Linli Yao[1], Long Xing[1], Yang Shi[1], Sida Li[1], Yuanxin Liu[1], Yuhao Dong[1], Yi-Fan Zhang[1], Lei Li[1], Qingxiu Dong[1], Xiaoyi Dong[1], Qidong Huang[1], Haotian Wang[1], Feng Wu[1], Yuanxing Zhang[1], Pengfei Wan[1], Zhouchen Lin[1], and Xu Sun[1]

[1]Affiliation not available

January 12, 2026

Towards Efficient Multimodal Large Language Models: A Survey on Token Compression

Linli Yao*‡, Long Xing*, Yang Shi*, Sida Li, Yuanxin Liu, Yuhao Dong, Yi–Fan Zhang, Lei Li, Qingxiu Dong, Xiaoyi Dong, Qidong Huang, Haotian Wang, Feng Wu, Yuanxing Zhang, Pengfei Wan, Zhouchen Lin†, Xu Sun†

Abstract—Multimodal Large Language Models (MLLMs) have made significant strides in integrating vision-language perception, alignment, and reasoning. However, the increasing complexity of tasks such as high-resolution image processing and long video understanding has led to an exponential rise in visual context length within MLLMs. The resulting long-context token sequences impose substantial computational demands on large language models (LLMs), leading to quadratic complexity growth, heightened GPU resource consumption, and slower inference speeds. To address these challenges, token compression has emerged as a promising research direction that reduces the number of tokens processed within MLLMs while preserving essential cross-modal semantic information, thereby enhancing both training and inference efficiency. This survey provides a comprehensive review of token compression techniques for MLLMs, examining the current state of research and exploring future directions. We propose a taxonomy of token compression methods based on their application modules within the MLLM system, including the vision encoder, projector, LLM backbone, and hybrid approaches. We analyze the strengths and limitations of widely adopted algorithms, offering practitioners a structured framework for selecting appropriate token compression strategies. Finally, we discuss practical applications of token compression, identify key challenges in the field, and propose potential directions for future research and development. All related resources are available at https://github.com/yaolinli/MLLM-Token-Compression.

Index Terms—Multimodal Large Language Model, Token Compression, Token Reduction, Efficient Multimodal Learning, Long-Context Modeling, Video Large Language Model, Vision and Language.

1 Introduction

Multimodal Large Language Models (MLLMs) [1]–[11] rapidly advanced the frontier of vision-language joint perception, alignment, reasoning, and generation [12]–[17] [12]–[17]. By integrating the remarkable language understanding capabilities of Large Language Models (LLMs) [18]–[22] with comprehensive visual perception abilities from vision encoders [23], contemporary systems such as LLaVA [24], Qwen-VL [25] and GPT-4o [26] exhibit strong performance on diverse tasks spanning open-ended visual question answering, document understanding, and multi-step visual reasoning, among others.

However, these advanced cross-modal capabilities incur substantial computational costs. High-resolution images and long videos can generate hundreds to thousands of visual tokens, while multi-turn dialogue and chain-of-thought reasoning further extend the historical context [27]–[29]. As sequence lengths increase, the quadratic complexity of attention in Transformer-based MLLMs results in prohibitive memory consumption and latency, limiting both scalability and deployment. This tension between multimodal effectiveness and computational efficiency has made compressing multimodal token sequences an urgent research focus.

To build more efficient MLLMs, token compressing multimodal token sequences refers to methods that reduce the number of tokens processed by MLLMs while preserving critical cross-modal semantics. Conceptually, compression targets redundancy in spatial structure (e.g., repetitive background regions), temporal continuity (e.g., frame-to-frame similarities), and modality alignment (e.g., text-conditioned visual irrelevance), yielding shorter sequences with minimal essential information degradation. Historically, token compression originated in unimodal vision through patch dropping, token merging, and dynamic sparsification in Vision Transformers [30]–[36]. These approaches have since been extended to multimodal settings, where compression can operate on visual streams, textual streams, or their fusion. As depicted in Figure 1, multimodal token compression techniques [37]–[54] have evolved rapidly since 2022 and experienced significant growth from 2024 onward. Recent works [55]–[64] extend this research direction from spatial images to long-horizon video understanding with extreme compression ratios, where aggressive token compression must be balanced against fine-grained localization, temporal coherence, and temporal grounding performance.

Despite steady progress in token compression, practitioners still face critical challenges in selecting or designing token compression strategies for MLLMs. This survey systematically examines the fundamental issues of token compression from three perspectives.

- *L. Yao, Y. Shi, S. Li, Y. Liu, Q. Dong, Z. Lin, and X. Sun are with Peking University (Contact e-mail: linliyao@stu.pku.edu.cn).*
- *L. Xing and F. Wu are with the University of Science and Technology of China.*
- *Y. Dong is with Nanyang Technological University.*
- *Y.-F. Zhang is with the Institute of Automation, Chinese Academy of Science.*
- *L. Li is with the University of Hong Kong.*
- *X. Dong is with Microsoft.*
- *Q. Huang is with Alibaba Cloud.*
- *H. Wang is with the National University of Defense Technology.*
- *Y. Zhang and P. Wan are with the Kling Team, Kuaishou Technology.*
- **Equal contributions. †Corresponding authors. ‡Project Leader.*

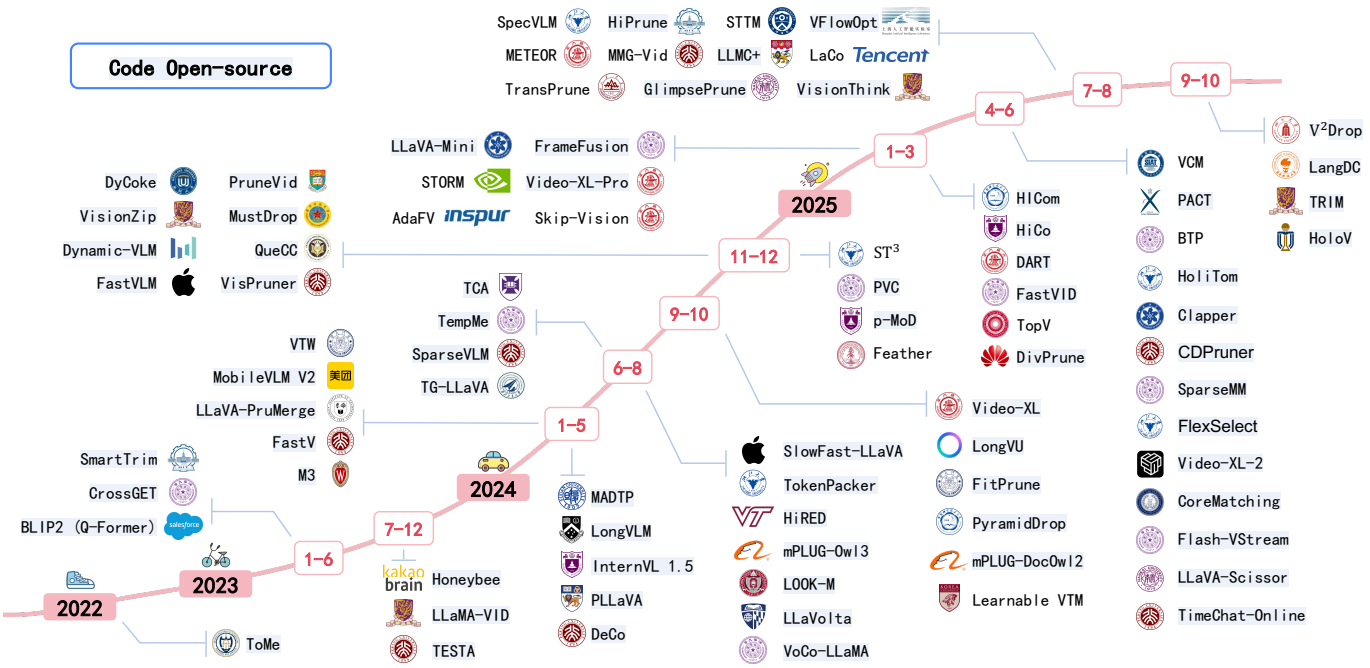

Fig. 1: A timeline of representative token compression methods for MLLMs is presented. The timeline is organized primarily based on the earliest arXiv submission dates. Methods with publicly available code are highlighted. Due to space constraints, only a subset of representative open-source approaches is included in the figure.

First, *where and how should token compression be applied within the MLLM architecture?*

Different modules in MLLMs, including the vision encoder, projector, and large language model, introduce distinct architectural characteristics, information bottlenecks, and computational trade-offs. The placement of compression strongly influences the preservation of visual semantics, the quality of cross-modal alignment, and downstream reasoning capability, yet lacks systematic analysis to guide architectural choices.

Second, *which compression mechanism best suits specific deployment scenarios?* The commonly-adopted design space spans token merging versus pruning, text-guided versus purely visual compression, objectives for training versus inference acceleration, and plug-in modules versus end-to-end retraining. Each paradigm offers distinct benefits and limitations that must be aligned with application-specific constraints. We aim to clarify these trade-offs and provide decision guidelines for practitioners.

Third, *what are the remaining open challenges and promising future directions?* As token compression represents an active research field undergoing rapid development, it is essential to identify unresolved issues and emerging opportunities. We discuss key challenges including the lack of theoretical foundations, adaptation to dynamic compression requirements, efficiency-effectiveness trade-offs in fine-grained tasks (e.g., chart understanding and OCR), and the need for more rigorous evaluation protocols. Based on these perspectives, we aim to shed light on promising future research directions

This survey addresses these fundamental questions through structured analysis. Compared to existing surveys on efficient MLLMs [159], [160] and efficient vision transformers [161], this work focuses specifically on token compression techniques for MLLMs, with emphasis on efficient long-context sequence processing. A concurrent survey [162] examines token compression across comprehensive modalities including image, video, and audio. Our work is distinctly motivated by the goal of systematically organizing existing token compression methods according to MLLM architectural components (*where to compress*) and providing a practical roadmap of compression techniques (*how to select*).

Our main contributions are summarized as follows:

(i) **Taxonomy of token compression by MLLM architectural placement (§3).** We introduce a systematic taxonomy that categorizes token compression methods by their application location within MLLMs—vision encoder, projector, or large language model—clarifying how architectural placement interacts with compression objectives and how hybrid strategies can synergistically combine approaches across different modules.

(ii) **Methodological analysis and design roadmap (§4).** Complementing the architectural taxonomy, we analyze the prevailing token compression *mechanisms* employed across these locations. We dissect critical design dimensions, including text-guided versus vision-only compression, token pruning versus merging, modular plug-ins versus end-to-end retraining, and training-centric versus inference-centric optimization. Based on this methodological breakdown, we provide a selection roadmap to guide researchers in choosing the optimal compression techniques tailored to specific tasks, accuracy targets, and latency constraints.

Grounded in the above analysis, we further summarize open challenges in this field and aim to shed light on efficient next-generation MLLMs. We highlight pivotal future directions, such as task-aware adaptivity and refined

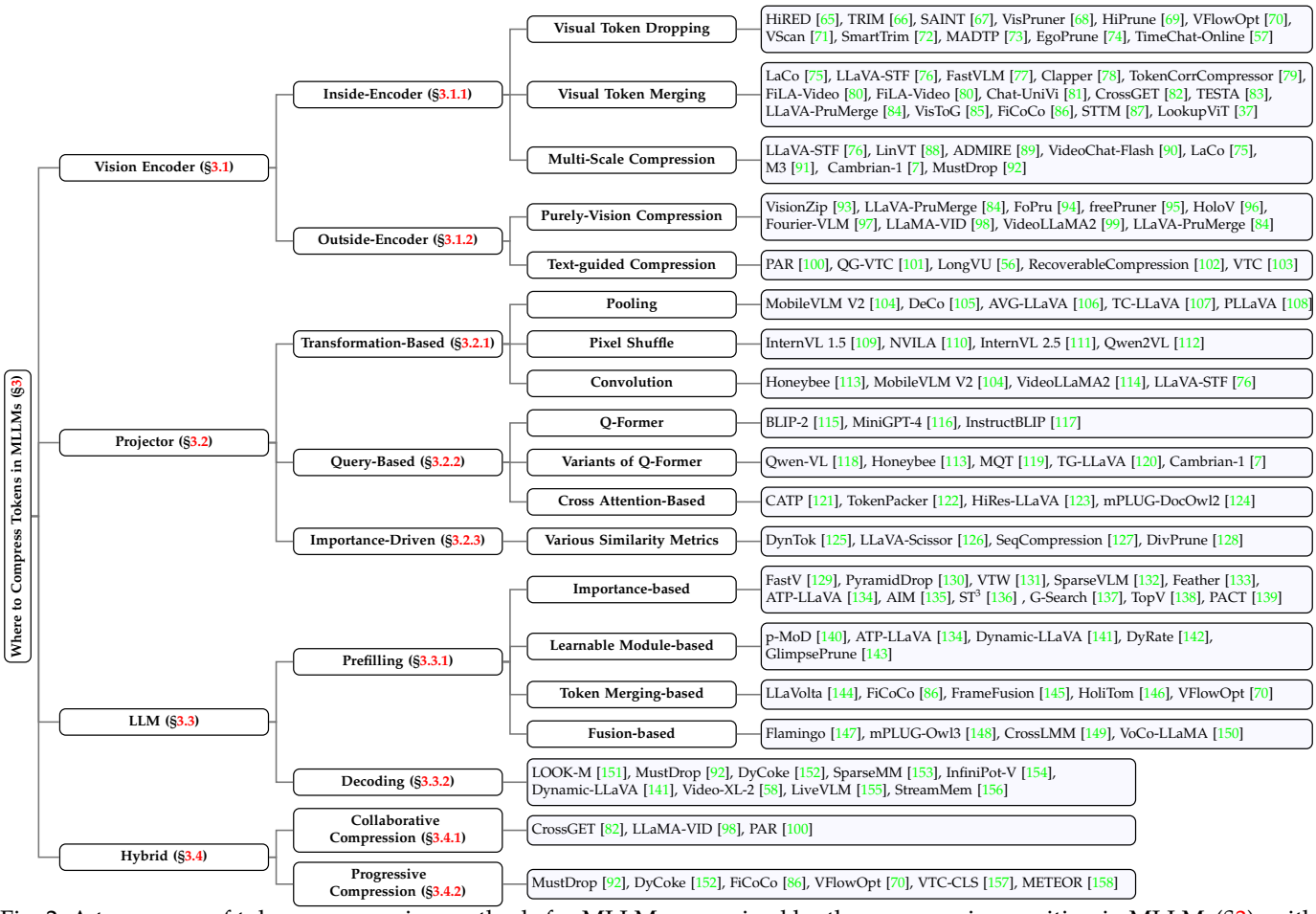

Fig. 2: A taxonomy of token compression methods for MLLMs, organized by the compression position in MLLMs(§3), with leaf nodes illustrating representative works.

evaluation protocols, with the ultimate goal of making multimodal intelligence both powerful and affordable at scale.

2 Preliminaries

This section lays the foundation for token compression in Multimodal Large Language Models (MLLMs). We begin with an overview of typical MLLM architectures (§2.1), followed by a formal definition of token compression techniques (§2.2).

2.1 Multimodal Large Language Models

The rapid advancement of artificial intelligence has witnessed a paradigm shift from unimodal models to sophisticated multimodal systems capable of understanding and reasoning across diverse data modalities. MLLMs represent a significant milestone in this evolution, combining the remarkable language understanding capabilities of Large Language Models (LLMs) [18], [163]–[166] with comprehensive visual perception abilities to create systems that can process, understand, and generate responses based on both textual and visual information.

Modern MLLMs typically adopt a three-component architecture: A *vision encoder (VE)* (often based on SigLIP [167] or CLIP [168]) that processes visual inputs into high-dimensional feature representations, a *projector* that aligns visual features with the language model's embedding space, and a powerful *LLM* that performs multimodal alignment, reasoning and generation. This architectural design enables end-to-end training and seamless integration of visual and textual information processing. Throughout this survey, we focus on token compression techniques designed for this mainstream three-component architecture. Alternative architectural paradigms [169], [170] that deviate from this design are beyond the scope of our discussion.

Formally, let $\mathcal{X}^{\mathrm{v}} = \{I_1, I_2, \ldots, I_{n_{\mathrm{v}}}\}$ with $n_{\mathrm{v}} \geq 1$ denote the input image sequence or video frames, and $\mathcal{X}^{\mathrm{t}} = \{x_1, x_2, \ldots, x_{n_{\mathrm{t}}}\}$ represent the textual token sequence comprising system prompts, user instructions, or dialogue history. The MLLM architecture consists of three key components:

Vision Encoder. The vision encoder $\mathcal{E}_{\mathrm{v}}$ transforms raw visual inputs into a sequence of dense visual token representations:

$$\mathbf{Z}^{\mathrm{v}} = \mathcal{E}_{\mathrm{v}}(\mathcal{X}^{\mathrm{v}}) \in \mathbb{R}^{n_{\mathrm{v}} \times d_{\mathrm{v}}}, \tag{1}$$

where n_{v} denotes the number of visual tokens and d_{v} represents the feature dimension of each visual token.

Projector. To bridge the modality gap between visual and textual representations, a projector $\mathcal{P}$ transforms visual features from dimension d_{v} to the LLM's embedding space:

$$\mathbf{H}^{\mathrm{v}} = \mathcal{P}(\mathbf{Z}^{\mathrm{v}}) \in \mathbb{R}^{n_{\mathrm{v}} \times d_{\mathrm{t}}}, \tag{2}$$

where d_{t} denotes the embedding dimension of the target language model.

Large Language Model. The LLM $\mathcal{G}$ processes the concatenated sequence of projected visual tokens and embedded textual tokens:

$$\mathbf{Y} = \mathcal{G}\big([\mathbf{H}^{\mathrm{v}}; \mathcal{E}_{\mathrm{t}}(\mathcal{X}^{\mathrm{t}})]\big), \tag{3}$$

where $\mathcal{E}_{\mathrm{t}}(\cdot)$ represents the embedding layer of the LLM, $[\cdot;\cdot]$ denotes concatenation along the sequence dimension, and $\mathbf{Y}$ is the generated output sequence.

Computational Complexity. The aforementioned components in MLLMs primarily employ Transformer-based architectures [171], renowned for their powerful representation capabilities but also characterized by high computational costs for processing long input sequences. The computational complexity predominantly stems from the self-attention mechanism and feed-forward networks (FFNs) within Transformer layers.

Given a sequence of length n, a hidden dimension size d, and an intermediate dimension m in the FFN, the computational cost per Transformer layer can be approximated as

$$\text{Layer FLOPs} = 8nd^2 + 4n^2d + 4ndm. \tag{4}$$

Thus, for an L-layer Transformer, the total cost is

$$\text{Total FLOPs} = L \times \left(8nd^2 + 4n^2d + 4ndm\right), \tag{5}$$

where $n = n_t + n_v$ is the overall sequence length (text tokens n_t plus visual tokens n_v).

As the sequence length n increases, the **quadratic complexity** term $2n^2d$ in the attention mechanism grows rapidly, leading to prohibitive computational overhead. This computational bottleneck is particularly pronounced in scenarios involving: (1) high-resolution images or long videos, where n_{v} typically dominates n_{t} in MLLMs, and (2) multi-turn conversations or complex reasoning tasks requiring extensive contextual history.

2.2 Token Compression

The quadratic computational complexity in MLLMs naturally motivates the development of *token compression* techniques (also known as *token reduction*), which aim to reduce the total context length in the MLLM while preserving essential visual and textual semantics, thereby achieving computational efficiency without remarkably compromising model performance.

Formally, denote the total visual and textual token number in the MLLM as $N = n_{\mathrm{t}} + n_{\mathrm{v}}$, token compression aims to reduce the N to a smaller M to improve efficiency by selecting or aggregating original tokens, where $M < N$. The token compression process can be represented as a function $\mathcal{C}$ that maps the original token sequence to a compressed sequence:

$$\mathbf{H}_{\mathrm{comp}} = \mathcal{C}(\mathbf{H}) \in \mathbb{R}^{M \times d_{\mathrm{t}}}, \tag{6}$$

where $\mathbf{H} = [\mathbf{H}^{\mathrm{v}}; \mathbf{H}^{\mathrm{t}}] \in \mathbb{R}^{N \times d_{\mathrm{t}}}$ is the concatenated sequence of projected visual tokens and embedded textual tokens, and $\mathbf{H}_{\mathrm{comp}}$ is the compressed token sequence.

Compression Ratio is a widely-mentioned concept in token compression, defined as:

$$R_{\mathrm{comp}} = \frac{N}{M}, \tag{7}$$

where higher values (e.g., $4\times$ or $8\times$) indicate greater compression levels, more compact semantic representations, and consequently larger efficiency gains.

Since the number of visual tokens typically exceeds that of textual tokens by substantial margins (e.g., by $20\times$ [93]) in MLLMs, most existing token compression methods primarily focus on reducing n_{v}. To achieve more compact visual representations within MLLMs, two main types of redundancy can be exploited:

(i) Intra-Visual Redundancy. Visual content inherently contains redundant information. In images, numerous patches may represent background elements that are not crucial for understanding the primary subject matter. Similarly, in videos, consecutive frames often exhibit substantial similarity, resulting in temporal redundancy. This redundancy can be leveraged to reduce the number of visual tokens requiring processing, thereby improving computational efficiency while maintaining information quality.

(ii) Cross-Modal Redundancy. In multimodal tasks, particularly question-answering scenarios, textual input provides contextual guidance that can identify the most relevant visual tokens. For instance, when a question focuses on a specific object within an image, only visual tokens corresponding to that object may be necessary for accurate comprehension and response generation. By exploiting textual information, it becomes possible to selectively retain only those visual tokens that are pertinent to the specific task requirements.

3 Where to Compress Tokens in MLLMs

Based on the taxonomy illustrated in Figure 2, we systematically categorize existing token compression methods according to *where compression is applied* within the MLLM architecture. Throughout the processing procedure from visual input to textual output, token compression strategies can be progressively deployed at three architectural modules: (1) the **Vision Encoder (§3.1)**, where compression reduces computational overhead at the visual perception stage; (2) the **Projector (§3.2)**, which integrates token reduction during the transformation from visual to linguistic representation space; and (3) the **Large Language Model (§3.3)**, where compression achieves holistic cross-modal efficiency optimization.

3.1 Token Compression in Vision Encoder

In MLLMs, visual data are inherently more redundant than text [191]–[193], leading to a substantially larger number of tokens on the vision side than on the language side. For instance, a single high-resolution image can be divided into thousands of patch tokens [10], [112]. If these tokens are simply concatenated with text tokens and processed as an "interleaved long sequence", the subsequent pre-filling and decoding stages of the LLM incur quadratic computational complexity with respect to the sequence length. Since the vision encoder (VE) is the first module to encode visual

TABLE 1: Summary of representative token compression works (venues up to Oct. 2025). **Modality** denotes the primary application scenario. **Compression Position** indicates the application stage (Vision Encoder, Projector, or LLM). **Text Query-based** marks dependency on text token guidance. **Re-train** vs. **Plug-in** distinguishes methods requiring additional training from plug-and-play modules.

