UniCode : Learning a Unified Codebook for Multimodal Large Language Models

Sipeng Zheng[1], Bohan Zhou[2], Yicheng Feng[2], Ye Wang[1], Zongqing Lu[2,1⋆]
[1]Beijing Academy of Artificial Intelligence (BAAI)
[2]School of Computer Science, Peking University
spzheng@baai.ac.cn zhoubh@stu.pku.edu.cn yewang@stu.ecnu.edu.cn
{fyc813, zongqing.lu}@pku.edu.cn

Abstract. In this paper, we propose UniCode, a novel approach within the domain of multimodal large language models (MLLMs) that learns a unified codebook to efficiently tokenize visual, text, and potentially other types of signals. This innovation addresses a critical limitation in existing MLLMs: their reliance on a text-only codebook, which restricts MLLMs' ability to generate images and texts in a multimodal context. Towards this end, we propose a language-driven iterative training paradigm, coupled with an in-context pre-training task we term "image decompression", enabling our model to interpret compressed visual data and generate high-quality images. The unified codebook empowers our model to extend visual instruction tuning to non-linguistic generation tasks. Moreover, UniCode is adaptable to diverse stacked quantization approaches in order to compress visual signals into a more compact token representation. Despite using significantly fewer parameters and less data during training, UniCode demonstrates promising capabilities in visual reconstruction and generation. It also achieves performance comparable to leading MLLMs across a spectrum of VQA benchmarks.

Keywords: Multimodal Learning · Large Model · Visual Generation

1 Introduction

The rapid development of large language models (LLMs) [39,50,51] has spurred growing interest in their multimodal counterparts [1, 29, 33]. Empowered by LLMs, existing foundation models have shown remarkable capabilities in multimodal understanding, spanning from basic image classification [41,63] and captioning [29,54], to more intricate tasks such as making strategic high-level plans for open-world agents [17,67]. As illustrated in Figure 1 (a), these works rely on a lightweight module such as a multimodal projector [29,33] to seamlessly map visual signals into LLMs' textual space with minimal training cost.

Progress has been made, though most multimodal large language models (MLLMs) are still limited to language generation. This limitation stems from their reliance on text-only codebooks, which restricts their application across

⋆ Corresponding author.

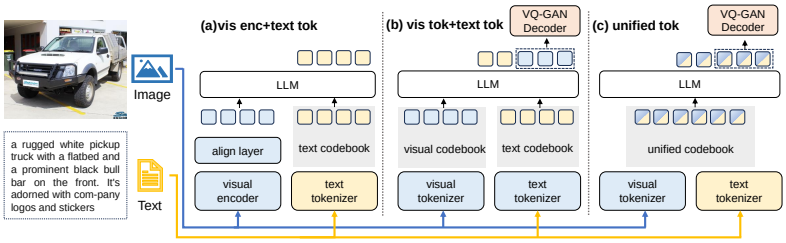

Fig. 1: Three paradigms of MLLMs: **(a) vis enc+text tok** incorporates a lightweight module to align visual signals with the LLM, specifically designed for language generation; **(b) vis tok+text tok** concatenates the text codebook with quantized visual tokens, significantly increasing the computational cost and complexity; **(c) unified tok** learns a unified codebook to interpret both visual and text modalities without additional modules. We explore the last option by proposing UniCode in this work.

diverse scenarios, such as image generation [27]. Note that images, like text, can be tokenized into a series of discrete codes through Vector Quantization (VQ) [15, 27, 52], which suggests a straightforward strategy to enhance MLLMs: extending the LLMs' codebook to include visual codes [8, 36], as shown in Figure 1 (b). However, this approach introduces new challenges. Firstly, it requires considerable effort to overcome the substantial modality gap between visual and text codes. Secondly, enlarging the codebook leads to an upsurge in model parameters and risks "codebook collapse" [12], where the model overly relies on a limited set of codes, posing significant obstacles in the training of MLLMs.

Instead of expanding the codebook size, we pose a question: "*Is it feasible to learn a unified codebook capable of quantizing language, vision, and potentially other modalities?*" As illustrated in Figure 1 (c), a unified codebook could seamlessly integrate various data types, thereby equipping MLLMs with the ability to generate non-linguistic content without the need for additional parameters or specialized modules. Recent initiatives have explored mapping visual signals into the text token space of a frozen LLM. Yet, these efforts often yield inferior results compared with those using a learned visual codebook [26, 31], or they require an exponential number of tokens to accurately represent an image [62]. More importantly, relying on a frozen LLM means stopping adaptation to follow human instructions through alignment with high-quality, instruction-following data [6, 49]. Hence, being trainable is an indispensable feature for MLLMs.

Considering this, we introduce **UniCode**, the first attempt to craft a **Uni**fied **Code**book for MLLMs by integrating a VAE-style visual tokenizer [52] with the LLM. To achieve this, we alternate the training process between these two modules, iteratively synchronizing the visual tokenizer's codebook with the LLM's to maintain consistency. This process, which we term "language-driven iterative training", utilizes a smooth moving average to update the visual codebook. To further enhance the fidelity of images generated by UniCode, we introduce a novel pre-training task: in-context image decompression. This task leverages in-context instructions to transform compressed image data into discrete visual tokens. Furthermore, UniCode is designed to support stacked quantization [27, 62]

to optimize visual tokenization efficiency, which compresses images into stacked code maps to reduce the feature resolution. The unified codebook effectively converts visual inputs into language tokens. Based on it, UniCode broadens the scope of visual instruction tuning [31] by reformulating multimodal generation in the context of instruction-following format.

Our key contributions can be summarized as follows: ❶ We propose UniCode, an innovative paradigm for MLLMs, featuring a unified codebook capable of tokenizing both visual and textual inputs. To achieve this, we adopt language-driven iterative training to learn such a codebook without additional parameters for visual-text alignment. ❷ We enrich the model's tuning with non-linguistic data integrated into the existing visual instructional dataset. The tuning process is augmented by a unique in-context image decompression task, designed to improve the model's ability to interpret and generate complex multimodal content. ❸ Experimental analysis shows the effectiveness of UniCode compared to state-of-the-art MLLMs. Notably, this is achieved using a more efficient visual encoder that requires significantly fewer parameters and training samples.

