Language-Guided Image Tokenization for Generation

Kaiwen Zha[1,2*] Lijun Yu[1] Alireza Fathi[1] David A. Ross[1]
Cordelia Schmid[1] Dina Katabi[2] Xiuye Gu[1]
[1]Google DeepMind [2]MIT CSAIL

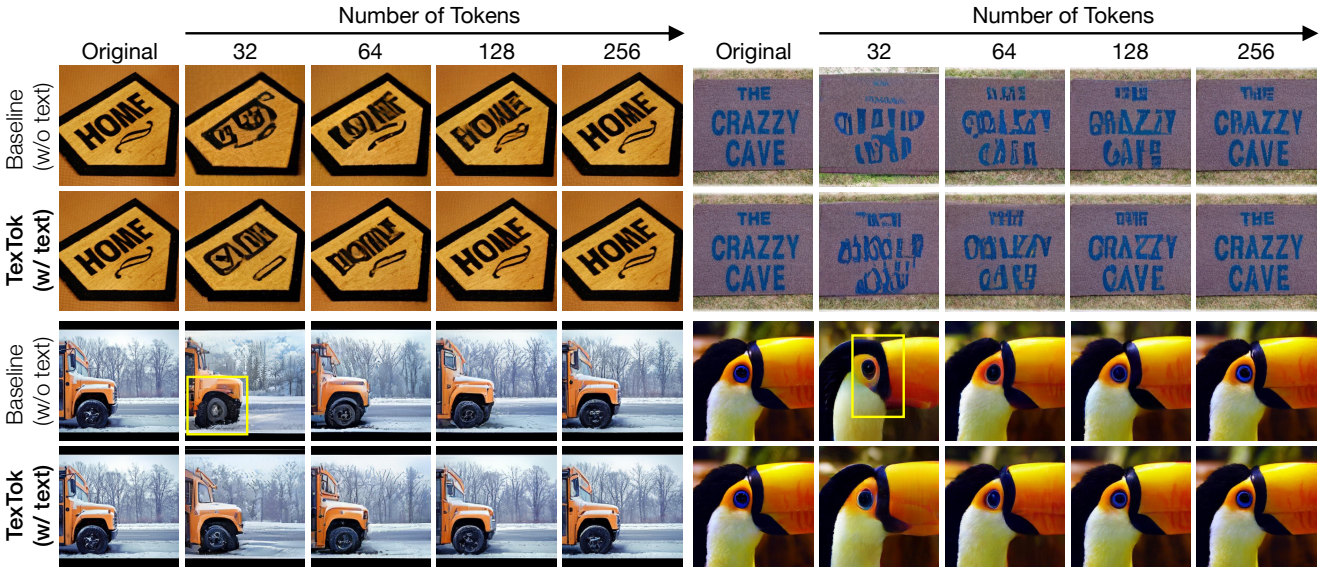


Figure 1. **Reconstruction samples of TexTok compared with Baseline (w/o text)** on ImageNet 256×256 using different number of image tokens. TexTok enables the tokenizer to encode finer visual details into image tokens, achieving better reconstruction quality across various token counts, such as improved text in images, car wheels, and bird beaks. The improvement is particularly significant in the low-token domain. The yellow-boxed regions highlight the significant enhancements.

Abstract

Image tokenization, the process of transforming raw image pixels into a compact low-dimensional latent representation, has proven crucial for scalable and efficient image generation. However, mainstream image tokenization methods generally have limited compression rates, making high-resolution image generation computationally expensive. To address this challenge, we propose to leverage language for efficient image tokenization, and we call our method Text-Conditioned Image Tokenization (TexTok). TexTok is a simple yet effective tokenization framework that leverages language to provide a compact, high-level semantic representation. By conditioning the tokenization process on descriptive text captions, TexTok simplifies semantic learning, allowing more learning capacity and token space to be allocated to capture fine-grained visual details, leading to enhanced reconstruction quality and higher compression rates. Compared to the conventional tokenizer without text conditioning, TexTok achieves average reconstruction FID improvements of 29.2% and 48.1% on ImageNet-256 and -512 benchmarks respectively, across varying numbers of tokens. These tokenization improvements consistently translate to 16.3% and 34.3% average improvements in generation FID. By simply replacing the tokenizer in Diffusion Transformer (DiT) with TexTok, our system can achieve a 93.5× inference speedup while still outperforming the original DiT using only 32 tokens on ImageNet-512. TexTok with a vanilla DiT generator achieves state-of-the-art FID scores of 1.46 and 1.62 on ImageNet-256 and -512 respectively. Furthermore, we demonstrate TexTok's superiority on the text-to-image generation task, effectively utilizing the off-the-shelf text captions in tokenization.

1. Introduction

Image generation has made remarkable progress in recent years, enabling high-quality synthesis across diverse appli-

[*] Work done during an internship at Google DeepMind.

cations [9, 11, 32, 37]. Central to this success is the evolution of image tokenization, a process that compresses raw image data into a compact yet expressive latent representation through training an autoencoder. Tokenization allows generative models, such as diffusion models [9, 32, 37] and autoregressive models [11, 28, 43] to operate directly in this compressed latent space instead of the high-dimensional pixel space, significantly improving computational efficiency while enhancing generation quality and fidelity.

Despite various image tokenization efforts aimed at improving training objectives [11, 36, 45] and refining the autoencoder architecture [47, 50], current methods remain fundamentally limited by a trade-off between compression rate and reconstruction quality, especially in high-resolution generation. High compression reduces computational costs but often sacrifices reconstruction quality, while prioritizing quality leads to prohibitively high computational expenses.

Addressing this limitation requires a fundamental shift in the tokenization process. At its core, tokenization involves finding a compact and effective representation of an image. The most concise and meaningful representation of an image often comes from its language description–i.e., captioning. When describing an image, humans naturally start with high-level semantics before elaborating on finer details. Inspired by this insight, we introduce *Text-Conditioned Image Tokenization (TexTok)*, a novel framework that leverages text captions to guide the tokenizer in learning image semantics. This simplifies semantic learning, allowing more learning capacity and token space to be allocated to capture fine-grained visual details, thereby enhancing reconstruction quality without compromising compression rate.

To the best of our knowledge, we are the **first** to condition on detailed captions in the tokenization stage, an approach typically reserved for the generation phase. Text captions are easy to obtain from online image-text pairs or using a vision-language model to caption the images. Since text conditioning is widely used in image generation, e.g., text-to-image generation, our method can seamlessly incorporate these captions into the tokenization process without incurring additional annotation overhead.

