Efficient Large Multi-modal Models via Visual Context Compression

Jieneng Chen*, **Luoxin Ye***, **Ju He***, **Zhao-Yang Wang**, **Daniel Khashabi**†, **Alan Yuille**†
Johns Hopkins University

Abstract

While significant advancements have been made in compressed representations for text embeddings in large language models (LLMs), the compression of *visual* tokens in multi-modal LLMs (MLLMs) has remained a largely overlooked area. In this work, we present the study on the analysis of redundancy concerning visual tokens and efficient training within these models. Our initial experiments show that eliminating up to 70% of visual tokens at the testing stage by simply average pooling only leads to a minimal 3% reduction in visual question answering accuracy on the GQA benchmark, indicating significant redundancy in visual context. Addressing this, we introduce *Visual Context Compressor*, which reduces the number of visual tokens to enhance training and inference efficiency without sacrificing performance. To minimize information loss caused by the compression on visual tokens while maintaining training efficiency, we develop *LLaVolta* as a light and staged training scheme that incorporates stage-wise visual context compression to progressively compress the visual tokens from heavily to lightly compression during training, yielding no loss of information when testing. Extensive experiments demonstrate that our approach enhances the performance of MLLMs in both image-language and video-language understanding, while also significantly cutting training costs and improving inference efficiency.

Website https://beckschen.github.io/llavolta.html
Code https://github.com/Beckschen/LLaVolta

1 Introduction

The advent of LLMs [33, 34, 44] has marked a new era in the field of artificial intelligence and natural language processing. LLMs can play a role as a universal interface for a general-purpose assistant, where various task instructions can be explicitly represented in language and guide the end-to-end trained neural assistant to solve a task of interest. For example, the recent success of ChatGPT [33] and GPT-4 [34] have demonstrated the power of aligned LLMs in following human instructions, and have stimulated tremendous interest in developing open-source LLMs [41, 43]. As the horizon of LLM applications broadens and the availability of open-source LLMs increases, the integration of multi-modality into these models presents a new frontier in expanding their capabilities. Multi-modal LLMs [1, 28, 40, 54] (MLLMs), which can process and understand not just text but also visual information, stand at the cutting edge of this evolution.

While MLLMs have made significant strides, a crucial aspect that remains relatively unexplored is the efficient representation and processing of visual information within these models. Substantial efforts [18, 35, 53] have been dedicated to optimizing the efficient representation of text tokens through various compression techniques [18, 35, 53], aimed at enhancing inference efficiency by

*Contributed equally.
†Advised equally.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-2353

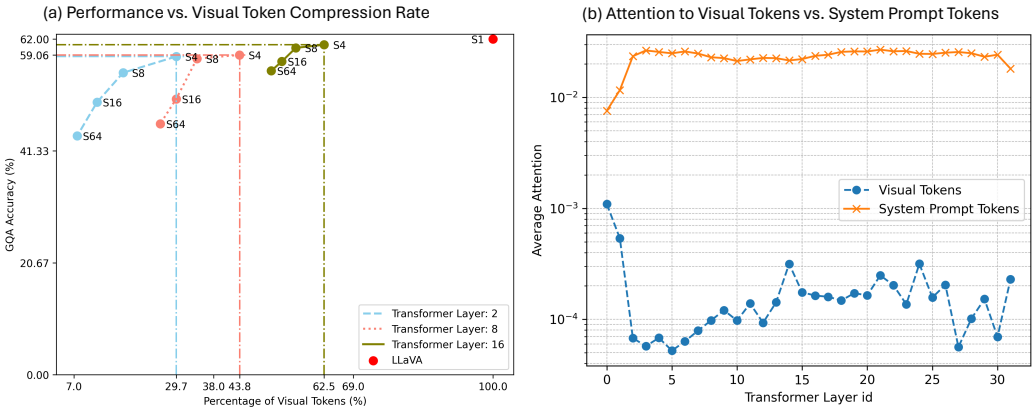

Figure 1: **Visual tokens are redundant in MLLMs. Left:** The accuracy of the LLaVA-1.5-7B [28] model(without re-train) on the GQA [20] benchmarks varies with different percentages of retained visual tokens. The x-axis represents the percentage of original visual tokens preserved after applying 1D average pooling with varying stride sizes S applied in i-th Transformer layer. **Right:** Visual tokens receive less attention from the `[ANS]` token as we go deeper into its layers of LLaVA-1.5-7B model. These findings collectively suggest a significant redundancy within the visual tokens of the MLLMs.

attentively selecting important tokens. However, the efficient learning of *visual* tokens in MLLM has not garnered comparable attention. Naturally, this raises questions about the potential redundancy present in visual tokens and its implications for the overall computational efficiency of MLLMs.

We start our work by addressing the question: *Are visual tokens redundant in multi-modal LLMs?* To explore this, we first experiment with simply reducing the number of visual tokens in a pre-trained LLaVA-1.5-7B [28] at the inference stage via average pooling (§3.2). As shown in Fig.1 (left), our initial results demonstrate that eliminating up to 70% of visual tokens by pooling them with a stride of 4 starting from Transformer layer 2 incurs only a minimal performance loss on the GQA benchmark, specifically a 3% accuracy reduction. Additionally, we compute and present the average attention values from the `[ANS]` token to visual tokens and system prompt tokens across different Transformer layers in the pre-trained LLaVA-1.5-7B [28]. As revealed in Fig. 1 (right; blue trends), the visual tokens are generally less attended to, measured based on average attention from the `[ANS]` token, as the layers get deeper. These two early explorations indicate significant redundancy in visual tokens.

Addressing this, in this work we develop an effective *Visual Context Compressor* that can be integrated into the training of MLLMs. Surprisingly, a simple average pooler nested in LLMs stands out as the most effective compressor, outperforming the attention-based [18, 53] and parametric [23] counterparts. We attribute this to two reasons: (1) The simple pooling operation makes training stable, whereas prior attention-based approaches [18, 53] are specifically designed for accelerating inference rather than training. (2) Visual tokens in the deeper Transformer layers are less attended to (see Fig. 1 (right)) and particularly redundant, making a simple compressor placed in a deeper Transformer layer effective enough. At a lower training cost, the LLaVA-1.5-7B [28] trained with the proposed *Visual Context Compressor* is competitive with the non-compressed baseline across various multi-modal benchmarks (*e.g.*, GQA [20] and MM-Vet [50]). This dual achievement highlights *Visual Context Compressor*'s role as a pivotal advancement in enhancing the efficiency and performance of MLLMs across various multi-modal question-answering benchmarks.

To further mitigate the information loss caused by compressing visual tokens, especially under a large compression ratio (CR), we have devised a **LLaV**A-p**o**wered **l**ite **tra**ining scheme, dubbed ***LLaVolta***, which progressively employs *Visual Context Compressor* at multiple training stages with different compression ratios (§3.3). Specifically, *LLaVolta* progresses through several stages, beginning with a high level of visual token compression and gradually reducing the compression ratio until the final stages, where full visual tokens are utilized. This multi-stage approach allows for adaptive compression levels that ensure training efficiency without losing information at testing, thus maintaining the overall effectiveness of the model.

Extensive experimental evaluations of *LLaVolta* have been conducted on thirteen widely-adopted MLLM benchmarks for both image-language understanding and video-language understanding ,

https://doi.org/10.52202/079017-2353

showing promising results. We observe that *LLaVolta* not only enhances the performance of MLLMs, but also achieves a substantial reduction in training costs. These experiments validate the effectiveness of our method, demonstrating its capability to optimize resource utilization while maintaining or even improving model performance.

In summary, our paper makes the following contributions:

- We present two initial studies to verify the redundancy of visual tokens in MLLMs.
- We propose the *Visual Context Compressor*, a simple yet effective compression technique that utilizes an average pooler, enhancing the efficiency of multi-modal models.
- We propose the *LLaVolta* as an efficient training scheme by leveraging *Visual Context Compressor* at multiple training stages with a progressively decreasing compression ratio. To the best of our knowledge, we are among the first to explore efficient training of MLLMs.
- Extensive experiments show that our approach not only improves the performance of MLLMs in image-language and video-language understanding across various benchmarks but also showcases efficiency gains by reducing training costs by 16% and inference latency by 24%.

