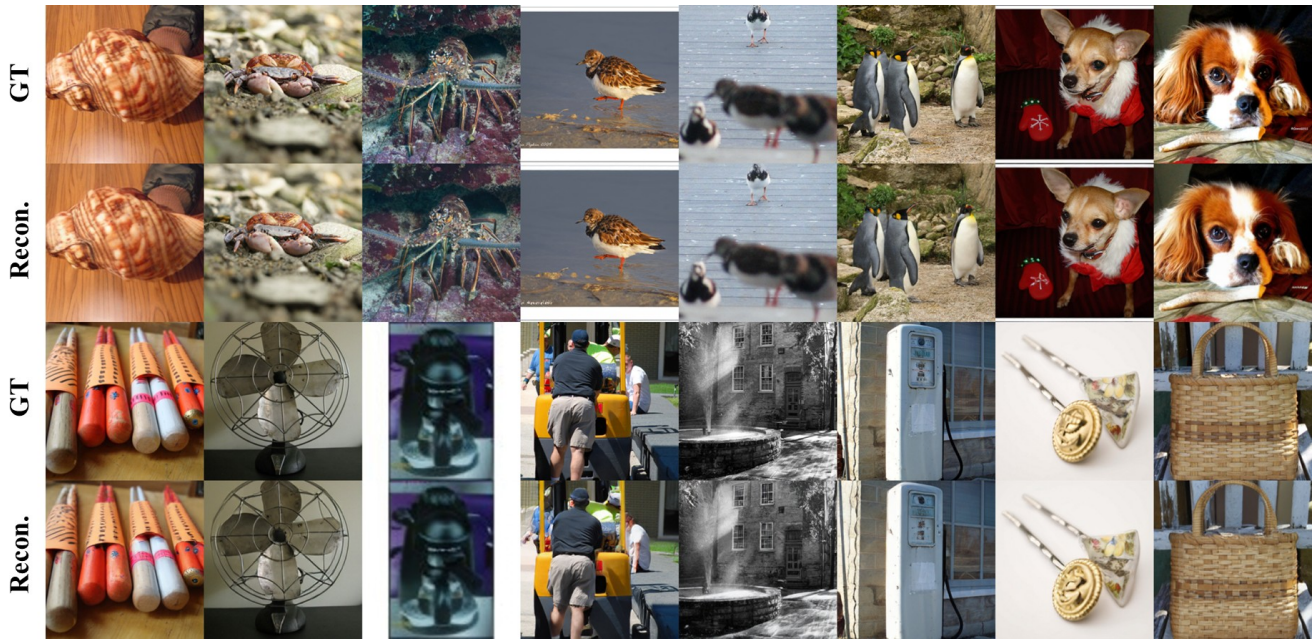


Figure 9. Reconstruction results of MacTok with 64 tokens.

Figure 10. Reconstruction results of MacTok with 128 tokens.

