

UniTok: A Unified Tokenizer for Visual Generation and Understanding

Chuofan Ma^{1,2} Yi Jiang^{2†} Junfeng Wu^{2,3} Jihan Yang¹
Xin Yu¹ Zehuan Yuan^{2*} Bingyue Peng² Xiaojuan Qi^{1†*}

¹The University of Hong Kong ²ByteDance Inc.

³Huazhong University of Science and Technology

Abstract

Visual generative and understanding models typically rely on distinct tokenizers to process images, presenting a key challenge for unifying them within a single framework. Recent studies attempt to address this by connecting the training of VQVAE (for autoregressive generation) and CLIP (for understanding) to build a unified tokenizer. However, directly combining these training objectives has been observed to cause severe loss conflicts. In this paper, we show that reconstruction and semantic supervision do not inherently conflict. Instead, the underlying bottleneck stems from limited representational capacity of discrete token space. Building on these insights, we introduce UniTok, a unified tokenizer featuring a novel multi-codebook quantization mechanism that effectively scales up the vocabulary size and bottleneck dimension. In terms of final performance, UniTok sets a new record of 0.38 rFID and 78.6% zero-shot accuracy on ImageNet. Besides, UniTok can be seamlessly integrated into MLLMs to unlock native visual generation capability, without compromising the understanding performance. Additionally, we show that UniTok favors *cfg-free generation*, reducing gFID from 14.6 to 2.5 on ImageNet 256×256 benchmark. GitHub: <https://github.com/FoundationVision/UniTok>.

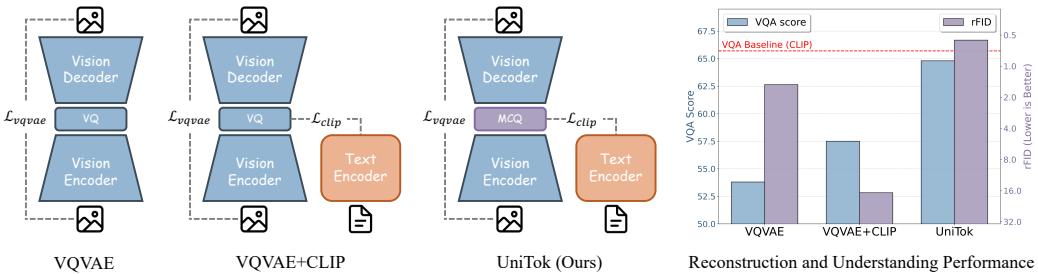


Figure 1: **The major challenge in unified tokenizer training.** CLIP supervision cannot be easily incorporated into VQVAE training – This provides only marginal improvements in understanding performance, while drastically degrading reconstruction FID.

1 Introduction

The advent of GPT-4o [33] highlights the immense potential of Multimodal Large Language Models (MLLMs) with native visual generation capabilities [12, 46, 62, 73, 58]. These unified models offer precise control in multimodal interactions, enabling exceptional fluency in tasks such as multi-turn image editing and visual in-context learning. However, a fundamental dilemma remains in the choice of visual tokenizers for unified MLLMs – e.g., the CLIP [38, 71] tokenizer excels in multimodal understanding but complicates generative modeling due to its high-dimensional, continuous feature

*: Corresponding authors; †: Project lead.

space; Conversely, the discrete VQVAE [8] tokenizer fits autoregressive generation but struggles to capture essential semantics for understanding [62].

In this work, we aim to design a unified visual tokenizer to bridge the gap in multimodal generation and understanding. Intuitively, this can be achieved by integrating CLIP supervision into VQVAE training, resulting in a discrete tokenizer capturing both fine-grained details and high-level semantics. However, we empirically find this training recipe confronts severe convergence issues [61] and largely falls behind the CLIP baseline in multimodal understanding (Figure 1). While prior studies commonly attribute these challenges to conflicts between semantic and pixel-level feature learning [61, 37, 58], recent progress in visual generation suggests the opposite, showing semantic regularization could benefit tokenizers in reconstruction-oriented training [63, 4, 19]. Such disparity motivates us to question: **Do reconstruction and semantic losses truly conflict in tokenizer training?**

To study the problem, we conduct a comprehensive ablation on the unified tokenizer training paradigm (Figure 3), which yields several intriguing findings: First, we show that removing reconstruction supervision, which leads to a vector-quantized CLIP model, does not improve understanding performance compared to the unified tokenizer. This observation indicates that the performance gap between unified and CLIP tokenizers mainly arises from vector quantization, rather than conflicts between learning objectives; Further analysis reveals that this gap is driven by two key factors: token factorization, which projects tokens into a lower-dimensional space for code index lookup [65], and discretization. These operations are essential for vector quantization but inevitably compromise the expressiveness of visual tokens. We thus argue that **the primary bottleneck of unified tokenizers lies in the limited representational capacity of discrete token space.**

In light of the issue, we consider expanding the vocabulary size and latent code dimension, which allows for a closer approximation of the continuous feature space. However, extensive studies have shown that doing so could result in low codebook utilization [74, 65] and diminishing performance gains [67]. To address this, we introduce multi-codebook quantization to partition the visual token into several chunks, each discretized using a small, separate sub-codebook, akin to the multi-head attention mechanism [54]. This design exponentially scales the vocabulary size with the number of sub-codebooks, while avoiding the optimization problems of large monolithic codebooks. Besides, we replace traditional linear projection layers with adapted attention modules for token factorization, which is observed to consistently improve training stability and understanding performance.

Building upon these techniques, we train a unified tokenizer called UniTok to bridge visual generation and understanding. Through extensive experiments, we demonstrate that UniTok achieves comparable or even better performance to domain-specific tokenizers: On ImageNet evaluation, UniTok records an impressive 0.38 reconstruction FID and 78.6% zero-shot accuracy at 256×256 resolution; In building unified MLLMs, UniTok enables the MLLM with native visual generation capabilities while maintaining decent understanding performance. It outperforms the Liquid [59] baseline with a VQGAN tokenizer by 5.5% on VQAv2 [13], 9.2% on TextVQA [42], and 339 points on MME [64]; In addition, we demonstrate that semantic supervision leads to improved latent space structure for autoregressive generation, i.e., for class-conditional image generation on ImageNet 256×256 , UniTok significantly reduces generation FID *without classifier-free guidance* from 14.6 to 2.5 under the LlamaGen [43] framework, which aligns with recent findings in diffusion modeling [19, 4, 63].

2 Related Work

Image Tokenization for Generation. In the domain of visual generation, image tokenization plays an important role in encoding raw pixels into compact latent features for generative modeling [53, 39]. Among a variety of tokenizers, the vector-quantized tokenizer [53] is favored for its discrete latent space and compatibility with autoregressive or masked generative models [48, 43, 3, 66]. The pioneering work VQVAE [53] initially introduced the concept of discretizing continuous tokens by mapping them to the nearest neighbors in a learnable codebook. Built on this, VQGAN [8] added perceptual loss [72] and discriminator loss [16] to improve the reconstruction quality. ViT-VQGAN [65] subsequently advanced the framework with the transformer architecture. In recent literature, considerable efforts have been devoted to developing better quantization methods such as residual quantization [18] and lookup-free quantization [67], which also constitute a focal point of this paper.

