# VoCo-LLaMA: Towards Vision Compression with Large Language Models

Xubing Ye[1], Yukang Gan[2], Xiaoke Huang[3], Yixiao Ge[2*], Yansong Tang[1*]
[1]Tsinghua Shenzhen International Graduate School, Tsinghua University
[2]ARC Lab, Tencent PCG [3]UC Santa Cruz
{yxb23@mails.,tang.yansong@sz.}tsinghua.edu.cn
{brucegan,yixiaoge}@tencent.com xhuan192@ucsc.edu

## Abstract

*Vision-Language Models (VLMs) have achieved remarkable success in various multi-modal tasks, but they are often bottlenecked by the limited context window and high computational cost of processing high-resolution image inputs and videos. Vision compression can alleviate this problem by reducing the vision token count. Previous approaches compress vision tokens with external modules and force LLMs to understand the compressed ones, leading to visual information loss. However, the LLMs' understanding paradigm of vision tokens is not fully utilised in the compression learning process. We propose VoCo-LLaMA, the first approach to compress vision tokens using LLMs. By introducing **Vi**sion **Co**mpression tokens during the vision instruction tuning phase and leveraging attention distillation, our method distill how LLMs comprehend vision tokens into their processing of VoCo tokens. VoCo-LLaMA facilitates effective vision compression and improves the computational efficiency during the inference stage. Specifically, our method can achieve a 576× compression rate while maintaining 83.7% performance. Furthermore, through continuous training using time-series compressed token sequences of video frames, VoCo-LLaMA demonstrates the ability to understand temporal correlations, outperforming previous methods on popular video question-answering benchmarks. Our approach presents a promising way to unlock the full potential of VLMs' contextual window, enabling more scalable multi-modal applications.*

## 1. Introduction

The advent of visual-language models [3, 5, 13, 25, 29, 30, 32, 47, 59, 60] has led to significant advancements in visual understanding. Particularly, high-resolution image encoding [5, 29] and the incorporation of more video frames [32, 47] have been shown to enhance the capabilities of both large visual-language models and large video-language models, respectively. However, the large number of vision tokens occupies a substantial portion of the valuable context window of the large language model, leading to expensive computational costs. For instance, when using high-resolution image inputs in LLaVA-1.6 [29], a single image with a resolution of 672 × 672 is divided into smaller patches, each encoded with a 336 × 336 resolution input. This process yields an image representation consisting of 2880 vision tokens, occupying over half of the context length. As the number of input images increases, the context window for text will be further bottle-

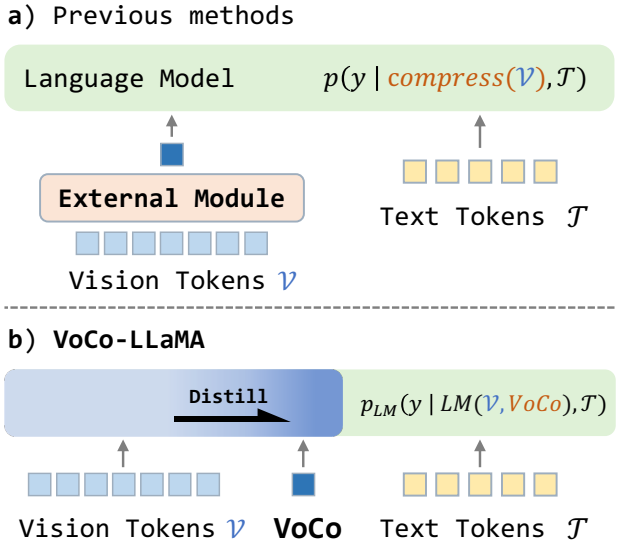


Figure 1. (a) Previous methods exploit external module, such as Q-Former [25] or average pooling [28], to "compress" vision tokens with substantial loss. (b) Illustration of VoCo-LLaMA, which empowers LLM to compress vision tokens and understand compressed tokens via intrinsic token distillation.

Work was done when the author interned at ARC Lab, Tencent PCG.
*Corresponding author.

necked. [32, 47] investigate the efficacy of extending the context length to the million-level mark to mitigate this issue, but this approach necessitates expensive computational resources (*e.g.,* [32] requires over 1000 v4 TPUs) and engineering efforts in data and framework development.

To address this issue, previous methods [11, 13, 25, 28, 59, 60] have exploited Q-Former [25] or Re-sampler [1] to "compress" the encoded vision tokens. As illustrated in Fig. 1 (a), these kind of methods compress the vision tokens with **external** modules and force LLMs to understand the compressed ones.

Given that the LLM can effectively understand uncompressed vision tokens [31], it has great potential to perform token compression on its own. Therefore, we propose **VoCo-LLaMA**, the first vision compression method that leverages the inherent capabilities of large language models to our best knowledge. As demonstrated in Fig. 1 (b), we introduce **Vis**ion **Co**mpression (**VoCo**) tokens between visual and text tokens. By modifying the attention mechanism, we ensure that VoCo tokens attend exclusively to visual tokens, while text tokens attend solely to VoCo tokens. Subsequently, we establish an exclusive interaction pathway between the visual and text tokens via VoCo tokens. This facilitates the LLM itself to compress and distill the parsing vision tokens, specifically the transformer activations on top of them, into compact VoCo tokens.

Building upon this, we further investigate the efficacy of VoCo-LLaMA in handling video input. The total number of visual tokens for each video can be substantial, far exceeding the context length of large language models (LLMs), making it impractical to utilize VoCo-LLaMA to compress all the tokens simultaneously. To address this issue, we first employ VoCo-LLaMA to compress the visual tokens of each frame into voco tokens. These voco tokens are subsequently concatenated to form a sequential token series. VoCo-LLaMA then extracts both visual and temporal information from this series to facilitate video understanding tasks. With this effective design, the LLMs can handle much longer videos within the same context length.