#	Method	Date	Venue	Modality	Compression Position	Text Query-based	Re-train/Plug-in
1	ToMe [34]	2022.01	ICLR	image	Vision Encoder	no	re-train,plug-in
2	BLIP2 [172]	2023.01	ICML	image	Projector	yes	re-train
3	MovieChat [173]	2023.07	CVPR	video	Vision Encoder	no	plug-in
4	MobileVLM V2 [104]	2024.02	arXiv	image	Projector	no	re-train
5	LLaVA-PruMerge [84]	2024.03	ICCV	image,video	Vision Encoder	no	re-train,plug-in
6	FastV [129]	2024.03	ECCV	image,video	LLM	yes	plug-in
7	M3 [91]	2024.05	ICLR	image,video	Vision Encoder	no	re-train
8	DeCo [105]	2024.05	arXiv	image	Projector	no	re-train
9	VoCo-LLaMA [150]	2024.06	CVPR	image,video	LLM	no	re-train
10	TokenPacker [122]	2024.07	IJCV	image	Projector	no	re-train
11	HiRes-LLaVA [123]	2024.07	CVPR	image	Projector	no	re-train
12	mPLUG-Owl3 [148]	2024.08	arXiv	image,video	LLM	yes	re-train
13	HiRED [65]	2024.08	AAAI	image	Vision Encoder	no	re-train,plug-in
14	TempMe [174]	2024.09	ICLR	video	Vision Encoder	no	re-train
15	Video-XL [55]	2024.09	CVPR	video	LLM	no	re-train
16	PyramidDrop [130]	2024.10	CVPR	image,video	LLM	yes	plug-in
17	SparseVLM [132]	2024.10	ICML	image,video	LLM	yes	plug-in
18	LongVU [56]	2024.10	ICML	video	Vision Encoder, Projector	yes	re-train
19	TCA [175]	2024.10	ICCV	image	Vision Encoder	no	plug-in
20	QueCC [176]	2024.11	ICLR	image	Projector	yes	re-train
21	ATP-LLaVA [177]	2024.12	CVPR	image	LLM	yes	re-train
22	VisPruner [68]	2024.12	ICCV	image,video	Vision Encoder	no	plug-in
23	VisionZip [93]	2024.12	CVPR	image,video	Vision Encoder, Projector	no	re-train,plug-in
24	Dynamic-VLM [41]	2024.12	ICCV	image,video	Vision Encoder	no	re-train
25	PVC [178]	2024.12	CVPR	image,video	Vision Encoder,Projector	no	re-train
26	PruneVid [179]	2024.12	ACL	video	Projector,LLM	yes	plug-in
27	Feather [133]	2024.12	ICCV	image	LLM	yes	plug-in
28	HiCo [180]	2025.01	arXiv	video	Vision Encoder,LLM	yes	re-train
29	LLaVA-Mini [181]	2025.01	ICLR	image,video	Projector	yes	re-train
30	FALCON [44]	2025.01	ICCV	image	Vision Encoder	no	re-train
31	FCoT-VL [182]	2025.02	arXiv	image	Projector	no	re-train
32	DART [183]	2025.02	EMNLP	image	Vision Encoder	no	plug-in
33	DivPrune [128]	2025.03	CVPR	image,video	Projector	no	plug-in
34	FastVID [184]	2025.03	NeurIPS	video	Projector	no	plug-in
35	TopV [138]	2025.03	CVPR	image,video	LLM	no	plug-in
36	Skip-Vision [185]	2025.03	ICCV	image	Vision Encoder,LLM	no	plug-in
37	TimeChat-Online [57]	2025.04	ACM MM	video	Projector	no	re-train,plug-in
38	VCM [186]	2025.04	arXiv	image	Projector	yes	re-train
39	HoliTom [146]	2025.05	NeurIPS	video	Projector,LLM	no	plug-in
40	ToDRE [187]	2025.05	arXiv	image,video	Projector,LLM	no	plug-in
41	BTP [188]	2025.05	NeurIPS	image	LLM	yes	plug-in
42	DynTok [125]	2025.06	arXiv	video	Projector	no	plug-in
43	LLaVA-Scissor [126]	2025.06	arXiv	video	Projector	no	plug-in
44	SparseMM [153]	2025.06	ICCV	image	LLM	yes	plug-in
45	Video-XL-2 [58]	2025.06	arXiv	video	Vision Encoder,LLM	yes	re-train
46	FlexSelect [189]	2025.06	NeurIPS	video	Vision Encoder	yes	re-train,plug-in
47	VisionThink [190]	2025.07	NeurIPS	image	Projector	yes	re-train
48	STTM [87]	2025.07	ICCV	video	Vision Encoder	no	plug-in
49	METEOR [158]	2025.07	ICCV	image	Vision Encoder, Projector, LLM	yes	re-train
50	VFlowOpt [70]	2025.08	ICCV	image,video	Vision Encoder,LLM	yes	re-train,plug-in
51	CATP [121]	2025.08	arXiv	image	Projector,LLM	yes	plug-in
52	SpecVLM [61]	2025.08	EMNLP	video	LLM	yes	plug-in
53	LangDC [64]	2025.09	EMNLP	video	Projector	yes	re-train
54	HoloV [96]	2025.10	NeurIPS	image,video	Vision Encoder	no	plug-in

inputs, reducing visual tokens at this initial stage yields disproportionately large efficiency gains throughout the entire MLLM system. As shown in Figure 3, we first review and categorize vision-side token compression methods applied at the vision encoder module into two broad categories:

- **Inside Vision Encoder Compression (Inside-VE, §3.1.1):** Compression is applied within the ViT or video encoder itself. Methods in this category either discard redundant tokens or merge similar ones. Since different layers capture multi-scale semantics—ranging from low-level textures to high-level concepts—multi-scale compression schemes have been developed to coordinate compression across layers.
- **Outside Vision Encoder Compression (Outside-VE, §3.1.2):** Compression occurs after the vision encoder produces its output tokens but before the projector maps these tokens into the language model space. This design is plug-and-play and minimally invasive to the original architecture. Depending on whether textual signals are incorporated, methods can be grouped into purely vision-based approaches and text-guided approaches.

3.1.1 Inside-Encoder Compression

Inside-VE compression directly alters token flow *within* the encoder, reducing self-attention complexity at an early stage and shortening the propagation path of tokens. The design revolves around two questions: (1) how to handle "unimportant" tokens through pruning or merging; and (2) how to coordinate compression across multiple layers or encoders to leverage multi-scale visual features. Here we focus exclusively on methods applied in multimodal LLMs, and do not review token compression for pure vision tasks [161].

Visual Token Dropping. Token dropping methods compute importance scores for visual tokens within the vision encoder and retain only the most salient ones, directly discarding the remainder. Implementation typically follows a "ranking + Top-K" paradigm with defined thresholds. To identify important visual tokens within the encoder, existing methods employ three principal scoring strategies:

Similarity-based scoring. These methods quantify token redundancy by measuring the similarity between each visual token and a global representation (e.g., CLS token or aggregated feature vector). Tokens exhibiting high similarity are deemed redundant and removed. Representative works include TRIM [66] and SAINT [67], which employ global similarity metrics with layer-adaptive thresholds. TRIM leverages CLIP embeddings to measure the relevance between textual queries and visual tokens, employing an adaptive Interquartile Range (IQR)-based thresholding mechanism to select the most query-relevant tokens. SAINT advances this paradigm by leveraging token similarity within a graph-based formulation to dynamically optimize both pruning rates and redundancy thresholds, offering greater flexibility than fixed strategies.

Attention-based scoring. These approaches leverage attention weights from the vision transformer to derive token saliency. The first category restricts pruning decisions to vision-only attention patterns. VisPruner [68] and

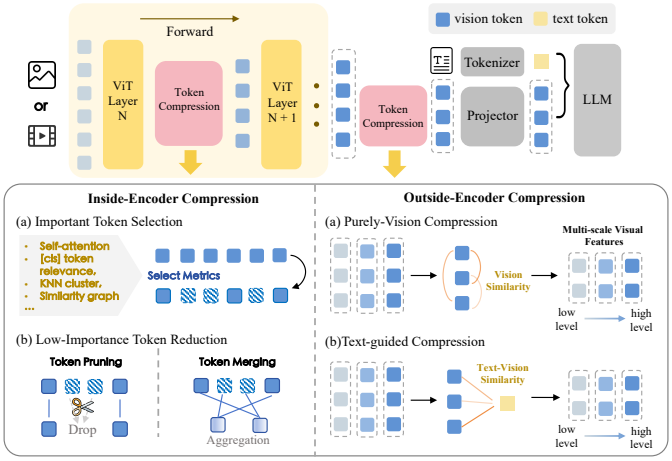

Fig. 3: Illustration of token compression strategies applied at the vision encoder module in MLLMs.

HiPrune [69] leverage the CLS token attention in the vision transformer to assess the visual importance of image partitions. VFlowOpt [70] constructs an importance map by integrating visual attention-derived context relevance with patch-level information entropy to determine which tokens to prune. The second category incorporates cross-modal attention to evaluate token significance. MADTP [73] introduces a Token Importance Score (TIS) that integrates three attention mechanisms—class attention, self-attention, and cross-modal alignment attention—and employs learnable thresholds with sparsemax activation to dynamically determine pruning masks. SmartTrim [72] adopts a cross-modal guidance approach by feeding the CLS token into a lightweight policy network that learns importance scores based on cross-modal information.

Heuristic-based scoring. These methods exploit task-specific priors to guide token selection. EgoPrune [74] leverages domain-specific heuristics from egocentric videos, utilizing geometric stability and field-of-view dynamics to prioritize motion-relevant regions while pruning static backgrounds. METEOR [158] adopts a layer-adaptive strategy based on the prior that shallow and deep layers encode fundamentally different types of information. Specifically, METEOR employs similarity to the average token as the pruning criterion in shallow layers, where low-level redundancies dominate, and class attention scores in deep layers, where semantic information is more concentrated.

Visual Token Merging. Unlike pruning, which deletes tokens outright, merging aggregates similar tokens into compact representations to preserve information while shortening sequences [34]. A fundamental principle underlying merging operations is *proximity-based redundancy*: tokens that are close to each other spatially or temporally tend to exhibit high redundancy. Beyond proximity heuristics, more sophisticated methods leverage explicit similarity measurements or hybrid compression pipelines to achieve semantic merging.

Proximity-based Merging. Spatial and temporal adjacency provide natural bases for identifying redundant visual tokens, as neighboring patches or consecutive frames typically

share similar features. For spatial merging, structured approaches perform deterministic aggregation through downsampling operations [77] or pixel-shuffle with channel merging [75], while learnable methods adopt adaptive convolution kernels [76] or density-based clustering [81] to capture task-specific patterns beyond uniform averaging.

In video understanding, temporal proximity enables cross-frame consolidation through two complementary strategies: joint temporal-spatial aggregation that merges similar frames and patches simultaneously [71], [83], and frame-level fusion that adaptively integrates consecutive frames with learnable importance weighting [80], [87]. By exploiting the inductive bias that adjacent tokens exhibit high correlation across both spatial and temporal dimensions, these proximity-based methods achieve efficient compression while preserving local coherence.

Similarity-based Merging. While proximity heuristics provide strong inductive bias, semantic redundancy often transcends geometric or temporal adjacency, focusing explicit feature-space similarity. Global similarity methods compute token importance via patch-to-class correlation [79] or cluster semantically similar patches into abstracted representations [85], enabling merging of spatially distant yet semantically related regions.

Cross-modal merging methods leverage textual context to refine token merging decisions. This can be achieved through bidirectional tokens that exchange language-aware signals between modalities [82] or through pipelines that combine semantic and spatial similarity [84]. By prioritizing semantic relationships over spatial proximity, these methods enable compression that adapts to content meaning rather than token positions.

Hybrid Strategies. Combining multiple compression techniques can achieve better efficiency-quality trade-offs than individual methods alone. Sequential approaches [86] first apply attention-based pruning to remove coarse-grained redundancy, then use weighted merging to recover information from discarded tokens and integrate it into retained ones. Learnable abstraction methods [37] employ a small set of trainable compressed tokens while maintaining cross-attention with high-resolution lookup tokens for fine-grained details, allowing flexible compression ratios without architectural changes. These hybrid strategies show that pruning, merging, and learnable abstraction work synergistically when properly combined.

Multi-Scale Visual Compression. Single-scale compression methods operate at fixed granularity, struggling to obtain comprehensive visual details. Multi-scale approaches address this limitation by coordinating compression across layers, encoders, or resolutions, enabling flexible exploitation of hierarchical visual semantics.

Multi-Layer Compression. While most MLLMs extract visual features from the penultimate ViT layer, aggregating multi-layer features complements high-level visual semantics with low-level visual details. LLaVA-STF [76] extracts tokens from multiple ViT blocks, fusing them via channel concatenation and convolutions to combine spatial and semantic information across layers. METEOR [158] applies hierarchical pruning, using token-to-average similarity in shallow layers and CLS-to-token attention in deep layers for layer-adaptive compression. Chat-UniVi [81] employs three-level cascade aggregation, progressively extracting coarse, medium, and fine-grained token sets for unified multi-scale representation. LaCo [75] performs aggressive early-layer compression followed by pixel shuffle and MLP-based detail recovery.

Multi-Encoder Compression. Combining vision encoders with different architectures or training paradigms yields complementary representations. Cambrian-1 [7] demonstrates that integrating self-supervised models (e.g., DINOv2 [194]) with language-supervised encoders (e.g., CLIP [195]) consistently improves performance on vision-centric and OCR tasks, underscoring the value of diverse visual representations. METEOR [158] proposes a systematic multi-encoder framework that eliminates cross-encoder redundancy to maximize complementarity while minimizing computational overhead.

Multi-Resolution Compression. Processing inputs at multiple resolutions balances efficiency with visual detail preservation. High-resolution inputs capture fine-grained information for vision-sensitive tasks, while low-resolution inputs provide efficient global context. FastVLM [77] achieves optimal token-resolution balance through a novel hybrid vision encoder called FastViTHD. ADMIRE [89] employs dual-path Multi-Resolution Adaptation, comprising a low-resolution backbone for global processing and a high-resolution bypass for detail injection, excelling at document understanding and small object detection with minimal overhead.

For video understanding, LinVT [88] and M3 [91] apply multi-scale temporal pooling to capture both short-term dynamics and long-term context across different timescales. VideoChat-Flash [180] introduces Hierarchical Condensation (HiCo), progressively refining video semantics from clip-level to segment-level through selective filtering and backfill.

3.1.2 Outside-Encoder Compression

Outside-encoder compression occurs after vision encoder output but before the projector. At this stage, visual tokens are encoded but not yet aligned with the language modality. This position offers stronger plug-and-play capability than inside-encoder approaches, requiring no modification to encoder layers. Compression here reduces visual token count by measuring semantic relevance between vision-vision or vision-text representations. We categorize methods into purely-vision and text-guided compression.

Purely-Vision Compression. Purely-vision methods downsample or aggregate encoder outputs based solely on vision-vision semantic relevance, independent of user queries or prompts. A widely adopted paradigm is "selection-then-merge". VisionZip [93] identifies reusable tokens through importance estimation and representativeness constraints. Fourier-VLM [97] suppresses high-frequency redundancy via low-pass filtering in the frequency domain before mapping back to token space. LLaVA-STF [76] generates compact visual summaries through cross-layer concatenation and Multi-Block Token Fusion (MBTF).

Visual Attention Bias Problem. Early works such as LLaVA-PruMerge [84], VTC-CLS [157], and FasterVLM [196]

leverage the CLS token for patch attention and representation similarity-based sparsification. Similarly, FoPru [94] and freePruner [95] calculate token contribution via self-attention scores, selecting high-contribution tokens as pivots. However, recent works [71], [96], [133] reveal that attention-based selection exhibits bias toward salient regions (e.g., foreground objects), neglecting global context. HoloV [96] addresses this by incorporating global visual context to balance foreground and background tokens from a holistic perspective.

Extreme Compression. For long videos, LLaMA-VID [98] compresses each frame into a single Content Token, providing fixed-budget compression. Flash-VStream [197] employs K-means clustering of low-resolution features as Context Synopsis Memory to retain global temporal information. VideoLLaMA 2 [99] integrates frame-level patches via Spatial-Temporal Convolution (STC) with separable convolution and local aggregation. LLaVA-PruMerge [84] performs learnable token merging via nearest-neighbor clustering, maintaining near-uncompressed performance under 10x compression.

These methods share a common principle: enhancing per-token information density without text reliance, demonstrating particular advantages in multi-image and multi-turn scenarios.

Text-Guided Compression. When textual prompts provide semantic priors, compression can focus on question-relevant regions or frames, realizing context-oriented efficiency. PAR [100] parses queries into entities and actions and re-weights visual tokens accordingly. QG-VTC [101] computes question-to-vision similarity to guide token retention, enabling from 4× to 8× compression with minimal performance loss. LongVU [56] integrates cross-modal queries with frame or region candidates, first filtering at the segment level and then refining token-level selection.

Text-guided compression methods demonstrate particular robustness at the Outside-VE position: visual token semantics are fully encoded while cross-modal interaction has not yet begun, minimizing textual bias interference with low-level visual encoding. These methods often cascade with purely-vision approaches: first applying text-agnostic compression, then refining based on query relevance, achieving both stronger generalization and higher effective compression rates.

Token Recovery Mechanisms. Under aggressive compression, dynamic recovery mechanisms enable closed-loop refinement for enhanced robustness. When MLLMs detect semantic uncertainty by confidence or entropy, they can trigger resampling visual information, reinjecting tokens to compensate for missing visual evidence. Recoverable-Compression [102] triggers targeted resampling based on confidence and conflict thresholds post-compression. MustDrop [92] integrates recovery throughout a multi-stage pipeline via uncertainty gating, balancing aggressive reduction with stability. Beyond runtime recovery, ToCom [198] addresses train-test compression mismatches as a plug-and-play layer. It bridges performance gaps across compression ratios without retraining.

VTC [103] and Video-XL-Pro [199] optimize compression via visual reconstruction supervision. For example, VTC employs a Stable Diffusion [200] decoder to reconstruct images from completed tokens, using reconstruction error to recover missing visual information.

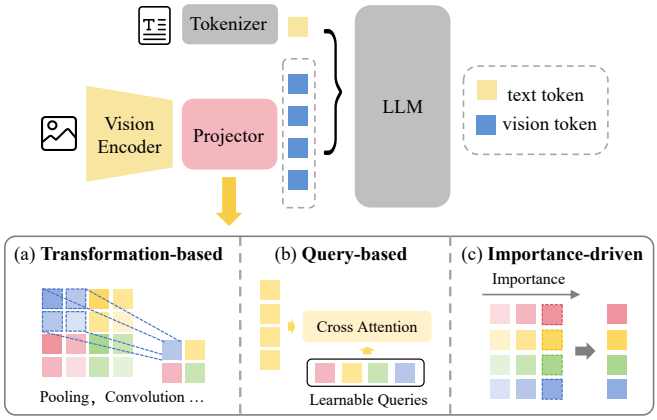

Fig. 4: Illustration of token compression strategies applied at the projector module in MLLMs.

3.2 Token Compression in Projector

The projector module plays a pivotal role in bridging the vision encoder and the language model in MLLMs. It acts as the interface that transforms raw visual embeddings into language-compatible representations, ensuring that the information extracted by the vision backbone can be effectively leveraged by the LLM. While projector architectures such as Q-Former [115] inherently perform token compression by distilling a large set of visual embeddings into a compact set of query tokens, subsequent research [104], [105], [108], [113] has introduced additional design enhancements to the projector to enable more fine-grained and task-adaptive compression. Therefore, in this section, we focus on token compression within the projector, referring to methods that operate on the visual features produced by the vision encoder before they are fed into the language model. As Figure 4 shows, these approaches can be broadly categorized into three main types: transformation-based (Sec. 3.2.1), query-based (Sec. 3.2.2), and importance-driven (Sec. 3.2.3).

3.2.1 *Transformation-Based Compression*

Transformation-based token compression methods reduce the number of visual tokens by directly transforming the spatial structure of visual feature maps. Instead of relying on learnable queries or complex attention mechanisms, these approaches perform lightweight, deterministic transformations to achieve token reduction while preserving essential visual information. In this section, we review representative transformation-based techniques, including pooling-based, pixel shuffle-based, and convolution-based methods.

Pooling-Based. Pooling is a widely used downsampling operation in computer vision, which can directly and effectively reduce the number of tokens while preserving the main semantic information.

Given an input feature map $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$, a pooling window of size $k \times k$, and an output feature map

$\mathbf{Y} \in \mathbb{R}^{H' \times W' \times C}$, the average pooled feature at spatial position (i, j) in channel c is defined as:

$$Y_{i,j,c} = \frac{1}{|\Omega_{i,j}|} \sum_{(u,v) \in \Omega_{i,j}} X_{u,v,c}, \tag{8}$$

where $\Omega_{i,j}$ denotes the set of spatial locations within the $k \times k$ neighborhood centered at (i, j).

Owing to its parameter-free nature and computational efficiency, pooling has been widely employed in many token compression approaches [104]–[108]. MobileVLM V2 [104] proposes the Lightweight Downsample Projector (LDP), which performs a simple 2×2 average pooling to effectively reduce the number of image tokens. DeCo [105] validates the effectiveness of the adaptive average pooling through extensive experimental analysis, showing that it not only facilitates stable and efficient convergence but also effectively extracts visual features. Following this line of pooling-based compression, AVG-LLaVA [106] proposes the Visual Granularity Scaler, which constructs multi-granularity visual features by stacking average pooling layers and employs the Visual Granularity Router to select the most appropriate granularity.

For video-focused models, pooling also serves as a simple yet effective way to reduce the number of tokens. TC-LLaVA [107] employs simple global average pooling to reduce the number of tokens per frame, while PLLaVA [108] applies adaptive average pooling across both spatial and temporal dimensions.

Pixel Shuffle-Based. Pixel shuffle is a method that trades token count for channel dimensionality, rearranging high-resolution spatial tokens into fewer tokens with increased channel depth.

Given an input feature map $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ and a downsampling factor r, this approach rearranges the spatial resolution into the channel dimension as:

$$\begin{aligned} \mathbf{Y} &= \mathrm{PixelShuffle}(\mathbf{X}, r) \\ &= \mathrm{reshape}\big(\mathbf{X},\, H/r,\, W/r,\, C \cdot r^2\big), \end{aligned} \tag{9}$$

where $\mathbf{Y} \in \mathbb{R}^{\frac{H}{r} \times \frac{W}{r} \times (Cr^2)}$.

This operation reduces the spatial token count by a factor of r^2 while increasing the channel dimension accordingly, thus effectively trading token number for richer per-token channel representation. An additional module, typically an MLP, is then applied to align the expanded channel dimension with the embedding dimension required by the LLM. Such a token compression strategy has also been employed in many well-known models, including InternVL 1.5 [109] and NVLM [201].

Convolution-Based. Compared with parameter-free methods such as pooling or pixel shuffle, convolutions selectively integrate local information through learnable weights rather than merely taking the mean or maximum, thus preserving more task-relevant details. By stacking convolutional layers or using variable kernel sizes, the model can also capture multi-scale abstract features, offering greater flexibility than simple pooling.

Formally, a 2D convolution that maps an input feature map $\mathbf{X} \in \mathbb{R}^{H \times W \times C_{\text{in}}}$ to an output feature map $\mathbf{Y} \in \mathbb{R}^{H' \times W' \times C_{\text{out}}}$ can be defined as:

$$\mathbf{Y}^{(o)}_{i,j} = \sum_{c=1}^{C_{\text{in}}} \sum_{m=1}^{k_h} \sum_{n=1}^{k_w} \mathbf{W}^{(o)}_{m,n,c} \cdot \mathbf{X}^{(c)}_{i+m-1,\, j+n-1} + b^{(o)}, \tag{10}$$

where $\mathbf{W} \in \mathbb{R}^{k_h \times k_w \times C_{\text{in}} \times C_{\text{out}}}$ are the learnable convolutional kernels, k_h and k_w denote the kernel height and width, and $b^{(o)}$ is the bias term for the o-th output channel.