2 Related Works

Multi-modal LLMs. The evolution of large language models [10, 33, 34] into their multi-modal counterparts [28, 40] represents a significant leap in their ability to follow instructions and generalize across tasks. This transition has been marked by seminal works such as Flamingo [1], BLIP-2 [23] and LLaVA [28], which have extended LLM capabilities to encompass visual tasks, demonstrating impressive zero-shot generalization and in-context learning abilities. Progress in multi-modal LLMs has primarily been driven by advancements in visual instruction tuning [28, 54], leveraging vision-language datasets and refining visual instruction-following data. Additionally, efforts have been made to enhance the grounding capabilities of multi-modal LLMs through the use of specialized datasets aimed at improving task-specific performance. Despite these advancements, the exploration of visual compression within multi-modal LLMs remains relatively underdeveloped. The design and optimization of compression strategies are crucial for maximizing the effectiveness and efficiency of multi-modal LLMs, suggesting a potential area for future research and development.

Visual Redundancy. In computer vision, reducing redundancy is crucial for creating efficient yet effective models without losing accuracy [4]. Redundancy in images often arises from the inherent characteristics of natural scenes, including repetitive patterns, textures, and areas of uniform color. These features, while contributing to the richness and detail of visual perception, can lead to inefficiencies in both storage and processing when not adequately addressed. Image compression algorithms [46] can reduce file size by eliminating or efficiently encoding redundant data. These methods take advantage of human visual perception's tolerances to subtly reduce data without significantly impacting image quality. Advanced machine learning models, particularly CNNs and autoencoders [3], offer sophisticated approaches to minimizing redundancy. Transformers [45], as a fundamental architecture for LLMs [10, 34], apply self-attention mechanisms to dynamically bind the most informative parts of tokents. Vision Transformers [6, 7, 8, 12, 16] trained with CLIP objective [7, 36] encode an image to a sequence of visual features for multi-modal LLMs [28]. Nevertheless, visual tokens receive less attention in LLMs due to attention shrinkage [47], resulting a waste of computation. In this work, we focus on reducing the redundancy of visual tokens in MLLMs.

Efficient LLMs. Efficient inference and training for LLMs are important. Compressing input sequences for efficiency reasons in Transformers is not a new idea for NLP. Much work is being done to accelerate the inference of LMs. For example, Pyramid Transformer variants [11] and [19] are proposed in Encoder-Decoder LMs that progressively compress the sequence as the layers grow deeper via pooling or core-set selection. Nawrot et al. [32] propose adaptively compressing the sequence based on the predicted semantic boundaries within the sequence. Rae et al. [37] propose compressing the fine-grained past activations to coarser memories. VCC [53] compress the sequence into a much smaller representation at each layer by prioritizing important tokens. Besides efficient inference, accelerating training for LLMs attracts attention as well. A staged training setup [38] is proposed which begins with a small model and incrementally increases the amount of compute used

 https://doi.org/10.52202/079017-2353

for training by applying a growth operator to increase the model depth and width. However, efficient training for LLMs in multi-modal scenarios is rarely explored.

3 Method

In this section, we first introduce an overview of multi-modal LLMs in § 3.1. Then, we define the problem of visual redundancy and introduce *Visual Context Compressor* in § 3.2. Finally, we present our proposed *LLaVolta* in § 3.3.

3.1 Preliminaries: A Multi-modal LLM

We start by reviewing the design of the LLaVA family [27, 28]. For processing an input image $\mathbf{X}_v$, we utilize the pre-trained CLIP visual encoder ViT-L/14, as detailed by [36], to extract the visual feature $\mathbf{Z}_v = g(\mathbf{X}_v)$, where $g(.)$ indicates the visual encoder. To bridge the gap between visual and linguistic modalities, the LLaVA [27, 28] framework as an MLLM implements a straightforward linear/MLP transformation. This involves a trainable projection matrix $\mathbf{W}$, which maps the visual features $\mathbf{Z}_v$ into the linguistic embedding space, producing language embedding tokens $\mathbf{H}_v = \mathbf{W}\mathbf{Z}_v$. These tokens are designed to match the dimensionality of the word embeddings within the LLM.

For each image $\mathbf{X}_v$, one can generate multi-turn conversation data $(\mathbf{X}_q^1, \mathbf{X}_a^1, \cdots, \mathbf{X}_q^T, \mathbf{X}_a^T)$ with T as the number of turns. One can organize them as a sequence, by treating all answers as the assistant's response and the instruction $\mathbf{X}_{\texttt{instruct}}^t$ at the t-th turn as:

$$\mathbf{X}_{\texttt{instruct}}^t = \begin{cases} \text{Random Choose}[\mathbf{X}_q^1, \mathbf{X}_v] \text{ or } [\mathbf{X}_v, \mathbf{X}_q^1], & t = 1 \\ \mathbf{X}_q^t, & t > 1 \end{cases} \tag{1}$$

This approach establishes a standardized format for the multi-modal instruction-following sequence. It allows for the instruction-based tuning of the LLM to be applied to the prediction tokens, utilizing the model's native auto-regressive training objective. Specifically, for a sequence with length L, the likelihood of the target responses $\mathbf{X}_a$ is calculated as:

$$p(\mathbf{X}_a|\mathbf{X}_v, \mathbf{X}_{\texttt{instruct}}) = \prod_{i=1}^{L} p_\theta(x_i|\mathbf{X}_v, \mathbf{X}_{\texttt{instruct},<i}, \mathbf{X}_{a,<i}), \tag{2}$$

3.2 *Visual Context Compressor*

Problem Formulation: The redundancy observed in images often arises from inherent traits of natural scenes, including repetitive patterns, textures, and regions with uniform color. While these traits enrich visual perception by offering detail and depth, they can also present challenges in terms of storage and processing efficiency. Considering the inherent limitations of Transformers in handling long sequences [2, 29, 49], it is critical to minimize any length redundancies to obtain a more effective accuracy/efficiency trade-off.

The objective of this study is to decrease the length of visual tokens $\mathbf{X}_v$ (*i.e.,* its hidden states $\mathbf{H}_v$ if inside LLMs), while simultaneously maximizing the probability of the target response $p(\mathbf{X}_a|\mathbf{X}_v, \mathbf{X}_{\texttt{instruct}})$ as described in Equation (2).

Visual Context Compressor: A key design change that we introduce is a compressor layer that compresses the dimensions of the visual inputs by reducing the effective number of visual tokens. As depicted in Fig. 2, the compressor is simply an average pooler in our setting. It is applied to the visual tokens in k-th Transformer layer of an LLM. Formally, given the hidden visual tokens at k-th Transformer layer $\mathbf{H}_k \in \mathbb{R}^{B\times C\times L}$, the compressor is expected to fulfill the following projection: $f : \mathbb{R}^{B\times C\times L} \mapsto \mathbb{R}^{B\times C\times L_{out}}$, which results in compressed visual tokens $\tilde{\mathbf{H}}_k \in \mathbb{R}^{B\times C\times L_{out}}$, where $L_{out} = \frac{L}{S}$ with s as the compression stride. In §4, we explore multiple variants of compressor f to reduce the token length, including random token dropping [17] with dropping ratio $1 - \frac{1}{S}$, K-Means [21] with number of centroids set to $N_C = \frac{L}{S}$, attention-based token-centric compression [53], attention-based token dropping [9, 18], and average pooling with stride s. To our surprise, we find that the simple average pooler is the most effective compressor for vision tokens within MLLMs, due to its stability during training detailed in § 4.4. Thus, we choose average pooler as the compressor.

Note that the proposed *Visual Context Compressor* can be directly applied to any off-the-shelf MLLMs to assess the visual redundancy, as conducted in §4.2. One can also train an MLLM with *Visual Context Compressor* to reduce the number of visual tokens while maintaining competitive multi-modal performance.

Compression Ratio (CR)[3]. For an LLM with N Transformer decoder layers, the compression ratio for visual tokens can be calculated as:

$$\text{CR} = \frac{N \cdot L}{(N-K) \cdot L_{out} + K \cdot L} \quad , \tag{3}$$

where K is the K-th Transformer layer of a multi-modal LLM; L is the the length of visual tokens input into Visual Context Compressor; L_{out} is the compressed length of visual tokens generated by Visual Context Compressor, as illustrated in Fig. 2.