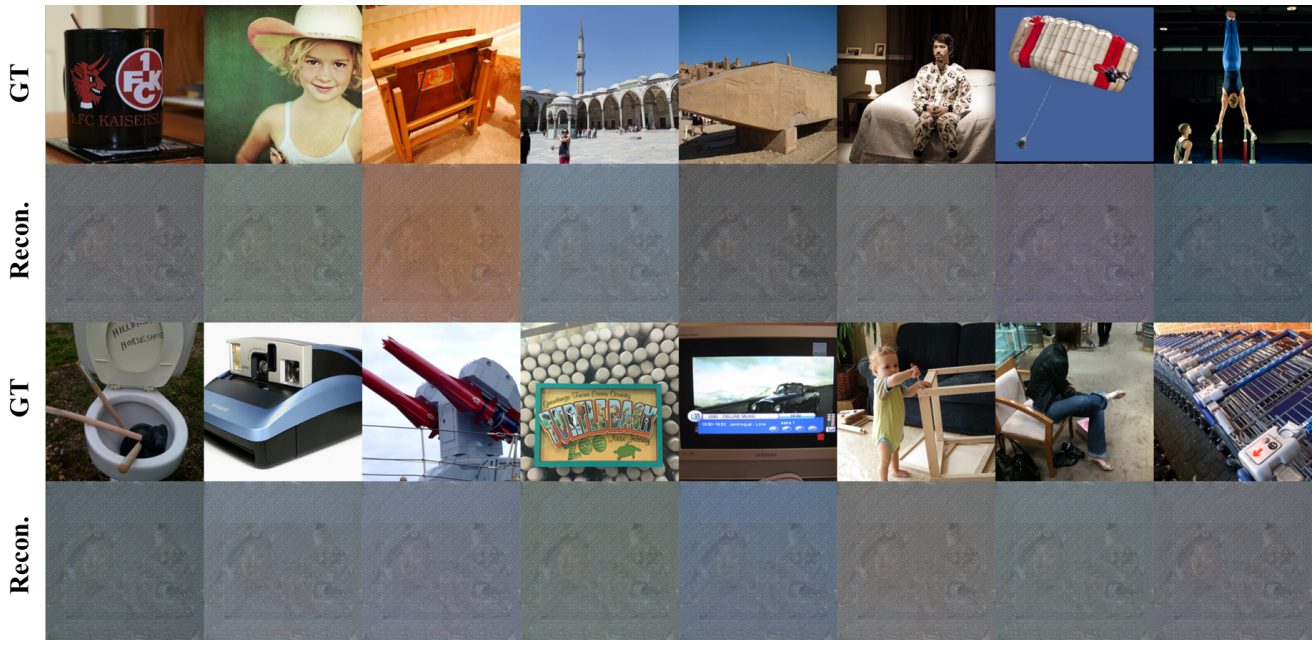


Figure 11. Reconstruction results of collapsed KL-VAE.

Figure 12. Uncurated 256×256 generation results of SiT-XL with MacTok 128 tokens. We use CFG with 4.0. Class label =“loggerhead turtle” (33).

Figure 13. Uncurated 256×256 generation results of SiT-XL with MacTok 128 tokens. We use CFG with 4.0. Class label =“macaw” (88).

Figure 14. Uncurated 256×256 generation results of SiT-XL with MacTok 128 tokens. We use CFG with 4.0. Class label =“Kakatoe galerita” (89).

Figure 15. Uncurated 256×256 generation results of SiT-XL with MacTok 128 tokens. We use CFG with 4.0. Class label =“golden retriever” (207).

Figure 16. Uncurated 256×256 generation results of SiT-XL with MacTok 128 tokens. We use CFG with 4.0. Class label =“Arctic wolf” (270).

