Compress & Cache: Vision token compression for efficient generation and retrieval

Adrian Bulat[*1,2] **Yassine Ouali**[*1] **Georgios Tzimiropoulos**[1,3]
[1]Samsung AI Cambridge [2]Technical University of Iasi [3]QMUL

Abstract

This work aims to compress the vision tokens of an LVLM into a representation that is simultaneously suitable for (a) generative and (b) discriminative tasks, (c) is nearly lossless, and (d) storage-efficient. To this end, we propose C&C, a novel compression method that leverages the LVLM itself for task-agnostic visual token compression. Unlike prior methods that perform token reduction on-the-fly, our approach offloads computation to a dedicated, upfront indexing stage, effectively decoupling compression from generation. This enables learning more powerful representations for generation during inference. At the core of C&C is a "double-forward pass" training strategy. During the first forward pass, the LLM (of the LVLM) creates a bottleneck by compressing the dense visual tokens into a few summary tokens. Subsequently, the second forward pass processes the language instruction(s) alongside the summary tokens, used as a direct replacement for the image ones. The training of C&C is guided by two key losses: an autoregressive loss applied after the second pass that provides a direct optimization objective for reconstructing the original information flow, and a contrastive loss applied after the first pass to bolster the representational strength of the summary tokens, particularly for discriminative tasks. Moreover, we propose stage-specific adapters for further enhancing performance. C&C produces highly informative compressed representations. An in-depth ablation study confirms the efficacy of our approach. For generative tasks, we achieve a 2× higher compression rate without compromising capabilities, setting a new state-of-the-art. For discriminative tasks, we establish new state-of-the-art results on image retrieval and compositionality benchmarks.

1 Introduction

Large Vision Language Models (LVLMs) are LLMs [8, 19] that, in addition to text, are capable of integrating and processing visual information as input context [24]. Being able to reason across both vision and language, they are suitable for a wide range of use cases such as image captioning, VQA, and multimodal chatbots. A key bottleneck for their efficient deployment is the large number of input visual tokens, which often dominate the sequence length compared to the language instruction(s). Recent efforts to improve their efficiency have primarily focused on *on-the-fly* token compression [27, 47, 3, 16]. These approaches aim to prune or dynamically condense visual tokens during the online inference process for a given input image and query. While beneficial for single-pass efficiency, these methods operate without a dedicated upfront compression or caching stage and thus limit the capacity of the compressor. Moreover, in the context of retrieval-augmented generation (RAG), they are not aligned with a typical RAG setting, whereby the image and documents are available a priori.

In this work, we explore a different paradigm for LVLM visual token compression that leverages offline processing and caching. Instead of performing token reduction during every inference step, we propose to perform a computationally more intensive compression step once for a given image

[*]Denotes equal contribution.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

to generate a small set of general-purpose summary tokens. These summary tokens are cached and then used directly for subsequent inference queries (*i.e*. online processing, RAG). See Fig. 1. This decoupling of compression and generation allows for a more powerful and versatile representation to be learned during the offline caching phase. Importantly, the representations learned are also suitable for discriminative tasks (*i.e*. retrieval), unlike all prior works, which focus solely on generation.

To this end, we propose Compress and Cache (C&C), a novel token compression approach constructed to support the decoupled indexing (caching) and generation stages. Our core methodological insight and contribution is that the LVLM itself can be adapted to perform the necessary visual compression, leveraging a newly proposed "double-forward pass" training strategy. Specifically, the first forward pass through the LVLM functions as the offline compression phase: trainable summary tokens are processed alongside the image tokens and a predefined prompt guiding the general-purpose visual compression, creating an information bottleneck. The second forward pass simulates the online inference phase: instead of passing the image, the produced summary tokens and the language instruction are fed into the LLM (of the LVLM) for optimization with next-token prediction loss.

A second methodological contribution of ours is to optimize the summary tokens not just for autoregressive generation, but also for discriminative tasks (*e.g*. image-text retrieval). This is achieved through the incorporation of a contrastive loss applied on the summary tokens, after the first forward pass. A significant finding is that this contrastive loss not only enables discriminative capabilities but also proves beneficial for improving the accuracy of the generative tasks.

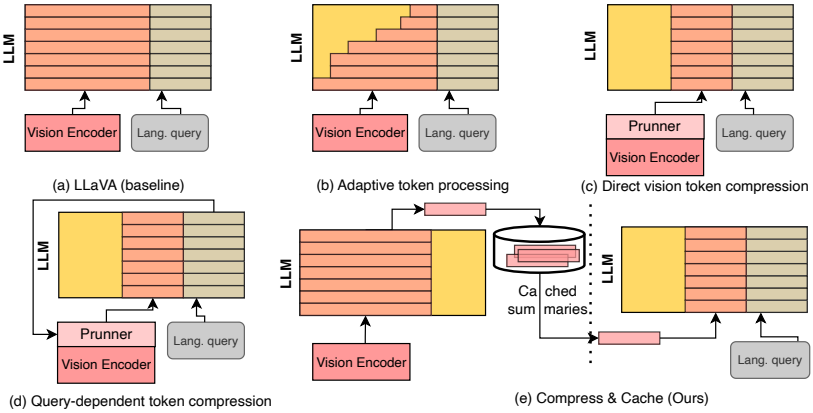

Figure 1: Different paradigms for compressing the visual tokens in LVLMs: (a) LLaVA (baseline) [34]: all vision tokens are used; (b) Adaptive token processing (*e.g*. [4]): the vision tokens are pruned dynamically within each LLM layer. It requires all vision tokens (hence, storage inefficient); (c) Direct vision token compression (*e.g*. [47]): a separate module is learned separately or jointly with the vision encoder; (d) Query-dependent token compression (*e.g*. [27]): the compression depends on the LLM features of each query; (e) Our setting: the LVLM itself produces the compressed representations, trading-off offline processing with superior compression performance.

Overall, **we make the following contributions**:

- We introduce C&C, a novel two-staged method that leverages the LVLM itself for token compression. C&C employs a novel "double-forward pass" training strategy to learn a compact visual representation in the form of condensed summary tokens during an offline phase. The summary tokens are then cached and later used for efficient online inference. C&C effectively disentangles compression from generation, allowing for learning powerful compressed representations.
- We show that the summary tokens learned by C&C are effective for *both* generation and discriminative tasks, such as image-text retrieval. We achieve this dual capability by incorporating a contrastive loss along the autoregressive cross-entropy loss during training. This combined loss is shown to be crucial not only for discrimination but also for enhanced generative performance.
- For generative tasks, C&C achieves a 2× higher compression rate compared to prior methods without compromising capabilities, setting a new state-of-the-art. For discriminative tasks, our method establishes new state-of-the-art results on key image retrieval and compositionality benchmarks. For Visual RAG, we outperform the state-of-the-art VisRAG retriever with a 3.8× smaller model, and almost match the uncompressed baseline for generation, despite using 24× fewer tokens.

2 Related work

Token compression/reduction in LVLMs: While achieving remarkable multimodal capabilities, LVLMs [35, 34, 56, 65, 55, 2, 30, 7] often face significant computational cost, largely due to the LLM having to process a substantial number of visual tokens (*e.g.* 576 in [34]). To alleviate this, recent research has focused on reducing the number of visual tokens fed as input into the LLM [27, 47, 59, 3, 16]. These works operate under an *on-the-fly* paradigm, performing the reduction during inference (see Fig. 1 for a conceptual comparison). Various strategies have been proposed: PruMerge [47] and [64] use training-free heuristics based on spatial token similarities with the global token or with the text query, while methods like [29] and Matryoshka-style techniques [3, 16, 7] train specific modules (e.g., attention layers or convolutions) to learn fixed compressed or nested representations. Other methods implement dynamic token reduction within the LLM layers [55, 4] or condition the reduction on the language query [27]. [57] adapts the attention pattern of the LVLM to store the visual information as part of a compressed KV representation, incurring high storage costs. QueCC [27] conditions its token selection on an LLM-produced embedding derived from the user query, a design that requires recomputation of the visual token reduction for every new instruction.

The proposed C&C fundamentally differs from these prior works by operating under an offline compression and caching paradigm (rather than on-the-fly token reduction), making it ideal for RAG and on-device deployment. Specifically, C&C performs an upfront compression step to generate a task-agnostic, cached summary representation of the image using a newly proposed "double-forward pass" training strategy that leverages the whole LVLM for compression. C&C offers several key advantages. Firstly, it decouples the compression step from online inference, allowing for a more sophisticated offline compression process. Secondly, C&C's summary tokens are optimized (via a contrastive loss) to support both generative and discriminative tasks. This dual capability is a key distinction. Finally, C&C achieves state-of-the-art results surpassing query-dependent methods like QueCC without incurring the computational cost of recomputing visual tokens for every new query.

Discriminative LVLMs: Very recently, a series of works [20, 21, 17] have explored the task of converting LVLMs into discriminative models. For example, [17] directly aligns a pretrained LLM with a pretrained CLIP vision encoder. E5-V [20] through text-only contrastive training converts a generative LVLM into a discriminative one, while [21] expands it to multi-modal retrieval. One major limitation of these works is the loss of generative abilities post-adaptation. In this work, we address this very issue, creating a unified model that excels at both generative and discriminative tasks, surpassing recently proposed LVLM adaptations for image-text retrieval and compositionality.