In token compression methods, convolution is often combined with other operations such as average pooling. For example, the C-Abstractor proposed in Honeybee [113] integrates convolution with average pooling to achieve improved local context modeling. Similarly, MobileVLM V2 [104] employs an LDP that combines pointwise and depthwise convolutions with average pooling.

3.2.2 Query-Based Compression

Query-based token compression leverages a limited number of learnable query embeddings to attend to dense visual features and distill them into a compact representation for the subsequent processing. This paradigm provides a flexible and parameter-efficient alternative to purely transformation-based methods, as the queries can adaptively select task-relevant information while discarding redundancy. In the following, we discuss the canonical Q-Former framework, explore its enhanced and simplified variants, and introduce other cross-attention–based token compression approaches.

Q-Former. Q-Former, introduced in BLIP-2 [115], is a lightweight Transformer designed for query-based token compression and vision–language alignment.

Q-Former employs a small set of learnable query vectors that interact with frozen visual features via stacked self-attention and cross-attention layers. In this mechanism, the queries (*Q*) are trainable embeddings initialized as a small set of tokens that aim to retrieve task-relevant information; the keys and values (*K/V*) are the fixed output features from the frozen vision encoder (e.g., patch embeddings of the image). Through this process, the queries selectively aggregate task-relevant visual information into a compact set of embeddings, which are then linearly projected into the language embedding space and fed to the LLM as visual tokens. Building upon this principle, Q-Former efficiently compresses hundreds of visual tokens into only a few while preserving essential semantics, providing a parameter-efficient and highly adaptable bridge between vision encoders and LLMs. This design not only enables effective multimodal understanding but has also been widely adopted and extended in subsequent works such as MiniGPT-4 [116] and InstructBLIP [117].

Variants of Q-Former. Some later works [113], [118]–[120], [181] have proposed simplified and enhanced variants of the Q-Former architecture. For example, Qwen-VL [118] adopts a single-layer cross-attention module, reducing architectural complexity while retaining the ability to aggregate visual information and perform token compression. Honeybee [113] further observes that the conventional Q-Former may lead to the loss of fine-grained spatial information. To address this issue, it introduces two locality-enhanced projectors:

C-Abstractor and D-Abstractor. The C-Abstractor combines ResNet blocks with average pooling to perform downsampling while preserving local structures, whereas the D-Abstractor leverages the idea of Deformable Attention [202], using reference points and sampling offsets to enhance locality while maintaining flexibility in the number of output tokens. MQT [119] proposes a variant of the Q-Former architecture that allows a variable number of query tokens. Specifically, given M query tokens, MQT randomly samples the first m ($m < M$) tokens during training, enabling the model to learn visual representations at varying granularities. On average, this strategy reduces the number of visual tokens by about half compared with the original Q-Former, while maintaining effective information compression. TG-LLaVA [120] emphasizes the role of textual instructions in guiding key visual feature extraction. It introduces learnable latent embeddings to encode global text semantics and employs a single-layer Q-Former to integrate textual and visual information. The resulting mask is applied to the visual features, refining them under text guidance. Considering that using a standalone Q-Former may still lead to the loss of visual information, LLaVA-Mini [181] introduces an additional Modality-Pre Fusion module, which fuses visual representations with the instruction tokens before feeding them into the LLM, thereby mitigating such information loss.

Cross-Attention-Based. Instead of relying solely on the Q-Former's compressed token representations, some methods [203] utilize the cross-attention mechanism to identify or extract task-relevant tokens.

CATP [203] performs voting based on the cross-attention probabilities between query tokens and image tokens, accumulates the scores across different layers and heads, and prunes tokens according to their aggregated importance.

Several works [122]–[124], [176] move away from relying on learnable queries for token compression. TokenPacker [122] employs a coarse-to-fine visual information extraction strategy. It first downsamples the original visual features to obtain low-resolution representations that act as point-based queries. These queries are then paired with their corresponding regions in the high-resolution features to form point–region pairs, which iteratively interact through Point-to-Region cross-attention, progressively injecting rich visual information. Similarly, HiRes-LLaVA [123] abandons learnable queries and leverages downsampled visual features as queries that interact with the original visual features via cross-attention, resulting in a compact, compressed sequence. mPLUG-DocOwl2 [124] uses global visual features as queries and the cropped image features as keys and values, performing cross-attention to aggregate text semantics while significantly reducing the number of visual tokens for high-resolution images. QueCC [176] further injects the textual features of the user query into the visual representations, enabling subsequent queries to carry task-specific semantics. Combined with the cross-attention between the downsampled tokens and their respective visual token regions, it achieves extreme visual token compression while maintaining strong relevance to the textual task.

Other studies argue that relying solely on visual features is insufficient to accurately identify the most informative tokens. To address this limitation, they introduce cross-modal attention mechanisms, enabling more precise identification and extraction of task-relevant information. AdaFV [204] proposes a self-adaptive cross-modality attention mixture mechanism that dynamically selects visual tokens based on visual saliency and text–image similarity. VCM [186] introduces the concept of Vision Concept Modeling, which dynamically determines the number and spatial locations of required visual concepts according to a given instruction. It employs a multi-head cross-attention layer as a key component for semantic alignment in keyword selection, aligning visual features with training signals to guide subsequent token retention or aggregation. Based on the number and relevance of the selected keywords, VCM further estimates the optimal number of tokens to retain.

3.2.3 *Importance-Driven Compression*

Importance-driven token compression refers to methods that reduce visual token redundancy by estimating the importance of each token and selectively retaining the most valuable ones. Rather than relying on fixed-length queries or simple pooling, these approaches identify the relative importance of tokens and selectively prune or merge less informative ones. Existing strategies include similarity-based methods, attention-based methods, saliency-based methods, and innovative metrics-based methods, which will be discussed in detail below. This perspective highlights how importance estimation shapes the trade-off between efficiency and information preservation in MLLMs.

Various Similarity Metrics. There exist various approaches to measuring token similarity. DynTok [125] introduces a dynamic token compression method based on local token similarity. Its core idea is to exploit the varying information density of image patches across video frames: DynTok adaptively groups visual tokens and merges them within each group, thereby preserving more tokens in high-information-density regions while achieving higher compression ratios in less informative areas. Experiments show that computing cosine similarity on CLIP-generated visual representations yields better performance than directly measuring similarity in the LLM embedding space. LLaVA-Scissor [126] proposes Semantic Connected Components (SCC), reframing token compression as a graph connected components partitioning task. By explicitly covering all semantic regions, this method alleviates the common bias of attention-based approaches, which often overemphasize only the most salient objects. In SCC, token similarity is likewise measured using the cosine similarity of visual embeddings.

Saliency-Based. SeqCompression [127] conducts a comparative study between saliency-based and importance-agnostic token compression strategies, demonstrating that the saliency-based "Cluster and Aggregate" method offers clear performance gains. Specifically, after the vision encoder and projector, visual tokens are clustered with K-means++ according to embedding similarity, and tokens within each cluster are subsequently merged into a single representative token by averaging their embedding vectors.

Innovative Metrics-Based. Some studies [128] depart from the common attention-based similarity measures and

instead propose novel definitions and designs for token importance, similarity, or diversity. For example, DivPrune [128] formulates token pruning as a Max–Min Diversity Problem (MMDP), aiming to construct a token subset with the maximum minimum distance based on the original token set.

3.3 Token Compression in LLM

Currently, the mainstream architectures for MLLMs typically follow a classic design wherein visual information, after being processed by a vision encoder and a projector, generates a large number of vision tokens. Given that the LLM component generally contains significantly more parameters than the vision encoder and projector, the resulting sequence incurs substantial computational overhead when forwarded through the LLM.

To address this issue, a growing body of research has focused on reducing token redundancy within the LLM component. These methods can be broadly categorized based on the generation stage at which token reduction is applied, as illustrated in Figure 5. The first category (§3.3.1) performs token compression during the prefilling stage, i.e., it reduces the number of vision tokens at the first forward pass of the sequence through the LLM. This approach was primarily motivated by early use cases such as short-form visual question answering (VQA), where the cost of the prefilling stage dominates that of decoding. However, with the rapid advancement of chain-of-thought (CoT) and the increasing demand for long-form generation, attention has shifted to methods that apply token reduction during the decoding stage (§3.3.2). These techniques typically reduce the memory and computational cost by selectively pruning or merging parts of the key-value cache (KV cache), which proves especially beneficial for long-sequence generation tasks.

3.3.1 Compression in Prefilling Stage

The prefilling stage refers to the first forward pass of all tokens through the LLM. Once a vision token is removed in the shallow layers of the LLM, deeper layers are no longer able to access information from the corresponding image region. As a result, achieving significant acceleration during this stage while maintaining model performance is particularly challenging. Existing approaches often rely on observations about the inherent behavior of LLMs when processing tokens. By analyzing patterns in vision token redundancy, researchers have proposed four representative methods.

One common line of work involves **importance-based** approaches, which rank and retain vision tokens according to predefined metrics that estimate their significance. Another line involves **learnable module-based** approaches, where additional trainable components are used to determine token importance and the appropriate compression ratio. Both of these strategies primarily apply direct pruning, retaining only a subset of tokens without integrating information from the discarded ones. In contrast, **token merging-based** methods adopt a softer strategy by merging similar vision tokens to preserve information. Lastly, **fusion-based** approaches inject visual information through cross-attention or self-attention mechanisms, avoiding the need

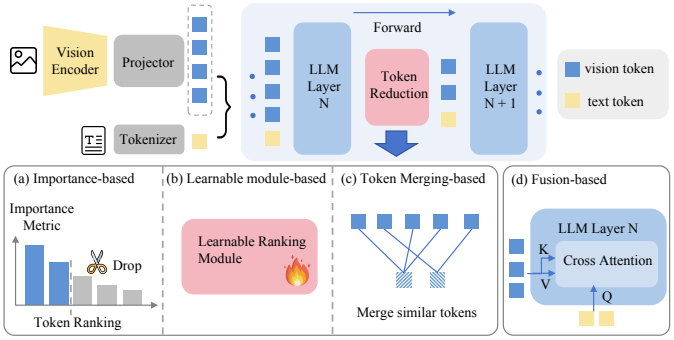

Fig. 5: Illustration of token compression strategies applied at the LLM module.

to concatenate all vision and text tokens into a single long sequence.

Importance-based This category of methods typically utilizes importance metrics to score vision token, followed by a ranking process to retain only the most important tokens. Among these metrics, the most commonly adopted is the attention from textual tokens to visual tokens, which helps preserve vision tokens that are relevant to the query. This allows for aggressive pruning of redundant visual information without significantly affecting model performance. FastV [205] was among the first to observe that vision tokens receive substantially lower attention scores compared to text tokens within the LLM, revealing the extreme sparsity in the information carried by vision tokens. Based on this observation, FastV prunes half of the vision tokens at the second layer of the LLM using the attention from the last textual token. PyramidDrop [206] further extended this line of analysis by identifying that the redundancy of vision tokens tends to increase with LLM depth. Leveraging this insight, it introduced a multi-stage progressive pruning strategy. Following these pioneering works, a number of subsequent studies, including [71], [134], [136], [137], [207]–[210], have adopted text-to-image attention ranking as a straightforward and effective approach, applying either single-stage or multi-stage pruning schemes. Beyond simple attention-based ranking, some recent efforts have focused on refining the evaluation of token importance [135], [188], [211], [212]. For instance, SparseVLM [132] and AdaptInfer [213] propose more fine-grained methods for selecting text tokens that are most relevant to visual content, using these to more accurately assess the significance of vision tokens. Other methods such as TransPrune [214] and VFlowOpt [214] combine attention scores with additional indicators, like information entropy maps, to improve the robustness of token importance estimation.

Additionally, VTW [131] takes a more radical approach by entirely removing vision tokens from certain layers of the model. In contrast, CrossMisalign [215] leverages vision-to-vision attention mechanisms to assess token importance, bypassing reliance on textual signals altogether.

Attention Bias Problem. As attention-based pruning methods have advanced, several inherent issues have emerged. One such issue is the attention bias observed during importance estimation. Feather [133] first noted that vision tokens located near output tokens in the input sequence tend to receive disproportionately high attention scores in the

shallow layers of the LLM. This phenomenon is attributed to the long-term decay property of Rotary Position Embedding (RoPE). To mitigate this, Feather proposes computing importance without applying RoPE to eliminate positional bias. AdaTP [216] addresses attention bias challenge by introducing a dedicated text encoder to compute cosine similarity between textual and visual features, thereby offering a more balanced measure of token importance. VScan [71], on the other hand, avoids the issue altogether by initiating pruning from the intermediate layers of the LLM, rather than at the shallow layers where attention bias is more pronounced.

Flash Attention Compatibility Problem. A technical challenge arises when integrating these attention-based pruning strategies with Flash Attention [217], which does not directly expose attention scores due to its design. Using standard attention mechanisms across all layers would degrade inference efficiency. A common solution involves applying Flash Attention at all layers, but selectively recomputing the queries, keys, values, and attention maps only at the specific layers where attention-based ranking is needed. While this solution mitigates the overhead for single-layer pruning, inference latency increases significantly if pruning is conducted at multiple layers. To address this more fundamentally, some studies have proposed alternative metrics that bypass the need for attention scores entirely [218]. TopV [138], for instance, ranks tokens using a combination of feature similarity, relative spatial distance, and absolute central distance. PACT [139] incorporates hidden state norms in conjunction with a global query vector to assess token importance. GreedyPrune [219] employs cosine similarity between text and vision tokens as a ranking criterion. It is also worth noting that CATP [121] takes into account the differences across layers, proposing a composite ranking method that combines semantic relevance with layer-wise attention variations to produce more robust token importance estimations.

Learnable Module-based Unlike importance-based methods that rely on predefined metrics to rank tokens, learnable module-based approaches introduce trainable components that learn to assess token importance or determine the appropriate compression ratio during training, thereby enabling dynamic compression. This paradigm was widely adopted in early Vision Transformer (ViT) research. For instance, DynamicViT [33] and AdaViT [220] attach lightweight decision networks to the ViT backbone as learnable modules, and employ Gumbel-Softmax [221] during training to render the framework fully differentiable. This design significantly improves the computational efficiency of ViTs.

In the MLLM domain, several works have adopted similar strategies [141]. In p-MoD [140], a weight predictor is proposed to assign importance scores to each token. Before each layer, tokens are sorted by their predicted weights, and only the top R% of tokens are retained for further processing. This allows the model to dynamically preserve informative visual tokens and skip less relevant ones in a flexible manner. Similarly, GlimpsePrune [143] utilizes a visual token importance predictor to estimate the significance of each token at a given layer, based on the attention scores of all visual tokens. Beyond predicting token importance, learnable modules have also been applied to estimate the compression rate of the entire sequence. For example, DyRate [142] incorporates a lightweight classifier to predict the optimal pruning ratio for each input sequence. ATP-LLaVA [177] employs a MLP with dual prediction heads to learn instance-specific thresholds for token pruning. This design enables adaptive token reduction during the generation process.

Token Merging-based In contrast to the previous two approaches that perform direct pruning by discarding less important tokens, token merging offers a softer compression strategy. Token merging techniques compute similarity measures and apply grouping or clustering algorithms to fuse multiple vision tokens into fewer representative ones, thereby achieving compression. Such methods were initially popularized in the context of accelerating Vision Transformers (ViTs). For example, ToMe [34] introduces a bipartite soft matching algorithm to perform efficient token merging based on pairwise similarity.

In the MLLM setting, LLaVolta [144] is one of the first works to apply token merging, using a simple and direct average pooling strategy to aggressively compress vision tokens. To mitigate the loss of performance caused by heavy compression, LLaVolta employs progressively lower compression ratios with multiple training stages. Subsequent methods have proposed more sophisticated designs in both similarity computation and clustering mechanisms. FiCoCo [86], for instance, first selects a subset of important tokens and then computes a correlation matrix between these preserved tokens and the remaining ones. The merging process is then guided by minimizing information loss based on this matrix. In CrossMisalign [215], token merging is used primarily as a visual information recovery mechanism. This method introduces a specialized recovery scheme that merges semantically redundant tokens with their most similar counterparts, based on a dot-product similarity calculated from reused attention key embeddings. In the video domain, to address inter-frame redundancy, FrameFusion [145] computes cosine similarity between each visual token and its spatially corresponding token from the preceding frame. This approach aims to minimize repetitive information across consecutive frames by merging tokens that represent similar spatial regions over time. Compared with such a sophisticated design, HoliTom [146] takes a relatively simple approach by directly merging those tokens with lower attention scores.

Fusion-based The previously discussed methods achieve compression by directly pruning or merging tokens, thereby shortening the overall sequence length. In contrast, fusion-based approaches implement compression indirectly by leveraging cross-attention or self-attention module to integrate visual information into other tokens, effectively avoiding excessively long input sequences. An early example is Flamingo [147], which introduced cross-attention layers called GATED XATTN-DENSE, between layers of a pretrained language model. In this setup, the original text tokens serve as queries, while visual features are treated as keys and values, enabling deep interaction between vision and language representations. Building on this idea,

mPLUG-Owl3 [148] adopts a similar architecture by combining intra-text self-attention with cross-modal attention between text and image features. This design leverages textual tokens as queries to selectively extract relevant visual information, eliminating the need to pass through a long sequence of visual tokens. More recently, CrossLMM [149] has advanced this direction further by introducing a design where compressed visual tokens and text tokens are used as queries, while the original long-sequence visual representations act as keys and values. It incorporates visual-to-visual and text-to-visual cross-attention, ensuring that the LLM can access high-resolution visual content while mitigating the performance degradation.

In addition to cross-attention-based fusion, another line of work explores extracting visual information via learnable tokens through self-attention. For example, VoCo-LLaMA [150] introduces a single Vision Compression token and modifies the attention mechanism such that textual tokens only attend to the VoCo token, effectively forcing the model to abstract visual information into this compressed representation. Following a similar philosophy, Victor [222] appends learned visual register tokens after the visual tokens, which absorb visual content through attention. All original visual tokens are subsequently discarded in deeper layers, thereby achieving efficient compression.

3.3.2 Compression in Decoding Stage

Compression in the Decoding Stage typically refers to KV-cache compression, which aims to reduce the memory and computational overhead of cached key and value tensors in transformers during autoregressive decoding. This is commonly achieved through pruning, quantization, or merging strategies, with the goal of preserving generation quality while improving efficiency. Due to the inherently long outputs produced by LLMs, there has long been a pressing need for KV-cache optimization. Consequently, a wide range of KV-cache compression techniques have been developed in the LLM domain, as seen in works such as StreamLLM [223], FastGen [224], and H2o [225].

In the multimodal setting, this challenge has become increasingly significant with the rise of multimodal chain-of-thought (CoT) reasoning. Output lengths have expanded from a few sentences to hundreds or even thousands of tokens, making both computational load and KV-cache memory consumption critical bottlenecks in the generation process. As a result, a growing number of studies have focused on KV-cache compression tailored for multimodal models [136], [141], [226]. One of the earliest works, LOOK-M [151], proposes using cumulative attention scores to estimate token importance. It preserves the KV pairs of the most recent window and additionally retains a proportion of visual KV pairs ranked by their importance. Another example, MustDrop [92], addresses both the prefilling and decoding stages. It stores only the KV pairs of visual tokens that are retained in the final layer during the prefilling stage. Recognizing that not all attention heads equally contribute to visual understanding, SparseMM [153] first identifies visual heads using an OCR-based task, then allocates more KV-cache budget to these heads. For non-visual heads, it adopts a more aggressive compression policy, striking a balance between performance and efficiency.

Promising progress has also been made in KV-cache compression in video domain, where the visual input is especially long and redundant. DyCoke [152] proposes a dynamic compression mechanism based on the text-vision attention. In each decoding step, only the KV pairs with high attention scores are retained. If the attention distribution shifts significantly in subsequent decoding steps, the KV cache is updated accordingly. Given the substantial redundancy present in video frames, Video-XL-2 [58] introduces a novel Bi-level KVs decoding. Based on the current query, the model dynamically selects whether to retrieve from dense or sparse KV representations, allowing it to discard a large number of query-irrelevant KV pairs. In video streaming scenarios, several specialized strategies have been proposed. LiveVLM [155], for instance, first discards KV pairs of unimportant visual tokens based on attention scores, then merges the original KVs of each frame into a single KV tuple. InfiniPot-V [154] explores the estimation of token importance by integrating two distinct evaluation criteria: Temporal-axis Redundancy (TaR) and Value Norm (VaN). These jointly guide the model to retain only the most critical tokens during KV-cache compression. In addition, StreamMem [156] implements KV-cache compression based on attention scores between visual tokens and generic queries. This is done while operating within a fixed-size KV memory to allow for efficient question answering.

3.4 Token Compression in Multi-Module

Beyond applying token compression within individual components such as the vision encoder (Sec. 3.1), the projector (Sec. 3.2), or the LLM (Sec. 3.3), an increasing number of recent approaches explore compression strategies across multiple modules to achieve higher compression efficiency and improved representational quality.

Since most multi-module token compression approaches are essentially built by combining the single-module techniques introduced earlier, we do not revisit their low-level technical details here. Instead, we focus on how these methods coordinate compression across different components and organize it as a multi-stage process to maximize overall efficiency and representational quality. In the following, we analyze two emerging design paradigms: **multi-module collaborative compression**, which emphasizes the joint and coordinated reduction of tokens across vision and language pathways, and **multi-stage progressive compression**, which structures token reduction as a progressive pipeline spanning early visual processing to late-stage LLM inference.

3.4.1 Collaborative Compression

CrossGET [82] is one of the earliest works to adopt multi-module token compression. It inserts CrossGET modules between the self-attention and FFN layers of both the visual and language branches, reducing the token count across layers. This design addresses the limitation of previous methods that required extracting visual information first and thus lacked text-guided supervision in early stages, enabling the earlier visual processing layers to be informed by subsequent textual features. LLaMA-VID [98] leverages cross-modal interaction between visual tokens and textual queries to extract task-relevant visual information and generate context tokens. It further applies pooling on the visual

features to obtain content tokens, enabling each video frame to be represented by only two tokens (a context token and a content token), which facilitates efficient understanding of long videos. PAR [100] provides a finer-grained analysis of visual token redundancy, categorizing it into external redundancy and internal redundancy. To address external redundancy, PAR removes task-irrelevant tokens through query rewriting, semantic clustering of visual tokens, and semantic retrieval. For internal redundancy, it introduces a token router mechanism that further eliminates redundant tokens by applying predefined similarity and redundancy thresholds.

3.4.2 Progressive Compression

To further improve the inference efficiency of MLLMs, several studies [86], [92], [152], [227] have proposed rigorous multi-stage token compression strategies that span multiple steps and phases of the inference process. MustDrop [92] adopts a multi-stage token compression strategy with carefully designed mechanisms across the vision encoding, prefilling, and decoding stages. It combines techniques such as merging highly similar spatial tokens, dual-attention filtering, and output-aware KV cache to achieve end-to-end acceleration throughout the entire inference pipeline. DyCoke [152] also employs a two-stage token compression strategy. In the first stage, it merges and removes tokens by computing the cosine similarity between corresponding tokens in adjacent video frames. In the second stage, it performs dynamic pruning within the KV cache, adaptively evaluating and retaining tokens based on their attention scores. FiCoCo [86] formulates token compression as a three-stage process: filter, correlate, and compress, addressing three key questions: which tokens to discard, where to preserve discarded information, and how to fuse remaining tokens while retaining critical information.

In summary, token compression in multi-module architectures represents a shift from isolated, single-stage reduction to a more holistic and system-level optimization of MLLMs. Rather than limiting compression to a single component such as the vision encoder, projector, or LLM, these approaches strategically integrate multiple stages of reduction — from early spatial downsampling and semantic clustering to query-guided selection and late-stage pruning — in order to maximize both efficiency and representational quality. This trend highlights an important direction for future research: optimizing token reduction as a coordinated, end-to-end process rather than a set of independent, module-specific techniques.