Image Tokenization for Understanding. The unprecedented success of large language models (LLMs) [57, 1, 51, 47] has catalyzed the development of multimodal large language models (MLLMs)

[28, 25, 31]. As a critical component of MLLMs, the selection of an effective vision tokenizer has been the subject of extensive study [55, 49]. A common choice of the vision tokenizer is the pretrained CLIP model [38], which undergoes alignment with language during its pretraining phase. While self-supervised learning models, such as DINOv2 [34], are shown to be advantageous at region-level tasks [30]. However, these tokenizers predominantly encode images into a continuous feature space, presenting challenges for uniformly modeling both vision and text tokens. To address this, some works have explored discretizing CLIP tokens [10] or employing VQVAE encoders [27, 62]. Yet, these methods have been observed to substantially impair understanding performance of MLLMs.

Unified Vision-Language Models. The rise of MLLMs is not limited to the realm of visual understanding. Recent advancements have witnessed an increasing focus on unifying visual generation and understanding within one MLLM [7, 60, 46, 73, 62, 50, 21]. Specifically, a line of works employs continuous visual tokenizers for image encoding, and leverages pretrained diffusion models for image synthesis [7, 11, 44]. This approach inevitably increases model complexity and disconnects the visual sampling process from the MLLM. In contrast, another stream of research adopts VQVAE models to encode images into discrete tokens [46, 56, 62, 61, 59]. These tokens are subsequently modeled using the same cross-entropy loss that is applied to text tokens, facilitating a unified approach to multimodal learning. However, as reconstruction-oriented VQVAE does not naturally align with the LLM token space, these models typically suffer from degraded visual comprehension capabilities. Our research aligns with the second approach, with a particular focus on the tokenizer design that is suitable for both generation and understanding tasks.

3 Method

In this section, we introduce UniTok, a unified tokenizer well-suited for both visual generation and understanding tasks. We start with a unified training recipe that integrates reconstruction (VQVAE) and semantic (CLIP) supervisions (Section 3.1). However, we find that simply combining both training objectives leads to severe performance degradation, which can be mainly attributed to limited representational capacity of discrete tokens (Section 3.2). To this end, we propose multi-codebook quantization and attention projection to enhance the latent feature space and derive unified visual representations (Section 3.3). An overview of the framework is presented in Figure 2.

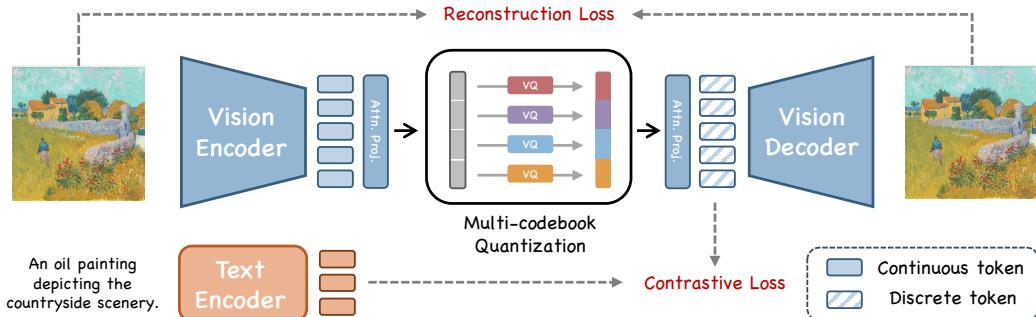


Figure 2: **An overview of UniTok.** The tokenizer is trained to reconstruct the input image while aligning its discrete latent features with the text caption. For vector quantization, each visual token is split into multiple chunks, which then undergo code index lookup on corresponding sub-codebooks.

3.1 Unified Supervision

Visual generative and understanding models typically impose distinct demands on the visual tokenizers. For instance, generation emphasizes precise encoding of the visual signals, whereas understanding prioritizes capturing high-level semantics. To accommodate both requirements, we jointly train the tokenizer with (i) a VQVAE-based reconstruction loss to preserve low-level information, and (ii) an image-text contrastive loss that enhances high-level semantics of the features.

To be specific, the VQVAE-based loss term $\mathcal{L}_{\text{recon}}$ consists of a pixel-level reconstruction loss \mathcal{L}_R , a perceptual loss \mathcal{L}_P based on the LPIPS metric [72], a discriminator loss \mathcal{L}_G to enhance reconstruction

fidelity [16], and a vector quantization loss \mathcal{L}_{VQ} to minimize distance between the encoder output and its nearest code entry. It is denoted as:

$$\mathcal{L}_{\text{recon}} = \mathcal{L}_{\text{R}} + \lambda_{\text{VQ}} \mathcal{L}_{\text{VQ}} + \lambda_{\text{P}} \mathcal{L}_{\text{P}} + \lambda_{\text{G}} \mathcal{L}_{\text{G}}, \quad (1)$$

where λ is the weight factor for the corresponding loss term. The image-text contrastive loss term $\mathcal{L}_{\text{contra}}$ is basically the same as in CLIP [38]. Therefore, the final loss term can be written as:

$$\mathcal{L} = \mathcal{L}_{\text{recon}} + \lambda_{\text{contra}} \mathcal{L}_{\text{contra}}. \quad (2)$$

We simply choose $\lambda_{\text{contra}} = 1$ in this paper.

3.2 Quantization Bottleneck

Despite being augmented with CLIP supervision, we find that the unified tokenizer exhibits unsatisfactory performance in visual understanding tasks, significantly lagging behind the commonly used CLIP tokenizer. To figure out the underlying cause of this underperformance, we break down the key components involved in training a unified tokenizer, as illustrated in Figure 3. Starting with the CLIP baseline, we provide a step-by-step walk-through of all changes in following paragraphs.

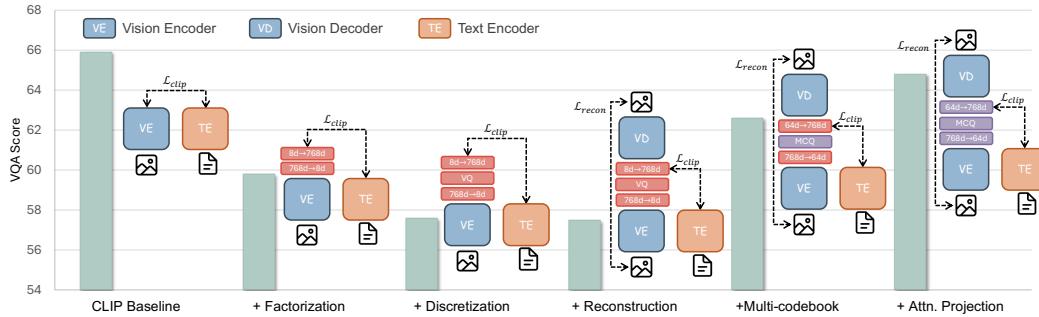


Figure 3: **Roadmap from CLIP to UniTok.** It is observed that major degradation in understanding performance comes from token factorization and discretization, rather than reconstruction supervision. The proposed multi-codebook quantization and attention projection effectively address this by scaling up the vocabulary size and bottleneck dimension. The VQA score is measured using the average score across the VQAv2, GQA, TextVQA, and POPE benchmarks. All tokenizers are trained from scratch on 512m image-text pairs from DataComp.

