Efficient Multi-modal Large Language Models via Progressive Consistency Distillation

Zichen Wen[1,2] **Shaobo Wang**[1] **Yufa Zhou**[3] **Junyuan Zhang**[4] **Qintong Zhang**[5]
Yifeng Gao[1] **Zhaorun Chen**[6] **Bin Wang**[2] **Weijia Li**[7,2]
Conghui He[2*] **Linfeng Zhang**[1*]
[1]EPIC Lab, Shanghai Jiao Tong University [2]Shanghai AI Laboratory
[3]Duke University [4]The University of Hong Kong
[5]Peking University [6]University of Chicago [7]Sun Yat-sen University
heconghui@pjlab.org.cn, zhanglinfeng@sjtu.edu.cn

Code: https://github.com/ZichenWen1/EPIC

Abstract

Visual tokens consume substantial computational resources in multi-modal large models (MLLMs), significantly compromising their efficiency. Recent works have attempted to improve efficiency by compressing visual tokens during training, either through modifications to model components or by introducing additional parameters. However, they often overlook the increased learning difficulty caused by such compression, as the model's parameter space struggles to quickly adapt to the substantial perturbations in the feature space induced by token compression. In this work, we propose to develop **E**fficient MLLMs via **P**rogress**I**ve **C**onsistency Distillation (`EPIC`), a progressive learning framework. Specifically, by decomposing the feature space perturbations introduced by token compression along the token-wise and layer-wise dimensions, we introduce token consistency distillation and layer consistency distillation, respectively, aiming to reduce the training difficulty by leveraging guidance from a teacher model and following a progressive learning trajectory. Extensive experiments demonstrate the superior effectiveness, robustness, and generalization capabilities of our proposed framework.

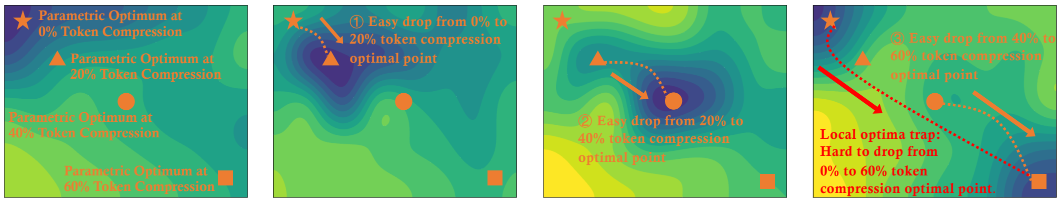
1 Introduction

Multi-modal large language models (MLLMs) [32, 81, 39, 42] equip large language models (LLMs) [54, 23] with the ability to understand visual information, exhibiting remarkable capabilities across a diverse range of multi-modal tasks, including image captioning, visual question answering (VQA), video understanding [61], and multi-modal reasoning [62].

However, unlike LLMs [54, 2, 66, 59], which only need to process a small number of information-dense text tokens [48], the introduction of substantial visual tokens in MLLMs [3, 4] presents significant computational challenges. This issue becomes particularly pronounced when handling high-resolution images [35] or multi-frame videos [53].

To address this issue, a natural approach is to reduce visual tokens [44, 65, 6, 67]. Recent advances have introduced various token compression techniques to eliminate vision tokens in a training-free approach [78, 63]. Most of them adopt either a token importance-based strategy [7, 24] or a token redundancy-based strategy [57, 52]. Although the aforementioned non-parametric methods avoid additional training costs, they inevitably incur significant performance degradation. To achieve

*Corresponding authors.


Parametric Optimum at 0% Token Compression
Parametric Optimum at 20% Token Compression
Parametric Optimum at 40% Token Compression
Parametric Optimum at 60% Token Compression
① Easy drop from 0% to 20% token compression optimal point
② Easy drop from 20% to 40% token compression optimal point
③ Easy drop from 40% to 60% token compression optimal point
Local optima trap: Hard to drop from 0% to 60% token compression optimal point.

(a) Loss Landscape of Compression Ratio = 0% (b) Loss Landscape of Compression Ratio = 20% (c) Loss Landscape of Compression Ratio = 40% (d) Loss Landscape of Compression Ratio = 60%

Figure 1: Progressive Consistency Distillation vs. Direct Training. Each subplot shows the loss landscape under the corresponding token compression ratio, with the **optimum** indicated. Our method reaches the objective via **progressive** learning trajectories, while **direct** training remains challenging.

a better performance-efficiency trade-off, recent training-aware token compression methods have attracted significant attention [5, 14]. Beyond early approaches [32] that employed Q-former, MQT-LLaVA [26] proposes a more flexible Q-former capable of dynamically encoding visual information into variable-length visual tokens. [33] proposes TokenPacker, a coarse-to-fine visual projector that progressively refines coarse tokens with visual information. In addition to refining the model architecture and upgrading model components, VoCo-LLaMA [70] and LLaVA-Mini [76] build upon observations of the transfer of visual information within language models, attempting to transfer visual information into a small set of VoCo tokens via attention modification, or into textual tokens with a transformer-based pre-fusion module.

Although these methods improve model efficiency while largely preserving performance, their improvements primarily come from architectural enhancements or newly introduced modules, often overlooking the learning challenges posed by token compression during training. As illustrated in Figure 1, when token compression is applied during training, the distribution of the compressed token sequence inevitably differs from that of the full token set. This discrepancy can be regarded as a perturbation in the feature space, which shifts the model's optimal point in the parameter space. The higher the compression ratio, the greater the perturbation introduced, and consequently, the further the optimal point drifts. The goal of training-aware token compression is to guide the model to progressively adapt from the original optimum (under full tokens) to a new optimum corresponding to the compressed token distribution. Figure 1 (d) shows that directly training a model with compressed tokens often leads to suboptimal solutions, as the optimization process can easily get trapped in poor local minima, making it difficult to reach the desired optimum under heavy compression.

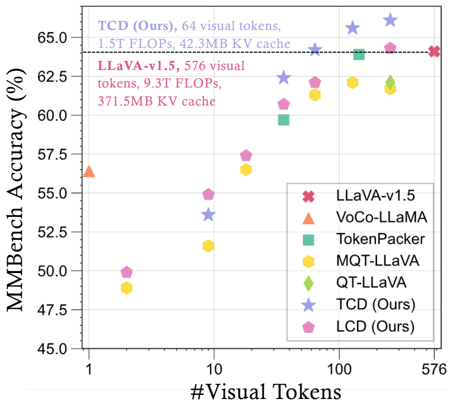


Figure 2: MMBench accuracy vs. number of visual tokens for various methods. TCD (Ours) and LCD (Ours) achieve competitive accuracy with far fewer tokens, lower FLOPs, and smaller KV cache compared to LLaVA-v1.5, highlighting its efficiency.

In this work, we propose a progressive consistency distillation learning framework `EPIC` tailored for token compression, where a single MLLM simultaneously acts as both teacher and student through weight sharing. From a token-wise perspective, we introduce **Token Consistency Distillation (TCD)**. At the early stages of training, both the teacher and student models adopt a very low token compression ratio, indicating a relatively easy learning task without significant optimal point drifts (as shown in Fig. 1 (b)). As training progresses, the compression becomes increasingly aggressive, forming a progressive learning trajectory (see Fig. 1 (b) to (d)). Although the final optimal point under heavy compression remains far from the initial optimum, each intermediate optimum along the trajectory is relatively close to its predecessor, making each transition more manageable and easier to optimize. Moreover, the teacher consistently uses a slightly lower compression ratio than the student, introducing a compression ratio gap between them. We argue that when the gap is too large, the student may struggle to benefit effectively from the teacher's guidance [20]. Therefore, the compression ratio gap is also designed to follow a progressive learning strategy, gradually increasing over time to ease the learning process. From a layer-wise perspective, we introduce **Layer Consistency Distillation (LCD)**. Based on observations from prior work [7], the significance of visual tokens diminishes notably in deeper layers, suggesting that compressing tokens at these layers has minimal impact on the model's feature space and output. Therefore, in LCD, the token compression is progressively shifted from deeper to shallower layers as training progresses, implicitly following an easy-to-hard learning paradigm. Meanwhile, a compression gap between the teacher and the student is maintained to encourage effective guidance from the teacher.

Our primary contribution lies in proposing `EPIC`, a progressive consistency distillation learning framework, which demonstrates compatibility with diverse token compression techniques. Within this framework, we introduce Token Consistency Distillation and Layer Consistency Distillation from token-wise and layer-wise dimensions, respectively. This approach enables the training of robust and highly generalizable models through a progressive learning strategy without requiring modifications to the model architecture. Compared to prior approaches, comprehensive experiments validate the superior effectiveness, robustness, and generalization capabilities of our proposed framework.

2 Related Works

Multi-modal Large Language Models. Multi-modal Large Language Models (MLLMs), pioneered by [41, 81, 15, 71, 8] have successfully showcased promising results on a wide variety of vision-language perception and reasoning tasks. Existing MLLMs typically employ a pre-trained vision encoder (*e.g.*, CLIP [49] and SigLIP [73, 55]) to extract visual features, which are then projected into the LLM's input space via a visual projector (*e.g.*, MLP and Q-former [32]), enabling the model to process both visual embeddings and user instructions for multimodal understanding and response generation. Recent studies [75, 74] have highlighted the limitations of MLLMs in fine-grained visual perception tasks. To this end, more advanced MLLMs have attempted to increase the number of encoded visual tokens by employing dynamic resolutions [11, 10, 40, 31] or arbitrary resolutions [58, 4] to process high-resolution images, thereby enhancing their performance in visual understanding tasks. However, due to the quadratic complexity of the attention mechanism [56], the resulting longer token sequences pose significant challenges to both inference speed and memory usage.

Visual Token Compression. Visual tokens typically outnumber text tokens by orders of magnitude while containing greater spatial redundancy than information-dense text [48]. Recent work has explored both training-free [24, 79, 80, 43, 77] and training-aware [5, 13, 3, 9, 60] compression approaches, with the latter showing superior performance potential despite requiring additional training [63]. Training-free methods generally follow two paradigms: importance-based strategies like FastV [7] and SparseVLM [78] that use attention scores, and redundancy-based approaches such as DART [64] and G-Prune [28] that assess token similarity. Early training-aware work focused on parameter-free strategies, including LLaVA-PruMerge's attention-based merging [50] and VoCo-LLaMA's compressed VoCo tokens [70]. Subsequent research explored architectural modifications through lightweight component replacements [16, 19, 34]. More advanced approaches include MQT-LLaVA's dynamic Q-former for variable-length token encoding [26] and TokenPacker's coarse-to-fine iterative condensation [33]. Recent methods like LLaVA-Mini [76] achieve near-lossless token compression through extra auxiliary modules, though they often overlook the training challenges introduced by feature space perturbations during token compression.