We demonstrate the effectiveness of TexTok across a diverse set of tasks and settings. Compared to conventional tokenizers without text conditioning, TexTok achieves substantial gains in reconstruction quality, with average reconstruction FID improvements of **29.2%** and **48.1%** on ImageNet 256×256 and 512×512 resolutions, respectively. These enhancements in tokenization lead to consistent boosts in generation performance, with average improvements of **16.3%** and **34.3%** in generation FID for the two resolutions. By simply replacing the tokenizer in Diffusion Transformer (DiT) with TexTok, our system achieves a **93.5**× inference speedup while still outperforming the original DiT using only **32** tokens on ImageNet 512×512. Our best TexTok variant with a vanilla DiT generator achieves state-of-the-art FID scores of **1.46** and **1.62** on ImageNet 256×256 and 512×512 respectively.

We further demonstrate that incorporating text during the tokenization stage significantly enhances **text-to-image** generation, achieving 2.82 FID and 29.23 CLIP score on ImageNet 256×256. Since text captions are inherently available for this task, TexTok boosts performance without adding any extra annotation overhead.

2. Related Work

Image tokenization. Image tokenizers build a bidirectional mapping between high-resolution pixels and a low-dimensional latent space, significantly improving the learning efficiency of downstream tasks, such as image generation [4, 11, 26, 49], and understanding [47, 49]. Image tokenizers are usually formulated as an AutoEncoder (AE) [1] framework with an optional quantizer [45] and potentially in a variational [23] setup. These AutoEncoders are trained to minimize the discrepancy between the output and input images, measured by pixel-space distances, latent-space distances [52], or jointly trained discriminators [11]. Architectural variants for the encoder and decoder include ResNet [15] and vision transformers [10]. Spatial correspondence has been a common property of modern tokenizer designs, where one token largely refers to a square neighborhood of pixels. Recently, there has also been development of transformer-based models producing global tokens as a more compact representation [50]. In this work, we follow this paradigm to tokenize an image into a set of global tokens to flexibly control token budgets. However, unlike prior work, we are the first to propose to condition the tokenization process on image captions, which greatly improves the reconstruction quality and compression rate.

Image generation. Generative learning of pixels has been explored under adversarial [3, 40], autoregressive [6], and diffusion [9, 18, 22] setups. For higher resolutions, generative learning in compressed latent spaces has become popular given its efficiency advantages. Among them, autoregressive [11, 26] and masked prediction [4, 49] models often operate in discrete token spaces following the practice of GPT [34] and BERT [8] in language modeling. Recent variants [28] could also use continuous latent spaces, akin to those used in latent diffusion models (LDMs) [37]. For LDMs, the architecture has evolved from convolution-based U-Net [38] to transformer-based DiT [32]. In this paper, we focus on diffusion-based image generation with DiT architecture, leveraging the flexible token lengths of TexTok.

Leveraging external semantic information in image generation and tokenization. Many recent studies start to leverage external semantic information, such as image representations and semantic maps, to improve image genera-

tion [27, 33, 51]. Unlike these methods, which use external semantics to aid the generation process, our approach focuses on enhancing the tokenization process through conditioning on text semantics. Some recent efforts [29, 30, 48, 53] also consider aligning image tokens with text semantics in image tokenization to improve multimodal understanding. They either directly map images to text tokens in a frozen LLM codebook [30, 48, 53] or align the features of image tokens with text features [29], to produce semantically meaningful tokens. However, by enforcing strict image-text alignments, these works suffer from limited image reconstruction quality due to the inherent divergence between vision and language representations, resulting in undesirable image generation quality. In contrast, our work takes a complementary approach. We leverage text as external semantic conditioning, significantly boosting the image reconstruction and generation performance.

3. Method

3.1. Preliminary

Based on the format of latent representation, image tokenizers can be broadly classified into: 1) *Vector-Quantized (VQ) Tokenizers*, such as VQ-VAE [45] and VQGAN [11], which represent images using a set of discrete tokens, and 2) *Continuous Latent Tokenizers* [37] which use a variational autoencoder (VAE) [23] to embed images into a continuous latent space. In this work, we focus primarily on continuous latent tokenizers. As shown in Appendix A, TexTok also works well on VQ tokenizers.

The standard continuous latent tokenizer typically consists of an encoder (tokenizer) E and a decoder (detokenizer) D. Given an image $\mathbf{I} \in \mathbb{R}^{H \times W \times 3}$, the encoder E compresses it into a 2D latent space $\mathbf{Z} = E(\mathbf{I}) \in \mathbb{R}^{h \times w \times d}$, where $h = \frac{H}{f}$, $w = \frac{W}{f}$, and f is the spatial downsampling factor. Each latent embedding $\mathbf{z} \in \mathbb{R}^d$ is treated as a continuous token, with the image represented by a total of hw tokens. For decoding, these embeddings $\mathbf{Z}$ are fed into the decoder D to reconstruct the image $\hat{\mathbf{I}} = D(\mathbf{Z})$. Recently, 1D tokenizers [50] were introduced to allow flexible token budgets for image representation, directly compressing $\mathbf{I}$ into 1D latent embeddings $\mathbf{Z}_{1D} = E(\mathbf{I}) \in \mathbb{R}^{N \times d}$ with N tokens. Reconstruction, perceptual [52], and GAN [11] losses are applied to train the tokenizer by minimizing the distance between $\mathbf{I}$ and $\hat{\mathbf{I}}$.

In this work, we adopt the 1D tokenizer paradigm to allow more flexible compression rates, demonstrating TexTok's efficacy and efficiency across varying token budgets.

3.2. TexTok: Text-Conditioned Image Tokenization

We introduce *Text-Conditioned Image Tokenization (TexTok)*, a simple yet effective tokenization framework. Unlike existing methods that compress all visual information into latent tokens, we use text captions to represent high-level semantics and guide the tokenization process.