Factorization. Modern VQ-tokenizers typically project continuous tokens to a lower-dimensional latent space for code index lookup (e.g. from 768-d to 8-d), known as token factorization [65]. This increases the relative density of codes by compressing the latent code space, thereby reducing quantization error. To evaluate the impact of factorization in CLIP training, we add two linear projection layers on top of the CLIP vision encoder (right before average pooling), which transforms tokens from 768-d to 16-d and then back to 768-d. Notably, vector quantization and reconstruction supervision are not included at this stage. Surprisingly, it turns out that this channel compression operation significantly compromises the expressiveness of tokens, leading to severe performance degradation in downstream VQA tasks.

Discretization. Based on the implementation described above, we further introduce vector quantization to CLIP training, which maps factorized tokens to their nearest code entries. Compared to language tokenizers with vocabularies exceeding 200k entries, the vocabulary size of modern VQ-tokenizers is markedly smaller (i.e., typically ranging from 4k to 16k). Mapping continuous tokens to such a small codebook results in considerable information loss. This is validated in our experiment, which demonstrates that discretizing the factorized tokens with a 16k codebook causes an average accuracy drop of 2.1 in VQA tasks.

Reconstruction Supervision. Finally, we integrate reconstruction losses into the training process to build a unified tokenizer, as outlined in Section 3.1. Previous literature suggests that loss conflict between VQVAE and CLIP is a major cause of performance degradation in joint training [61]. We observe a similar phenomenon where joint training results in sub-optimal ImageNet zero-shot classification accuracy and reconstruction FID compared to specialized training. However, surprisingly, we find that this degradation has negligible impacts on downstream understanding performance.

Moreover, the degradation in classification accuracy and reconstruction FID diminishes after we improve the quantization methods (detailed in the next section). Based on these observations, we speculate that the perceived loss conflict is only a superficial issue, and the primary cause of the underperformance lies in the limited representational capacity of discrete tokens.

3.3 UniTok

A straightforward solution to breaking the quantization bottleneck could be increasing the codebook size and the latent code dimension. However, current studies on VQVAE tokenizers suggest that there is diminishing gain in scaling and the performance saturates after the codebook size reaches 16k [67, 43]. Continuing expansion results in a substantial portion of codes being rarely used or becoming ‘dead’ during training, which negatively impacts downstream task performance [65]. To address this, we propose multi-codebook quantization and attention projection in the following paragraphs.

Multi-codebook quantization (MCQ) discretizes the latent tokens with a set of independent codebooks. Specifically, the latent vector $f \in \mathbb{R}^d$ is first evenly split into n chunks $\{f_1, f_2, \dots, f_n\}$, where $f_i \in \mathbb{R}^{\frac{d}{n}}$. The subsequent quantization process is denoted as:

$$\hat{f} = \text{Concat}(\mathcal{Q}(Z_1, f_1), \mathcal{Q}(Z_2, f_2), \dots, \mathcal{Q}(Z_n, f_n)) \quad (3)$$

where \hat{f} is the discretized latent vector, \mathcal{Q} is the code index lookup operation, and Z_i is i -th sub-codebook. Compared to conventional quantization methods, the proposed MCQ effectively scales up the vocabulary size. For instance, by increasing the number of sub-codebooks from 1 to 4, and suppose each sub-codebook contains 16k code entries, the theoretical vocabulary size exponentially increases from 2^{14} to 2^{56} (i.e., there are up to $2^{14 \times 4}$ possible combinations of codes for each token). As the size of each individual codebook remains constant, it circumvents the optimization problem associated with large codebooks. Besides, the dimensionality of the latent codes also scales proportionally with the number of codebooks (i.e., increasing from 16-d to 64-d in this case), which further enhances the representational capacity of discrete representations.

Discussions. MCQ shares a similar concept with residual quantization (RQ) [18] in using multiple codes to quantize a token, but differs fundamentally in design philosophy: RQ follows a coarse-to-fine quantization order, whereas MCQ adopts a divide-and-conquer strategy. This distinction gives MCQ unique advantages when operating in high-dimensional latent spaces, where codes tend to become increasingly sparse. For instance, with a latent dimension of 64-d, we observe that MCQ’s quantization loss is 15 to 45 times lower than that of RQ. This is because MCQ partitions the original latent space into multiple low-dimensional subspaces for quantization. Our ablation study in Table 7 further confirms the superiority of MCQ in unified tokenizer training.

Attention projection. Existing VQ methods usually employ linear or convolutional projection layers for token factorization. But as shown in Figure 3, this over-simplified design fails to preserve rich semantics when compressing the feature dimensions, leading to degraded understanding performance. To alleviate this problem, we suggest adapting the multi-head attention modules for factorization. Specifically, instead of concatenating features from multiple heads after the attention calculation, we replace the concatenation operation with average pooling to realize channel compression. Figure 6 provides a detailed illustration of the adaptation. Despite its simplicity, we find this design effectively strengthens the representational power of factorized tokens and stabilizes training.

3.4 Unified MLLM

We proceed to develop a unified multimodal model with UniTok. Particularly, we leverage the unified framework introduced in Liquid [59], which models (discrete-valued) vision and language sequences with a universal next-token prediction loss. But instead of learning the visual codebook from scratch, we reuse code embeddings of UniTok by projecting them to the MLLM token space with an MLP projector. Notably, despite UniTok encodes an image into $H \times W \times K$ codes (where K represents the number of sub-codebooks), we simplify this for MLLM input by merging every K consecutive codes into a single visual token. Similarly, when it comes to visual token prediction, we make each token autoregressively predict the next K codes, using a depth transformer head as implemented in RQ-Transformer [18] and VILA-U [61]. This design maintains efficiency for visual generation in the context of multi-codebooks.

4 Experiments

4.1 Implementation Details

Tokenizer Setup. Leading VQVAE tokenizers predominantly adopt the CNN architecture, while ViT is preferred in CLIP training for its scalability. To take advantage of both, we choose a hybrid architecture, ViTamin-L/16 [5], to instantiate UniTok. We configure UniTok with eight sub-codebooks, each containing 4,096 code entries and a latent dimension set to 8-d (the global latent dimension is thus 64-d). The discriminator is initialized with pretrained DINoV2-S [34]. We train the tokenizer for one epoch on the public dataset DataComp-1B [9] consisting of 1.28B image-text pairs, with all images resized to 256×256 resolution and a global batch size of 16k. The learning rate is set to 1e-3 for the tokenizer and 2e-4 for the discriminator. Besides, we prepare two settings for evaluation: one with pretrained CLIP weight initialization and one with random initialization (the default setting).

