

Rethinking Token Reduction for Large Vision-Language Models

Yi Wang^{1*}, Haofei Zhang^{2,3*}, Qihan Huang¹, Anda Cao¹, Gongfan Fang⁵,
Wei Wang⁶, Xuan Jin⁶, Jie Song⁴, Mingli Song^{1,2,3,4}, Xinchao Wang^{5†}

¹College of Computer Science and Technology, Zhejiang University

²State Key Laboratory of Blockchain and Data Security, Zhejiang University

³Hangzhou High-Tech Zone (Binjiang) Institute of Blockchain and Data Security

⁴School of Software Technology, Zhejiang University

⁵National University of Singapore ⁶Alibaba Group

Abstract

*Large Vision-Language Models (LVLMs) excel in visual understanding and reasoning, but the excessive visual tokens lead to high inference costs. Although recent token reduction methods mitigate this issue, they mainly target single-turn Visual Question Answering (VQA), leaving the more practical multi-turn VQA (MT-VQA) scenario largely unexplored. MT-VQA introduces additional challenges, as subsequent questions are unknown beforehand and may refer to arbitrary image regions, making existing reduction strategies ineffective. Specifically, current approaches fall into two categories: prompt-dependent methods, which bias toward the initial text prompt and discard information useful for subsequent turns; prompt-agnostic ones, which, though technically applicable to multi-turn settings, rely on heuristic reduction metrics such as attention scores, leading to suboptimal performance. In this paper, we propose a learning-based prompt-agnostic method, termed *MetaCompress*, overcoming the limitations of heuristic designs. We begin by formulating token reduction as a learnable compression mapping, unifying existing formats such as pruning and merging into a single learning objective. Upon this formulation, we introduce a data-efficient training paradigm capable of learning optimal compression mappings with limited computational costs. Extensive experiments on MT-VQA benchmarks and across multiple LVLM architectures demonstrate that *MetaCompress* achieves superior efficiency–accuracy trade-offs while maintaining strong generalization across dialogue turns. Our code is available at <https://github.com/MARShall147/MetaCompress>.*

1. Introduction

LVLMs [1, 4, 35–37, 69, 70] have become powerful AI systems enabling natural human interaction with visual data such as images and videos. They encode both textual and

visual modalities into tokens jointly processed by a unified Large Language Model (LLM). Recent works [37, 70] further extend LVLMs toward multi-scale visual inputs that integrate both global and local tokens to enhance visual understanding. However, visual tokens greatly increase computation and memory costs, as token numbers grow by thousands and attention scales quadratically with sequence length [13, 30], making low-latency or resource-constrained deployment challenging [64].

Although numerous token reduction techniques [9, 49, 65] have been proposed and have achieved considerable success, they are primarily developed for single-turn VQA. Meanwhile, the more realistic MT-VQA setting, which involves multi-round conversational question answering, remains largely underexplored thus far. Compared to single-turn VQA, which focuses on answering one-shot questions and can greedily discard image tokens irrelevant to the current query, MT-VQA poses additional challenges due to its open-ended nature. In MT-VQA, future questions are entirely unpredictable, and their relevant regions may arise anywhere in the image, making existing token reduction methods difficult to apply directly.

Existing token reduction methods can be broadly categorized into prompt-dependent and prompt-agnostic approaches. Prompt-dependent methods, such as FastV [9], retain tokens that are highly relevant to only the first-turn question prompt. This strategy may inadvertently discard visual information that could be crucial for answering subsequent questions; for example, the first question might focus on the foreground while later questions may reference the background. In contrast, prompt-agnostic approaches like PruMerge [49] reduce tokens solely based on attention scores within the image sequence itself, which are technically applicable to multi-turn interactions. However, a critical limitation of existing prompt-agnostic methods lies in their reliance on heuristic reduction criteria derived from human priors, and the lack of theoretical guidance, often resulting in suboptimal performance.

*Equal Contribution. Email: y_w@zju.edu.cn, haofeizhang@zju.edu.cn

†Corresponding Author. Email: xinchao@nus.edu.sg

In response to this, we propose a learning-based prompt-agnostic token reduction approach, termed MetaCompress, which overcomes the drawbacks of heuristic designs. To achieve this, a key question is how the learning objective should be defined. By analyzing the reduction formats of current approaches, including both pruning and merging, we find that they can be unified by formulating the visual token reduction task as an optimization problem. The goal is to identify an optimal compression mapping of the input visual tokens, under conditions such as language conditioning in prompt-dependent approaches and image-only conditioning in prompt-agnostic approaches, so that the model’s responses exhibit minimal discrepancy after token reduction.

Based on this formulation, we first simplify the problem by learning an optimal compression matrix for each image and conduct a preliminary investigation into the guiding role of attention information, as commonly employed in previous methods. Surprisingly, our findings reveal that the tokens retained by the learned matrix do not exhibit an obvious relationship with the heuristic attention cues commonly used in prior methods, such as [CLS] token attention and prompt-token attention, further validating that heuristic reduction criteria are suboptimal.

Furthermore, to fully implement MetaCompress, a practical challenge arises from the need to generate multiple compression matrices, since actual image inputs can vary in resolution. And learning specific compression matrices for every possible resolution is not an especially elegant or practical solution. To address this, we ultimately design to learn a compression matrix generator compatible with dynamic resolutions, trained in a data-efficient paradigm. Extensive experiments on three MT-VQA benchmarks using five LVLM architectures demonstrate that MetaCompress outperforms existing token reduction methods while achieving high computational efficiency.

The contributions of our paper are summarized as follows:

- We first explore token reduction in the MT-VQA scenario, revealing that heuristic methods relying on visual token attention scores are suboptimal.
- We propose MetaCompress, a novel learning-based and prompt-agnostic token reduction method, overcoming the reliance on suboptimal heuristic reduction criteria.
- MetaCompress leverages a data-efficient training paradigm to learn the optimal compression mapping for visual sequences, demonstrating the effectiveness and efficiency through extensive experiments.

2. Related Work

2.1. Efficient Large Vision-Language Models

LVLMs. Transformers [57] have unified architectures across language [6, 16, 46] and vision [7, 17, 23, 26, 53], then CLIP [45] bridges both modalities through contrastive pre-

training, enabling zero-shot visual understanding. Based on this, LVLMs [2, 35–37, 58, 66, 70] integrate visual encoders with large language models to perform multimodal tasks such as captioning and VQA. LLaVA [35, 36] achieves image-to-text generation by feeding CLIP-encoded visual tokens and language tokens into an LLM *e.g.*, Llama [54, 55], but its fixed global resolution restricts fine-grained perception. Recent models such as LLaVA-NeXT [37] and InternLM-XComposer-2.5 [70] enhance visual understanding by incorporating multi-scale visual inputs that combine global and local tokens, but this substantially increases token counts, leading to heavy memory and computation overhead in multi-head attention and auto-regressive decoding, particularly on resource-constrained devices.

Model Quantization. To deploy LVLMs to low-memory devices such as mobile while preserving the model’s performance, a natural approach is to quantize the model and inference process into 4/8-bit [15, 19, 67] or even 1-bit [42]. Another line of work focuses on reducing the computational burden of MHA by employing efficient attention mechanisms [10, 12, 13, 32] or sparse attention [11, 59, 60]. However, quantization methods are limited by optional fine-tuning and hardware support, and more importantly, they do not solve the overall computational inefficiency caused by the increasing number of visual tokens.

Model Pruning. Model pruning [20, 40, 43, 71] and knowledge distillation [25, 48, 61] methods compress the given model to arbitrary size by removing redundant parameters or transferring knowledge from a large model to a smaller one. These approaches are generally effective at reducing model size and inference cost, but often require careful hyperparameter tuning and expensive retraining procedure.

2.2. Visual Token Reduction

Recent studies show that image representations contain substantial redundancy [14, 24], enabling feature reduction without significant performance loss [5, 8, 29]. This observation has motivated the development of token reduction techniques for LVLMs, which can be broadly categorized into prompt-dependent and prompt-agnostic approaches. Prompt-dependent methods, such as FastV [9], identify redundant tokens by measuring their attention to language prompts and remove them at specific layers. FitPrune [65] identifies redundant tokens by minimizing the divergence of attention distributions before and after pruning. IVTP [27] and TRIM [51] employ CLIP’s text encoder to guide token reduction. However, these methods are less applicable to general scenarios such as MT-VQA, as they require re-compression for each question. In contrast, prompt-agnostic methods rely solely on image sequences and are technically applicable to MT-VQA tasks. Nevertheless, existing approaches like LLaVA-PruMerge [49] overlook compatibility with modern LVLMs that incorporate multi-scale vision towers (*e.g.*,

LLaVA-NeXT). More importantly, these methods rely heavily on heuristic reduction criteria derived from human priors, which often lead to suboptimal performance. To address this, this paper introduces *a novel learning-based, prompt-agnostic token reduction method that avoids heuristic designs* (e.g., attention to [CLS] or other tokens as reduction guidance, which will be shown suboptimal later in this paper) and can integrate seamlessly with modern LVLMs.

3. Preliminaries

In this section, we first give a brief review of the inference process of LVLMs, particularly in the context of multi-turn dialogue scenarios. We then introduce the problem definition of visual sequence compression mapping.