4 How to Select the Desirable Token Compression Strategy

The proliferation of token compression designs necessitates guidelines to help practitioners select optimal strategies for specific deployment scenarios. As Figure 6 illustrates, this section provides a comprehensive comparison of critical selection factors: (i) temporal-enhanced compression for video inputs, which focuses on unique challenges of processing long temporal sequences (§ 4.1); (ii) text-guided versus purely visual compression, examining the trade-offs between cross-modal guidance and visual-only approaches

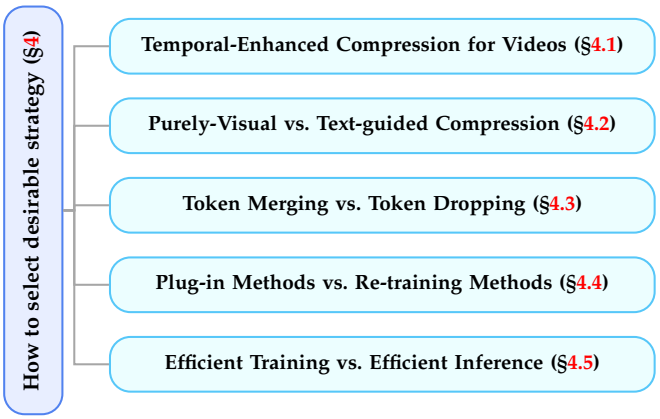

Fig. 6: Decision taxonomy for selecting an appropriate token compression strategy (see §4).

(§ 4.2); (iii) token merging versus token pruning, comparing fundamental compression paradigms (§ 4.3); (iv) plug-in versus retraining methods, weighing deployment flexibility against performance optimization (§ 4.4); and (v) efficient training versus efficient inference, distinguishing between optimization objectives (§ 4.5).

For each factor, we analyze underlying technical advantages and disadvantages, providing practical recommendations based on deployment constraints.

4.1 Temporal-Enhanced Compression for Video Input

Compared with static images, video input introduces an additional temporal dimension that substantially increases computational demands. As video duration grows or frame sampling rates rise, the number of visual tokens fed into language models increases explosively, creating a fundamental tension between inference efficiency and modeling fidelity. Although existing spatial compression strategies (refer to Sec 3.1) can be directly applied to individual frames, they often fail to exploit cross-frame redundancy. To address this gap, recent research has proposed *temporal-enhanced token compression* methods that explicitly consider temporal structure for efficient long-sequence modeling. Three central challenges emerge:

1) **Spatial-temporal interaction:** How to jointly compress across spatial (h, w) and temporal (t) dimensions to form compact yet expressive representations (§ 4.1.1).
2) **Temporal structure preservation:** How to retain spatiotemporal structure after compression for fine-grained perception tasks such as motion direction estimation [239], [240] and temporally grounded QA [9], [241] (§ 4.1.2).
3) **Scalability to extreme lengths:** How to design compression and memory mechanisms that scale to hour-long videos containing tens of thousands of frames (§ 4.1.3).

4.1.1 Spatial-Temporal Compression

Joint spatial-temporal compression strategies can be broadly divided into *fixed* and *dynamic* approaches, with hybrid strategies emerging at the intersection.

TABLE 2: Overview of temporal-enhanced compression strategies for video input (details refer to §4.1).

Category	Method	Key Idea	Representative Works
Fixed	**Pooling**	Average neighbor tokens across temporal dimension.	PLLaVA [108], Video-ChatGPT [228]
	Convolution	2D/3D convolutions for joint spatio-temporal downsampling.	VideoLLaMA2 [114], Qwen2-VL [25]
	Query-based	Learnable queries that attend over all video tokens (e.g., Q-former, Token Learner, Resampler).	Clapper [78], LinVT [88], CrossLMM [149]
	Sequential Models	Process tokens in temporal order with explicit timestamp embeddings and recurrent memory.	BLIP-3-Video [229], STORM [230]
Dynamic	**Token Merging**	Merge redundant tokens across frames	TESTA [83], AuroraCap [231], DyCoke [152]
	Token Dropping	Drop temporally low-saliency or redundant tokens	LongVU [56], TimeChat-Online [57]
Hybrid	**Global-Local Fusion**	Global event-level clustering with local frame-level aggregation.	LongVLM [232], Video-XL [55], HiCom [233], PruneVid [179], Chat-UniVi [81], FiLA-Video [80], TempMe [174], Quicksviewer [234]
	Slow-Fast Pathways	Two-Stream architecture: a high-resolution slow pathway for spatial detail and a low-resolution fast pathway for motion dynamics.	SlowFast-LLaVA [235], LLaVA-Video [236], Clapper [78], Keye-VL-1.5 [237]
	Memory-bank	Long-term memory complemented by a short-term memory.	MovieChat [173], VidCompress [238], Flash-VStream [197]

Fixed Temporal Compression. Fixed strategies reduce the number of tokens per frame or clip from N to a predefined M ($M \ll N$). Early Video-LLMs commonly adopted uniform frame sampling or downsampling to bound token budgets. Pooling-based designs (e.g., PLLaVA [108], Video-ChatGPT [228], TC-LLaVA [107]) average patches across adjacent frames to suppress redundancy, but often at the cost of motion detail. Convolution-based designs integrate temporal information more explicitly: VideoLLaMA2 introduces a 3D spatio-temporal convolution (STC Connector) combined with RegStage to preserve local dynamics under reduced cost, while Qwen2-VL [25] applies 2D convolutions to fuse adjacent-frame features. To enhance temporal embedding, Qwen2.5-VL [4] adopts 3D convolutional module to downsample both spatially (4x) and temporally (2x).

Query-based designs represent another reserach line. Instead of pooling all tokens, they learn a compact set of *query tokens* (e.g., Q-former, Resampler, Token Learner) that aggregate salient information through attention. For instance, Clapper's TimePerceiver applies cross-attention to capture inter-frame dynamics, while LinVT and CrossLMM [149] leverage user queries to guide compression, producing lightweight yet semantically aligned representations.

Sequential models leverage linear complexity $O(n)$ to efficiently encode long video token sequences by first enhancing temporal modeling before compressing token counts. BLIP-3-Video [229] proposes a Grouped Sequential Model that processes video tokens in temporal order while grouping them by spatial location. Each group maintains independent temporal memory augmented with timestamp positional encodings at each update step, ultimately aggregating into merely 16-32 video-level tokens via attention mechanisms. Through systematic comparison against various fixed compression methods, BLIP-3-Video demonstrates that its Grouped Sequential Model outperforms traditional pooling and attentional pooling by preserving absolute temporal order alongside semantic completeness. Similarly, STORM [230] leverages the Mamba State Space Model [242] to integrate temporal information, employing bidirectional scanning to simultaneously capture spatial and temporal dependencies, yielding token representations enriched with historical context.

Dynamic Temporal Compression. In contrast to fixed-ratio compression that uniformly compresses videos of varying information density into identical token counts, dynamic compression methods adaptively adjust the number of retained tokens based on video content, enabling differentiated modeling between static and dynamic segments.

Temporal Token Merging. TESTA [83], AuroraCap [231], and DyCoke [152] merge similar or redundant tokens across frames, typically identifying merge candidates via token similarity. Building upon this foundation, learnable merging strategies have emerged. InTI [243] introduces a lightweight weight prediction network that generates dynamic weights for spatially co-located tokens in adjacent frames, enabling more adaptive fusion. Similarly, Learnable VTM [244] assigns learnable saliency scores to each token, supporting dynamic merge ratios that substantially reduce token counts while preserving critical information.

Temporal Token Pruning. Unlike merging, pruning directly discards less important tokens rather than fusing them [56], [57], [245]. LongVU [56] proposes a three-stage compression pipeline where the final temporal-dependency-based spatial token pruning uses the first frame as an anchor within each sliding window, computing cosine similarity between spatially aligned tokens across frames and discarding highly similar ones to achieve extreme spatial compression. Similarly, TimeChat-Online [57] retains only temporally dynamic information by measuring redundancy between temporally adjacent, spatially co-located tokens and discarding redundant tokens in subsequent frames. This work also demonstrates that feature-level redundancy measures outperform pixel-level approaches.

Hybrid Strategies combine multiple principles to balance global coverage with local detail. A prevalent design is *global-local fusion*, which clusters video segments into key events and then performs intra-event aggregation, thereby capturing both coarse event structure and fine-grained dynamics. Representative works include PruneVid [179], Chat-UniVi [81], and FiLA-Video [80]. LongVLM [232] combines

local token merging within clips with global semantic representations across all frames. TempMe [174] and Video-XL [55] employ hierarchical merging or visual summarization tokens (VSTs) to reduce redundancy while preserving temporal context. HiCom [233] groups sampled frames in the spatiotemporal domain and performs instruction-conditioned compression, whereas Quicksviewer [234] uses Gumbel-Softmax to determine information density and performs block-wise resampling to reduce irrelevant redundancy.

Slow-fast dual streams. Inspired by action recognition, SlowFast-LLaVA [235], LLaVA-Video [236], and Clapper process video through two pathways: a slow pathway with low frame rate but high spatial detail, and a fast pathway with high frame rate but compact tokens. Keye-VL 1.5 [237] further refines this by dynamically routing salient frames to the slow branch while assigning static frames to the fast branch, significantly improving token efficiency.

Memory-bank mechanisms. Flash-VStream [197] introduces STAR memory, consisting of (i) a Context Synopsis Memory that clusters low-resolution features into centroids to preserve global temporal trends, and (ii) a Detail Augmentation Memory that selectively retains high-resolution tokens for keyframes. This design offers flexible token budgets while balancing coverage and detail. Similarly, MovieChat [173] combines sliding windows with long-term and short-term memory, periodically merging tokens when capacity is exceeded, while VidCompress [238] enhances this approach with memory-augmented cross-clip attention.

4.1.2 Temporal Structure Preservation

During video compression, atomic operations such as token merging and pruning can blur or discard the spatiotemporal positional information of visual tokens, thereby disrupting the original temporal structure of videos. This degradation impairs MLLMs' ability to perceive precise timestamp information, adversely affecting tasks that require absolute temporal localization, such as temporal grounding [9], [246]. To mitigate this issue, several works have introduced explicit time-aware mechanisms, which can be categorized into three main approaches: augmenting video tokens with temporal positional embeddings, incorporating dedicated temporal encoding modules within the overall architecture, and inserting special timestamp tokens.

Temporal Positional Embeddings. The most direct approach is to enrich visual tokens with temporal positional information. BLIP-3-Video's Grouped Sequential Model [229] processes frames sequentially with timestamp positional encodings and grouped memory mechanisms, maintaining separate temporal memory across different token groups to preserve both local temporal details and absolute temporal order. TimeChat-Online [57] retains the original Video-RoPE [4], [247] positional encodings for important tokens selected based on visual dynamics, thereby preserving spatiotemporal information relative to the original video even after pruning operations. PVC [178] also employs relative timestamps to indicate video frames and obtains temporal embeddings via MLP. Specifically, it uses either absolute positional embedding $t = [0, 1, \ldots, T]$ or relative positional embedding as follows:

$$t = \left[0, \frac{1}{T-1}, \ldots, \frac{T-2}{T-1}, 1\right] \tag{11}$$

Temporal Encoding Modules. Beyond positional embeddings, dedicated architectural components can explicitly model temporal dependencies. STORM [230] leverages Mamba-based [248] state-space layers (MambaMixer) that inject temporal awareness through bidirectional scanning of token sequences, simultaneously capturing both spatial and temporal dependencies. PVC [178] adopts a progressive encoding strategy where tokens of each frame are sequentially encoded and adaptively compressed to supplement information not extracted from previous frames, ensuring cumulative temporal context preservation.

Special Timestamp Tokens. An alternative strategy introduces explicit timestamp representations as separate tokens. Video-XL-2 [58] interleaves timestamp tokens within the visual token sequence to enhance temporal awareness throughout the model. Qwen3-VL [249] advances this approach by adopting a textual token-based time encoding strategy [193], [250], [251], wherein each video temporal patch is prefixed with a timestamp expressed as a formatted text string (e.g., `<3.0 seconds>`), moving beyond traditional Video-RoPE to achieve precise, timestamp-grounded event localization for stronger video temporal modeling.

4.1.3 Extreme-Long Video Compression

In hour-long video scenarios, MLLMs must process thousands of frames, posing severe challenges to computational efficiency and memory management. Addressing these challenges necessitates specialized designs across multiple dimensions, including input sampling, encoding compression, memory storage, and inference acceleration.

Early explorations into long video understanding primarily focused on memory bank-based approaches to store long-term temporal semantics. MovieChat [173] pioneered the integration of sliding windows with dual-memory mechanisms, where short-term memory captures fine-grained details within the current window while long-term memory aggregates global semantics from historical segments, enabling processing over 10,000 frames on a 24GB GPU. Similarly, FlashVStream [252] proposes a more elaborate flash memory architecture to achieve real-time responses to user queries.

Beyond the memory-bank foundations, the Video-XL series demonstrates a clear evolution from adaptive compression to comprehensive optimization. Video-XL [55] introduces dynamic interval partitioning that assigns varying numbers of Visual Summarization Tokens (VSTs) to compress visual semantics. It enables processing 2,048 frames with near-lossless 16x compression and 95% accuracy at 32x compression. Video-XL-Pro [199] advances this by introducing reconstructive capability through the ReCoT framework, which synthesizes dynamic tokens to capture motion patterns, employs semantic-guided masking to focus on dense regions, and incorporates query-aware selection to prune low-relevance tokens. This enables handling over 8,000 frames with near 99% accuracy. While these efforts optimize training-time compression, Video-XL-2 [58] shifts

TABLE 3: Comparison between Purely-Visual and Text-Guided token compression strategies with representative works (details refer to §4.2).

	Purely-Visual	Text-Guided
Method	Retain informative visual tokens according to inherent vision redundancy	Select text semantic aligned visual tokens according to textual instruction or query
Features	(i) Suitable for multi-turn dialogues, streaming video understanding, visual captioning, (ii) Easy to deployment	Suitable for single-turn dialogues, long VideoQA, high-ratio compression scenario, visual grounding
Works	DeCo [105], VisionZip [93], DART [183], HoloV [96], TimeChat-Online [57]	FastV [129], SparseVLM [132], Q-Former [257], QueCC [176], PyramidDrop [130], LLaVA-Mini [181]

focus to inference efficiency through KV cache sparsification, enabling processing over 10,000 frames on a single GPU.

Under the video question-answering scenario, only a subset of frames is typically relevant to a given question. This has motivated query-aware compression strategies. LinVT [88] identifies candidate regions through spatiotemporal saliency analysis, then filters and aggregates them according to the text query to ensure retained tokens capture both visual saliency and semantic relevance.

Beyond query-aware innovations, practical deployment demands system-level efficiency. Long-VMNet [253] employs a fixed-size memory bank (e.g., 5,880 tokens) enabling memory reuse across queries after a single video scan, requiring less than 1GB memory while supporting 10-hour videos. ReTaKe [254] detects keyframes via inter-frame distance peaks and marks them as pivots, while compressing non-pivot frames by pruning low-attention tokens in their KV cache. Leveraging LLM prior knowledge, it enables plug-and-play adaptation to existing VideoLLMs for processing 8x longer sequences. TimeViper [255] adopts a hybrid Mamba-Transformer architecture, combining linear complexity with precise attention to process over 10,000 frames.

Summary. Extreme-long video understanding exhibits multi-dimensional synergy: 1) adaptive key frame sampling [56], [193], [256] and adaptive partitioning reduce input redundancy; 2) multi-module collaboration enables progressive encoding compression; 3) query-aware strategies dynamically adjust based on user intent; 4) KV-cache sparsification improves inference efficiency. This evolution from isolated optimizations to systematic, task-aware, end-to-end design establishes the foundation for practical hour-long video understanding.

4.2 Purely-Visual vs. Text-guided Compression

As discussed in § 2.2, token compression in MLLMs aims to reduce two major types of redundancy: *intra-visual* (vision-to-vision) and *cross-modal* (text-to-vision). Accordingly, existing methods can be grouped into two branches depending on whether they leverage textual information, such as user instructions or questions, compared in Table 3.

Purely-visual Compression. These methods rely solely on visual cues to eliminate redundant information. They systematically reduce tokens representing duplicate objects, uniform backgrounds, or semantically equivalent regions. For video sequences, such approaches compress temporally static content while preserving dynamic motions. Specifically, recent studies identify distinctive visual tokens [57], [96], [183] or aggregate repetitive semantic tokens into compact representations [34], [83]. As VisionZip [93] pointed out, these more compact visual tokens lead to better visual representations.

Since purely-visual approaches are text-agnostic and perform one-time compression, they are efficient for low-latency applications such as multi-turn dialogue, online responses and streaming video understanding [258]–[262]. Moreover, their general applicability to visually rich scenes makes them effective for captioning [263], [264] tasks. For deployment, they directly reduce visual tokens before the LLM, avoiding extensive computation and memory consumption in the LLM's shallow layers and enabling seamless adaptation across different LLM architectures.

Text-Guided Compression. In contrast, text-guided strategies [148], [265] use cross-modal information to select text-relevant tokens according to a given instruction or query. Typical methods estimate text-to-vision attention or similarity or introduce proxy tokens for better cross-modal interaction [73], [82]. By only focusing on task-relevant visual semantics, these methods can achieve high compression ratios while maintaining accuracy in tasks such as visual question answering, grounding [9], [266], [267], and long-video reasoning [28], [29], [268]. However, since user queries always vary across turns, text-guided compression often requires re-encoding historical tokens, limiting efficiency and reusability in multi-turn dialogue settings.

Takeaway. Purely-visual and text-guided strategies are complementary. A practical design is to first derive compact visual representations via purely-visual compression and then apply text-guided selection within the language module to refine tokens relevant to the given textual query.

4.3 Token Merging vs. Token Dropping

Token merging and token dropping (also referred to as pruning) are two fundamental operations in the token compression paradigm. Their core distinction lies in the compression manner: *merging* is a *soft* strategy that aggregates less informative tokens into representative ones, while *dropping* is a *hard* operation that directly discards them. A natural question arises: *should these two operations be treated identically?* This subsection discusses their conceptual differences, selection mechanisms, and practical implications.

Merging or Dropping. Token merging and dropping each possess distinct advantages and drawbacks. As Table 4 summarizes, token merging maintains holistic semantics by smoothing token representations but may blur spatial or temporal locality. Token dropping, in contrast, preserves sparse and salient semantics yet risks losing fine-grained contextual information. Quantitative analyses from LLMC+ [270] reveal that for spatial redundancy, drop-based

TABLE 4: Comparison between token merging and token dropping strategies with representative works (details refer to §4.3).

	Token Merging	Token Dropping
Method	A *soft* strategy that aggregates visually redundant tokens into compact and representative embeddings.	A *hard* strategy that directly discards tokens considered less informative or task-irrelevant.
Pros	(i) Preserves holistic and fine-grained semantics, (ii) Suitable for compressing low-level visual features, (iii) Effective for spatial redundancy.	(i) Retains sparse and salient semantics (ii) Suitable for compressing high-level visual features.
Cons	May blur spatial or temporal locality due to averaging across multiple tokens.	May overlook subtle contextual cues that are removed during pruning.
Works	ToMe [34], TESTA [83], HoliTom [146], MustDrop [92]	VisPruner [68], DivPrune [128], MADTP [73], DART [183], FlexSelect [189], CDPruner [269], DTD [57]

strategies generally outperform merge-based ones in both the vision encoder and the LLM component.

Attention-based or Similarity-based strategies for token selection. Both token merging and dropping rely on identifying *"unimportant"* tokens to aggregate or discard. Early works primarily used attention scores as indicators of token importance. However, recent studies have exposed several limitations of attention-based selection. DART [183] and FEATHER [133] reported that attention scores introduce a positional bias, favoring tokens located at the lower-right region of the image—typically appearing later in the sequence—regardless of their semantic significance. HoloV [96] further highlighted that MLLMs often overfit to "highlighted tokens" and overlook holistic context, leading to local overemphasis on salient regions. Moreover, attention-based selection can be incompatible with Flash Attention implementations, reducing efficiency and sometimes even underperforming random reduction baselines. To address these issues, recent approaches increasingly adopt similarity-based token selection [57], [145], [271], where redundancy is measured via feature-level similarity rather than attention magnitude, enabling more stable and context-aware compression.

Takeaway. Merging and dropping are complementary rather than competing strategies [79], [84], [175], [272], [273]. Merging provides smooth aggregation suitable for dense or temporally redundant visual inputs, whereas dropping is preferable when sparse, high-level semantics suffice. Future frameworks may benefit from adaptive hybrid designs that dynamically switch between soft aggregation and hard pruning according to modality characteristics and redundancy types.

TABLE 5: Comparison between plug-in and re-training methods with representative works (details refer to §4.4).

	Plug-in	Re-training
Method	A *parameter-free* strategy that can be directly integrated into existing models without additional training.	A *trainable* strategy that requires additional training to obtain learnable compression capability.
Features	(i) Training-free and parameter-free, (ii) Lightweight and efficient for deployment, (iii) Performance degradation on fine-grained tasks.	(i) Higher performance ceiling, (ii)Require additional training, (iii) Limited transferability across models
Works	FastV [129], SparseVLM [132], PyramidDrop [130], MustDrop [92]	Honeybee [113], DeCo [105], TokenPacker [122], HiCo [180]

4.4 Plug-in Methods vs. Re-training Methods

From the perspective of model adaptation, existing token compression methods can be broadly categorized into two groups: plug-in methods, which can be seamlessly integrated into pre-trained models without the need for extra training, and re-training approaches, which require additional fine-tuning or end-to-end optimization. As illustrated in Tab. 5, although both aim to reduce token redundancy and accelerate inference, they differ markedly in design philosophy, deployment cost, and the level of performance they can ultimately achieve.

Plug-in Methods. Plug-in approaches focus on lightweight modules that require minimal or no training and can be seamlessly integrated into frozen backbones. Representative strategies include: (1) parameter-free spatial transformations, such as global or adaptive pooling employed in TC-LLaVA [107], PLLaVA [108], DeCo [105], and AVG-LLaVA [106]; (2) pixel rearrangement operations, exemplified by pixel shuffle and space-to-depth transformations in NVLM [201] and InternVL 1.5 [109]; (3) similarity-based token compression, where DynTok [125] dynamically groups video tokens and performs intra-group merging, LLaVA-Scissor [126] leverages Semantic Connected Components to preserve semantic regions while reducing redundancy, and DivPrune [128] selects informative tokens by maximizing diversity; and (4) inference-time KV cache compression, where DyCoke [152] prunes the KV cache guided by attention scores, and MustDrop [92] adopts an output-aware KV Cache policy to reduce memory consumption and accelerate decoding without backbone modification.

These methods are easy to deploy, highly compatible across models, and cost-efficient, making them ideal for rapid inference acceleration or flexible deployment. However, because they are often task-agnostic and rely on heuristics such as similarity thresholds or clustering, their semantic retention and performance upper bound can degrade under aggressive compression or complex task demands.

Re-training Methods. Re-training approaches introduce learnable modules or require end-to-end optimization, aim-

TABLE 6: Comparison between Efficient Training and Efficient Inference strategies (details refer to §4.5).

	Efficient Training	Efficient Inference
Method	Aim to mitigate training costs by reducing the number of image tokens during the forward process.	Aim to lower inference costs by performing token reduction during the prefill or decoding stage.
Features	(i) The methodological design is relatively simple, with a limited number of studies in this area. (ii) The validation cost is substantial.	(i) The methodological design is more diverse, with a greater body of research in this field. (ii) The validation cost is minimal.
Works	Flamingo [147], Q-Former [257], LLaVA-OneVision [10], Qwen2.5-VL [274], InternVL3.5 [52]	FastV [129], SparseVLM [132], PyramidDrop [130], VisionZip [93], SparseMM [153]

TABLE 7: Representative MLLMs and their efficiency-oriented training compression strategies.