MLLM Setup. We instantiate a unified MLLM described in Section 3.4 with the Llama-2-7B base model [52]. Following Liquid, we first pretrain the model on a mix of multimodal data, which is composed of 10M language data from DCLM [22], 30M internal MidJourney-style synthetic data, and 30M re-captioned image-text pairs from COYO [32] and Laion [41]. Subsequently, we finetune the model on 1.5M text-to-image data and 1.5M multimodal instruction tuning data introduced in Mini-Gemini [23]. Specifically, the learning rate is set to 5e-5 in the pretraining stage and 2e-5 in the finetuning stage. For visual understanding evaluation, we report results on standard VQA benchmarks including VQAv2 [13], GQA [14], TextVQA [42], POPE [24], MME [64], and MM-Vet [70]. For visual generation evaluation, we report results on GenAI-Bench [26] and MJHQ-30K [20].

4.2 Tokenizer Comparison

We benchmark UniTok on ImageNet using two primary metrics: Fréchet Inception Distance (FID) to evaluate reconstruction quality, and top-1 zero-shot accuracy to assess image-text alignment. The results are presented in Table 1. To provide a fair comparison with tokenizers trained on small datasets, we also train a version of UniTok on OpenImages [17] solely with reconstruction supervision. It can be seen that UniTok excels in reconstruction quality compared to both unified and domain-specific tokenizers, recording an impressive 0.38 rFID on ImageNet with $16 \times$ downsampling ratio. As a discrete tokenizer, UniTok even surpasses the continuous VAE tokenizer from Stable Diffusion v2.1 [40], showcasing the superiority of the proposed multi-codebook quantization. For the perception performance, we observe that randomly initialized UniTok demonstrates suboptimal zero-shot classification accuracy. This is expected as current training schedule (i.e., one epoch on 1.28B samples) is insufficient for CLIP training to fully converge. It can be seen that initializing the model with pretrained CLIP weights largely alleviates the problem, boosting the zero-shot accuracy from 70.8% to 78.6%. In complement to quantitative results, we provide examples of reconstructed images in Figure 4.

4.3 Class-Conditional Image Generation

Recent studies on diffusion models indicate that injecting semantics into VAE training leads to a better-structured latent space, considerably enhancing guidance-free generation performance [63, 4, 19]. To evaluate whether UniTok possesses similar properties, we test it within the LlamaGen framework for class-conditional image generation. As shown in Table 2, UniTok reduces the FID by 12.11 compared to the VQGAN baseline in CFG-free generation, under the same generator setup. This implies that UniTok learns a more structured code distribution, benefiting autoregressive modeling.

Table 1: Comparison on ImageNet reconstruction FID and zero-shot classification accuracy. rFID is measured at 256×256 resolution with $16 \times$ downsample ratio. \dagger indicates model using pretrained CLIP weights for initialization. * indicates model trained on OpenImages.

Method	#Tokens	rFID \downarrow	Accuracy
<i>VQVAE Model</i>			
VQ-GAN* [8]	256	4.98	–
RQ-VAE [18]	256	1.30	–
VAR* [48]	680	0.90	–
UniTok*	256	0.33	–
<i>CLIP Model</i>			
CLIP [38]	256	–	76.2
SigLIP [71]	256	–	80.5
ViTamin [5]	256	–	81.2
<i>Unified Model</i>			
TokenFlow † [37]	680	1.37	–
VILA-U † [61]	256	1.80	73.3
UniTok	256	0.41	70.8
UniTok †	256	0.38	78.6

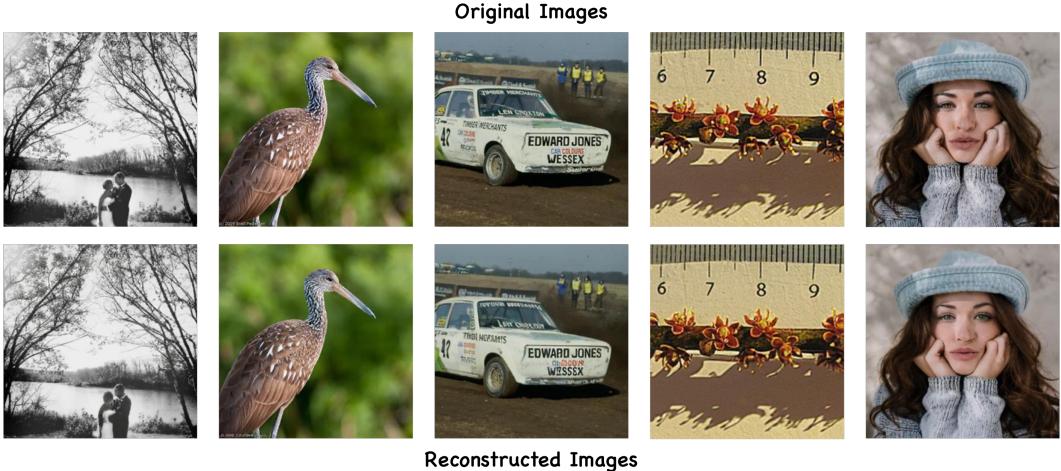


Figure 4: Qualitative results on image reconstruction in a resolution of 256×256 .

Table 2: Class-conditional image generation results on ImageNet 256×256 . \dagger : VQGAN tokenizer from LlamaGen. \ddagger : Images are generated at 384×384 resolution and then resized to 256×256 for evaluation. ‘Pre.’: precision; ‘Rec.’: recall; ‘CFG’: classifier-free-guidance.

Tokenizer	rFID	Generator	#Params.	Generation w/o CFG				Generation w/ CFG			
				gFID \downarrow	IS \uparrow	Pre.	Rec.	gFID \downarrow	IS \uparrow	Pre.	Rec.
<i>Diffusion Models</i>											
SD-VAE [40]	0.61	DiT [35]	675M	9.62	121.5	0.67	0.67	2.27	278.2	0.83	0.57
VAVAE [63]	0.28	LightningDiT [63]	675M	2.17	205.6	0.77	0.65	1.35	295.3	0.79	0.65
<i>Masked Generative Models</i>											
LFQ [67]	0.9	MAGVIT-v2 [67]	307M	3.07	213.1	–	–	1.91	324.3	–	–
TiTok-L [69]	2.21	MaskGIT [3]	177M	3.15	173.0	–	–	2.77	199.8	–	–
<i>Autoregressive Models</i>											
VQGAN †	2.19	LlamaGen ‡ [43]	1.4B	14.65	86.3	0.63	0.68	2.34	253.9	0.81	0.60
UniTok (Ours)	0.41	LlamaGen [43]	1.4B	2.51	216.7	0.82	0.57	2.77	227.5	0.81	0.57

4.4 Unified Understanding and Generation

Understanding Performance. We evaluate the understanding performance of UniTok on diverse VQA benchmarks in Table 3. Our unified MLLM showcases clear advantages when compared to other unified models that also utilize a discrete visual tokenizer. Specifically, UniTok significantly outperforms the Chameleon model, which relies on a traditional VQVAE tokenizer, by 7.2% higher accuracy on VQAv2. Additionally, it surpasses VILA-U, another model with a unified tokenizer, by 3.3% in accuracy on the TextVQA benchmark and by a notable margin of 112 points on the MME-Perception scores. Furthermore, we can see that UniTok largely narrows the performance gap with MLLMs that incorporate continuous visual tokenizers. These strong results confirm the candidacy of UniTok as a unified visual tokenizer for multimodal models.