3.1. Large Vision-Language Models

Given an input image I_{IMG} , LVLMs are required to generate a series of responses (R_1, \dots, R_t) to user’s prompts (P_1, \dots, P_t) . The image and the language context are tokenized separately by a vision tower $T_{\text{IMG}}(\cdot)$, e.g., vision Transformer (ViT) [17, 45] and a language tokenizer $T_{\text{TXT}}(\cdot)$, e.g., SentencePiece [31]. The tokenized image and language sequence are then embedded into a common space by a vision projector $V_{\text{IMG}}(\cdot)$ and an embedding layer $E_{\text{TXT}}(\cdot)$, producing X_{IMG} and X_{TXT} , respectively.

To fully capture detailed information, current prevalent LVLMs, such as LLaVA-NeXT and InternLM-XComposer-2.5, employ a ViT to encode images from both global and local views, generating multi-scale visual sequences. Despite enhancing the model’s capability to capture the details of the image, such an approach significantly increases the token number, severely impairing the inference efficiency due to the $O(n^2)$ computational and memory complexity of MHA [57].

To increase LLMs’ inference efficiency, KV cache methods [34, 62] are proposed for reusing intermediate attentions in the auto-regressive decoder. Specifically, for generating the i -th response token with query q_i , the original computation is to concatenate the previous queries for MHA layer

$$\text{MHA}(Q_i, K_i, V_i) = \sigma \left(\frac{Q_i K_i^\top}{\sqrt{d_k}} \right) V_i, \quad (1)$$

where $\{Q, K, V\}_i = (\{Q, K, V\}_{i-1} | \{q, k, v\}_i)$ are the concatenated inputs, and σ denotes the row-wise Softmax operation. To decrease the computation complexity, KV caches store the intermediate key-value pairs $\{K, V\}_{i-1}$ and only compute the attention for q_i :

$$\text{MHA}(q_i, k_i, v_i) = \sigma \left(\frac{q_i (K_{i-1} | k_n)^\top}{\sqrt{d_k}} \right) (V_{i-1} | v_i), \quad (2)$$

which significantly reduce the computation burden. Such techniques can be seamlessly integrated with the MT-VQA setting, where the caches are reused across multiple turns.

3.2. Visual Token Reduction

However, the aforementioned cache mechanism is still insufficient to address the memory and computation overhead caused by the large number of image tokens, resulting in an $O(n^2)$ cost for generating the first token and an $O(nT)$ cost for producing T tokens during multi-turn dialogues.

To alleviate the issue, token reduction methods are proposed to compress the image sequence. For simplicity, we only consider reducing image tokens right before feeding into the LLM, e.g., LLama [54], which can be formulated as

$$\tilde{X}_{\text{IMG}} = \mathcal{P}_{\text{reduce}}(X_{\text{IMG}} | I_{\text{IMG}}, I_{\text{TXT}}), \quad (3)$$

where guiding information is extracted from the input image I_{IMG} and the language context I_{TXT} . Depending on whether to rely on the prompt I_{TXT} , token reduction methods can be categorized into prompt-dependent and prompt-agnostic methods. However, in real-world applications, LVLMs are often required to respond to multiple prompts. Prompt-dependent methods, however, tend to bias toward the initial query and discard information beneficial for subsequent turns, leading to suboptimal performance in multi-turn dialogue scenarios. Furthermore, many existing methods require the intermediate attention matrices in MHA layers to guide token reduction, whereas modern LVLMs commonly employ FlashAttention [12, 13] or Memory-Efficient Attention [32], which do not support returning them.

To further investigate the optimal token reduction strategy, we first unify the token pruning and merging methods by formulating the reduction process as a linear projection to the input X_{IMG} :

$$\tilde{X}_{\text{IMG}} = P X_{\text{IMG}}, \quad (4)$$

where $P \in \mathbb{R}_+^{m \times n}$ ($m \ll n$) is the sparse compression matrix. In Section 4, we set P as a learnable matrix for each image. By optimizing P in a data-driven manner, we first explore the relationship between the retained tokens and the attention weights employed by heuristics-designed methods. Then in Section 5, we present a novel token reduction method that does not rely on the intermediate attention matrices, while can be seamlessly integrated with modern LVLMs.

4. Which Tokens to Keep?

To objectively analyze the optimal token reduction scheme without relying on hand-crafted designs, we start by looking at a simpler case: *Given an input image I_{IMG} and a conversation context I_{TXT} , find the optimal compression matrix P^* as defined in Equation (4) so that the response discrepancy between the LLM using the compressed and uncompressed visual sequence is minimized.*

To achieve this, let $P_{\text{raw}} \in \mathbb{R}^{m \times n}$ be the trainable reduction parameters, with each element independently drawn from Gaussian distribution $\mathcal{N}(0, \sigma_{\text{raw}}^2)$. We normalize P_{raw}

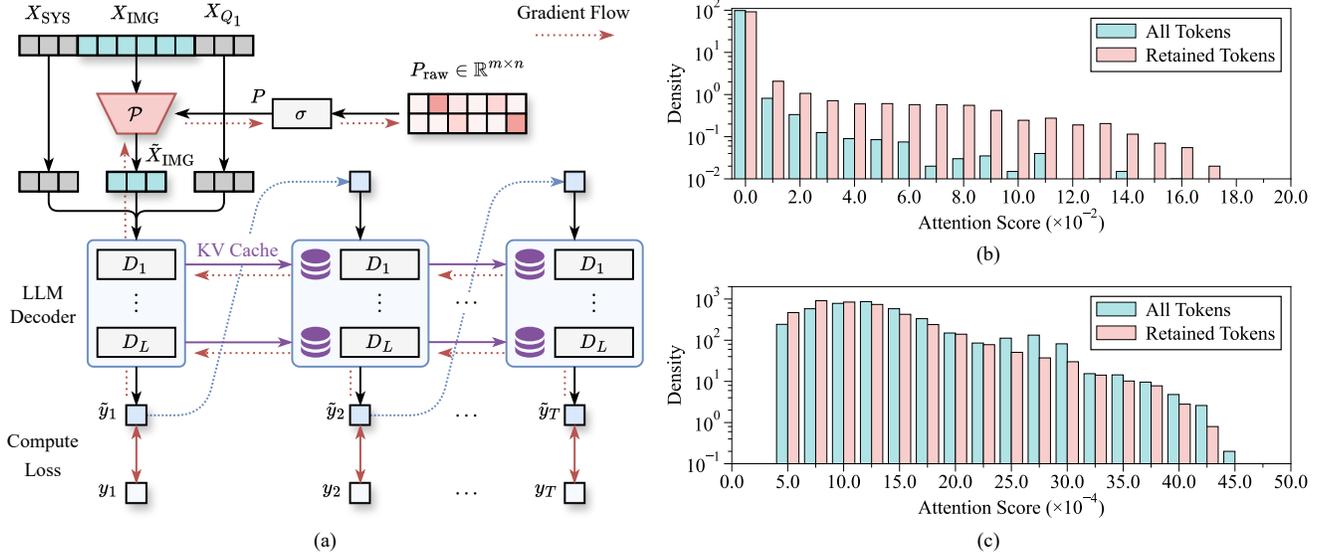


Figure 1. (a) Overall pipeline of the compression projection training process. (b) Attention distribution over the [CLS] token for retained and all visual tokens. The image tokens are extracted from the last layer of the vision tower of LLaVA-1.5-13b running on VQA-v2 dataset. (c) Attention distribution over the prompt tokens for retained and all visual tokens. The attention scores are averaged to prompt tokens extracted from the first layer of the LLM decoder.

with row-wise Softmax to obtain the compression matrix $P = \sigma(P_{raw})$. Let $p(y|X_{IMG}, X_{TXT})$ denotes the original prediction distribution T tokens $y = (y_1, \dots, y_T)$. Then we force the LLM to generate T tokens $p(\tilde{y}|\tilde{X}_{IMG}, X_{TXT})$ using the compressed visual sequence \tilde{X}_{IMG} . Figure 1a illustrates the overall training pipeline, where P_{raw} is trained to minimize the KL divergence between the two response distributions $\mathcal{L}_{pred} = D_{KL}(p(y)||p(\tilde{y}))$ and the distribution entropy $\mathcal{L}_{entropy} = \frac{1}{m} \sum_{i=0}^m \mathcal{H}(P_{i,:})$.

The training objective is formulated as

$$P^* = \arg \min_{P_{raw}} \mathcal{L}_{pred} + \alpha \mathcal{L}_{entropy}. \quad (5)$$

The training algorithm and detailed implementation are provided in Section 8.

Figure 1b and 1c visualize the attention distribution over the [CLS] and prompt tokens, respectively. Despite that some tokens with high attention to the [CLS] token are retained (accounting for approximately 1.71% of the total retained tokens), the vast majority of the retained tokens are unrelated to their attention scores, especially with regards to the attention to the language prompts. More results are in Section 6.6, drawing to the same conclusion. This observation suggests that using attention scores as guidance for token reduction is suboptimal in the MT-VQA scenario, which explains the experimental results in Section 6.2, where token pruning methods such as FastV perform worse than uniform or even random pruning. Therefore, it is essential to explore a novel token reduction approach that does not rely on heuristic metrics such as attention scores, while being seamlessly compatible with modern LVLMs.

5. Method

Results from Section 4 inspire us to construct the compression matrix in a data-driven manner. To this end, we propose MetaCompress, a lightweight module learning the compression matrix P conditioned only on the input image for MT-VQA scenarios. Section 5.1 details the MetaCompress module, Section 5.2 provides theoretical analysis, and Section 5.3 presents the optimization objective and training algorithm.