	Representative MLLMs	Compression Strategy
2022	Flamingo [147]	GATED XATTN–DENSE
2023	BLIP-2 [115], mPLUG-Owl [275], Qwen-VL [118], Video-LLaMA [276], MiniGPT-4 [116]	Q-former and its variants
	Video-ChatGPT [228]	Temporal and Spatial Pooling
2024	PLLaVA [108], LongVLM [232], VideoLLaMA 2 [99]	Temporal and Spatial Pooling
	LLaVA-OneVision [10]	Bilinear Interpolation
	LLaVA-Video [236]	Average Spatial Pooling
2025	InternVL series [52], [109], Qwen2VL series [4], [112]	Pixel Shuffle
	Seed1.5-VL [277]	Average Pooling

ing for task-adaptive and semantically aware token compression. Representative methods include query-based designs, such as Q-Former and its variants, including the BLIP-2 [115], the simplified single-layer cross-attention in Qwen-VL [118], the C-/D-Abstractor modules in Honeybee [113] for better locality modeling, and MQT [119], which adapts the number of query tokens. Another line of work employs downsampled-as-query cross-attention, as seen in TokenPacker [122] and HiRes-LLaVA [123], which use downsampled features as queries to interact with high-resolution regions and achieve coarse-to-fine information injection. There are also text- and concept-guided compression methods. TG-LLaVA [120] performs text-driven masking, QueCC [176] incorporates user query semantics through local cross-attention aggregation, and VCM [186] models vision concepts to dynamically determine concept granularity and spatial alignment. Finally, several multi-module and multi-stage token compression frameworks have been proposed. CrossGET [82] breaks the sequential visual-first processing paradigm, LLaMA-VID [98] constructs context and content tokens for each video frame, PAR [100] differentiates between external and internal redundancy through query rewriting and token routing, and MustDrop [92] accelerates inference through a three-stage "vision-prefill-decoding" merging strategy with dual-attention filtering.

These methods typically achieve stronger semantic preservation and task relevance because they are able to leverage cross-modal attention, textual guidance, and progressive refinement. They also tend to reach higher compression ratios without causing severe performance degradation. However, their use comes with additional training costs, greater data requirements, and increased engineering complexity, and their effectiveness may vary depending on the target task or application domain.

Comparative Insights. Plug-in methods are well-suited for rapid deployment and inference acceleration when training resources are limited or when task requirements are relatively moderate. However, their performance upper bound is relatively limited. Recent studies such as FCoT-VL [182] further empirically demonstrate that current training-free token compression methods suffer from notable performance drops on tasks requiring high-resolution visual understanding and complex text reasoning. In contrast, re-training methods excel in task-specific scenarios and fine-grained multimodal understanding, offering higher performance ceilings and greater stability under aggressive compression, though at the cost of substantial additional training overhead and poor transferability across models.

In practice, hybrid strategies have gained increasing attention as a promising compromise between efficiency and adaptability. A common design is to apply lightweight plug-in techniques such as pooling or pixel unshuffle for early spatial reduction, then incorporate re-trained modules such as cross-attention or query-guided compression for semantic refinement, and finally adopt key-value cache pruning to improve decoding efficiency. This progressive integration, exemplified by the multi-stage design of MustDrop [92], reflects a trend toward combining the deployment flexibility of plug-in methods with the task adaptivity and performance advantages of re-training approaches.

4.5 Efficient Training vs. Efficient Inference

Efficient training and efficient inference respectively address the problem of token reduction during the training and inference phases. In this section, we focus on discussing the distinctions between these two approaches, compared in Table 6.

Efficient training typically aims to mitigate costs during pretraining and SFT, which demands hundreds of billions to trillions of tokens. In practice, most state-of-the-art multimodal models rely on relatively simple mechanisms for token reduction as presented in Table 7. LLaVA-OneVision [10] utilizes bilinear interpolation to reduce tokens per frame. The InternVL [278] and Qwen2 series [274] apply pixel shuffle strategies, reducing the number of vision tokens while expanding their feature dimensions. Similarly, Seed1.5-VL [277] employs a basic average pooling method. It is worth noting that a variety of new methods have recently been proposed to accelerate training. For example, LLaVolta [144] introduces staged training, where more

TABLE 8: Summary of benchmarks widely-used in visual token pruning studies. MQA denotes multiple-choice question answering, Open denotes open-ended question answering, Y/N denotes Yes/No question answering.

Benchmark	Answer Type	Metric	Num Examples	Focus	Data Link
Image Domain					
GQA-testdev-balanced	Open	Accuracy	12,578	General Image Perception	Link
VQA-v2-testdev	Open	Accuracy	107,394	General Image Perception	Link
VizWiz-val	Open	Accuracy	4,319	General Image Perception	Link
POPE	Y/N	F1-Score	3,000	General Image Perception	Link
TextVQA-val	Open	Accuracy	5,000	OCR	Link
ScienceQA-Image-test	MQA,Y/N	Accuracy	2,017	Knowledge	Link
MathVista-testmini	MQA,Open	Accuracy	1,000	Knowledge,Reasoning	Link
MathVerse-testmini	MQA,Open	Accuracy	3,940	Knowledge,Reasoning	Link
MMMU	MQA,Open	Accuracy	11,550	Knowledge,Reasoning	Link
MME	Y/N	Perception Score	2,374	Integrated	Link
MMBench-en-dev	MQA	Accuracy	4,329	Integrated	Link
MM-Vet	Open	GPT-Score	218	Integrated	Link
SeedBench-Image	MQA	Accuracy	14,280	Integrated	Link
LLaVA-Bench^{W}	Open	GPT-Score	60	Integrated	Link
Video Domain					
ActivityNet-QA-test	Open	Accuracy,GPT-Score	8,000	Integrated	Link
MVBench	MQA	Accuracy	4,000	Temporal Understanding	Link
EgoSchema	MQA	Accuracy	5,063	Long Video	Link
LongVideoBench-val	MQA	Accuracy	1,337	Long Video,Integrated	Link
MLVU-dev	MQA,Open	Accuracy,GPT-Score	2,593	Long Video,Integrated	Link
Next-QA-MC-test	MQA	Accuracy	8,564	Integrated	Link
Video-ChatGPT	Open	GPT-Score	3,493	Integrated	Link
Video-MME	MQA	Accuracy	2,700	Integrated	Link

aggressive token reduction is applied in the early stages and the compression ratio is gradually decreased over time. PyramidDrop [130] removes tokens layer by layer inside the LLM. From the perspective of task similarity, both LLM prefilling and training involve a single forward pass of a sequence through the LLM. Therefore, in principle, all strategies that can be applied during LLM prefilling could also be used for efficient training.

However, **why have these diverse methods not been widely adopted by mainstream LVLMs?** We identify three main reasons. First, *compatibility issues*: many prefilling acceleration methods are not compatible with Flash Attention, which directly affects training efficiency. Second, *validation cost*: adopting a new strategy requires validation during training, which is far more expensive than inference, making researchers more conservative. As long as current costs remain acceptable, new methods are unlikely to be adopted unless they prove to be a breakthrough. Third, *inductive bias*: existing compression techniques often design customized strategies based on observations from certain tasks or benchmarks, thereby introducing strong inductive bias. Such methods may lead to performance degradation in scenarios where visual information is denser or task distributions differ significantly. Since current MLLMs are intended for general-purpose use, any degradation in certain capabilities is unacceptable.

In the field of efficient inference, nearly all of the aforementioned methods are designed for this scenario, and the area is evolving rapidly. The popularity of this topic is largely due to its low exploration cost. Moreover, for the practical deployment of large multimodal models, the volume of API requests is extremely high, and minimizing latency is crucial. As a result, controlling inference cost has become an urgent requirement.

5 Benchmarks and Metrics

In this section, we first provide a detailed overview of the benchmarks (§5.1) and evaluation metrics (§5.2) commonly used in MLLM token compression studies.

5.1 Benchmarks

Table 8 summarizes the image and video understanding benchmarks commonly used in token pruning studies. Depending on the primary capability being evaluated, these benchmarks can be grouped into several categories.

For image understanding benchmarks, the categories include:

- **General Image Perception**: Evaluates basic visual recognition skills in natural images, such as identifying objects, scenes, attributes, and spatial relationships.
- **Optical Character Recognition** (OCR): Measures the ability to recognize and interpret textual content embedded in unstructured visual formats. This skill is crucial for enabling effective interaction between MLLMs and humans.
- **Knowledge**: Assesses the integration of visual perception with domain-specific or general world knowledge across diverse disciplines.
- **Reasoning**: Goes beyond perception, requiring logical inference and problem-solving based on visual content combined with specific prior knowledge.
- **Integrated Image Understanding**: Provides a holistic evaluation by combining visual perception and reasoning tasks into a single benchmark, thereby testing comprehensive multimodal understanding.

For video understanding benchmarks, the categories include:

- **Temporal Understanding**: Measures the ability to capture and interpret temporal dynamics, such as action sequences, motion patterns, and event localizations.
- **Long Video Understanding**: Evaluates the capacity to process and reason over long-form videos, ranging from several to tens of minutes.
- **Integrated Video Understanding**: Offers a holistic assessment of perception and reasoning skills in video contexts by combining multiple evaluation dimensions.

5.2 Metrics

The evaluation of MLLM token compression methods primarily considers two perspectives: downstream task performance (**effectiveness**) and computational efficiency (**efficiency**), either theoretical or practical.

5.2.1 Effectiveness

Effectiveness evaluation typically follows the standard of original benchmarks. Most benchmarks adopt *Accuracy* as the primary metric, which measures whether the model's prediction matches the ground-truth answer. For open-ended tasks without a single correct answer (e.g., image captioning), *GPT-Score* is often employed to provide a numerical rating of the MLLM's response.

5.2.2 Efficiency

Efficiency can be evaluated from several complementary aspects:

- **Token Retention Count/Ratio**: Measures the absolute number or relative percentage of visual tokens preserved after compression. Token compression methods are commonly compared under the same retention count/ratio in downstream tasks. However, identical retention levels do not guarantee equal inference latency, as factors such as the compression position can significantly influence runtime.
- **Prefilling/Decoding FLOPs**: Captures the theoretical computational cost of the query prefilling and decoding stages, measured in floating-point operations.
- **Prefilling/Decoding Latency**: Reports the actual wall-clock time required for the model to process input (prefilling) and generate output tokens (decoding). Unlike FLOPs, which is hardware-agnostic, this metric is dependent on the specific infrastructure and implementation.
- **Memory Usage**: This metric quantifies the peak memory footprint during inference, which is especially critical for deploying MLLMs on resource-constrained devices. Token compression can reduce the memory required for attention key–value caches and intermediate representations, but the reduction is highly dependent on how compression is implemented.

6 Application Scenarios

Following the development of advanced algorithms that significantly enhance the efficiency of MLLMs, subsequent efforts have focused on translating these methods into real-world applications, thereby maximizing their societal impact. In this section, we provide a concise overview of the key aspects that are critical for the applications.

6.1 Image Understanding

In image understanding, current algorithms primarily focus on accelerating the processing of high-resolution inputs, which is essential for downstream tasks.

Medical Image Processing. A key application lies in medical imaging, where MLLMs must rapidly and accurately interpret clinical data, underscoring the need to balance efficiency and accuracy. Extensive research [279], [280] has been devoted to evaluating the capabilities of these models. However, despite the rapid advancement, current models remain limited in effectively handling high-resolution medical imaging examination results. The incorporation of efficient token compression algorithms presents a promising avenue to further improve both efficiency and effectiveness in such settings.

Multi-page Document Understanding. Another valuable application is document understanding, where models must process long documents and generate concise summaries or meaningful solutions from the input. Prior studies [281] have primarily focused on improving accuracy and expanding the range of document lengths that models can handle. Inspired by advances in high-resolution image processing, where algorithms accelerate computation without sacrificing accuracy, similar techniques [124], [148] can be applied to document understanding. Such integration would allow models to manage longer inputs within limited context lengths while also improving overall efficiency.

Satellite and Remote Sensing Imagery. In industrial applications, MLLMs have been deployed to interpret satellite and remote sensing imagery [282]. These images typically contain rich structural information at high resolutions, yet practical deployments face computational resource constraints. Efficiently processing such imagery remains a significant challenge. Recent studies [283], [284] have explored token compression strategies to address this bottleneck, achieving notable progress by enabling models to handle higher-resolution inputs more efficiently—an advancement of considerable importance for industrial deployment.

6.2 Video Understanding

In the realm of video understanding tasks, previous research has primarily concentrated on addressing the inherent challenges associated with comprehending lengthy videos. Efforts have been made to reduce data redundancy and enhance efficiency in processing extended video content.

Embodied AI. A practical application of such algorithms lies in the development of robot learning and embodied AI. In these settings, embodied agents or robots must respond in real time to the visual input they receive during continuous video perception. The token compression strategy [74] addresses this challenge by efficiently capturing both spatial and temporal information, thereby enabling fine-grained

video understanding while maintaining computational efficiency. This capability is essential for the real-world deployment of robots and embodied agents, making them more suitable for practical applications.

Streaming Video Understanding. Another significant application domain is streaming video understanding, where models must process continuous video streams and deliver real-time responses with minimal latency. Prior studies [57], [285]–[287] have adopted token compression techniques to address the high temporal redundancy in dense video streams (e.g., 1-10 FPS), store compact historical representations through memory mechanisms, and efficiently retrieve question-relevant KV caches during inference. These strategies enable models to maintain responsiveness and accuracy while managing computational resources effectively, a critical requirement for real-time applications.

Instructional Video Summary. Other real-world applications, such as meeting summarization and lecture keypoint extraction, also require models to achieve efficient video understanding while preserving fine-grained details. Several studies [5], [6] have investigated these challenging scenarios and proposed a variety of solutions. A central idea underlying these approaches is the selective retention of informative tokens while discarding redundant ones, thereby improving overall efficiency and facilitating the practical adoption of such methods in real-world tasks.

6.3 Other Applications

Beyond accelerating the processing of high-resolution images and long videos through redundant token reduction, token pruning demonstrates considerable potential across diverse applications. A key advantage of this approach is its ability to guide model attention toward the most relevant image or video regions [288]. By filtering out background noise and irrelevant objects, models can allocate computational capacity to critical visual information essential for accurately interpreting and responding to prompts. Prior studies [289], [290] have shown that this improved focus can *mitigate visual hallucinations*, where models generate text inconsistent with visual input. Through selective token pruning, these strategies improve the grounding of model outputs in the actual visual context.

7 Open Challenges and Future Work

Despite the rapid progress in token compression for MLLMs, several open challenges remain that warrant further investigation. We discuss remaining open challenges and future research directions in this section.

7.1 Lack of Theoretical Understanding

Although token compression has achieved notable empirical success, most existing approaches remain largely experience-driven and lack rigorous theoretical grounding. Apart from a few works, such as DeCo [105] and DART [183], which analyze how compression influences representation learning within MLLMs, the majority of methods rely on heuristic intuition and limited empirical validation. Consequently, they often exhibit poor transferability across datasets, architectures, and modalities, as well as insufficient robustness under distribution shift.

A key weakness lies in the absence of a principled theory of token importance. Current practices—such as ranking tokens by attention weights, pairwise similarity, or mutual information—lack causal or generalization-based justification. These metrics indicate correlation rather than necessity, offering little explanation of whether the retained tokens are truly sufficient for the downstream objective or merely coincidental with good performance.

By connecting token selection to sufficiency, causality, and robustness, future work can move beyond ad-hoc heuristics toward a principled understanding of why compression works, enabling generalizable and theoretically sound compression strategies for MLLMs.

7.2 Lack of Task- and Content-Aware Adaptivity

Most existing token compression strategies operate in a task-agnostic and content-agnostic manner, applying a fixed compression ratio or heuristic rule regardless of the task type or the visual complexity of the input. However, the granularity of information required to fulfill a given question varies substantially. As M^3 [91] observed, for most benchmarks, especially those mainly crafted from natural scenes (such as COCO [291]), can be handled well with only 9 tokens per image. In contrast, dense visual perception tasks such as document understanding or OCR require a greater amount of tokens ($144 \sim 576$ tokens) per image to handle the task well. A uniform compression policy thus risks either retaining redundant tokens for simple tasks or discarding crucial details for complex ones, leading to inefficiency and degraded understanding. Similarly, multimodal inputs such as images or video clips exhibit vastly different levels of informational richness. Compressing them under a single fixed strategy ignores variations in object density, scene complexity, and visual salience. Yet few existing methods explicitly model this heterogeneity or incorporate adaptive mechanisms conditioned on either the task semantics or the visual content itself.

Future research should explore task- and content-aware compression, where the model dynamically determines the degree and manner of token reduction. Some recent studies [100], [101], [186] have begun to move in this direction, introducing adaptive mechanisms that modulate compression according to textual queries or visual content complexity. However, how to further couple such adaptive compression with improved training strategies to achieve stronger generalization across diverse tasks remains an open question. For instance, VisionThink [190] proposes a reinforcement learning-based approach that enables the model to autonomously decide whether the higher-resolution visual input is necessary for a given task, offering a promising step toward fully adaptive token compression. Such adaptive strategies would align the compression process more closely with the cognitive demands of multimodal understanding, improving both efficiency and fidelity across diverse tasks.

7.3 Performance Degradation in Practical Tasks

Although many token compression methods demonstrate competitive results on general Visual QA tasks [292], often

maintaining comparable accuracy even when reducing visual tokens to $1/3$ or $1/4$ of the original, this performance stability does not generalize well to real-world applications. Tasks that require fine-grained perception, such as OCR [293], [294], document understanding [295], and dense reasoning over structured visual layouts, tend to experience a substantial drop in accuracy after compression. These scenarios demand precise localization, text recognition, and structural alignment, where the loss of subtle spatial or semantic cues introduced by aggressive compression becomes detrimental. This performance gap highlights a key limitation: current compression schemes prioritize average efficiency rather than task-specific fidelity, which constrains their applicability in practical multimodal systems requiring high-resolution understanding or domain-level precision.

7.4 Limitations of Existing Evaluation

From an evaluation perspective, the efficiency and effectiveness of existing token compression methods are primarily assessed through downstream multimodal tasks. We identify three key limitations in current MLLM token compression evaluation practices:

Lack of systematic task categorization. As shown in Table 8, benchmarks are grouped into broad categories, offering limited insight into how token compression affects specific visual understanding capabilities (e.g., spatial relation reasoning or object motion tracking) and content domains (e.g., table or chart interpretation).

Inefficient evaluation processes. Current evaluations typically employ at least ten benchmarks encompassing tens of thousands of examples. Many benchmarks exhibit substantial overlap in evaluation focus, leading to redundant assessments and inefficient resource utilization.

Absence of consistent evaluation standards. The selection of benchmarks and metrics varies widely across studies, with each work emphasizing different strengths. This inconsistency hinders fair cross-method comparison.

Although recent efforts have introduced more challenging evaluation settings tailored for token compression approaches [296], a systematic and standardized evaluation framework remains necessary to enable fair comparisons and advance progress in this field.

8 Conclusion

MLLMs represent a significant advancement in cross-modal understanding, yet computational efficiency remains a critical bottleneck. Token compression emerges as a promising solution by reducing redundancy across MLLM components, enhancing both training and inference efficiency while alleviating long-context reasoning complexity. The field has evolved from single-module to multi-module compression, from fixed-rate to adaptive dynamic approaches, and from static images to complex video sequences. However, key challenges persist: the absence of unified evaluation frameworks for token compression, limited integration with mainstream training or inference acceleration libraries, and insufficient synergy with other MLLM efficiency techniques. This survey provides a systematic foundation for advancing efficient, scalable, and practically deployable multimodal large language models through strategic token compression methodologies.

References

[1] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. 1

[2] H. Liu, C. Li, Y. Li, B. Li, Y. Zhang, S. Shen, and Y. J. Lee, "Llava-next: Improved reasoning, ocr, and world knowledge," 2024. 1

[3] K. Ataallah, X. Shen, E. Abdelrahman, E. Sleiman, D. Zhu, J. Ding, and M. Elhoseiny, "Minigpt4-video: Advancing multimodal llms for video understanding with interleaved visual-textual tokens," *ArXiv preprint*, vol. abs/2404.03413, 2024. 1

[4] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang, H. Zhong, Y. Zhu, M. Yang, Z. Li, J. Wan, P. Wang, W. Ding, Z. Fu, Y. Xu, J. Ye, X. Zhang, T. Xie, Z. Cheng, H. Zhang, Z. Yang, H. Xu, and J. Lin, "Qwen2.5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025. 1, 15, 16, 19

[5] K. Li, Y. He, Y. Wang, Y. Li, W. Wang, P. Luo, Y. Wang, L. Wang, and Y. Qiao, "Videochat: Chat-centric video understanding," *ArXiv preprint*, vol. abs/2305.06355, 2023. 1, 22

[6] B. Zhang, K. Li, Z. Cheng, Z. Hu, Y. Yuan, G. Chen, S. Leng, Y. Jiang, H. Zhang, X. Li *et al.*, "Videollama 3: Frontier multimodal foundation models for image and video understanding," *arXiv preprint arXiv:2501.13106*, 2025. 1, 22

[7] P. Tong, E. Brown, P. Wu, S. Woo, A. J. V. IYER, S. C. Akula, S. Yang, J. Yang, M. Middepogu, Z. Wang *et al.*, "Cambrian-1: A fully open, vision-centric exploration of multimodal llms," *Advances in Neural Information Processing Systems*, vol. 37, pp. 87 310–87 356, 2024. 1, 3, 7

[8] D. Li, Y. Liu, H. Wu, Y. Wang, Z. Shen, B. Qu, X. Niu, F. Zhou, C. Huang, Y. Li *et al.*, "Aria: An open multimodal native mixture-of-experts model," *arXiv preprint arXiv:2410.05993*, 2024. 1

[9] S. Ren, L. Yao, S. Li, X. Sun, and L. Hou, "Timechat: A time-sensitive multimodal large language model for long video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 313–14 323. 1, 14, 16, 17

[10] B. Li, Y. Zhang, D. Guo, R. Zhang, F. Li, H. Zhang, K. Zhang, P. Zhang, Y. Li, Z. Liu *et al.*, "Llava-onevision: Easy visual task transfer," *arXiv preprint arXiv:2408.03326*, 2024. 1, 4, 19

[11] Y. Zhang, J. Wu, W. Li, B. Li, Z. Ma, Z. Liu, and C. Li, "Video instruction tuning with synthetic data," 2024. [Online]. Available: https://arxiv.org/abs/2410.02713 1

[12] S. Tong, E. L. Brown II, P. Wu, S. Woo, A. J. IYER, S. C. Akula, S. Yang, J. Yang, M. Middepogu, Z. Wang *et al.*, "Cambrian-1: A fully open, vision-centric exploration of multimodal llms," in *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. 1

[13] Z. Sun, S. Shen, S. Cao, H. Liu, C. Li, Y. Shen, C. Gan, L.-Y. Gui, Y.-X. Wang, Y. Yang, K. Keutzer, and T. Darrell, "Aligning large multimodal models with factually augmented rlhf," *ArXiv preprint*, vol. abs/2309.14525, 2023. 1

[14] L.-C.-T. Xiaomi, "Mimo-vl technical report," 2025. [Online]. Available: https://arxiv.org/abs/2506.03569 1