3 Methodology

3.1 Preliminary

Multi-modal Large Language Models (MLLMs). An MLLM typically consists of three modules: a *visual encoder* VE, a *modality projector* (*e.g.*, MLP), and a *language model* LM. Given image $\mathcal{I}$, the visual encoder extracts patch-level features $\mathbf{z}_v \in \mathbb{R}^{N\times d_v}$, projected into visual tokens $\mathbf{e}_v \in \mathbb{R}^{N\times d_h}$:

$$\mathbf{e}_v = \mathrm{MLP}(\mathrm{VE}(\mathcal{I})), \tag{1}$$

where N is the number of image patches, d_v is the encoder output dimension, and d_h is the LM hidden size. Meanwhile, a text prompt $\mathcal{P}$ is tokenized and embedded into a sequence of text tokens $\mathbf{e}_t \in \mathbb{R}^{L\times d_h}$. The visual and text tokens are then concatenated to form the full input sequence:

$$\mathbf{x} = [\mathbf{e}_v; \mathbf{e}_t] \in \mathbb{R}^{(N+L)\times d_h}. \tag{2}$$

Positional embeddings are added to $\mathbf{x}$ to encode spatial and sequential structure. The language model then autoregressively generates output tokens $y_i \in \mathcal{V}$ (with vocabulary $\mathcal{V}$) one token at a time:

$$y_i = \mathrm{LM}(\mathbf{x}, y_{<i}), \quad \text{for } i = 0, 1, 2, \ldots \tag{3}$$

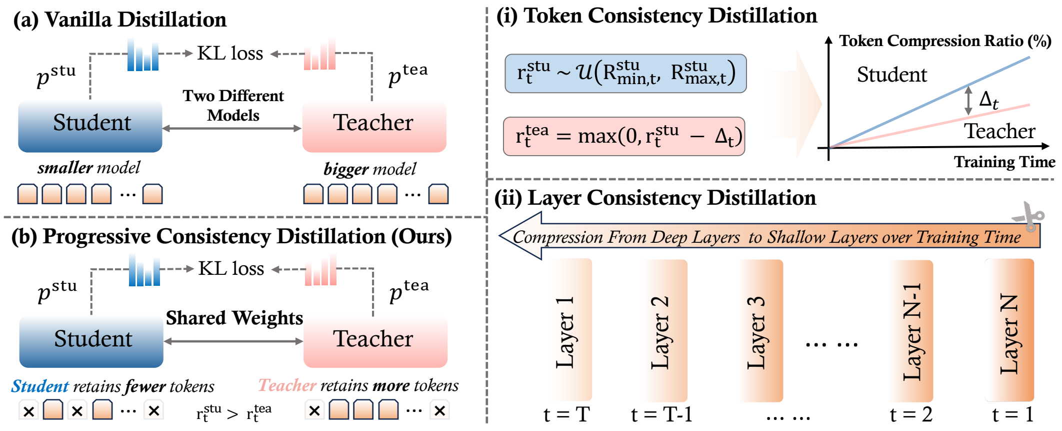

(a) Vanilla Distillation
KL loss
p^{stu}
p^{tea}
Two Different Models
Student
Teacher
smaller model
bigger model
(b) Progressive Consistency Distillation (Ours)
KL loss
p^{stu}
p^{tea}
Shared Weights
Student
Teacher
Student retains fewer tokens
Teacher retains more tokens
$r_t^{stu} > r_t^{tea}$
(i) Token Consistency Distillation
$r_t^{stu} \sim \mathcal{U}(R_{min,t}^{stu}, R_{max,t}^{stu})$
$r_t^{tea} = \max(0, r_t^{stu} - \Delta_t)$
Token Compression Ratio (%)
Student
Δ_t
Teacher
Training Time
(ii) Layer Consistency Distillation
Compression From Deep Layers to Shallow Layers over Training Time
Layer 1
Layer 2
Layer 3
… …
Layer N-1
Layer N
t = T
t = T-1
… …
t = 2
t = 1

Figure 3: **An overview of Progressive Consistency Distillation.** (i) Token Consistency Distillation progressively increases token compression ratio over time. (ii) Layer Consistency Distillation shifts token compression from deep to shallow layers, promoting layer-wise consistency during training.

where $y_{<i} := \{y_0, y_1, \dots, y_{i-1}\}$ denotes the previously generated tokens.

Progressive Consistency Distillation Learning. Training MLLMs under aggressive token compression introduces significant feature space perturbations that hinder convergence. To address this, we propose *Progressive Consistency Distillation Learning* (PCDL), which gradually increases compression difficulty and facilitates the model's progressive convergence to the final objective (see Fig. 1). Our **Token Consistency Distillation (TCD)** progressively increases the token compression ratio over training steps from a token-wise perspective, while **Layer Consistency Distillation (LCD)** initially applies token compression in deeper layers (minimal impact [7, 76]) before gradually shifting to shallower layers, performing progressive learning in a layer-wise manner. Both employ *Consistency Distillation with Shared Weights*, where teachers with slightly lower compression ratios (*e.g.*, 5% less) provide manageable guidance [20], progressively transitioning to stronger teachers (*e.g.*, 10% lower ratio) for staged mentorship.

3.2 Theoretical Intuition: A 1D Prototype for Progressive Consistency Distillation Learning

Scalar center path. To provide intuition for *Progressive Consistency Distillation Learning* (PCDL), we introduce a one-dimensional prototype where model predictions are scalar values $\theta \in \mathbb{R}$, and each target is given by a compression-dependent center $c_r \in \mathbb{R}$ for compression ratio $r \in [0, r_{\max}]$. We assume that the mapping $c : [0, r_{\max}] \to \mathbb{R}$ satisfies:

(S1) *Monotonicity:* c is differentiable with $c'_r \geq 0$ for all r.

(S2) *Lipschitz slope:* There exists $\gamma > 0$ such that $|c'_r| \leq \gamma$, and $|c_{r_1} - c_{r_2}| \leq \gamma |r_1 - r_2|$ for all r_1, r_2.

(S3) *Convexity:* c is convex, i.e., $c''(r) \geq 0$.

Two quadratic objectives. We compare two learning objectives:

$$\mathcal{L}_{\text{dir}}(r, \theta) = \tfrac{1}{2}(\theta - c_r)^2, \qquad \mathcal{L}_{\text{prog}}(r, \theta) = \tfrac{1}{2}(\theta - c_r)^2 + \tfrac{\lambda}{2}(\theta - c_{r-\Delta})^2,$$

with constants $0 < \lambda < 1$, $0 < \Delta \leq r_{\max}$. The second term acts as a regularizer pulls the prediction toward a slightly less compressed teacher target, mimicking the KL-based distillation loss in `EPIC`.

Analogy to EPIC. In `EPIC`, the teacher operates at a slightly lower compression ratio $(r - \Delta)$ and provides smoother targets via KL consistency. The scalar prototype captures this: $\mathcal{L}_{\text{dir}}$ corresponds to direct supervision from the current target c_r, while $\mathcal{L}_{\text{prog}}$ encodes progressive distillation by incorporating a past (easier) target $c_{r-\Delta}$. As in `EPIC`, the regularizer improves training stability by reducing sensitivity to abrupt changes in the input space induced by token compression.

Exact minimizers. Both objectives are strongly convex in θ with closed-form solutions:

$$\theta_r^{\text{dir}} = c_r, \qquad \theta_r^{\text{prog}} = \frac{c_r + \lambda\, c_{r-\Delta}}{1 + \lambda}.$$

Schedule and path length. Let $0 = r_0 < r_1 < \cdots < r_T = r_{\max}$ denote any compression schedule. The total variation of a path $\{x_t\}$ is defined as:

$$\mathrm{TV}(\{x_t\}) := \sum_{t=0}^{T-1} |x_{t+1} - x_t|.$$

We show below that the progressive path yields strictly shorter total variation, reflecting the smoother optimization trajectory induced by PCDL. We delay the full proof to Appendix B.

Theorem 1 (Scalar path gain, Proof in Appendix B)**.** *Under assumptions (S1)–(S3), the total variation of the progressive path is strictly smaller:*

$$\begin{aligned}
&\mathrm{TV}\big(\{\theta^{\mathrm{dir}}_{r_t}\}\big) \le \gamma r_{\max}, \\
&\mathrm{TV}\big(\{\theta^{\mathrm{prog}}_{r_t}\}\big) \le \frac{1+\lambda\kappa}{1+\lambda} \cdot \mathrm{TV}\big(\{\theta^{\mathrm{dir}}_{r_t}\}\big) \; < \; \mathrm{TV}\big(\{\theta^{\mathrm{dir}}_{r_t}\}\big), \\
&\textit{where} \qquad \kappa := \sup_{r\in[\Delta,\, r_{\max}-\Delta]} \frac{c(r) - c(r-\Delta)}{c(r+\Delta) - c(r)} \in [0, 1).
\end{aligned}$$

3.3 Token Consistency Distillation

We consider a single MLLM f_θ that plays both teacher and student roles by sharing parameters θ. Let $\mathcal{I}$ denote the input image and $\mathcal{P}$ denote the language prompt. Let $C(\mathcal{I}, r, \ell)$ be a token compression operator[1] that, at layer ℓ, compresses the visual tokens extracted from $\mathcal{I}$ with ratio $r \in [0, 1]$, *i.e.*, retaining only a fraction $1 - r$ of visual tokens. Let $t \in \{0, \ldots, T\}$ index training steps.

At iteration t, we sample a student compression ratio from a gradually shifting uniform distribution:

$$r_t^{\mathsf{stu}} \sim \mathcal{U}\big(R^{\mathsf{stu}}_{\min,t},\ R^{\mathsf{stu}}_{\max,t}\big), \tag{4}$$

where $R^{\mathsf{stu}}_{\max,t}$ linearly increases from $R^{\mathsf{stu}}_{\max,0} = \epsilon$ (*e.g.*, 5%) to $R^{\mathsf{stu}}_{\max,T} = R_{\max}$ (*e.g.*, 90%) with training steps, and $R^{\mathsf{stu}}_{\min,t}$ grows more slowly from 0% to at most 50% as training progresses. This creates an easy-to-hard curriculum for the student model by gradually narrowing and shifting the sampling range towards more aggressive compression.