During inference, VoCo-LLaMA mitigates the issue of limited context length in LLM by employing a two-stage forward process. The fist stage compresses visual tokens of each image into a reduced set of VoCo tokens, while the second stage completes the task by utilizing both VoCo tokens and text tokens. Moreover, VoCo tokens can be cached and reused when handling various tasks involving identical visual inputs, thereby enhancing computational efficiency and reducing storage requirements compared to maintaining the entire KV-cache for uncompressed visual tokens. Experimental results on various benchmarks demonstrate that VoCo-LLaMA achieves a 576x compression rate while maintaining 83.7% of the original performance. Additionally, significant reductions in inference computation costs were observed, including up to 99.8% in cache storage, 94.8% in FLOPs, and 69.6% in inference time.

Our core contributions are summarized as follows:

- We propose VoCo-LLaMA, the first approach to compress vision tokens by leveraging the inherent capabilities of large language models, thereby eliminating the need for any external modules.
- We extend VoCo-LLaMA from image input to video input, which allows the LLM to handle approximately 200 times more video frames while maintaining its video understanding capabilities.
- Extensive experiments on image and video benchmarks demonstrate the effectiveness of our method, showcasing superior performance in both token compression and inference efficiency compare to various existing baselines.

## 2. Related Work

**LLMs and Text Compression.** In recent years, large language models (LLMs) have sparked a technological revolution. As the scale of training data and model size continue to expand, models [6, 10, 20, 22, 41, 48, 49, 51] have demonstrated exceptional capabilities in understanding and generating language. In particular, models such as the LLaMA series [10, 20, 48, 49] have emerged as foundational models or main components in many research works. However, the limited context window size in LLMs has long been a widely discussed topic. Text compression has been proven to be an efficient approach. Long-standing works, including [12, 33, 44, 52, 57], focus on storing text representations in transformers to achieve dense information representation. [2, 46] have demonstrated the effectiveness of distilling long text information into prompt-free student models. In a similar vein, recent studies [9, 16, 40, 50] have explored the potential applications of compressing text in large language models. However, the discussion of compressing visual information has been relatively understudied compared to the language model domain. Our work pioneers the use of LLMs' learning capabilities to compress vision information, aiming to bridging this gap in the field of VLMs.

**VLMs and Vision Compression.** The success of LLMs has inspired significant progress in vision language models (VLMs). By integrating visual encoders with LLMs, VLMs can effectively achieve cross-modal understanding through instruction tuning. Previous methods [1, 3, 5, 11, 13, 25, 29, 30, 59, 60] have substantiated the success of this training paradigm in visual understanding. The successful application of VLMs on images has also been rapidly extended to the video domain [19, 23, 26, 28, 32, 34, 37, 39, 47, 58]. With the input of higher-resolution images [5, 29] and more video frames [32, 47], VLMs can capture rich visual information. However, as the number of vision tokens representing an input image increases, they take up a significant portion of the limited context window of lan-

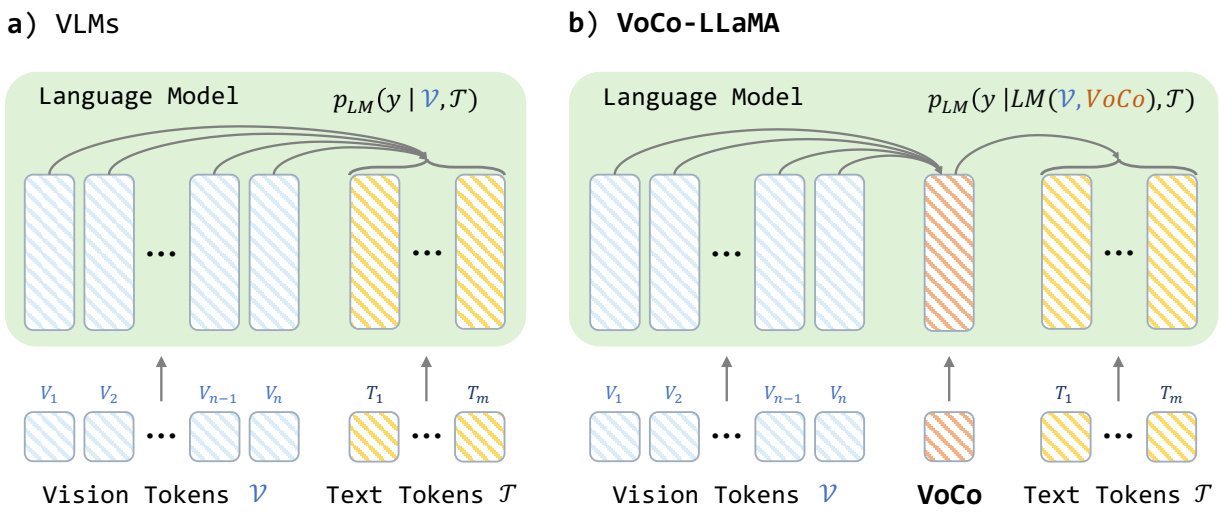


Figure 2. Illustration of the VoCo-LLaMA framework. Based on standard VLMs (a), VoCo-LLaMA (b) first isolate visual and text tokens by injecting VoCo tokens, and then establishes a dedicated interaction pathway between the two modalities via VoCo tokens, enabling effective compression of vision tokens into the transformer activations upon the compact VoCo tokens.

guage models, and can even exceed it. To address this, previous methods [11, 13, 25, 59, 60] have largely employed Q-Former [25], which maps images to fixed-length tokens in the language embedding space through learnable queries, compressing visual information. A more recent approach [28] has applied average pooling with a learnable linear layer to compress visual features through multi-stage training strategy. Although these methods perform moderately well at lower compression multiples, they cause a significant loss of valuable visual information when the number of compressed tokens reduces. VoCo-LLaMA distills the approach of LLMs in understanding vision tokens into their processing of compressed tokens, significantly reducing information loss during the vision compression process.

## 3. Method

We first introduce VoCo-LLaMA, a large language model capable of compressing lengthy vision tokens into compact VoCo tokens through attention distillation, which enables the efficient representation of visual information. Then, we build upon these compressed tokens to continue training VoCo-LLaMA, enabling our model to capture temporal dependencies within video data.

