Reducing Token Redundancy in LVLMs: A Systematic Review of Token Pruning Methods

Hanzhang Yuan[1] **Mengxuan Hu**[1] **Wenhao Zhang**[1] **Tianlong Wang**[1]
Zhongliang Zhou[2] **Jiasen Lu**[3] **Sheng Li**[1]
[1]School of Data Science, University of Virginia
[2]Merck & Co., Inc., Biometrics Research [3]Apple Inc.
{ynx9zm,qtq7s,bdu8us,jqf8qm,shengli}@virginia.edu
zhongliang.zhou@merck.com jiasen_lu@apple.com

Abstract

Large Vision-Language Models (LVLMs) excel at visual understanding but face severe computational bottlenecks when processing high-resolution images and long videos due to massive visual token counts. Token pruning mitigates this by selectively removing less informative tokens while maintaining performance. However, existing methods vary widely in pruning location (vision encoder vs. LLM decoder), importance criteria (attention vs. similarity vs. learned scores), and application strategy, lacking systematic comparison. This survey presents the first comprehensive review of token pruning for LVLMs. We propose a taxonomy categorizing methods into vision-side, LLM-side, and hybrid paradigms, systematically analyze token selection mechanisms and pruning strategy. We further discuss evaluation protocols and identify key challenges including prompt-adaptive pruning and hardware-aware design. Our survey provides a structured foundation for this rapidly growing research area.

1 Introduction

Large Vision–Language Models (LVLMs), such as GPT-4V (Achiam et al., 2023), the LLaVA family (Liu et al., 2023a, 2024a,b), Qwen-VL (Wang et al., 2024), and the BLIP family (Li et al., 2022), have demonstrated remarkable capabilities in multimodal understanding, reasoning, and generation. By integrating powerful visual encoders with large language models, LVLMs enable a wide range of applications, including visual question answering (Singh et al., 2019), video understanding (Zhou et al., 2025), multimodal retrieval (Abootorabi et al., 2025), and agentic reasoning (Yao et al., 2025). These advances make LVLMs a promising foundation for real-world systems that require both perception and reasoning.

However, this performance comes at a significant computational cost. Modern LVLMs typically rely on Vision Transformer (ViT)-based encoders (e.g., CLIP (Radford et al., 2021), SigLIP (Zhai et al., 2023)) that partition an image into a large number of patch tokens. The number of visual tokens scales rapidly with image resolution, video length, and multi-image inputs, leading to long multimodal input sequences. During inference, these tokens must be processed by the language model in the *prefill* stage, where full self-attention is computed over all visual and textual tokens. As a result, inference cost scales quadratically with the total token length, making visual tokens a dominant bottleneck and limiting the scalability of LVLMs in latency-sensitive and resource-constrained settings.

A key challenge underlying this inefficiency is **visual token redundancy**: many visual tokens contribute little to the final prediction for a given prompt or task, yet still incur substantial computational and memory cost. Visual token pruning has emerged as a distinct and promising direction that directly targets redundancy at the token level. It select a compact subset of visual tokens from the original visual sequence in order to reduce computational and memory costs while preserving task-relevant information. This process can be implemented through token dropping, masking, routing, replacement, or reweighting mechanisms, and can be applied at different stages of the LVLM pipeline.

Despite the rapid growth of research in this area, there is currently no systematic and comprehensive survey that focuses specifically on visual token pruning in LVLMs. Existing surveys primarily address efficiency in large language models through model compression (Zhu et al., 2024; Cheng et al., 2025), or discuss token compression in multimodal models at a high level without detailed analysis of pruning mechanisms and design choices (Shao et al., 2025). This leaves an important gap in understanding the design space, trade-offs, and practical implications of visual token pruning. In this survey, we aim to fill this gap by providing the first structured and in-depth review of visual token pruning

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 7231–7251
July 2-7, 2026

methods for LVLMs. We organize existing approaches into a principled taxonomy consisting of *vision-side token pruning*, *LLM-side visual token pruning*, and *hybrid vision–LLM pruning*, reflecting where pruning is applied and what signals it relies on. This novel classification highlights fundamental differences in pruning granularity, adaptivity, and computational impact, and provides a unified framework for comparing methods across architectures and tasks. We further analyze token importance estimation strategies, compare pruning with related efficiency techniques, review evaluation protocols, and discuss open challenges and future directions.

The remainder of this article is organized as follows. Section 2 introduces background on LVLMs and visual token pruning. Section 3 reviews pruning methods according to our taxonomy. Section 4 compares pruning with related efficiency techniques and discusses evaluation practices. Section 5 presents widely used benchmarks and experiment protocols. Finally, Sections 6 and 7 discuss future directions and limitations, respectively.

2 Background

In this section, we introduce background knowledge on Large Vision–Language Models and the corresponding problem formulation for Visual Token Pruning in LVLMs.