We then define the teacher model's compression ratio as

$$r_t^{\mathsf{tea}} = \max\big(0,\ r_t^{\mathsf{stu}} - \Delta_t\big), \tag{5}$$

where the compression gap Δ_t also increases gradually from $\Delta_0 = \delta_{\min}$ to $\Delta_T = \delta_{\max}$ (*e.g.*, up to 30%), ensuring that the teacher consistently sees slightly less compressed inputs than the student.

Concretely, we form two forward passes through the shared model f_θ:

$$\mathbf{h}^{\mathsf{tea}} = f_\theta\big(C(\mathcal{I}, r_t^{\mathsf{tea}}, \ell); \mathcal{P}\big); \quad \mathbf{h}^{\mathsf{stu}} = f_\theta\big(C(\mathcal{I}, r_t^{\mathsf{stu}}, \ell); \mathcal{P}\big), \tag{6}$$

where both token compressions occur at the fixed Transformer layer ℓ.

We incorporate the Token Consistency Distillation (TCD) loss as an auxiliary objective alongside the main supervised fine-tuning (SFT) loss. Specifically, the overall training objective is formulated as:

$$\mathcal{L}_{\text{total}}(\theta) = (1-\lambda) \cdot \mathcal{L}_{\text{SFT}}(\theta) + \lambda \cdot \mathcal{L}_{\mathrm{TCD}}(\theta), \tag{7}$$

where $\mathcal{L}_{\text{SFT}}(\theta)$ denotes the autoregressive cross-entropy loss over language outputs, and λ is a balancing coefficient for the distillation loss, empirically set to 0.7.

The TCD loss is defined as the Kullback–Leibler (KL) divergence [29, 25] between the output logits distributions of the teacher and student:

$$\mathcal{L}_{\mathrm{TCD}}(\theta) = \mathbb{E}_{\mathcal{I},\mathcal{P},t}\left[\mathrm{KL}\left(p^{\mathsf{tea}} \parallel p^{\mathsf{stu}}\right)\right], \tag{8}$$

where $p^{\mathsf{tea}} = \mathrm{Softmax}(\mathbf{h}^{\mathsf{tea}}/\tau)$ and $p^{\mathsf{stu}} = \mathrm{Softmax}(\mathbf{h}^{\mathsf{stu}}/\tau)$ are the temperature-scaled output logits distributions from the teacher and student, respectively, and τ is a temperature hyperparameter.

By progressively increasing both the student's compression range $\big[R^{\mathsf{stu}}_{\min,t}, R^{\mathsf{stu}}_{\max,t}\big]$ and the teacher–student gap Δ_t, we implement an progressive learning in a token-wise manner. At early training stages, the student experiences mild compression and benefits from a closely aligned teacher, while in later stages, it endures stronger compression with increasingly distinct teacher guidance.

[1]Any plug-and-play token compressor (*e.g.*, FastV [7], DART [64]) can serve as compression operator here.

3.4 Layer Consistency Distillation

Prior work [7, 76] shows visual tokens receive negligible attention in deeper layers while shallow layers play a more critical role for visual modality. This motivates our **Layer Consistency Distillation (LCD)** strategy: performing compression in deeper layers first minimizes output perturbation and feature space distortion, then progressively moves to shallower layers.

Let L be the total number of Transformer layers in the language model. Define a normalized training progress: $\beta_t = \frac{t}{T}$. We then select a single compression layer for both teacher and student:

$$\ell_t = \mathrm{Round}\big(L - \beta_t(L - \ell_{\min})\big), \quad \ell_t \in \ell_{\min}, \ell_{\min}+1, \ldots, L \tag{9}$$

where $\ell_{\min}$ denotes the shallowest compression layer. Thus at $t = 0$ we compress at the deepest layer $\ell_0 = L$, and by $t = T$ at the shallowest layer $\ell_T = \ell_{\min}$, realizing a layer-wise progressive learning.

At training iteration t, we sample the student model's token compression ratio from $[r_{\min}, r_{\max}]$:

$$r_t^{\mathsf{stu}} \sim \mathcal{U}(r_{\min},\, r_{\max}), \tag{10}$$

Typically, $r_{\min}$ is set to 0.2, while $r_{\max}$ is configured as 0.9 and define the teacher model's compression ratio by subtracting the compression ratio gap Δ_t:

$$r_t^{\mathsf{tea}} = \max\big(0,\ r_t^{\mathsf{stu}} - \Delta_t\big). \tag{11}$$

We then perform two forward passes through the shared model f_θ at layer ℓ_t:

$$\mathbf{h}^{\mathsf{tea}} = f_\theta\big(C(\mathcal{I}, r_t^{\mathsf{tea}}, \ell_t); \mathcal{P}\big); \quad \mathbf{h}^{\mathsf{stu}} = f_\theta\big(C(\mathcal{I}, r_t^{\mathsf{stu}}, \ell_t); \mathcal{P}\big). \tag{12}$$

The Layer Consistency Distillation (LCD) loss is defined analogously to TCD, as the KL divergence between the teacher and student output logits distributions:

$$\mathcal{L}_{\mathrm{LCD}}(\theta) = \mathbb{E}_{\mathcal{I},\mathcal{P},t}\big[\mathrm{KL}(p^{\mathsf{tea}} \,\|\, p^{\mathsf{stu}})\big]; \quad p^{\mathsf{tea}} = \mathrm{Softmax}(\mathbf{h}^{\mathsf{tea}}/\tau), p^{\mathsf{stu}} = \mathrm{Softmax}(\mathbf{h}^{\mathsf{stu}}/\tau). \tag{13}$$

Finally, we integrate both distillation terms into the overall training objective:

$$\mathcal{L}_{\mathrm{total}}(\theta) = (1-\lambda)\cdot\mathcal{L}_{\mathrm{SFT}}(\theta) + \lambda\cdot\mathcal{L}_{\mathrm{LCD}}(\theta). \tag{14}$$

4 Experiments

4.1 Experimental Setting

Implementation Details. We implement `EPIC` based on LLaVA [41, 40] without introducing any modifications to the model architecture. Specifically, we adopt CLIP ViT-L/14 [49] as our vision encoder, utilizing its officially pretrained projector, and Vicuna-v1.5 [12] as our LLM. `algname` only requires performing the second stage training, which involves visual instruction tuning on the LLaVA-665K instruction fine-tuning dataset. To demonstrate the effectiveness and generalizability of our method, we incorporate three representative token compression techniques (DART [64], FastV [7], and Random token pruning) into `EPIC` for training. More implementation details about our proposed method and baselines are provided in Appendix D.

Evaluation Benchmarks. We evaluate our model across 10 representative visual understanding benchmarks. Further details about benchmarks can be found in the Appendix D.3.

Baselines. As shown in Table 1, we compare our method with various visual token compression techniques, including QT-LLaVA, MQT-LLaVA [26], LLaMA-VID [34], VoCo-LLaMA [70], TokenPacker [33], and LLaVA-Mini [76]. We also list other MLLM's results for comparison, including BLIP-2 [32], InstructBLIP [15], IDEFICS [30], Qwen-VL [3], Qwen-VL-Chat [3], SPHINX [38], mPLUG-Owl2 [68], and vanilla LLaVA [41, 40]. Please refer to Appendix D.4 for more details.

4.2 Experimental Results on Benchmarks

Table 1 presents experimental results on 10 representative visual benchmarks. Notably, our method and MQT-LLaVA are among the few approaches enabling flexible control over token compression ($36 \sim 256$ tokens) with a single trained model, adapting efficiently to varying resource constraints.

Table 1: Performance on 10 visual understanding benchmarks. “Res.” is resolution, and ‘#Vision Tokens’ is the number of vision tokens. Both training and inference employ DART as the token compression strategy for our methods. Parentheses in Avg.(%) column show diffs vs. LLaVA-v1.5.