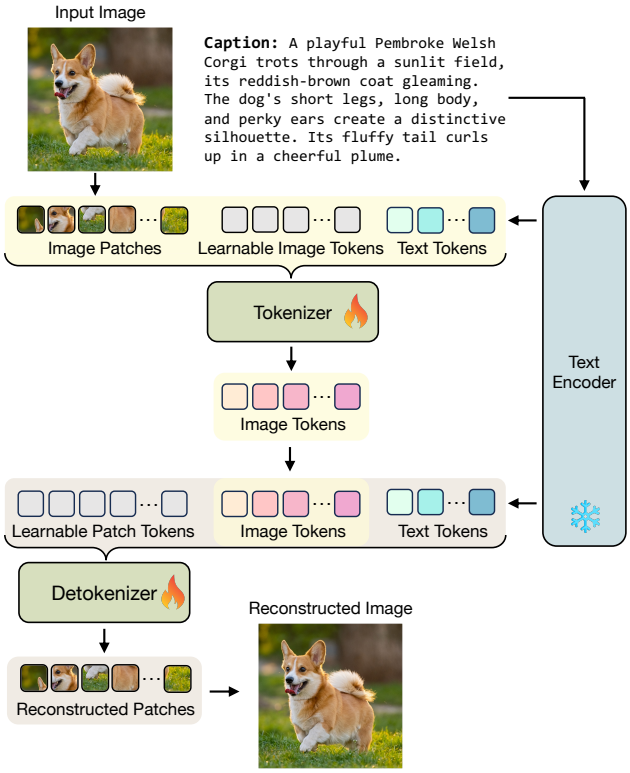


Figure 2. **TexTok architecture**. During training, a frozen text encoder (e.g., T5 [35]) extracts text embeddings (tokens) from the given image caption. The image patches, learnable image tokens, and text tokens are fed into the tokenizer (a ViT [10]) to produce the image tokens. During detokenization, the image tokens are concatenated with the same text tokens fed to the tokenizer and learnable patch tokens to reconstruct the image. For generation, *only image tokens* need to be generated.

Tokenization stage. Given an image caption, we use a frozen T5 [35] text encoder to extract text embeddings. These embeddings are injected into both the tokenizer and detokenizer to providing semantic guidance throughout the tokenization process.

As shown in Figure 2, TexTok adopts a Vision Transformer (ViT) backbone for both the encoder (tokenizer) E and the decoder (detokenizer) D to enable flexible control of token numbers. The input to the tokenizer is a concatenation of three components: 1) image patch tokens $\mathbf{P} \in \mathbb{R}^{hw \times D}$ from patchifying and flattening the input image with a projection layer, where $h = \frac{H}{s}$, $w = \frac{W}{s}$, and s is the patch size, 2) N randomly-initialized learnable image token $\mathbf{L} \in \mathbb{R}^{N \times D}$, where N is the number of output image tokens, and 3) *linearly projected text tokens*, $\mathbf{T} \in \mathbb{R}^{N_t \times D}$, derived from the text embeddings, where N_t is the number of text tokens. In the tokenizer's output, only the learned image tokens are retained and linearly projected to produce

the output image tokens $\mathbf{Z} \in \mathbb{R}^{N \times d}$.

The detokenizer also takes three concatenated inputs: 1) hw learnable patch tokens $\mathbf{P}' \in \mathbb{R}^{hw \times D}$, 2) linearly projected image tokens $\mathbf{Z}' \in \mathbb{R}^{N \times D}$ from the input image tokens, and 3) *linearly projected text tokens* $\mathbf{T}' \in \mathbb{R}^{N_t \times D}$ that come from the same text tokens fed to the tokenizer. In the detokenizer's output, only the learned image patch tokens are retained, unpatchified, and projected to reconstruct the image patches.

We train the tokenizer and detokenizer using the combination of ℓ_2 reconstruction, GAN, perceptual, and LeCAM regularization [44] losses, following [49].

By directly injecting text tokens containing high-level semantic information into both the tokenizer and detokenizer, TexTok alleviates the need for the tokenizer and detokenizer to learn image semantics.

Generation stage. Since this work focuses on continuous latent tokens, we use the Diffusion Transformer (DiT) [32] as the generation framework and train the DiT on top of the latent tokens produced by TexTok. Note that *only latent image tokens* need to be generated in the generation stage, while the text tokens will be provided in detokenization.

DiT is trained to model the distribution of TexTok latent tokens, conditioned either on a class category (for class-conditional generation) or on the text embeddings (for text-to-image generation).

During inference, the process differs by the generation task. For text-to-image generation, we use the provided captions for both generation and detokenization, feeding the text embeddings and generated latent image tokens into the detokenizer to produce the output image. For class-conditional generation, DiT generates latent tokens based on the specified class; we then sample an unseen caption for that class from a pre-generated list, and inject it into the detokenizer along with the generated latent tokens to produce the final image. Notably, only the class category is used during generation, in line with standard practice.

4. Experiments

4.1. Implementation Details

The implementation details of TexTok are described below. Please refer to Appendix C for further details.

Text caption acquision. Text captions are readily available for text-to-image generation tasks, where they can be directly used in the tokenization process. For other generation tasks without captions, such as our use of ImageNet [7], we employ a vision language model (VLM), Gemini v1.5 Flash [42], to generate detailed captions offline. For the training set, we caption each given image. For the evaluation set, in class-conditional generation, we pre-generate unseen captions for each category using a sampled caption list of this category from the training set as reference. By default, each image is captioned with up to 75 words, which are encoded into a 128-token sequence using the T5 text encoder [35] (XL for ImageNet-256 and XXL for ImageNet-512 experiments). Please see Appendix D for more details.

Tokenization & generation. By default, all TexTok experiments employ ViT-Base for both tokenizer and detokenizer, each comprising 12 layers with a hidden size of 768 and 12 attention heads ($\sim$176M parameters). For the GAN loss, we follow [47] and use the StyleGAN discriminator [19]($\sim$24M parameters). Unless otherwise specified, the image token channel dimension in TexTok is set to $d = 8$.

We use Diffusion Transformer (DiT) [32] as our default generator due to its effectiveness and flexibility of handling 1D tokens. We use a DiT patch size of 1 for all TexTok generation experiments, and by default, we train DiT for 350 epochs. Specifically, for class-conditional generation, we use the original DiT architecture. For text-to-image generation, referring to [5], we modify DiT architecture by adding an additional multi-head cross-attention layer following the multi-head self-attention layer in the DiT block to accept text embeddings. We refer to this architecture as "DiT-T2I".

4.2. Experiment Setup

Model variants. We compare two setups to demonstrate the effectiveness of using text conditioning: *TexTok* incorporates text tokens in both the tokenizer and detokenizer, corresponding to the architecture shown in Figure 2. In contrast, *Baseline (w/o text)* does not condition on text tokens in both the tokenizer and detokenizer. For each image, we tokenize it into "#tokens" number of latent tokens and train the generator to generate these tokens.