3 Method

3.1 Preliminaries

We implement C&C on top of LLaVA-1.5 [34], leaving all architectural components unchanged. The LLaVA model consists of a pretrained CLIP vision encoder $g(.)$, a projection matrix $\mathbf{W}$, and an LLM $f(.)$. The input image $\mathbf{X}_v$ is passed to CLIP to produce vision embeddings $\mathbf{H}_v = g(\mathbf{X}_v)\mathbf{W}$. The language embeddings $\mathbf{H}_q$ are obtained from the input language instruction $\mathbf{X}_q$. Finally, the concatenated vision and language embeddings are passed to the LLM to compute the answer (output) embeddings $\mathbf{H}_a = f(\mathbf{H}_v; \mathbf{H}_q)$, which is decoded to the corresponding answer (output) sequence $\mathbf{X}_a$.

Although autoregressive in nature, recently, it was shown that the model can be run in discriminative mode, producing image-text embeddings for matching *à la* CLIP [20]. Using the prompt $\mathbf{X}_p$ "summarize the above image in one word" (or similar), the image embedding is produced as $\mathbf{e}_v = \mathbf{H}_a[-1],\ \mathbf{H}_a = f(\mathbf{H}_\mathbf{v}; \mathbf{H}_p)$ ($\mathbf{H}_p$ is the language embedding of $\mathbf{X}_p$). Analogously, and given a text query $\mathbf{X}_{query}$, the text embedding is constructed as $\mathbf{e}_t = \mathbf{H}_a[-1],\ \mathbf{H}_a = f(\mathbf{H}_{query}, \mathbf{H}_p)$ ($\mathbf{H}_{query}$ is the embedding of a $\mathbf{X}_{query}$). The image-text similarity is computed as $s = \cos\_\text{sim}(\mathbf{e}_v, \mathbf{e}_t)$.

3.2 Double forward bottleneck algorithm

Assuming a fixed LVLM architecture, its inference cost is defined by the input sequence length, which turns out to be dominated by the length (number) k of the vision embeddings $\mathbf{H}_\mathbf{v} \in \mathbb{R}^{k \times d}$. For a LLaVA model, $k = 576$, which is significantly higher than the typical text query length

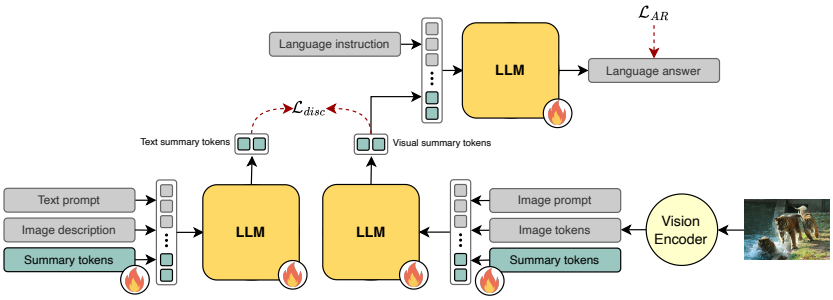

Figure 2: **C&C training pipeline:** A first forward pass from the LLM creates a bottleneck by condensing the visual information into a small number of visual summary tokens. Then, using the same LLM with shared weights, a second forward pass processes the language instruction(s) alongside the summary tokens for training with a next-token prediction loss $\mathcal{L}_{\text{AR}}$ (see Sec. 3.2). A contrastive loss $\mathcal{L}_{\text{disc}}$, applied after the first pass, further boosts the representation strength, especially for discriminative tasks (see Sec. 3.3). Trainable components are marked with 🔥. Note, that different LoRA adapters are used depending on the stage: compression or generation.

and answer [27]. In this work, our goal is to derive a compressed visual token representation $\mathbf{H}_v^c \in \mathbb{R}^{k' \times d}$ where $k' \ll k$ without compromising the model's accuracy. Importantly, besides improving subsequent runs, a small sequence length also opens the path to offline pre-processing, whereby one can pre-compute, cache, and re-use the compressed representation $\mathbf{H}_v^c$ without having to process the original image again.

Departing from all previous approaches for token compression/reduction, we take a totally different path by proposing to leverage the LVLM itself (*i.e.* the LLM of the LVLM) to self-compress the visual tokens. Our motivation for this is multifold. Firstly, LLMs have already excelled at text summarization [63, 36], hence, we propose to utilize them for image summarization (*i.e.* compression). However, as summarization with text, due to quantization (*i.e.*, tokenization), is inefficient with respect to the sequence length, we instead perform this summarization in a continuous latent space. Secondly, the LLM of the LVLM has already been recently utilized to compute discriminant image-text embeddings [20]. However, these embeddings cannot be used for generation. To this end, we propose a "double-forward pass" training strategy whereby visual summary tokens at the output of the model are directly trained to highly compress visual information for both generation and discrimination. See Fig. 2 for an overview.

More specifically, given an input image $\mathbf{X}_v$, we introduce the summary tokens *i.e.* learnable input embeddings $\mathbf{H}_r \in \mathbb{R}^{k' \times d}$ which evolve into the compressed vision embeddings $\mathbf{H}_v^c \in \mathbb{R}^{k' \times d}$ after *a first forward pass from the LLM of the LVLM*:

$$[;,;,\mathbf{H}_v^c] = f(\mathbf{H}_v; \mathbf{H}_p; \mathbf{H}_r), \tag{1}$$

where $\mathbf{H}_p$ are the embeddings of the prompt $\mathbf{X}_p$ "Summarize the image in a few words.". Clearly, during this pass, $\mathbf{H}_r$ interact with both $\mathbf{H}_v$ and $\mathbf{H}_p$. As the transformed embeddings $\mathbf{H}_v^c$ are query-agnostic, for subsequent instructions/queries, the LLM simply takes as input $\mathbf{H}_v^c$ instead of $\mathbf{H}_v$.

To learn the compressed representation $\mathbf{H}_v^c$, during training, we perform *a second forward pass from the LLM of the LVLM* where this time only $\mathbf{H}_v^c$ and the language instructions/embeddings are passed to the LLM. An autoregressive loss is applied at the output of the second forward pass:

$$\mathcal{L}_{\text{AR}} = -\sum_{i=1}^{L} \log\left(p_\theta(x_i | \mathbf{H}_v^c, \mathbf{X}_{q,<i}, \mathbf{X}_{a,<i})\right), \tag{2}$$

where θ are the trainable parameters, $\mathbf{X}_{q,<i}$ and $\mathbf{X}_{a,<i}$ are the query and, respectively, answer tokens located before the current predicted token x_i. $\mathbf{H}_v$ is obtained from Eq. 1. Note that the weights of LLM are shared between the two forward passes. The flow of gradients through $\mathbf{H}_v^c$ results in a single model that can both compress and generate answers by looking solely at the compressed tokens.

Intuitively, our algorithm can also be interpreted as a form of implicit chain-of-thought in the latent space [12], with the LLM "rephrasing" the content of the vision sequence in a condensed manner for itself. Notably, while the input and output spaces of the LLM are not perfectly aligned, they are sufficiently close to resulting in good alignment of the compressed representations in just

a few hundred iterations, making the whole training process efficient. That is, the compressed representations simultaneously lie in the input and output space of the LVLM.

3.3 Discriminative adaptation

Because the compressed representations in C&C lie simultaneously in both the input and output space of the LVLM (unlike previous approaches), this enables us to directly leverage them for CLIP-like discrimination in a zero-shot manner, as detailed in Sec. 3.1. However, in this case, the discriminant performance is suboptimal as there is no explicit loss to encourage the separability of concepts. To address this, and create a unified compressed representation suitable for both generative and discriminative tasks, we also propose to apply a contrastive loss over $\mathbf{H}_v^c$, at the output of the first forward pass. Importantly, this loss also turns out to enhance the generative ability of the model thanks to learning a better underlying representation.

Given a dataset consisting of paired image-text samples, the contrastive loss, for a given batch containing B elements, is defined as:

$$\mathcal{L}_{\text{disc}} = \frac{1}{B}\sum_{k=1}^{b}(-\log\frac{\exp(s_v^{k,k})}{\sum_j \exp(s_v^{k,j})} - \log\frac{\exp(s_t^{k,k})}{\sum_j \exp(s_t^{j,k})}), \tag{3}$$

where $s_v^{k,j} = \cos\_\text{sim}(\mathbf{e}_v^k, \mathbf{e}_t^j)$ computes the cosine similarity between the k-th image and the j-th caption. $\mathbf{e}_v = \frac{1}{k'}\sum \mathbf{H}_v^c$ and, $\mathbf{e}_t = \frac{1}{k'}\sum \mathbf{H}_t^c$, respectively. $\mathbf{H}_t^c$ is computed analogously to $\mathbf{H}_v^c$ (Eq. 1), except that it encodes textual data instead of visual, *i.e.* $\mathbf{H}_t^c = f(\mathbf{H}_{query}, \mathbf{H}_p, \mathbf{H}_r)$. $\mathbf{H}_t^c$ is only used as part of the discriminative loss.