Generation Performance. Table 4 presents the text-to-image generation performance of our unified MLLM on the GenEval benchmark. We show that UniTok not only outperforms most of the unified MLLMs, but also demonstrates competitive performance against domain experts (diffusion models) trained on billions of images. Besides, UniTok achieves non-trivial improvements over Liquid while using exactly the same set of text-to-image training data, highlighting the importance of a unified tokenizer. We also provide results on GenAI-Bench in Table 10 and Table 11 in Appendix.

We further evaluate the quality of images generated by our model on the MJHQ-30K benchmark, details of which are presented in Table 5. Notably, as this benchmark primarily relies on the FID score

Table 3: Comparison with unified multi-modal large language models on VQA benchmarks.

Method	LLM	Token Type	Res.	VQAv2	GQA	TextVQA	POPE	MME	MM-Vet
Emu [45]	Llama-13B	Continuous	224	52.0	-	-	-	-	-
LaVIT [15]	Llama-7B	Continuous	224	66.0	46.8	-	-	-	-
DreamLLM [7]	Vicuna-7B	Continuous	224	72.9	-	41.8	-	-	26.6
Unified-IO 2 [29]	6.8B from scratch	Continuous	384	79.4	-	-	87.7	-	-
Janus [58]	DeepSeek-1.3B	Continuous	384	77.3	59.1	-	87.0	1338	34.3
CM3Leon [68]	7B from scratch	Discrete	256	47.6	-	-	-	-	-
LWM [27]	Llama-2-7B	Discrete	256	55.8	44.8	18.8	75.2	-	-
Show-o [62]	Phi-1.5-1.3B	Discrete	256	59.3	48.7	-	73.8	948	-
Chameleon [46]	34B from scratch	Discrete	512	69.6	-	-	-	-	-
Liquid [59]	Gemma-7B	Discrete	512	71.3	58.4	42.4	81.1	1119	-
VILA-U [61]	Llama-2-7B	Discrete	256	75.3	58.3	48.3	83.9	1336	27.7
UniTok	Llama-2-7B	Discrete	256	76.8	61.1	51.6	83.2	1448	33.9

Table 4: Comparison with other visual generation methods on the GenEval benchmark.

Method	Type	#Data	Single Obj.	Two Obj.	Counting	Colors	Position	Color Attri.	Overall↑
SD v2.1 [39]	Diffusion	2000M	0.98	0.51	0.44	0.85	0.07	0.17	0.50
SD-XL [36]	Diffusion	2000M	0.98	0.74	0.39	0.85	0.15	0.23	0.55
DALL-E 3 [2]	Diffusion	-	0.96	0.87	0.47	0.83	0.43	0.45	0.67
Show-o [62]	Discrete Diff.	36M	0.95	0.52	0.49	0.82	0.11	0.28	0.53
		2.0B	0.98	0.80	0.66	0.84	0.31	0.50	0.68
LWM [27]	Autoregressive	-	0.93	0.41	0.46	0.79	0.09	0.15	0.47
Janus [58]	Autoregressive	-	0.97	0.68	0.30	0.84	0.46	0.42	0.61
Liquid [59]	Autoregressive	30M	0.98	0.73	0.32	0.76	0.17	0.37	0.55
UniTok	Autoregressive	30M	0.99	0.71	0.36	0.79	0.26	0.45	0.59

for evaluation, high-resolution images are preferred because they potentially capture more fine-grained details. Despite this makes FID across different resolutions less comparable, we show that our model achieves impressive performance even at the the smallest resolution, showcasing its ability to generate high-quality, detail-rich images.

We present some examples of the images generated by our model in Figure 5, using text prompts sampled from MJHQ-30K. The visualization results demonstrate our model is capable of synthesizing photo-realistic and visually appealing images. Moreover, the model is able to comprehend a wide spectrum of concepts, such as ‘Vincent van Gogh painting style’ and ‘bitcoin’, and flexibly combine these concepts to synthesize creative images.

4.5 Ablation Studies

Impact of Supervision Types. To ablate the impact of contrastive and reconstruction losses in UniTok training, we conduct experiments on tokenizers trained with different supervision types, as shown in Table 6. It is worth noting that all the tokenizers are vector-quantized even though some do not have reconstruction supervision. First, we show that reconstruction-oriented tokenizer significantly lags behind tokenizers with contrastive supervision in visual understanding performance. This observation evidences the limitations of traditional VQVAE. Second, we demonstrate that reconstruction and contrastive training objectives do not inherently conflict, or can be addressed by enhancing discrete feature space. With multi-codebook quantization, the jointly trained tokenizer not only exhibits understanding performance on par with the tokenizer trained solely with contrastive loss, but also slightly improves generation performance over the reconstruction-oriented tokenizer.

MCQ v.s. RQ. Following discussions in Section 3.3, we provide an apple-to-apple comparison between multi-codebook quantization and residual quantization in Table 7. For fair comparisons, we

Table 5: Results on MJHQ-30K.

Method	Type	Res.	FID↓
SD-XL [36]	Diffusion	1024	9.55
PixArt [6]	Diffusion	1024	6.14
Playground [20]	Diffusion	1024	4.48
Liquid [59]	Autoregressive	512	5.47
Janus [58]	Autoregressive	384	10.10
LWM [27]	Autoregressive	256	17.77
Show-o [62]	Discrete Diff.	256	15.18
VILA-U [61]	Autoregressive	256	12.81
UniTok	Autoregressive	256	7.46



Figure 5: Images generated in a resolution of 256×256 with our unified MLLM.

Table 6: Impact of different supervision types on downstream generation and understanding performance. The rFID and gFID are measured on the ImageNet (256×256) validation set. LlamaGen-L [43] is adopted as the generator for gFID evaluation.

Supervision	Generation		Understanding					
	rFID \downarrow	gFID \downarrow	VQAv2	GQA	SciQA	TextVQA	POPE	MME
Contrastive	—	—	68.95	56.89	65.64	49.89	82.34	1373
Reconstruction	0.82	3.59	56.33	47.53	63.26	43.65	77.09	902
Recon. + Contra.	0.72	3.26	69.14	56.06	65.25	49.22	81.42	1333

directly implement RQ on our codebase, keeping all the training settings the same as UniTok. Both tokenizers are trained on a 512M subset of DataComp-1B. Notably, unlike MCQ, RQ by default uses a shared large codebook. To keep the global codebook size the same as UniTok, we set the codebook size of RQ to 32768. It can be seen that RQ demonstrates inferior reconstruction performance and lower classification accuracy compared to MCQ under the high bottleneck dimension (64-d) setting.