5.1. MetaCompress

Our goal is to learn a compression matrix generator \mathcal{P}_{meta} in a data-driven manner, so that the overall prediction discrepancy on the given dataset $\mathcal{D} = \{(I_{IMG}^{(i)}, I_{TXT}^{(i)})\}_{i=1}^N$ is minimized. To this end, we propose a lightweight meta generator \mathcal{P}_{meta} which computes a compression matrix $P = \mathcal{P}_{meta}(X_{IMG})$ for each input image I_{IMG} , independent of the prompt. One major challenge is that \mathcal{P}_{meta} is required to generate the compression matrix P , whose shape can adapt to the varying length of X_{IMG} , thereby accommodating multiple resolution scales for LVLMs such as LLaVA-NeXT and InternLM-XComposer-2.5.

Figure 2 shows the overall architecture of \mathcal{P}_{meta} , which consists of a position embedding layer, a query down-sample projection \tilde{D}_q , a key projection D_k , and a weighted inner product layer. The core idea is to compute the inner product between the spatially down-sampled queries $\tilde{X}_q \in \mathbb{R}^{m \times d_c}$ and keys $X_k \in \mathbb{R}^{n \times d_c}$ to get the compression matrix $P \in \mathbb{R}^{m \times n}$. Here, queries

$$\tilde{X}_q = \tilde{D}_q(X_{IMG} + E_{pos}) = \text{Pool}(X_{IMG} + E_{pos}|k, s)W_q \quad (6)$$

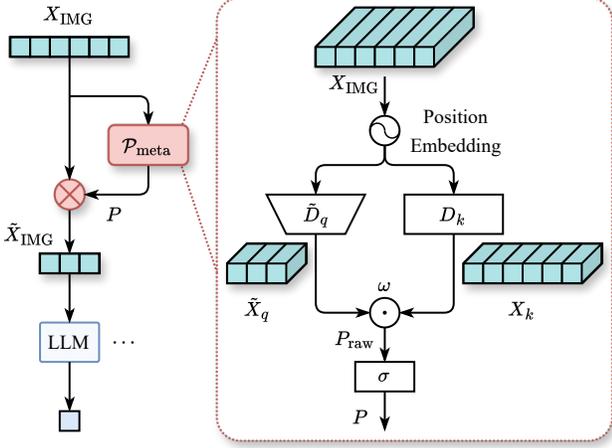


Figure 2. Illustration of our proposed MetaCompress, where module $\mathcal{P}_{\text{meta}}$ generate the compression projection P solely according to the image sequence X_{IMG} .

are down-sampled from the image sequence encoded with absolute position embeddings E_{pos} by average pooling $\text{Pool}(\cdot|k, s)$ with kernel size k and stride s^1 , and keys

$$X_k = D_k(X_{\text{IMG}} + E_{\text{pos}}) = (X_{\text{IMG}} + E_{\text{pos}})W_k \quad (7)$$

are linearly projected from X_{IMG} for computational efficiency (by setting $d_c \ll d$). Finally, the computation of compression matrix P is formulated as:

$$P = \sigma \left(\frac{\tilde{X}_q \text{diag}(\omega) X_k^\top}{\sqrt{d_c}} \right), \quad (8)$$

where diagonal matrix $\omega \in \mathbb{R}^{d_c}$ is learnable.

Regarding the module placement, following the setting of LLaVA-PruMerge [49], we apply our reduction module only before the LLM decoder, although MetaCompress can in principle be inserted at any layer. This placement reduces the additional MHA computation incurred in earlier LLM layers compared with inserting it at intermediate layers, which is particularly beneficial for long visual inputs such as videos. Moreover, our work focuses on developing a learning-based reduction method rather than a full-stage compression strategy across both the vision tower and the LLM, as explored in IVTP [27]. Such full-stage optimization requires costly pretraining and instruction-tuning, which we leave as future work under our lightweight framework.

For the module design, since our primary goal is to reduce the inference burden of LVLMs, we intentionally avoid constructing complex reduction modules, such as those required for auto-regressive generation, as they would significantly increase latency and reduce computational efficiency. As

¹Section 8.2 provides the detailed implementation for down-sampling X_{IMG} to arbitrary length m .

Algorithm 1 Training algorithm for MetaCompress.

Require: $E_{\text{TXT}}(\cdot)$: the language encoder; $V_{\text{IMG}}(\cdot)$: the image encoder; $\text{LLM}(\cdot, \cdot)$: the vision-language decoder; $\mathcal{P}_{\text{meta}}(\cdot|\Theta)$: the proposed MetaCompress module with learnable parameters Θ .

- 1: **for** $(I_{\text{IMG}}, I_{\text{TXT}}) \in \mathcal{D}_{\text{train}}$ **do**
 - 2: $X_{\text{TXT}} \leftarrow E_{\text{TXT}}(I_{\text{TXT}})$
 - 3: $X_{\text{IMG}} \leftarrow V_{\text{IMG}}(I_{\text{IMG}})$
 - 4: $\tilde{X}_{\text{IMG}} \leftarrow \mathcal{P}_{\text{meta}}(X_{\text{IMG}}|\Theta)$ # with gradients
 - 5: $\mathbf{y} \leftarrow \text{LLM}(X_{\text{TXT}}, X_{\text{IMG}})$
 - 6: $\tilde{\mathbf{y}} \leftarrow \text{LLM}(X_{\text{TXT}}, \tilde{X}_{\text{IMG}})$ # with gradients
 - 7: Compute the final loss and gradient ∇_{Θ} w.r.t. Θ .
 - 8: Update Θ with SGD optimizer.
 - 9: **end for**
-

the first learning-based token reduction framework, there are currently few non-learning approaches available for direct comparison. Nevertheless, we further discuss the relationship between our method and other data-driven approaches for efficient model inference in Section 12.

5.2. Theoretical Analysis

Now we provide a theoretical analysis of MetaCompress to explain the design motivation and further introduce the optimization objectives and constraints. To begin with, we expand Equation (8) as

$$P_{\text{raw}} = \text{Pool}(X|k, s)W_q \text{diag}(\omega)W_k^\top X^\top. \quad (9)$$

Further, suppose all elements in W are drawn independently from a Gaussian distribution $\mathcal{N}(0, \sigma_c^2)$ and with a specific initialization (i.e., $W_q = W_k$), MetaCompress will initially behave as a weighted pooling of the input image sequence (controlled by the kernel size k), as we prove in Section 9. Moreover, the meta generator will learn, in a data-driven manner, how to select and merge the visual tokens to minimize the prediction discrepancy. Since we do not rely on any annotation for the compression matrix, the low-rank positive semi-definite form presented in Equation (9) provides a good starting point and facilitates gradient decent optimization.

5.3. Training MetaCompress

Similar to the training objective introduced in Section 4, we train MetaCompress by minimizing the prediction discrepancy $\mathcal{L}_{\text{pred}}$ with additional sparsity regularization $\mathcal{L}_{\text{entropy}}$. However, due to the lack of ground-truth compression matrix, the generated compression matrix P tends to collapse to trivial solutions where the compressed tokens all derive from the same input source. To avoid this, we add a collapse regularization term $\mathcal{L}_{\text{collapse}} = \max_j \sum_{i=1}^m P_{i,j}$. Therefore, the final optimization objective is

$$\mathcal{L} = \mathcal{L}_{\text{pred}} + \alpha_{\text{entropy}} \mathcal{L}_{\text{entropy}} + \alpha_{\text{collapse}} \mathcal{L}_{\text{collapse}}, \quad (10)$$

where α_{entropy} and α_{collapse} are hyperparameters. Algorithm 1 delineates the training procedure.

Table 1. The comparison of visual token reduction methods on three MT-VQA benchmarks with the reduction rate of 90%. The best and the second-best results are highlighted in bold and underline, respectively.