### 3.1. Vision Compression

Given a paired image and text input, we follow the design of most vision-language models (VLMs) and encode the image into a sequence of vision tokens $\mathcal{V} = \{V_1, \ldots, V_n\}$, where $n$ is the number of the output patches from the visual encoder. Similarly, the text input is encoded into a sequence of text tokens $\mathcal{T} = \{T_1, \ldots, T_m\}$. Consider an original, unmodified standard large vision language model (denoted as $LM_o$), exemplified by LLaVA [30], depicted in Fig. 2 (a). During visual instruction tuning, $LM_o$ leverages both vision tokens $\mathcal{V}$ and text tokens $\mathcal{T}$ to predict the output $y$, and learns the distribution $p_{LM_o}(y \mid \mathcal{V}, \mathcal{T})$. For image compression models, our goal is to employ a compact set of compressed tokens $\mathcal{C}$ to efficiently represent the vision token set $\mathcal{V}$. Additionally, we aim to generate outputs that closely approximates the outputs of the original model $LM_o$ when presented with identical image and text inputs.

With an image encoded as vision tokens $\mathcal{V}$, we formulate the image compression distillation process as learning a compression model $LM_c$ that generates the output $y$ conditioned on the compressed tokens $\mathcal{C}$ and the text tokens $\mathcal{T}$. This is achieved by learning a probability distribution $p_{LM_c}(y \mid \mathcal{C}, \mathcal{T})$. The optimization objective of $LM_c$ is to minimize the loss function:

$$E_{\mathcal{V},\mathcal{T}}[D_{KL}(p_{LM_o}(y \mid \mathcal{V}, \mathcal{T}) \parallel p_{LM_c}(y \mid \mathcal{C}, \mathcal{T}))] \quad (1)$$

With above distillation objective, how to further distill the information within the vision tokens $\mathcal{V}$ into the compressed token $\mathcal{C}$ is the key of vision compression.

### 3.2. VoCo-LLaMA

As illustrated in Fig. 2 (b), VoCo-LLaMA leverages the LLM's ability to compress visual tokens into compact **Vis**ion **Co**mpression (VoCo) tokens and learns to understand image through these VoCo tokens. The input sequence to the large language model is formed by concatenating the vision tokens, the special $VoCo$ tokens, and the

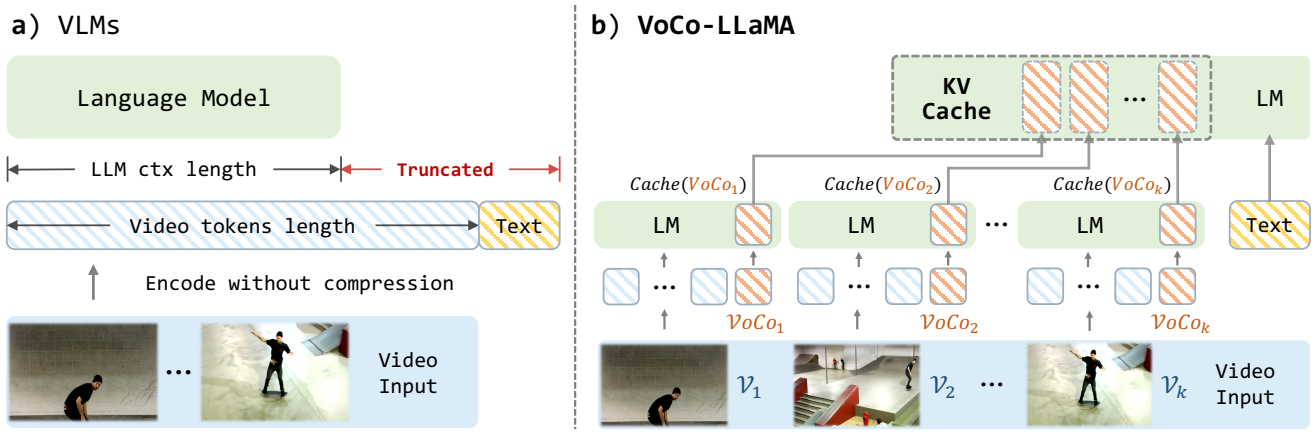


Figure 3. (a) VLMs are bottlenecked by the limited context window when processing video frames. (b) Extension of VoCo-LLaMA to video domain: Enabling more frames input with a limited context length.

text tokens, yielding a sequence:

$$(\mathcal{V}, VoCo, \mathcal{T}) = (V_0, \ldots, V_n, VoCo, T_0, \ldots, T_m) \quad (2)$$

In the training phase, we employ a two-stage attention mechanism. Initially, we impose a constraint on the text tokens, explicitly preventing them from attending to the original vision tokens, and requiring them to exclusively attend to the compressed and distilled VoCo tokens. Subsequently, the vision tokens are subjected to continuous attention from the VoCo tokens due to the casual attention mechanism. This deliberate design ensures that the text tokens solely capture the distilled visual information encoded in the VoCo tokens, rather than directly interacting with the original vision tokens, thereby achieving effective compression from vision tokens to compressed tokens.

The compression process of VoCo-LLaMA can be elegantly implemented by modifying the attention mask. Specifically, we set the attention weights between the text tokens and the vision tokens to $False$, effectively rendering the text tokens "**isolated**" to the vision tokens. Formally, let $\mathbf{M} \in \mathbb{R}^{(m+n+1)\times(m+n+1)}$ denote the attention mask, where $M_{ij} = True$ if token $i$ attends to token $j$, and $M_{ij} = False$ otherwise. We define the attention mask as:

$$M_{ij} = \begin{cases} True, & \text{if } i \in \mathcal{T} \text{ and } j \in VoCo, \\ False, & \text{if } i \in \mathcal{T} \text{ and } j \in \mathcal{V}, \\ True, & \text{otherwise.} \end{cases} \quad (3)$$

In practice, VoCo-LLaMA can be effectively trained under the standard supervised fine-tuning paradigm, leveraging the abundant image-text data readily available in VLMs. Furthermore, the VoCo token can be compactly represented as a set of Transformer activations, allowing them to be cached to enhance inference efficiency, which will be discussed in Sec. 3.3.