Methods	LLM	Res.	#Vision Tokens	VQA^{V2}	GQA	VizWiz	SQA^{I}	VQA^{T}	POPE	MME	MMB	MMB-CN	OCR Bench	Avg. (%)
BLIP-2 [32]	Vicuna-13B	224	32	65.0	41.0	19.6	61.0	42.5	85.3	–	–	–	–	–
InstructBLIP [16]	Vicuna-7B	224	32	66.3	49.2	34.5	60.5	50.1	83.9	1500	36.0	–	259	–
InstructBLIP [16]	Vicuna-13B	224	32	64.2	49.5	33.4	63.1	–	84.1	1530	36.9	17.4	252	–
IDEFICS-9B [30]	LLaMA-7B	224	64	50.9	38.4	35.5	–	25.9	75.3	1027	48.2	–	245	–
IDEFICS-80B [30]	LLaMA-65B	224	64	60.0	45.2	36.0	–	30.9	–	1076	54.5	29.1	277	–
Qwen-VL [3]	Qwen-7B	448	256	–	59.3	35.2	67.1	63.8	–	1708	38.2	–	133	–
Qwen-VL-Chat [3]	Qwen-7B	448	256	–	57.5	38.9	68.2	61.5	–	1891	60.6	–	267	–
SPHINX [38]	LLaMA-13B	224	289	78.1	62.6	39.9	69.3	51.6	80.7	–	66.9	–	–	–
SPHINX-2k [38]	LLaMA-13B	762	2890	80.7	63.1	44.9	70.6	61.2	87.2	–	65.9	–	–	–
mPLUG-Owl2 [68]	LLaMA-7B	448	1024	79.4	56.1	54.5	68.7	54.3	-	–	64.5	–	–	–
Video-LLaVA [37]	Vicuna-7B	224	256	65.9	60.3	48.1	66.4	51.8	83.1	1542	60.6	49.3	161	55.7
LLaVA-v1.5 [41]	Vicuna-7B	336	576	72.2	61.9	52.5	68.3	58.1	85.9	1785	64.1	55.8	319	61.4
LMMs with fewer vision tokens														
Average-Pooling	Vicuna-7B	336	64	63.0	55.5	48.4	68.6	52.6	79.2	1579	59.6	49.5	258	55.9 (-5.5)
MQT-LLaVA [26]	Vicuna-7B	336	2	51.4	49.6	50.0	66.1	14.8	75.4	1402	48.9	40.5	169	46.4 (-15)
MQT-LLaVA [26]	Vicuna-7B	336	36	62.0	57.7	53.6	69.2	28.6	82.9	1777	60.5	51.6	244	55.4 (-6.0)
MQT-LLaVA [26]	Vicuna-7B	336	64	65.6	58.7	54.3	68.4	32.5	83.1	1810	61.3	53.7	260	56.8 (-4.6)
MQT-LLaVA [26]	Vicuna-7B	336	128	66.2	59.8	54.6	69.3	35.7	84.3	1773	62.1	53.6	266	57.6 (-3.8)
MQT-LLaVA [26]	Vicuna-7B	336	192	66.9	59.9	54.6	69.1	35.8	85.1	1784	62.0	53.9	263	57.7 (-3.7)
MQT-LLaVA [26]	Vicuna-7B	336	256	68.3	60.1	54.6	69.0	37.1	84.6	1740	61.7	53.0	273	57.8 (-3.6)
QT-LLaVA	Vicuna-7B	336	256	–	60.3	51.5	68.1	36.9	84.1	1771	62.1	53.9	265	–
LLaMA-VID [34]	Vicuna-7B	336	2	–	55.5	54.2	68.8	49.0	83.1	–	–	–	–	–
VoCo-LLaMA [70]	Vicuna-7B	336	1	–	55.6	54.6	68.4	31.7	80.8	1594	56.4	46.2	69	–
TokenPacker [33]	Vicuna-7B	336	144	71.3	62.0	56.6	70.5	43.8	86.2	1716	63.9	53.4	303	59.9 (-1.5)
TokenPacker [33]	Vicuna-7B	336	36	–	58.6	50.2	–	–	83.7	–	62.8	–	–	–
LLaVA-Mini [76]	Vicuna-7B	336	144	58.1	56.3	14.8	25.3	26.0	82.3	1325	24.8	–	132	–
LLaVA-Mini [76]	Vicuna-7B	336	64	-	56.6	10.4	27.4	28.1	82.3	1324	23.8	–	145	–
Ours														
LLaVA-v1.5 + TCD	Vicuna-7B	336	256	72.7	61.4	54.1	69.8	57.0	85.8	1807	66.1	54.8	310	61.7 (+0.3)
LLaVA-v1.5 + TCD	Vicuna-7B	336	192	71.6	60.9	54.0	70.0	56.9	85.3	1813	65.8	54.6	304	61.4 (+0.0)
LLaVA-v1.5 + TCD	Vicuna-7B	336	128	69.7	59.9	54.9	70.8	56.6	84.5	1861	65.6	54.3	299	61.3 (-0.1)
LLaVA-v1.5 + TCD	Vicuna-7B	336	64	66.1	57.1	55.1	71.1	54.8	79.2	1809	64.2	53.0	286	59.4 (-2.0)
LLaVA-v1.5 + TCD	Vicuna-7B	336	36	62.1	54.9	55.2	71.3	53.6	75.8	1747	62.4	51.5	262	57.5 (-3.9)
LLaVA-v1.5 + LCD	Vicuna-7B	336	256	72.6	62.0	57.4	69.8	56.8	86.1	1834	64.3	56.0	312	62.2 (+0.8)
LLaVA-v1.5 + LCD	Vicuna-7B	336	192	71.3	61.5	57.6	70.0	56.7	85.3	1830	64.4	55.9	316	62.0 (+0.6)
LLaVA-v1.5 + LCD	Vicuna-7B	336	128	69.2	60.6	57.9	69.8	56.3	84.2	1832	64.1	55.3	306	61.3 (-0.1)
LLaVA-v1.5 + LCD	Vicuna-7B	336	64	66.0	58.3	57.8	69.7	54.3	81.2	1794	62.1	52.9	280	59.4 (-2.0)
LLaVA-v1.5 + LCD	Vicuna-7B	336	36	62.8	56.5	56.5	70.3	52.9	77.7	1711	60.7	51.0	265	57.6 (-3.8)

When retaining 128 tokens, our framework achieves performance comparable to vanilla LLaVA-v1.5-7B, while surpassing it with 192+ visual tokens, strongly suggesting significant redundancy in visual tokens. Compared to other training-aware methods involving model modifications (*e.g.*, MQT-LLaVA, TokenPacker), our approach maintains superior average performance, particularly excelling on MME, MMBench, and VQA V2. This indicates that effective training strategies are as crucial as architectural modifications for token compression. The results also demonstrate our method's robustness across compression ratios: with just 64 tokens, performance degrades by merely 2% versus vanilla LLaVA, while maintaining minimal ($< 1\%$) variation at $128 \sim 256$ visual tokens.

4.3 Efficiency

Table 2: Inference efficiency analysis of `EPIC`. Δ denotes the reduction ratio. All experiments are on POPE ($8,910$ samples) using an A100 GPU. Token compression is fixed at the 2nd layer.

Method	**Visual Tokens**	**KV cache (MB)** ↓		**CUDA Time (s)** ↓		**FLOPs (T)** ↓	
		Value	**Δ**	**Value**	**Δ**	**Value**	**Δ**
LLaVA-v1.5-7B	576	367.2	–	1103.5	–	9.3	–
`EPIC` + FastV [7]	64	40.9	88.9%	749.1	32.1%	1.5	83.9%
`EPIC` + DART [64]	64	40.9	88.9%	744.3	32.6%	1.5	83.9%
`EPIC` + Random	64	40.9	88.9%	697.3	36.8%	1.5	83.9%

We discuss the efficiency of `EPIC`, including **training-time efficiency** and **inference-time efficiency**. As shown in Table 6 in Appendix D.2, in contrast to other training-aware token compression method, especially those that alter the model architecture, our approach only requires supervised fine-tuning, completing training in approximately 12 hours on 8 A100 GPUs. Most token compression methods that modify the model architecture require two or even three training stages. Furthermore, the replacement or addition of new model components necessitates more training iterations to properly

Table 3: Ablation study on Token Consistency Distillation. “w/o Distillation Loss” disables teacher supervision by zeroing $\mathcal{L}_{\text{TCD}}$. “w/o Progressive Compression Ratio” uses a fixed 88.9% compression ratio. DART is employed as the token compression strategy during training.

Method	VQA$^{\text{V2}}$	GQA	VizWiz	SQA$^{\text{I}}$	VQA$^{\text{T}}$	POPE	MME	MMB	MMB-CN	OCRBench	Avg. (%)
Retain 128 Tokens (↓ 77.8%)											
TCD (Ours)	69.7	59.9	54.9	70.8	56.6	84.5	1861	65.6	54.3	299	61.3
w/o Distillation Loss	67.2	60.4	53.2	68.6	57.3	83.2	1745	63.8	53.5	289	59.8 (-1.5)
w/o Progressive Compression Ratio	67.1	58.9	49.7	70.0	54.1	84.3	1788	63.8	51.5	277	59.1 (-2.2)
Retain 64 Tokens (↓ 88.9%)											
TCD (Ours)	66.1	57.1	55.1	71.1	54.8	79.2	1809	64.2	53.0	286	59.4
w/o Distillation Loss	65.7	57.9	53.6	69.8	54.9	78.3	1671	61.6	52.1	272	58.1 (-1.3)
w/o Progressive Compression Ratio	64.3	57.9	50.3	70.3	52.0	82.7	1774	62.8	52.5	255	58.2 (-1.2)

adapt these parameters, resulting in significantly greater computational expenditure (*e.g.*, $30 \sim 48$ hours on 8 A100 GPUs). This substantially increases the overall training cost. Focus on the inference-time efficiency, which in our framework is primarily influenced by the token compression strategies employed during the inference process. Table 2 presents the KV cache memory usage, CUDA time, and FLOPs obtained when applying three different token compression methods during inference for models trained using Token Consistency Distillation. The FLOPs are calculated using `calflops` [69], while the KV cache memory usage is estimated with the help of `LLM-Viewer` [72]. It can be observed that when retaining 64 tokens, all methods achieve improvements in reducing KV cache memory, FLOPs, and latency. In particular, Random token compression, which incurs no additional computational overhead, achieves an actual speedup of nearly $1.6\times$.

5 Analyses

In this section, we conduct a thorough investigation aimed at addressing the following key questions: **(1)** How much does the weight-sharing teacher guidance contribute to performance? **(2)** What occurs without progressive learning in token/layer-wise dimensions? **(3)** How generalizable is `EPIC` across various token compression strategies? **(4)** Is extreme token compression (1 or 2 tokens) necessary?

5.1 Ablation Studies

Table 4: Ablation study on Layer Consistency Distillation. “w/o Progressive Compression Layer”: fixed at the 2nd layer. DART is employed as the token compression strategy during training.

Method	VQA$^{\text{V2}}$	GQA	VizWiz	SQA$^{\text{I}}$	VQA$^{\text{T}}$	POPE	MME	MMB	MMB-CN	OCRBench	Avg. (%)
Retain 128 Tokens (↓ 77.8%)											
LCD (Ours)	69.2	60.6	57.9	69.8	56.3	84.2	1832	64.1	55.3	306	61.3
w/o Distillation Loss	67.1	60.6	55.4	70.3	56.1	84.3	1761	62.9	55.4	301	60.5 (-0.8)
w/o Progressive Compression Layer	68.7	59.3	54.3	70.6	56.2	82.2	1776	63.1	54.9	298	60.3 (-1.0)
Retain 64 Tokens (↓ 88.9%)											
LCD (Ours)	66.0	58.3	57.8	69.7	54.3	81.2	1794	62.1	52.9	280	59.4
w/o Distillation Loss	64.4	58.2	55.9	69.6	54.8	80.9	1735	61.3	52.7	275	58.7 (-0.7)
w/o Progressive Compression Layer	63.5	56.5	54.7	71.5	54.2	75.6	1734	61.9	51.7	260	57.8 (-1.6)

To validate the guiding role of the weight-sharing teacher model (RQ1) and the importance of the progressive learning strategy in training-aware token compression (RQ2), we conducted ablation studies on both components. Specifically, for token consistency distillation (TCD) and layer consistency distillation (LCD), in addition to eliminating the distillation loss to remove the teacher model’s influence, we also fix the compression ratio (*e.g.*, 88.9%) and compression layer (*e.g.*, the 2nd layer) to eliminate the progressive learning strategy in both the token-wise and layer-wise dimensions. This essentially degenerates the process into imposing a significant perturbation in the feature space, and the model is trained to adapt in the parameter space, like direct training. As demonstrated in Tables 3 and 4, the experimental results indicate that without teacher guidance, both TCD and LCD exhibit significant performance degradation across multiple benchmarks on average, with particularly pronounced declines on vision-centric benchmarks such as MME and MMBench.