3.4 Overall training loss and data

The final model is trained using both losses, autoregressive and discriminative/contrastive:

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{AR}} + \mathcal{L}_{\text{disc}}. \tag{4}$$

At a given iteration, depending on the sampled training data, the applicable losses are used. That is, for conversational data sampled from the LLaVA-665k dataset, we apply $\mathcal{L}_{\text{AR}}$. For data sampled from CC3M, we apply $\mathcal{L}_{\text{disc}}$. If a conversational sample also has a caption associated with it, both losses are applied within the same iteration. For efficiency, the sampler will group together such cases. This also ensures that we have sufficiently large batches for contrastive training. The sampler aims for a 1:1 ratio between discriminative and autoregressive.

3.5 Stage-specific adaptation

To enable efficient adaptation, we train our models using LoRA [14] adapters which restrict the weight updates to a low-rank representation, $\Delta W = BA, \Delta W \in \mathbb{R}^{d\times m}, B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times m}$, with $r << \min(d, m)$. Although this works well, we use stage-specific adapters to further enhance the plasticity of the LVLM. We distinguish two stages that correspond to the two forward passes used during training: compression, which summarizes $\mathbf{H}_v$ into $\mathbf{H}_v^c$, and generation, which produces $\mathbf{X}_a$ given $\mathbf{H}_v^c$ and $\mathbf{X}_q$. Depending on the stage, different LoRA adapter weights A and B are used.

4 Results

4.1 Generative and Discriminative results with LLaVA-1.5

Implementation details: Our model is LLaVA-1.5 [34], consisting of a ViT-L@336px vision encoder [44] and Vicuna LLM [6] decoder/compressor. Unless otherwise stated, the models are trained for 10,000 iterations, using a batch size of 1024, AdamW [38] with no weight decay and a learning rate of $2e-4$ decayed to 0 using a cosine scheduler. All other layers remain frozen except for the LoRA adapters (rank = 64, α = 128). At a given iteration, depending on the loss (*i.e.* autoregressive or discriminative), we sample a batch either from LLaVA-665K [34] (for generative) or from CC3M [48] (for discriminative). The sampling ratio between the two is 1:1. The training runs were performed on 24 AMD MI300X GPUs using pytorch [43] and deepspeed [45].

Generative benchmarks: Following [34], we evaluate our approach on a diverse collection of datasets, mainly: GQA [18], MMB [37], MME [32], POPE [31], SQA [39], TextVQA [50],

Table 1: Comparison with various token reduction methods on vision-language understanding.

Method	# Tokens	GQA	MMB	MME	POPE	SQA	TextVQA	VisWiz	VQAv2
LLAVA-1.5 [34]	576	62.0	64.3	1510.7	85.9	66.8	58.2	50.0	78.5
PruMerge [47]	≈32	57.2	60.9	1350.3	76.3	68.5	**56.0**	45.2	72.0
TokenPacker [29]	36	59.6	62.8	1440.9	83.3	**71.0**	53.2	50.2	75.0
Matryoshka Multi. [3]	36	60.3	64.8	-	85.5	-	-	52.8	-
Matryoshka Query [16]	36	58.8	63.4	1416.3	81.9	66.8	-	51.0	73.7
QueCC [27]	36	60.5	62.5	1442.0	84.5	70.6	53.3	50.1	75.8
C&C (Ours)	32	**61.6**	64.6	**1472.1**	**85.9**	68.5	55.8	**53.1**	**77.1**
TokenPacker [29]	16	58.9	62.7	1378.8	83.7	68.1	52.5	50.5	74.4
Matryoshka Query [16]	16	57.6	61.9	1408.5	80.8	67.5	-	49.8	71.1
QueCC [27]	16	59.0	62.2	1408.0	83.4	**70.7**	51.3	47.7	74.5
C&C (Ours)	16	**61.0**	**64.4**	**1470.0**	**85.6**	67.7	**54.2**	49.8	**76.5**
TokenPacker [29]	4	56.2	61.5	1347.6	81.7	68.5	49.2	45.7	70.5
Matryoshka Query [16]	4	53.0	56.5	1176.1	77.6	65.1	-	49.4	64.1
QueCC [27]	4	56.5	62.1	1390.3	81.8	**68.6**	48.7	45.0	70.6
C&C (Ours)	4	**58.6**	**63.3**	**1403.0**	**84.3**	67.7	**52.5**	**51.6**	**74.5**

Table 2: Zero-shot text-image retrieval accuracy on Flickr30K, COCO and nocaps.

Method	**image** retrieval						**text** retrieval					
	Flickr30K		COCO		nocaps		Flickr30K		COCO		nocaps	
	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10
Contrastive approaches												
CLIP (`ViT-L`) [44]	67.3	93.3	37.0	71.5	48.6	85.7	87.2	99.4	58.1	87.8	70.0	96.2
BLIP (`ViT-L`) [25]	70.0	95.2	48.4	83.2	62.3	93.4	75.5	97.7	63.5	92.5	72.1	97.7
BLIP2 (`ViT-L`) [26]	74.5	97.2	50.0	86.1	63.0	93.8	86.1	99.4	63.0	93.1	74.4	98.3
OpenCLIP (`ViT-G/14`) [46]	77.8	96.9	48.8	81.5	63.7	93.2	91.5	99.6	66.3	91.8	81.0	98.7
OpenCLIP (`ViT-BigG/14`) [46]	79.5	97.1	51.3	83.0	65.1	93.5	92.9	97.1	67.3	92.6	82.3	98.8
EVA-02-CLIP (`ViT-E/14+`) [51]	78.8	96.8	51.1	82.7	64.5	92.9	93.9	99.8	68.8	92.8	83.0	98.9
EVA-CLIP [52]	80.3	97.2	52.0	82.9	65.3	93.2	94.5	99.7	70.1	93.1	83.5	98.6
LVLM-based approaches												
LLaVA-1.5-7B [34]	59.6	89.3	34.4	69.6	46.9	83.3	65.6	92.3	35.6	70.5	52.1	88.1
E5-V (`LLaVA-1.5-7B`) [20]	76.7	96.9	48.2	82.1	62.0	93.0	86.6	99.0	57.4	88.4	71.9	97.0
VLM2Vec (`Mistral-7B`) [21]	80.1	97.3	52.0	85.6	65.9	94.5	90.3	99.6	68.2	93.2	79.2	98.5
C&C (Ours) (`LLaVA-1.5-7B`)	**83.8**	**98.5**	**56.8**	**86.6**	**70.2**	**96.1**	94.3	**99.9**	**72.9**	**94.4**	**85.7**	**99.5**

VisWiz [11] and VQAv2 [10]. To ensure fairness, in all cases, we fully align the test-time settings and processing with [34]. In addition to this, we also evaluate our approach for captioning on MS-COCO [33], Flickr30k [58] and NoCaps [1], comparing it to token reduction methods that have models openly available. See supplementary material for results on TextCaps [49].

When comparing our approach with the state-of-the-art token reduction methods for visual-language understanding, as the results from Tab. 1 show, we set a new best result, outperforming prior works using 2.25× fewer tokens (16 vs 36). Our results for 32 and even 16 tokens nearly match the uncompressed LlaVA [34] baseline.

Similarly, when evaluated for zero-shot captioning (Tab. 3), our approach matches LLaVA's accuracy, significantly outperforming prior methods. This suggests that the proposed approach encodes more information in its compressed tokens. We note that LLaVA saw some MS-COCO images during training; hence, the MS-COCO evaluation is not fully zero-shot for all methods listed.

Discriminative benchmarks: We evaluate our model on a diverse set of retrieval benchmarks: Flicr30k [58], MS-COCO [33], NoCaps [1] and SugarCrepe [13], against state-of-the-art two-tower independent models. The last one measures the compositional capabilities of the model, an area where CLIP and CLIP-like models tend to underperform.

As Tab. 2 shows, we match and outperform several state-of-the-art contrastive models including larger models, *i.e.* EVA-CLIP (8B vs. 7.06B), despite using 3 orders of magnitude fewer samples for training (2,700M for EVA-CLIP vs. ∼3M for ours). A similar trend can be observed when evaluated for compositionality on SugarCreppe (Tab. 4). Interestingly, models derived from LVLMs (*e.g.*, E5-V and ours) demonstrate superior compositionality. This suggests that the LVLM in discriminative mode inherits the strong vision-language understanding of the underlying generative model.

Table 3: Comparison with various token reduction/compression methods on image captioning in terms of CIDEr score. See supplimentary material for results using additional metrics.

Method	# Tokens	Flickr30K	COCO	nocaps
LLAVA-1.5 [34]	576	81.2	115.4	105.3
PruMerge [47]	≈32	36.3	66.3	58.6
Matryoshka Multi. [3]	36	68.7	102.2	93.6
Matryoshka Query [16]	36	69.5	101.3	90.0
C&C (Ours)	32	**78.9**	**113.1**	**105.9**
Matryoshka Query [16]	16	65.2	99.2	90.0
C&C (Ours)	16	**78.2**	**112.0**	**104.7**
Matryoshka Query [16]	4	47.5	81.0	63.2
C&C (Ours)	4	**74.5**	**111.4**	**103.4**

Table 4: Comparison with state-of-the-art on the SugarCrepe compositionality benchmark.