Model	Method	MT-VQA-v2				MT-GQA				ConvBench			
		Acc_1	Acc_2	Acc_3	Avg	Acc_1	Acc_2	Acc_3	Avg	S_1	S_2	S_3	Avg
LLaVA-1.5-7b	Base	76.72	77.51	77.30	77.18	61.76	64.07	65.35	63.73	4.33	5.72	5.55	5.20
	Random	66.36	66.94	66.68	66.66	54.60	57.07	59.31	56.99	3.73	3.99	4.08	3.93
	Sample	67.11	67.52	67.63	67.42	55.06	57.89	59.74	<u>57.56</u>	3.64	4.85	3.81	<u>4.10</u>
	FastV	45.98	48.56	49.65	48.06	40.98	46.71	49.30	<u>45.66</u>	1.56	1.39	3.12	<u>2.02</u>
	PruMerge	69.03	69.93	69.73	<u>69.56</u>	55.26	57.23	60.13	57.54	4.51	3.47	3.47	3.82
	Ours	70.27	70.31	71.36	70.65	55.95	58.71	60.64	58.43	4.33	3.99	4.16	4.16
LLaVA-1.5-13b	Base	78.35	79.47	78.92	78.91	62.47	65.21	67.22	64.97	4.33	7.11	5.72	5.72
	Random	67.26	68.05	67.63	67.65	54.63	57.87	60.23	57.58	3.56	5.55	4.42	4.51
	Sample	68.12	68.82	68.47	68.47	55.41	58.53	60.26	58.07	4.16	5.03	4.33	4.51
	FastV	55.36	56.80	57.08	56.41	49.08	53.34	56.14	52.85	2.19	3.47	4.20	3.29
	PruMerge	70.18	71.16	70.70	<u>70.68</u>	55.70	57.94	60.70	<u>58.11</u>	4.51	3.99	5.55	<u>4.68</u>
	Ours	72.70	73.24	72.88	72.94	57.02	59.26	62.16	59.48	4.68	4.51	6.41	5.20
LLaVA-NeXT-7b	Base	80.20	80.86	80.71	80.59	63.83	66.68	67.94	66.15	7.95	11.46	7.58	9.00
	Random	70.65	72.26	72.44	71.78	58.60	61.04	63.00	60.88	5.81	6.59	4.42	<u>5.61</u>
	Sample	70.88	72.32	72.35	<u>71.85</u>	58.46	61.39	63.24	<u>61.03</u>	3.81	7.28	5.72	5.60
	FastV	57.09	59.00	59.27	58.45	46.42	50.55	53.95	50.31	0.00	1.85	1.85	1.23
	Ours	73.83	75.24	75.18	75.18	59.43	63.49	65.19	62.70	4.16	8.67	9.01	7.28
	LLaVA-NeXT-13b	Base	81.02	82.32	81.64	81.66	65.45	67.32	69.12	67.30	12.48	13.17	7.97
Random		71.86	73.44	73.22	72.84	59.61	61.86	63.43	61.63	7.20	9.19	6.33	7.57
Sample		71.97	73.84	73.43	<u>73.08</u>	59.69	62.28	63.88	<u>61.95</u>	6.07	10.92	6.07	<u>7.69</u>
FastV		57.07	59.09	59.14	58.43	47.23	51.34	53.36	50.64	5.17	5.17	1.72	4.02
Ours		74.62	75.73	75.42	75.26	60.78	63.41	65.16	63.12	6.93	11.27	6.76	8.32
XComposer-2.5-7b		Base	78.80	81.12	81.24	80.39	60.21	63.38	65.01	62.87	12.48	7.00	7.97
	Random	67.88	70.20	70.49	69.52	52.52	57.01	60.67	56.73	11.35	9.88	7.28	9.50
	Sample	68.46	70.92	70.78	70.05	52.94	57.20	59.69	56.61	12.13	9.53	7.63	<u>9.76</u>
	FastV	72.65	75.02	75.02	<u>74.23</u>	54.84	57.33	58.83	<u>57.00</u>	4.17	4.17	0.00	<u>2.78</u>
	Ours	73.91	76.68	76.70	75.76	55.99	58.49	61.55	58.68	12.31	9.71	7.63	9.88

6. Experiments

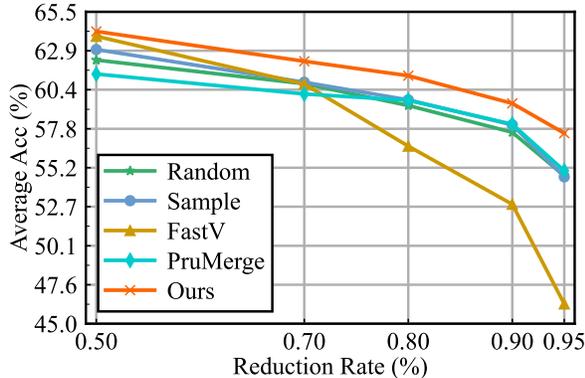
6.1. Implementation

Datasets. We evaluate our method on three MT-VQA benchmarks: MT-VQA-v2, MT-GQA, and ConvBench². MT-VQA-v2 is constructed based on the validation set of VQA-v2 [3, 68] with 25k three-turn image-dialogue pairs. Similarly, MT-GQA is constructed from the testdev-balanced set of GQA [28] with 4061 three-turn dialogues. ConvBench [38] is a native multi-turn conversation evaluation benchmark with 577 conversations that adopts a three-level multimodal capability hierarchy. Instead of training on the entire dataset, which is time-consuming, we only train MetaCompress on a small subset (about 20k items) drawn from the training-balanced split of MT-GQA and the training set of MT-VQA-v2. We utilize the pre-trained weights on MT-VQA-v2 to evaluate on ConvBench.

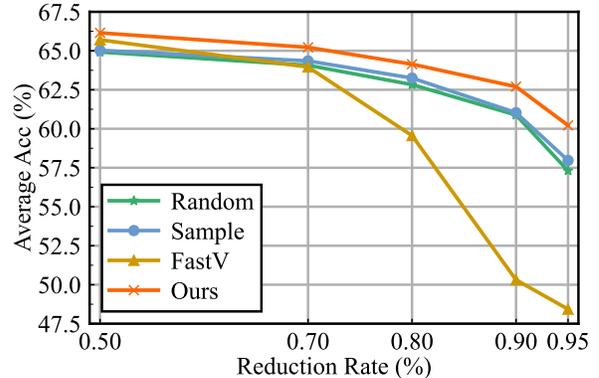
²As ConvBench relies on GPT-3.5-turbo’s commercial API for evaluation, we replace it with the recently released open-source LLM, Llama-3.1-8B-Instruct [18].

LVLMS. To evaluate the generalizability of our method, we choose five different LVLMS: LLaVA-1.5-7b/13b [35], LLaVA-NeXT-7b/13b [37], and InternLM-XComposer-2.5-7b [70]. Of these models, LLaVA-1.5 employs a single-scale vision tower with a fixed visual sequence length, while the others adopt multi-scale perception, resulting in variable visual sequence lengths, which brings further challenges to the token reduction method.

Training Details and Selection of Hyperparameters. We implement our method with PyTorch [44] and optimize the proposed MetaCompress with SGD [47] with a learning rate of 10^{-3} . Gradient clipping is adopted with a maximum value of 10^{-2} . We train all the settings for 2 epochs with a batch size of 36 on four commercial NVIDIA RTX A6000 GPUs. The training of LLaVA-NeXT-7B with a 90% reduction rate takes approximately 30 GPU hours, which corresponds to about only 9 hours on a 4-GPU machine. We initialize $W_q = W_k$ and drawn from Gaussian distribution $\mathcal{N}(0, \frac{1}{\sqrt{d_c}}^2)$; ω is set to all ones; $\alpha_{\text{entropy}} = \alpha_{\text{collapse}} = 1$ as the default setting.



(a) LLaVA-1.5-13b



(b) LLaVA-NeXT-7b

Figure 3. Comparison of average accuracy on MT-GQA with reduction rate from 50% to 95%.

6.2. Comparison Results

We evaluate the proposed MetaCompress and comparison baselines with the following settings:

- **Base:** The base LLaVA evaluated directly on the MT-VQA benchmarks without token reduction.
- **Random:** Randomly prune visual tokens before the first layer of the LLM decoder. We report the average performance over 3 random seeds.
- **Sample:** Perform equidistant down-sampling on the visual sequence before the LLM decoder.
- **FastV:** Our implementation of FastV [17] for multi-scale vision tower. The guidance attention weights are extracted from the first layer of the LLM decoder and visual tokens are pruned at the second layer.
- **PruMerge:** Perform LLaVA-PruMerge [35] only for LLaVA-1.5, as it is not compatible with the multi-scale visual tower.
- **Ours:** Perform our proposed MetaCompress before the LLM decoder.

The comparison results with reduction rate 90% and 70% are shown in Table 1 and Table 4, where we compare the accuracy of each turn conversation and the overall accuracy on three MT-VQA benchmarks. *It is noticeable that the proposed MetaCompress consistently outperforms the baseline methods.* While not trained on ConvBench, our method still surpasses the baseline methods by a large margin, demonstrating the transferability of MetaCompress. For LLaVA-1.5, experimental results show that LLaVA-PruMerge which is designed specifically for it performs slightly better than Sample, but still lags behind our approach. On the other hand, FastV performs significantly worse than both the Sample and even the Random methods. This further supports our findings in Section 4, where we have revealed that using attention as guidance for compression results in a loss of critical tokens. Although FastV shows some improvement for XComposer-2.5-7b, it still performs poorly on ConvBench.

Table 2. Efficiency comparison of different token reduction methods. The time to first token (TTFT, ms), end-to-end generation time (E2ET, ms), GPU memory usage (Mem. GB), and TFLOPs are reported on MT-GQA dataset with a reduction rate of 90%.

Model	Setting	TTFT	E2ET	Mem.	TFLOPs
LLaVA-1.5-7b	Base	232 (± 5.8)	676 (± 8.0)	26.9	71.4
	Random	98.2 (± 5.7)	487 (± 4.6)	26.2	13.3
	Sample	96.9 (± 5.8)	482 (± 4.9)	26.2	13.3
	FastV	102 (± 4.52)	528 (± 6.0)	26.3	13.5
	PruMerge	107 (± 5.23)	509 (± 4.5)	26.2	13.3
	Ours	97.8 (± 5.41)	480 (± 5.1)	26.1	13.3
LLaVA-NeXT-7b	Base	484 (± 4.7)	830 (± 13.5)	16.7	95.3
	Random	174 (± 3.4)	481 (± 4.5)	14.8	12.7
	Sample	176 (± 3.2)	484 (± 5.3)	14.8	12.7
	FastV	219 (± 5.0)	529 (± 5.33)	19.2	12.9
	Ours	174 (± 6.1)	501 (± 4.8)	14.9	12.7

Figure 3 illustrates the performance curve of different methods with varying reduction rates from 50% to 95%. It is clear that our method consistently outperforms the baselines across different reduction rates, while FastV performs better for low reduction rate and LLaVA-PruMerge performs better for high reduction rate.