5.2 How Well Does Proposed Framework Generalize across Different Methods?

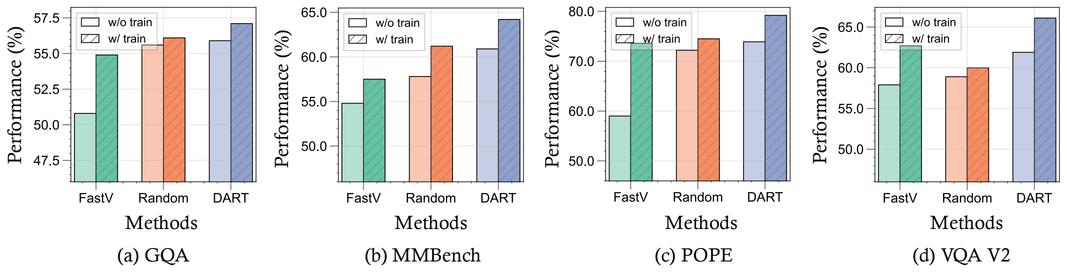


Figure 4: Following LLaVA-v1.5's architecture and data, we apply **DART** for token consistency distillation. "w/o train" denotes vanilla LLaVA. At inference, all methods use 88.9% token compression.

Beyond the generalization across different compression ratios observed in our comparative experiments of Sec. 4.2, we further investigate the adaptability of our proposed framework to diverse token compression strategies (RQ3). Specifically, as outlined in Sec. 4.1, we integrate three plug-and-play token compression strategies into our framework during training. We then conduct cross-strategy evaluations to assess how models trained with one specific strategy (*e.g.*, FastV, DART) perform when applied with alternative compression methods during inference. As shown in Figure 4, token consistency distillation consistently improves model performance across all benchmarks and compression methods. Notably, even when trained solely with DART-based compression, the model generalizes well to FastV and Random compression, yielding consistent performance gains. Furthermore, after training with our proposed framework, the performance gap between token compression strategies is significantly reduced. Notably, previously underperforming strategies exhibit more substantial improvements than their stronger counterparts. For additional experiments and discussions on the generalization capability of our method, please refer to Appendix A.

5.3 Is Extreme Token Compression Necessary?

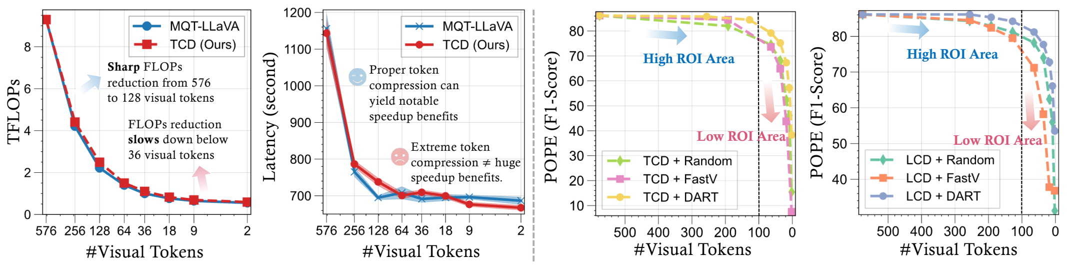


Figure 5: All experiments use the model trained following LLaVA-v1.5. FLOPs and latency are measured on the POPE. Visual token and latency experiments are repeated three times for reliability.

We observe that many token compression methods pursue aggressive compression (*e.g.*, one or two tokens). Table 1 shows that extreme token compression still leads to notable performance degradation for many methods. While this consistently reduces KV cache memory, it raises the question: *does such extreme compression always translate to faster inference* (RQ4)? To answer this, we conducted detailed analysis and experiments. Figure 5 shows the relationship between FLOPs and the number of retained visual tokens. When reducing tokens from the full set (576 tokens) to 128, FLOPs drop significantly—from 9.3T to around 2T. However, under more extreme compression, FLOPs reduction becomes noticeably smaller. A similar or even more pronounced trend is observed between token count and actual latency. In some cases, retaining 64 tokens yields better performance than fewer tokens (*e.g.*, 36 or 18). A possible hypothesis is that overly fragmented feature slices increase memory access time. Moreover, reducing tokens to 64 largely preserves vanilla model performance. We refer to this range as the High Return-on-Investment **(High ROI)** Area. Further reduction beyond 64 offers only marginal latency gains but sharply degrades performance—this is the **Low ROI** Area. Model efficiency depends on whether it is computation or memory-bound. With heavily reduced tokens, GPU compute is underutilized and latency is dominated by memory access, making the system memory-bound, where further reduction brings little speedup. Overall, we argue that extreme compression is unnecessary; instead, focus should be on balancing latency and performance.

6 Conclusion

In this paper, we propose EPIC, a learning framework that enhances the efficiency of multi-modal large language models (MLLMs) via Progressive Consistency Distillation. It integrates with existing token compression strategies without modifying the model architecture, achieving efficiency in both training and inference. Experimental results demonstrate that MLLMs trained with our framework achieve comparable average performance to the vanilla model using only 128 visual tokens. Notably, on 4 out of 10 visual understanding benchmarks, our approach even outperforms the vanilla model despite using significantly fewer visual tokens. Furthermore, extensive experiments validate the effectiveness of EPIC, highlighting its robustness across token compression ratios and generalization across strategies. Meanwhile, our analysis reveals that while token compression offers significant benefits, excessive compression may lead to a poor latency-performance trade-off, underscoring the importance of balancing compression levels for optimal performance.

References

[1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

[2] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. Qwen technical report. *arXiv preprint arXiv:2309.16609*, 2023.

[3] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-VL: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*, 2023.

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

[5] Junbum Cha, Wooyoung Kang, Jonghwan Mun, and Byungseok Roh. Honeybee: Locality-enhanced projector for multimodal llm. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13817–13827, 2024.

[6] Junjie Chen, Xuyang Liu, Zichen Wen, Yiyu Wang, Siteng Huang, and Honggang Chen. Variation-aware vision token dropping for faster large vision-language models. *arXiv preprint arXiv:2509.01552*, 2025.

[7] Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, pages 19–35. Springer, 2024.

[8] Zhaorun Chen, Yichao Du, Zichen Wen, Yiyang Zhou, Chenhang Cui, Zhenzhen Weng, Haoqin Tu, Chaoqi Wang, Zhengwei Tong, Qinglan Huang, et al. Mj-bench: Is your multimodal reward model really a good judge for text-to-image generation? *arXiv preprint arXiv:2407.04842*, 2024.

[9] Zhaorun Chen, Francesco Pinto, Minzhou Pan, and Bo Li. Safewatch: An efficient safety-policy following video guardrail model with transparent explanations. In *The Thirteenth International Conference on Learning Representations*.

[10] Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

[11] Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *Science China Information Sciences*, 67(12):220101, 2024.

[12] 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 opensource chatbot impressing gpt-4 with 90% chatgpt quality. *ArXiv preprint*, 2023.

[13] Xiangxiang Chu, Limeng Qiao, Xinyang Lin, Shuang Xu, Yang Yang, Yiming Hu, Fei Wei, Xinyu Zhang, Bo Zhang, Xiaolin Wei, et al. Mobilevlm: A fast, reproducible and strong vision language assistant for mobile devices. *arXiv preprint arXiv:2312.16886*, 1(2):3, 2023.

[14] Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, et al. Mobilevlm v2: Faster and stronger baseline for vision language model. *arXiv preprint arXiv:2402.03766*, 2024.

[15] 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.

[16] 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.

[17] Tri Dao. Flashattention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*, 2023.

[18] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022.

[19] Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, et al. Internlm-xcomposer2-4khd: A pioneering large vision-language model handling resolutions from 336 pixels to 4k hd. *Advances in Neural Information Processing Systems*, 37:42566–42592, 2024.

[20] Tayebeh Fani and Farid Ghaemi. Implications of vygotsky's zone of proximal development (zpd) in teacher education: Zptd and self-scaffolding. *Procedia-Social and Behavioral Sciences*, 29:1549–1554, 2011.

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

[22] 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.

[23] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[24] Yefei He, Feng Chen, Jing Liu, Wenqi Shao, Hong Zhou, Kaipeng Zhang, and Bohan Zhuang. Zipvl: Efficient large vision-language models with dynamic token sparsification and kv cache compression. *arXiv preprint arXiv:2410.08584*, 2024.

[25] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[26] Wenbo Hu, Zi-Yi Dou, Liunian Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. Matryoshka query transformer for large vision-language models. *Advances in Neural Information Processing Systems*, 37:50168–50188, 2024.

[27] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, 2019.

[28] Yutao Jiang, Qiong Wu, Wenhao Lin, Wei Yu, and Yiyi Zhou. What kind of visual tokens do we need? training-free visual token pruning for multi-modal large language models from the perspective of graph. *arXiv preprint arXiv:2501.02268*, 2025.

[29] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[30] Hugo Laurençon, Lucile Saulnier, Léo Tronchon, Stas Bekman, Amanpreet Singh, Anton Lozhkov, Thomas Wang, Siddharth Karamcheti, Alexander Rush, Douwe Kiela, et al. Obelics: An open web-scale filtered dataset of interleaved image-text documents. *Advances in Neural Information Processing Systems*, 36:71683–71702, 2023.

[31] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024.

[32] 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 *International conference on machine learning*, pages 19730–19742. PMLR, 2023.

[33] Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm. *arXiv preprint arXiv:2407.02392*, 2024.

[34] Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pages 323–340. Springer, 2024.

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

[36] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proc. of EMNLP*, 2023.

[37] Bin Lin, Bin Zhu, Yang Ye, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. *arXiv preprint arXiv:2311.10122*, 2023.

[38] Ziyi Lin, Chris Liu, Renrui Zhang, Peng Gao, Longtian Qiu, Han Xiao, Han Qiu, Chen Lin, Wenqi Shao, Keqin Chen, et al. Sphinx: The joint mixing of weights, tasks, and visual embeddings for multi-modal large language models. *arXiv preprint arXiv:2311.07575*, 2023.

[39] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26296–26306, 2024.

[40] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024.

[41] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc., 2023.

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

[43] Xuyang Liu, Ziming Wang, Yuhang Han, Yingyao Wang, Jiale Yuan, Jun Song, Bo Zheng, Linfeng Zhang, Siteng Huang, and Honggang Chen. Compression with global guidance: Towards training-free high-resolution mllms acceleration. *arXiv preprint arXiv:2501.05179*, 2025.

[44] Xuyang Liu, Zichen Wen, Shaobo Wang, Junjie Chen, Zhishan Tao, Yubo Wang, Xiangqi Jin, Chang Zou, Yiyu Wang, Chenfei Liao, et al. Shifting ai efficiency from model-centric to data-centric compression. *arXiv preprint arXiv:2505.19147*, 2025.