Method	Params (B)	Replace	Swap	Add
Contrastive approaches				
NegCLIP [61]	0.15	85.0	75.3	85.8
CLIP (`ViT-L`) [44]	0.43	79.5	61.3	74.9
BLIP (`ViT-L`) [25]	0.23	82.4	71.7	88.6
BLIP2 (`ViT-L`) [26]	1.17	85.7	63.8	89.9
OpenCLIP (`ViT-G/14`) [46]	1.37	84.4	67.1	86.8
OpenCLIP (`ViT-BigG/14`) [46]	2.54	86.5	68.9	88.4
EVA-02-CLIP (`ViT-E/14+`) [51]	5.04	86.6	70.7	87.9
EVA-CLIP [52]	8.22	85.9	70.4	86.7
LVLM-based approaches				
LLaVA-1.5-7B [34]	7.06	81.9	59.9	64.7
E5-V (`LLaVA-1.5-7B`) [20]	7.06	88.0	63.5	90.8
VLM2Vec (`Mistral-7B`) [21]	7.3	89.3	67.7	91.7
C&C (Ours) (`LLaVA-1.5-7B`)	7.06	**90.1**	**77.9**	**94.2**

4.2 Visual RAG results with LLaVA-OneVision

Our method offers distinct advantages for Vision-based RAG, stemming from its upfront indexing step and a unified representation for both retrieval and generation. This allows us to use a single model, in contrast to prior methods, which require separate models for retrieval and generation. Below, we compare our approach with the state-of-the-art following the protocol of VisRAG [60].

Implementation details: Since LLaVA-1.5 model is not suitable for high-resolution image analysis, we adopted the improved LLaVA-OneVision-0.5B [23]. We keep the previous hyperparameters fixed, changing only the training data to reflect the nature of the task and model. In particular, for the generative objective, we use the same collection of vision-language datasets encompassing 3.2M samples introduced in [23] and previously used to train the LLaVA-OneVision model that we start from. For the discriminative objective, to allow for fair comparisons, we use the same data as in [60].

Comparison with state-of-the-art: We report results on all datasets from [60] (*i.e.* ArxivQA [28], ChartQA [40], DocVQA [54], InfoVQA [41], PlotQA [42], SlideVQA [53]), first in terms of retrieval and then in terms of retrieval-augmented generation (RAG). For retrieval, as Tab. 11 shows, despite using a 3.8× smaller model, capable of performing generation too, we match and outperform the state-of-the-art VisRAG-Ret [60]. For RAG evaluation, for the generator we consider the following options: MiniCPM-V 2.6 [15], LLaVA-OV-0.5B (original), and our C&C (based on LLaVA-OV-0.5B). For the retriever: VisRAG-Ret [60] and C&C. Notice that in our case, we use the same model for both generation and retrieval. We report in Tab. 6 all combinations formed by them. Our approach gets close to the original LLaVA-OV-0.5B model, using 24× fewer vision tokens. Furthermore, for larger generators (MiniCPM-V 2.6), our improved retriever translates into better RAG performance.

Table 5: Overall retrieval performance in MRR@10. Corresponding Recall@10 performance can be found in the supplementary material. All prior methods results are taken from [60].

Model	# Param.	ArxivQA	ChartQA	DocVQA	InfoVQA	PlotQA	SlideVQA	Average
MiniCPM (OCR) [15]	2.72B	58.43	77.74	72.54	83.45	64.78	91.74	74.78
MiniCPM (Captioner) [15]	2.72B	56.15	74.06	67.57	81.22	55.43	84.27	69.78
SigLIP [2023]	0.88B	59.16	81.34	64.60	74.59	61.32	89.08	71.68
ColPali [2024]	2.92B	72.50	73.49	**82.79**	81.15	55.32	**93.99**	76.54
VisRAG-Ret [60]	3.43B	**75.11**	76.63	75.37	86.37	62.14	91.85	77.91
C&C (Ours)	0.89B	74.63	**87.04**	74.79	**86.40**	**68.73**	90.99	**80.38**

Table 6: Overall generation performance in accuracy (%) using two retrievers: VisRAG-Ret and C&C (Ours). Our variant uses a `LLaVA-OV-0.5B` model and compresses each patch from 768 to 32 tokens. Note that unlike prior works, our model can perform both retrieval and generation.

Generator	Retriever	ArxivQA	ChartQA	DocVQA	InfoVQA	PlotQA	SlideVQA	Average
MiniCPM-V 2.6	VisRAG-Ret	**66.67**	46.88	**54.31**	63.34	47.57	50.54	54.89
MiniCPM-V 2.6	**C&C (Ours)**	66.42	**54.69**	53.33	**64.19**	**49.19**	**50.71**	**56.42**
LLaVA-OV-0.5B	VisRAG-Ret	46.81	31.25	24.72	38.01	21.06	33.21	32.51
C&C (Ours)	VisRAG-Ret	46.32	34.38	22.92	30.91	23.26	28.04	30.97
LLaVA-OV-0.5B	**C&C (Ours)**	45.83	34.38	25.00	38.18	21.18	32.68	32.88
C&C (Ours)	**C&C (Ours)**	45.93	39.06	22.50	30.24	22.92	28.39	31.49

5 Ablation studies & analysis

Impact of each loss function: As detailed in Secs. 3.2 and 3.3, our models are trained using two losses: one autoregressive, applied after the second forward pass, and one contrastive, applied

Table 7: Effect of generative and discriminative losses for generation (MMB, MME, TextVQA) and retrieval (Flickr30K, MS-COCO).

Method	MMB	MME	Text	Flickr30K		MS-COCO	
			VQA	T2I	I2T	T2I	I2T
Discrim.	46.2	624.3	13.5	84.3	94.8	56.3	73.2
Generative	64.1	1420.1	54.2	61.3	76.0	33.9	47.0
Both	64.4	1470.0	54.2	83.8	94.4	56.8	70.2

Table 8: Single vs. stage-specific LoRA v.s full finetuning for generation (MMB, MME, TextVQA) and retrieval (Flickr30K, MS-COCO).

Method	MMB	MME	Text	Flickr30K		MS-COCO	
			VQA	T2I	I2T	T2I	I2T
Fine-tuning	64.3	1413.1	52.9	83.1	94.0	56.2	70.4
Single LoRA	64.3	1410.5	51.8	83.8	94.1	56.5	69.9
Stage LoRA	64.4	1470.0	54.2	83.8	94.4	56.8	70.2

over the compressed tokens, after the first pass. In Tab. 7, we report results for the LLaVA model evaluated using 16 tokens on generative and discriminative tasks. Intuitively, training solely with the discriminative loss (1st row) results in degraded generative performance, as no alignment between the input and output space of the LLM is performed. Moreover, discriminative losses applied over short captions tend to focus on coarse details, missing out on finer-grained details. Conversely, applying only the generative loss (2nd row) results in degraded retrieval abilities, as no loss explicitly encourages concept separation. We note that the longer the training scheduler is, the more pronounced these degradations are for the two cases.

Finally, combining the two losses (3rd row) results in the best performance across the board. Notice that the two losses are complementary when applied jointly and boost the model's accuracy on both sets of benchmarks.

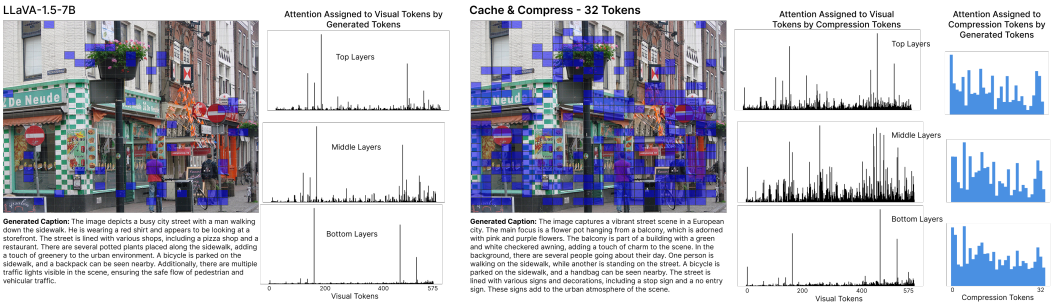

Figure 4: Visualization of attention weights assigned to the 576 visual tokens and the 32 compressed tokens. On the left, we show the cumulative weights assigned to each visual token by the generated tokens for the base model. For C&C, on the right, we first display the per-visual-token weights assigned by the summary tokens during the 1st forward pass for compression. We then show the weights assigned to the compressed tokens by the generated ones during the 2nd forward pass.

Single vs. stage-specific LoRA vs. full fine-tuning: Herein, we compare the effect of training using (a) a single shared LoRA adapter, (b) stage-specific adapters, as proposed in Sec. 3.5, and (c) full fine-tuning. We present the results of these choices in Tab. 8. The best results are obtained using the stage-specific adapters. The fine-tuning run suffers from overfitting to some extent, and its larger training cost makes optimization more difficult.

Fig. 3 further solidifies the need for stage-specific LoRAs, as the optimal representations required during compression (first forward pass) vs downstream inference (second forward pass) are different, especially for earlier layers.

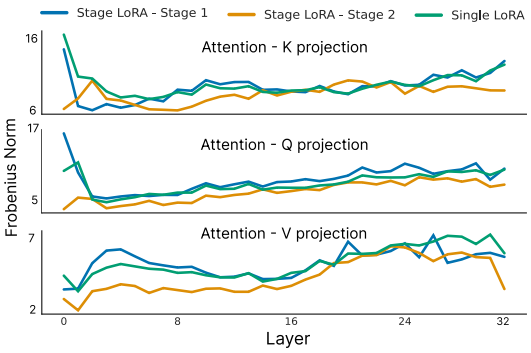

Figure 3: The norm of the learned LoRA weights adjustment $\Delta W = BA$ for a model trained with either a single LoRA or stage-specific LoRAs.