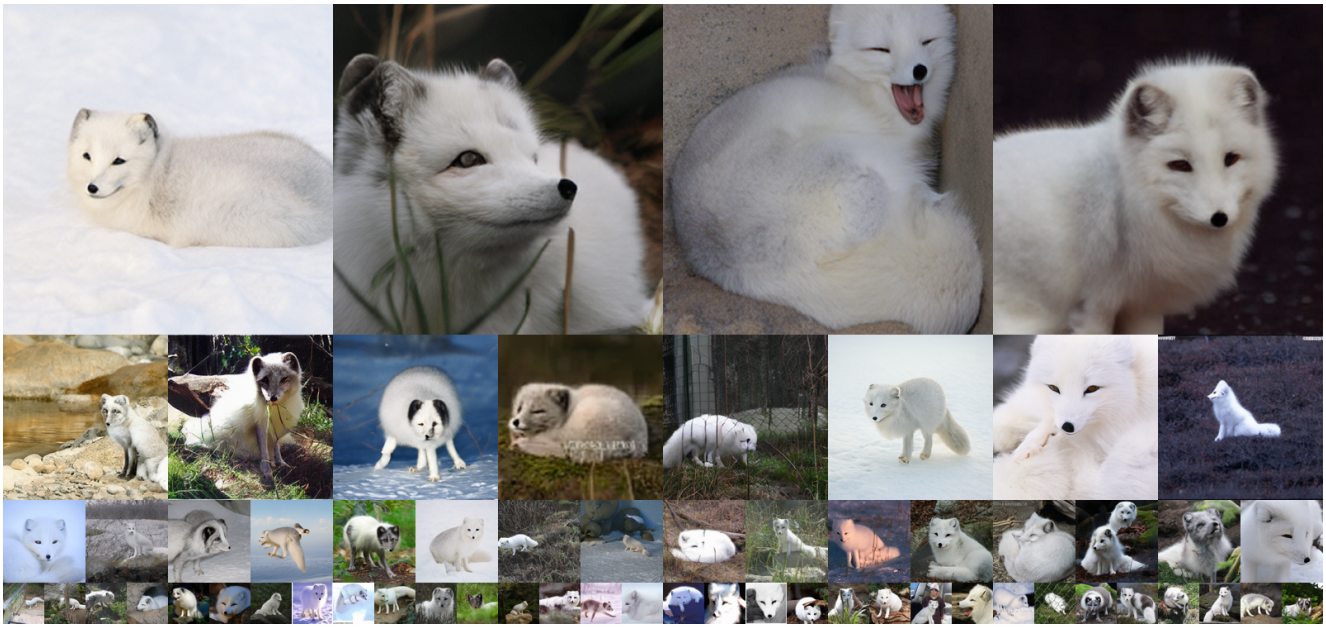


Figure 17. Uncurated 256×256 generation results of SiT-XL with MacTok 128 tokens. We use CFG with 4.0. Class label =“Arctic fox” (279).

Figure 18. Uncurated 256×256 generation results of SiT-XL with MacTok 128 tokens. We use CFG with 4.0. Class label =“otter” (360).

Figure 19. Uncurated 256×256 generation results of SiT-XL with MacTok 128 tokens. We use CFG with 4.0. Class label =“panda” (388).

Figure 20. Uncurated 256×256 generation results of SiT-XL with MacTok 64 tokens. We use CFG with 4.0. Class label =“fire engine” (555).

Figure 21. Uncurated 256×256 generation results of SiT-XL with MacTok 64 tokens. We use CFG with 4.0. Class label =“space shuttle” (812).

Figure 22. Uncurated 256×256 generation results of SiT-XL with MacTok 64 tokens. We use CFG with 4.0. Class label =“ice cream” (928).

Figure 23. Uncurated 256×256 generation results of SiT-XL with MacTok 64 tokens. We use CFG with 4.0. Class label =“cheeseburger” (933).

Figure 24. Uncurated 256×256 generation results of LightningDiT-XL with MacTok 128 tokens. We use CFG with 3.0. Class label ="white shark" (2).

Figure 25. Uncurated 256×256 generation results of LightningDiT-XL with MacTok 128 tokens. We use CFG with 3.0. Class label ="Dungeness crab" (118).

Figure 26. Uncurated 256×256 generation results of LightningDiT-XL with MacTok 128 tokens. We use CFG with 3.0. Class label ="Chesapeake Bay retriever" (209).

Figure 27. Uncurated 256×256 generation results of LightningDiT-XL with MacTok 128 tokens. We use CFG with 3.0. Class label ="burrito" (965).

Figure 28. Uncurated 256×256 generation results of LightningDiT-XL with MacTok 64 tokens. We use CFG with 3.0. Class label =“geyser” (974).

Figure 29. Uncurated 256×256 generation results of LightningDiT-XL with MacTok 64 tokens. We use CFG with 3.0. Class label =“valley” (979).

Figure 30. Uncurated 512×512 generation results of SiT-XL with MacTok 128 tokens. We use CFG with 4.0. Class label =“castle” (483).

Figure 31. Uncurated 512×512 generation results of SiT-XL with MacTok 128 tokens. We use CFG with 4.0. Class label =“cliff” (972).