Our architecture modifications thus far mostly impacts the inference efficiency of MLLM, however, its impact on performance-compression trade-off remains unclear. We will study this question in the context of **training** MLLMs with a goal of enhancing efficiency without compromising performance. We then move on to further utilize *Visual Context Compressor* to design an efficient training scheme to incorporates Visual Context Compressor at various stages of the training process.

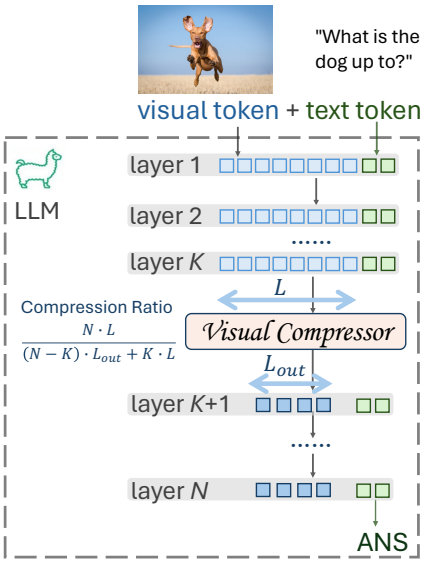

Figure 2: Example of Visual Context Compressor in a multi-modal LLM.

3.3 *LLaVolta* as a Light, Staged Training Scheme

Training with *Visual Context Compressor* not only facilitates efficient inference but also enhances training efficiency. However, devising an effective training scheme poses challenges when ensuring fair comparisons with the original LLaVA [27], primarily due to differences in the number of tokens involved in inference. This discrepancy may lead to information loss, particularly when operating under a scenario with a high compression ratio. To tackle this issue, we have developed a lite training scheme for LLaVA, dubbed as *LLaVolta*, which employs stage-wise visual context compression. Generally, assuming there are N_s total stages, stage i involves $\frac{1}{N_s}$ of the total training epochs with a compression ratio of r_i, and the final stage proceeds without any compression. Essentially, as training progresses, i increases while r_i decreases.

In this work, as depicted in Fig. 3, we primarily explore a three-stage training pipeline that progressively reduces the compression ratio, as detailed below:

Training Stage I: Heavy Compression. The MLLM training at the first one-third of the total training iterations commences with a heavy compression ratio (> 500%), where *Visual Context Compressor* is applied in an early layer of the LLM with a large pooling stride. This setup enables a very fast training speed.

Training Stage II: Light Compression. The MLLM continues training with another one-third of the total training epochs. At this stage, *Visual Context Compressor* is applied at only the deeper layers of the LLM with a smaller pooling stride compared to Training Stage I.

Training Stage III: No/subtle Compression. The MLLM continues training during the final one-third of the total epochs, with either no compression or subtle compression applied. This stage is designed to align with the inference process, where visual tokens may also undergo compression. By maintaining consistency between training and inference, this approach ensures that critical information is preserved while still allowing for compression, minimizing any potential discrepancies between training and real-world use.

Given the above meta framework, we can instantiate a family of training schemes, as demonstrated in Tab. 1. The single-stage (non-compression) scheme is equivalent to the MLLM baseline. For

[3]Definition of compression ratio from Wikipedia

https://doi.org/10.52202/079017-2353

multi-stage training, the compression stage can either go deeper or wider. "deeper" implies an increase in K (Transformer layer), while "wider" means a decrease in the stride of the pooler.

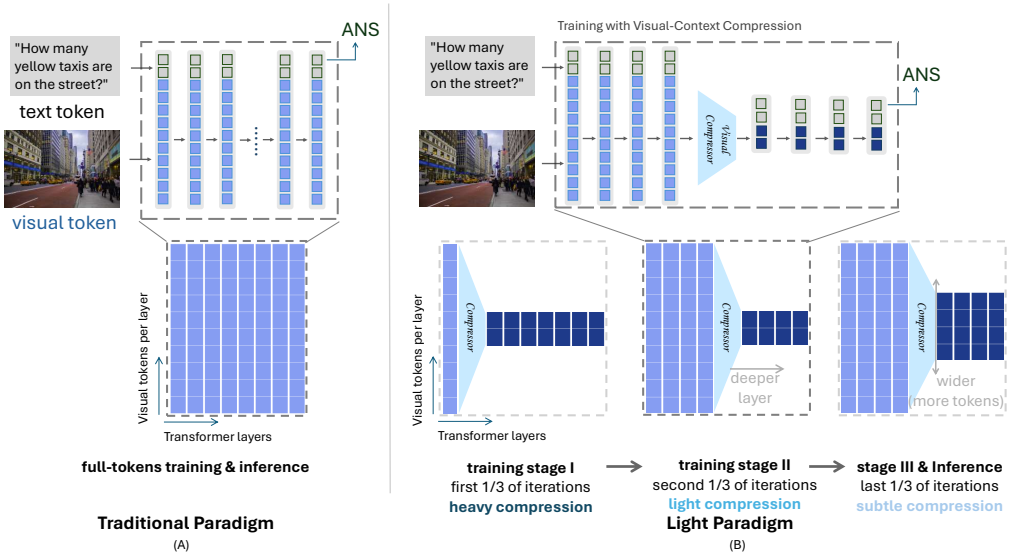

Figure 3: Training & inference paradigm comparison for conventional setting (A) and ***LLaVolta*** (B). Meta framework of ***LLaVolta*** consists three training stages: Stage I with heavy visual compression; Stage II with light visual compression in *deeper* layer; Stage III with subtle compression with *wider* token window without loss of performance. This can accelerate the training and inference by 18+% while maintaining performance.

#Stages	Scheme	Stage	Layer	Stride	CR	#Epoch
Single	no compression	$S1$	/	/	100%	1
Two	compression	$S1$	2	8	557%	0.5
		$S2$	/	/	100%	0.5
Three	compr. deeper	$S1$	2	8	557%	0.33
		$S2$	16	8	178%	0.33
		$S3$	/	/	100%	0.33
Three	compr. wider	$S1$	2	8	557%	0.33
		$S2$	2	2	188%	0.33
		$S3$	/	/	100%	0.33

#Stages	Scheme	Stage	Layer	Stride	CR	#Epoch
Four	wider then deeper	$S1$	2	8	557%	0.25
		$S2$	2	2	188%	0.25
		$S3$	16	2	133%	0.25
		$S4$	/	/	100%	0.25
Four	deeper then wider	$S1$	2	8	557%	0.25
		$S2$	16	8	178%	0.25
		$S3$	16	2	133%	0.25
		$S4$	/	/	100%	0.25
Three	last stage compression	$S1$	2	16	825%	0.33
		$S2$	16	16	188%	0.33
		$S3$	16	4	160%	0.33

Table 1: **Instantiations of *LLaVolta* schemes**. deeper indicates that the compressor's position in the LLM shifts from the shallow layer (*e.g.*, 2) to a deeper layer (*e.g.*, 16). wider indicates that the compressor's stride decreases while the number of visual tokens increases. Last stage compression refers to using compressor at last stage for efficient inference.

Note that all training schemes will be standardized to complete just one epoch. Thus, in the three-stage training, each stage will receive one third of an epoch, while in the four-stage training, each stage will receive one fourth of an epoch. Effects of non-uniform stage splitting are presented in the Appendix.

4 Experiments

In this section, we begin by detailing the experimental setup in § 4.1. Next, we elaborate on the proof-of-concept in Section § 4.2. Following this, we validate the proposed *LLaVolta* in § 4.3 with an ablation study in § 4.4. Finally, we assess the extensibility to video-language in § 4.5.

https://doi.org/10.52202/079017-2353

4.1 Experimental Setup

We adopt the Vicuna-v1.5-7B [10] as the language model, leveraging the LLaMA2 codebase [43]. We leverage the pre-trained CLIP ViT-L/14 [12, 36] with an input resolution of 336×336, resulting in 576 visual tokens. We employ the LLaVA framework [27] to connect the frozen CLIP vision encoder and the Vicuna LLMs. Along with the projector, we train the entire LLM instead of parameter-efficient finetuning. We follow LLaVA-1.5 [27] to perform data preparation and training schedule for pretraining and instruction tuning. We conduct all the experiments with the machine of $8\times$ Nvidia RTX 6000 Ada. Due to multiple invalid image links in the dataset of instruction tuning stage, the scores of LLaVA-1.5 reported in our analysis are reproduced by ourselves to ensure a fair comparison under the same experimental environment.