Double forward vs single forward: To showcase the importance of our "double-forward pass" training strategy, we conducted the following experiment: instead of using the LLM itself to compress the vision summary tokens, we use the CLIP vision encoder only. In this case, the loss is directly applied after the LLM, as in LLaVA, using a single forward pass. As shown in Tab. 9, this baseline (1st row) vs ours (2nd row) performs significantly worse.

How does the model's behavior change? To shed light on the changes the model undergoes to act as a self-compressor, we analyze the attention patterns before and after our fine-tuning. The results of this visual analysis are presented in Fig. 4. Looking on the left side, we can observe that LLaVA exhibits a sparse attention pattern across all layers, particularly early on. In contrast, during self-compression, our model attends to all visually important parts of the image, having a significantly denser attention pattern at all layers. Intuitively, in the first case, as the model has access to all tokens, during generation, the model can peek back at the vision tokens as needed. In contrast, during compression, the LLM must ensure that all visually important details are stored in the compressed representation. Finally, on the right-most part of the figure, we showcase the attention pattern between the generated tokens and the compressed representation obtained during text generation from compressed representations. We observe that early and late summary tokens generally receive higher attention weights.

Efficiency analysis: Unlike prior works that perform the compression on-the-fly, our approach offloads the compression cost to a dedicated up-front indexing stage. This disentanglement allows for a more expensive and highly accurate compressor that is run ahead of time. This scenario is aligned with RAG and on-device deployment, where most images from a gallery can be indexed overnight. While we note that some of the methods we compare with could be run offline too (*i.e.* [16, 3]), they (a) don't make this distinction, (b) have significantly worse accuracy, and (c) produce representations unsuitable for retrieval.

Table 9: Double vs single forward pass for generation (MMB, MME, TextVQA) and retrieval (Flickr30K, MS-COCO).

Method	MMB	MME	Text	Flickr30K		MS-COCO	
			VQA	T2I	I2T	T2I	I2T
Single-fwd stage	61.3	1305.0	42.1	80.8	92.1	53.1	66.2
Double-fwd (Ours)	64.4	1470.0	54.2	83.8	94.4	56.8	70.2

With this in mind, in Fig. 5 we report the FLOPs count for a series of state-of-the-art methods during the indexing (caching) and generation phase. The FLOPs count is estimated as in [22, 27] under the following setting: only the prefilling FLOPs are captured, all token compression methods use 16 tokens, the LLaVA-1.5 baseline uses all 576 tokens, for QueCC, which performs query-dependent compression, we assume an average query length of 25 tokens. As the figure shows, our method is the only one to leverage a slower but highly accurate compressor during an indexing phase with a compute cost similar to running the baseline model. During generation, our approach is the fastest as it directly loads the cached tokens, bypassing the need to recompute the vision tokens using the vision encoder and a compression module. In contrast, the current state-of-the-art approach, QueCC [27], requires nearly $2\times$ more FLOPs due to the dependency on the user query/instruction for compression. Moreover, from a storage point of view, in [27], all V (576) tokens must be stored or, alternatively, recomputed if an image from the database is queried again.

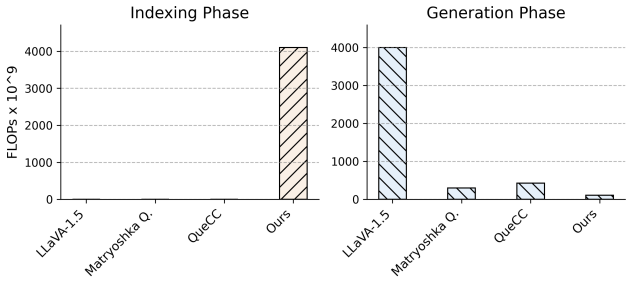

Figure 5: FLOPs estimate for various methods for the indexing and generation phase.

Limitations and broader impact: As our work reduces the inference cost, it allows the deployment of highly performant LVLMs and RAG systems on-device, reducing costs and democratizing the use of AI. In terms of limitations, our work builds on top of existing pre-trained LVLMs. As our goal is to explicitly preserve their characteristics, any potential biases present in the original data and model are likely to propagate to ours too. Therefore, we recommend caution before deploying such models. Moreover, while the proposed method is well-suited for on-device deployment and RAG systems, it's less so for scenarios that don't allow for offline preprocessing and caching.

6 Conclusions

In this work, we introduced C&C, a novel LVLM visual token compression approach that uses the LVLM itself to compress the visual information in a task-agnostic manner, which is trained using a

new "double-forward pass" training strategy. This results in a compressed visual representation that is simultaneously suitable for (a) generative and (b) discriminative tasks, (c) is nearly lossless, and (d) is storage-efficient. Performance-wise, for generative tasks, we offer a 2× higher compression rate without compromising the generative capabilities, setting a new state-of-the-art. For discriminative tasks, we also set a new state-of-the-art result on image retrieval and compositionality.

References

[1] Harsh Agrawal, Karan Desai, Yufei Wang, Xinlei Chen, Rishabh Jain, Mark Johnson, Dhruv Batra, Devi Parikh, Stefan Lee, and Peter Anderson. Nocaps: Novel object captioning at scale. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8948–8957, 2019. 6

[2] 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. 3

[3] Mu Cai, Jianwei Yang, Jianfeng Gao, and Yong Jae Lee. Matryoshka multimodal models. In *Workshop on Video-Language Models@ NeurIPS 2024*, 2024. 1, 3, 6, 7, 9, 22, 23

[4] 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. 2, 3

[5] Shuo Chen, Jindong Gu, Zhen Han, Yunpu Ma, Philip Torr, and Volker Tresp. Benchmarking robustness of adaptation methods on pre-trained vision-language models. *Advances in Neural Information Processing Systems*, 36:51758–51777, 2023. 24

[6] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023. 5

[7] 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. 3

[8] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 1

[9] Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. Colpali: Efficient document retrieval with vision language models. In *The Thirteenth International Conference on Learning Representations*, 2024. 7, 21, 22

[10] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6904–6913, 2017. 6

[11] Danna Gurari, Qing Li, Abigale J Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P Bigham. Vizwiz grand challenge: Answering visual questions from blind people. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3608–3617, 2018. 6

[12] Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024. 4

[13] Cheng-Yu Hsieh, Jieyu Zhang, Zixian Ma, Aniruddha Kembhavi, and Ranjay Krishna. Sugarcrepe: Fixing hackable benchmarks for vision-language compositionality. *Advances in neural information processing systems*, 36, 2024. 6

[14] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 5

[15] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, et al. Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395*, 2024. 7, 21, 22

[16] Wenbo Hu, Zi-Yi Dou, Liunian Harold Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. Matryoshka query transformer for large vision-language models. *arXiv preprint arXiv:2405.19315*, 2024. 1, 3, 6, 7, 9, 22, 23

[17] Weiquan Huang, Aoqi Wu, Yifan Yang, Xufang Luo, Yuqing Yang, Liang Hu, Qi Dai, Xiyang Dai, Dongdong Chen, Chong Luo, et al. Llm2clip: Powerful language model unlock richer visual representation. *arXiv preprint arXiv:2411.04997*, 2024. 3

[18] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709, 2019. 5

[19] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023. 1

[20] Ting Jiang, Minghui Song, Zihan Zhang, Haizhen Huang, Weiwei Deng, Feng Sun, Qi Zhang, Deqing Wang, and Fuzhen Zhuang. E5-v: Universal embeddings with multimodal large language models. *arXiv preprint arXiv:2407.12580*, 2024. 3, 4, 6, 7, 22, 23

[21] Ziyan Jiang, Rui Meng, Xinyi Yang, Semih Yavuz, Yingbo Zhou, and Wenhu Chen. Vlm2vec: Training vision-language models for massive multimodal embedding tasks. *arXiv preprint arXiv:2410.05160*, 2024. 3, 6, 7, 23

[22] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020. 9

[23] 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. 7

[24] Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *arXiv preprint arXiv:2407.07895*, 2024. 1

[25] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pages 12888–12900. PMLR, 2022. 6, 7, 23

[26] 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. 6, 7, 23

[27] Kevin Y Li, Sachin Goyal, Joao D Semedo, and J Zico Kolter. Inference optimal vlms need only one visual token but larger models. *arXiv preprint arXiv:2411.03312*, 2024. 1, 2, 3, 4, 6, 9

[28] Lei Li, Yuqi Wang, Runxin Xu, Peiyi Wang, Xiachong Feng, Lingpeng Kong, and Qi Liu. Multimodal arxiv: A dataset for improving scientific comprehension of large vision-language models. *arXiv preprint arXiv:2403.00231*, 2024. 7

[29] 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. 3, 6

[30] Yanwei Li, Yuechen Zhang, Chengyao Wang, Zhisheng Zhong, Yixin Chen, Ruihang Chu, Shaoteng Liu, and Jiaya Jia. Mini-gemini: Mining the potential of multi-modality vision language models. *arXiv preprint arXiv:2403.18814*, 2024. 3

[31] Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. *arXiv preprint arXiv:2305.10355*, 2023. 5