[15] T. Wan, A. Wang, B. Ai, B. Wen, C. Mao, C.-W. Xie, D. Chen, F. Yu, H. Zhao, J. Yang, J. Zeng, J. Wang, J. Zhang, J. Zhou, J. Wang, J. Chen, K. Zhu, K. Zhao, K. Yan, L. Huang, M. Feng, N. Zhang, P. Li, P. Wu, R. Chu, R. Feng, S. Zhang, S. Sun, T. Fang, T. Wang, T. Gui, T. Weng, T. Shen, W. Lin, W. Wang, W. Wang, W. Zhou, W. Wang, W. Shen, W. Yu, X. Shi, X. Huang, X. Xu, Y. Kou, Y. Lv, Y. Li, Y. Liu, Y. Wang, Y. Zhang, Y. Huang, Y. Li, Y. Wu, Y. Liu, Y. Pan, Y. Zheng, Y. Hong, Y. Shi, Y. Feng, Z. Jiang, Z. Han, Z.-F. Wu, and Z. Liu, "Wan: Open and advanced large-scale video generative models," *arXiv preprint arXiv:2503.20314*, 2025. 1

[16] L. Li, Y. Liu, L. Yao, P. Zhang, C. An, L. Wang, X. Sun, L. Kong, and Q. Liu, "Temporal reasoning transfer from text to video," in *ICLR 2025*. OpenReview.net, 2025. [Online]. Available: https://openreview.net/forum?id=sHAvMp5J4R 1

[17] K. Ouyang, Y. Liu, L. Yao, Y. Cai, H. Zhou, J. Zhou, F. Meng, and X. Sun, "Conan: Progressive learning to reason like a detective over multi-scale visual evidence," *arXiv preprint arXiv:2510.20470*, 2025. 1

[18] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023. 1, 3

[19] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, A. Goyal, A. Hartshorn, A. Yang, A. Mitra, A. Sravankumar, A. Korenev, A. Hinsvark, A. Rao, A. Zhang, A. Rodriguez, A. Gregerson, A. Spataru, B. Rozière, B. Biron, B. Tang, B. Chern, C. Caucheteux, C. Nayak, C. Bi, C. Marra, C. McConnell, C. Keller, C. Touret, C. Wu, C. Wong, C. C. Ferrer, C. Nikolaidis, D. Allonsius, D. Song, D. Pintz, D. Livshits, D. Esiobu, D. Choudhary, D. Mahajan, D. Garcia-Olano, D. Perino, D. Hupkes, E. Lakomkin, E. AlBadawy, E. Lobanova, E. Dinan, E. M. Smith, F. Radenovic, F. Zhang, G. Synnaeve, G. Lee, G. L. Anderson, G. Nail, G. Mialon, G. Pang, G. Cucurell, H. Nguyen, H. Korevaar, H. Xu, H. Touvron, I. Zarov, I. A. Ibarra, I. M. Kloumann, I. Misra, I. Evtimov, J. Copet, J. Lee, J. Geffert, J. Vranes, J. Park, J. Mahadeokar, J. Shah, J. van der Linde, J. Billock, J. Hong, J. Lee, J. Fu, J. Chi, J. Huang, J. Liu, J. Wang, J. Yu, J. Bitton, J. Spisak, J. Park, J. Rocca, J. Johnstun, J. Saxe, J. Jia, K. V. Alwala, K. Upasani, K. Plawiak, K. Li, K. Heafield, K. Stone, and et al., "The llama 3 herd of models," *ArXiv preprint*, vol. abs/2407.21783, 2024. 1

[20] OpenAI, "Introducing chatgpt," 2022. 1

[21] A. Yang, B. Yang, B. Hui, B. Zheng, B. Yu, C. Zhou, C. Li, C. Li, D. Liu, F. Huang, G. Dong, H. Wei, H. Lin, J. Tang, J. Wang, J. Yang, J. Tu, J. Zhang, J. Ma, J. Xu, J. Zhou, J. Bai, J. He, J. Lin, K. Dang, K. Lu, K. Chen, K. Yang, M. Li, M. Xue, N. Ni, P. Zhang, P. Wang, R. Peng, R. Men, R. Gao, R. Lin, S. Wang, S. Bai, S. Tan, T. Zhu, T. Li, T. Liu, W. Ge, X. Deng, X. Zhou, X. Ren, X. Zhang, X. Wei, X. Ren, Y. Fan, Y. Yao, Y. Zhang, Y. Wan, Y. Chu, Y. Liu, Z. Cui, Z. Zhang, and Z. Fan, "Qwen2 technical report," *ArXiv preprint*, vol. abs/2407.10671, 2024. 1

[22] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025. 1

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *International Conference on Learning Representations*, 2020. 1

[24] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," *arXiv preprint arXiv:2304.08485*, 2023. 1

[25] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin, "Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution," 2024. 1, 15

[26] OpenAI, "Hello gpt-4o," 2024. [Online]. Available: https://openai.com/index/hello-gpt-4o/ 1

[27] P. Wu and S. Xie, "V*: Guided visual search as a core mechanism in multimodal llms," *arXiv preprint arXiv:2312.14135*, 2023. 1

[28] C. Fu, Y. Dai, Y. Luo, L. Li, S. Ren, R. Zhang, Z. Wang, C. Zhou, Y. Shen, M. Zhang *et al.*, "Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis," *ArXiv preprint*, vol. abs/2405.21075, 2024. 1, 17

[29] J. Zhou, Y. Shu, B. Zhao, B. Wu, S. Xiao, X. Yang, Y. Xiong, B. Zhang, T. Huang, and Z. Liu, "Mlvu: A comprehensive benchmark for multi-task long video understanding," *ArXiv preprint*, vol. abs/2406.04264, 2024. 1, 17

[30] Y. Liang, C. Ge, Z. Tong, Y. Song, J. Wang, and P. Xie, "Not all patches are what you need: Expediting vision transformers via token reorganizations," *arXiv preprint arXiv:2202.07800*, 2022. 1

[31] H. Yin, A. Vahdat, J. Alvarez, A. Mallya, J. Kautz, and P. Molchanov, "Adavit: Adaptive tokens for efficient vision transformer," *arXiv preprint arXiv:2112.07658*, 2021. 1

[32] M. Fayyaz, S. A. Koohpayegani, F. R. Jafari, S. Sengupta, H. R. V. Joze, E. Sommerlade, H. Pirsiavash, and J. Gall, "Adaptive token sampling for efficient vision transformers," in *European conference on computer vision*. Springer, 2022, pp. 396–414. 1

[33] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C.-J. Hsieh, "Dynamicvit: Efficient vision transformers with dynamic token sparsification," *Advances in neural information processing systems*, vol. 34, pp. 13 937–13 949, 2021. 1, 12

[34] D. Bolya, C.-Y. Fu, X. Dai, P. Zhang, C. Feichtenhofer, and J. Hoffman, "Token merging: Your vit but faster," *arXiv preprint arXiv:2210.09461*, 2022. 1, 5, 6, 12, 17, 18

[35] S. Peng, D. Fu, B. Wei, Y. Cao, L. Gao, and Z. Tang, "Vote&mix: Plug-and-play token reduction for efficient vision transformer," *arXiv preprint arXiv:2408.17062*, 2024. 1

[36] R. Choudhury, G. Zhu, S. Liu, K. Niinuma, K. Kitani, and L. Jeni, "Don't look twice: Faster video transformers with run-length tokenization," *Advances in Neural Information Processing Systems*, vol. 37, pp. 28 127–28 149, 2024. 1

[37] R. Koner, G. Jain, P. Jain, V. Tresp, and S. Paul, "Lookupvit: Compressing visual information to a limited number of tokens," in *European Conference on Computer Vision*. Springer, 2024, pp. 322–337. 1, 3, 7

[38] K. Zhou, "Lvp: Language-guide visual projector for efficient multimodal llm." 1

[39] Y. He, F. Chen, J. Liu, W. Shao, H. Zhou, K. Zhang, and B. Zhuang, "Zipvl: Efficient large vision-language models with dynamic token sparsification and kv cache compression," 2024. 1

[40] Q. Wu, W. Lin, Y. Zhou, W. Ye, Z. Zen, X. Sun, and R. Ji, "Accelerating multimodal large language models via dynamic visual-token exit and the empirical findings," *arXiv preprint arXiv:2411.19628*, 2024. 1

[41] H. Wang, Y. Nie, Y. Ye, D. GuanYu, Y. Wang, S. Li, H. Yu, J. Lu, and C. Huang, "Dynamic-vlm: Simple dynamic visual token compression for videollm," *arXiv preprint arXiv:2412.09530*, 2024. 1, 5

[42] Y. Jiang, Q. Wu, W. Lin, W. Yu, and Y. Zhou, "What kind of visual tokens do we need? training-free visual token pruning for multi-modal large language models from the perspective of graph," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 4, 2025, pp. 4075–4083. 1

[43] ——, "What kind of visual tokens do we need? training-free visual token pruning for multi-modal large language models from the perspective of graph," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 4, 2025, pp. 4075–4083. 1

[44] R. Zhang, R. Shao, G. Chen, M. Zhang, K. Zhou, W. Guan, and L. Nie, "Falcon: Resolving visual redundancy and fragmentation in high-resolution multimodal large language models via visual registers," *arXiv preprint arXiv:2501.16297*, 2025. 1, 5

[45] H. Wang, Z. Yu, G. Spadaro, C. Ju, V. Quétu, S. Xiao, and E. Tartaglione, "Folder: Accelerating multi-modal large language models with enhanced performance," *arXiv preprint arXiv:2501.02430*, 2025. 1

[46] Z. Wen, Y. Gao, W. Li, C. He, and L. Zhang, "Token pruning in multimodal large language models: Are we solving the right problem?" *arXiv preprint arXiv:2502.11501*, 2025. 1

[47] H. Zhang, M. Lyu, C. He, Y. Ao, and Y. Lin, "Towards adaptive visual token pruning for large multimodal models," *arXiv preprint arXiv:2509.00320*, 2025. 1

[48] J. Ma, Q. Zhang, M. Lu, Z. Wang, Q. Zhou, J. Song, and S. Zhang, "Mmg-vid: Maximizing marginal gains at segment-level and token-level for efficient video llms," *arXiv preprint arXiv:2508.21044*, 2025. 1

[49] K. Zeng, G. Zhong, J. Cheng, J. Yuan, and Z. Li, "Avam: Universal training-free adaptive visual anchoring embedded into multimodal large language model for multi-image question answering," *arXiv preprint arXiv:2508.17860*, 2025. 1

[50] Z. Tang, Z. Ma, S. Wang, Z. Li, L. Zhang, H. Zhao, Y. Li, and Q. Wang, "Covipal: Layer-wise contextualized visual token pruning for large vision-language models," *arXiv preprint arXiv:2508.17243*, 2025. 1

[51] K. Zhao, W. Yuan, A. L. Hung, and D. Zeng, "Pore: Position-reweighted visual token pruning for vision language models," *arXiv preprint arXiv:2508.17807*, 2025. 1

[52] W. Wang, Z. Gao, L. Gu, H. Pu, L. Cui, X. Wei, Z. Liu, L. Jing, S. Ye, J. Shao *et al.*, "Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency," *arXiv preprint arXiv:2508.18265*, 2025. 1, 19

[53] J. Liu, J. Lin, Y. Wei, K. Shao, K. Tao, J. Huang, X. Yang, Z. Chen, H. Wang, and X. Jin, "Revisiting mllm token technology through the lens of classical visual coding," *arXiv preprint arXiv:2508.13460*, 2025. 1

[54] Z. Zhang, S. Liu, W. Yu, X. Wang *et al.*, "Top-down compression: Revisit efficient vision token projection for visual instruction tuning," *arXiv preprint arXiv:2505.11945*, 2025. 1

[55] Y. Shu, Z. Liu, P. Zhang, M. Qin, J. Zhou, Z. Liang, T. Huang, and B. Zhao, "Video-xl: Extra-long vision language model for hour-scale video understanding," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 26 160–26 169. 1, 5, 15, 16

[56] X. Shen, Y. Xiong, C. Zhao, L. Wu, J. Chen, C. Zhu, Z. Liu, F. Xiao, B. Varadarajan, F. Bordes *et al.*, "Longvu: Spatiotemporal

adaptive compression for long video-language understanding," *arXiv preprint arXiv:2410.17434*, 2024. 1, 3, 5, 8, 15, 17

[57] L. Yao, Y. Li, Y. Wei, L. Li, S. Ren, Y. Liu, K. Ouyang, L. Wang, S. Li, S. Li *et al.*, "Timechat-online: 80% visual tokens are naturally redundant in streaming videos," *arXiv preprint arXiv:2504.17343*, 2025. 1, 3, 5, 15, 16, 17, 18, 22

[58] M. Qin, X. Liu, Z. Liang, Y. Shu, H. Yuan, J. Zhou, S. Xiao, B. Zhao, and Z. Liu, "Video-xl-2: Towards very long-video understanding through task-aware kv sparsification," *arXiv preprint arXiv:2506.19225*, 2025. 1, 3, 5, 13, 16

[59] Z. Liu, Y. Dong, Z. Liu, W. Hu, J. Lu, and Y. Rao, "Oryx mllm: On-demand spatial-temporal understanding at arbitrary resolution," *arXiv preprint arXiv:2409.12961*, 2024. 1

[60] Z. Liu, Y. Dong, J. Wang, Z. Liu, W. Hu, J. Lu, and Y. Rao, "Ola: Pushing the frontiers of omni-modal language model," *arXiv preprint arXiv:2502.04328*, 2025. 1

[61] Y. Ji, J. Zhang, H. Xia, J. Chen, L. Shou, G. Chen, and H. Li, "Specvlm: Enhancing speculative decoding of video llms via verifier-guided token pruning," *arXiv preprint arXiv:2508.16201*, 2025. 1, 5

[62] S. Dong, J. Hu, M. Zhang, M. Yin, Y. Fu, and Q. Qian, "Mmtok: Multimodal coverage maximization for efficient inference of vlms," *arXiv preprint arXiv:2508.18264*, 2025. 1

[63] J. Chen, X. Liu, Z. Wen, Y. Wang, S. Huang, and H. Chen, "Variation-aware vision token dropping for faster large vision-language models," *arXiv preprint arXiv:2509.01552*, 2025. 1

[64] X. Wang, J. Zhang, T. Wang, H. Zhang, and F. Zheng, "Seeing more, saying more: Lightweight language experts are dynamic video token compressors," *arXiv preprint arXiv:2509.00969*, 2025. 1, 5

[65] K. H. I. Arif, J. Yoon, D. S. Nikolopoulos, H. Vandierendonck, D. John, and B. Ji, "Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 2, 2025, pp. 1773–1781. 3, 5

[66] D. Song, W. Wang, S. Chen, X. Wang, M. Guan, and B. Wang, "Less is more: A simple yet effective token reduction method for efficient multi-modal llms," 2024. [Online]. Available: https://arxiv.org/abs/2409.10994 3, 6

[67] A. Jeddi, N. Baghbanzadeh, E. Dolatabadi, and B. Taati, "Similarity-aware token pruning: Your vlm but faster," *arXiv preprint arXiv:2503.11549*, 2025. 3, 6

[68] Q. Zhang, A. Cheng, M. Lu, R. Zhang, Z. Zhuo, J. Cao, S. Guo, Q. She, and S. Zhang, "Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms," *arXiv preprint arXiv:2412.01818*, 2024. 3, 5, 6, 18

[69] J. Liu, F. Du, G. Zhu, N. Lian, J. Li, and B. Chen, "Hiprune: Training-free visual token pruning via hierarchical attention in vision-language models," *arXiv preprint arXiv:2508.00553*, 2025. 3, 6

[70] S. Yang, R. Xu, C. Cui, T. Wang, D. Lin, and J. Pang, "Vflowopt: A token pruning framework for lmms with visual information flow-guided optimization," *arXiv preprint arXiv:2508.05211*, 2025. 3, 5, 6

[71] C. Zhang, K. Ma, T. Fang, W. Yu, H. Zhang, Z. Zhang, Y. Xie, K. Sycara, H. Mi, and D. Yu, "Vscan: Rethinking visual token reduction for efficient large vision-language models," *arXiv preprint arXiv:2505.22654*, 2025. 3, 7, 8, 11, 12

[72] Z. Wang, J. Chen, W. Zhou, H. Zhu, J. Liang, L. Shan, M. Liu, D. Xu, Q. Yang, and B. Qin, "Smarttrim: Adaptive tokens and attention pruning for efficient vision-language models," *arXiv preprint arXiv:2305.15033*, 2023. 3, 6

[73] J. Cao, P. Ye, S. Li, C. Yu, Y. Tang, J. Lu, and T. Chen, "Madtp: Multimodal alignment-guided dynamic token pruning for accelerating vision-language transformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 15 710–15 719. 3, 6, 17, 18

[74] J. Li, K. Li, C. Gao, Y. Li, and X. Chen, "Egoprune: Efficient token pruning for egomotion video reasoning in embodied agent," *arXiv preprint arXiv:2507.15428*, 2025. 3, 6, 21

[75] J. Liu, L. Niu, W. Chen, J. Zhou, and F. Meng, "Laco: Efficient layer-wise compression of visual tokens for multimodal large language models," *arXiv preprint arXiv:2507.02279*, 2025. 3, 7

[76] H. Tang and C. Shen, "Learning compact vision tokens for efficient large multimodal models," *arXiv preprint arXiv:2506.07138*, 2025. 3, 7

[77] P. K. A. Vasu, F. Faghri, C.-L. Li, C. Koc, N. True, A. Antony, G. Santhanam, J. Gabriel, P. Grasch, O. Tuzel *et al.*, "Fastvlm: Efficient vision encoding for vision language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 19 769–19 780. 3, 7

[78] L. Kong, H. Zhang, J. Zhang, J. Huang, K. Li, Q. Wang, and F. Zhang, "Clapper: Compact learning and video representation in vlms," *arXiv preprint arXiv:2505.15529*, 2025. 3, 15

[79] R. Zhang, Y. Lyu, R. Shao, G. Chen, W. Guan, and L. Nie, "Token-level correlation-guided compression for efficient multimodal document understanding," *arXiv preprint arXiv:2407.14439*, 2024. 3, 7, 18

[80] Y. Guo, W. Dong, J. Song, S. Zhu, X. Zhang, H. Yang, Y. Wang, Y. Du, X. Chen, and B. Zheng, "Fila-video: Spatio-temporal compression for fine-grained long video understanding," *arXiv preprint arXiv:2504.20384*, 2025. 3, 7, 15

[81] P. Jin, R. Takanobu, W. Zhang, X. Cao, and L. Yuan, "Chat-univi: Unified visual representation empowers large language models with image and video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 13 700–13 710. 3, 7, 15

[82] D. Shi, C. Tao, A. Rao, Z. Yang, C. Yuan, and J. Wang, "Crossget: Cross-guided ensemble of tokens for accelerating vision-language transformers," *arXiv preprint arXiv:2305.17455*, 2023. 3, 7, 13, 17, 19

[83] S. Ren, S. Chen, S. Li, X. Sun, and L. Hou, "Testa: Temporal-spatial token aggregation for long-form video-language understanding," *arXiv preprint arXiv:2310.19060*, 2023. 3, 7, 15, 17, 18

[84] Y. Shang, M. Cai, B. Xu, Y. J. Lee, and Y. Yan, "Llava-prumerge: Adaptive token reduction for efficient large multimodal models," *arXiv preprint arXiv:2403.15388*, 2024. 3, 5, 7, 8, 18

[85] M. Huang, R. Huang, H. Shi, Y. Chen, C. Zheng, X. Sun, X. Jiang, Z. Li, and H. Cheng, "Efficient multi-modal large language models via visual token grouping," *arXiv preprint arXiv:2411.17773*, 2024. 3, 7

[86] Y. Han, X. Liu, P. Ding, D. Wang, H. Chen, Q. Yan, and S. Huang, "Rethinking token reduction in mllms: Towards a unified paradigm for training-free acceleration," *arXiv e-prints*, pp. arXiv–2411, 2024. 3, 7, 12, 14

[87] J. Hyun, S. Hwang, S. H. Han, T. Kim, I. Lee, D. Wee, J.-Y. Lee, S. J. Kim, and M. Shim, "Multi-granular spatio-temporal token merging for training-free acceleration of video llms," *arXiv preprint arXiv:2507.07990*, 2025. 3, 5, 7

[88] L. Gao, Y. Zhong, Y. Zeng, H. Tan, D. Li, and Z. Zhao, "Linvt: Empower your image-level large language model to understand videos," *arXiv preprint arXiv:2412.05185*, 2024. 3, 7, 15, 17

[89] Q. Zhu, X. Wang, Z. Lu, J. Lao, C. Jin, J. Chen, Y. Peng, Q. Zhu, L. Zhong, J. Liu *et al.*, "Admire: Adaptive method to enhance multiple image resolutions in text-rich multi-image understanding," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 5237–5248. 3, 7

[90] X. Li, Y. Wang, J. Yu, X. Zeng, Y. Zhu, H. Huang, J. Gao, K. Li, Y. He, C. Wang *et al.*, "Videochat-flash: Hierarchical compression for long-context video modeling," *arXiv preprint arXiv:2501.00574*, 2024. 3

[91] M. Cai, J. Yang, J. Gao, and Y. J. Lee, "Matryoshka multimodal models," *arXiv preprint arXiv:2405.17430*, 2024. 3, 5, 7, 22

[92] T. Liu, L. Shi, R. Hong, Y. Hu, Q. Yin, and L. Zhang, "Multi-stage vision token dropping: Towards efficient multimodal large language model," *arXiv preprint arXiv:2411.10803*, 2024. 3, 8, 13, 14, 18, 19

[93] S. Yang, Y. Chen, Z. Tian, C. Wang, J. Li, B. Yu, and J. Jia, "Visionzip: Longer is better but not necessary in vision language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 19 792–19 802. 3, 4, 5, 7, 17, 19

[94] L. Jiang, W. Huang, T. Liu, Y. Zeng, J. Li, L. Cheng, and X. Xu, "Fopru: Focal pruning for efficient large vision-language models," *arXiv preprint arXiv:2411.14164*, 2024. 3, 8

[95] B. Xu, Y. Shang, Y. Ge, Q. Lou, and Y. Yan, "freepruner: A training-free approach for large multimodal model acceleration," *arXiv preprint arXiv:2411.15446*, 2024. 3, 8

[96] X. Zou, D. Lu, Y. Wang, Y. Yan, Y. Lyu, X. Zheng, L. Zhang, and X. Hu, "Don't just chase "highlighted tokens" in mllms: Revisiting visual holistic context retention," 2025. [Online]. Available: https://arxiv.org/abs/2510.02912 3, 5, 8, 17, 18

[97] H. Wang, J. Kai, H. Bai, L. Hou, B. Jiang, Z. He, and Z. Lin, "Fourier-vlm: Compressing vision tokens in the fre-

quency domain for large vision-language models," *arXiv preprint arXiv:2508.06038*, 2025. 3, 7

[98] Y. Li, C. Wang, and J. Jia, "Llama-vid: An image is worth 2 tokens in large language models," in *European Conference on Computer Vision*. Springer, 2024, pp. 323–340. 3, 8, 13, 19

[99] Z. Cheng, S. Leng, H. Zhang, Y. Xin, X. Li, G. Chen, Y. Zhu, W. Zhang, Z. Luo, D. Zhao *et al.*, "Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms," *arXiv preprint arXiv:2406.07476*, 2024. 3, 8, 19

[100] Y. Liu, F. Wu, R. Li, Z. Tang, and K. Li, "Par: Prompt-aware token reduction method for efficient large multimodal models," *arXiv preprint arXiv:2410.07278*, 2024. 3, 8, 14, 19, 22