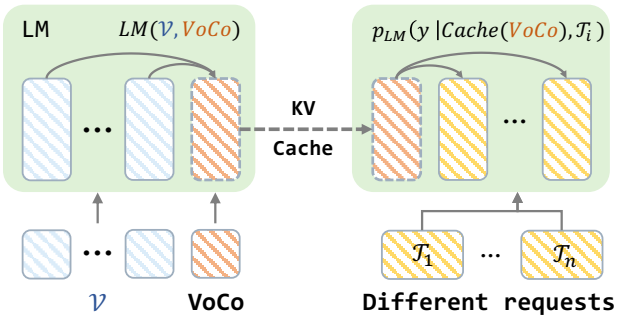


Figure 4. Illustration of the two stage forward operation with KV cache for VoCo-LLaMA during inference. The first forward pass extract image into VoCo cache. The cached VoCo tokens can be utilized to handle different taksk that involve same image.

VoCo-LLaMA enables the large language models to learn the compression process of vision tokens, $LM(\mathcal{V}, VoCo)$, while simultaneously learning to understand the compressed VoCo tokens. We define the target learning distribution as follows:

$$p_{VoCo-LLaMA} = p_{LM}(y \mid LM(\mathcal{V}, VoCo), \mathcal{T}) \quad (4)$$

the optimization objective in Eq. (1) can be defined as:

$$E_{\mathcal{V},\mathcal{T}}[D_{KL}(p_{LM_o}(y \mid \mathcal{V}, \mathcal{T}) \,||\, p_{VoCo-LLaMA}] \quad (5)$$

### 3.3. Reuse of VoCo Cache

During inference, VoCo-LLaMA mitigates the issue of limited context window size by dividing the single forward pass into two phases. As illustrated in Fig. 4, the first forward pass takes [vision tokens, VoCo tokens] as input to compress visual information into Transformer activations upon VoCo tokens. The second forward pass takes [text tokens] as input and load VoCo activations as KV Cache.

Moreover, VoCo tokens derived from the first forward pass can be cached and reused when handling various tasks involving identical image/video inputs, thereby enhancing computational efficiency and reducing storage requirements compared to maintaining the entire KV-cache for uncompressed visual tokens. For more details on the inference implementation, please refer to the *supplementary material*.

### 3.4. Temporal Modeling

When giving a sequence of video frames $Vid = \{\mathcal{V}_1, \ldots, \mathcal{V}_k\}$ and a corresponding text input, the token length for the entire video far exceeds LLM context length, as shown in Fig. 3 (a). To solve this issue, VoCo-LLaMA divides video input into smaller segments and input these segments into LLM with VoCo tokens $\{VoCo_1, \ldots, VoCo_k\}$. As shown in Fig. 3 (b), all the frames are compressed into VoCo activations. Specifically, we obtain the compressed representation $Cache_t$ for video segment tokens $\mathcal{V}_t$ through $Cache(VoCo_t) = LM(\mathcal{V}_t, VoCo_t)$. This yields a sequence of KV Cache representing compressed video tokens, denoted by $\mathcal{F} = \{Cache(VoCo_1), \ldots, Cache(VoCo_k)\}$.

Having obtained the time-series compressed cache sequences $\mathcal{F}$, we enable language model to capture and comprehend the temporal correlations among the compressed video tokens. With the inclusion of text tokens $\mathcal{T}$, VoCo-LLaMA learns the distribution $p(y \mid \mathcal{F}, \mathcal{T})$. We adopt a continue training process based on VoCo-LLaMA with image compression capabilities, allows the model to focus on temporal modeling, thereby streamlining the video understanding process.

### 3.5. Implementation Details

Regarding the training strategy and data, as mentioned earlier in Sec. 3.2, VoCo-LLaMA only requires learning to insert and compress VoCo tokens during the vision instruction tuning stage. We follow the common VLMs [29, 30] to encode the image input into vision tokens with vision encoder and a linear projector. We employ the pre-trained CLIP-ViT-L [43] as our visual encoder. For pre-trained large language models, we utilize Vicuna-7B [10]. Without introducing VoCo tokens, we first align the visual encoder and language model using the LLaVA-filtered CC3M [45] dataset with visual encoder and language model keeping frozen. During the instruction tuning phase of VoCo-LLaMA, incorporating multiple image understanding tasks is crucial for learning a scalable image compression model. Therefore, we construct the instruction pairs inspired by [28] using [29]. For video tuning, we further utilize WebVid [4] and the QA-pairs of Video-ChatGPT [39]. Moreover, gradient checkpointing strategies are employed to reduce computational cost during training.

We conducted experiments on several common compression strategies with the same training setting and data for comparison. For the compression strategy with Q-Former, we employ the architecture in [25] and configure the query number to one, resulting in a single compression token. For the compression strategy with average pooling, we follow the design of the single content token in [28]. For more details on the training and inference implementation, please refer to the *supplementary material*.

## 4. Experiments

### 4.1. Datasets

In this work, we conduct experiments on several common visual understanding benchmarks for vision compression. In particular, we report results on GQA [21], MMB (MMBench) [35], MME [15], POPE [27], SEED-Bench [24], $SQA^I$ (Image-based setting in ScienceQA) [36] and $VQA^{v2}$ (VQA V2) [17]. By observing the model's performance on these image understanding benchmarks before and after compression (*i.e.* with initial vision tokens / VoCo tokens), we can observe the effects of the visual information loss that occurs during the vision compression process. We evaluate the performance on these visual understanding benchmarks in accordance with the details outlined in [30]. As for the video domain, we evaluate the zero-shot performance on several video question-answering benchmarks. MSVD-QA [53] is a video QA dataset consisting of 1,970 video clips with 50,505 QA pairs, built upon the Microsoft Research Video Description Corpus [7]. MSRVTT-QA [53] is a large-scale video QA dataset featuring 10K videos and 243K question-answering pairs with complex scenes, based on the MSR-VTT dataset [54]. ActivityNet-QA [56] is a fully annotated video QA dataset containing 58K question-answering pairs derived from 5,800 complex web videos from the ActivityNet dataset [18].

### 4.2. Vision Compression Configuration

In the primary experiment of vision compression, we present the results of compressing all vision tokens of an image into a single VoCo token. To rigorously quantify the performance loss of VoCo-LLaMA during compression, we designed two comparative training settings: the Upper Bound model, which represents the best compression performance. The ideal case for a visual compression model is to obtain the same visual understanding capability as the upper bound model. And the Lower Bound model, which represents the worst compression performance.