2 Related Work

2.1 Visual Quantization

Vector quantization (VQ) has achieved remarkable success in creating high-resolution images [5, 52, 61] and videos [20, 56, 57]. VQ-VAE [52] first converts images into discrete representations and autoregressively models their distribution. Following this work, Razavi *et al.* [43] adopt learned hierarchical representations, while Esser *et al.* [15] introduce perceptual adversarial loss [53] to refine the perceptual quality of reconstructed images. Inspired by residual quantization [24,35], Lee *et al.* [27] develop residual quantization (RQ), a technique that encodes images into a stacked map of discrete codes, thereby efficiently reducing the spatial resolution of features. You *et al.* [59] propose hierarchical vector quantization (HQ), which employs a pyramid scheme with two-level codes for image encoding. Despite these advancements, a limitation of these methods is that their codebooks, being jointly trained with the encoder and decoder, lack direct interpretability in natural language. To address this, recent research has investigated leveraging frozen LLMs for image understanding [26, 31]. Liu *et al.* [31] innovate with LQAE, replacing the learned codebook with a text vocabulary from the frozen BERT [11]. Despite its novelty, LQAE falls short in the fidelity of image reconstruction, underscoring the challenges of using a frozen LLM for content generation across modalities. Yu *et al.* [62] aim to solve this challenge by arranging quantized tokens in a multi-layer, coarse-to-fine pyramid.

2.2 Multimodal Instruction Tuning

In the field of natural language processing (NLP), previous studies have made significant strides in enabling LLMs [4, 7, 42, 64] to comprehend and execute natural language instructions through a process known as instruction tuning [40].

Following this practice, recent efforts have extended its application to the multimodal realm [55, 58]. Among these works, Liu *et al.* [33] introduce LLaVA, the first model to apply the concept of visual instruction tuning to build a versatile visual assistant. Following this, Li *et al.* [28] propose Mimic-it, enhancing the model's capability by incorporating multimodal in-context information directly into instruction data. Zhang *et al.* [65] and Zhao *et al.* [66] have furthered research in this area by scaling instructional data and enriching it with text-dense images. In addition to simply increasing data volume, Dai *et al.* [9] develop InstructBLIP based on BLIP-2 [29], which introduces an advanced visual feature extraction mechanism to bolster performance across vision-language tasks.

While existing foundation models have marked impressive strides in multimodal benchmarks, their capabilities are still limited to text-only generation. Recently, a notable advancement, Emu, was introduced by Sun *et al.* [48], a model crafted for generative pretraining across multiple modalities. Despite its innovation, Emu necessitates a robust visual encoder with 1 billion parameters and relies on 80 million samples for effective pretraining. Meanwhile, Lu *et al.* [36] propose Unified-IO 2, an MLLM akin to our approach, which encodes and generates text, vision, audio, and interleaved sequences. Yet, it also requires significant computational demands with 1 billion image-text pairs. Instead, UniCode diverges from the above approaches by focusing on learning a unified codebook. We demonstrate through experiments that our model substantially decreases resource requirements while still achieving competitive results.

3 Unified Codebook Learning

UniCode is built without bells and whistles for easy replication based on the arbitrary transformer-based architecture of LLMs. Our primary objective is to craft a unified codebook that efficiently tokenizes multimodal information. To achieve this, we first provide a brief overview of visual tokenization in Section 3.1. Then, in Section 3.2, we propose our language-driven iterative training paradigm, discussing various alternatives for synchronizing the learning of both visual and linguistic codebooks. In Section 3.3, we further introduce a novel image decompression task designed for generation enhancement.

3.1 Visual Tokenization

Visual tokenization [52] is a process that compresses visual signals (e.g., images) into a series of discrete tokens. It generally consists of an encoder $\mathbb{E}$, a decoder $\mathbb{D}$, and a codebook $\mathbb{C} = \{(k, e(k))|k \in \{1, \cdots, \mathrm{K}\}\}$, where K denotes the codebook size. Here, $\mathbb{C}$ is a finite set of pairs, each consisting of a code k and its corresponding n-dimensional code embedding $e(k) \in \mathbb{R}^n$. Similar to an LLM, for each vector $z \in \mathbb{R}^n$, the operation of visual tokenization $Q(z; \mathbb{C})$ is defined to select the code from $\mathbb{C}$ whose embedding is closest to z, which is denoted as:

$$Q(z; \mathbb{C}) = \underset{k \in \{1, \cdots, \mathrm{K}\}}{\operatorname{argmin}} \|z - e(k)\|_2^2. \tag{1}$$

Given an image $\mathcal{I} \in \mathbb{R}^{\mathrm{H}\times\mathrm{W}\times 3}$, the visual tokenizer first uses the encoder $\mathbb{E}$ to derive its feature map $Z_0 \in \mathbb{R}^{h\times w\times c}$, with c representing the embedding dimension. Subsequently, each vector $z \in Z_0$ is assigned to the closest code within the codebook $\mathbb{C}$, yielding a code map $M_{ij} = \mathcal{Q}(Z_{0ij}; \mathbb{C})$ and its quantized feature map $Z_{ij} = e(M_{ij})$, where $i \in [1, h]$ and $j \in [1, w]$. The decoder then utilizes Z to reconstruct the image. In this work, the LLM is designed to either interpret these quantized embeddings as input or to directly generate discrete tokens that can signify visual semantic concepts.

Efficient Stack Quantization. As the resolution $(h \times w)$ of the code map M increases, the computational demand on the LLM grows quadratically. Given the LLM's inherent constraint on processing only a finite length of token sequences, reducing the resolution of the code map becomes crucial. However, the fidelity of reconstruction is deeply influenced by the tokens' bit-depth [45]. To strike a balance between efficiency and quality in visual tokenization, we consider stacked quantization as a viable solution [38] to decrease the resolution of M. Specifically, stacked quantization preserves the visual information by generating a D-layer code map $\hat{M}_d \in \mathbb{N}^{\hat{h}_d\times\hat{w}_d\times D}$, where $d \in [1, D]$ and the dimensions $\hat{h}$, $\hat{w}$ are significantly reduced compared to h, w. For each element (i, j) in the code map, its ultimate embedding is an aggregation of D quantized vectors $\hat{z}_{ij} = \mathcal{F}_{d=1}^{D} e(\hat{M}_{i,j,d})$, with $\mathcal{F}$ denoting the aggregation function (e.g., concatenation in HQ [59], cumulative sum in RQ [27]). In our study, HQ serves as a prime example for illustration. Note that UniCode is adaptable to various variants of stacked quantization, making it a fertile area for further research.