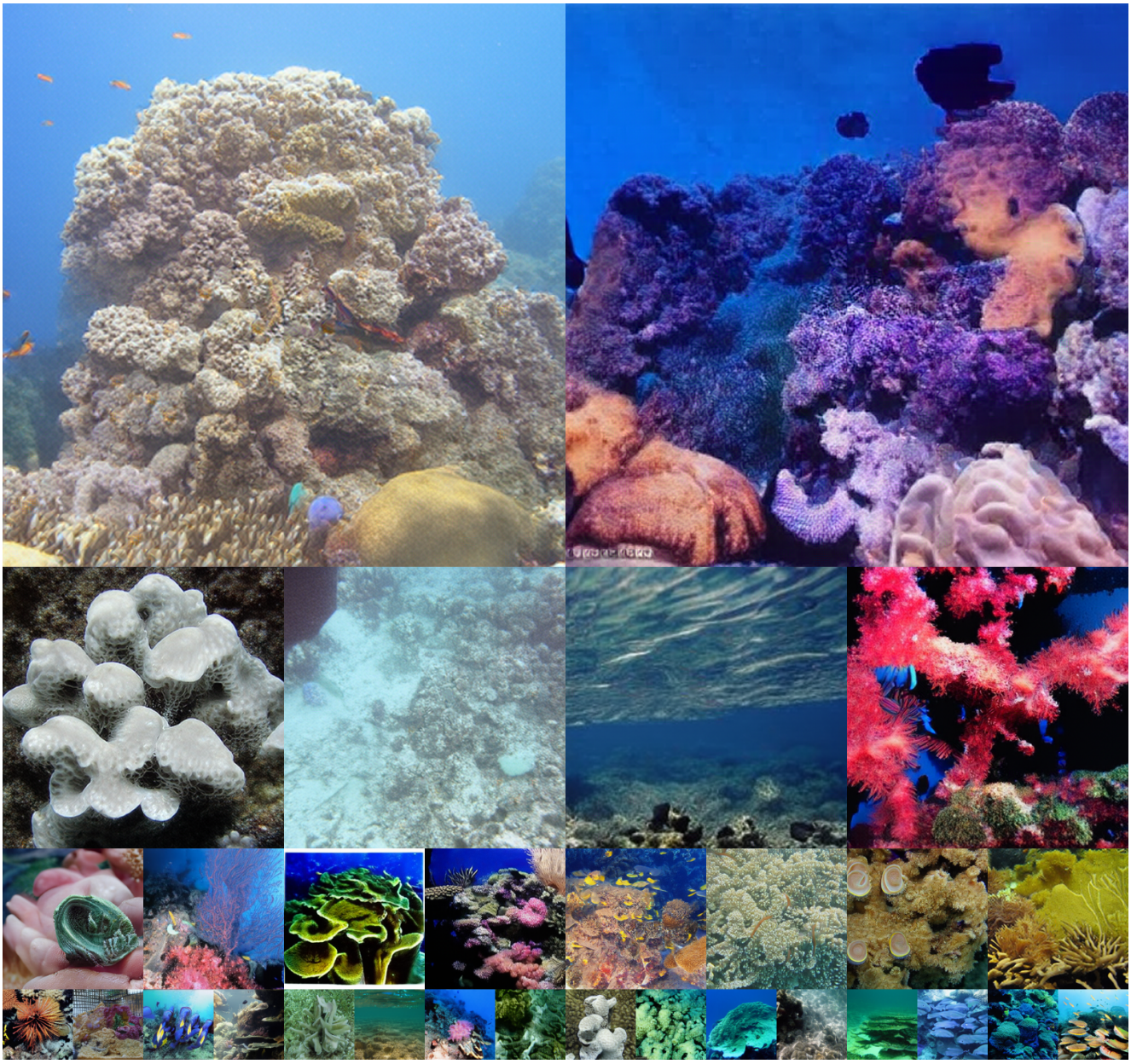


Figure 32. Uncurated 512×512 generation results of SiT-XL with MacTok 64 tokens. We use CFG with 4.0. Class label =“coral reef” (973).