[45] Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. Mmbench: Is your multi-modal model an all-around player? *arXiv preprint arXiv:2307.06281*, 2023.

[46] Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024.

[47] Pan Lu, Swaroop Mishra, Tanglin 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. *NeurIPS*, 2022.

[48] David Marr. *Vision: A computational investigation into the human representation and processing of visual information*. MIT press, 2010.

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

[50] Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*, 2024.

[51] Amanpreet Singh, Vivek Natarjan, Meet Shah, Yu Jiang, Xinlei Chen, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019.

[52] Xudong Tan, Peng Ye, Chongjun Tu, Jianjian Cao, Yaoxin Yang, Lin Zhang, Dongzhan Zhou, and Tao Chen. Tokencarve: Information-preserving visual token compression in multimodal large language models. *arXiv preprint arXiv:2503.10501*, 2025.

[53] Yunlong Tang, Jing Bi, Siting Xu, Luchuan Song, Susan Liang, Teng Wang, Daoan Zhang, Jie An, Jingyang Lin, Rongyi Zhu, et al. Video understanding with large language models: A survey. *arXiv preprint arXiv:2312.17432*, 2023.

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

[55] Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, et al. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *arXiv preprint arXiv:2502.14786*, 2025.

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

[57] Haicheng Wang, Zhemeng Yu, Gabriele Spadaro, Chen Ju, Victor Quétu, and Enzo Tartaglione. Folder: Accelerating multi-modal large language models with enhanced performance. *arXiv preprint arXiv:2501.02430*, 2025.

[58] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

[59] Shaobo Wang, Xiangqi Jin, Ziming Wang, Jize Wang, Jiajun Zhang, Kaixin Li, Zichen Wen, Zhong Li, Conghui He, Xuming Hu, et al. Data whisperer: Efficient data selection for task-specific llm fine-tuning via few-shot in-context learning. *arXiv preprint arXiv:2505.12212*, 2025.

[60] Shaobo Wang, Jiaming Wang, Jiajun Zhang, Cong Wang, Yue Min, Zichen Wen, Fei Huang, Huiqiang Jiang, Junyang Lin, Dayiheng Liu, and Linfeng Zhang. Winning the pruning gamble: A unified approach to joint sample and token pruning for efficient supervised fine-tuning. *arXiv preprint arXiv:2509.23873*, 2025.

[61] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Jilan Xu, Zun Wang, et al. Internvideo2: Scaling video foundation models for multimodal video understanding. *Arxiv e-prints*, pages arXiv–2403, 2024.

[62] Yiqi Wang, Wentao Chen, Xiaotian Han, Xudong Lin, Haiteng Zhao, Yongfei Liu, Bohan Zhai, Jianbo Yuan, Quanzeng You, and Hongxia Yang. Exploring the reasoning abilities of multimodal large language models (mllms): A comprehensive survey on emerging trends in multimodal reasoning. *arXiv preprint arXiv:2401.06805*, 2024.

[63] Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. Token pruning in multimodal large language models: Are we solving the right problem? *arXiv preprint arXiv:2502.11501*, 2025.

[64] Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for important tokens in multimodal language models: Duplication matters more. *arXiv preprint arXiv:2502.11494*, 2025.

[65] Minhao Xiong, Zichen Wen, Zhuangcheng Gu, Xuyang Liu, Rui Zhang, Hengrui Kang, Jiabing Yang, Junyuan Zhang, Weijia Li, Conghui He, et al. Prune2drive: A plug-and-play framework for accelerating vision-language models in autonomous driving. *arXiv preprint arXiv:2508.13305*, 2025.

[66] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

[67] Yantai Yang, Yuhao Wang, Zichen Wen, Luo Zhongwei, Chang Zou, Zhipeng Zhang, Chuan Wen, and Linfeng Zhang. Efficientvla: Training-free acceleration and compression for vision-language-action models. *arXiv preprint arXiv:2506.10100*, 2025.

[68] Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. In *Proceedings of the ieee/cvf conference on computer vision and pattern recognition*, pages 13040–13051, 2024.

[69] Xiaoju Ye. calflops: a flops and params calculate tool for neural networks in pytorch framework, 2023.

[70] Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, and Yansong Tang. Voco-llama: Towards vision compression with large language models. *arXiv preprint arXiv:2406.12275*, 2024.

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

[72] Zhihang Yuan, Yuzhang Shang, Yang Zhou, Zhen Dong, Chenhao Xue, Bingzhe Wu, Zhikai Li, Qingyi Gu, Yong Jae Lee, Yan Yan, Beidi Chen, Guangyu Sun, and Kurt Keutzer. Llm inference unveiled: Survey and roofline model insights, 2024.

[73] Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11975–11986, 2023.

[74] Jiarui Zhang, Jinyi Hu, Mahyar Khayatkhoei, Filip Ilievski, and Maosong Sun. Exploring perceptual limitation of multimodal large language models. *arXiv preprint arXiv:2402.07384*, 2024.

[75] Jiarui Zhang, Mahyar Khayatkhoei, Prateek Chhikara, and Filip Ilievski. Visual cropping improves zero-shot question answering of multimodal large language models. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023.

[76] Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token. *arXiv preprint arXiv:2501.03895*, 2025.

[77] Yiming Zhang, Zhuokai Zhao, Zhaorun Chen, Zenghui Ding, Xianjun Yang, and Yining Sun. Beyond training: Dynamic token merging for zero-shot video understanding. *arXiv preprint arXiv:2411.14401*, 2024.

[78] Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*, 2024.

[79] Zeliang Zhang, Phu Pham, Wentian Zhao, Kun Wan, Yu-Jhe Li, Jianing Zhou, Daniel Miranda, Ajinkya Kale, and Chenliang Xu. Treat visual tokens as text? but your mllm only needs fewer efforts to see. *arXiv preprint arXiv:2410.06169*, 2024.

[80] Shiyu Zhao, Zhenting Wang, Felix Juefei-Xu, Xide Xia, Miao Liu, Xiaofang Wang, Mingfu Liang, Ning Zhang, Dimitris N Metaxas, and Licheng Yu. Accelerating multimodel large language models by searching optimal vision token reduction. *arXiv preprint arXiv:2412.00556*, 2024.

[81] 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.

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: The main contributions and scope of this paper are outlined in the abstract and introduction sections.

   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: In Appendix E, we outline the potential limitations of this work.

   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: [Yes]

   Justification: In the main paper and Appendix, we provide the full set of assumptions and complete proof.

   Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: We provide details of all model versions, dataset versions, hyperparameters, and experimental settings in Sec. 4.1, and Appendix D.

   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: [No]

   Justification: In accordance with the requirements of the supporting organization, the code will be released after the review process is completed. We are committed to providing sufficient instructions to ensure reproducibility at that time.

   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: All the training and evaluation details have been provided in Sec. 4.1 and Appendix D.

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: Some experiments, including latency statistics, were conducted multiple times, and the corresponding error bars or bands are reported in Sec. 5.

Guidelines:

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

8. **Experiments compute resources**

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

Answer: [Yes]

Justification: We include descriptions of the computational resources used in Sec. 4.3 and Appedix D.

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: We followed 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: We describe broader impacts of our work in Appendix F.

    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: [NA]

    Justification: The paper poses no such risks.

    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: The creators or original owners of assets (e.g., code, data, models) used in the paper have been properly credited, and the corresponding papers have been cited.

    Guidelines:

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

13. **New assets**

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

    Answer: [NA]

    Justification: 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.

16. **Declaration of LLM usage**

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

    Answer: [NA]

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

    Guidelines:

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

Appendix

A More Generalizability Experiments

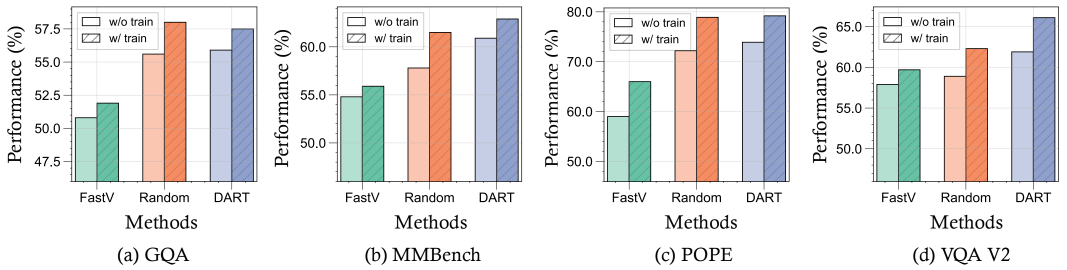


Figure 6: Following model architecture and training data of LLaVA-v1.5-7B, we apply the **Random token compression** for token consistency distillation during training. Here, “w/o train” denotes the vanilla LLaVA. During inference, all methods achieve a token compression ratio of 88.9%.

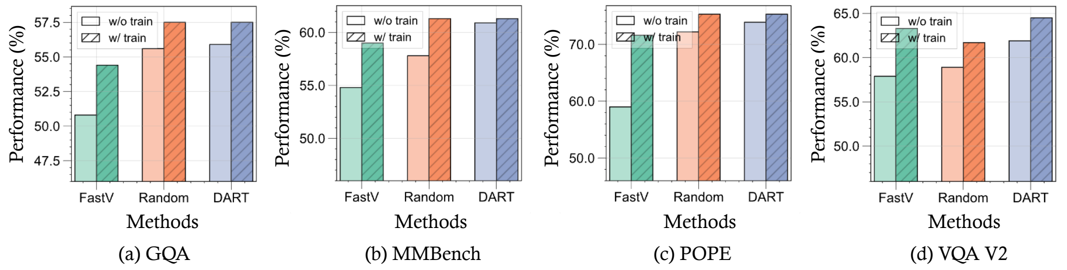


Figure 7: Following model architecture and training data of LLaVA-v1.5-7B, we apply the **FastV** for token consistency distillation during training. Here, “w/o train” denotes the vanilla LLaVA. During inference, all methods achieve a token compression ratio of 88.9%.

In addition to the cross-method generalization validation of our proposed framework conducted in Sec. 5.2, we performed more comprehensive and meticulous experiments to further verify its effectiveness. Specifically, as illustrated in Figures 6 and 7, we employ Random Token Compression and FastV, respectively, to perform token consistency distillation on the multi-modal large language model. Experimental results demonstrate that regardless of the token compression strategy used during training with our proposed method, the trained model consistently achieves improved inference performance across various token compression strategies. This strongly demonstrates the generalizability and effectiveness of our approach, indicating that models trained under our framework genuinely adapt to the pattern of missing visual tokens without being constrained to a fixed set of preserved tokens.