Number of Sub-Codebooks. To gain deeper insights into multi-codebook quantization, we evaluate how tokenizer performance changes with the number of sub-codebooks in Table 8. Specifically, the size of a codebook is denoted as $A \times B$, where A is the number of sub-codebook and B is the size of sub-codebook. For rFID evaluation, we train the tokenizer solely with reconstruction loss on OpenImages [17], and evaluated it on ImageNet (256×256) validation set. While for ImageNet zero-shot accuracy evaluation, the tokenizer is trained on DataComp-1B 128m subset using only contrastive loss. Given a constant global codebook size, we see that increasing the number of sub-codebooks consistently improves reconstruction FID and classification accuracy. This indicates that MCQ generally benefits vector-quantized models, independent of the training objectives.

Table 7: MCQ v.s. RQ.

Method	Code Shape	Code Dim.	rFID \downarrow	Accuracy
RQ	$16 \times 16 \times 8$	64	3.46	58.8
MCQ	$16 \times 16 \times 8$	64	0.55	63.7

Table 8: Ablation on number of sub-codebooks.

Codebook / Vocabulary	$1 \times 16384 / 2^{14}$	$2 \times 8192 / 2^{26}$	$4 \times 4096 / 2^{48}$	$8 \times 2048 / 2^{88}$
rFID \downarrow	1.50	0.98	0.54	0.33

CLIP Weight Initialization. We notice that higher ImageNet accuracy does not guarantee superior downstream performance. In Table 9, we ablate the impact of CLIP weight initialization on visual understanding performance. Specifically, we adopt the classic LLaVA framework for evaluation,

replacing the original CLIP tokenizer with UniTok while keeping all other the training settings unchanged. One tokenizer is initialized with the pretrained ViTamin-L-256 [5] weights, while the other is randomly initialized. To our surprise, UniTok that is trained from scratch surpasses the one initialized with pretrained CLIP weights, despite the latter actually achieves better zero-shot classification accuracy. This suggests downstream VQA performance may not be highly correlated with ImageNet classification accuracy. More importantly, it also implies that CLIP weight initialization may serve as a negative prior for unified tokenizers, as the unified visual feature space could drastically differ from CLIP feature space.

Table 9: Comparison of different initialization methods under the LLaVA framework. \dagger indicates the model uses CLIP weights for initialization. We highlight the default setting of UniTok in gray.

Tokenizer	VQAv2	GQA	TextVQA	POPE	MME
UniTok \dagger	69.9	56.2	49.3	81.2	1331
UniTok	72.4	58.2	51.6	82.4	1392

5 Limitations and Conclusion

This paper studies unified visual tokenization for generation and understanding, which serves as the cornerstone of unified multimodal large language models. We investigate the training paradigm of unified tokenizers and identify that the current challenge in unification mainly arises from the limited representational power of discrete tokens. To address this limitation, we introduce multi-codebook quantization and attention projection to build a unified tokenizer called UniTok. We show that UniTok excels in downstream visual generation and understanding tasks. The ablation study further reveals that discriminative and generative representation learning does not inherently conflict. We hope our findings could inspire future research in this domain.

However, due to limited computational resources, UniTok is only trained for one epoch, which is not sufficient for CLIP-based semantic representation learning. We believe extending the training schedule could further benefit the tokenizer, especially in understanding performance.

Acknowledgments This work has been supported by the National Key R&D Program of China (Grant No. 2022YFB3608300), Hong Kong Research Grant Council - Early Career Scheme (Grant No. 27209621), General Research Fund Scheme (Grant No. 17202422, 17212923, 17215025) Themebased Research (Grant No. T45-701/22-R) and Shenzhen Science and Technology Innovation Commission (SGDX20220530111405040). We sincerely thank Seng Ye for the insightful discussions and valuable contributions to this project.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] James Betker, Gabriel Goh, Li Jing, Tim Brooks, Jianfeng Wang, Linjie Li, Long Ouyang, Juntang Zhuang, Joyce Lee, Yufei Guo, et al. Improving image generation with better captions. *Computer Science*. <https://cdn.openai.com/papers/dall-e-3.pdf>, 2(3):8, 2023.
- [3] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11315–11325, 2022.
- [4] Hao Chen, Yujin Han, Fangyi Chen, Xiang Li, Yidong Wang, Jindong Wang, Ze Wang, Zicheng Liu, Difan Zou, and Bhiksha Raj. Masked autoencoders are effective tokenizers for diffusion models. *arXiv preprint arXiv:2502.03444*, 2025.
- [5] Jieneng Chen, Qihang Yu, Xiaohui Shen, Alan Yuille, and Liang-Chieh Chen. Vitamin: Designing scalable vision models in the vision-language era. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12954–12966, 2024.
- [6] Junsong Chen, Jincheng Yu, Chongjian Ge, Lewei Yao, Enze Xie, Yue Wu, Zhongdao Wang, James Kwok, Ping Luo, Huchuan Lu, et al. Pixart-alpha: Fast training of diffusion transformer for photorealistic text-to-image synthesis. *arXiv preprint arXiv:2310.00426*, 2023.
- [7] Runpei Dong, Chunrui Han, Yuang Peng, Zekun Qi, Zheng Ge, Jinrong Yang, Liang Zhao, Jianjian Sun, Hongyu Zhou, Haoran Wei, et al. Dreamllm: Synergistic multimodal comprehension and creation. *arXiv preprint arXiv:2309.11499*, 2023.
- [8] 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.
- [9] Samir Yitzhak Gadre, Gabriel Ilharco, Alex Fang, Jonathan Hayase, Georgios Smyrnis, Thao Nguyen, Ryan Marten, Mitchell Wortsman, Dhruba Ghosh, Jieyu Zhang, et al. Datacomp: In search of the next generation of multimodal datasets. *Advances in Neural Information Processing Systems*, 36, 2024.
- [10] Yuying Ge, Yixiao Ge, Ziyun Zeng, Xintao Wang, and Ying Shan. Planting a seed of vision in large language model. *arXiv preprint arXiv:2307.08041*, 2023.
- [11] Yuying Ge, Sijie Zhao, Jinguo Zhu, Yixiao Ge, Kun Yi, Lin Song, Chen Li, Xiaohan Ding, and Ying Shan. Seed-x: Multimodal models with unified multi-granularity comprehension and generation. *arXiv preprint arXiv:2404.14396*, 2024.
- [12] Google. Experiment with gemini 2.0 flash native image generation, 2025.
- [13] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017.
- [14] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[15] Yang Jin, Kun Xu, Liwei Chen, Chao Liao, Jianchao Tan, Bin Chen, Chenyi Lei, An Liu, Chengru Song, Xiaoqiang Lei, et al. Unified language-vision pretraining with dynamic discrete visual tokenization. *arXiv preprint arXiv:2309.04669*, 2023.

[16] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019.

[17] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, et al. The open images dataset v4: Unified image classification, object detection, and visual relationship detection at scale. *International journal of computer vision*, 128(7):1956–1981, 2020.