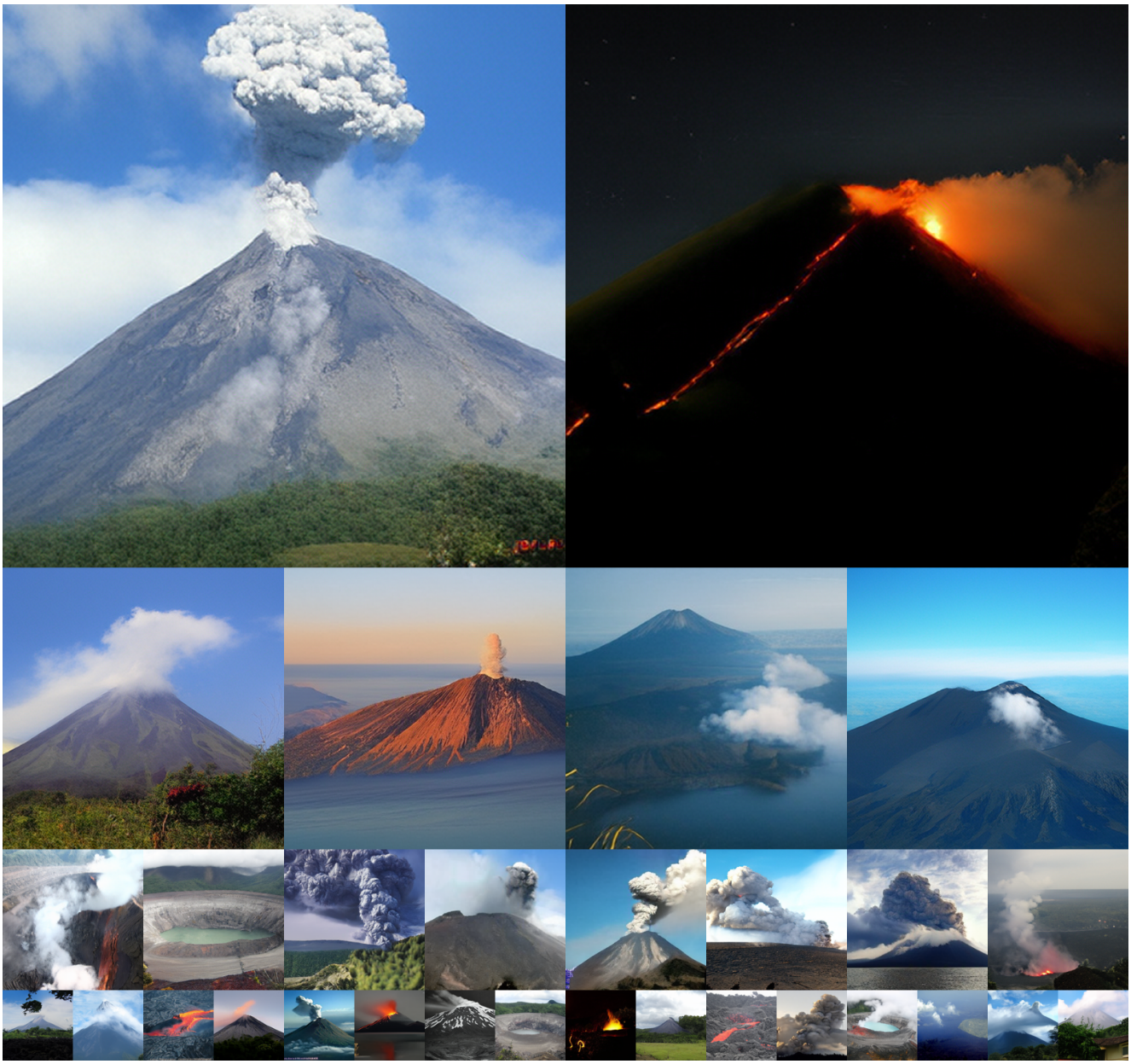


Figure 33. Uncurated 512×512 generation results of SiT-XL with MacTok 64 tokens. We use CFG with 4.0. Class label =“volcano” (980).