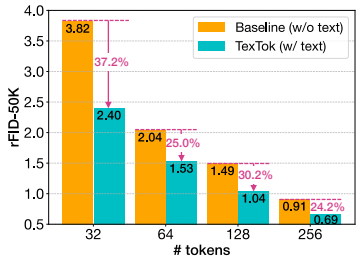
Evaluation protocol. To evaluate reconstruction performance of the tokenizer, we report reconstruction Frechet inception distance (rFID) [17], reconstruction inception score (rIS) [39], peak signal-to-noise ratio (PSNR), structural similarity index measure (SSIM), and learned perceptual image patch similarity (LPIPS) [52] on 50K samples from ImageNet training set. To evaluate class-conditional generation performance, we report generation Frechet inception distance (gFID) [17], generation inception score (gIS) [39], precision and recall [24] following the evaluation protocol and suite provided by ADM [9]. To evaluate text-to-image generation performance, we report FID and CLIP Score [16] on 50K samples from ImageNet validation set.

4.3. Effectiveness of Text Conditioning

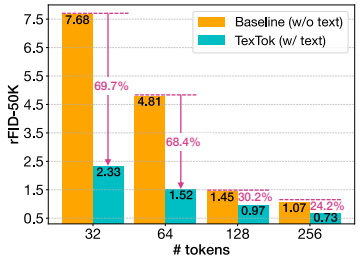
We begin by evaluating the effectiveness of text conditioning in image tokenization and generation. We compare our method, TexTok, with a Baseline (w/o text) that uses the same settings but excludes text conditioning, on ImageNet at resolutions of 256×256 and 512×512. We experiment with varying numbers of tokens, presenting the quantitative

tokenizer	# tokens	Reconstruction rFID ↓	rIS↑	PSNR↑	SSIM ↑	LPIPS↓	Generation gFID ↓	gIS ↑
(a) ImageNet 256×256								
SD-VAE-f8 [37]	1024 (d=4)	1.20†	-	-	-	-	9.62	121.5
Baseline-32 (w/o text)	32 (d=8)	3.82	117.1	17.67	0.4281	0.3270	4.97	170.3
TexTok-32 (w/ text)		**2.40**	**156.2**	**18.32**	**0.4463**	**0.2884**	**3.55**	**205.3**
Baseline-64 (w/o text)	64 (d=8)	2.04	147.2	19.52	0.4801	0.2343	3.30	188.9
TexTok-64 (w/ text)		**1.53**	**169.8**	**20.10**	**0.4971**	**0.2126**	**2.88**	**209.2**
Baseline-128 (w/o text)	128 (d=8)	1.49	160.5	20.51	0.5102	0.1913	3.19	190.1
TexTok-128 (w/ text)		**1.04**	**183.3**	**22.05**	**0.5618**	**0.1499**	**2.75**	**210.9**
Baseline-256 (w/o text)	256 (d=8)	0.91	178.3	23.05	0.5950	0.1225	2.91	197.2
TexTok-256 (w/ text)		**0.69**	**192.6**	**24.38**	**0.6454**	**0.0998**	**2.68**	**219.6**
(b) ImageNet 512×512								
SD-VAE-f8 [37]	4096 (d=4)	-	-	-	-	-	12.03	105.3
Baseline-32 (w/o text)	32 (d=8)	7.68	82.6	16.21	0.5046	0.4771	9.22	119.0
TexTok-32 (w/ text)		**2.33**	**161.5**	**18.55**	**0.5488**	**0.3772**	**3.61**	**215.6**
Baseline-64 (w/o text)	64 (d=8)	4.81	104.0	17.81	0.5341	0.4029	7.26	141.8
TexTok-64 (w/ text)		**1.52**	**171.7**	**20.19**	**0.5786**	**0.3093**	**3.30**	**210.2**
Baseline-128 (w/o text)	128 (d=8)	1.45	163.2	21.59	0.6086	0.2624	3.64	191.1
TexTok-128 (w/ text)		**0.97**	**185.5**	**22.27**	**0.6230**	**0.2365**	**3.16**	**210.7**
Baseline-256 (w/o text)	256 (d=8)	1.07	174.9	23.15	0.6410	0.2180	3.14	204.2
TexTok-256 (w/ text)		**0.73**	**192.0**	**24.45**	**0.6682**	**0.1875**	**2.87**	**218.5**

Table 1. **Image reconstruction and generation performance comparison of TexTok with Baseline (w/o text)** on ImageNet 256×256 and 512×512. TexTok consistently delivers significant improvements in image reconstruction and generation performance, with more pronounced gains as the number of tokens decreases. Class-conditional generation results are reported without classifier-free guidance (Baseline and TexTok use DiT-L as the generator, while SD-VAE uses DiT-XL/2). †: number taken from [28].



(a) ImageNet 256×256



(b) ImageNet 512×512

Figure 3. **Reconstruction FID of TexTok *v.s.* Baseline (w/o text)** on ImageNet 256×256 and 512×512 for different number of image tokens. With text conditioning, TexTok can use **half**, **1/4** of the token number (**2**×, **4**× compression rates) to achieve similar rFID compared to Baseline (w/o text) on ImageNet-256 and -512 respectively.

results in Table 1 and visualizing the relative improvement in rFID in Figure 3.

On ImageNet 256×256, across all settings, TexTok significantly enhances both reconstruction and generation performance. Specifically, TexTok achieves 37.2%, 25.0%, 30.2%, 24.2% improvements in rFID using 32, 64, 128, and 256 tokens respectively, which consistently translates to 28.6%, 12.7%, 13.8%, and 7.9% improvements in gFID. Notably, the fewer tokens used, the higher the gains from text conditioning. As shown in Figure 2a, TexTok can achieve similar rFID using **half** the number of tokens compared to the baseline (**2× compression rate**). We note that our Baseline (w/o text) is highly competitive. As shown in Table 1(a), with 8× fewer number of tokens, Baseline (w/o text) outperforms the widely used SD-VAE tokenizer [37] in both reconstruction and generation.

On higher resolution images, *i.e.*, ImageNet 512×512, TexTok exhibits stronger efficacy. As shown in Table 1 and Figure 2b, TexTok achieves more significant improvement in the reconstruction quality and enables higher compression rates under this high-resolution setting. Specficially, it achieves 69.7%, 68.4%, 30.2%, and 24.2% improvements in rFID and 60.8%, 54.5%, 13.2% and 8.6% improvements in gFID, across 32, 64, 128 and 256 tokens respectively. As shown in Figure 2b, TexTok achieves similar rFID to the baseline using only **1/4 of the token number (4× compression rate)**.