In our approach, these visual tokens directly correspond to entries in the LLM's codebook, enabling our proposed UniCode to seamlessly interpret these aggregated quantized embeddings. Furthermore, we propose an image decompression task to enhance the LLM's capability for converting quantized embeddings into language tokens.

3.2 Codebook Learning Paradigm

Before introducing our proposed paradigm, we first discuss two alternatives for obtaining a unified codebook:

Frozen LLM Codebook. We start with a straightforward approach, as illustrated in Figure 2 (a), where the visual tokenizer's codebook is initialized with a pretrained LLM and remains frozen during training. While this approach directly links the visual tokenizer with the language vocabulary, it falls short in accurately capturing the semantic nuances in images. Our empirical study further reveals that employing a frozen codebook adversely impacts the quality of reconstruction, especially for stacked quantization methods such as hierarchical quantization (HQ). This can be primarily attributed to two factors: the absence of an explicit mechanism to synchronize the encoder/decoder with the frozen codebook, and the varying scales of multi-layer embeddings that divide the codebook into multiple parts [31]. The pursuit of optimal reconstruction fidelity motivates the development of dynamic alignment between the codebook and the encoder/decoder of the visual tokenizer.

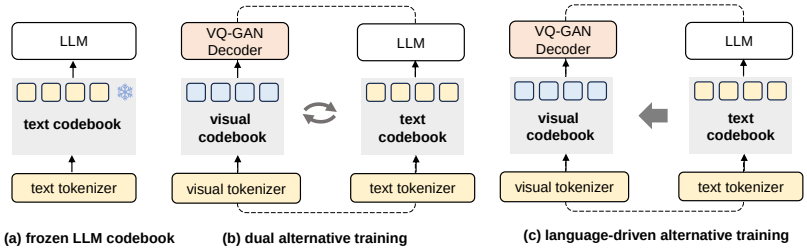

Fig. 2: Illustration of multiple paradigms to obtain a unified codebook. The dotted line indicates the training loop: **(a) frozen LLM codebook**, which initiates the codebook with a pretrained LLM and freezes it during training; **(b) dual alternative training**, which jointly trains both the visual tokenizer and the LLM by alternatively updating each one's codebook using the other's parameters; **(c) language-driven iterative training**, which smoothly updates the codebook of the visual tokenizer with the LLM's through a moving average manner.

Dual Alternative Training. As shown in Figure 2 (b), this approach dynamically aligns the visual tokenizer and LLM by alternating their training. In each training step of the visual tokenizer, its codebook is directly replaced by that of the LLM, and vice versa. This approach ensures both modules are progressively optimized in a unified direction using a shared codebook. However, a new challenge arises from the disparity in the codebook change rate between the two modules. Specifically, the codebook change in the visual tokenizer is significantly greater than that of the LLM. This issue becomes even more severe for stacked quantization due to their multi-layer code map, where each additional layer requires one more update of the codebook. Such disparity ultimately leads to misalignment between the codebook and the LLM, impairing the LLM's language generation capabilities.

Language-driven Iterative Training. To overcome the above issues and facilitate unified codebook learning, we introduce this paradigm as illustrated in Figure 2 (c). Unlike dual alternative training, this approach does not employ the visual tokenizer to update the LLM's codebook. Instead, we apply the exponential moving average (EMA) method [27] to ensure the codebook's alignment with the visual encoder, dynamically updating the visual tokenizer's codebook at a certain decay rate λ:

$$\mathbb{C}^{'} = \lambda\mathbb{C} + (1-\lambda)\mathbb{I}\cdot Z. \tag{2}$$

$Z \in \mathbb{R}^{hw\times c}$ represents the flattened features of a given image, as generated by the encoder. The indicator map $\mathbb{I} \in \mathbb{R}^{\mathrm{K}\times hw}$ summarizes the usage of each code in the codebook $\mathbb{C}$ within the feature map Z. Crucially, at regular intervals, we integrate the codebook $\mathbb{C}_L$ in the LLM to replace $\mathbb{I}\cdot Z$ to update $\mathbb{C}$:

$$\mathbb{C}^{'} = \lambda\mathbb{C} + (1-\lambda)\mathbb{C}_L. \tag{3}$$

Equation 3 ensures the gradual convergence of the visual tokenizer's codebook towards $\mathbb{C}_L$ during training. Our paradigm not only aids in the efficient acquisition of a unified codebook, but also ensures that the training of the LLM remains undisturbed by the updates in the visual tokenizer. Note that our paradigm is adaptable to various LLM-tuning approaches, including full parameter tuning, LoRA [22], or even freezing the LLM. A significant distinction of our approach, compared to other MLLMs, is that it does not need additional modules for visual-text alignment. We believe this could be an alternative for unified MLLMs, especially considering the recent breakthrough of visual sequential modeling [3].

3.3 In-context Image Decompression

Since we adopt the stacked quantization introduced in Section 3.1 to represent images with fewer tokens, UniCode encounters a misalignment issue when aggregating word embeddings with the LLM, which can hinder the learning of semantically meaningful tokens. To tackle this issue, we propose an image decompression pretraining task, as shown in Figure 3. The objective is to reconstruct the multi-layer code map $\hat{M}$ by feeding the LLM with the aggregated quantized embeddings $\hat{Z}$. Initially, $\hat{Z}$ is processed into a flattened sequence of length $\hat{h} \times \hat{w}$. We then define the target sequence as $\{u_1, u_2, ..., u_{\hat{h}\times\hat{w}\times D}\}$, which is derived from $\hat{M}$, where each $u_l \in \hat{M}$. Our goal is to maximize the likelihood of generation in an auto-regressive manner:

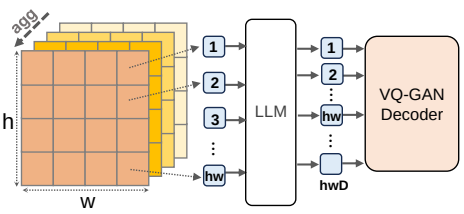

Fig. 3: Illustration of the procedure for the in-context image decompression task, which accepts the compressed quantized embeddings $\hat{Z} \in \mathbb{R}^{\hat{h}\times\hat{w}}$ as inputs, and then proceeds to transform these embeddings into their flattened codes $\hat{M} \in \mathbb{R}^{\hat{h}\times\hat{w}\times D}$ that are subsequently used for visual decoding.