It is worth mentioning that assessing visual token redundancy only necessitates the inference of existing off-the-shelf models, whereas the other experiments involve the training of multi-modal LLMs, specifically projectors and LLMs.

Benchmarks and Metrics: We adopt thirteen benchmarks specifically designed for MLLM evaluation, including GQA [20], MM-Vet [50], ScienceQA (SQA)[31], MME[13], TextVQA [39], POPE [24], MMBench [30], MMBench-CN [30], VQA-v2 [14], LLaVA-Bench-in-the-Wild (LLaVA^W) [28], VisWiz [15], SEED-Image [22] and MMMU [52]. GQA and VQA-v2 evaluate the model's visual perception capabilities on open-ended short answers. MME-Perception evaluates model's visual perception with yes/no questions. ScienceQA with multiple choice are used to evaluate the zero-shot generalization on scientific question answering. TextVQA contains text-rich visual question answering. MMBench and the CN version evaluate a model's answer robustness with all-round shuffling on multiple choice answers. MM-Vet evaluates a model's capabilities in engaging in visual conversations. Additionally, we extend *LLaVolta* to video-language understanding, and follow Video-LLaVA [26] to evaluate the models on MSVD-QA [5], MSRVTT-QA [48] and ActivityNet-QA [51], where the accuracy and score are assessed using GPT-Assistant.
We report the official metrics calculated using the standard implementations provided for each benchmark for a fair comparison. Latency is reported as the time taken during inference until the first answer token is produced. When reporting average performance in Table 2, the score of MME is divided by 2000, as its range is from 800 to 2000. TFLOPs are profiled via DeepSpeed. For total number of tokens, $\#\text{Tokens} = \sum_i^N \#\text{Token}^i$. The training time is reported for one epoch of training during the LLaVA instruction-tuning stage. The Compression Ratio (CR) is defined as in Equation 3.

4.2 Proof of Concept: Visual Context Redundancy

To assess the redundancy of visual tokens, we perform average pooling within an off-the-shelf LLaVA-1.5-7B checkpoint at the testing stage, using different pooling stride sizes S across various Transformer layers K. As shown in Fig. 1, the model still exhibits strong performance even when retaining only 62.5% of the visual tokens ($S = 4, K = 16$) in the MM-Vet benchmark, without the need for additional training. When adopting the same setting ($S = 4, K = 16$), a similar trend can be observed in the GQA benchmark as well, where the compressed model only has 1% performance drop than the uncompressed counterpart. Surprisingly, in the GQA benchmark, eliminating up to 70% of visual tokens ($S = 4, K = 16$) results in a mere 3% decrease in performance. This proof-of-concept shows a certain level of redundancy in the visual tokens within MLLMs.

4.3 Main Results: *LLaVolta*

In this section, we present the main results of *LLaVolta* schemes instantiated in § 3.3. We conduct a thorough evaluation of the multi-modal capability across 13 benchmarks. Tab. 2 demonstrates that our proposed *LLaVolta* not only consistently lowers training costs by 19% (15.3 hours *vs*. 12.4 hours) but also surpasses the non-compression baseline. The last-stage-compression training schemes achieves the best performance across thirteen benchmarks and obtains 62.1% average performance, improving LLaVA-v1.5-7B [27] with much less inference TFLOPs and training time. This indicates the necessity of designing an optimally lite training scheme.

https://doi.org/10.52202/079017-2353

#Stages	Scheme	#Tokens†	CR†	Last Stage TFLOPs†	Latency (ms)	Train Time	GQA	MMVet	SQA	MME	VQA^T	POPE	MMB	MMB^{CN}	VQA^{v2}	LLaVA^w	VisWiz	SEED^I	MMMU	Avg.
Single	no compression	18432	-	8.26	68.5	15.3h	**62.6**$_{.49}$	**31.9**$_{1}$	70.8$_{.59}$	1467$_{13}$	58.3$_{.15}$	86.1$_{.24}$	65.3$_{.93}$	59.4$_{.92}$	**78.9**$_{.37}$	65.5$_{.56}$	49.8$_{.6}$	**66.7**$_{.25}$	35.1$_{.86}$	61.8$_{.32}$
Two	compression	10062	183%	8.26	68.5	12.8h	61.9$_{.23}$	31.7$_{1.5}$	70.9$_{.34}$	**1480**$_{23}$	58.3$_{.46}$	86.5$_{.33}$	64.8$_{.23}$	59.0$_{1.1}$	78.5$_{.20}$	67.3$_{.91}$	47.2$_{1.8}$	64.9$_{.17}$	34.9$_{.11}$	61.5$_{.40}$
Three	compr. deeper	10597	174%	8.26	68.5	12.8h	62.1$_{.01}$	30.5$_{.40}$	70.5$_{.23}$	1477$_{13}$	58.4$_{.07}$	86.6$_{.14}$	65.6$_{.26}$	59.9$_{.27}$	78.5$_{.22}$	67.5$_{1.4}$	49.2$_{.56}$	65.9$_{.17}$	35.0$_{.19}$	61.8$_{.10}$
Three	compr. wider	10407	177%	8.26	68.5	12.8h	61.1$_{1.6}$	31.8$_{.61}$	71.0$_{.28}$	1434$_{12}$	58.5$_{.04}$	86.6$_{.06}$	64.8$_{.23}$	59.1$_{.83}$	78.7$_{.02}$	64.3$_{4.8}$	49.8$_{1.1}$	65.3$_{.04}$	34.3$_{.75}$	61.3$_{.28}$
Four	wider then deeper	11088	166%	8.26	68.5	12.9h	62.1$_{.09}$	31.6$_{.58}$	**71.4**$_{.36}$	1444$_{15}$	58.7$_{.24}$	**86.8**$_{.21}$	65.3$_{.30}$	59.3$_{.26}$	78.8$_{.05}$	67.7$_{3.1}$	**50.1**$_{.21}$	65.6$_{.15}$	33.8$_{.78}$	61.8$_{.35}$
Four	deeper then wider	10863	170%	8.26	68.5	12.8h	62.1$_{.07}$	31.5$_{.20}$	70.5$_{.16}$	1472$_{16}$	**58.7**$_{.08}$	86.3$_{.33}$	**65.6**$_{.52}$	**59.9**$_{.61}$	78.8$_{.03}$	68.2$_{2.1}$	48.3$_{1.3}$	66.1$_{.20}$	35.1$_{.02}$	61.9$_{.47}$
Three	last stage compression	7848	235%	5.47	52.2	12.4h	62.3$_{.26}$	31.5$_{.35}$	71.0$_{.17}$	**1519**$_{14}$	58.0$_{.12}$	86.5$_{.30}$	65.3$_{.45}$	59.1$_{.60}$	78.2$_{.02}$	**69.2**$_{0.15}$	**50.2**$_{1.0}$	65.4$_{.16}$	**35.4**$_{.22}$	**62.1**$_{.07}$

Table 2: **Performance of *LLaVolta***. See the definition of each training scheme in Tab. 1. †: average across stages. First five derived schemes for training acceleration achieve competitive results while reducing 16% training time. The last scheme, last stage compression, achieved the shortest training time (12.4 hours) and the lowest inference cost (5.47 TFLOPs), but also the highest average performance (62.1%). We report average results across three runs, with the standard deviation written at the bottom right of the average result. *The last stage compression training achieves the best average performance across thirteen benchmarks, outperforming the baseline (LLaVA-v1.5-7B) while requiring significantly less training time.*

4.4 Ablation Study

In this section, we perform an ablation study on the choice of visual compressors by comparing different compression methods. Additionally, we examine the effects of varying the stride and LLM layer in training *Visual Context Compressor*.