Finally, the qualitative results in Figure 1 across varying token counts show that TexTok significantly enhances reconstruction quality, particularly for text within images and specific visual details, such as car wheels and beaks. This indicates that TexTok encodes finer visual details using the same number of tokens.

4.4. System-level Image Generation Comparison

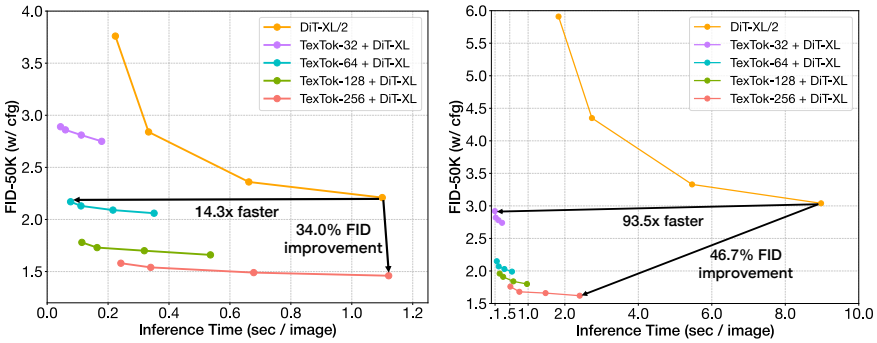
We experiment with image generation using TexTok as the tokenizer and adopt a vanilla DiT image generator [32] (denoted by TexTok + DiT), to study how this system performs against other leading image generation systems. We evaluate on class-conditional ImageNet 256×256 and 512×512 settings with varying number of tokens (compression rates).

On ImageNet 256×256 class-conditional image generation, as shown in Table 2(a), our TexTok-256 + DiT-XL achieves an FID of **1.46**, surpassing previous state-of-the-art systems, even though using a simpler, vanilla DiT as the image generator. As we reduce the number of tokens and increase image compression rate, TexTok + DiT maintains generation performance. Notably, TexTok-64 + DiT-XL, where the diffusion transformer generates only 64 image tokens, outperforms the original DiT-XL/2, which uses 256 tokens after patchification in the diffusion transformer.

On higher resolution images, *i.e.*, ImageNet 512×512, as shown in Table 2(b), TexTok-256 + DiT-XL also achieves state-of-the-art **1.62** gFID compared with previous methods, using only **256** image tokens. On the most compressed side, TexTok-32 + DiT-XL only uses 32 tokens yet achieves better generation performance than the original DiT that uses 1024 tokens after patchification.

			(a) ImageNet 256×256					(b) ImageNet 512×512				
Model	#Params (G)	#Params (T)	FID↓	IS↑	Precision↑	Recall↑	#tokens	FID↓	IS↑	Precision↑	Recall↑	#tokens
GAN												
StyleGAN-XL [40]	168M	-	2.30	265.1	0.78	0.53	-	2.41	267.8	0.77	0.52	-
pixel diffusion												
ADM-U [9]	731M	-	3.94	215.8	0.83	0.53	-	3.85	221.7	0.84	0.53	-
simple diffusion [18]	2B	-	2.44	256.3	-	-	-	3.02	248.7	-	-	-
VDM++ [22]	2B	-	2.12	267.7	-	-	-	2.65	278.1	-	-	-
masked image modeling												
MaskGIT [4]	227M	66M	6.18	182.1	0.80	0.51	256	7.32	156.0	0.78	0.50	1024
RCG [27]	512M	66M	2.25	300.7	-	-	256	-	-	-	-	-
TiTok-L-32 [50]	177M	644M	2.77	-	-	-	32	-	-	-	-	-
TiTok-64 (B/L) [50]	177M	202M / 644M	2.48	-	-	-	64	2.74	-	-	-	64
TiTok-128 (S/B) [50]	287M / 177M	72M / 202M	1.97	-	-	-	128	2.13	-	-	-	128
MAGVIT-v2 [49]	307M	116M	1.78	319.4	-	-	256	1.91	**324.3**	-	-	1024
MaskBit [46]	305M	54M	1.52	**328.6**	-	-	256	-	-	-	-	-
autoregressive												
VQGAN [11]	1.4B	23M	15.78	78.3	-	-	256	-	-	-	-	-
ViT-VQGAN [47]	1.7B	64M	4.17	175.1	-	-	1024	-	-	-	-	-
LlamaGen-3B [41]	3.1B	72M	2.18	263.3	0.81	0.58	576	-	-	-	-	-
VAR (*d30*/*d36*-s) [43]	2B / 2.4B	109M	1.92	323.1	0.82	0.59	256	2.63	303.2	-	-	1024
MAR (H/L) [28]	943M / 481M	66M	1.55	303.7	0.81	0.62	256 (d=16)	1.73	279.9	-	-	1024 (d=16)
latent diffusion												
LDM-4 [37]	400M	55M	3.60	247.7	0.87	0.48	4096 (d=3)	-	-	-	-	-
U-ViT-H [2]	501M	84M	2.29	263.9	0.82	0.57	1024* (d=4)	4.05	263.8	0.84	0.48	4096* (d=4)
DiT-XL/2 [32]	675M	84M	2.27	278.2	0.83	0.57	1024* (d=4)	3.04	240.8	0.84	0.54	4096* (d=4)
DiffiT [14]	-	-	1.73	276.5	0.80	0.62	-	2.67	252.1	0.83	0.55	-
MDTv2-XL/2 [12]	676M	84M	1.58	314.7	0.79	0.65	1024* (d=4)	-	-	-	-	-
REPA + SiT-XL/2 [51]	675M	84M	1.80	284.0	0.81	0.61	1024* (d=4)	-	-	-	-	-
EDM2-XXL [21]	1.5B	84M	-	-	-	-	-	1.81	-	-	-	4096 (d=4)
Ours												
TexTok-32 + DiT-XL	675M	176M	2.75	294.6	0.83	0.56	32 (d=8)	2.74	303.2	0.83	0.56	32 (d=8)
TexTok-64 + DiT-XL	675M	176M	2.06	290.0	0.81	0.60	64 (d=8)	1.99	301.9	0.82	0.6	64 (d=8)
TexTok-128 + DiT-XL	675M	176M	1.66	294.4	0.80	0.61	128 (d=8)	1.80	305.4	0.81	0.63	128 (d=8)
TexTok-256 + DiT-XL	675M	176M	**1.46**	303.1	0.79	0.64	256 (d=8)	**1.62**	313.8	0.80	0.64	256 (d=8)