$$\max_{\theta} \sum_{l=1}^{\hat{h}\times\hat{w}\times D} \log P_{\Theta}(u_l | u_{<l}; \hat{Z}), \tag{4}$$

where Θ denotes the trainable parameters of the LLM. Moreover, to enhance our model's capability in interpreting and generating across various modalities, we adopt a strategy similar to Liu *et al.* [33]. Specifically, we construct instruction-following pairs of multi-turn, conversation-style data for in-context learning:$\{\mathcal{X}_m^1, \mathcal{X}_z^1, ..., \mathcal{X}_m^T, \mathcal{X}_z^T\}$. Here, an image is segmented into T pieces, with $\mathcal{X}_z^t$ and $\mathcal{X}_m^t$ representing the quantized embeddings and their corresponding visual codes for each segment t. We organize these segments sequentially and consider each $\mathcal{X}_m^t$ as the response from the LLM. For a sequence of length L, the probability of generating the target codes $\mathcal{X}_m$ is computed as:

$$P(\mathcal{X}_m^t|X_z^t) = \prod_{i=1}^{L} P_\Theta(x_i|\mathcal{X}_m^{<i}, \mathcal{X}_z^{<i}), \tag{5}$$

where $\mathcal{X}_m^{<i}$ and $\mathcal{X}_z^{<i}$ denote the visual codes and their compressed quantized embeddings in all segments before the current prediction token x_i, respectively. We incorporate this task with our multimodal instruction tuning data, which mimics the misalignment between the compressed image embeddings and the LLM, encouraging our UniCode to generate images with higher quality.

4 Training

Following Liu *et al.* [32], we first leverage pairs of image-text data for multimodal instruction tuning. Specifically, we organize each instructional instance into a sequence of multi-round dialogues, represented as $\{\mathcal{X}_q^1, \mathcal{X}_a^1, \ldots, \mathcal{X}_q^N, \mathcal{X}_a^N\}$. In this sequence, each pair $\{\mathcal{X}_q^i, \mathcal{X}_a^i\}$ signifies a question-answer round between a human and the chatbot assistant, with N indicating the total number of dialogue rounds. This structured format is consistently applied throughout our instructional dataset. Additionally, with the advancement that allows images to be represented as discrete language tokens, our model is capable of converting text-to-image samples (e.g., CC3M [46]) into instruction-answer pairs. Lastly, we also prepare task-specific data using the same format for in-context image decompression as described in Section 3.3. We combine all the above data for multimodal instruction tuning. To train our model efficiently, we adopt a negative log-likelihood objective over the prediction tokens:

$$\mathcal{L}(\Theta) = -\sum_{j=1}^{L} \log P_\Theta(y_j|\mathcal{I}, \hat{y}_{1:j-1}). \tag{6}$$

Here, y and $\hat{y}$ represent the target and input token sequences, respectively, while Θ denotes the model parameters and L denotes the length of the target sequence. Depending on the specific instruction provided, the input visual content, represented as $\mathcal{I}$, may correspond to an empty image. A notable aspect is the restriction of loss computation exclusively to the answer tokens $\mathcal{X}_a$, designed to avoid oversimplifying the training process and to ensure that the model remains focused on generating accurate and coherent responses. We adopt a two-stage instruction-tuning process to train UniCode. It is important to note that our training process does not include the multimodal alignment stage, which is different from Liu *et al.* [33].

Stage I: Unified Codebook Learning. Our goal in this stage is to align the visual tokenizer with the LLM to share one codebook. We train the visual tokenizer through the image reconstruction task. There is no limitation on the type of training images, and in practice, more diverse and large-scale data can bring better performance to the visual tokenizer. To strike a balance between performance and efficiency, we only consider a limited scale in this work. For the

LLM, it requires textual instruction-answer data to enhance its ability to follow instructions [6]. We alternate the training process between these two modules, updating the codebook parameters using our language-driven iterative paradigm. After Stage 1, UniCode obtains a unified codebook that can simultaneously represent non-linguistic signals to achieve multimodal Input/Output (I/O).

Stage II: Multimodal Instruction Tuning. In the second stage, we keep the visual encoder and decoder frozen while exclusively fine-tuning the LLM. This stage fully utilizes the comprehensive multimodal instructional dataset, which augments the model's effectiveness in interpreting and responding to intricate multimodal instructions. The focus of this stage is on the model's ability to produce multimodal outputs, thereby significantly enriching its multimodal comprehension and response capabilities.

5 Experiments

To thoroughly evaluate the expansive multimodal capabilities of UniCode, we first conduct a series of ablation studies in Section 5.2. We then carry out comparison experiments across several key benchmarks: image generation (Section 5.3), image reconstruction (Section 5.4), and multimodal understanding (Section 5.5). Due to space limitations, more details, visualizations, and experimental results can be found in our appendix.

5.1 Implemented Details

During Stage I, we train the visual tokenizer on the LCS-558K dataset introduced by LLaVA [33]. We directly employ a pretrained LLM [6], opting not to conduct further instruction tuning on text-only data. It's worth mentioning that this stage is designed with the flexibility to extend and allow the LLM to undergo pretraining on an extensive text corpus with full parameter tuning. In Stage II, we focus on fine-tuning the LLM using a curated combined dataset, which includes Mixed-665K [33], the text-to-image dataset CC3M [46], and our specially tailored data for the in-context image decompression task.

Table 1: Comparisons of different paradigms for MLLMs on VQA and image generation benchmarks. Here, "tok" is used as an abbreviation for "tokenizer".

paradigm	VQA Benchmarks					Image Gen (FID ↓)		
	VQA^2	VizWiz	SQA	VQA^T	POPE	ImgNet	LSUN-cat	LSUN-church
vis enc+text tok	52.3	45.4	62.2	42.1	69.7	-	-	-
vis tok+text tok	49.0	44.5	56.7	37.8	65.4	9.82	10.28	10.78
unified tok	53.1	46.2	62.9	42.5	71.8	6.72	8.07	6.96

5.2 Ablation Study

Comparison of different paradigms for MLLMs. In Table 1, we compare three MLLM paradigms as depicted in Figure 1. Notably, the use of a unified

codebook (Row 3) achieves stable improvements in both VQA and image generation benchmarks compared to the separate use of visual and text tokenizers (Row 2). This can be attributed to the aligned distribution of shared tokens and the pretrained LLM, which also results in greater resource efficiency during both training and inference. Additionally, the unified codebook exhibits a slight improvement in VQA tasks compared to using "visual encoder + text tokenizer" (Row 1). Note that Row 1 lacks direct applicability to image generation. Instead, our unified codebook enables LLMs to produce multimodal outputs.