B Detailed Theoretical Analysis

Proof of Theorem 1. **1. Direct path.** Since $\theta_{r_t}^{\mathrm{dir}} = c(r_t)$, the total variation is

$$\mathrm{TV}(\{\theta_{r_t}^{\mathrm{dir}}\}) = \sum_{t=0}^{T-1} (c(r_{t+1}) - c(r_t)).$$

By monotonicity **(S1)** and Lipschitz continuity **(S2)**, each term satisfies

$$c(r_{t+1}) - c(r_t) \le \gamma(r_{t+1} - r_t),$$

so summing yields:

$$\mathrm{TV}(\{\theta_{r_t}^{\mathrm{dir}}\}) \le \gamma \sum_{t=0}^{T-1} (r_{t+1} - r_t) = \gamma r_{\max}.$$

2. Progressive path. From the closed-form minimizer,

$$\theta_{r_t}^{\mathrm{prog}} = \frac{c(r_t) + \lambda\, c(r_t - \Delta)}{1+\lambda},$$

we have:

$$\theta_{r_{t+1}}^{\mathrm{prog}} - \theta_{r_t}^{\mathrm{prog}} = \frac{(c(r_{t+1}) - c(r_t)) + \lambda(c(r_{t+1} - \Delta) - c(r_t - \Delta))}{1+\lambda}.$$

Define $\Delta_t := c(r_{t+1}) - c(r_t) \ge 0$ and $\tilde{\Delta}_t := c(r_{t+1} - \Delta) - c(r_t - \Delta) \ge 0$. Then:

$$\left|\theta_{r_{t+1}}^{\mathrm{prog}} - \theta_{r_t}^{\mathrm{prog}}\right| \le \frac{\Delta_t + \lambda \tilde{\Delta}_t}{1+\lambda} = \frac{1+\lambda\kappa_t}{1+\lambda} \cdot \Delta_t, \quad \text{where } \kappa_t := \frac{\tilde{\Delta}_t}{\Delta_t}.$$

3. Bounding κ_t. Define the worst-case slope ratio:

$$\kappa := \sup_{r \in [\Delta,\, r_{\max} - \Delta]} \frac{c(r) - c(r-\Delta)}{c(r+\Delta) - c(r)}.$$

Each κ_t compares the lagged slope to the current slope. If the schedule is appropriately spaced so that $r_t \in [\Delta, r_{\max} - \Delta]$, then we have:

$$\kappa_t = \frac{c(r_{t+1} - \Delta) - c(r_t - \Delta)}{c(r_{t+1}) - c(r_t)} \le \kappa.$$

Convexity **(S3)** ensures that $c'(r)$ is non-decreasing, so

$$c(r) - c(r-\Delta) \le c(r+\Delta) - c(r) \quad \Rightarrow \quad \kappa \le 1.$$

If c is not affine (i.e., $c''(r) > 0$ on a set of positive measure), the inequality is strict for some r, and thus $\kappa < 1$.

4. Total variation bound. Summing over steps gives:

$$\mathrm{TV}(\{\theta_{r_t}^{\mathrm{prog}}\}) = \sum_{t=0}^{T-1} \left|\theta_{r_{t+1}}^{\mathrm{prog}} - \theta_{r_t}^{\mathrm{prog}}\right| \le \sum_{t=0}^{T-1} \frac{1+\lambda\kappa}{1+\lambda} \cdot \Delta_t = \frac{1+\lambda\kappa}{1+\lambda} \cdot \mathrm{TV}(\{\theta_{r_t}^{\mathrm{dir}}\}),$$

with strict inequality when $\kappa < 1$, completing the proof. $\square$

Table 5: Performance on 10 visual understanding benchmarks. 'Res.' is resolution, and '#Vision Tokens' is the number of vision tokens fed to the LLM backbone. Both training and inference of **Integrated Progressive Consistency Distillation (ICD)** employ DART as the token compression strategy for our methods. Parentheses in Avg.(%) column show diffs vs. LLaVA-v1.5.

Methods	**LLM**	**Res.**	**#Vision Tokens**	**VQA^{V2}**	**GQA**	**VizWiz**	**SQA^{I}**	**VQA^{T}**	**POPE**	**MME**	**MMB**	**MMB-CN**	**OCR Bench**	**Avg. (%)**
BLIP-2 [32]	Vicuna-13B	224	32	65.0	41.0	19.6	61.0	42.5	85.3	–	–	–	–	–
InstructBLIP [16]	Vicuna-7B	224	32	66.3	49.2	34.5	60.5	50.1	83.9	1500	36.0	–	259	–
InstructBLIP [16]	Vicuna-13B	224	32	64.2	49.5	33.4	63.1	–	84.1	1530	36.9	17.4	252	–
IDEFICS-9B [30]	LLaMA-7B	224	64	50.9	38.4	35.5	–	25.9	75.3	1027	48.2	–	245	–
IDEFICS-80B [30]	LLaMA-65B	224	64	60.0	45.2	36.0	–	30.9	–	1076	54.5	29.1	277	–
Qwen-VL [3]	Qwen-7B	448	256	–	59.3	35.2	67.1	63.8	–	1708	38.2	–	133	–
Qwen-VL-Chat [3]	Qwen-7B	448	256	–	57.5	38.9	68.2	61.5	–	1891	60.6	–	267	–
SPHINX [38]	LLaMA-13B	224	289	78.1	62.6	39.9	69.3	51.6	80.7	–	66.9	–	–	–
SPHINX-2k [38]	LLaMA-13B	762	2890	80.7	63.1	44.9	70.6	61.2	87.2	–	65.9	–	–	–
mPLUG-Owl2 [68]	LLaMA-7B	448	1024	79.4	56.1	54.5	68.7	54.3	-	–	64.5	–	–	–
Video-LLaVA [37]	Vicuna-7B	224	256	65.9	60.3	48.1	66.4	51.8	83.1	1542	60.6	49.3	161	55.7
LLaVA-v1.5 [41]	Vicuna-7B	336	576	72.2	61.9	52.5	68.3	58.1	85.9	1785	64.1	55.8	319	61.4
LMMs with fewer vision tokens														
Average-Pooling	Vicuna-7B	336	64	63.0	55.5	48.4	68.6	52.6	79.2	1579	59.6	49.5	258	55.9 (-5.5)
MQT-LLaVA [26]	Vicuna-7B	336	2	51.4	49.6	50.0	66.1	14.8	75.4	1402	48.9	40.5	169	46.4 (-15)
MQT-LLaVA [26]	Vicuna-7B	336	36	62.0	57.7	53.6	69.2	28.6	82.9	1777	60.5	51.6	244	55.4 (-6.0)
MQT-LLaVA [26]	Vicuna-7B	336	64	65.6	58.7	54.3	68.4	32.5	83.1	1810	61.3	53.7	260	56.8 (-4.6)
MQT-LLaVA [26]	Vicuna-7B	336	128	66.2	59.8	54.6	69.3	35.7	84.3	1773	62.1	53.6	266	57.6 (-3.8)
MQT-LLaVA [26]	Vicuna-7B	336	192	66.9	59.9	54.6	69.1	35.8	85.1	1784	62.0	53.9	263	57.7 (-3.7)
MQT-LLaVA [26]	Vicuna-7B	336	256	68.3	60.1	54.6	69.0	37.1	84.6	1740	61.7	53.0	273	57.8 (-3.6)
QT-LLaVA	Vicuna-7B	336	256	–	60.3	51.5	68.1	36.9	84.1	1771	62.1	53.9	265	–
LLaMA-VID [34]	Vicuna-7B	336	2	–	55.5	54.2	68.8	49.0	83.1	–	–	–	–	–
VoCo-LLaMA [70]	Vicuna-7B	336	1	–	55.6	54.6	68.4	31.7	80.8	1594	56.4	46.2	69	–
TokenPacker [33]	Vicuna-7B	336	144	71.3	62.0	56.6	70.5	43.8	86.2	1716	63.9	53.4	303	59.9 (-1.5)
TokenPacker [33]	Vicuna-7B	336	36	–	58.6	50.2	–	–	83.7	–	62.8	–	–	–
LLaVA-Mini [76]	Vicuna-7B	336	144	58.1	56.3	14.8	25.3	26.0	82.3	1325	24.8	–	132	–
LLaVA-Mini [76]	Vicuna-7B	336	64	-	56.6	10.4	27.4	28.1	82.3	1324	23.8	–	145	–
Ours														
LLaVA-v1.5 + ICD	Vicuna-7B	336	256	72.2	61.7	54.7	70.0	58.1	85.7	1788	64.9	55.7	312	61.8 (+0.4)
LLaVA-v1.5 + ICD	Vicuna-7B	336	192	71.9	61.4	54.8	70.3	57.6	84.8	1783	64.8	55.9	316	61.7 (+0.3)
LLaVA-v1.5 + ICD	Vicuna-7B	336	128	69.6	60.3	55.2	70.1	57.1	83.5	1770	63.9	55.2	298	60.8 (-0.6)
LLaVA-v1.5 + ICD	Vicuna-7B	336	64	66.5	57.5	55.3	70.4	54.7	79.3	1720	63.8	53.2	275	59.0 (-2.4)
LLaVA-v1.5 + ICD	Vicuna-7B	336	36	62.1	55.6	54.4	70.9	52.8	74.0	1635	61.2	51.1	246	56.5 (-4.9)

C Integrated Progressive Consistency Distillation

As outlined in Sec. 1, our Progressive Consistency Distillation framework incorporates two progressive learning mechanisms: Token Consistency Distillation (TCD) in a token-wise manner and Layer Consistency Distillation (LCD) in a layer-wise manner. Experimental results demonstrate that both approaches achieve strong performance across all key metrics: model accuracy, efficiency, robustness, and generalization capability. To further investigate this direction, we explore the integration of progressive learning from both token-wise and layer-wise perspectives. A critical question arises: *Can this integrated approach continue to maintain or even enhance the model's performance?* Building upon these two approaches, we design Integrated Progressive Consistency Distillation (ICD). Unlike the original Token Consistency Distillation (TCD), where token-wise progression is governed by global training progress, our method enables layer-wise iterative application of TCD, effectively integrating both token-wise and layer-wise dimensions. Specifically, during training, the token compression layer progressively shifts from deeper to shallower layers. Within each layer, the compression ratio follows the TCD schedule, sampling from small to large values. Upon transitioning to the next layer, the ratio resets to its initial value, and the process repeats. As shown in Table 5, the Integrated Progressive Consistency Distillation approach achieves comparable performance on representative visual benchmarks. Notably, it even surpasses the vanilla LLaVA in average performance while retaining only 192 visual tokens (66.7% ↓). Furthermore, when compared to other training-aware token compression approaches (such as MQT-LLaVA and TokenPacker), ICD demonstrates superior performance under identical retained visual tokens.