Table 2. **System-level comparison of class-conditional image generation** on ImageNet 256×256 and 512×512. TexTok-256 + DiT-XL achieves ***state-of-the-art*** performance on both image resolutions. All entries *use* classifier-free guidance if applicable. Note that our method is orthogonal to both latent generation models and classifier-free guidance techniques, more advanced latent generators [31] and guidance mechanisms [20, 25] can also be applied to TexTok to further improve our performance. "#Params (G)": the number of generator's parameters. "#Params (T)": the number of tokenizer's parameters. "#tokens": the number of latent image tokens used during generation. "/" in the first three columns indicates different configurations used at different image resolutions respectively. * denotes the number of tokens before patchification.



(a) ImageNet 256×256 (b) ImageNet 512×512

Figure 4. **Speed/performance tradeoff** of TexTok + DiT-XL compared to the original DiT-XL/2 on ImageNet 256×256 and 512×512. TexTok achieves the same generation performance 14.3×/93.5× faster, or gains 34.0%/46.7% FID improvements using similar inference time. As image resolution scales up, this improvement is more pronounced. Each curve is obtained by using different sampling steps (50, 75, 150, 250). The inference time includes latent token generation, T5 text embedding extraction (for TexTok), and detokenization, measured on a single TPUv5e chip with a batch size of 32.

Tokenizer	FID↓	CLIP Score↑
Baseline-32	5.09	28.08
TexTok-32	**4.36**	**28.73**
Baseline-64	3.74	28.49
TexTok-64	**3.34**	**28.92**
Baseline-128	3.01	28.95
TexTok-128	**2.82**	**29.23**

Table 3. **Text-to-image generation performance comparison of TexTok with Baseline (w/o text)** using DiT-XL-T2I on ImageNet 256×256. TexTok achieves better FID and CLIP scores on all 32/64/128-token settings, indicating that TexTok produces image tokens that improve text-to-image generation results using the exactly *same* image generation setups. Classifier-free guidance is applied.

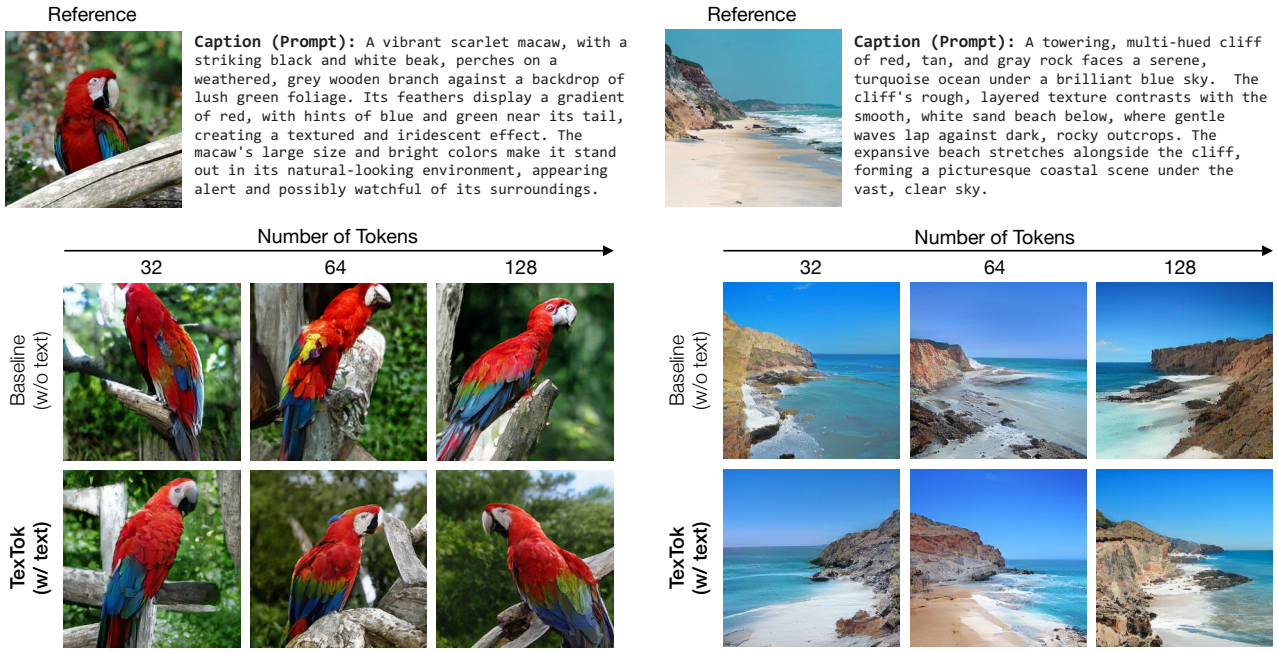


Figure 5. **Qualitative text-to-image generation results of TexTok compared with Baseline (w/o text)** on ImageNet 256×256. TexTok generates higher-quality images that better follow the prompts compared to Baseline (w/o text). It even captures some fine-grained visual details presented in the reference images. The first row shows reference images from the ImageNet validation set along with their captions. Both TexTok and Baseline (w/o text) use the same generation settings and are conditioned on the same captions.

Figure 6. **Qualitative class-conditional image generation results** on ImageNet 512×512. TexTok generates semantically meaningful images with delicate fine-grained details. Results are generated with our class-conditional TexTok-256 + DiT-XL model.

Our system not only achieves superior generation performance, but is also very efficient given its great compression rates. We plot in Figure 4a the speed v.s. performance tradeoffs of TexTok + DiT-XL compared to the original DiT on ImageNet 256×256. Simply replacing the tokenizer in DiT with TexTok can achieve a **14.3× speedup** with better FID, or **34.3% FID improvement** with similar inference time. This verifies the effectiveness and efficiency of TexTok. This improved speed/performance tradeoff is further reflected on ImageNet 512×512 (Figure 4b), where we demonstrate that simply replacing the tokenizer in DiT with TexTok variants, it achieves a **93.5×** speedup with better FID using **32** tokens, or **46.7%** FID improvement with **3.7×** less inference time using 256 tokens. This shows that as image resolution increases, providing the tokenization process with explicit text semantics yields greater improvements in generation performance and inference speedup.