Comparison of different visual setups. UniCode employs a relatively lightweight visual encoder and is trained on a modest dataset of 558K images [32]. While efficient, this setup restricts its ability to extract comprehensive visual features and hinders its generalization to novel contexts. This limitation becomes more evident when compared to models like CLIP [41], which benefit from training on a vast collection of 400 million image-text pairs. In Table 2, we demonstrate that the limitation can be mitigated. By enriching the dataset with additional images from CC3M (Row 2) and evaluation ground truth (Row 3), UniCode manifests consistent improvements across various VQA benchmarks. Additional improvement can be obtained by replacing the visual encoder trained from scratch with a pretrained and advanced version (Row 4).

Table 2: Comparisons of different visual encoder setups, where "cc3m imgs" and "GT imgs" refer to using additional images from CC3M [46] and evaluation ground truth to train the visual encoder, "w/ ViT*" denotes using pretrained and larger ViT encoder [16] instead of training it from scratch.

Setup	VQA Benchmarks				
	VQA^2	VizWiz	SQA	VQA^T	POPE
UniCode	53.1	46.2	62.9	42.5	71.8
+ cc3m imgs	53.6	47.4	64.3	45.6	74.3
++ GT imgs	53.7	47.4	64.8	44.9	75.1
+++ w/ ViT*	56.2	47.1	65.4	47.3	77.6

In addition to the visual encoder, we also verify the influence of different visual tokenizers, as demonstrated in Table 3. It is crucial to note again that UniCode is designed to be compatible with a wide range of visual quantization approaches. Furthermore, we observe that as we keep upgrading the visual tokenizer (from Row 1 to Row 3), the performance of UniCode is also improved. These observations confirm that UniCode's overall capabilities can be continuously enhanced by continually improving the visual setup.

Table 3: Comparison of different visual tokenizers. A Better tokenizer brings better performance.

	VQA^2	VizWiz	SQA	VQA^T	POPE
VQ-GAN [15]	49.1	42.6	60.8	41.2	65.1
RQ-VAE [27]	49.8	44.0	61.5	41.6	67.5
HQ-VAE [59]	53.1	46.2	62.9	42.5	71.8

Table 4: Comparisons of different paradigms for learning a unified codebook.

paradigm	VQA Benchmarks					Image Gen (FID ↓)		
	VQA^2	VizWiz	SQA	VQA^T	POPE	ImgNet	LSUN-cat	LSUN-church
frozen	44.2	35.1	56.8	36.3	63.9	34.45	33.84	34.26
dual	9.3	5.2	11.2	8.5	13.2	8.87	9.76	9.54
iter	53.1	46.2	62.9	42.5	71.8	6.72	8.07	6.96

Comparison of different paradigms to learn the unified codebook. We include the relevant results in Table 4. The dual alternative training (dual) results in a performance collapse, particularly in VQA benchmarks. This issue arises from a disruption in the consistency between the LLM architecture and the codebook, as discussed in Section 3.2. In addition, our paradigm (iter) produces representative visual tokens, leading to notable improvements in visual generation compared to the frozen LLM codebook (frozen).

Effect of in-context image decompression task. The results in Table 5 demonstrate that our pretraining task clearly enhances the visual generation quality of our model across various configurations: class-conditioned (ImageNet), text-conditioned (CC3M), and unconditioned (LSUN-Cat). We posit that this enhancement in performance can be attributed to the pretraining task's ability to prevent premature convergence. It achieves this by escalating the complexity of the training process and enriching the diversity of the training samples.

Table 5: Ablation of in-context image decompression task ("ImgDe") on image generation tasks. We use FID as the metric.

Method	ImageNet	CC3M	LSUN-Cat
w/o ImgDe	7.08	11.91	8.53
w/ ImgDe	6.72	11.54	8.07

Effect of different code map resolution. In Table 6. spanning Column 1-5, UniCode is pretrained with images of resolution 256×256, and reaches optimal performance when tested at this identical resolution. Notably, there is a marked decrease in performance when test resolutions are increased beyond this point, even though these larger resolutions do not exceed the LLM's token length capacity. We deduce that this drop in performance stems from a misalignment between training and testing conditions. Specifically, testing with resolutions significantly larger than those used in training creates a disparity in how each element of the code map represents image areas. In Column 6, we verify this hypothesis by pretraining UniCode using 320×320 images (320*), and the results of our model are improveed to our expectation.

Table 6: Ablation of different code map resolutions on VQA benchmarks.

	192	256	320	384	Raw	320*
VQA^2	38.2	53.1	41.3	42.6	36.4	54.5
VizWiz	42.8	46.2	43.9	41.3	39.1	47.1
SQA^I	61.1	62.9	63.7	62.0	59.2	63.8

Table 7: Comparison of FIDs for unconditioned image generation on LSUN-{Cat, Bedroom, Church} [60].

Method	FID ↓		
	Cat	Bedroom	Church
ImageBART	15.09	4.90	7.89
StyleGAN2 [25]	7.25	2.35	3.86
VQ-GAN	17.31	6.35	7.81
RQ-Transformer	8.64	3.04	7.45
HQ-TVAE*	8.35	2.89	7.12
HQ-UniCode	8.07	2.65	6.96

Table 8: Comparison of FIDs and CLIP score for text-conditioned image generation on CC3M validation set.

Method	Params	FID ↓	CLIP-s ↑
ImageBART	2.8B	22.61	0.23
LDM [44]	645M	17.01	0.24
VQ-GAN	1.5B	28.86	0.20
RQ-Transformer	654M	12.33	0.26
HQ-TVAE	579M	12.86	0.26
HQ-TVAE*	7B	12.13	0.28
HQ-UniCode	7B	11.54	0.30

5.3 Comparison on Image Generation

We first assess the capability of our model in unconditioned image generation in Table 7, utilizing three subsets of the LSUN dataset. Initially, we combine ImageNet with the LCS-558K dataset to pretrain our visual tokenizer, then finetune the model for another one epoch on the downstream dataset. Given the extensive size of the dataset, we opt for LoRA [23] to finetune LLM to avoid overfitting. Due to the lack of training details for HQ-TVAE, we have implemented its 7B version (HQ-TVAE*) and compare it with our model (HQ-UniCode) for a fair comparison based on the same parameter setup. Our model performs clearly better than HQ-TVAE*. Furthermore, we carry out experiments on text-conditioned image generation in Table 8 and class-conditioned generation in Table 9, UniCode obtains similar improvements on these two benchmarks. As can be seen, the improved results demonstrate the benefit of employing a unified codebook to enhance visual generation, especially considering that our reconstruction quality is suboptimal compared with original HQ as discussed in Section 5.4. We attribute this benefit to the alignment between the unified codebook and LLM's textual space. Lastly, we present some qualitative examples as shown in Figure 4. More visualization cases can be seen in our appendix.