[18] Doyup Lee, Chiheon Kim, Saehoon Kim, Minsu Cho, and Wook-Shin Han. Autoregressive image generation using residual quantization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11523–11532, 2022.

[19] Xingjian Leng, Jaskirat Singh, Yunzhong Hou, Zhenchang Xing, Saining Xie, and Liang Zheng. Repa-e: Unlocking vae for end-to-end tuning with latent diffusion transformers. *arXiv preprint arXiv:2504.10483*, 2025.

[20] Daiqing Li, Aleks Kamko, Ehsan Akhgari, Ali Sabet, Linmiao Xu, and Suhail Doshi. Playground v2. 5: Three insights towards enhancing aesthetic quality in text-to-image generation. *arXiv preprint arXiv:2402.17245*, 2024.

[21] Hao Li, Changyao Tian, Jie Shao, Xizhou Zhu, Zhaokai Wang, Jinguo Zhu, Wenhan Dou, Xiaogang Wang, Hongsheng Li, Lewei Lu, et al. Synergen-vl: Towards synergistic image understanding and generation with vision experts and token folding. *arXiv preprint arXiv:2412.09604*, 2024.

[22] Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Gadre, Hritik Bansal, Etrash Guha, Sedrick Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *arXiv preprint arXiv:2406.11794*, 2024.

[23] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024.

[24] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023.

[25] Ji Lin, Hongxu Yin, Wei Ping, Pavlo Molchanov, Mohammad Shoeybi, and Song Han. Vila: On pre-training for visual language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26689–26699, 2024.

[26] Zhiqiu Lin, Deepak Pathak, Baiqi Li, Jiayao Li, Xide Xia, Graham Neubig, Pengchuan Zhang, and Deva Ramanan. Evaluating text-to-visual generation with image-to-text generation. In *European Conference on Computer Vision*, pages 366–384. Springer, 2025.

[27] Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with ringattention. *arXiv preprint arXiv:2402.08268*, 2024.

[28] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[29] Jiasen Lu, Christopher Clark, Sangho Lee, Zichen Zhang, Savya Khosla, Ryan Marten, Derek Hoiem, and Aniruddha Kembhavi. Unified-io 2: Scaling autoregressive multimodal models with vision language audio and action. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26439–26455, 2024.

[30] Chuofan Ma, Yi Jiang, Jiannan Wu, Zehuan Yuan, and Xiaojuan Qi. Groma: Localized visual tokenization for grounding multimodal large language models. In *European Conference on Computer Vision*, pages 417–435. Springer, 2025.

[31] Brandon McKinzie, Zhe Gan, Jean-Philippe Fauconnier, Sam Dodge, Bowen Zhang, Philipp Dufter, Dhruti Shah, Xianzhi Du, Futang Peng, Floris Weers, et al. Mm1: Methods, analysis & insights from multimodal llm pre-training. *arXiv preprint arXiv:2403.09611*, 2024.

[32] Byeon Minwoo, Park Beomhee, Kim Haecheon, Lee Sungjun, Baek Woonhyuk, and Kim. Saehoon. Coyo-700m: Image-text pair dataset., 2022.

[33] OpenAI. Introducing 4o image generation, 2025.

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

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

[36] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023.

[37] Liao Qu, Huichao Zhang, Yiheng Liu, Xu Wang, Yi Jiang, Yiming Gao, Hu Ye, Daniel K Du, Zehuan Yuan, and Xinglong Wu. Tokenflow: Unified image tokenizer for multimodal understanding and generation. *arXiv preprint arXiv:2412.03069*, 2024.

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[39] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[40] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10684–10695, 2022.

[41] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022.

[42] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019.

[43] 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.

[44] Quan Sun, Yufeng Cui, Xiaosong Zhang, Fan Zhang, Qiying Yu, Yueze Wang, Yongming Rao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative multimodal models are in-context learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14398–14409, 2024.

[45] Quan Sun, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, Yueze Wang, Hongcheng Gao, Jingjing Liu, Tiejun Huang, and Xinlong Wang. Generative pretraining in multimodality. *arXiv preprint arXiv:2307.05222*, 2023.

[46] Chameleon Team. Chameleon: Mixed-modal early-fusion foundation models. *arXiv preprint arXiv:2405.09818*, 2024.

[47] Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivièvre, Mihir Sanjay Kale, Juliette Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[48] 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.

[49] Shengbang Tong, Ellis Brown, Penghao Wu, Sanghyun Woo, Manoj Middepogu, Sai Charitha Akula, Jihan Yang, Shusheng Yang, Adithya Iyer, Xichen Pan, et al. Cambrian-1: A fully open, vision-centric exploration of multimodal llms. *arXiv preprint arXiv:2406.16860*, 2024.

[50] Shengbang Tong, David Fan, Jiachen Zhu, Yunyang Xiong, Xinlei Chen, Koustuv Sinha, Michael Rabbat, Yann LeCun, Saining Xie, and Zhuang Liu. Metamorph: Multimodal understanding and generation via instruction tuning. *arXiv preprint arXiv:2412.14164*, 2024.

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

[52] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[53] Aaron Van Den Oord, Oriol Vinyals, et al. Neural discrete representation learning. *Advances in neural information processing systems*, 30, 2017.

[54] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[55] Guangzhi Wang, Yixiao Ge, Xiaohan Ding, Mohan Kankanhalli, and Ying Shan. What makes for good visual tokenizers for large language models? *arXiv preprint arXiv:2305.12223*, 2023.

[56] Xinlong Wang, Xiaosong Zhang, Zhengxiong Luo, Quan Sun, Yufeng Cui, Jinsheng Wang, Fan Zhang, Yueze Wang, Zhen Li, Qiying Yu, et al. Emu3: Next-token prediction is all you need. *arXiv preprint arXiv:2409.18869*, 2024.

[57] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837, 2022.

[58] Chengyue Wu, Xiaokang Chen, Zhiyu Wu, Yiyang Ma, Xingchao Liu, Zizheng Pan, Wen Liu, Zhenda Xie, Xingkai Yu, Chong Ruan, et al. Janus: Decoupling visual encoding for unified multimodal understanding and generation. *arXiv preprint arXiv:2410.13848*, 2024.

[59] Junfeng Wu, Yi Jiang, Chuofan Ma, Yuliang Liu, Hengshuang Zhao, Zehuan Yuan, Song Bai, and Xiang Bai. Liquid: Language models are scalable and unified multi-modal generators. *arXiv preprint arXiv:2412.04332*, 2024.

[60] Shengqiong Wu, Hao Fei, Leigang Qu, Wei Ji, and Tat-Seng Chua. Next-gpt: Any-to-any multimodal llm. *arXiv preprint arXiv:2309.05519*, 2023.

[61] Yecheng Wu, Zhuoyang Zhang, Junyu Chen, Haotian Tang, Dacheng Li, Yunhao Fang, Ligeng Zhu, Enze Xie, Hongxu Yin, Li Yi, et al. Vila-u: a unified foundation model integrating visual understanding and generation. *arXiv preprint arXiv:2409.04429*, 2024.