D Experimental Setup

D.1 Token Compression Techniques During Training

We adopt three representative token compression methods in our proposed framework: FastV, which follows an importance-based strategy; DART, which leverages redundancy-based pruning; and random token pruning, the simplest form of token compression.

- DART [64] is a training-free and plug-and-play method that prunes visual tokens based on token duplication while maintaining compatibility with efficient attention mechanisms like Flash Attention [18, 17].
- FastV [7] is a plug-and-play token compression technique that builds on the observation that visual tokens tend to have diminishing contributions to model outputs in deeper layers. By utilizing attention scores to assess token importance, it prunes less critical visual tokens at earlier layers of the language model.
- Random is a token compression method that requires no additional signals or computation, and is primarily used to validate the effectiveness and generalizability of our proposed approach.

Table 6: Method Comparison Across Different Stages

Method	**Stage 1**	**Stage 2**	**Stage 3**	**Training Time ↓ (h)**
QT-LLaVA	✓	✓		$\sim$ 30h $\times$ 8 A100s
MQT-LLaVA	✓	✓		$\sim$ 34h $\times$ 8 A6000s
LLaMA-VID	✓	✓	✓	$\sim$ 48h $\times$ 8 A100s
LLaVA-Mini	✓	✓		$\sim$ 26h $\times$ 8 A100s
`EPIC` (Ours)		✓		$\sim$ 12.2h $\times$ 8 A100s

D.2 Training Details

For a fair comparison, our framework not only adheres to the same model architecture and instruction-tuning data as vanilla LLaVA but also maintains identical hyperparameter settings. Moreover, since we have not modified the model architecture, we can directly use the pre-trained projector—unlike MQT-LLaVA, QT-LLaVA, and LLaVA-Mini, which require mandatory Stage 1 pre-training. The training process, conducted on 8 $\times$ A100 GPUs, takes approximately 12 hours. Furthermore, we faithfully reproduced the entire LLaVA training process following the official LLaVA-v1.5 training guidelines. All other token compression baselines are either trained following the settings provided in the original papers or evaluated using publicly available model checkpoints. For the ablation study in Section 5.1, the experiment without the distillation loss was trained using the same LLaVA-665K SFT data, with all other training parameters and procedures kept identical to those of `EPIC`. For the variants without a progressive compression ratio (Table 3) or progressive compression layer (Table 4), we fixed the second layer as the compression layer. Overall, our ablation studies were conducted under a strictly fair and consistent experimental setup.

Table 7: Detailed hyperparameter settings.

Settings	**Stage 2**
Batch size	128
Learning rate	2e-5
Learning schedule	Cosine decay
Warmup ratio	0.03
Weight decay	0
Epoch	1
Optimizer	AdamW
DeepSpeed stage	3
Max token	2048

D.3 Benchmarks

- MME [21] is a comprehensive benchmark for evaluating the performance of MLLMs in multi-modal tasks. It measures models' capabilities across two key areas: perception and cognition, using 14 specially designed subtasks that test interpretative and analytical skills.
- MMBench [45] employs a dual approach: it provides an extensive dataset that broadens the range and variety of evaluation questions, and introduces the innovative CircularEval strategy, which uses ChatGPT to convert free-form predictions into structured choices. MMBench-CN is the Chinese version of the benchmark.
- ScienceQA [47] is a multi-modal benchmark aimed at assessing and diagnosing AI systems' multi-hop reasoning and interpretability in the science domain. It includes a dataset of around 21K multiple-choice questions across various scientific topics, complete with detailed answer annotations, related lectures, and explanations.
- GQA [27] is a dataset designed for advanced visual reasoning in real-world scenarios, using scene graph-based structures to generate 22 million diverse, semantically-programmed questions. It features a novel set of evaluation metrics focused on consistency, grounding, and plausibility, setting a high standard for vision-language task assessment.
- POPE [36] is an evaluation method for examining object hallucination in MLLMs. It transforms the evaluation into a binary classification task, asking MLLMs simple Yes-or-No questions to identify hallucinated objects. POPE employs various object sampling strategies to reveal model tendencies towards hallucination.
- VQA V2 [22] evaluates the model's visual perception capabilities through open-ended questions. It consists of 265,016 images, covering a wide variety of real-world scenes and objects, providing rich visual contexts for the questions. For each question, there are 10 ground truth answers provided by human annotators, which allows for a comprehensive evaluation of the performance of different models in answering the questions accurately.
- TextVQA [51] focuses on the comprehensive integration of diverse text information within images. It meticulously evaluates the model's text understanding and reasoning abilities through a series of visual question-answering tasks with rich textual information. Models need to not only understand the visual content of the images but also be able to read and reason about the text within the images to answer the questions accurately.
- OCRBench [46] is a comprehensive benchmark for evaluating the OCR capabilities of multi-modal language models across five key tasks: text recognition, scene text-centric and document-oriented VQA, key information extraction, and handwritten mathematical expression recognition.

D.4 Overview of the Baselines

D.4.1 General MLLMs

- BLIP-2 [32] is a vision-language pretraining framework that efficiently combines frozen image encoders with large language models (LLMs). It adopts a two-stage training strategy leveraging a lightweight Querying Transformer to bridge the vision-language modality gap, enabling compute-efficient, zero-shot image-to-text generation aligned with natural language instructions.
- InstructBLIP [15] builds on BLIP-2 by introducing instruction tuning and an instruction-aware Query Transformer. This enhances the model's ability to extract features for a wide range of vision-language tasks. It achieved state-of-the-art zero-shot performance across 13 benchmarks and demonstrated strong results on fine-tuned tasks such as ScienceQA.
- IDEFICS [30] is an open-access vision-language model based on the Flamingo [1] architecture. Available in both base and instruction-tuned variants (9B and 80B parameters), IDEFICS is trained entirely on publicly available data and models, promoting transparency and accessibility.
- Qwen-VL & Qwen-VL-Chat [3] extend the Qwen-LM [2] foundation with a visual encoder and a specialized input-output interface. Through a three-stage training pipeline and a rich multilingual, multimodal corpus, the models achieve strong capabilities in grounding and optical character recognition (OCR) tasks.
- SPHINX [38] is a multimodal large language model that applies joint mixing across model weights, tuning objectives, visual embeddings, and image scales. By unfreezing the LLM during pretraining

and integrating diverse instructional and visual signals, SPHINX demonstrates strong performance in fine-grained vision-language understanding and reasoning tasks.

- mPLUG-Owl2 [68] features a modular architecture with a language decoder interface for unified modality coordination. It employs shared cross-modal modules alongside modality-adaptive components to enhance feature retention and generalization in both unimodal and multimodal settings.
- Video-LLaVA [37] extends multimodal language modeling to unified video and image understanding. By aligning visual modalities into a shared language feature space prior to projection, the model enables effective joint training across visual domains, achieving state-of-the-art results on diverse video and image benchmarks without requiring paired video-image data.
- LLaVA-v1.5 [39] improves upon the original LLaVA [41] model through enhanced visual instruction tuning. Utilizing a CLIP-ViT-L-336px visual backbone and MLP-based projection, it achieves strong performance with high data efficiency. Trained on only 1.2 million publicly available images, it excels at academic-oriented VQA tasks using straightforward prompting strategies.

D.4.2 Training-aware Token Compression Methods

- Average Pooling, inspired by the token merging strategy in Qwen2-VL [58], merges every four adjacent visual patches into a single token, effectively reducing the number of visual tokens.
- QT-LLaVA replaces the original modality projector in LLaVA with a Q-former [32] that uses learnable queries to project the input token sequence into a shorter sequence. In this work, we fix to retain 256 visual tokens.
- MQT-LLaVA [26] proposes a flexible Query Transformer that enables encoding images into a variable number of visual tokens (up to a predefined maximum), allowing dynamic adaptation to different tasks and computational budgets.
- LLaMA-VID [34] compresses both the instruction and the image into a single token each, resulting in just two tokens per image. This design enables efficient understanding of longer video sequences.
- VoCo-LLaMA [70] utilizes language models to compress all vision tokens, significantly improving computational efficiency while maintaining multimodal understanding.
- TokenPacker [33] introduces a visual projector that adopts a coarse-to-fine strategy to reduce the number of visual tokens by up to 80%, substantially decreasing the computational cost.
- LLaVA-Mini [76] employs a modality pre-fusion module, constructed with Transformer blocks, to integrate visual information into the text tokens in advance. This approach reduces the number of visual tokens fed into the language model, thereby lowering computational cost.

E Limitations and Future Works

As discussed in the Appendix D.2, our proposed progressive consistency distillation framework was only applied during the visual instruction tuning phase when training the multi-modal large language models (MLLMs). While this approach has already demonstrated remarkable effectiveness in terms of both model performance and training efficiency, several promising research directions remain unexplored. For instance, how might our method perform if applied during the model's pretraining stage? Specifically, implementing our framework during projector pretraining, rather than initializing it with publicly available projector weights, could potentially yield greater performance improvements. We hypothesize that this would allow the modality projector's parameter space to adapt to token compression-induced feature space perturbations prior to fine-tuning, thereby further facilitating subsequent supervised fine-tuning. In future work, we will systematically investigate extending our framework to all training stages (including pretraining and supervised finetuning), with rigorous analysis of both the performance gains and impacts on training-time efficiency.

F Broader Impacts

In this work, we propose a framework to develop efficient multi-modal large language models (MLLMs) through Progressive Consistency Distillation, which can be seamlessly integrated with various token compression strategies without modifying the original model architecture. Our approach demonstrates strong effectiveness, robustness, and generalizability. On one hand, this contributes to improving the efficiency of MLLMs, thereby facilitating their deployment and practical applications at a societal level, especially for resource-constrained edge devices. On the other hand, after training under our framework with token-compressed inputs, the resulting model parameters shift from the optimum of the vanilla model (trained on full inputs) to a new optimum suited for compressed inputs. However, we do not further align these models with human preferences after token compression-based training. While the models perform well across a range of multi-modal tasks, they may carry potential risks of adversarial vulnerabilities or undesirable outputs.