6.3. Efficiency Results

Table 2 compares the inference efficiency of different token reduction methods. As we can observe that our method achieves compatible efficiency with the ‘Sample’ setting, which is the most efficient baseline thanks to the explicit low-ranking mechanism as described in Equation (9).

6.4. Transfer Results

Beyond the transfer results on ConvBench in Table 1, we further evaluate the transfer capability of MetaCompress through comprehensive cross-dataset validation. Specifically, we perform transfer learning experiments between MT-GQA and MT-VQA-V2, with the results summarized in Table 7. This table reports the average accuracy under a

Table 3. Ablation study of training MetaCompress for LLaVA-NeXT-7b using different loss terms on MT-GQA. Gradient clipping is only applied for the ‘ $\mathcal{L}_{\text{collapse}} + \text{Grad Clip}$ ’ setting.

Settings				MT-GQA
$\mathcal{L}_{\text{pred}}$	$\mathcal{L}_{\text{entropy}}$	$\mathcal{L}_{\text{collapse}}$	$\mathcal{L}_{\text{collapse}} + \text{Grad Clip}$	Avg
✓	✗	✗	✗	61.98
✓	✓	✗	✗	62.42
✓	✗	✓	✗	56.34
✓	✗	✗	✓	62.13
✓	✓	✗	✓	62.70

90% token reduction rate for both directions of transfer, from MT-GQA to MT-VQA-V2 and vice versa. These results indicate that MetaCompress is not heavily dependent on a specific training dataset, demonstrating robust generalization. We also conduct transfer experiments on the video question answering task, as reported in Table 8 of Section 10.4.

6.5. Ablation Study

As delineated in Section 5.3, we utilize three optimization objectives to train the proposed method. To investigate the effectiveness of each objective, we conduct an ablation study by removing one of the objectives at a time. The results in Table 3 (with additional results for various LVLMs in Table 6) demonstrate that each objective contributes positively to the overall performance. In particular, training utilizing the $\mathcal{L}_{\text{collapse}}$ alone leads to divergence because of the relatively high penalty on the collapse objective, especially when the reduction rate is small (less than 70%). To tackle this, we introduce gradient clipping to stabilize the training process.

Besides, we also investigate the sensitivity of the hyperparameters α_{entropy} and α_{collapse} when training MetaCompress. Figure 4 shows the performance curves for different weight settings, demonstrating that the performance remains relatively stable (within a 0.5 percentage point variation).

6.6. Visualization

Figure 5 visualizes the attention distribution for LLaVA-NeXT-7b, similar to Figure 1. As directly computing the attention to [CLS] token is not feasible for multi-scale vision towers, we compute FastV’s style image token importance instead. Nevertheless, we observe that only a small number of tokens with high attention are retained, which is consistent with the conclusion in Section 4 and further demonstrates that using token attention to guide reduction is suboptimal.

7. Conclusion and Outlook

This paper proposes a novel token reduction approach for multi-turn VQA scenarios. To this end, we first unify token pruning and merging under the framework of compression projection to visual sequences and explore the optimal compression mapping for a single image. Preliminary results reveal that existing methods guided by attention are subopti-

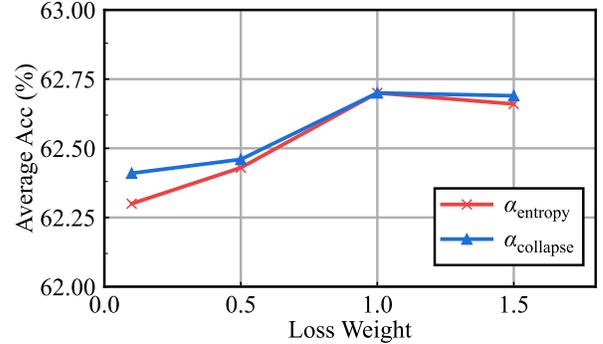


Figure 4. Sensitivity analysis in training MetaCompress for LLaVA-NeXT-7b with different weights α_{entropy} and α_{collapse} on MT-GQA.

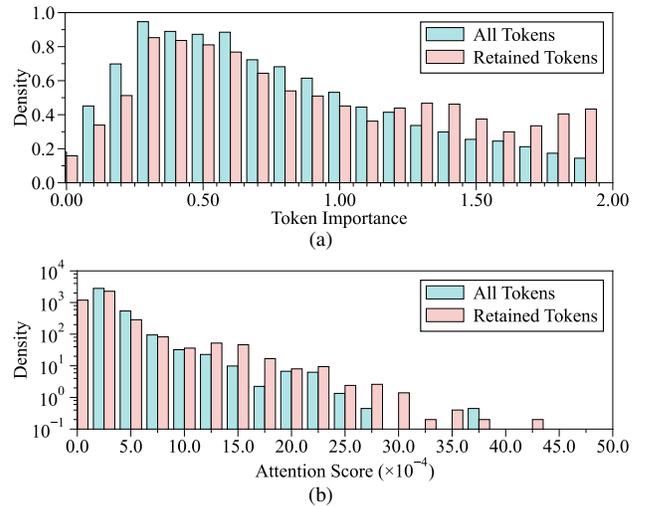


Figure 5. (a) Token importance distribution. (b) Attention distribution over prompt tokens. Image tokens are extracted from the last layer of the vision tower of LLaVA-NeXT-7b on VQA-v2 dataset.

mal, as a large number of retained tokens do not correspond to the highest attention scores. This motivates us to further explore the construction of an optimal compression mapping for the entire dataset. To achieve this, we propose MetaCompress, a meta generator conditioned solely on the visual sequence, and optimized in a data-driven manner. Extensive experiments demonstrate the efficiency and effectiveness of our method. In future work, we will explore the token reduction strategy for all LLM layers without a hand-crafted design, and investigate the transferability of our method to more challenging tasks, such as video understanding.

Acknowledgments

This work is funded by National Natural Science Foundation of China (62576305), the Alibaba Group through Alibaba Innovative Research Program, and the National Research Foundation, Singapore, under its Medium Sized Center for Advanced Robotics Technology Innovation.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 1
- [2] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, et al. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 2
- [3] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, pages 2425–2433, 2015. 6
- [4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1
- [5] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your vit but faster. In *ICLR*, 2023. 2, 1
- [6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, et al. Language models are few-shot learners. In *NeurIPS*, pages 1877–1901, 2020. 2
- [7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 2
- [8] Kaixuan Chen, Jie Song, Shunyu Liu, Na Yu, Zunlei Feng, and Mingli Song. Distribution knowledge embedding for graph pooling. *IEEE Transactions on Knowledge and Data Engineering*, 35:7898–7908, 2021. 2
- [9] 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. *arXiv preprint arXiv:2403.06764*, 2024. 1, 2
- [10] Zigeng Chen, Gongfan Fang, Xinyin Ma, Ruonan Yu, and Xinchao Wang. dparallel: Learnable parallel decoding for dllms. In *International Conference on Learning Representations*, 2026. 2
- [11] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with sparse transformers. *ArXiv*, abs/1904.10509, 2019. 2
- [12] Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *ICLR*, 2024. 2, 3
- [13] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022. 1, 2, 3
- [14] Huiqi Deng, Qihan Ren, Hao Zhang, and Quanshi Zhang. Discovering and explaining the representation bottleneck of DNNs. In *ICLR*, 2022. 2
- [15] Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *NeurIPS*, pages 30318–30332, 2022. 2
- [16] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186, Minneapolis, Minnesota, 2019. 2
- [17] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2, 3, 7
- [18] 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. 6
- [19] Kazuki Egashira, Mark Vero, Robin Staab, Jingxuan He, and Martin Vechev. Exploiting LLM quantization. *arXiv preprint arXiv:2405.18137*, 2024. 2
- [20] Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. Depgraph: Towards any structural pruning. In *CVPR*, pages 16091–16101, 2023. 2
- [21] Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, et al. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118, 2025. 2
- [22] Benjamin Graham. Fractional max-pooling. *arXiv preprint arXiv:1412.6071*, 2014. 1
- [23] Le Han, Kaixuan Chen, Minchen Ye, and Nenggan Zheng. Hi-motion: Hierarchical intention guided conditional motion synthesis. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 9628–9637, 2025. 2
- [24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *CVPR*, pages 16000–16009, 2022. 2
- [25] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531, 2015. 2
- [26] Chaoqun Hong, Liang Chen, Yuxin Liang, and Zhiqiang Zeng. Stacked capsule graph autoencoders for geometry-aware 3d head pose estimation. *Computer Vision and Image Understanding*, 208-209:103224, 2021. 2
- [27] Kai Huang, Hao Zou, Ye Xi, BoChen Wang, Zhen Xie, and Liang Yu. IVTP: Instruction-guided visual token pruning for large vision-language models. In *ECCV*, pages 214–230, Cham, 2025. Springer Nature Switzerland. 2, 5
- [28] Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *CVPR*, pages 6700–6709, 2019. 6
- [29] Cao Jianjian, Ye Peng, Li Shengze, Yu Chong, Tang Yansong, Lu Jiwen, and Chen Tao. MADTP: Multimodal alignment-guided dynamic token pruning for accelerating vision-language transformer. *CVPR*, 2024. 2
- [30] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020. 1