Table 9: Comparisons of FIDs and ISs for class-conditioned image generation on ImageNet [10].

Method	Params	FID ↓	IS ↑
ADM [13]	554M	10.94	101.0
ImageBART [14]	3.5B	21.19	61.6
VQ-Diffusion [19]	370M	11.89	
VQ-VAE-2 [43]	13.5B	≈ 31	≈ 45
VQ-GAN [15]	1.4B	15.78	74.3
RQ-Transformer [27]	3.8B	7.55	134.0
HQ-TVAE [59]	1.4B	7.15	-
HQ-TVAE*	7B	7.04	171.4
HQ-UniCode	7B	6.72	208.9

5.4 Comparison on Image Reconstruction

Table 10 validates the reconstruction quality of the visual tokenizer of our model. Such validation is crucial to ensure that the tokenizer preserves essential semantics after visual quantization. In this table, we observe that multi-layer stacking, as a form of stacked quantization structure, substantially boosts the model's ability to efficiently represent images. However, this benefit comes with a trade-off: an increased number of layers significantly lengthens the sequence, posing a greater challenge for decoding by LLM. In comparison with HQ or RQ, the reconstruction quality of UniCode, when using the unified codebook, is nearly on par, indicating that our learning paradigm for the unified codebook

Table 10: Comparison of reconstruction quality on ImageNet and LCS-558K datasets, according to their codebook size (K), resolution (Res), number of used layers and tokens.

Tokenizer	Res	Layers: Tokens	K	rFID ↓	
				Imagenet	LCS-558K
VQ-GAN	64	1:64	16384	17.95	23.83
VQ-GAN	256	1:256	16384	4.9	11.26
SPAE [62]	256	5:341	16384	9.49	-
SPAE	256	6:597	16384	4.41	-
RQ-VAE	64	4:256	32000	6.82	12.09
HQ-VAE	256	2:320	32000	2.61	8.35
RQ-UniCode	64	8:512	32000	3.78	9.33
HQ-UniCode	256	2:320	32000	2.83	7.91

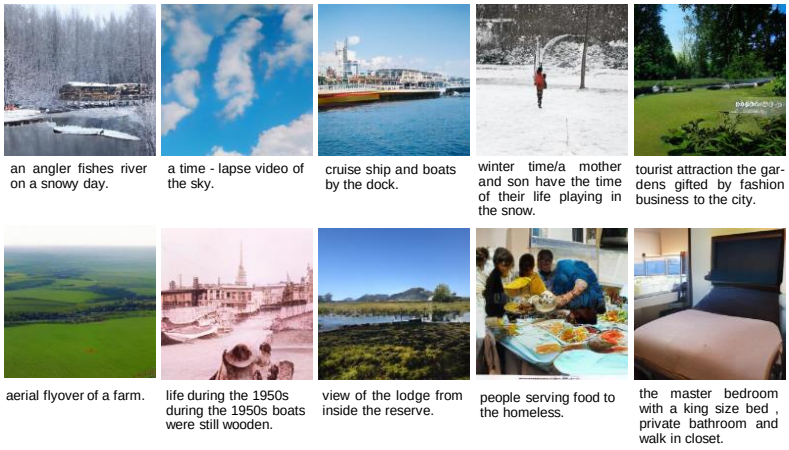

Fig. 4: Qualitative examples of text-conditioned image generation on CC3M.

does not significantly damage VAE training. In Figure 5, we present qualitative examples of image reconstruction on LCS-558K [32]. When compared to ImageNet, there is a significant decline in reconstruction quality on LCS-558K, probably attributed to LCS-558K's more diverse scenes.

5.5 Comparison on Multimodal Understanding

We first carry out experiments on a diverse set of seven benchmarks in Table 11, including VQA-v2 (VQA^{v2}) [18], VizWiz [21], ScienceQA-IMG (SQA^{I}) [37], TextVQA (VQA^{T}) [47], POPE [30], MMB [34] and MMB^{CN}. Experimental results on more benchmarks are provided in our appendix. It is encouraging that our model performs considerably well even with the smallest scale of training data and fewer parameters. UniCode outperforms many recently proposed MLLMs in several benchmarks. More importantly, it obtains stable improvement on both VQA^{v2} and VizWiz benchmarks when compared to another multimodal generation model Emu [48]. Through these experiments, we validate the feasibility of a unified codebook as an alternative paradigm for multimodal generative models. When compared to the current state-of-the-art model LLaVA-1.5, UniCode shows significant performance vari-

Fig. 5: Qualitative examples of image reconstruction generated by our proposed UniCode. Their raw images can be seen in the appendix.

Table 11: Comparison with MLLMs on VQA benchmarks. UniCode outperforms another multimodal generation model Emu. It achieves competitive results against other methods while requiring less data and fewer parameters for its visual tokenizer. Here, "M2T" and "M2M" refer to the model's capability to generate either text only or multiple modalities. "Vis-P", "PT" and "IT" represent the number of parameters in the visual encoder, the number of samples for multimodal alignment, and instruction tuning, respectively. Results on more benchmarks are provided in the appendix.

Method	Type	LLM	Vis-P	PT	IT	VQA^{v2}	VizWiz	SQA^{I}	VQA^{T}	POPE	MMB	MMB^{CN}
BLIP-2 [29]	M2T	Vicuna-13B	303M	129M	0	41.0	19.6	61	42.5	85.3	-	-
InstructBLIP [9]	M2T	Vicuna-7B	303M	129M	1.2M	-	34.5	60.5	50.1	-	36	23.7
Qwen-VL [2]	M2T	Qwen-7B	1.8B	1.4B	50M	78.8	35.2	67.1	63.8	-	38.2	
Emu [48]	M2M	LLaMA-13B	1B	82M	240K	52.0	34.2	-	-	-	-	-
Emu-I [48]	M2M	LLaMA-13B	1B	82M	240K	40.0	35.4	-	-	-	-	-
LLaVA-1.5	M2T	Vicuna-7B	303M	558K	665K	79.1	47.8	68.4	58.2	86.4	64.3	58.3
UniCode	M2M	Vicuna-7B	104M	0	665K	53.1	46.2	62.9	42.5	71.8	33.7	25.5
UniCode+	M2M	Vicuna-7B	1B	0	665K	56.2	47.1	65.4	47.3	77.6	37.2	29.1

ations across different benchmarks. For example, UniCode's performance is competitive with LLaVA-1.5 in the VQA^{T} and SQA^{I} benchmarks. However, it lags significantly (nearly 20%) behind in the POPE [30] benchmark. We speculate that this is likely due to the insufficient training data provided for the visual tokenizer, which leads to the limitation of the tokenizer in terms of generalization.