The initialization model is trained by integrating VoCo tokens in a manner analogous to VoCo-LLaMA, without modifying the attention mask strategy. During inference, we employ a standard causal attention mask. This setting effectively controls for performance fluctuations induced by the introduction of additional special tokens. In contrast, the random compression model is trained under

| Method | Token | GQA | MMB | MME | POPE | SEED | SQA$^I$ | VQA$^{v2}$ | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| Upper Bound | 576 | 61.1<br>100% | 64.0<br>100% | 1487.2<br>100% | 85.0<br>100% | 57.9<br>100% | 66.5<br>100% | 77.7<br>100% | -<br>100% |
| **VoCo-LLaMA** | 1 | **57.0**<br>**82.5%** | **58.8**<br>**87.5%** | **1323.3**<br>**81.2%** | **81.4**<br>**88.4%** | **53.7**<br>**80.0%** | **65.4**<br>**81.0%** | **72.3**<br>**85.2%** | -<br>**83.7%** |
| Avg. Pool [28]<br>+ Linear | 1 | 52.9<br>65.0% | 55.5<br>79.6% | 1210.3<br>68.1% | 79.1<br>81.0% | 50.3<br>63.8% | 62.2<br>25.8% | 65.0<br>65.2% | -<br>64.1% |
| Q-Former [25] | 1 | 51.1<br>57.3% | 51.7<br>70.5% | 1079.7<br>53.2% | 77.3<br>75.2% | 47.2<br>49.0% | 62.7<br>34.5% | 63.4<br>60.8% | -<br>57.2% |
| Lower Bound | 1 | 37.7<br>0% | 22.3<br>0% | 617.3<br>0% | 53.9<br>0% | 36.9<br>0% | 60.7<br>0% | 41.2<br>0% | -<br>0% |

Table 1. Comparison with previous approaches on vision compression using common visual understanding benchmarks. All methods compress 576 vision tokens (from $(336/14)^2 = 576$) into one. We further report the compression performance mentioned in Sec. 3.5.

| Token | MMB | GQA | VQA$^{v2}$ | SEED | Avg. |
|---|---|---|---|---|---|
| 576 | 64.0 | 61.1 | 77.7 | 57.9 | 100% |
| 128 | **61.0** | 59.8 | **76.9** | **59.1** | **97.7%** |
| 64 | 60.5 | **60.4** | 75.4 | 56.3 | 93.7% |
| 32 | 59.4 | 60.2 | 75.3 | 56.2 | 92.6% |
| 16 | 58.6 | 59.4 | 75.4 | 56.2 | 91.3% |
| 8 | 58.7 | 59.2 | 75.3 | 56.3 | 91.3% |
| 4 | 60.4 | 58.4 | 74.5 | 56.0 | 90.4% |
| 2 | 60.1 | 57.7 | 73.5 | 55.0 | 87.8% |
| 1 | 58.8 | 57.0 | 72.3 | 53.7 | 83.8% |
| 1 | 22.3 | 37.7 | 41.2 | 36.9 | 0% |

Table 2. Effect of VoCo tokens count on widely used benchmarks. The number of VoCo tokens increases from 1 to 128. Green and red represent the Upper and Lower Bound, respectively.

identical settings as the initialization model. During inference, we restrict the visibility of text tokens to only the VoCo token, isolating the visual information. This setup represents a scenario without vision compression training, providing a baseline for evaluating. Based on the performance boundary model, the compression retention rate can be subsequently calculated as (result of VoCo-LLaMA − Lower Bound)/(Upper Bound − Lower Bound).

### 4.3. Results

**Vision Compression.** Tab. 1 presents the results of VoCo-LLaMA on vision compression. To explore the maximum potential of our method, we report the highest achievable compression ratio, which compresses vision tokens into one single VoCo token. We report results of our compression model on various common visual understanding benchmarks, as well as the compression retention rates defined based on upper and lower bound models introduced in Sec. 4.2. It can be observed that our method preserves the original visual information to a large extent, even at an extremely high compression ratio of 576×. Specifically, we

| Method | $N$ | GQA | POPE | SQA$^I$ | VQA$^T$ |
|---|---|---|---|---|---|
| LLaMA-VID [28] | 16 | 58.2 | 83.1 | 67.4 | 50.8 |
| | 4 | 56.2 | 83.5 | 68.7 | 49.1 |
| | 1 | 55.5 | 83.1 | 68.8 | 49.0 |
| **VoCo-LLaMA** | 1 | **58.3** | **85.0** | **69.5** | **52.7** |

Table 3. Comparison with previous compression methods which compress image into single token. $N$ means the number of "content" tokens in LLaMA-VID or the VoCo tokens in our method. The input resolution is set to 224 for fair comparison.

| Method | Token | MMB | GQA | VQA$^{v2}$ | SEED |
|---|---|---|---|---|---|
| **VoCo-LLaMA** | 32 | **59.4** | **60.2** | **75.3** | **56.2** |
| | 16 | 58.3 | 58.9 | 74.9 | 55.8 |
| | 4 | 59.7 | 58.0 | 73.5 | 55.2 |
| | 1 | 57.9 | 56.1 | 71.2 | 53.0 |

Table 4. Compression performance with adjusted VoCo token numbers during inference on model trained with fixed numbers.

achieved an average compression retention rate of 83.7% across seven widely used benchmarks. Especially on MMBench, POPE and VQA$^{v2}$, our method retained more than 85% of the performance during compression. The results indicate that VoCo-LLaMA can effectively compress vision tokens. Moreover, our method consistently outperforms the performance lower bound model of random compression across all benchmarks. This demonstrates that the advantages of VoCo-LLaMA, such as significant reductions in context length and improved calculation efficiency, outweigh the performance loss caused by compression.