- [31] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *EMNLP*, pages 66–71, Brussels, Belgium, 2018. Association for Computational Linguistics. 3
- [32] Benjamin Lefaudeux, Francisco Massa, Diana Liskovich, Wenhan Xiong, Vittorio Caggiano, Sean Naren, Min Xu, Jieru Hu, Marta Tintore, Susan Zhang, Patrick Labatut, Daniel Haziza, Luca Wehrstedt, Jeremy Reizenstein, and Grigory Sizov. xFormers: A modular and hackable transformer modelling library. <https://github.com/facebookresearch/xformers>, 2022. 2, 3
- [33] Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm. *International Journal of Computer Vision*, pages 1–19, 2025. 3
- [34] Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Hafari, and Bohan Zhuang. MiniCache: KV cache compression in depth dimension for large language models. *arXiv preprint arXiv:2405.14366*, 2024. 3
- [35] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning, 2023. 1, 2, 6, 7
- [36] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *NeurIPS*, 2023. 2
- [37] Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge, 2024. 1, 2, 6
- [38] Shuo Liu, Kaining Ying, Hao Zhang, Yue Yang, Yuqi Lin, Tianle Zhang, Chuanhao Li, Yu Qiao, Ping Luo, Wenqi Shao, et al. Convbench: A multi-turn conversation evaluation benchmark with hierarchical capability for large vision-language models. *arXiv preprint arXiv:2403.20194*, 2024. 6
- [39] Songhua Liu, Weihao Yu, Zhenxiong Tan, and Xinchao Wang. Linfusion: 1 gpu, 1 minute, 16k image. *arXiv preprint arXiv:2409.02097*, 2024. 3
- [40] Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, K. Cheng, and Jian Sun. Metapruning: Meta learning for automatic neural network channel pruning. *ICCV*, pages 3295–3304, 2019. 2
- [41] Zhijian Liu, Ligeng Zhu, Baifeng Shi, Zhuoyang Zhang, Yuming Lou, Shang Yang, Haocheng Xi, Shiyi Cao, Yuxian Gu, Dacheng Li, et al. Nvlla: Efficient frontier visual language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4122–4134, 2025. 3
- [42] Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 2024. 2
- [43] Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-pruner: On the structural pruning of large language models. In *NeurIPS*, 2023. 2, 3
- [44] 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. *NeurIPS*, 32, 2019. 6
- [45] 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 *ICML*, pages 8748–8763. PMLR, 2021. 2, 3
- [46] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67, 2020. 2
- [47] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 6
- [48] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019. 2
- [49] 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, 5
- [50] Zhuoran Shen, Mingyuan Zhang, Haiyu Zhao, Shuai Yi, and Hongsheng Li. Efficient attention: Attention with linear complexities. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 3531–3539, 2021. 3
- [51] Dingjie Song, Wenjun Wang, Shunian Chen, Xidong Wang, Michael Guan, and Benyou Wang. Less is more: A simple yet effective token reduction method for efficient multi-modal llms. *arXiv preprint arXiv:2409.10994*, 2024. 2
- [52] Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. A simple and effective pruning approach for large language models. *arXiv preprint arXiv:2306.11695*, 2023. 3
- [53] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 2
- [54] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 2, 3
- [55] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruiti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 2
- [56] Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokula Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, et al. Fastvlm: Efficient vision encoding for vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19769–19780, 2025. 3
- [57] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 2, 3
- [58] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. Git: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022. 2

- [59] Zeqing Wang, Gongfan Fang, Xinyin Ma, Xingyi Yang, and Xinchao Wang. Sparsed: Sparse attention for diffusion language models. In *International Conference on Learning Representations*, 2026. [2](#)
- [60] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *ICLR*, 2024. [2](#)
- [61] Chuanpeng Yang, Yao Zhu, Wang Lu, Yidong Wang, Qian Chen, Chenlong Gao, Bingjie Yan, and Yiqiang Chen. Survey on knowledge distillation for large language models: Methods, evaluation, and application. *ACM Trans. Intell. Syst. Technol.*, 2024. [2](#)
- [62] Dongjie Yang, XiaoDong Han, Yan Gao, Yao Hu, Shilin Zhang, and Hai Zhao. Pyramidinfer: Pyramid KV cache compression for high-throughput llm inference. *arXiv preprint arXiv:2405.12532*, 2024. [3](#)
- [63] Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. Visionzip: Longer is better but not necessary in vision language models. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19792–19802, 2024. [2](#)
- [64] Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, et al. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*, 2024. [1](#)
- [65] Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. *arXiv preprint arXiv:2409.10197*, 2024. [1](#), [2](#)
- [66] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. CoCa: Contrastive captioners are image-text foundation models. *Transactions on Machine Learning Research*, 2022. [2](#)
- [67] Zhihang Yuan, Yuzhang Shang, and Zhen Dong. PB-LLM: Partially binarized large language models. In *ICLR*, 2024. [2](#)
- [68] Peng Zhang, Yash Goyal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Yin and yang: Balancing and answering binary visual questions. In *CVPR*, pages 5014–5022, 2016. [6](#)
- [69] Pan Zhang, Xiaoyi Dong, Bin Wang, Yuhang Cao, Chao Xu, et al. InternLM-XComposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*, 2023. [1](#)
- [70] Pan Zhang, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Rui Qian, et al. InternLM-XComposer-2.5: A versatile large vision language model supporting long-contextual input and output. *arXiv preprint arXiv:2407.03320*, 2024. [1](#), [2](#), [6](#)
- [71] Mingjian Zhu, Kai Han, Yehui Tang, and Yunhe Wang. Visual transformer pruning. *ArXiv*, abs/2104.08500, 2021. [2](#)

Rethinking Token Reduction for Large Vision-Language Models

Supplementary Material

8. Implementation Details

8.1. Fixed Compression Matrix

The training algorithm for P_{raw} is shown in Algorithm 2, where compression projection matrix P_{raw} is optimized to minimize the objective as described in Equation (5).

There are several hyperparameters involved in the training of P_{raw} , including α in Equation (5) and σ_{raw} in the initialization of P_{raw} . We set $\alpha = 1$ and $\sigma_{\text{raw}} = 0.1$ in all experiments. The learning rate is set to 10 and we train 500 epochs for each image-text pair.

Algorithm 2 Training algorithm for MetaCompress.

Require: $(I_{\text{IMG}}, I_{\text{TXT}})$: the image-text pair; $E_{\text{TXT}}(\cdot)$: the language encoder; $V_{\text{IMG}}(\cdot)$: the image encoder; $\text{LLM}(\cdot, \cdot)$: the vision-language decoder; P_{raw} : the learnable compression matrix with shape $m \times n$.

- 1: Initialize P_{raw} with Gaussian distribution $\mathcal{N}(0, \sigma_{\text{raw}}^2)$.
 - 2: $X_{\text{TXT}} \leftarrow E_{\text{TXT}}(I_{\text{TXT}})$
 - 3: $X_{\text{IMG}} \leftarrow V_{\text{IMG}}(I_{\text{IMG}})$
 - 4: $\mathbf{y} \leftarrow \text{LLM}(X_{\text{TXT}}, X_{\text{IMG}})$
 - 5: **while** not converged **do**
 - 6: $\tilde{X}_{\text{IMG}} \leftarrow \sigma(P_{\text{raw}})X_{\text{IMG}}$ # with gradients
 - 7: $\tilde{\mathbf{y}} \leftarrow \text{LLM}(X_{\text{TXT}}, \tilde{X}_{\text{IMG}})$ # with gradients
 - 8: Compute the final loss and gradient $\nabla_{P_{\text{raw}}}$ w.r.t. P_{raw} .
 - 9: Update P_{raw} with SGD optimizer.
 - 10: **end while**
-

8.2. MetaCompress

To adapt to arbitrary compression rate, the stride k in the pooling operation is set to a float value $\frac{n}{m}$, which can be easily implemented by the fractional max pooling operation [22]. We set the kernel size $s = 3$ for all experiments.

9. Properties of MetaCompress

Now, we analyze the properties of MetaCompress when $W_q = W_k = W$ and are drawn from $\mathcal{N}(0, \sigma_w^2)$ in Equation (9).

We start by considering when kernel size $k = 1$, meaning that $\text{Pool}(X)$ is a down-sampling operation to X , and we let \tilde{X} denotes the down-sampled image sequence. In this case, Equation (9) can be simplified as

$$P = \tilde{X}S X^\top, \quad (11)$$

where S is a positive semi-definite matrix. Therefore, the expectation $\mathbb{E}[p_{i,j}] = 0$ for all positions that $\tilde{x}_i \neq x_j$, and

for $\tilde{x}_i = x_j = x$

$$\mathbb{E}[p_{i,j}] = \mathbb{E}[xSx^\top] = \mathbb{E}[xW \text{diag}(\omega)W^\top x^\top]. \quad (12)$$

Here, we notice that $y = xW$ is still a random vector, with all elements subject to $\mathcal{N}(0, d\sigma_c^2\sigma_w^2)$. Hence,

$$\mathbb{E}[y \text{diag}(\omega)y^\top] = dd_c\sigma_c^2\sigma_w^2. \quad (13)$$

Considering that the embedding dimension of LVLMs is a large number (e.g., 4096 for LLaVA-1.5-7b), $\sigma(P_{\text{raw}})$ is close to the down-sampling projection to the input image sequence controlled by the stride s .