Compressor	#Tokens	CR	GQA	MM-Vet	SQA	MME	VQA^T	POPE	MMB	MMB^{CN}	VQA^{v2}	LLaVA^w	VisWiz	SEED^I	MMMU	Avg.
*Train **without** compression; Testing with compression*																
Random Dropping	3312	556%	50.6	21.4	69.3	1142	46.5	55.8	39.7	33.3	59.3	47.6	47.2	52.2	34.3	47.3
K-Means	3312	556%	54.4	25.9	**69.7**	1155	49.0	**78.6**	55.3	46.1	**69.3**	**57.6**	48.9	56.1	32.9	54.0
FastV [9]	3312	556%	52.1	**30.6**	69.4	**1298**	**53.4**	65.6	60.1	**53.0**	68.6	54.8	**50.0**	56.3	**34.9**	**54.9**
VCC [53]	3582	514%	**54.7**	26.9	69.2	1246	49.2	72.3	**60.8**	52.0	68.1	55.6	47.8	**57.0**	34.8	54.7
Average Pooling	3312	556%	53.7	25.6	69.4	1150	47.7	70.1	56.4	46.5	67.0	55.6	50.0	55.7	34.3	53.0
*Train **with** compression; Testing with compression*																
Random Dropping	3312	556%	53.4	25.0	69.4	1186	49.4	64.9	52.0	41.1	59.7	51.5	47.9	52.6	34.6	50.8
K-Means	3312	556%	57.5	25.9	55.6	1279	51.4	79.4	62.6	54.6	75.7	59	46.1	59.2	34.1	57.9
FastV [9]	3312	556%	55.9	27.9	70.4	1327	49.7	79.8	62.9	55.9	69.5	61.7	**49.6**	56.8	**35.1**	57.0
VCC [53]	3582	514%	57.7	29.3	70.7	1398	53.0	83.6	65.0	55.8	74.1	58.0	48.2	60.1	35.0	58.5
Average Pooling	3312	556%	**60.0**	**30.7**	**70.8**	**1450**	**55.1**	**85.5**	**65.0**	**59.5**	**75.9**	**66.9**	46.4	**62.6**	33.8	**60.4**

Table 3: **Comparison among different visual compressors**. Higher values are preferred. All methods except VCC are set to the compression ratio of 556% to approximate VCC's 514% [53] for a fair comparison. The best scores are marked as gray and the second best are underlined. Attention-based compressors (*i.e.*, FastV and VCC) excel during the inference phase, yet their application to the training phase proves challenging. *Average pooling shows a more stable performance during the training phase.*

Choice of Visual Compressors. The design choices include (1) random token dropping, (2) K-Means clustering, (3) average pooling, (4) FastV [9], (5) VCC [18], (6) parametric pre-trained Q-Former [23]. We have the following three observations. Firstly, Tab. 3 shows that the attention-based methods, including FastV and VCC win 9/13 best and second best scores, showcasing the high performance when compressing visual tokens in inference. However, they are ineffective when applied to training because the in-training attention scores are unstable. Secondly, and surprisingly, the average pooling obtains the highest scores on eleven out of thirteen benchmarks when it is used to train MLLMs with a high CR. Thirdly, Tab. 4 shows that both Q-Former and average pooling can obtain reasonably good performance when trained with extremely high CRs, and the average pooling performs better with less training cost. The reason could be that the Q-Former resamples tokens outside the LLM, potentially causing the LLM to overlook crucial information relevant to the response. In contrast, our approach employs average pooling subsequent to Transformer layer K, allowing the initial K layers of the LLM to effectively retain important information from uncompressed tokens. Given these three insights, we select average pooling as our favored approach for visual compression.

Performance Across Compression Ratios. Herein, we train the multi-modal LLM with our *Visual Context Compressor* in various settings. As demonstrated in Tab. 5, the proposed method offers certain improvements and trade-offs compared to the state-of-the-art method, LLaVA-1.5-7B. We have the following two observations. Firstly, in the heavy compression level, the performance of MLLM is inversely proportional to the compression ratio (linearly scaling to the number of visual

https://doi.org/10.52202/079017-2353

Method	#Param	#Tokens	CR	Train Time	GQA	MMVet	SQA	MME	VQA^T	POPE	MMB	MMB^{CN}	VQA^{v2}	LLaVA^w	VisWiz	SEED^I	MMMU	Avg.
Q-Former [23]	105M	1024	1800%	10.4h	55.7	26.4	69.3	1217	49.2	83.0	57.7	50.7	71.4	64.6	52.6	55.1	34.0	56.2
Ours	0	855	2156%	9.2h	55.9	26.3	71.0	1321	51.6	82.5	63.3	55.9	74.5	63.1	47.8	57.3	35.7	57.8

Table 4: **Parametric *vs*. nonparametric visual compressor.** We follow miniGPT-4 [54] that uses Q-Former pre-trained from BLIP-2 [23] as the parametric compressor (All other aspects are maintained as in LLaVA to ensure a fair comparison). Ours: pooling with stride 64 on LLM layer 1 to ensure comparable CRs. *Our nonparametric compressor outshines the parametric Q-Former counterpart in terms of both performance and training efficiency.*

tokens). Secondly, the performance of MLLMs at the light compression level does not correlate directly with the number of visual tokens, making this observation somewhat unexpected. We attribute this to the MLLMs at this level of compression being relatively insensitive to changes in the compression ratio. This indicates that MLLMs trained at a light compression level will not hurt the model performance at all. For instance, the setting of stride 16 in light compression level attains a 188% CR and also outperforms the baseline LLaVA-v1.5-7B across all four metrics. The above observations pave the way for developing a more systematic training scheme.

Stride	#Tokens	CR	Latency	TFLOPs	Train time	GQA	MMVet	SQA	MME	VQA^T	POPE	MMB	MMB^{CN}	VQA^{v2}	LLaVA^w	VisWiz	SEED^I	MMMU	Avg.
Heavy compression in LLM layer 2																			
8	3312	557%	37.9ms	2.14	12.0	$59.9_{.13}$	$30.1_{.92}$	$70.9_{.17}$	1443_{11}	$55.3_{.3}$	$85.3_{.21}$	$65.2_{.25}$	$\mathbf{59.5}_{.06}$	$76.0_{.09}$	$65.9_{2.0}$	$46.6_{.2}$	$62.6_{.0}$	$34.2_{.54}$	$60.3_{.2}$
4	5472	337%	39.1ms	3.02	12.2	$61.3_{.23}$	$\mathbf{32.3}_{.35}$	$\mathbf{71.4}_{.16}$	$1456_{5.4}$	$56.6_{.42}$	$85.6_{.01}$	$65.8_{.54}$	$\mathbf{59.5}_{1.1}$	$77.4_{.02}$	$\mathbf{67.3}_{2.7}$	$50.4_{.38}$	$63.9_{.49}$	$34.9_{.08}$	$61.5_{.1}$
2	9792	188%	48.6ms	4.77	12.6	$61.9_{.43}$	$30.9_{1.1}$	$71.6_{.69}$	1450_{18}	$57.6_{.08}$	$86.3_{.22}$	$67.2_{.05}$	$59.9_{.4}$	$78.0_{.17}$	$66.4_{.85}$	$48.7_{.25}$	$65.9_{.49}$	$34.1_{.34}$	$61.6_{.08}$
Light compression in LLM layer 16																			
8	10368	178%	51.3ms	5.00	12.8	$\mathbf{62.6}_{.03}$	$30.4_{.54}$	$71.1_{.27}$	1462_{9}	$58.2_{.01}$	$86.0_{.09}$	$65.3_{.52}$	$58.9_{.57}$	$78.8_{.12}$	$63.9_{1.1}$	$\mathbf{51.4}_{.15}$	$66.8_{.23}$	$\mathbf{35.8}_{1.4}$	$61.8_{.04}$
4	11520	160%	52.2ms	5.47	13.2	$62.4_{.10}$	$\mathbf{32.0}_{.87}$	$70.5_{.20}$	1458_{19}	$58.3_{.14}$	$86.2_{.15}$	65.9_{66}	$59.1_{.65}$	$78.7_{.09}$	$66.0_{.57}$	$49.6_{1.4}$	$\mathbf{67.1}_{.09}$	$35.0_{.65}$	$61.8_{.17}$
2	13824	133%	58.8ms	6.40	14.2	$61.9_{.45}$	$31.5_{1.0}$	$70.8_{.49}$	1462_{24}	$\mathbf{58.5}_{.02}$	$\mathbf{86.4}_{.12}$	$\mathbf{66.4}_{.33}$	$\mathbf{59.6}_{.47}$	$78.9_{.02}$	$65.3_{.46}$	$49.5_{.97}$	$66.7_{.23}$	$35.1_{.87}$	$\mathbf{61.8}_{.01}$
Base [27]	18432	100%	68.5ms	8.26	15.3h	$\mathbf{62.6}_{.49}$	$31.9_{1.0}$	$70.8_{.59}$	$\mathbf{1467}_{13}$	$58.3_{.15}$	$86.1_{.24}$	$65.3_{.93}$	$59.4_{.92}$	$\mathbf{78.9}_{.37}$	$65.5_{.56}$	$49.8_{.6}$	$66.7_{.25}$	$35.1_{.86}$	$61.8_{.32}$

Table 5: **Training MLLMs with *Visual Context Compressor* in various compression levels**. We report the average results across three runs, with the standard deviation written at the bottom right of the average result. In the heavy compression range, the performance is inversely proportional to the compression ratio. In the light compression range, the performance is not sensitive to compression. *Performance remains high for models at the light compression level.*

Scalability to Larger Models. As modern multimodal LLMs (MLLMs) continue to grow in size and complexity, it is crucial to determine whether the performance gains observed in smaller models can be extended to larger architectures. This ablation allows us to verify if our compression strategies maintain or even enhance their effectiveness as the model scales, ensuring their applicability to more complex real-world scenarios. As demonstrated in Tab. 6, our four-stage scheme achieved comparable performance with standard training while saving 16%(21.1 vs 17.6) training time.