[101] S. Li, J. Xu, X.-H. Li, C. Deng, and L.-L. Huang, "Qg-vtc: Question-guided visual token compression in mllms for efficient vqa," *arXiv preprint arXiv:2504.00654*, 2025. 3, 8, 22

[102] Y. Chen, J. Xu, X.-Y. Zhang, W.-Z. Liu, Y.-Y. Liu, and C.-L. Liu, "Recoverable compression: A multimodal vision token recovery mechanism guided by text information," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 2, 2025, pp. 2293–2301. 3, 8

[103] D. Wang, J. Cui, M. Li, W. Lin, B. Chen, and H. Zhang, "Instruction tuning-free visual token complement for multimodal llms," in *European Conference on Computer Vision*. Springer, 2024, pp. 446–462. 3, 8

[104] X. Chu, L. Qiao, X. Zhang, S. Xu, F. Wei, Y. Yang, X. Sun, Y. Hu, X. Lin, B. Zhang *et al.*, "Mobilevlm v2: Faster and stronger baseline for vision language model," *arXiv preprint arXiv:2402.03766*, 2024. 3, 5, 8, 9

[105] L. Yao, L. Li, S. Ren, L. Wang, Y. Liu, X. Sun, and L. Hou, "Deco: Decoupling token compression from semantic abstraction in multimodal large language models," *arXiv preprint arXiv:2405.20985*, 2024. 3, 5, 8, 9, 17, 18, 22

[106] Z. Lan, L. Niu, F. Meng, W. Li, J. Zhou, and J. Su, "Avg-llava: A large multimodal model with adaptive visual granularity," *arXiv preprint arXiv:2410.02745*, 2024. 3, 9, 18

[107] M. Gao, J. Liu, M. Li, J. Xie, Q. Liu, B. Zhao, X. Chen, and H. Xiong, "Tc-llava: Rethinking the transfer from image to video understanding with temporal considerations," 2024. [Online]. Available: https://arxiv.org/abs/2409.03206 3, 9, 15, 18

[108] L. Xu, Y. Zhao, D. Zhou, Z. Lin, S. K. Ng, and J. Feng, "Pllava: Parameter-free llava extension from images to videos for video dense captioning," *arXiv preprint arXiv:2404.16994*, 2024. 3, 8, 9, 15, 18, 19

[109] Z. Chen, W. Wang, H. Tian, S. Ye, Z. Gao, E. Cui, W. Tong, K. Hu, J. Luo, Z. Ma *et al.*, "How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites," *Science China Information Sciences*, vol. 67, no. 12, p. 220101, 2024. 3, 9, 18, 19

[110] Z. Liu, L. Zhu, B. Shi, Z. Zhang, Y. Lou, S. Yang, H. Xi, S. Cao, Y. Gu, D. Li *et al.*, "Nvila: Efficient frontier visual language models," *arXiv preprint arXiv:2412.04468*, 2024. 3

[111] Z. Chen, W. Wang, Y. Cao, Y. Liu, Z. Gao, E. Cui, J. Zhu, S. Ye, H. Tian, Z. Liu *et al.*, "Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling," *arXiv preprint arXiv:2412.05271*, 2024. 3

[112] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Dang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin, "Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution," *arXiv preprint arXiv:2409.12191*, 2024. 3, 4, 19

[113] J. Cha, W. Kang, J. Mun, and B. Roh, "Honeybee: Locality-enhanced projector for multimodal llm," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 13 817–13 827. 3, 8, 9, 18, 19

[114] Z. Cheng, S. Leng, H. Zhang, Y. Xin, X. Li, G. Chen, Y. Zhu, W. Zhang, Z. Luo, D. Zhao *et al.*, "Videollama 2: Advancing spatial-temporal modeling and audio understanding in video-llms," *arXiv preprint arXiv:2406.07476*, 2024. 3, 15

[115] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742. 3, 8, 9, 19

[116] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023. 3, 9, 19

[117] W. Dai, J. Li, D. Li, A. Tiong, J. Zhao, W. Wang, B. Li, P. N. Fung, and S. Hoi, "Instructblip: Towards general-purpose vision-language models with instruction tuning," *Advances in neural information processing systems*, vol. 36, pp. 49 250–49 267, 2023. 3, 9

[118] J. Bai, S. Bai, S. Yang, S. Wang, S. Tan, P. Wang, J. Lin, C. Zhou, and J. Zhou, "Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond," 2023. [Online]. Available: https://arxiv.org/abs/2308.12966 3, 9, 19

[119] W. Hu, Z.-Y. Dou, L. Li, A. Kamath, N. Peng, and K.-W. Chang, "Matryoshka query transformer for large vision-language models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 50 168–50 188, 2024. 3, 9, 10, 19

[120] D. Yan, P. Li, Y. Li, H. Chen, Q. Chen, W. Luo, W. Dong, Q. Yan, H. Zhang, and C. Shen, "Tg-llava: Text guided llava via learnable latent embeddings," 2024. [Online]. Available: https://arxiv.org/abs/2409.09564 3, 9, 10, 19

[121] Y. Li, J. Yang, Z. Shen, L. Han, H. Xu, and R. Tang, "Catp: Contextually adaptive token pruning for efficient and enhanced multimodal in-context learning," *arXiv preprint arXiv:2508.07871*, 2025. 3, 5, 12

[122] W. Li, Y. Yuan, J. Liu, D. Tang, S. Wang, J. Qin, J. Zhu, and L. Zhang, "Tokenpacker: Efficient visual projector for multimodal llm," *International Journal of Computer Vision*, pp. 1–19, 2025. 3, 5, 10, 18, 19

[123] R. Huang, X. Ding, C. Wang, J. Han, Y. Liu, H. Zhao, H. Xu, L. Hou, W. Zhang, and X. Liang, "Hires-llava: Restoring fragmentation input in high-resolution large vision-language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 29 814–29 824. 3, 5, 10, 19

[124] A. Hu, H. Xu, L. Zhang, J. Ye, M. Yan, J. Zhang, Q. Jin, F. Huang, and J. Zhou, "mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding," *arXiv preprint arXiv:2409.03420*, 2024. 3, 10, 21

[125] H. Zhang, J. Zhang, X. Ji, Q. Wang, and F. Zhang, "Dyntok: Dynamic compression of visual tokens for efficient and effective video understanding," *arXiv preprint arXiv:2506.03990*, 2025. 3, 5, 10, 18

[126] B. Sun, J. Zhao, X. Wei, and Q. Hou, "Llava-scissor: Token compression with semantic connected components for video llms," *arXiv preprint arXiv:2506.21862*, 2025. 3, 5, 10, 18

[127] Y. Omri, P. Shroff, and T. Tambe, "Token sequence compression for efficient multimodal computing," *arXiv preprint arXiv:2504.17892*, 2025. 3, 10

[128] S. R. Alvar, G. Singh, M. Akbari, and Y. Zhang, "Divprune: Diversity-based visual token pruning for large multimodal models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 9392–9401. 3, 5, 10, 11, 18

[129] L. Chen, H. Zhao, T. Liu, S. Bai, J. Lin, C. Zhou, and B. Chang, "An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models," in *European Conference on Computer Vision*. Springer, 2024, pp. 19–35. 3, 5, 17, 18, 19

[130] L. Xing, Q. Huang, X. Dong, J. Lu, P. Zhang, Y. Zang, Y. Cao, C. He, J. Wang, F. Wu *et al.*, "Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction," *arXiv preprint arXiv:2410.17247*, 2024. 3, 5, 17, 18, 19, 20

[131] Z. Lin, M. Lin, L. Lin, and R. Ji, "Boosting multimodal large language models with visual tokens withdrawal for rapid inference," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, no. 5, 2025, pp. 5334–5342. 3, 11

[132] Y. Zhang, C.-K. Fan, J. Ma, W. Zheng, T. Huang, K. Cheng, D. Gudovskiy, T. Okuno, Y. Nakata, K. Keutzer *et al.*, "Sparsevlm: Visual token sparsification for efficient vision-language model inference," *arXiv preprint arXiv:2410.04417*, 2024. 3, 5, 11, 17, 18, 19

[133] M. Endo, X. Wang, and S. Yeung-Levy, "Feather the throttle: Revisiting visual token pruning for vision-language model acceleration," *arXiv preprint arXiv:2412.13180*, 2024. 3, 5, 8, 11, 18

[134] X. Ye, Y. Gan, Y. Ge, X.-P. Zhang, and Y. Tang, "Atp-llava: Adaptive token pruning for large vision language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 24 972–24 982. 3, 11

[135] Y. Zhong, Z. Liu, Y. Li, and L. Wang, "Aim: Adaptive inference of multi-modal llms via token merging and pruning," *arXiv preprint arXiv:2412.03248*, 2024. 3, 11

[136] J. Zhuang, L. Lu, M. Dai, R. Hu, J. Chen, Q. Liu, and H. Hu, "St3: Accelerating multimodal large language model by spatial-temporal visual token trimming," in *Proceedings of the AAAI*

Conference on Artificial Intelligence, vol. 39, no. 10, 2025, pp. 11 049–11 057. 3, 11, 13

[137] S. Zhao, Z. Wang, F. Juefei-Xu, X. Xia, M. Liu, X. Wang, M. Liang, N. Zhang, D. N. Metaxas, and L. Yu, "Accelerating multimodal large language models by searching optimal vision token reduction," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 29 869–29 879. 3, 11

[138] C. Yang, Y. Sui, J. Xiao, L. Huang, Y. Gong, C. Li, J. Yan, Y. Bai, P. Sadayappan, X. Hu *et al.*, "Topv: Compatible token pruning with inference time optimization for fast and low-memory multimodal vision language model," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 19 803–19 813. 3, 5, 12

[139] M. Dhouib, D. Buscaldi, S. Vanier, and A. Shabou, "Pact: Pruning and clustering-based token reduction for faster visual language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 14 582–14 592. 3, 12

[140] J. Zhang, D. Meng, Z. Zhang, Z. Huang, T. Wu, and L. Wang, "p-mod: Building mixture-of-depths mllms via progressive ratio decay," *arXiv preprint arXiv:2412.04449*, 2024. 3, 12

[141] W. Huang, Z. Zhai, Y. Shen, S. Cao, F. Zhao, X. Xu, Z. Ye, Y. Hu, and S. Lin, "Dynamic-llava: Efficient multimodal large language models via dynamic vision-language context sparsification," *arXiv preprint arXiv:2412.00876*, 2024. 3, 12, 13

[142] X. Liang, C. Guan, J. Lu, H. Chen, H. Wang, and H. Hu, "Dynamic token reduction during generation for vision language models," *arXiv preprint arXiv:2501.14204*, 2025. 3, 12

[143] Q.-S. Zeng, Y. Li, Q. Wang, P.-T. Jiang, Z. Wu, M.-M. Cheng, and Q. Hou, "A glimpse to compress: Dynamic visual token pruning for large vision-language models," *arXiv preprint arXiv:2508.01548*, 2025. 3, 12

[144] J. Chen, L. Ye, J. He, Z.-Y. Wang, D. Khashabi, and A. Yuille, "Efficient large multi-modal models via visual context compression," *Advances in Neural Information Processing Systems*, vol. 37, pp. 73 986–74 007, 2024. 3, 12, 19

[145] T. Fu, T. Liu, Q. Han, G. Dai, S. Yan, H. Yang, X. Ning, and Y. Wang, "Framefusion: Combining similarity and importance for video token reduction on large visual language models," *arXiv preprint arXiv:2501.01986*, 2024. 3, 12, 18

[146] K. Shao, K. Tao, C. Qin, H. You, Y. Sui, and H. Wang, "Holitom: Holistic token merging for fast video large language models," *arXiv preprint arXiv:2505.21334*, 2025. 3, 5, 12, 18

[147] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Advances in neural information processing systems*, vol. 35, pp. 23 716–23 736, 2022. 3, 12, 19

[148] J. Ye, H. Xu, H. Liu, A. Hu, M. Yan, Q. Qian, J. Zhang, F. Huang, and J. Zhou, "mplug-owl3: Towards long image-sequence understanding in multi-modal large language models," *arXiv preprint arXiv:2408.04840*, 2024. 3, 5, 13, 17, 21

[149] S. Yan, J. Han, J. Tsai, H. Xue, R. Fang, L. Hong, Z. Guo, and R. Zhang, "Crosslmm: Decoupling long video sequences from lmms via dual cross-attention mechanisms," *arXiv preprint arXiv:2505.17020*, 2025. 3, 13, 15

[150] X. Ye, Y. Gan, X. Huang, Y. Ge, and Y. Tang, "Voco-llama: Towards vision compression with large language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 29 836–29 846. 3, 5, 13

[151] J. Chen, L. Ye, J. He, Z.-Y. Wang, D. Khashabi, and A. Yuille, "Efficient large multi-modal models via visual context compression," *Advances in Neural Information Processing Systems*, vol. 37, pp. 73 986–74 007, 2024. 3, 13

[152] K. Tao, C. Qin, H. You, Y. Sui, and H. Wang, "Dycoke: Dynamic compression of tokens for fast video large language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 18 992–19 001. 3, 13, 14, 15, 18

[153] J. Wang, Z. Liu, Y. Rao, and J. Lu, "Sparsemm: Head sparsity emerges from visual concept responses in mllms," *arXiv preprint arXiv:2506.05344*, 2025. 3, 5, 13, 19

[154] M. Kim, K. Shim, J. Choi, and S. Chang, "Infinipot-v: Memory-constrained kv cache compression for streaming video understanding," *arXiv preprint arXiv:2506.15745*, 2025. 3, 13

[155] Z. Ning, G. Liu, Q. Jin, W. Ding, M. Guo, and J. Zhao, "Livevlm: Efficient online video understanding via streaming-oriented kv cache and retrieval," *arXiv preprint arXiv:2505.15269*, 2025. 3, 13

[156] Y. Yang, Z. Zhao, S. N. Shukla, A. Singh, S. K. Mishra, L. Zhang, and M. Ren, "Streammem: Query-agnostic kv cache memory for streaming video understanding," *arXiv preprint arXiv:2508.15717*, 2025. 3, 13

[157] A. Wang, F. Sun, H. Chen, Z. Lin, J. Han, and G. Ding, "[cls] token tells everything needed for training-free efficient mllms," *arXiv preprint arXiv:2412.05819*, 2024. 3, 7

[158] Y. Liu, Y. Wang, B. Shi, X. Zhang, W. Dai, C. Li, H. Xiong, and Q. Tian, "Meteor: Multi-encoder collaborative token pruning for efficient vision language models," *arXiv preprint arXiv:2507.20842*, 2025. 3, 5, 6, 7

[159] Y. Jin, J. Li, Y. Liu, T. Gu, K. Wu, Z. Jiang, M. He, B. Zhao, X. Tan, Z. Gan *et al.*, "Efficient multimodal large language models: A survey," *arXiv preprint arXiv:2405.10739*, 2024. 2

[160] G. Shinde, A. Ravi, E. Dey, S. Sakib, M. Rampure, and N. Roy, "A survey on efficient vision-language models," *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 15, no. 3, p. e70036, 2025. 2

[161] P. Nguyen and N.-M. Cheung, "Token compression meets compact vision transformers: A survey and comparative evaluation for edge ai," *arXiv preprint arXiv:2507.09702*, 2025. 2, 6

[162] K. Shao, K. Tao, K. Zhang, S. Feng, M. Cai, Y. Shang, H. You, C. Qin, Y. Sui, and H. Wang, "When tokens talk too much: A survey of multimodal long-context token compression across images, videos, and audios," *arXiv preprint arXiv:2507.20198*, 2025. 2

[163] OpenAI., "Gpt-4 technical report," 2023. 3

[164] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *NeurIPS*, 2020. 3

[165] R. Taori, I. Gulrajani, T. Zhang, Y. Dubois, X. Li, C. Guestrin, P. Liang, and T. B. Hashimoto, "Stanford alpaca: An instruction-following llama model," 2023. 3

[166] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023. 3

[167] X. Zhai, B. Mustafa, A. Kolesnikov, and L. Beyer, "Sigmoid loss for language image pre-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11 975–11 986. 3

[168] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PmLR, 2021, pp. 8748–8763. 3

[169] R. Bavishi, E. Elsen, C. Hawthorne, M. Nye, A. Odena, A. Somani, and S. Taşırlar, "Introducing our multimodal models," 2023. 3

[170] C. Team, "Chameleon: Mixed-modal early-fusion foundation models," *arXiv preprint arXiv:2405.09818*, 2024. 3

[171] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017. 4

[172] J. Li, D. Li, S. Savarese, and S. C. H. Hoi, "BLIP-2: bootstrapping language-image pre-training with frozen image encoders and large language models," in *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202, 2023, pp. 19 730–19 742. 5

[173] E. Song, W. Chai, G. Wang, Y. Zhang, H. Zhou, F. Wu, H. Chi, X. Guo, T. Ye, Y. Zhang *et al.*, "Moviechat: From dense token to sparse memory for long video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 221–18 232. 5, 15, 16

[174] L. Shen, T. Hao, T. He, S. Zhao, Y. Zhang, P. Liu, Y. Bao, and G. Ding, "Tempme: Video temporal token merging for efficient text-video retrieval," *arXiv preprint arXiv:2409.01156*, 2024. 5, 15, 16

[175] Z. Wang, D. Gong, S. Wang, Z. Huang, and Y. Luo, "Is less more? exploring token condensation as training-free test-time adaptation," *arXiv preprint arXiv:2410.14729*, 2024. 5, 18

[176] K. Y. Li, S. Goyal, J. D. Semedo, and J. Z. Kolter, "Inference optimal vlms need fewer visual tokens and more parameters," *arXiv preprint arXiv:2411.03312*, 2024. 5, 10, 17, 19

[177] X. Ye, Y. Gan, Y. Ge, X.-P. Zhang, and Y. Tang, "Atp-llava: Adaptive token pruning for large vision language models," in

Proceedings of the Computer Vision and Pattern Recognition Conference, 2025, pp. 24 972–24 982. 5, 12

[178] C. Yang, X. Dong, X. Zhu, W. Su, J. Wang, H. Tian, Z. Chen, W. Wang, L. Lu, and J. Dai, "Pvc: Progressive visual token compression for unified image and video processing in large vision-language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 24 939–24 949. 5, 16

[179] X. Huang, H. Zhou, and K. Han, "Prunevid: Visual token pruning for efficient video large language models," 2024. [Online]. Available: https://arxiv.org/abs/2412.16117 5, 15

[180] X. Li, Y. Wang, J. Yu, X. Zeng, Y. Zhu, H. Huang, J. Gao, K. Li, Y. He, C. Wang *et al.*, "Videochat-flash: Hierarchical compression for long-context video modeling," *arXiv preprint arXiv:2501.00574*, 2024. 5, 7, 18

[181] S. Zhang, Q. Fang, Z. Yang, and Y. Feng, "Llava-mini: Efficient image and video large multimodal models with one vision token," *arXiv preprint arXiv:2501.03895*, 2025. 5, 9, 10, 17

[182] J. Li, J. Fan, F. Tang, G. Huang, S. Zhu, S. Liu, N. Xie, W. Liu, and Y. Liao, "Fcot-vl: Advancing text-oriented large vision-language models with efficient visual token compression," *arXiv preprint arXiv:2502.18512*, 2025. 5, 19

[183] Z. Wen, Y. Gao, S. Wang, J. Zhang, Q. Zhang, W. Li, C. He, and L. Zhang, "Stop looking for important tokens in multimodal language models: Duplication matters more," *arXiv preprint arXiv:2502.11494*, 2025. 5, 17, 18, 22

[184] L. Shen, G. Gong, T. He, Y. Zhang, P. Liu, S. Zhao, and G. Ding, "Fastvid: Dynamic density pruning for fast video large language models," *arXiv preprint arXiv:2503.11187*, 2025. 5

[185] W. Zeng, Z. Huang, K. Ji, and Y. Yan, "Skip-vision: Efficient and scalable acceleration of vision-language models via adaptive token skipping," *arXiv preprint arXiv:2503.21817*, 2025. 5

[186] R. Luo, R. Shan, L. Chen, Z. Liu, L. Wang, M. Yang, and X. Xia, "Vcm: Vision concept modeling based on implicit contrastive learning with vision-language instruction fine-tuning," *arXiv preprint arXiv:2504.19627*, 2025. 5, 10, 19, 22

[187] D. Li, Z. Yang, and S. Lu, "Todre: Visual token pruning via diversity and task awareness for efficient large vision-language models," *arXiv preprint arXiv:2505.18757*, 2025. 5

[188] K. Li, X. Chen, C. Gao, Y. Li, and X. Chen, "Balanced token pruning: Accelerating vision language models beyond local optimization," *arXiv preprint arXiv:2505.22038*, 2025. 5, 11

[189] Y. Zhang, Y. Lu, T. Wang, F. Rao, Y. Yang, and L. Zhu, "Flexselect: Flexible token selection for efficient long video understanding," *arXiv preprint arXiv:2506.00993*, 2025. 5, 18

[190] S. Yang, J. Li, X. Lai, B. Yu, H. Zhao, and J. Jia, "Visionthink: Smart and efficient vision language model via reinforcement learning," *arXiv preprint arXiv:2507.13348*, 2025. 5, 22

[191] S. Yin, C. Fu, S. Zhao, K. Li, X. Sun, T. Xu, and E. Chen, "A survey on multimodal large language models," *National Science Review*, vol. 11, no. 12, p. nwae403, 2024. 4

[192] J. Wu, W. Gan, Z. Chen, S. Wan, and P. S. Yu, "Multimodal large language models: A survey," in *2023 IEEE International Conference on Big Data (BigData)*. IEEE, 2023, pp. 2247–2256. 4

[193] L. Yao, H. Wu, K. Ouyang, Y. Zhang, C. Xiong, B. Chen, X. Sun, and J. Li, "Generative frame sampler for long video understanding," *arXiv preprint arXiv:2503.09146*, 2025. 4, 16, 17

[194] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby *et al.*, "Dinov2: Learning robust visual features without supervision," *arXiv preprint arXiv:2304.07193*, 2023. 7

[195] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763. 7

[196] H. Wu, M. Tang, X. Zheng, and H. Jiang, "When language overrules: Revealing text dominance in multimodal large language models," *arXiv preprint arXiv:2508.10552*, 2025. 7

[197] H. Zhang, Y. Wang, Y. Tang, Y. Liu, J. Feng, and X. Jin, "Flash-vstream: Efficient real-time understanding for long video streams," *arXiv preprint arXiv:2506.23825*, 2025. 8, 15, 16

[198] S. Jie, Y. Tang, J. Guo, Z.-H. Deng, K. Han, and Y. Wang, "Token compensator: Altering inference cost of vision transformer without re-tuning," in *European conference on computer vision*. Springer, 2024, pp. 76–94. 8

[199] X. Liu, Y. Shu, Z. Liu, A. Li, Y. Tian, and B. Zhao, "Video-xl-pro: Reconstructive token compression for extremely long video understanding," *arXiv preprint arXiv:2503.18478*, 2025. 8, 16

[200] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695. 8

[201] W. Dai, N. Lee, B. Wang, Z. Yang, Z. Liu, J. Barker, T. Rintamaki, M. Shoeybi, B. Catanzaro, and W. Ping, "Nvlm: Open frontier-class multimodal llms," 2024. [Online]. Available: https://arxiv.org/abs/2409.11402 9, 18

[202] Z. Xia, X. Pan, S. Song, L. E. Li, and G. Huang, "Vision transformer with deformable attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 4794–4803. 10

[203] R. Liao, C. Zhao, J. Li, W. Feng, Y. Lyu, B. Chen, and H. Yang, "Catp: Cross-attention token pruning for accuracy preserved multimodal model inference," in *2025 IEEE Conference on Artificial Intelligence (CAI)*. IEEE, 2025, pp. 1100–1104. 10