Further, when the kernel size $k > 1$, the expectation of $p_{i,j}$ is still zero when \tilde{x}_i is not captured by the pooling kernel located in x_i . To sum up, the initialization to Equation (9) converting MetaCompress to a interpretable pooling operation to the input image sequence.

However, as training progresses, W_q diverges from W_k , breaking the positive semi-definiteness of matrix S , enabling MetaCompress to further explore more effective compression strategies, ultimately enhancing performance. Besides, we choose the compression embedding dimension d_c to be smaller than the original embedding dimension d to reduce the computational cost and number of parameters.

Essentially, Equation (9) is a specialized form of the dot-product attention $XW_QW_K^\top X^\top$, making it easier to optimize and less prone to over-fitting to the training dataset, as we only adopt a few-shot subset for training efficiency.

10. More Results

10.1. Performance of Fixed Compression Matrix

Because we train the compression matrix P_{raw} for a single image on the training dataset, which is a straightforward optimization problem, we do not compare it with other methods. Here, we present the overall accuracy about compressing LLaVA-Next-7b on the MT-VQA-v2 dataset for reference. The accuracy of the base setting is 82.44, and when reducing 90% of the image token the accuracy decreases to 80.89.

10.2. More comparison Results

Table 4 presents the comparison results of different token reduction method with the compression rate of 70%. Our method achieves the best overall performance, the same as the results in Table 1.

As a supplement, Table 5 compares the effectiveness of token reduction methods. Here, ‘Spatial’ represents applying spatial pooling to the image sequence (the kernel size k is set to the same as the stride s). ‘ToMe’ [5] is a token merging

Table 4. The comparison of visual token reduction methods on three MT-VQA benchmarks with the reduction rate of 70%. The best and the second-best results are highlighted in bold and underline, respectively.

Model	Method	MT-VQA-v2				MT-GQA				ConvBench			
		Acc ₁	Acc ₂	Acc ₃	Avg	Acc ₁	Acc ₂	Acc ₃	Avg	S ₁	S ₂	S ₃	Avg
LLaVA-1.5-7b	Base	76.72	77.51	77.30	77.18	61.76	64.07	65.35	63.73	4.33	5.72	5.55	5.20
	Random	72.72	73.57	73.11	73.13	57.79	61.07	63.09	<u>60.65</u>	4.17	4.73	5.51	4.80
	Sample	72.96	73.71	73.33	73.33	58.48	61.12	62.30	<u>60.63</u>	4.16	5.03	3.81	4.33
	FastV	69.30	69.57	69.41	69.43	54.79	57.65	60.03	57.49	3.99	5.03	3.47	4.16
	PruMerge	72.79	73.90	73.42	<u>73.37</u>	57.89	59.54	61.81	59.75	3.64	4.68	4.68	4.33
	Ours	75.67	76.63	76.46	76.25	58.62	60.96	63.64	61.07	3.99	5.03	4.85	<u>4.62</u>
LLaVA-1.5-13b	Base	78.35	79.47	78.92	78.91	62.47	65.21	67.22	64.97	4.33	7.11	5.72	5.72
	Random	73.63	74.56	73.96	74.05	57.74	61.24	63.33	<u>60.77</u>	4.03	4.89	5.24	4.72
	Sample	73.81	74.79	74.51	74.37	58.51	61.29	62.82	<u>60.87</u>	3.64	5.03	5.37	4.68
	FastV	73.85	75.18	74.58	<u>74.54</u>	57.99	60.75	63.51	<u>60.75</u>	4.37	6.92	5.10	<u>5.46</u>
	PruMerge	73.80	75.16	74.57	<u>74.51</u>	57.67	60.45	62.18	60.10	4.51	6.41	5.37	5.43
	Ours	74.03	76.98	76.31	75.77	59.48	61.91	65.25	62.21	4.33	6.93	5.37	5.55
LLaVA-NeXT-7b	Base	80.20	80.86	80.71	80.59	63.83	66.68	67.94	66.15	7.95	11.46	7.58	9.00
	Random	76.18	77.43	77.64	77.08	61.96	63.95	66.29	64.07	7.97	9.19	6.93	<u>8.03</u>
	Sample	76.60	77.93	77.96	<u>77.50</u>	62.28	64.61	66.17	<u>64.35</u>	7.63	8.32	4.33	6.76
	FastV	75.96	76.86	76.39	76.40	61.54	64.37	65.97	63.96	0.00	0.00	2.50	0.83
	Ours	77.75	78.06	78.54	78.12	63.38	64.69	67.59	65.22	7.63	9.88	7.45	8.32
	LLaVA-NeXT-13b	Base	81.02	82.32	81.64	81.66	65.45	67.32	69.12	67.30	12.48	13.17	7.97
Random		77.30	78.77	78.65	78.24	62.89	64.64	67.22	64.92	11.44	11.27	8.15	10.29
Sample		77.51	79.15	79.03	<u>78.56</u>	63.95	64.54	67.20	<u>65.23</u>	10.40	14.04	6.76	<u>10.40</u>
FastV		75.78	77.16	76.66	<u>76.53</u>	62.10	64.37	65.06	63.84	20.00	5.00	3.33	9.44
Ours		78.14	80.98	80.21	79.78	64.86	65.89	67.59	66.11	10.92	13.86	7.11	10.63
XComposer-2.5-7b		Base	78.80	81.12	81.24	80.39	60.21	63.38	65.01	62.87	12.48	7.00	7.97
	Random	74.63	77.23	77.44	76.43	56.96	61.59	63.63	60.73	15.42	10.23	7.97	11.21
	Sample	75.07	77.68	78.04	76.93	57.79	61.17	63.36	60.77	15.94	11.79	6.07	<u>11.27</u>
	FastV	77.93	80.18	80.05	<u>79.39</u>	58.95	61.34	62.67	<u>60.99</u>	12.50	4.17	12.50	9.72
	Ours	78.24	80.53	80.79	79.85	60.11	62.28	64.14	62.18	15.77	12.13	6.24	11.38

Table 5. Comparison results of different token merging methods for LLaVA-1.5-7b.

Setting	MT-GQA			
	Acc ₁	Acc ₂	Acc ₃	Avg
Base	61.76	64.07	65.35	63.73
Sample	54.60	57.07	59.31	56.99
Spatial	51.05	54.62	56.24	53.97
ToMe	53.64	56.91	57.62	56.06
VisionZip	55.08	57.82	59.89	57.60
Ours	55.95	58.71	60.64	58.43

method proposed for ViTs rather than LVLMs, and thus performs ineffectively in our setting. ‘VisionZip’ [63] is a hybrid token compression method that integrates both token pruning and merging, yet it does not take MT-VQA scenarios into account and therefore also underperforms our method.

10.3. More Ablation Results

The results in Table 6 provide additional ablation studies across various LVLMs, further supporting the results in Table 3 and demonstrating that each objective contributes positively to the overall performance.

10.4. More Transfer Results

Table 7 reports more transfer validation experiments across LVLMs on MT-VQA-v2 and MT-GQA, showing that MetaCompress does not heavily depend on the specific training dataset, thereby demonstrating robust generalization ability. Furthermore, Table 8 reports transfer results on video question answering task. In detail, we transformed the video QA dataset Video-MME [21] into a 3-turn dialogue version, referred to as MT-Video-MME (including 500 dialogs for validation), and conducted comparative evaluations against baseline methods at a 70% compression rate. Since XComposer-2.5-7B natively supports video input, we directly use the

Table 6. Additional ablation study of training MetaCompress for various LVLMs using different loss terms on MT-GQA. Gradient clipping is only applied for the ‘ $\mathcal{L}_{\text{collapse}} + \text{Grad Clip}$ ’ setting.

$\mathcal{L}_{\text{pred}}$	$\mathcal{L}_{\text{entropy}}$	$\mathcal{L}_{\text{collapse}}$	$\mathcal{L}_{\text{collapse}} + \text{Grad Clip}$	LLaVA-1.5-7b	LLaVA-NeXT-7b	XComposer-2.5-7b
✓	✗	✗	✗	56.63	61.98	56.77
✓	✓	✗	✗	57.99	62.42	58.01
✓	✗	✓	✗	52.26	56.34	52.53
✓	✗	✗	✓	57.57	62.13	58.24
✓	✓	✗	✓	58.43	62.70	58.68

Table 7. Transfer validation experiments. Average accuracy is reported for cross-dataset transfer between MT-VQA-V2 and MT-GQA, all under a 90% token reduction rate.

Settings	LLaVA-1.5-7b	LLaVA-1.5-13b	LLaVA-NeXT-7b	LLaVA-NeXT-13b	XComposer-2.5-7b
MT-VQA-v2	70.65	72.94	75.18	75.26	75.76
MT-GQA → MT-VQA-v2	69.06	71.89	73.61	73.25	74.41
MT-GQA	58.43	59.48	62.70	63.12	58.68
MT-VQA-v2 → MT-GQA	57.45	58.78	61.43	62.60	58.06

Table 8. Transfer results on MT-Video-MME. Average accuracy is reported across different methods.

Metrics	Base	Random	Sample	FastV	Ours
Acc_1	44.3	26.8	25.5	26.2	28.5
Acc_2	46.7	25.3	26.4	27.3	27.3
Acc_3	48.2	30.7	31.3	31.6	34.6
Avg	46.4	27.6	27.7	28.4	30.1

pre-trained weights obtained from training on the small combined dataset of MT-VQA-v2 and MT-GQA (only around 20k samples in total as we mentioned) to evaluate on the MT-Video-MME benchmark. As shown in the table, MetaCompress outperforms baseline approaches even without any task-specific training on MT-Video-MME, further demonstrating its strong transferability.