We additionally compare our method with previous common learning-based approaches (*i.e.,* Q-Former and average pooling) for vision token compression. Our method significantly outperforms previous methods across all benchmarks. Specifically, we observe an improvement of 19.6% in average compression retention rate, surpassing the av-

| Method | Token | RefCOCO | | | RefCOCO+ | | | RefCOCOg | | GRIT | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | val | test A | test B | val | test A | test B | val (U) | test (U) | refexp | |
| Upper Bound | 256 | 87.01 | 90.61 | 80.24 | 81.60 | 87.36 | 72.12 | 82.27 | 82.19 | 69.34 | 100% |
| **VoCo-LLaMA** | 8 | **85.17** | **88.92** | **79.21** | **80.02** | **85.13** | **70.22** | **80.36** | **80.64** | **68.59** | **90.7%** |
| | 1 | 83.29 | 86.89 | 77.87 | 77.62 | 83.02 | 67.74 | 78.32 | 78.06 | 67.69 | 79.9% |
| Lower Bound | 1 | 68.34 | 72.96 | 68.03 | 62.58 | 64.77 | 50.65 | 62.30 | 62.99 | 60.50 | 0% |

Table 5. Compression performance on REC task. Avg. means the average compression retention rate on all benchmarks.

| Method | Token | RefCOCO | | | RefCOCO+ | | | RefCOCOg | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|
| | | val | test A | test B | val | test A | test B | val (U) | test (U) | |
| Upper Bound | 256 | 75.61 | 44.26 | 104.83 | 56.42 | 40.98 | 68.25 | 62.71 | 65.58 | 100% |
| **VoCo-LLaMA** | 8 | **73.87** | **43.13** | **102.71** | **55.34** | **39.91** | **67.00** | **61.59** | **64.45** | **91.3%** |
| | 1 | 71.92 | 41.81 | 94.50 | 53.98 | 38.96 | 65.35 | 60.46 | 63.17 | 81.6% |
| Lower Bound | 1 | 56.73 | 31.82 | 78.09 | 43.71 | 30.26 | 52.22 | 50.49 | 53.22 | 0% |

Table 6. Compression performance on REG task. Avg. means the average compression retention rate on all benchmarks.

erage pooling compression strategy. In contrast, while Q-Former has demonstrated impressive capabilities in capturing visual features with 32 queries, its performance undergoes a substantial decline when the query count is reduced to a single digit. This proves that our VoCo-LLaMA, which utilizes the knowledge distillation from language models itself, maintains more valuable vision information than that of average pooling or query-based compression.

**Number of VoCo tokens.** We evaluate the impact of the number of VoCo tokens on vision compression performance. Tab. 2 illustrates the trend of compression performance retention as the number of VoCo tokens varies, where the green and red lines represent the upper and lower bounds of compression performance, respectively. We adopted the same training settings and data as in the main experiments. It can be observed that as the number of VoCo tokens grows, the overall compression performance of the model shows an upward trend. Increasing the number of tokens within the range of fewer than 10 tokens results in a significant improvement in compression performance. Finally, when conducting 128 VoCo tokens, the model achieves an average compression performance retention rate of 97.7%, indicating that the performance loss due to compression is almost negligible when compressing into more than 100 tokens. Interestingly, we observe that when training with 128 VoCo tokens, the result on the SEED-Bench exceeds the performance upper bound model.

**Method of Compression.** We compare our method with LLaMA-VID on vision compression, specifically evaluating its full model that utilizes both context and content tokens. For a fair comparison, VoCo-LLaMA is trained under the exact same settings and applied the same visual encoder, EVA-G [14], in this experiment. As shown in Tab. 3, our method outperforms the previous approach when using a single content compression token, even surpassing the performance of LLaMA-VID when it uses multiple context tokens. In particular, we could observe an improvement of 2.8 and 3.7 on GQA and VQA$^T$ benchmarks, respectively.

**Adaptability of VoCo Number.** To assess the model's adaptability to varying numbers of compression tokens, we trained the model with a fixed number of tokens and evaluated its performance with different token numbers. As demonstrated in Tab. 4, we conducted experiments by fixing the number of VoCo tokens (32) during training and varying the number of tokens during inference. Our method achieves better performance with an increasing number of compressed tokens, without requiring specialized training for elastic compressed token.

**Results on fine-grained tasks.** We analyze the extent of loss of fine-grained visual information after high-magnification compression of vision tokens in our approach. Here, we apply our method to [8] which is a cleanly structured MLLM trained on fine-grained task data such as REG, REC, and PointQA. As shown in Tab. 5 and Tab. 6, when compressing vision tokens to 1 VoCo token, our method maintained an impressive average compression retention rate of 79.9% and 81.6% for REC and REG tasks, respectively. Furthermore, by increasing the number of VoCo tokens to 8, we observed a significant improvement in the average compression retention rate. We observe that VoCo-LLaMA achieves similar compression retention rate to other benchmarks on fine-grained tasks, mainly because the Lower Bound model incurs more information loss on fine-grained tasks. Please refer to the *supplementary material* for additional fine-grained benchmarks, including VisWiz, OCRBench and others.

**Inference Efficiency.** We discuss the inference efficiency under the scenarios that images are cached as discussed in Sec. 3.3. Due to our model's design, the representation of compressed image (*i.e.,* transformer activations on

| Method | Token | KV Cache Length | Storage Memory (MB) | Δ | CUDA Time (ms) ↓ | Δ | FLOPs (T) ↓ | Δ |
|---|---|---|---|---|---|---|---|---|
| Baseline | 576 | - | - | - | 440.5 | - | 9.6 | - |
| Full Caching | 576 | 576 | 302.4 | - | 154.9 | 64.8% | 1.2 | 87.5% |
| **VoCo-LLaMA** | **1** | **1** | **0.525** | 99.8% | **134.0** | 69.6% | **0.5** | 94.8% |

Table 7. Efficiency analysis of VoCo-LLaMA including cache storage memory, CUDA time and the FLOPs. Δ denotes the reduction ratio.