[204] J. Han, L. Du, Y. Wu, X. Zhou, H. Du, and W. Zheng, "Adafv: Rethinking of visual-language alignment for vlm acceleration," *arXiv preprint arXiv:2501.09532*, 2025. 10

[205] L. Chen, H. Zhao, T. Liu, S. Bai, J. Lin, C. Zhou, and B. Chang, "An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models," in *European Conference on Computer Vision*. Springer, 2024, pp. 19–35. 11

[206] L. Xing, Q. Huang, X. Dong, J. Lu, P. Zhang, Y. Zang, Y. Cao, C. He, J. Wang *et al.*, "Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction," *arXiv preprint arXiv:2410.17247*, 2024. 11

[207] Y. Zhu, C. Xie, S. Liang, B. Zheng, and S. Guo, "Focusllava: A coarse-to-fine approach for efficient and effective visual token compression," *arXiv preprint arXiv:2411.14228*, 2024. 11

[208] X. Huang, H. Zhou, and K. Han, "Prunevid: Visual token pruning for efficient video large language models," *arXiv preprint arXiv:2412.16117*, 2024. 11

[209] B. Cheng, Y. Ma, L. Wu, S. Liu, A. Ma, X. Wu, D. Leng, and Y. Yin, "Hico: Hierarchical controllable diffusion model for layout-to-image generation," *arXiv preprint arXiv:2410.14324*, 2024. 11

[210] T. Fu, T. Liu, Q. Han, G. Dai, S. Yan, H. Yang, X. Ning, and Y. Wang, "Framefusion: Combining similarity and importance for video token reduction on large visual language models," *arXiv preprint arXiv:2501.01986*, 2024. 11

[211] J. Lee, K. Xuan, C. Ekbote, S. Polisetty, Y. R. Fung, and P. P. Liang, "Tamp: Token-adaptive layerwise pruning in multimodal large language models," *arXiv preprint arXiv:2504.09897*, 2025. 11

[212] W. Ye, Q. Wu, W. Lin, and Y. Zhou, "Fit and prune: Fast and training-free visual token pruning for multi-modal large language models," 2024. [Online]. Available: https://arxiv.org/abs/2409.10197 11

[213] W. Zhang, Z. Zhu, N. Li, K. Liu, and Y. Liu, "Adaptinfer: Adaptive token pruning for vision-language model inference with dynamical text guidance," *arXiv preprint arXiv:2508.06084*, 2025. 11

[214] A. Li, Y. Duan, J. Zhang, C. Ma, Y. Xie, G. Carneiro, M. Yaqub, and H. Wang, "Transprune: Token transition pruning for efficient large vision-language model," *arXiv preprint arXiv:2507.20630*, 2025. 11

[215] R. Xu, Y. Wang, Y. Luo, and B. Du, "Rethinking visual token reduction in lvlms under cross-modal misalignment," *arXiv preprint arXiv:2506.22283*, 2025. 11, 12

[216] F. Sun, L. Shen, H. Chen, S. Zhao, J. Han, and G. Ding, "Adatp: Attention-debiased token pruning for video large language models," *arXiv preprint arXiv:2505.20100*, 2025. 12

[217] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré, "Flashattention: Fast and memory-efficient exact attention with io-awareness," *Advances in neural information processing systems*, vol. 35, pp. 16 344–16 359, 2022. 12

[218] Q. Wang, H. Ye, M.-Y. Chung, Y. Liu, Y. Lin, M. Kuo, M. Ma, J. Zhang, and Y. Chen, "Corematching: A co-adaptive sparse inference framework with token and neuron pruning for comprehensive acceleration of vision-language models," *arXiv preprint arXiv:2505.19235*, 2025. 12

[219] R. Pei, W. Sun, Z. Fu, and J. Wang, "Greedyprune: Retenting critical visual token set for large vision language models," *arXiv preprint arXiv:2506.13166*, 2025. 12

[220] L. Meng, H. Li, B.-C. Chen, S. Lan, Z. Wu, Y.-G. Jiang, and S.-N. Lim, "Adavit: Adaptive vision transformers for efficient image recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12 309–12 318. 12

[221] E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," *arXiv preprint arXiv:1611.01144*, 2016. 12

[222] Y. Wen, Q. Cao, Q. Fu, S. Mehta, and M. Najibi, "Efficient vision-language models by summarizing visual tokens into compact registers," *arXiv preprint arXiv:2410.14072*, 2024. 13

[223] G. Xiao, Y. Tian, B. Chen, S. Han, and M. Lewis, "Efficient streaming language models with attention sinks," *arXiv preprint arXiv:2309.17453*, 2023. 13

[224] S. Ge, Y. Zhang, L. Liu, M. Zhang, J. Han, and J. Gao, "Model tells you what to discard: Adaptive kv cache compression for llms," *arXiv preprint arXiv:2310.01801*, 2023. 13

[225] Z. Zhang, Y. Sheng, T. Zhou, T. Chen, L. Zheng, R. Cai, Z. Song, Y. Tian, C. Ré, C. Barrett *et al.*, "H2o: Heavy-hitter oracle for efficient generative inference of large language models," *Advances in Neural Information Processing Systems*, vol. 36, pp. 34 661–34 710, 2023. 13

[226] Z. Liu, B. Liu, J. Wang, Y. Dong, G. Chen, Y. Rao, R. Krishna, and J. Lu, "Efficient inference of vision instruction-following models with elastic cache," *arXiv preprint arXiv:2407.18121*, 2024. 13

[227] D. Zheng, M. Huang, B. Jiang, H. Hu, and X. Chen, "Towards lossless ultimate vision token compression for vlms," *arXiv preprint arXiv:2512.09010*, 2025. 14

[228] M. Maaz, H. Rasheed, S. Khan, and F. S. Khan, "Video-chatgpt: Towards detailed video understanding via large vision and language models," *arXiv preprint arXiv:2306.05424*, 2023. 15, 19

[229] M. S. Ryoo, H. Zhou, S. Kendre, C. Qin, L. Xue, M. Shu, J. Park, K. Ranasinghe, S. Savarese, R. Xu *et al.*, "xgen-mm-vid (blip-3-video): You only need 32 tokens to represent a video even in vlms," *arXiv preprint arXiv:2410.16267*, 2024. 15, 16

[230] J. Jiang, X. Li, Z. Liu, M. Li, G. Chen, Z. Li, D.-A. Huang, G. Liu, Z. Yu, K. Keutzer *et al.*, "Token-efficient long video understanding for multimodal llms," *arXiv preprint arXiv:2503.04130*, 2025. 15, 16

[231] W. Chai, E. Song, Y. Du, C. Meng, V. Madhavan, O. Bar-Tal, J.-N. Hwang, S. Xie, and C. D. Manning, "Auroracap: Efficient, performant video detailed captioning and a new benchmark," *arXiv preprint arXiv:2410.03051*, 2024. 15

[232] Y. Weng, M. Han, H. He, X. Chang, and B. Zhuang, "Longvlm: Efficient long video understanding via large language models," in *European Conference on Computer Vision*. Springer, 2024, pp. 453–470. 15, 19

[233] Z. Liu, C.-W. Xie, P. Li, L. Zhao, L. Tang, Y. Zheng, C. Liu, and H. Xie, "Hybrid-level instruction injection for video token compression in multi-modal large language models," in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 8568–8578. 15, 16

[234] J. Qi, Y. Yao, Y. Bai, B. Xu, J. Li, Z. Liu, and T.-S. Chua, "An lmm for efficient video understanding via reinforced compression of video cubes," *arXiv preprint arXiv:2504.15270*, 2025. 15, 16

[235] M. Xu, M. Gao, Z. Gan, H.-Y. Chen, Z. Lai, H. Gang, K. Kang, and A. Dehghan, "Slowfast-llava: A strong training-free baseline for video large language models," *arXiv preprint arXiv:2407.15841*, 2024. 15, 16

[236] Y. Zhang, J. Wu, W. Li, B. Li, Z. Ma, Z. Liu, and C. Li, "Llava-video: Video instruction tuning with synthetic data," *Transactions on Machine Learning Research*, 2025. 15, 16, 19

[237] B. Yang, B. Wen, B. Ding, C. Liu, C. Chu, C. Song, C. Rao, C. Yi, D. Li, D. Zang *et al.*, "Kwai keye-vl 1.5 technical report," *arXiv preprint arXiv:2509.01563*, 2025. 15, 16

[238] X. Lan, Y. Yuan, Z. Jie, and L. Ma, "Vidcompress: Memory-enhanced temporal compression for video understanding in large language models," *arXiv preprint arXiv:2410.11417*, 2024. 15, 16

[239] Y. Liu, S. Li, Y. Liu, Y. Wang, S. Ren, L. Li, S. Chen, X. Sun, and L. Hou, "Tempcompass: Do video llms really understand videos?" *arXiv preprint arXiv:2403.00476*, 2024. 14

[240] Z. Shangguan, C. Li, Y. Ding, Y. Zheng, Y. Zhao, T. Fitzgerald, and A. Cohan, "Tomato: Assessing visual temporal reasoning capabilities in multimodal foundation models," *arXiv preprint arXiv:2410.23266*, 2024. 14

[241] Y. Liu, Z. Ma, Z. Qi, Y. Wu, Y. Shan, and C. W. Chen, "Et bench: Towards open-ended event-level video-language understanding," *Advances in Neural Information Processing Systems*, vol. 37, pp. 32 076–32 110, 2024. 14

[242] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in *First conference on language modeling*, 2024. 15

[243] G. Zhang, J. Liu, S. Cao, X. Zhao, K. Zhao, K. Ma, and L. Wang, "Dynamic and compressive adaptation of transformers from images to videos," *arXiv preprint arXiv:2408.06840*, 2024. 15

[244] S.-H. Lee, J. Wang, Z. Zhang, D. Fan, and X. Li, "Video token merging for long-form video understanding," *arXiv preprint arXiv:2410.23782*, 2024. 15

[245] J. Cho, J. Lee, M. Hayat, K. Hwang, F. Porikli, and S. Choi, "Floc: Facility location-based efficient visual token compression for long video understanding," *arXiv preprint arXiv:2511.00141*, 2025. 15

[246] J. Gao, C. Sun, Z. Yang, and R. Nevatia, "Tall: Temporal activity localization via language query," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5267–5275. 16

[247] X. Wei, X. Liu, Y. Zang, X. Dong, P. Zhang, Y. Cao, J. Tong, H. Duan, Q. Guo, J. Wang *et al.*, "Videorope: What makes for good video rotary position embedding?" *arXiv preprint arXiv:2502.05173*, 2025. 16

[248] A. Gu and T. Dao, "Mamba: Linear-time sequence modeling with selective state spaces," in *First Conference on Language Modeling*, 2024. 16

[249] S. Bai, Y. Cai, R. Chen, K. Chen, X. Chen, Z. Cheng, L. Deng, W. Ding, C. Gao, C. Ge, W. Ge, Z. Guo, Q. Huang, J. Huang, F. Huang, B. Hui, S. Jiang, Z. Li, M. Li, M. Li, K. Li, Z. Lin, J. Lin, X. Liu, J. Liu, C. Liu, Y. Liu, D. Liu, S. Liu, D. Lu, R. Luo, C. Lv, R. Men, L. Meng, X. Ren, X. Ren, S. Song, Y. Sun, J. Tang, J. Tu, J. Wan, P. Wang, P. Wang, Q. Wang, Y. Wang, T. Xie, Y. Xu, H. Xu, J. Xu, Z. Yang, M. Yang, J. Yang, A. Yang, B. Yu, F. Zhang, H. Zhang, X. Zhang, B. Zheng, H. Zhong, J. Zhou, F. Zhou, J. Zhou, Y. Zhu, and K. Zhu, "Qwen3-vl technical report," *arXiv preprint arXiv:2511.21631*, 2025. 16

[250] S. Chen, X. Lan, Y. Yuan, Z. Jie, and L. Ma, "Timemarker: A versatile video-llm for long and short video understanding with superior temporal localization ability," *arXiv preprint arXiv:2411.18211*, 2024. 16

[251] Y. Wu, X. Hu, Y. Sun, Y. Zhou, W. Zhu, F. Rao, B. Schiele, and X. Yang, "Number it: Temporal grounding videos like flipping manga," *arXiv preprint arXiv:2411.10332*, 2024. 16

[252] H. Zhang, Y. Wang, Y. Tang, Y. Liu, J. Feng, J. Dai, and X. Jin, "Flash-vstream: Memory-based real-time understanding for long video streams," *arXiv preprint arXiv:2406.08085*, 2024. 16

[253] S. Gurukar and A. Kadav, "Long-vmnet: Accelerating long-form video understanding via fixed memory," *arXiv preprint arXiv:2503.13707*, 2025. 17

[254] X. Wang, Q. Si, J. Wu, S. Zhu, L. Cao, and L. Nie, "Retake: Reducing temporal and knowledge redundancy for long video understanding," *arXiv preprint arXiv:2412.20504*, 2024. 17

[255] B. Xu, Z. Xiao, J. Li, J. Ju, Z. Luo, J. Luan, and Q. Jin, "Timeviper: A hybrid mamba-transformer model for efficient long video understanding," *arXiv preprint arXiv:2511.16595*, 2025. 17

[256] S. Wang, T. Niu, R. Yang, D. Liu, X. He, Z. Wen, C. He, X. Hu, and L. Zhang, "Videocompressa: Data-efficient video understanding via joint temporal compression and spatial reconstruction," *arXiv preprint arXiv:2511.18831*, 2025. 17

[257] J. Li, D. Li, S. Savarese, and S. Hoi, "Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models," in *International conference on machine learning*. PMLR, 2023, pp. 19 730–19 742. 17, 19

[258] Z. Ning, J. Zhao, Q. Jin, W. Ding, and M. Guo, "Inf-mllm: Efficient streaming inference of multimodal large language models on a single gpu," *arXiv preprint arXiv:2409.09086*, 2024. 17

[259] J. Lin, Z. Fang, C. Chen, Z. Wan, F. Luo, P. Li, Y. Liu, and M. Sun, "Streamingbench: Assessing the gap for mllms to achieve streaming video understanding," *arXiv preprint arXiv:2411.03628*, 2024. 17

[260] H. Xiong, Z. Yang, J. Yu, Y. Zhuge, L. Zhang, J. Zhu, and H. Lu, "Streaming video understanding and multi-round interaction with memory-enhanced knowledge," *arXiv preprint arXiv:2501.13468*, 2025. 17

[261] Z. Huang, X. Li, J. Li, J. Wang, X. Zeng, C. Liang, T. Wu, X. Chen, L. Li, and L. Wang, "Online video understanding: A comprehensive benchmark and memory-augmented method," *arXiv preprint arXiv:2501.00584*, 2024. 17

[262] P. Zhang, X. Dong, Y. Cao, Y. Zang, R. Qian, X. Wei, L. Chen, Y. Li, J. Niu, S. Ding *et al.*, "Internlm-xcomposer2. 5-omnilive: A comprehensive multimodal system for long-term streaming

video and audio interactions," *arXiv preprint arXiv:2412.09596*, 2024. 17

[263] L. Yuan, J. Wang, H. Sun, Y. Zhang, and Y. Lin, "Tarsier2: Advancing large vision-language models from detailed video description to comprehensive video understanding," *arXiv preprint arXiv:2501.07888*, 2025. 17

[264] L. Yao, Y. Zhang, Z. Wang, X. Hou, T. Ge, Y. Jiang, X. Sun, and Q. Jin, "Edit as you wish: Video caption editing with multi-grained user control," in *Proceedings of the 32nd ACM International Conference on Multimedia*, 2024, pp. 1924–1933. 17

[265] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," in *Advances in Neural Information Processing Systems*, 2022. 17

[266] Z. Liu, P. Han, H. Yu, H. Li, and J. You, "Time-r1: Towards comprehensive temporal reasoning in llms," *arXiv preprint arXiv:2505.13508*, 2025. 17

[267] X. Zeng, K. Li, C. Wang, X. Li, T. Jiang, Z. Yan, S. Li, Y. Shi, Z. Yue, Y. Wang *et al.*, "Timesuite: Improving mllms for long video understanding via grounded tuning," *arXiv preprint arXiv:2410.19702*, 2024. 17

[268] H. Wu, D. Li, B. Chen, and J. Li, "Longvideobench: A benchmark for long-context interleaved video-language understanding," *ArXiv preprint*, vol. abs/2407.15754, 2024. 17

[269] Q. Zhang, M. Liu, L. Li, M. Lu, Y. Zhang, J. Pan, Q. She, and S. Zhang, "Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms," *arXiv preprint arXiv:2506.10967*, 2025. 18

[270] C. Lv, B. Zhang, Y. Yong, R. Gong, Y. Huang, S. Gu, J. Wu, Y. Shi, J. Guo, and W. Wang, "Llmc+: Benchmarking vision-language model compression with a plug-and-play toolkit," *arXiv preprint arXiv:2508.09981*, 2025. 17

[271] X. Liu, Y. Wang, J. Ma, and L. Zhang, "Video compression commander: Plug-and-play inference acceleration for video large language models," *arXiv preprint arXiv:2505.14454*, 2025. 18

[272] Q. Cao, B. Paranjape, and H. Hajishirzi, "Pumer: Pruning and merging tokens for efficient vision language models," *arXiv preprint arXiv:2305.17530*, 2023. 18

[273] X. Wu, F. Zeng, X. Wang, and X. Chen, "Ppt: Token pruning and pooling for efficient vision transformers," *arXiv preprint arXiv:2310.01812*, 2023. 18

[274] S. Bai, K. Chen, X. Liu, J. Wang, W. Ge, S. Song, K. Dang, P. Wang, S. Wang, J. Tang *et al.*, "Qwen2. 5-vl technical report," *arXiv preprint arXiv:2502.13923*, 2025. 19

[275] Q. Ye, H. Xu, G. Xu, J. Ye, M. Yan, Y. Zhou, J. Wang, A. Hu, P. Shi, Y. Shi, C. Jiang, C. Li, Y. Xu, H. Chen, J. Tian, Q. Qi, J. Zhang, and F. Huang, "mplug-owl: Modularization empowers large language models with multimodality," 2023. 19

[276] H. Zhang, X. Li, and L. Bing, "Video-llama: An instruction-tuned audio-visual language model for video understanding," *arXiv preprint arXiv:2306.02858*, 2023. 19

[277] D. Guo, F. Wu, F. Zhu, F. Leng, G. Shi, H. Chen, H. Fan, J. Wang, J. Jiang, J. Wang *et al.*, "Seed1. 5-vl technical report," *arXiv preprint arXiv:2505.07062*, 2025. 19

[278] Z. Chen, J. Wu, W. Wang, W. Su, G. Chen, S. Xing, M. Zhong, Q. Zhang, X. Zhu, L. Lu, B. Li, P. Luo, T. Lu, Y. Qiao, and J. Dai, "Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks," *ArXiv preprint*, vol. abs/2312.14238, 2023. 19

[279] H. Xiao, F. Zhou, X. Liu, T. Liu, Z. Li, X. Liu, and X. Huang, "A comprehensive survey of large language models and multimodal large language models in medicine," *Information Fusion*, p. 102888, 2024. 21

[280] Y. Hu, C. Xu, B. Lin, W. Yang, and Y. Y. Tang, "Medical multimodal large language models: A systematic review," *Intelligent Oncology*, 2025. 21

[281] Y. Ding, S. Luo, Y. Dai, Y. Jiang, Z. Li, G. Martin, and Y. Peng, "A survey on mllm-based visually rich document understanding: Methods, challenges, and emerging trends," *arXiv preprint arXiv:2507.09861*, 2025. 21

[282] F. Wang, H. Wang, Z. Guo, D. Wang, Y. Wang, M. Chen, Q. Ma, L. Lan, W. Yang, J. Zhang *et al.*, "Xlrs-bench: Could your multimodal llms understand extremely large ultra-high-resolution remote sensing imagery?" in *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025, pp. 14 325–14 336. 21

[283] J. Luo, Y. Zhang, X. Yang, K. Wu, Q. Zhu, L. Liang, J. Chen, and Y. Li, "When large vision-language model meets large remote sensing imagery: Coarse-to-fine text-guided token pruning," *arXiv preprint arXiv:2503.07588*, 2025. 21

[284] Y. Niu, Z. Song, Q. Luo, G. Chen, M. Ma, and F. Li, "Atmformer: An adaptive token merging vision transformer for remote sensing image scene classification," *Remote Sensing*, vol. 17, no. 4, p. 660, 2025. 21

[285] J. Chen, Z. Lv, S. Wu, K. Q. Lin, C. Song, D. Gao, J.-W. Liu, Z. Gao, D. Mao, and M. Z. Shou, "Videollm-online: Online video large language model for streaming video," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 18 407–18 418. 22

[286] S. Di, Z. Yu, G. Zhang, H. Li, T. Zhong, H. Cheng, B. Li, W. He, F. Shu, and H. Jiang, "Streaming video question-answering with in-context video kv-cache retrieval," *arXiv preprint arXiv:2503.00540*, 2025. 22

[287] Y. Wang, X. Liu, X. Gui, X. Lin, B. Yang, C. Liao, T. Chen, and L. Zhang, "Accelerating streaming video large language models via hierarchical token compression," *arXiv preprint arXiv:2512.00891*, 2025. 22

[288] L. Lei, J. Gu, X. Ma, C. Tang, J. Chen, and T. Xu, "Generic token compression in multimodal large language models from an explainability perspective," *arXiv preprint arXiv:2506.01097*, 2025. 22

[289] Z. Kong, Y. Li, F. Zeng, L. Xin, S. Messica, X. Lin, P. Zhao, M. Kellis, H. Tang, and M. Zitnik, "Token reduction should go beyond efficiency in generative models–from vision, language to multimodality," *arXiv preprint arXiv:2505.18227*, 2025. 22

[290] X. Zhang, L. Zhu, H. He, S. Zeng, O. Fu, J. Hu, Z. Yao, and Y. Lu, "Adatok: Adaptive token compression with object-aware representations for efficient multimodal llms," *arXiv preprint arXiv:2511.14169*, 2025. 22

[291] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. 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*. Springer, 2014, pp. 740–755. 22

[292] C. Fu, P. Chen, Y. Shen, Y. Qin, M. Zhang, X. Lin, J. Yang, X. Zheng, K. Li, X. Sun, Y. Wu, and R. Ji, "Mme: A comprehensive evaluation benchmark for multimodal large language models," 2024. [Online]. Available: https://arxiv.org/abs/2306.13394 22

[293] Y. Liu, Z. Li, M. Huang, B. Yang, W. Yu, C. Li, X.-C. Yin, C.-L. Liu, L. Jin, and X. Bai, "Ocrbench: on the hidden mystery of ocr in large multimodal models," *Science China Information Sciences*, vol. 67, no. 12, p. 220102, 2024. 23

[294] Y. Shi, H. Wang, W. Xie, H. Zhang, L. Zhao, Y.-F. Zhang, X. Li, C. Fu, Z. Wen, W. Liu *et al.*, "Mme-videoocr: Evaluating ocr-based capabilities of multimodal llms in video scenarios," *arXiv preprint arXiv:2505.21333*, 2025. 23

[295] M. Mathew, D. Karatzas, and C. Jawahar, "Docvqa: A dataset for vqa on document images," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, 2021, pp. 2200–2209. 23

[296] C. Liao, W. Wang, Z. Wen, X. Zheng, Y. Wang, H. He, Y. Lyu, L. Jiang, X. Zou, Y. Fu *et al.*, "Are we using the right benchmark: An evaluation framework for visual token compression methods," *arXiv preprint arXiv:2510.07143*, 2025. 23