Qualitative samples in Figure 6 demonstrate that TexTok enables class-conditional generation of semantically rich images with fine-grained details. More qualitative samples can be found in Appendix E.

4.5. Text-to-Image Generation

We now demonstrate TexTok's superiority on text-to-image generation. We use the same VLM-generated captions on ImageNet 256×256 with our modified DiT-T2I architecture (detailed in Section 4.1). During training, the tokenizer and generator share the same text embeddings extracted by the T5 text encoder. During inference, we generate images condition on captions from ImageNet validation set. We calculate FID between these generated images and the original ImageNet validation set. As shown in Table 3, compared with Baseline (w/o text), TexTok consistently and significantly improves text-to-image generation, across varying numbers of image tokens. Since text captions are already available for text-to-image tasks and the tokenizer can directly use the same text embeddings used in the generator, TexTok's performance boost comes at no additional cost for captioning and text embedding extraction.

Text conditioning	rFID↓	PSNR↑
none	1.49	20.51
class category	1.14	21.56
class text	1.15	21.58
25-word caption	1.08	21.63
75-word caption	**1.04**	**22.05**

(a) **Text conditioning**. More descriptive text captions yield better results.

T5 model size	rFID↓	PSNR↑
Small	1.06	22.01
XL	1.04	22.05
XXL	**0.99**	**22.28**

(b) **T5 text encoder size**. Larger text encoder model size is better.

Conditioning architecture	rFID↓	PSNR↑
none	1.49	20.51
cross-attention layer	1.31	21.42
in-context conditioning	**1.04**	**22.05**

(c) **Conditioning architecture**. In-context conditioning in the self-attention layers is better than adding a cross-attention layer in each ViT block.

Conditioning location	rFID↓	PSNR↑
none	1.49	20.51
tokenizer only	1.38	21.29
tokenizer & detokenizer	**1.04**	**22.05**

(d) **Conditioning location**. Injecting text conditioning to both tokenizer and detokenizer obtains the best results.

Model size	Layers	Hidden size	Heads	#Params	rFID↓	PSNR↑
TexTok-Small	8+8	512	8	54M	1.35	21.43
TexTok-Base	12+12	768	12	176M	1.04	22.05
TexTok-Large	24+24	1024	16	612M	**1.03**	**22.09**

(e) **TexTok model size**. TexTok-Base has the best performance/efficiency tradeoff.

Table 4. **Ablation studies**. We ablate key design choices affecting TexTok's reconstruction performance on ImageNet 256×256. Default setting: TexTok-Base-128, 75-word captions, T5-XL text encoder, with in-context conditioning applied to both tokenizer and detokenizer.

Qualitative samples in Figure 5 show that TexTok's generation is more realistic and follows the prompts better. More qualitative samples can be found in Appendix E.

4.6. Tokenization/Generation Inference Efficiency

We have demonstrated that TexTok significantly enhances reconstruction, class-conditional generation, and text-to-image generation quality. In text-to-image tasks, our text conditioning incurs no additional cost for text embedding extraction, as text embeddings are also used as conditioning in generation. For other tasks, it introduces minimal computational overhead to generate text embeddings and use them during tokenization. As shown in Table 5, this overhead is negligible (∼0.01 s/img). More importantly, the resulting reduction in generation computational cost compensates for this small increase, as evidenced by the comparison of computational costs between SD-VAE, Baseline (w/o text) and TexTok in Table 5 and the speedup results in Figure 4.

4.7. Ablation Studies

We ablate TexTok to analyze the contribution of our design choices. We use the following default settings: TexTok-128, Base model size, and T5-XL text encoder. Captions are 75 words long and applied to both the tokenizer and detokenizer using in-context conditioning.

Amount of text conditioning. In Table 4a, we ablate various types of class/text conditioning: (1) a learnable class embedding based on the class category, (2) text embeddings from a short text template with class names, (3) text embeddings from 25-word captions, and (4) (ours) text embeddings from 75-word captions. Our results show that more descriptive text conditioning improves performance.

T5 text encoder size. In Table 4b, we study the effect of the text encoder model size. We find that a larger encoder leads to better reconstruction quality. We use T5-XL as the default setting on ImageNet-256 for its efficiency.

Conditioning architecture. Another design choice is how we inject text into the tokenizer and detokenizer. In Table 4c, we find that in-context conditioning (concatenating text embeddings with other input tokens and feeding them into the self-attention layers) outperforms adding an additional multi-head cross-attention layer in each ViT block.

Conditioning location. In Table 4d, we ablate the locations for text conditioning injection and find that applying it to both the tokenizer and detokenizer yields the best results.

TexTok model size. In Table 4e, we investigate the influence of TexTok model size. We find using TexTok-Base performs much better than TexTok-Small, but increasing the model size further provides marginal improvements. Hence, we choose TexTok-Base as our default model size.

	ImageNet 256×256		ImageNet 512×512	
tokenizer	Tokenization	Generation	Tokenization	Generation
SD-VAE-f8 [37]	0.047	0.289	0.182	1.078
Baseline-32	0.051	0.030	0.169	0.031
TexTok-32	0.054	0.031	0.172	0.033
Baseline-64	0.051	0.066	0.171	0.067
TexTok-64	0.054	0.067	0.174	0.072
Baseline-128	0.052	0.110	0.175	0.109
TexTok-128	0.054	0.111	0.178	0.113
Baseline-256	0.052	0.289	0.181	0.282
TexTok-256	0.055	0.292	0.183	0.295

Table 5. **Tokenization & generation inference time**. We evaluate the tokenization and generation inference time of different tokenizers with a DiT-L generator. The tokenzation inference time includes T5 text embedding extraction (for TexTok), tokenization, and detokenization. The generation inference time includes latent token generation, T5 text embedding extraction (for TexTok), and detokenization. Both are measured on a single TPUv6e chip with a batch size of 32 (unit: second per image).

5. Conclusion

We present *Text-Conditioned Image Tokenization (TexTok)*, a new framework that leverages captions to guide the tokenizer in learning image semantics, allowing more learning capacity and token space to be allocated to capture visual details. TexTok significantly improves both reconstruction and generation performance, achieving state-of-the-art results in conditional and text-to-image generation on ImageNet with computational efficiency. By mitigating the trade-off between reconstruction quality and compression rate, TexTok enables more efficient image generation.