11. More Visualizations

Figures 6 and 7 show the visualization of the generated compression projection for LLaVA-1.5-7b and LLaVA-1.5-13b on MT-GQA with a compression rate of 90% (we randomly select two images as examples). The row and column indices in the figures represent the original and reduced token indices, respectively, with darker colors indicating higher retention weights. As observed, MetaCompress performs pruning and merging operations at different positions, but is primarily based on equidistant down-sampling, with specific adaptations for certain tokens.

12. Discussions

Most data-driven approaches for efficient model inference primarily focus on model pruning [43, 52], efficient at-

tention mechanisms [39, 50], and efficient model architectures [33, 41, 56], particularly in designing vision encoders for stronger and more compact visual representations. However, these methods typically require fine-tuning the entire model, resulting in substantial computational overhead. In contrast, our proposed MetaCompress trains only a small number of lightweight linear projection layers (D_q , D_k , and w), yet it surpasses existing token reduction approaches. Owing to its efficiency, MetaCompress requires only a modest amount of training data (approximately 20k samples) while exhibiting strong transferability across datasets, as demonstrated in our transfer experiments. This generalization capability stems from the fact that MetaCompress is trained to preserve as much general visual information as possible for multi-turn dialogues rather than being specialized for specific image domains.

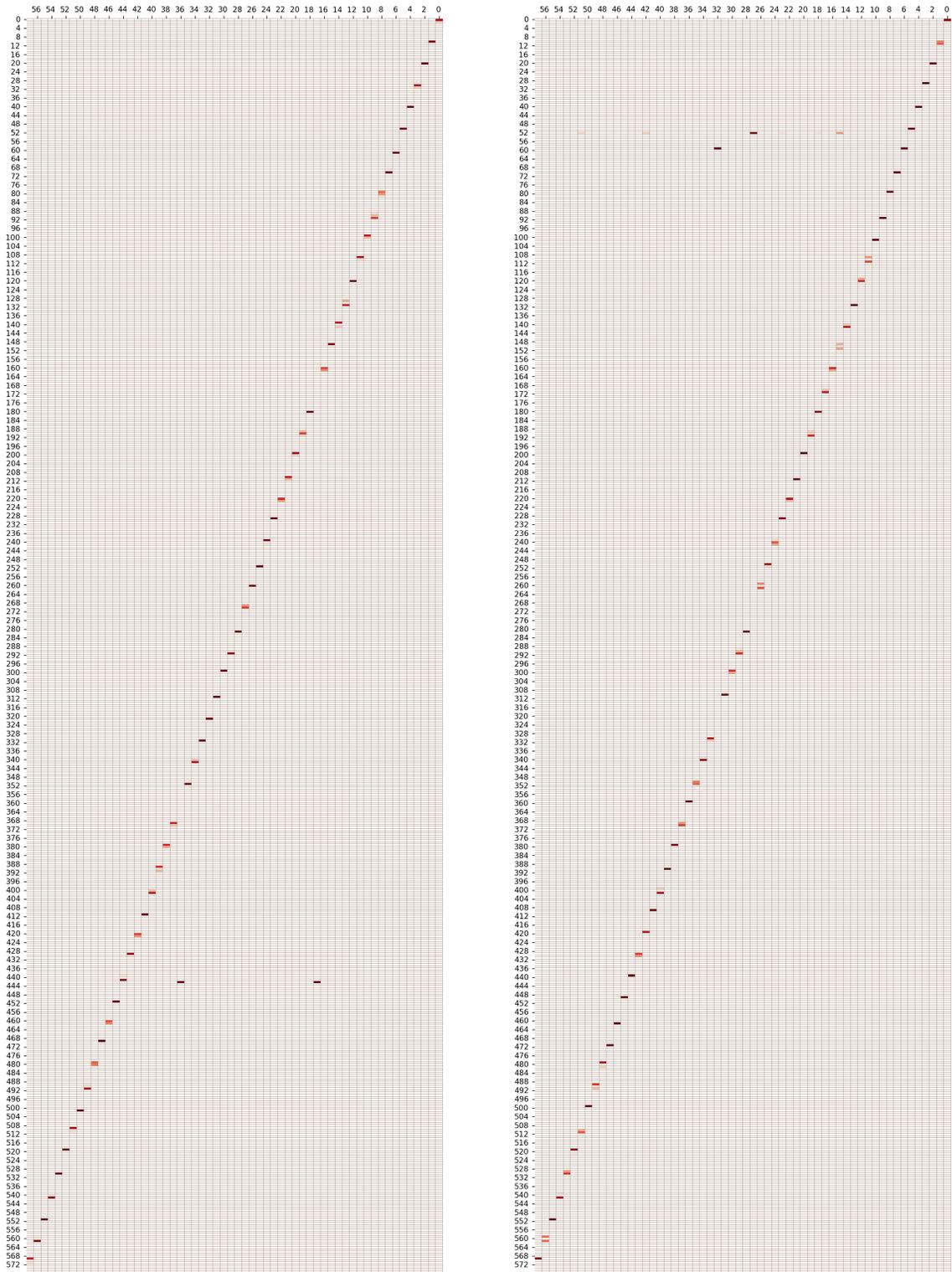


Figure 6. Visualization of the compression projection for LLaVA-1.5-7b on MT-GQA with the compression rate of 90%.

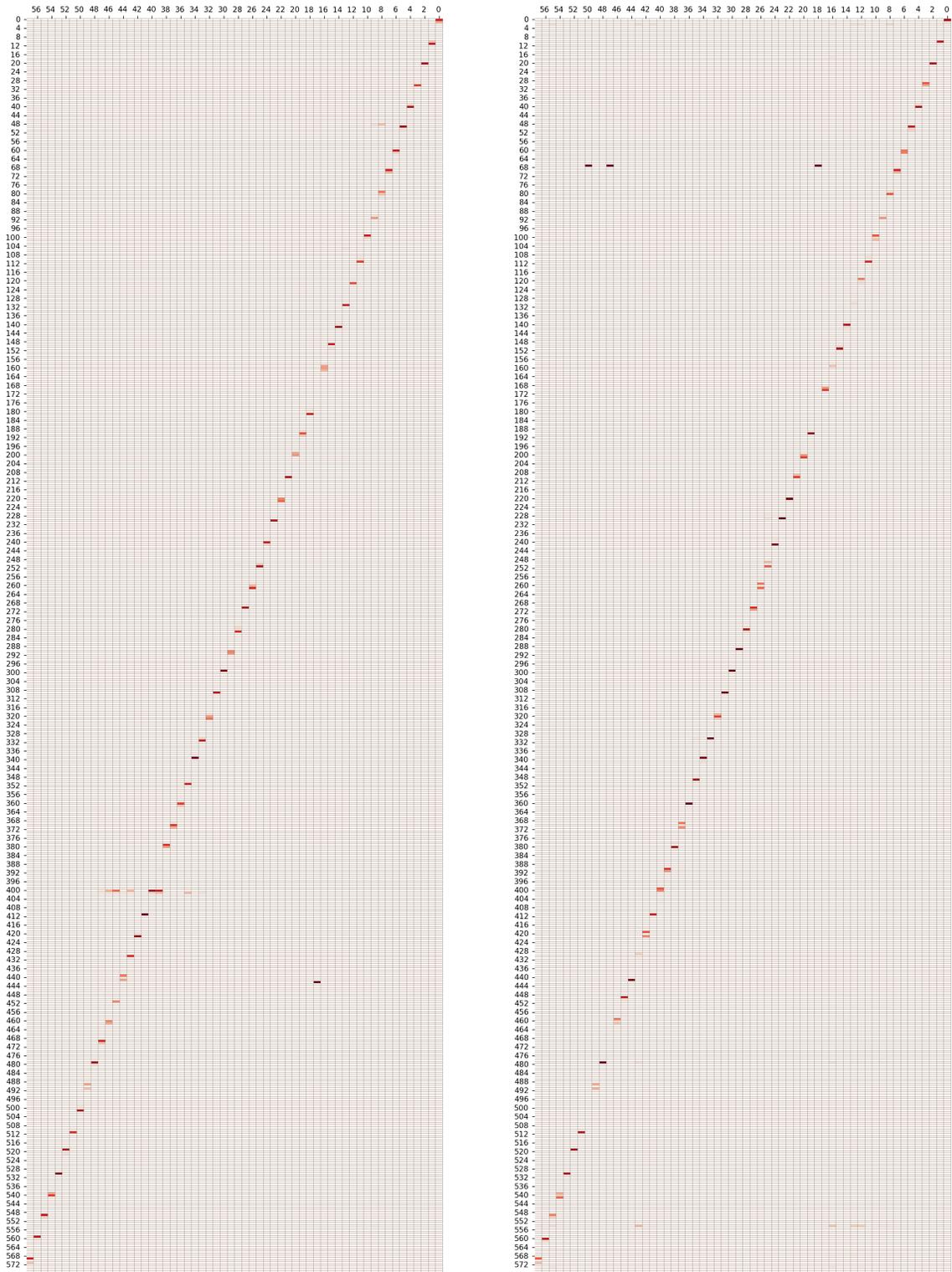


Figure 7. Visualization of the compression projection for LLaVA-1.5-13b on MT-GQA with the compression rate of 90%.