| Method | Visual Encoder | LLM | Res. | Image Token | MSVD-QA Acc | MSVD-QA Score | MSRVTT-QA Acc | MSRVTT-QA Score | ActivityNet-QA Acc | ActivityNet-QA Score |
|---|---|---|---|---|---|---|---|---|---|---|
| *Methods w/o Vision Compression* | | | | | | | | | | |
| FrozenBiLM [55] | CLIP-L | DeVERTa-V2 | 224 | 256 | 32.3 | - | 16.8 | - | 24.7 | - |
| Video-LLaMA [58] | EVA-G | Vicuna-7B | 224 | 256 | 51.6 | 2.5 | 29.6 | 1.8 | 12.4 | 1.1 |
| VideoChat [26] | - | Vicuna-7B | 224 | - | 56.3 | 2.8 | 45.0 | 2.5 | 26.5 | 2.2 |
| Video-ChatGPT [39] | CLIP-L | Vicuna-7B | 224 | 256 | 64.9 | 3.3 | 49.3 | 2.8 | 35.2 | 2.7 |
| BT-ADapter [34] | CLIP-L | Vicuna-7B | - | - | 67.5 | 3.7 | 57.0 | 3.2 | 45.7 | 3.2 |
| Vista-LLaMA [38] | EVA-G | Vicuna-7B | 224 | 256 | 65.3 | 3.6 | 60.5 | 3.3 | 48.3 | 3.3 |
| Chat-UniVi [23] | CLIP-L | Vicuna-7B | 224 | 256 | 69.3 | 3.7 | 55.0 | 3.1 | 46.1 | 3.3 |
| *Methods w/ Vision Compression* | | | | | | | | | | |
| LLaMA-VID [28] | EVA-G | Vicuna-7B | 224 | 2 | 69.7 | 3.7 | 57.7 | 3.2 | 47.4 | 3.3 |
| **VoCo-LLaMA** | CLIP-L | Vicuna-7B | 224 | 2 | **72.3** | **3.9** | **61.1** | **3.5** | 47.9 | **3.4** |
| | | | 336 | 2 | **72.6** | **3.9** | **61.2** | **3.5** | 47.9 | **3.4** |
| | | | 224 | 8 | **73.4** | **3.9** | **62.0** | **3.5** | **48.5** | **3.4** |
| | | | 336 | 8 | **73.5** | **3.9** | **62.3** | **3.5** | **48.6** | **3.4** |

Table 8. Comparison with leading video understanding methods, with and without vision compression, on three zero-shot benchmarks.

top of VoCo tokens) can be stored and repeatedly utilized in the form of a KV cache. We conduct a comparative analysis of CUDA time, FLOPs, and KV Cache storage size during the inference process, and compare our method with the baseline method and the full caching method. The baseline method, as its name suggests, does not employ any caching strategy and directly encodes and infers images. In contrast, the full caching method stores the uncompressed Transformer activations upon all vision tokens as KV caches. More specifically, we follow the approach of [42], storing the keys and values of each Transformer layer. As displayed in Tab. 7, we conduct an inference efficiency analysis on a single NVIDIA A100 using identical lengths of text prompts and single-image inputs. Compared to the baseline model without caching strategy, VoCo-LLaMA achieves a significant reduction of 69.6% in CUDA time and 94.8% in FLOPs. Relative to the full caching strategy, our method save 99.8% of cache storage while achieving lower CUDA time and FLOPs, demonstrating the inference efficiency gains brought by our approach. Please refer to the *supplementary material* for further discussion and details for inference efficiency.

**Video Understanding.** We further evaluate the performance of VoCo-LLaMA on three widely used video understanding benchmarks, reporting results for input image resolutions of 224 and 336, respectively. First, we discuss the video understanding methods that utilize vision compression. Ensuring fair comparison, we adopted the same compression ratio as previous method [28], compressing each video frame into 2 VoCo tokens for training and inference. Our method consistently outperforms previous video compression methods across all three benchmarks. Specifically, on the MSVD-QA and MSRVTT-QA datasets, VoCo-LLaMA achieved accuracies of 72.3% and 61.1%, respectively, corresponding to absolute gains of 3.7% and 5.9% over the previous best methods. Moreover, our method achieves the highest scores of 3.9 and 3.5, respectively.

In comparison to video understanding methods that do not employ vision compression, our approach, which represents each video frame with a mere 2 VoCo tokens, demonstrates strong competitiveness against methods that utilize 256 or more vision tokens per frame. To further explore the potential of VoCo-LLaMA, we opted to compress video frames into the number of VoCo tokens that exhibited the optimal compression performance within the 0 order of magnitude (*i.e.,* 8 tokens). Notably, as we increase the number of tokens, our method effectively leverages additional visual information. We also analyze the performance loss caused by vision compression and evaluate on other video QA benchmarks, as detailed in the *supplementary material*.

## 5. Conclusion

In this paper, we propose VoCo-LLaMA, the first approach to compress visual information using LLMs. By distilling the LLMs' understanding of vision tokens into a compact representation, our method can compress hundreds of vision

tokens into a single VoCo token, while minimizing information loss. VoCo-LLaMA significantly reduces cache storage and boosts efficiency during the inference stage. Moreover, our method exhibits promising performance in learning temporal understanding on video data with continuous training. In summary, our approach offers a promising solution for fully utilise the limited context window of VLMs, making them more efficient for multi-modal applications.

## References


[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, and et al. Flamingo: a visual language model for few-shot learning, 2022. 2

[2] Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Benjamin Mann, and et al. A general language assistant as a laboratory for alignment. *ArXiv*, abs/2112.00861, 2021. 2

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *arXiv preprint arXiv:2308.12966*, 2023. 1, 2

[4] Max Bain, Arsha Nagrani, Gül Varol, and Andrew Zisserman. Frozen in time: A joint video and image encoder for end-to-end retrieval. In *IEEE International Conference on Computer Vision*, 2021. 5

[5] Rohan Bavishi, Erich Elsen, Curtis Hawthorne, Maxwell Nye, Augustus Odena, Arushi Somani, and Sağnak Taşırlar. Introducing our multimodal models, 2023. 1, 2

[6] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and et al. Language models are few-shot learners, 2020. 2

[7] David L Chen and William B Dolan. Collecting highly parallel data for paraphrase evaluation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 190–200. Association for Computational Linguistics, 2011. 5

[8] Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic, 2023. 7

[9] Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3829–3846, Singapore, 2023. Association for Computational Linguistics. 2

[10] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, 2023. 2, 5

[11] Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning, 2023. 2, 3

[12] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, 2019. Association for Computational Linguistics. 2

[13] Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. Glm: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, 2022. 1, 2, 3