2.1 Large Vision-Language Models

Given an input image I and a text prompt T, LVLM inference proceeds in three stages. First, a vision encoder E_v maps the image into a sequence of visual tokens, where N equals the patch number:

$$V = E_v(I) = \{v_1, v_2, \ldots, v_N\}, \quad v_i \in \mathbb{R}^d.$$

Second, a multimodal projector f, which project the vision tokens to align with the language space: $H_v = f(V) \in \mathbb{R}^{N \times d}$. Finally, a language model g_θ accepts the concatenation of visual tokens H_v and text tokens $H_t = \{t_1, \ldots, t_M\}$ and autoregressively generates an output sequence $Y = \{y_1, \ldots, y_L\}$:

$$p(Y \mid I, T) = \prod_{\ell=1}^{L} p(y_\ell \mid y_{<\ell}, H_v, H_t).$$

From a computational perspective, inference in LVLMs can be divided into two phases. During the *prefill* stage, the language model processes the entire input sequence, comprising both visual tokens from the vision encoder and textual prompt tokens, in a single forward pass to compute their hidden representations, after which the first output token is generated. This stage incurs high computational of $\mathcal{O}(N_{\text{N+M}}^2)$ and activation memory cost because self-attention scales quadratically with the total number of input tokens. In the subsequent *autoregressive decoding* stage, new tokens are generated one at a time using cached key–value (KV) states; at decoding step ℓ, the forward pass only computes attention for the newly generated token against all previously cached tokens, resulting in linear per-step complexity $\mathcal{O}(N_{\text{N+M}} + \ell)$. Consequently, the prefill stage is substantially more expensive than decoding when the input contains many visual tokens. As a result, reducing the number of visual tokens *before or during prefill* yields significantly larger efficiency gains than pruning strategies that operate only during the decoding stage.

2.2 Visual Token Pruning

Visual token pruning refers to the process of selecting a compact subset of visual tokens from the original visual sequence in order to reduce computational and memory costs while preserving minimal performance loss through reserving task-relevant information.

Formally, given an image encoded as a sequence of N visual tokens $H_v = \{h_1, h_2, \ldots, h_N\}$, $h_i \in \mathbb{R}^d$, pruning constructs a reduced sequence $\tilde{H}_v = \{h_i \mid i \in \mathcal{I}\}$, $\mathcal{I} \subseteq \{1, \ldots, N\}$ with a *token budget* enforced either by a fixed number $|\mathcal{I}| = K$, $K \ll N$, or by a keep ratio $r \in (0, 1]$, $|\mathcal{I}| = \lceil rN \rceil$.

It is important to distinguish visual token pruning from related but different techniques: **token merging** combines similar tokens instead of removing them; **KV-cache pruning** reduces memory and attention cost during decoding by discarding cached key–value states without affecting prefill (Zhang et al., 2023, 2025b); **pooling or condensation** compresses groups of semantically related tokens into compact representations (Han et al., 2025); and **structural weight pruning** removes model parameters (e.g., attention heads or feed-forward blocks) to reduce FLOPs (Liang et al., 2025a). While complementary, these methods operate at different levels of the model and target different efficiency bottlenecks.