Model	#Tokens	CR	Train Time	GQA	MMVet	SQA	MME	VQA^T	POPE	MMB	MMB^{CN}	VQA^{v2}	LLaVA^w	VisWiz	SEED^I	MMMU	Avg.
LLaVA-13b	18432	100%	21.1h	63.0	35.0	74.1	1503	57.0	86.6	68.2	63.5	79.6	71.0	53.6	66.4	37.9	63.9
Ours	10863	170%	17.6h	63.0	35.4	74.2	1502	56.7	86.8	68.0	63.3	79.7	71.3	53.8	66.4	37.8	64.0

Table 6: **Training larger MLLMs with *LLaVolta***.Our method achieves comparable performance across various benchmarks while reducing the training time by 16% (21.1 hours vs. 17.6 hours) and increasing the compression ratio to 170%. These results demonstrate the scalability of our approach to larger models

Compairson with Layer-wise progressive Compression. Given the success of stage-wise compression in accelerating training, we hypothesize that it's also beneficial for layer-wise progressive compression. To explore this, we applied nested compressors with varying strides across layers, with smaller strides in the shallower layers, where visual tokens receive more attention. As shown in Tab. 7, we experimented with a multi-stage configuration: layers 0-3 with stride=1, layers 4-11 with stride=2, layers 12-23 with stride=4, and layers 24-31 with stride=8(CR=267%). This was compared to a single-stage compression setup: layer=8, stride=8(CR=266%). While the progressive layer-wise compression showed superior performance in direct inference, it underperformed when retrained. We

https://doi.org/10.52202/079017-2353

attribute this to the compounded pooling of visual tokens across layers, which imposes additional challenges on the model's learning, ultimately leading to suboptimal retraining outcomes.

Compressor	CR	GQA	MM-Vet	SQA	MME	VQA^T	POPE	MMB	MMB^{CN}	VQA^{v2}	LLaVA^W	VisWiz	SEED^I	MMMU	Avg.
Direct Inference															
Sinlge Stage	267%	57.8	25.3	70.2	1337	52.1	86.0	60.4	52.2	74.6	56.0	48.1	58.3	33.3	57.0
Multi Stage	266%	60.7	28.9	70.3	1403	55.4	85.1	65.2	57.1	77.7	60.6	49.1	64.8	35.2	60.0
Inference with Re-train															
Single Stage	267%	60.7	30.7	71.3	1456	56.9	86.4	64.6	58.0	77.9	67.0	48.8	66.0	35.3	61.3
Multi Stage	266%	60.9	29.5	70.5	1408	55.9	84.8	65.4	57.4	76.6	61.1	48.9	64.7	34.9	60.2

Table 7: **Comparison between single stage compressor and multi stage compressor.** mMlti-stage compression outperforming single-stage in direct inference across most tasks. However, in retrained models, multi-stage compression only shows marginal improvements, with a slight increase in the average performance.

Furthermore, we conduct an ablation study on the number of iterations in different stages (uniform *vs*. non-uniform stage splitting), which is detailed in the Appendix.

4.5 Extensibility to Video MLLMs

We extend our training scheme to VideoLLaVA [26] and the results in Tab. 8 reveal similar findings as before: the proposed training scheme achieve competitive results while reducing 9% training time. It is worth mentioning VideoLLaVA does not support DeepSpeed ZeRO-3, unlike LLaVA, which results in different relative efficiency gains.

#Stages	Scheme	#Tokens†	CR†	TFLOPs†	Train-time	MSVD-QA		MSRVTT-QA		ActivityNet-QA		Average	
						Score	Acc	Score	Acc	Score	Acc	Score	Acc
Single	no compression	147456	-	29.68	40.7h	3.69	69.1	3.48	56.8	3.28	47.5	3.48	57.8
Two	compression	80496	183%	17.73	37.1h	3.71	69.0	3.50	56.9	3.29	47.9	3.50	57.9
Three	compr. deeper	84776	174%	17.29	37.1h	3.73	69.3	**3.51**	**57.2**	3.28	47.4	**3.51**	58.0
Three	compr. wider	83256	177%	16.86	37.0h	3.72	69.0	**3.51**	**57.2**	**3.29**	47.7	**3.51**	58.0
Four	wider then deeper	88704	166%	18.32	37.2h	3.72	69.1	**3.51**	**57.2**	3.27	**48.0**	3.50	58.1
Four	deeper then wider	86904	170%	18.64	37.1h	**3.74**	**69.8**	3.49	56.9	3.27	47.8	3.50	**58.2**

Table 8: **Performance of *LLaVolta* on VideoLLaVA[26]**. See the definition of each training scheme in Tab. 1. †: average across stages. To implement our multi-stage training, we apply the same compression processing to the 8 frames representing the video respectively. *The derived six training schemes achieve competitive results while reducing 9% training time.*

5 Conclusion

In this work, we conduct two initial studies to investigate and verify the redundancy of visual tokens in multi-modal LLMs. To address this, we propose *Visual Context Compressor*, a straightforward yet effective compression technique that employs a simple average pooler, seamlessly integrating into the training of MLLMs. This approach enhances training efficiency without compromising performance. To further mitigate the information loss brought by the token compression, we introduce *LLaVolta*, a multi-stage training scheme that utilizes *Visual Context Compressor* with a progressively decreasing compression rate. Experimental results on various visual question answering benchmarks verify the effectiveness of *LLaVolta* in boosting performance while demonstrating efficiency gains by reducing training costs by 16% and inference latency by 24%. To the best of our knowledge, we are the first to accelerate the training of multi-modal LLM from the compression perspective. We hope that the proposed *Visual Context Compressor* and *LLaVolta* will inspire more in-depth analysis of visual redundancy existing in current MLLMs and call for future designs of efficient training for MLLMs.

Acknowledgement: We thank Zhanpeng Zeng for the discussions regarding the comparison with VCC. We are also grateful for the insightful advice from our anonymous reviewers. This work was supported by a Siebel Scholarship and ONR with N00014-23-1-2641.

References

[1] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[2] C. Anil, Y. Wu, A. Andreassen, A. Lewkowycz, V. Misra, V. Ramasesh, A. Slone, G. Gur-Ari, E. Dyer, and B. Neyshabur. Exploring length generalization in large language models. *arXiv preprint arXiv:2207.04901*, 2022.

[3] P. Baldi. Autoencoders, unsupervised learning, and deep architectures. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pages 37–49. JMLR Workshop and Conference Proceedings, 2012.

[4] H. Barlow. Redundancy reduction revisited. *Network: computation in neural systems*, 12(3):241, 2001.

[5] D. Chen and W. 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*, pages 190–200, 2011.

[6] J. Chen, J. Mei, X. Li, Y. Lu, Q. Yu, Q. Wei, X. Luo, Y. Xie, E. Adeli, Y. Wang, et al. Transunet: Rethinking the u-net architecture design for medical image segmentation through the lens of transformers. *Medical Image Analysis*, 97:103280, 2024.

[7] J. Chen, Q. Yu, X. Shen, A. Yuille, and L.-C. Chen. Vitamin: Designing scalable vision models in the vision-language era. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[8] J.-N. Chen, S. Sun, J. He, P. H. Torr, A. Yuille, and S. Bai. Transmix: Attend to mix for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12135–12144, 2022.