[62] Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. *arXiv preprint arXiv:2408.12528*, 2024.

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

[64] Shukang Yin, Chaoyou Fu, Sirui Zhao, Ke Li, Xing Sun, Tong Xu, and Enhong Chen. A survey on multimodal large language models. *arXiv preprint arXiv:2306.13549*, 2023.

- [65] 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 preprint arXiv:2110.04627*, 2021.
- [66] Lijun Yu, Yong Cheng, Kihyuk Sohn, José Lezama, Han Zhang, Huiwen Chang, Alexander G Hauptmann, Ming-Hsuan Yang, Yuan Hao, Irfan Essa, et al. Magvit: Masked generative video transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10459–10469, 2023.
- [67] Lijun Yu, José Lezama, Nitesh B Gundavarapu, Luca Versari, Kihyuk Sohn, David Minnen, Yong Cheng, Vighnesh Birodkar, Agrim Gupta, Xiuye Gu, et al. Language model beats diffusion-tokenizer is key to visual generation. *arXiv preprint arXiv:2310.05737*, 2023.
- [68] Lili Yu, Bowen Shi, Ramakanth Pasunuru, Benjamin Muller, Olga Golovneva, Tianlu Wang, Arun Babu, Binh Tang, Brian Karrer, Shelly Sheynin, et al. Scaling autoregressive multi-modal models: Pretraining and instruction tuning. *arXiv preprint arXiv:2309.02591*, 2(3), 2023.
- [69] Qihang Yu, Mark Weber, Xueqing Deng, Xiaohui Shen, Daniel Cremers, and Liang-Chieh Chen. An image is worth 32 tokens for reconstruction and generation. *Advances in Neural Information Processing Systems*, 37:128940–128966, 2024.
- [70] Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.
- [71] 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.
- [72] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.
- [73] Chunting Zhou, Lili Yu, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. *arXiv preprint arXiv:2408.11039*, 2024.
- [74] 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.

NeurIPS Paper Checklist

1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [\[Yes\]](#) .

Justification: We have included the motivation, empirical findings, technical innovations, and major achievements of our work in the abstract and introduction.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [\[Yes\]](#)

Justification: Please see 'Limitations and Conclusion' section.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [\[NA\]](#)

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: We have included the implementation details of our tokenizer and MLLM (such as training data, hyperparameter configurations, testing framework, etc) in the ‘Experiments’ part.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
 - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
 - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
 - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
 - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [\[Yes\]](#)

Justification: Most code and data for this paper are open-source, except the unified MLLM training data, which includes some internal, non-public data. However, our main focus – tokenizer training – uses only publicly available data.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [\[Yes\]](#)

Justification: Please see our code for the full set of hyperparameter configurations.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [\[No\]](#)

Justification: We mainly evaluate our methods using ImageNet FID and zero-shot accuracy, metrics known for low variance. Following prior work, we omit error bars in our experiments.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer “Yes” if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [\[Yes\]](#)

Justification: It takes roughly 50 hours with the equivalent computing power of 256 A100 GPUs to train the tokenizer.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [\[Yes\]](#)

Justification: I have read and confirmed that this paper complies with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [\[Yes\]](#)

Justification: Please see Appendix.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: **[No]**

Justification: We will add safeguards in our future release.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: **[Yes]**

Justification: We strictly followed the licenses to use other assets (like DataComp) with credits.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, paperswithcode.com/datasets has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: This paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. Institutional review board (IRB) approvals or equivalent for research with human subjects

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: This paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. Declaration of LLM usage

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

A Attention Projection Modules

Figure 6 illustrates the adaptations we made to the tradition MHA module, which enables channel compression and expansion in token factorization.

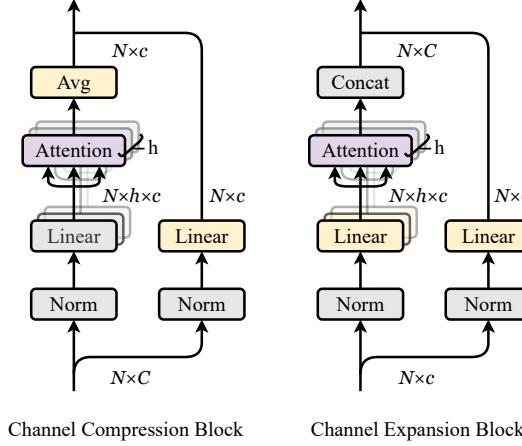


Figure 6: Modified attention blocks for factorization. Modules in yellow indicate a change in the number of channels. C and c stand for the channel dimension, h is the number of heads in the multi-head attention module. $C = h \times c$.

B More Generation Results

We provide the results on GenAI-Bench in Table 10 and Table 11. UniTok consistently delivers superior generation performance on this benchmark.

Table 10: Comparison with other visual generation methods on GenAI-Bench (basic prompts).

Method	Type	#Training Images	Attribute↑	Scene↑	Relation↑			Overall↑
					Spatial	Action	Part	
SD v2.1 [39]	Diffusion	2000M	0.80	0.79	0.76	0.77	0.80	0.78
SD-XL [36]	Diffusion	2000M	0.84	0.84	0.82	0.83	0.89	0.83
Midjourney v6	Diffusion	–	0.88	0.87	0.87	0.87	0.91	0.87
DALL-E 3 [2]	Diffusion	–	0.91	0.90	0.92	0.89	0.91	0.90
Show-o [62]	Discrete Diff.	36M	0.72	0.72	0.70	0.70	0.75	0.70
LWM [27]	Autoregressive	–	0.63	0.62	0.65	0.63	0.70	0.63
VILA-U [61]	Autoregressive	15M	0.78	0.78	0.77	0.78	0.79	0.76
Liquid [59]	Autoregressive	30M	0.84	0.86	0.81	0.83	0.91	0.83
UniTok	Autoregressive	30M	0.85	0.87	0.86	0.86	0.89	0.85

Table 11: Comparison with other visual generation methods on GenAI-Bench (advanced prompts).

Method	Type	#Training Images	Count↑	Differ↑	Compare↑	Logical↑		Overall↑
						Negate	Universal	
SD v2.1 [39]	Diffusion	2000M	0.68	0.70	0.68	0.54	0.64	0.62
SD-XL [36]	Diffusion	2000M	0.71	0.73	0.69	0.50	0.66	0.63
Midjourney v6	Diffusion	–	0.78	0.78	0.79	0.50	0.76	0.69
DALL-E 3 [2]	Diffusion	–	0.82	0.78	0.82	0.48	0.80	0.70
Show-o [62]	Discrete Diff.	36M	0.70	0.62	0.71	0.51	0.65	0.60
LWM [27]	Autoregressive	–	0.59	0.58	0.54	0.49	0.52	0.53
VILA-U [61]	Autoregressive	15M	0.70	0.71	0.74	0.53	0.66	0.64
Liquid [59]	Autoregressive	30M	0.76	0.73	0.74	0.46	0.74	0.65
UniTok	Autoregressive	30M	0.76	0.76	0.79	0.46	0.73	0.67

C Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.