Acknowledgements. We thank David Minnen, Eirikur Agustsson, Qihang Yu, Peng Cao, José Lezama, Long Zhao, Haotian Tang, Tianhong Li, Xuhui Jia, Ruben Villegas and Xingyi Zhou for helpful discussions and valuable feedback. KZ and DK are partly funded by Wistron Corporation.

References

[1] Dana H Ballard. Modular learning in neural networks. In *Proceedings of the sixth National conference on Artificial intelligence-Volume 1*, 1987. 2

[2] Fan Bao, Chongxuan Li, Yue Cao, and Jun Zhu. All are worth words: a ViT backbone for score-based diffusion models. In *NeurIPS 2022 Workshop on Score-Based Methods*, 2022. 6

[3] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale gan training for high fidelity natural image synthesis. In *ICLR*, 2019. 2

[4] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. MaskGIT: Masked generative image transformer. In *CVPR*, 2022. 2, 6

[5] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *ICLR*, 2024. 4

[6] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *ICML*, 2020. 2

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 4

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019. 2

[9] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *NeurIPS*, 2021. 2, 4, 6

[10] 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 *ICLR*, 2021. 2, 3

[11] Patrick Esser, Robin Rombach, and Bjorn Ommer. Taming transformers for high-resolution image synthesis. In *CVPR*, 2021. 2, 3, 6

[12] Shanghua Gao, Pan Zhou, Ming-Ming Cheng, and Shuicheng Yan. Masked diffusion transformer is a strong image synthesizer. In *ICCV*, 2023. 6, 1

[13] Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models. In *ECCV*, 2024. 1

[14] Ali Hatamizadeh, Jiaming Song, Guilin Liu, Jan Kautz, and Arash Vahdat. Diffit: Diffusion vision transformers for image generation. In *ECCV*, 2024. 6

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 2

[16] Jack Hessel, Ari Holtzman, Maxwell Forbes, Ronan Le Bras, and Yejin Choi. CLIPScore: A reference-free evaluation metric for image captioning. In *EMNLP*, 2021. 4

[17] 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 *NeurIPS*, 2017. 4

[18] Emiel Hoogeboom, Jonathan Heek, and Tim Salimans. simple diffusion: End-to-end diffusion for high resolution images. In *ICML*, 2023. 2, 6

[19] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *CVPR*, 2019. 4

[20] Tero Karras, Miika Aittala, Tuomas Kynkäänniemi, Jaakko Lehtinen, Timo Aila, and Samuli Laine. Guiding a diffusion model with a bad version of itself. In *NeurIPS*, 2024. 6

[21] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *CVPR*, 2024. 6

[22] Diederik Kingma and Ruiqi Gao. Understanding diffusion objectives as the ELBO with simple data augmentation. In *NeurIPS*, 2024. 2, 6

[23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *ICLR*, 2014. 2, 3

[24] Tuomas Kynkäänniemi, Tero Karras, Samuli Laine, Jaakko Lehtinen, and Timo Aila. Improved precision and recall metric for assessing generative models. In *NeurIPS*, 2019. 4

[25] 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. In *NeurIPS*, 2024. 6

[26] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *CVPR*, 2022. 2

[27] Tianhong Li, Dina Katabi, and Kaiming He. Self-conditioned image generation via generating representations. In *NeurIPS*, 2024. 3, 6

[28] Tianhong Li, Yonglong Tian, He Li, Mingyang Deng, and Kaiming He. Autoregressive image generation without vector quantization. *arXiv:2406.11838*, 2024. 2, 5, 6

[29] Guotao Liang, Baoquan Zhang, Yaowei Wang, Xutao Li, Yunming Ye, Huaibin Wang, Chuyao Luo, Kola Ye, et al. Lg-vq: Language-guided codebook learning. In *NeurIPS*, 2024. 3, 2

[30] Hao Liu, Wilson Yan, and Pieter Abbeel. Language quantized autoencoders: Towards unsupervised text-image alignment. In *NeurIPS*, 2024. 3, 2

[31] 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 *ECCV*, 2024. 6

[32] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *ICCV*, 2023. 2, 4, 5, 6, 1

[33] Pablo Pernias, Dominic Rampas, Mats L Richter, Christopher J Pal, and Marc Aubreville. Würstchen: An efficient architecture for large-scale text-to-image diffusion models. In *ICLR*, 2024. 3

[34] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018. 2

[35] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. In *JMLR*, 2020. 3, 4

[36] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with VQ-VAE-2. In *NeurIPS*, 2019. 2

[37] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 2, 3, 5, 6, 8

[38] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-Net: Convolutional networks for biomedical image segmentation. In *MICCAI*, 2015. 2

[39] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NeurIPS*, 2016. 4

[40] Axel Sauer, Katja Schwarz, and Andreas Geiger. StyleGAN-XL: Scaling StyleGAN to large diverse datasets. In *SIGGRAPH*, 2022. 2, 6

[41] 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:2406.06525*, 2024. 6

[42] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv:2403.05530*, 2024. 4

[43] Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. In *NeurIPS*, 2024. 2, 6

[44] Hung-Yu Tseng, Lu Jiang, Ce Liu, Ming-Hsuan Yang, and Weilong Yang. Regularizing generative adversarial networks under limited data. In *CVPR*, 2021. 4

[45] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. In *NeurIPS*, 2017. 2, 3

[46] Mark Weber, Lijun Yu, Qihang Yu, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. Maskbit: Embedding-free image generation via bit tokens. *arXiv:2409.16211*, 2024. 6

[47] 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. *arXiv:2110.04627*, 2021. 2, 4, 6

[48] Lijun Yu, Yong Cheng, Zhiruo Wang, Vivek Kumar, Wolfgang Macherey, Yanping Huang, David Ross, Irfan Essa, Yonatan Bisk, Ming-Hsuan Yang, et al. SPAE: Semantic pyramid autoencoder for multimodal generation with frozen llms. In *NeurIPS*, 2024. 3, 2

[49] 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 *ICLR*, 2024. 2, 4, 6, 1

[50] 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 *NeurIPS*, 2024. 2, 3, 6

[51] 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:2410.06940*, 2024. 3, 6

[52] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, 2018. 2, 3, 4

[53] Lei Zhu, Fangyun Wei, and Yanye Lu. Beyond text: Frozen large language models in visual signal comprehension. In *CVPR*, 2024. 3, 2