[9] L. Chen, H. Zhao, T. Liu, S. Bai, J. Lin, C. Zhou, and B. Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. *arXiv preprint arXiv:2403.06764*, 2024.

[10] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

[11] Z. Dai, G. Lai, Y. Yang, and Q. Le. Funnel-transformer: Filtering out sequential redundancy for efficient language processing. *Advances in neural information processing systems*, 33:4271–4282, 2020.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[13] C. Fu, P. Chen, Y. Shen, Y. Qin, M. Zhang, X. Lin, Z. Qiu, W. Lin, J. Yang, X. Zheng, et al. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

[14] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017.

[15] D. Gurari, Q. Li, A. J. Stangl, A. Guo, C. Lin, K. Grauman, J. Luo, and J. P. Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018.

[16] J. He, J.-N. Chen, S. Liu, A. Kortylewski, C. Yang, Y. Bai, and C. Wang. Transfg: A transformer architecture for fine-grained recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 852–860, 2022.

[17] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

[18] L. Hou, R. Y. Pang, T. Zhou, Y. Wu, X. Song, X. Song, and D. Zhou. Token dropping for efficient bert pretraining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

 https://doi.org/10.52202/079017-2353

[19] X. Huang, A. Khetan, R. Bidart, and Z. Karnin. Pyramid-bert: Reducing complexity via successive core-set based token selection. *arXiv preprint arXiv:2203.14380*, 2022.

[20] D. A. Hudson and C. D. Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019.

[21] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE transactions on pattern analysis and machine intelligence*, 24(7):881–892, 2002.

[22] B. Li, R. Wang, G. Wang, Y. Ge, Y. Ge, and Y. Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *arXiv preprint arXiv:2307.16125*, 2023.

[23] J. Li, D. Li, S. Savarese, and S. Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[24] Y. Li, Y. Du, K. Zhou, J. Wang, W. X. Zhao, and J.-R. Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023.

[25] Y. Li, Y. Zhang, C. Wang, Z. Zhong, Y. Chen, R. Chu, S. Liu, and J. Jia. Mini-gemini: Mining the potential of multi-modality vision language models, 2024.

[26] B. Lin, B. Zhu, Y. Ye, M. Ning, P. Jin, and L. Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[27] H. Liu, C. Li, Y. Li, and Y. J. Lee. Improved baselines with visual instruction tuning, 2023.

[28] H. Liu, C. Li, Q. Wu, and Y. J. Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024.

[29] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang. Lost in the middle: How language models use long contexts. *arXiv preprint arXiv:2307.03172*, 2023.

[30] Y. Liu, H. Duan, Y. Zhang, B. Li, S. Zhang, W. Zhao, Y. Yuan, J. Wang, C. He, Z. Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023.

[31] P. Lu, S. Mishra, T. Xia, L. Qiu, K.-W. Chang, S.-C. Zhu, O. Tafjord, P. Clark, and A. Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 2022.

[32] P. Nawrot, J. Chorowski, A. Łańcucki, and E. M. Ponti. Efficient transformers with dynamic token pooling. *arXiv preprint arXiv:2211.09761*, 2022.

[33] OpenAI. ChatGPT. `https://openai.com/blog/chatgpt/`, 2023.

[34] OpenAI. Gpt-4 technical report, 2023.

[35] G. Qin and B. Van Durme. Nugget: Neural agglomerative embeddings of text. In *International Conference on Machine Learning*, pages 28337–28350. PMLR, 2023.

[36] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[37] J. W. Rae, A. Potapenko, S. M. Jayakumar, and T. P. Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*, 2019.

[38] S. Shen, P. Walsh, K. Keutzer, J. Dodge, M. Peters, and I. Beltagy. Staged training for transformer language models. In *International Conference on Machine Learning*, pages 19893–19908. PMLR, 2022.

[39] A. Singh, V. Natarajan, M. Shah, Y. Jiang, X. Chen, D. Batra, D. Parikh, and M. Rohrbach. Towards vqa models that can read. In *CVPR*, 2019.

[40] G. Team, R. Anil, S. Borgeaud, Y. Wu, J.-B. Alayrac, J. Yu, R. Soricut, J. Schalkwyk, A. M. Dai, A. Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

[41] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, et al. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*, 2024.

[42] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, P. Tafti, L. Hussenot, P. G. Sessa, A. Chowdhery, A. Roberts, A. Barua, A. Botev, A. Castro-Ros, A. Slone, A. Héliou, A. Tacchetti, A. Bulanova, A. Paterson, B. Tsai, B. Shahriari, C. L. Lan, C. A. Choquette-Choo, C. Crepy, D. Cer, D. Ippolito, D. Reid, E. Buchatskaya, E. Ni, E. Noland, G. Yan, G. Tucker, G.-C. Muraru, G. Rozhdestvenskiy, H. Michalewski, I. Tenney, I. Grishchenko, J. Austin, J. Keeling, J. Labanowski, J.-B. Lespiau, J. Stanway, J. Brennan, J. Chen, J. Ferret, J. Chiu, J. Mao-Jones, K. Lee, K. Yu, K. Millican, L. L. Sjoesund, L. Lee, L. Dixon, M. Reid, M. Mikuła, M. Wirth, M. Sharman, N. Chinaev, N. Thain, O. Bachem, O. Chang, O. Wahltinez, P. Bailey, P. Michel, P. Yotov, R. Chaabouni, R. Comanescu, R. Jana, R. Anil, R. McIlroy, R. Liu, R. Mullins, S. L. Smith, S. Borgeaud, S. Girgin, S. Douglas, S. Pandya, S. Shakeri, S. De, T. Klimenko, T. Hennigan, V. Feinberg, W. Stokowiec, Y. hui Chen, Z. Ahmed, Z. Gong, T. Warkentin, L. Peran, M. Giang, C. Farabet, O. Vinyals, J. Dean, K. Kavukcuoglu, D. Hassabis, Z. Ghahramani, D. Eck, J. Barral, F. Pereira, E. Collins, A. Joulin, N. Fiedel, E. Senter, A. Andreev, and K. Kenealy. Gemma: Open models based on gemini research and technology, 2024.

[43] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[44] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, D. Bikel, L. Blecher, C. C. Ferrer, M. Chen, G. Cucurull, D. Esiobu, J. Fernandes, J. Fu, W. Fu, B. Fuller, C. Gao, V. Goswami, N. Goyal, A. Hartshorn, S. Hosseini, R. Hou, H. Inan, M. Kardas, V. Kerkez, M. Khabsa, I. Kloumann, A. Korenev, P. S. Koura, M.-A. Lachaux, T. Lavril, J. Lee, D. Liskovich, Y. Lu, Y. Mao, X. Martinet, T. Mihaylov, P. Mishra, I. Molybog, Y. Nie, A. Poulton, J. Reizenstein, R. Rungta, K. Saladi, A. Schelten, R. Silva, E. M. Smith, R. Subramanian, X. E. Tan, B. Tang, R. Taylor, A. Williams, J. X. Kuan, P. Xu, Z. Yan, I. Zarov, Y. Zhang, A. Fan, M. Kambadur, S. Narang, A. Rodriguez, R. Stojnic, S. Edunov, and T. Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[45] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[46] G. K. Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[47] G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis. Efficient streaming language models with attention sinks. *arXiv preprint arXiv:2309.17453*, 2023.

[48] J. Xu, T. Mei, T. Yao, and Y. 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.

[49] X. Ye, A. Wang, J. Choi, Y. Lu, S. Sharma, L. Shen, V. Tiyyala, N. Andrews, and D. Khashabi. AnaloBench: benchmarking the identification of abstract and long-context analogies. *arXiv preprint arXiv:2402.12370*, 2024.

[50] W. Yu, Z. Yang, L. Li, J. Wang, K. Lin, Z. Liu, X. Wang, and L. Wang. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*, 2023.

[51] Z. Yu, D. Xu, J. Yu, T. Yu, Z. Zhao, Y. Zhuang, and D. Tao. Activitynet-qa: A dataset for understanding complex web videos via question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9127–9134, 2019.

[52] X. Yue, Y. Ni, K. Zhang, T. Zheng, R. Liu, G. Zhang, S. Stevens, D. Jiang, W. Ren, Y. Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9556–9567, 2024.

[53] Z. Zeng, C. Hawkins, M. Hong, A. Zhang, N. Pappas, V. Singh, and S. Zheng. Vcc: Scaling transformers to 128k tokens or more by prioritizing important tokens. *Advances in Neural Information Processing Systems*, 36, 2024.