[32] Zijing Liang, Yanjie Xu, Yifan Hong, Penghui Shang, Qi Wang, Qiang Fu, and Ke Liu. A survey of multimodel large language models. In *Proceedings of the 3rd International Conference on Computer, Artificial Intelligence and Control Engineering*, pages 405–409, 2024. 5

[33] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 6

[34] 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. 2, 3, 5, 6, 7, 21, 22, 23

[35] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36, 2024. 3

[36] Yixin Liu, Alexander R Fabbri, Pengfei Liu, Yilun Zhao, Linyong Nan, Ruilin Han, Simeng Han, Shafiq Joty, Chien-Sheng Wu, Caiming Xiong, et al. Revisiting the gold standard: Grounding summarization evaluation with robust human evaluation. *arXiv preprint arXiv:2212.07981*, 2022. 4

[37] 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? In *European conference on computer vision*, pages 216–233. Springer, 2024. 5

[38] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[39] 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. *Advances in Neural Information Processing Systems*, 35: 2507–2521, 2022. 5

[40] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 7

[41] Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706, 2022. 7

[42] Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. Plotqa: Reasoning over scientific plots. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1527–1536, 2020. 7

[43] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[44] 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. 5, 6, 7, 23

[45] Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 3505–3506, 2020. 5

[46] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. *Advances in Neural Information Processing Systems*, 35:25278–25294, 2022. 6, 7, 23

[47] 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. 1, 2, 3, 6, 7, 22, 23

[48] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2556–2565, 2018. 5

[49] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. Textcaps: a dataset for image captioning with reading comprehension. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 742–758. Springer, 2020. 6, 22

[50] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 8317–8326, 2019. 5

[51] Quan Sun, Yuxin Fang, Ledell Wu, Xinlong Wang, and Yue Cao. Eva-clip: Improved training techniques for clip at scale. *arXiv preprint arXiv:2303.15389*, 2023. 6, 7, 23

[52] Quan Sun, Jinsheng Wang, Qiying Yu, Yufeng Cui, Fan Zhang, Xiaosong Zhang, and Xinlong Wang. Eva-clip-18b: Scaling clip to 18 billion parameters. *arXiv preprint arXiv:2402.04252*, 2024. 6, 7, 23

[53] Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. Slidevqa: A dataset for document visual question answering on multiple images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13636–13645, 2023. 7

[54] Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. Hierarchical multimodal transformers for multipage docvqa. *Pattern Recognition*, 144:109834, 2023. 7

[55] Weihan Wang, Qingsong Lv, Wenmeng Yu, Wenyi Hong, Ji Qi, Yan Wang, Junhui Ji, Zhuoyi Yang, Lei Zhao, Xixuan Song, et al. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*, 2023. 3

[56] Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*, 2023. 3

[57] 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. 3

[58] Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014. 6

[59] Gaotong Yu, Yi Chen, and Jian Xu. Balancing performance and efficiency: A multimodal large language model pruning method based image text interaction. *arXiv preprint arXiv:2409.01162*, 2024. 3

[60] Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, et al. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. *arXiv preprint arXiv:2410.10594*, 2024. 7, 21, 22

[61] Mert Yuksekgonul, Federico Bianchi, Pratyusha Kalluri, Dan Jurafsky, and James Zou. When and why vision-language models behave like bags-of-words, and what to do about it? *arXiv preprint arXiv:2210.01936*, 2022. 7, 23

[62] 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. 7, 21, 22

[63] Tianyi Zhang, Faisal Ladhak, Esin Durmus, Percy Liang, Kathleen McKeown, and Tatsunori B Hashimoto. Benchmarking large language models for news summarization. *Transactions of the Association for Computational Linguistics*, 12:39–57, 2024. 4

[64] 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. 3

[65] 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. 3

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: All claims made do accurately reflect the paper's contributions, with the proposed novel components detailed in Section 3. The claims are experimentally validated in detail in Section 4, with ample evaluations on both generative and discriminative tasks.

   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: A limitation section/paragraph was added before conclusions.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [NA] .

Justification: No theoretical results are included.

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: All implementation details are listed in Section 4.

   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: No code is provided alongside the submission, however all details are provided to allow for full reproducibility.

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: They are detailed as part of the results section (Section 4) and supplementary material.

Guidelines:

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

7. **Experiment statistical significance**

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

Answer: [No] .

Justification: Due to the cost of conducting the experiments no statistical significance is reported.

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 provide the amount and type of GPUs used to conduct the experiments in Section 4.

   Guidelines:

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

9. **Code of ethics**

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

   Answer:[Yes] .

   Justification: The work conducted is in full compliance with the ethics guideliness.

   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: Yes, included in a separate paragraph, right before conclusions.

    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: We do not train any models from scratch. Instead we finetune existing model with the goal of mantaining their original behaviour unchanged while increasing their efficiency.

    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: All datasets and tools used are appropiately cited throughout the paper.

    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: No new assets are introduced.

    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 work conducted 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: Not applicable.

    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: No LLM were used to help develop this work.

    Guidelines:

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

A Additional results and comparisons

Results for larger LLaVA-1.5 models: In the main manuscript, we conduct experiments using a LLaVA-1.5 (7B) model. Herein, we validate how our approach behaves when using a larger model, *i.e*. a LLaVA-1.5 (13B). As the results from Table 10 show, the proposed method nearly matches the full LLaVA model's accuracy using only 16 and even 4 tokens in this case, too.

Table 10: Token compression performance on vision-language understanding tasks using a LLaVA-1.5 13B model.

Method	**# Tokens**	**GQA**	**MMB**	**MME**	**POPE**	**SQA**	**TextVQA**	**VisWiz**	**VQAv2**
LLAVA-1.5 [34]	576	63.3	67.7	1531.0	86.2	71.6	61.3	53.6	80.0
C&C (Ours)	32	62.2	67.6	1465.1	85.3	72.5	59.6	54.0	78.7
C&C (Ours)	16	61.8	67.3	1473.5	85.0	72.4	57.5	54.2	78.4
C&C (Ours)	4	59.9	66.4	1390.1	84.4	71.1	53.6	52.7	75.9

Visual RAG results with Larger LLaVA-OneVision models In addition to the Visual RAG results from the main manuscript, which use a 0.5B LLaVA-OV skew, herein we report results for a larger 7B model (*i.e*. LLaVA-OV-7B). As shown in Table 11 and 12, for retrieval, our approach outperforms prior works in terms of both MRR@10 and Recall@10, showing good scaling with respect to the model size and establishing a new state-of-the-art result.

Table 11: Overall retrieval performance in terms of MRR@10. All prior methods results are taken from [60].

Model	# Param.	ArxivQA	ChartQA	DocVQA	InfoVQA	PlotQA	SlideVQA	Average
MiniCPM (OCR) [15]	2.72B	58.43	77.74	72.54	83.45	64.78	91.74	74.78
MiniCPM (Captioner) [15]	2.72B	56.15	74.06	67.57	81.22	55.43	84.27	69.78
SigLIP [2023]	0.88B	59.16	81.34	64.60	74.59	61.32	89.08	71.68
ColPali [2024]	2.92B	72.50	73.49	82.79	81.15	55.32	93.99	76.54
VisRAG-Ret [60]	3.43B	75.11	76.63	75.37	86.37	62.14	91.85	77.91
C&C (`LLaVA-OV-0.5B`) **(Ours)**	0.89B	74.63	87.04	74.79	86.40	68.73	90.99	80.38
C&C (`LLaVA-OV-7B`) **(Ours)**	7.0B	**83.65**	**90.56**	**85.77**	**91.97**	**71.41**	**95.07**	**86.41**

Table 12: Overall retrieval performance in terms of Recall@10. All prior methods results are taken from [60].

Model	# Param.	ArxivQA	ChartQA	DocVQA	InfoVQA	PlotQA	SlideVQA	Average
MiniCPM (OCR) [15]	2.72B	69.36	88.89	87.14	94.15	90.61	96.85	87.83
MiniCPM (Captioner) [15]	2.72B	69.00	85.71	84.26	94.29	84.24	93.08	85.10
SigLIP [2023]	0.88B	73.90	92.06	83.08	93.04	89.57	94.15	87.63
ColPali [2024]	2.92B	82.72	88.89	94.75	94.43	80.30	97.21	89.72
VisRAG-Ret [60]	3.43B	87.25	90.48	91.20	97.08	89.80	97.39	92.20
C&C (`LLaVA-OV-0.5B`) **(Ours)**	0.89B	84.64	92.06	91.71	97.21	93.51	96.67	92.97
C&C (`LLaVA-OV-7B`) **(Ours)**	7.0B	**93.38**	**98.41**	**96.45**	**98.75**	**94.67**	**98.47**	**96.69**

Additional discriminative comparisons with other token-summarization approaches. We note that our approach is the only one that compresses the vision tokens into a representation suitable for both generative and discriminative tasks, requiring no additional forward passes. However, herein, for completeness, we evaluate on our suite of discriminative tasks the current state-of-the-art token compression models that offered pretrained models. This is achieved by following the zero-shot setup described in the main manuscript in Section 3.1 and [20]. Unsurprisingly, as the results from Table 13 and 14 show, our approach significantly surpasses the other methods we compare with.

Table 13: Zero-shot text-image retrieval accuracy on Flickr30K, COCO and nocaps. Only our approach is specialised for both retrieval and generation, hence, except for our method, all other results are in a zero-shot manner following the protocol described in the main manuscript in Section 3.1 and [20].