UniCode initially employs a lightweight visual tokenizer, which, due to limitations in resolution, training data, and the scale of parameters, results in suboptimal performance. To address these shortcomings, we have developed an enhanced variant, referred to as 'UniCode+'. UniCode+ incorporates a more substantial dataset for training and integrating a pretrained and larger ViT encoder as detailed in Table 2. As demonstrated in Table 11, UniCode+ significantly outperforms the original UniCode across all VQA benchmarks. This improvement underscores the potential for elevating model performance through the adoption of a more sophisticated visual encoder.

6 Conclusion

We introduce UniCode, a pioneering effort in the Multimodal Language Learning Model (MLLM) field to create a unified codebook for both visual and textual tokenization. UniCode innovates with a language-driven iterative training paradigm and an in-context image decompression task, enabling the unified codebook to facilitate multimodal instruction tuning for non-linguistic generation tasks. Our comprehensive experiments in multimodal understanding and generation, coupled with an extensive ablation study, position UniCode as a promising new approach for advancing research within the MLLM community.

Acknowledgments

This work was supported by NSFC under grant 62250068.

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in Neural Information Processing Systems **35**, 23716–23736 (2022)
2. Bai, J., Bai, S., Yang, S., Wang, S., Tan, S., Wang, P., Lin, J., Zhou, C., Zhou, J.: Qwen-vl: A frontier large vision-language model with versatile abilities. arXiv preprint arXiv:2308.12966 (2023)
3. Bai, Y., Geng, X., Mangalam, K., Bar, A., Yuille, A., Darrell, T., Malik, J., Efros, A.A.: Sequential modeling enables scalable learning for large vision models. arXiv preprint arXiv:2312.00785 (2023)
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
5. Chang, H., Zhang, H., Jiang, L., Liu, C., Freeman, W.T.: Maskgit: Masked generative image transformer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11315–11325 (2022)
6. Chiang, W.L., Li, Z., Lin, Z., Sheng, Y., Wu, Z., Zhang, H., Zheng, L., Zhuang, S., Zhuang, Y., Gonzalez, J.E., et al.: Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. See https://vicuna. lmsys. org (accessed 14 April 2023) (2023)
7. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311 (2022)
8. Cui, Y., Yang, Z., Yao, X.: Efficient and effective text encoding for chinese llama and alpaca. arXiv preprint arXiv:2304.08177 (2023)
9. Dai, W., Li, J., Li, D., Tiong, A.M.H., Zhao, J., Wang, W., Li, B., Fung, P., Hoi, S.: Instructblip: Towards general-purpose vision-language models with instruction tuning. arXiv preprint arXiv:2305.06500 (2023)
10. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
11. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
12. Dhariwal, P., Jun, H., Payne, C., Kim, J.W., Radford, A., Sutskever, I.: Jukebox: A generative model for music. arXiv preprint arXiv:2005.00341 (2020)
13. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. Advances in neural information processing systems **34**, 8780–8794 (2021)
14. Esser, P., Rombach, R., Blattmann, A., Ommer, B.: Imagebart: Bidirectional context with multinomial diffusion for autoregressive image synthesis. Advances in neural information processing systems **34**, 3518–3532 (2021)
15. Esser, P., Rombach, R., Ommer, B.: Taming transformers for high-resolution image synthesis. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12873–12883 (2021)
16. Fang, Y., Wang, W., Xie, B., Sun, Q., Wu, L., Wang, X., Huang, T., Wang, X., Cao, Y.: Eva: Exploring the limits of masked visual representation learning at scale. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 19358–19369 (2023)

17. Feng, Y., Wang, Y., Liu, J., Zheng, S., Lu, Z.: Llama rider: Spurring large language models to explore the open world. arXiv preprint arXiv:2310.08922 (2023)
18. Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D.: 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. pp. 6904–6913 (2017)
19. Gu, S., Chen, D., Bao, J., Wen, F., Zhang, B., Chen, D., Yuan, L., Guo, B.: Vector quantized diffusion model for text-to-image synthesis. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10696–10706 (2022)
20. Gupta, A., Tian, S., Zhang, Y., Wu, J., Martín-Martín, R., Fei-Fei, L.: Maskvit: Masked visual pre-training for video prediction. arXiv preprint arXiv:2206.11894 (2022)
21. Gurari, D., Li, Q., Stangl, A.J., Guo, A., Lin, C., Grauman, K., Luo, J., Bigham, J.P.: Vizwiz grand challenge: Answering visual questions from blind people. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3608–3617 (2018)
22. Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
23. Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., Chen, W.: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 (2021)
24. Juang, B.H., Gray, A.: Multiple stage vector quantization for speech coding. In: ICASSP'82. IEEE International Conference on Acoustics, Speech, and Signal Processing. vol. 7, pp. 597–600. IEEE (1982)
25. Karras, T., Laine, S., Aittala, M., Hellsten, J., Lehtinen, J., Aila, T.: Analyzing and improving the image quality of stylegan. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8110–8119 (2020)
26. Koh, J.Y., Salakhutdinov, R., Fried, D.: Grounding language models to images for multimodal generation. arXiv preprint arXiv:2301.13823 (2023)
27. Lee, D., Kim, C., Kim, S., Cho, M., Han, W.S.: Autoregressive image generation using residual quantization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11523–11532 (2022)
28. Li, B., Zhang, Y., Chen, L., Wang, J., Pu, F., Yang, J., Li, C., Liu, Z.: Mimic-it: Multi-modal in-context instruction tuning. arXiv preprint arXiv:2306.05425 (2023)
29. Li, J., Li, D., Savarese, S., Hoi, S.: Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. arXiv preprint arXiv:2301.12597 (2023)
30. Li, Y., Du, Y., Zhou, K., Wang, J., Zhao, W.X., Wen, J.R.: Evaluating object hallucination in large vision-language models. arXiv preprint arXiv:2305.10355 (2023)
31. Liu, H., Yan, W., Abbeel, P.: Language quantized autoencoders: Towards unsupervised text-image alignment. arXiv preprint arXiv:2302.00902 (2023)
32. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. arXiv preprint arXiv:2310.03744 (2023)
33. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. arXiv preprint arXiv:2304.08485 (2023)
34. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? arXiv preprint arXiv:2307.06281 (2023)

35. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017)
36. Lu, J., Clark, C., Lee, S., Zhang, Z., Khosla, S., Marten, R., Hoiem, D., Kembhavi, A.: Unified-io 2: Scaling autoregressive multimodal models with vision, language, audio, and action. arXiv preprint arXiv:2312.17172 (2023)
37. Lu, P., Mishra, S., Xia, T., Qiu, L., Chang, K.W., Zhu, S.C., Tafjord, O., Clark, P., Kalyan, A.: Learn to explain: Multimodal reasoning via thought chains for science question answering. Advances in Neural Information Processing Systems **35**, 2507–2521 (2022)
38. Martinez, J., Hoos, H.H., Little, J.J.: Stacked quantizers for compositional vector compression. arXiv preprint arXiv:1411.2173 (2014)
39. OpenAI: Chatgpt. https://openai.com/blog/chatgpt (2022)
40. Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al.: Training language models to follow instructions with human feedback. Advances in Neural Information Processing Systems **35**, 27730–27744 (2022)
41. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
42. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. The Journal of Machine Learning Research **21**(1), 5485–5551 (2020)
43. Razavi, A., Van den Oord, A., Vinyals, O.: Generating diverse high-fidelity images with vq-vae-2. Advances in neural information processing systems **32** (2019)
44. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 10684–10695 (2022)
45. Shannon, C.E., et al.: Coding theorems for a discrete source with a fidelity criterion. IRE Nat. Conv. Rec **4**(142-163), 1 (1959)
46. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2556–2565 (2018)
47. Singh, A., Natarajan, V., Shah, M., Jiang, Y., Chen, X., Batra, D., Parikh, D., Rohrbach, M.: Towards vqa models that can read. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8317–8326 (2019)
48. Sun, Q., Yu, Q., Cui, Y., Zhang, F., Zhang, X., Wang, Y., Gao, H., Liu, J., Huang, T., Wang, X.: Generative pretraining in multimodality. arXiv preprint arXiv:2307.05222 (2023)
49. Taori, R., Gulrajani, I., Zhang, T., Dubois, Y., Li, X., Guestrin, C., Liang, P., Hashimoto, T.B.: Stanford alpaca: An instruction-following llama model (2023)
50. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023)
51. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023)
52. Van Den Oord, A., Vinyals, O., et al.: Neural discrete representation learning. Advances in neural information processing systems **30** (2017)

53. Wang, C., Xu, C., Wang, C., Tao, D.: Perceptual adversarial networks for image-to-image transformation. IEEE Transactions on Image Processing **27**(8), 4066–4079 (2018)
54. Wang, J., Yang, Z., Hu, X., Li, L., Lin, K., Gan, Z., Liu, Z., Liu, C., Wang, L.: Git: A generative image-to-text transformer for vision and language. arXiv preprint arXiv:2205.14100 (2022)
55. Xu, Z., Shen, Y., Huang, L.: Multiinstruct: Improving multi-modal zero-shot learning via instruction tuning. arXiv preprint arXiv:2212.10773 (2022)
56. Yan, W., Hafner, D., James, S., Abbeel, P.: Temporally consistent transformers for video generation. arXiv preprint arXiv:2210.02396 (2022)
57. Yan, W., Zhang, Y., Abbeel, P., Srinivas, A.: Videogpt: Video generation using vq-vae and transformers. arXiv preprint arXiv:2104.10157 (2021)
58. Ye, Q., Xu, H., Xu, G., Ye, J., Yan, M., Zhou, Y., Wang, J., Hu, A., Shi, P., Shi, Y., et al.: mplug-owl: Modularization empowers large language models with multimodality. arXiv preprint arXiv:2304.14178 (2023)
59. You, T., Kim, S., Kim, C., Lee, D., Han, B.: Locally hierarchical auto-regressive modeling for image generation. Advances in Neural Information Processing Systems **35**, 16360–16372 (2022)
60. Yu, F., Seff, A., Zhang, Y., Song, S., Funkhouser, T., Xiao, J.: Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. arXiv preprint arXiv:1506.03365 (2015)
61. Yu, J., Li, X., Koh, J.Y., Zhang, H., Pang, R., Qin, J., Ku, A., Xu, Y., Baldridge, J., Wu, Y.: Vector-quantized image modeling with improved vqgan. arXiv preprint arXiv:2110.04627 (2021)
62. Yu, L., Cheng, Y., Wang, Z., Kumar, V., Macherey, W., Huang, Y., Ross, D.A., Essa, I., Bisk, Y., Yang, M.H., et al.: Spae: Semantic pyramid autoencoder for multimodal generation with frozen llms. arXiv preprint arXiv:2306.17842 (2023)
63. Yuan, L., Chen, D., Chen, Y.L., Codella, N., Dai, X., Gao, J., Hu, H., Huang, X., Li, B., Li, C., et al.: Florence: A new foundation model for computer vision. arXiv preprint arXiv:2111.11432 (2021)
64. Zhang, S., Roller, S., Goyal, N., Artetxe, M., Chen, M., Chen, S., Dewan, C., Diab, M., Li, X., Lin, X.V., et al.: Opt: Open pre-trained transformer language models. arXiv preprint arXiv:2205.01068 (2022)
65. Zhang, Y., Zhang, R., Gu, J., Zhou, Y., Lipka, N., Yang, D., Sun, T.: Llavar: Enhanced visual instruction tuning for text-rich image understanding. arXiv preprint arXiv:2306.17107 (2023)
66. Zhao, B., Wu, B., Huang, T.: Svit: Scaling up visual instruction tuning. arXiv preprint arXiv:2307.04087 (2023)
67. Zheng, S., Liu, J., Feng, Y., Lu, Z.: Steve-eye: Equipping llm-based embodied agents with visual perception in open worlds. arXiv preprint arXiv:2310.13255 (2023)