[54] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023.

 https://doi.org/10.52202/079017-2353

Appendix

In the appendix, we provide additional information as listed below:

- § A provides the additional experimental results.
- § B provides the dataset information and licenses.

A Additional Experimental Results

A.1 Non-uniform Stage Splitting

By default, the training time is evenly divided across each stage. To explore how the compression stage affects total training time, we modify the relative proportion of different stages. This variation is tested in the two-stage setup referenced in Tab. 1, adjusting from the standard 50% in Stage 1 and 50% in Stage 2 to different distributions. Tab. 9 below displays the results of these experiments.

Stage 1	Stage 2	#Tokens	CR	GQA	MMVet	SQA	MME	VQA^T	POPE	MMB	MMB^{CN}
0%	100%	18432	-	62.0	31.1	70.1	1453.0	58.2	85.9	64.3	58.3
25%	75%	11088	166%	62.1	31.7	70.6	1474.5	58.8	86.4	65.1	59.6
50%	50%	10863	170%	62.2	30.0	70.3	1443.5	57.5	85.8	64.8	59.7
75%	25%	10597	174%	61.6	32.2	70.8	1471.5	57.5	86.6	65.2	58.9
90%	10%	10407	177%	61.2	31.0	70.5	1447.5	56.3	86.4	64.4	56.9
100%	0%	10062	183%	55.9	29.5	64.1	1257.8	49.1	86.6	47.4	29.2

Table 9: **Effects of non-uniform stage splitting at the two-stage set-up**. Performance decreases as the proportion of Stage 2 decreases, albeit at the expense of lower compression ratios.

We observe that as the Stage 2 increases from 0% to 100%, there is a gradual decrease in the model's performance across various metrics (such as GQA, MMVet, SQA, MME, VQA, POPE, MMB, and MMB^{CN}). Although there is a decline in performance, it is relatively minor when the compression stage makes up to 50% of the training duration. However, when the proportion of the compression stage is reduced below 50%, the decline in performance becomes more significant. In conclusion, keeping the compression stage between 0-50% of the training time minimizes performance loss while still achieving significant compression ratios.

A.2 Adaptability to Different Structures.

In addition to scaling across model sizes, it is essential to evaluate the adaptability of our approach to different model structures. As shown in Tab. 10, we conduct an experiment on Mini-Gemini [25], a structurally distinct baseline. Since Mini-Gemini employs a multi-resolution visual encoding strategy and Gemma [42] as language model. This ablation experiment assesses *LLaVolta*'s compatibility with different sophisticated visual encoding strategies.

Model	#Tokens	CR	Train Time	GQA	MMVet	SQA	MME	VQA^T	POPE	MMB	MMB^{CN}	VQA^{v2}	LLaVA^w	VisWiz	SEED^I	MMMU	Avg.
MGM-2B	18432	100%	18.1h	60.7	30.1	62.7	1327	57.1	86.0	61.9	50.6	76.3	65.9	48.3	63.8	28.1	58.3
Ours	10863	170%	14.8h	58.8	30.2	62.2	1325	54.3	87.0	62.5	52.5	76.3	65.7	48.9	63.1	27.3	58.1

Table 10: **Training struturally distinct MLLMs with *LLaVolta***.Comparison of our method with the Mini-Gemini (MGM-2B) baseline, which uses a multi-resolution visual encoding strategy. Our approach demonstrates competitive performance while reducing training time by 18% (18.1 hours vs. 14.8 hours) and achieving higher scores. This ablation highlights *LLaVolta*'s ability to adapt to different model structures and sophisticated visual encoding strategies.

https://doi.org/10.52202/079017-2353

B Datasets Information and Licenses

GQA: The GQA: [20] dataset, consists of 22M questions about various day-to-day images.

License: N/A

Dataset website: https://cs.stanford.edu/people/dorarad/gqa/download.html

MM-Vet: The MM-Vet [50] dataset, defining 6 core VL capabilities and examines the 16 integrations of interest derived from the capability combination.

License: Apache License. https://github.com/yuweihao/MM-Vet/blob/main/LICENSE

Dataset website: https://github.com/yuweihao/MM-Vet/tree/main

SQA: The SQA: [31] dataset, consisting of 21k multimodal multiple choice questions with a diverse set of science topics and annotations of their answers with corresponding lectures and explanations.

License: CC BY-NC-SA (Attribution-NonCommercial-ShareAlike) https://creativecommons.org/licenses/by-nc-sa/4.0/

Dataset website: https://scienceqa.github.io/#download

POPE: The POPE [24] dataset can evaluate the object hallucination in a more stable and flexible way.

License: MIT License. https://github.com/RUCAIBox/POPE?tab=MIT-1-ov-file#readme

Dataset website: https://github.com/RUCAIBox/POPE

MMBench: The MMBench [30] dataset is a collection of benchmarks to evaluate the multi-modal understanding capability of large vision language models.

License: Apache License. https://github.com/open-compass/MMBench?tab=Apache-2.0-1-ov-file#readme

Dataset website: https://github.com/open-compass/MMBench

MMBench-CN: The MMBench-CN [30] dataset is a collection of benchmarks to evaluate the multi-modal understanding capability of large vision language models.

License: Apache License. https://github.com/open-compass/MMBench?tab=Apache-2.0-1-ov-file#readme

Dataset website: https://github.com/open-compass/MMBench

MME: The MME [13] dataset containing 30 images with 60 instruction-answer pairs for coarse-grained recognition task; 917 images for fine-grained recognition task; 20 images with 40 instruction-answer pairs for OCR task.

Dataset website: https://github.com/QwenLM/Qwen-VL/blob/master/eval_mm/mme/EVAL_MME.md

License: Tongyi Qianwen LICENSE AGREEMENT. https://github.com/QwenLM/Qwen-VL/tree/master?tab=License-1-ov-file#readme

TextVQA: The TextVQA [39] dataset containing 30 images with 60 instruction-answer pairs for coarse-grained recognition task; 917 images for fine-grained recognition task; 20 images with 40 instruction-answer pairs for OCR task.

Dataset website: https://github.com/facebookresearch/mmf.git

License: BSD LICENSE. https://github.com/facebookresearch/mmf/blob/main/LICENSE

VQA-v2: The VQA-v2 [14] dataset, containing 265,016 images, dataset containing open-ended questions about images. These questions require an understanding of vision, language and commonsense knowledge to answer.

https://doi.org/10.52202/079017-2353

Dataset website: https://visualqa.org/

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: See Sec. 1 for the main claims and Sec. 3.2 and Sec. 4 for the detailed contributions and scope.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See Sec. 5 for the discussion on the limitations.

   Guidelines:

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

3. **Theory Assumptions and Proofs**

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

   Answer: [NA]

https://doi.org/10.52202/079017-2353

Justification: The paper does not include theoretical results.

Guidelines:

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

4. **Experimental Result Reproducibility**

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

   Answer: [Yes]

   Justification: See Sec. 4.1 for the information needed to reproduce the main experimental results.

   Guidelines:

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

5. **Open access to data and code**

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

Answer: [Yes]

Justification: We open-sourced the full code through github repo: `https://github.com/Beckschen/LLaVolta`.

Guidelines:

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

6. **Experimental Setting/Details**

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

   Answer: [Yes]

   Justification: See Sec. 4 for all the training and test details.

   Guidelines:

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

7. **Experiment Statistical Significance**

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

   Answer:[Yes]

   Justification: See Sec. 4.3, we report our experiment results with statistics by running 3 times and calculating mean and standard deviation.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

https://doi.org/10.52202/079017-2353

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

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

   Answer: [Yes]

   Justification: See Sec. 4 for the computer resources needed to reproduce the experiments.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader Impacts**

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

    Answer: [Yes]

    Justification: See Sec. 5 for the discussion on the broader impacts.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

    Answer: [Yes]

    Justification: See Sec. 5 for the discussion on the safeguards.

    Guidelines:

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

12. **Licenses for existing assets**

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

    Answer: [Yes]

    Justification: See Sec. B for dataset licenses.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

https://doi.org/10.52202/079017-2353

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

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

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

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

14. **Crowdsourcing and Research with Human Subjects**

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

    Answer: [NA]

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

    Guidelines:

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

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

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

    Answer: [NA]

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

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.