Method	**Tokens**	**image** retrieval						**text** retrieval					
		Flickr30K		COCO		nocaps		Flickr30K		COCO		nocaps	
		R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10	R@1	R@10
LLaVA-1.5-7B [34]	576	59.6	89.3	34.4	69.6	46.9	83.3	65.6	92.3	35.6	70.5	52.1	88.1
PruMerge [47]	18	34.7	67.9	18.4	47.9	25.8	62.7	38.3	74.3	19.8	49.9	28.2	65.2
Matryoshka Multi. [3]	16	57.9	88.5	34.1	69.7	45.5	83.2	63.8	91.7	36.4	72.5	48.0	86.2
Matryoshka Query [16]	16	53.6	85.9	29.8	65.4	40.5	80.0	59.4	90.3	34.1	69.6	45.4	84.7
C&C (Ours)	16	**83.8**	**98.5**	**59.0**	**88.6**	**72.3**	**96.5**	94.3	**99.9**	**72.9**	**94.4**	**85.7**	**99.5**

Table 14: Comparison on the SugarCrepe compositionality benchmark.

Method	Tokens	Replace			Swap		Add	
		Object	Attribute	Relation	Object	Attribute	Object	Attribute
LLaVA-1.5-7B [34]	576	88.0	81.6	76.1	60.9	58.8	67.0	62.4
PruMerge [47]	18	88.0	74.4	69.7	62.5	57.3	81.4	66.0
Matryoshka Multi. [3]	16	90.3	81.4	80.1	70.2	67.9	75.7	75.8
Matryoshka Query [16]	16	89.3	81.4	79.2	70.6	64.7	73.8	73.6
C&C (Ours) (`LLaVA-1.5-7B`)	7.06	**98.1**	**89.5**	**82.7**	**77.8**	**78.1**	**95.3**	**93.1**

Additional zero-shot image captioning evaluations: In addition to the evaluation from the main manuscript, herein, we evaluate our approach for zero-shot captioning on TextCaps [49], a dataset for image captioning with reading comprehension. As the results from Table 15 show, we generally match the full-tokens LLaVA's model performance. Importantly, our results remain stable as the number of compressed tokens decreases.

In-depth evalution results for captioning: In the main manuscript we report results for image captioning solely in terms of CIDEr score. For completeness, in Table 16 we also report Bleu@4 (B@4), METEOR (MET.), and ROUGE. The conclusions hold across all metrics.

In-depth evalution results for captioning: In the main manuscript we report results for image captioning solely in terms of CIDEr score. For completeness, in Table 16 we also report Bleu@4 (B@4), METEOR (MET.), and ROUGE. The conclusions hold across all metrics.

B Additional ablation studies and analyses

What do the compressed tokens encode? The compressed representation gradually encodes, from left to right, coarser to finer-grained concepts. This effect can be observed in Fig. 6, where, as the

Table 15: Comparison with various token compression methods on TextCaps dataset for image captioning in terms of BLEU-4 (B@4), CIDEr score, METEOR (MET.) and ROUGE-L.

Method	Tokens	B@4	CIDEr	MET.	ROUGE
LLAVA-1.5 [34]	576	27.1	90.4	21.9	46.2
PruMerge [47]	≈32	17.6	62.8	17.0	39.7
Matryoshka Multi. [3]	36	25.1	94.8	23.0	46.3
Matryoshka Query [16]	36	21.0	70.0	19.9	42.6
C&C (Ours)	32	26.5	90.6	22.4	46.1
Matryoshka Query [16]	16	20.1	62.5	19.3	41.7
C&C (Ours)	16	26.4	90.5	22.5	46.3
Matryoshka Query [16]	4	15.2	42.0	16.5	37.4
C&C (Ours)	4	25.4	86.1	22.0	45.7

Table 16: Comparison with various token reduction/compression methods on image captioning.

Method	# Tokens	Flickr30K				COCO				nocaps			
		B@4	CIDEr	MET.	ROUGE	B@4	CIDEr	MET.	ROUGE	B@4	CIDEr	MET.	ROUGE
LLAVA-1.5 [34]	576	30.6	81.2	25.0	53.4	32.9	115.4	27.7	56.3	42.9	105.3	28.9	59.8
PruMerge [47]	≈32	18.5	36.3	15.7	40.2	18.5	66.3	18.8	44.9	25.9	58.6	20.0	47.8
Matryoshka Multi. [3]	36	25.4	68.7	24.1	49.9	27.7	102.2	27.2	53.3	36.8	93.6	28.0	56.5
Matryoshka Query [16]	36	26.4	69.5	23.1	50.0	28.0	101.3	26.2	52.7	36.2	90.0	26.8	55.8
C&C (Ours)	32	**30.0**	**78.9**	**25.2**	**52.9**	**31.5**	**113.1**	**27.9**	**55.6**	**42.5**	**105.9**	**29.2**	**59.6**
Matryoshka Query [16]	16	24.8	65.2	22.7	49.0	27.6	99.2	26.0	52.5	36.2	90.0	26.8	55.8
C&C (Ours)	16	**29.0**	**78.2**	**25.3**	**52.7**	**31.0**	**112.0**	**27.9**	**55.4**	**42.0**	**104.7**	**29.3**	**59.5**
Matryoshka Query [16]	4	20.1	47.5	19.8	44.5	23.2	81.0	23.0	48.6	28.4	63.2	21.1	49.5
C&C (Ours)	4	**28.4**	**74.5**	**24.8**	**51.8**	**31.1**	**111.4**	**27.9**	**55.4**	**41.1**	**103.4**	**29.0**	**59.1**

Table 17: Comparison with state-of-the-art on the SugarCrepe compositionality benchmark.

Method	Params (B)	Replace			Swap		Add	
		Object	Attribute	Relation	Object	Attribute	Object	Attribute
Contrastive approaches								
NegCLIP [61]	0.15	92.7	85.9	76.5	75.2	75.4	88.8	82.8
CLIP (ViT-B) [44]	0.15	90.9	80.1	69.2	61.4	64.0	77.2	68.8
CLIP (ViT-L) [44]	0.43	94.1	79.2	65.2	60.2	62.3	78.3	71.5
BLIP (ViT-L) [25]	0.23	96.5	81.7	69.1	66.6	76.8	92.0	85.1
BLIP2 (ViT-L) [26]	1.17	97.6	81.7	77.8	62.1	65.5	92.4	87.4
OpenCLIP (ViT-G/14) [46]	1.37	95.8	85.0	72.4	63.0	71.2	91.5	82.1
OpenCLIP (ViT-BigG/14) [46]	2.54	96.6	87.9	74.9	62.5	75.2	92.2	84.5
EVA-02-CLIP (ViT-E/14+) [51]	5.04	97.1	88.5	74.2	67.3	74.1	91.8	83.9
EVA-CLIP [52]	8.22	96.4	86.6	74.8	66.1	74.6	91.3	82.0
LVLM-based approaches								
LLaVA-1.5-7B [34]	7.06	88.0	81.6	76.1	60.9	58.8	67.0	62.4
E5-V (LLaVA-1.5-7B) [20]	7.06	95.8	86.6	81.6	62.9	64.0	93.5	88.0
VLM2Vec (Mistral-7B) [21]	7.3	97.2	89.0	81.7	62.9	72.5	94.7	88.6
C&C (Ours) (LLaVA-1.5-7B)	7.06	**98.1**	**89.5**	**82.7**	**77.8**	**78.1**	**95.3**	**93.1**

number of tokens increases, the caption generated correctly captures more elements present in the photo, importantly reducing hallucinations. This effect is also corroborated in Fig. 7. There, we mask out different groups of (4 and 8) tokens, quantitatively measuring the impact of this: earlier tokens induce larger drops in performance (*e.g.* masking the first 8 tokens reduces performance by 10%). However, the performance does not drop to (near) 0, which suggests that there is also some degree of redundancy. The observed behavior can largely be attributed to the causal attention masking used by the LVLM, which encourages a directional information distribution.

Full-attention vs causal: Vicuna, and hence LLaVA, much like the rest of the generative LLMs, employs causal attention masking in order to restrict the past states from attending the future ones. While necessary for autoregressive modeling, it's unclear why it would be for vision token compression, too, as there is no preferential direction for image processing. Hence, herein, we explore the effect of changing the attention pattern from causal to bidirectional (*i.e.* full) attention for the compression forward pass, while keeping it causal for the subsequent answer generation ones. In this instance, the stage (*i.e.* compression vs generative) specific LoRAs also take the role of adjusting

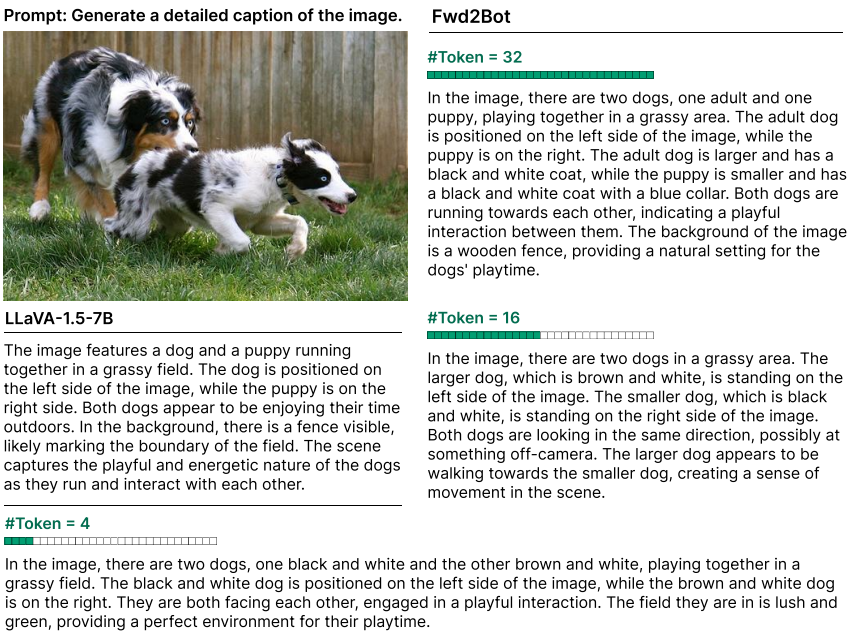

Figure 6: Captioning with variable number of summary tokens.

the attention pattern and information flow. Analyzing the results from Table 18 we can observe performance gains for discriminative tasks and degradation for generative ones. This suggests that a direct finetuning under a different attention pattern is suboptimal, likely requiring a pre-alignment step. Moreover, the LoRA adapters may limit the ability of the model to shift its attention pattern.