[14] Yuxin Fang, Wen Wang, Binhui Xie, Quan Sun, Ledell Wu, Xinggang Wang, Tiejun Huang, Xinlong Wang, and Yue Cao. Eva: Exploring the limits of masked visual representation learning at scale. *arXiv preprint arXiv:2211.07636*, 2022. 7

[15] Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023. 5

[16] Tao Ge, Hu Jing, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *The Twelfth International Conference on Learning Representations*, 2024. 2

[17] 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 *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 5

[18] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. *ArXiv*, abs/2211.11559, 2022. 5

[19] Bin Huang, Xin Wang, Hong Chen, Zihan Song, and Wenwu Zhu. Vtimellm: Empower llm to grasp video moments, 2023. 2

[20] Wei Huang, Xudong Ma, Haotong Qin, Xingyu Zheng, Chengtao Lv, Hong Chen, Jie Luo, Xiaojuan Qi, Xianglong Liu, and Michele Magno. How good are low-bit quantized llama3 models? an empirical study, 2024. 2

[21] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 5

[22] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and et al. Mistral 7b, 2023. 2

[23] Peng Jin, Ryuichi Takanobu, Wancai Zhang, Xiaochun Cao, and Li Yuan. Chat-univi: Unified visual representation empowers large language models with image and video understanding, 2024. 2, 8

[24] Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023. 5

[25] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023. 1, 2, 3, 5, 6

[26] Kunchang Li, Yinan He, Yi Wang, Yizhuo Li, Wenhai Wang, Ping Luo, Yali Wang, Limin Wang, and Yu Qiao. Videochat: Chat-centric video understanding. *arXiv preprint arXiv:2305.06355*, 2023. 2, 8

[27] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. 5

[28] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. *arXiv preprint arXiv:2311.17043*, 2023. 1, 2, 3, 5, 6, 8

[29] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 1, 2, 5

[30] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 1, 2, 3, 5

[31] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 2

[32] Hao Liu, Wilson Yan, Matei Zaharia, and Pieter Abbeel. World model on million-length video and language with ringattention. *arXiv preprint*, 2024. 1, 2

[33] Peter J. Liu, Mohammad Saleh, Etienne Pot, Ben Goodrich, Ryan Sepassi, Lukasz Kaiser, and Noam Shazeer. Generating wikipedia by summarizing long sequences, 2018. 2

[34] Ruyang Liu, Chen Li, Yixiao Ge, Ying Shan, Thomas H Li, and Ge Li. One for all: Video conversation is feasible without video instruction tuning. *arXiv preprint arXiv:2309.15785*, 2023. 2, 8

[35] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? *arXiv:2307.06281*, 2023. 5

[36] Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *The 36th Conference on Neural Information Processing Systems (NeurIPS)*, 2022. 5

[37] Ruipu Luo, Ziwang Zhao, Min Yang, Junwei Dong, Da Li, Pengcheng Lu, Tao Wang, Linmei Hu, Minghui Qiu, and Zhongyu Wei. Valley: Video assistant with large language model enhanced ability, 2023. 2

[38] Fan Ma, Xiaojie Jin, Heng Wang, Yuchen Xian, Jiashi Feng, and Yi Yang. Vista-llama: Reliable video narrator via equal distance to visual tokens, 2023. 8

[39] Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. Video-chatgpt: Towards detailed video understanding via large vision and language models. *arXiv:2306.05424*, 2023. 2, 5, 8

[40] Jesse Mu, Xiang Lisa Li, and Noah Goodman. Learning to compress prompts with gist tokens, 2024. 2

[41] OpenAI. Gpt-4 technical report. *arXiv:2303.08774*, 2023. 2

[42] Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. Efficiently scaling transformer inference, 2022. 8

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 5

[44] Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, and Timothy P. Lillicrap. Compressive transformers for long-range sequence modelling. *ArXiv*, abs/1911.05507, 2019. 2

[45] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of ACL*, 2018. 5

[46] Charles Burton Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *ArXiv*, abs/2209.15189, 2022. 2

[47] Gemini Team, Machel Reid, Nikolay Savinov, Denis Teplyashin, Dmitry, Lepikhin, Timothy Lillicrap, Jean baptiste Alayrac, Radu Soricut, Angeliki Lazaridou, Orhan Firat, Julian Schrittwieser, Ioannis Antonoglou, Rohan Anil, Sebastian Borgeaud, Andrew Dai, Katie Millican, Ethan Dyer, Mia Glaese, and et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context, 2024. 1, 2

[48] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. 2

[49] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, and et al. Llama 2: Open foundation and fine-tuned chat models, 2023. 2

[50] David Wingate, Mohammad Shoeybi, and Taylor Sorensen. Prompt compression and contrastive conditioning for controllability and toxicity reduction in language models, 2022. 2

[51] BigScience Workshop, :, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, and et al. Bloom: A 176b-parameter open-access multilingual language model, 2023. 2

[52] Yuhuai Wu, Markus N. Rabe, DeLesley Hutchins, and Christian Szegedy. Memorizing transformers, 2022. 2

[53] Dejing Xu, Zhou Zhao, Jun Xiao, Fei Wu, Hanwang Zhang, Xiangnan He, and Yueting Zhuang. Video question answer-

ing via gradually refined attention over appearance and motion. In *ACM Multimedia*, 2017. 5

[54] Jun Xu, Tao Mei, Ting Yao, and Yong Rui. Msr-vtt: A large video description dataset for bridging video and language. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5288–5296, 2016. 5

[55] Antoine Yang, Antoine Miech, Josef Sivic, Ivan Laptev, and Cordelia Schmid. Zero-shot video question answering via frozen bidirectional language models. In *NeurIPS*, 2022. 8

[56] Zhou Yu, Dejing Xu, Jun Yu, Ting Yu, Zhou Zhao, Yueting Zhuang, and Dacheng Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *AAAI*, pages 9127–9134, 2019. 5

[57] Hang Zhang, Yeyun Gong, Yelong Shen, Weisheng Li, Jiancheng Lv, Nan Duan, and Weizhu Chen. Poolingformer: Long document modeling with pooling attention. In *International Conference on Machine Learning*, 2021. 2

[58] Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023. 2, 8

[59] Pan Zhang, Xiaoyi Dong, Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Wenwei Zhang, Hang Yan, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*, 2023. 1, 2, 3

[60] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 1, 2, 3