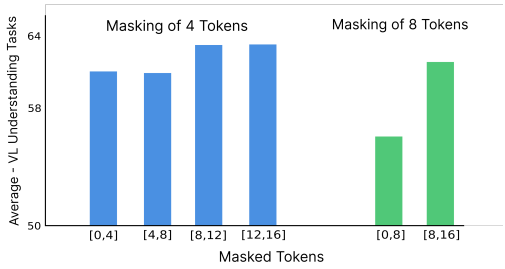

Figure 7: The relative importance of different subsets of visual tokens. We show the mean over the VL understanding tasks when masking specific subsets of the compressed visual tokens.

Table 18: Compression with Bidirectional vs Causal attention for generation (MMB, MME, TextVQA) and retrieval (Flickr30K, MS-COCO).

Method	MMB	MME	TextVQA	Flickr30K		MS-COCO	
				T2I	I2T	T2I	I2T
Bidirectional	60.2	1310.1	48.4	83.6	94.8	57.9	72.2
Causal	64.4	1470.0	54.2	83.8	94.4	56.8	70.2

Finetuning checkpoint choice: The natural starting point for our approach is the LLaVA model itself. However, for completeness, we also try to directly finetune from the Vicuna LLM itself. As the results from Table 19 show, starting from a model already optimized for vision-language understanding results in a notable performance boost. To compensate for this, likely, a longer training scheduler is needed and potentially a full model finetuning, as in LLaVA.

Robustness to noisy inputs: Following [5] we evaluate our approach under a various set of perturbations, e.g: zoom blur, elastic transformation, pixelation, JPEG compression, shot noise, brightness

Table 19: Impact of the pre-trained checkpoint for generation (MMB, MME, TextVQA) and retrieval (Flickr30K, MS-COCO).

Method	MMB	MME	TextVQA	Flickr30K		MS-COCO	
				T2I	I2T	T2I	I2T
Vicuna	60.3	1296.3	48.2	81.2	92.5	54.3	67.4
LLaVA	64.4	1470.0	54.2	83.8	94.4	56.8	70.2

jitter, contrast jitter, Gaussian noise, etc. For brevity, we include in Table 20 the results in terms of relative performance drop on a subset of them. Notice that both approaches, with and without compression, have similar robustness strength.

Latency measurements: In the main manuscript we reported FLOPs estimates because the timings themselves are subject to the specific implementation and underlying hardware architecture. For completeness, we benchmark the LLaVA-1.5 7B model on a RTX 4090 GPU. Each result is averaged over 100 runs following a warm-up period. Original LLaVA model cost: 0.0587 sec/image (out of which 0.00353 sec spent for the vision encoder); Caching cost: 0.0584 sec/image; Online C&C cost (16 tokens): 0.00158 sec/image; Online C&C cost (4 tokens): 0.000406 sec/image; Caching cost + Online C&C (16 tokens): 0.0599 sec/image; Caching cost + Online C&C (4 tokens): 0.0588 sec/image. Our approach is significantly faster once the embeddings are cached and comparable with the LLaVA baseline during caching.

C Additional details regarding the test-time inference

In Fig. 8, we depict the test-time inference flow for generative and discriminative tasks. The first step compresses the given image $\mathbf{X}_v$ into its compressed representation $\mathbf{H}_v^c$. This representation is then stored in a database. Note that while $\mathbf{H}_v^c$ can be computed on the fly, too, the scenario we are mostly interested in is pre-indexing, whereby the image representations are computed offline ahead of time.

Once stored, we can directly operate on this compressed representation for both generative and discriminative tasks. For generative tasks, the same LLM used for compression takes as input a user-provided instruction and the compressed image representation, producing an answer autoregressively (Fig. 8, top-right corner). For discriminative tasks, in order to measure the text-to-image similarity, á la CLIP, and again using the same LLM, we pass the user query (image description) to the LLM, producing a set of embeddings. We can measure the similarity between the given image description by taking the cosine similarity between the sum of the precomputed compressed vision tokens and the text embeddings newly produced by the LLM (Fig. 8, left side).

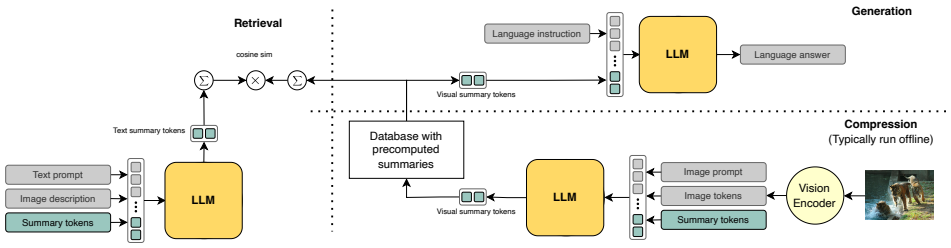

Figure 8: Test-time inference, depicting: compression (lower-right), generation (upper-right), and discrimination (left). Notice that in all cases we use the same LLM. The compressed embeddings are suitable for both sets of tasks and are generally expected to be pre-computed offline.

Table 20: Relative accuracy drop under various noise types across different datasets.

Noise type	**MMB**	**MME**	**POPE**	**SQA**	**TextVQA**	**realworldQA**
Zoom Blur (baseline)	20.45	0.0	0.0	7.31	0.0	17.15
Zoom Blur (compressed)	16.91	0.0	0.0	6.70	0.0	13.44
Snow (baseline)	11.04	0.0	0.0	2.51	0.0	7.73
Snow (compressed)	10.79	0.0	0.0	2.23	0.0	12.50
Defocus Blur (baseline)	12.50	0.0	0.0	3.17	0.0	7.49
Defocus Blur (compressed)	11.81	0.0	0.0	2.09	0.0	12.72
Blank Image (baseline)	73.38	42.21	43.42	9.52	90.14	22.95
Blank Image (compressed)	72.45	41.87	43.32	11.45	89.31	24.82
Saturate (baseline)	0.16	0.0	0.0	1.70	0.0	2.90
Saturate (compressed)	0.73	0.0	0.0	1.26	0.0	0.45
Elastic Transform (baseline)	5.52	0.0	0.0	2.36	0.0	3.62
Elastic Transform (compressed)	4.52	0.0	0.0	0.42	0.0	4.91
Pixelate (baseline)	8.44	1.99	9.00	0.89	68.08	13.77
Pixelate (compressed)	7.58	2.52	9.35	3.35	67.93	11.64
Spatter (baseline)	7.47	4.94	1.94	1.18	12.26	6.52
Spatter (compressed)	4.96	1.75	2.41	0.77	8.39	7.81
Speckle Noise (baseline)	10.88	3.01	3.29	2.36	15.24	8.21
Speckle Noise (compressed)	11.66	0.37	3.48	1.89	14.30	10.04
JPEG Compression (baseline)	2.60	-0.89	2.24	0.0	5.68	4.83
JPEG Compression (compressed)	1.60	1.80	2.82	-0.63	3.72	4.48
Shot Noise (baseline)	12.66	3.46	4.69	1.62	16.83	9.18
Shot Noise (compressed)	11.52	2.01	4.57	0.35	16.27	10.05
Impulse Noise (baseline)	12.01	4.64	4.35	1.99	16.30	8.21
Impulse Noise (compressed)	9.04	5.74	4.79	0.77	14.74	9.60
Brightness (baseline)	3.90	0.0	0.0	0.66	0.0	3.62
Brightness (compressed)	2.92	0.0	0.0	-0.07	0.0	-0.45
Contrast (baseline)	3.08	3.06	2.11	1.55	4.63	5.80
Contrast (compressed)	5.10	2.66	1.70	0.14	4.12	6.25
Gaussian Noise (baseline)	12.01	4.58	4.75	1.92	15.33	7.97
Gaussian Noise (compressed)	12.24	4.37	4.81	0.28	13.82	7.05
Motion Blur (baseline)	12.34	4.33	5.73	3.03	0.0	6.76
Motion Blur (compressed)	12.10	4.58	6.10	2.51	0.0	8.28