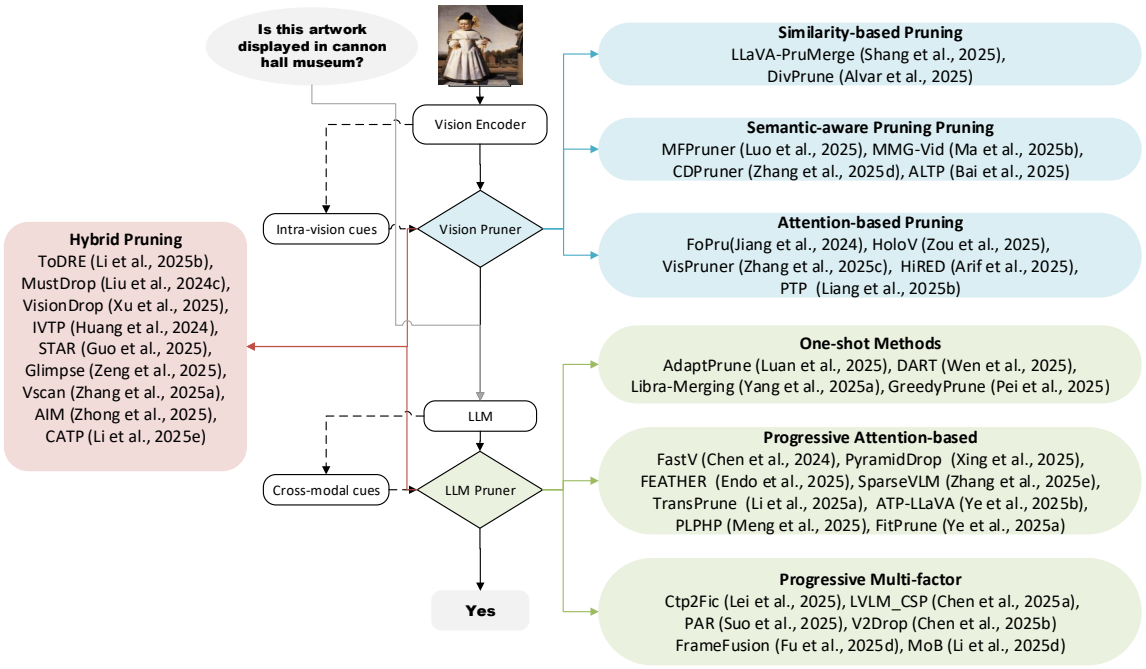

Figure 1: Comparison of three pruning paradigms. Pipelines flow top-to-bottom. Dashed arrows indicate what signals the pruner uses. Green blocks contain the LLM-side pruning methods; blue blocks contain the vision-side pruning methods; red blocks summarize the hybrid pruning methods.

3 LVLMs Token Pruning

In this section, we discuss token pruning strategy based on *where* and *how* the token number is reduced, divided the methods into **vision-side**, **LLM-side** and **hybrid** token pruning.

3.1 Vision-side Token Pruning

Vision-side token pruning operate entirely before cross-modal interaction, typically within or right after the ViT-based visual encoders, and aim to reduce the number of visual tokens injected into the LLM without relying on language guidance. In this work, we define the vision-side pruning as any pruning operation that reduces or transforms prior to the prefill stage.

3.1.1 Similarity-based Pruning

Similarity-based pruning mainly focuses on reducing token redundancy by identifying a minimal subset of tokens with high diversity, such that the selected tokens collectively preserve information close to that of the original token set. LLaVA-PruMerge combines `[cls]` attention sparsity with key-similarity clustering to prune and merge redundant visual tokens (Shang et al., 2025). DivPrune can also be seen as a similarity-based method with a diversity-driven criterion that maximizes the minimum pairwise distance among retained visual tokens (Alvar et al., 2025).

3.1.2 Semantic-aware Pruning

Beyond similarity-based redundancy reduction, another line of work focuses on visual-encoder-side pruning guided by alternative importance criteria. These methods explicitly incorporate task semantics or cross-modal signals to inform token selection, rather than relying solely on visual similarity.

Some approaches leverage text–image relevance as a complementary pruning signal. CDPruner formulates visual token pruning as a diversity maximization problem under textual guidance, using Determinantal Point Processes (DPP) to jointly balance visual token similarity and instruction relevance—measured by cosine similarity between text embeddings and image tokens—within a unified probabilistic framework (Zhang et al., 2025d). MFPruner fuses `[CLS]` attention, token similarity, and instruction relevance and make pruning decision via voting mechanism (Luo et al., 2025). ALTP targets grounded conversational generation by introducing locality-aware pruning, preserving object-centric tokens via region partitioning and density-adaptive token allocation (Bai et al., 2025). In video task, temporal becomes another semantic dimension. MMG-Vid exploits temporal coher-

ence via three-stage pruning: it segments videos by frame similarity, dynamically allocates budgets to maximize marginal gain (reducing allocation for redundant segments), and selects tokens with a strategy which prioritizes tokens novel to the selection history while salient in the current frame (Ma et al., 2025b).

3.1.3 Attention-based Pruning

Attention statistics from the visual encoder have also been widely adopted as importance indicators. FoPru estimates token importance by averaging attention maps across heads and selecting the row or column with higher variance to identify salient tokens, as these tokens may have more influence towards others (Jiang et al., 2024). PTP extends attention-based pruning with a pyramid-style importance modeling strategy by first computing region-level saliency using [CLS] attention to allocate token budgets across spatial regions, then performs token-level selection within each region based on [CLS]–to–patch attention, and finally refines the selected tokens using instruction-aware relevance (Liang et al., 2025b). In contrast, HoloV addresses the over-localization issue inherent in attention-first pruning by combining [CLS] attention for saliency estimation with intra-crop token variance to encourage semantic diversity. It further allocates token budgets adaptively across spatial crops to preserve holistic visual context (Zou et al., 2025). VisPruner performs two-stage pruning after the projector by first keeping important tokens based on [CLS] attention, and then iteratively removes redundant tokens by token similarity (cosine similarity) to retain a diverse complement under a fixed budget (Zhang et al., 2025d). HiRED similarly adopts a two-stage strategy, before the projector, first using early-layer [CLS] attention to estimate content distribution across image partitions, and then selecting top-K informative tokens based on [CLS] attention of layer 22 (Arif et al., 2025).

By aggregating complementary signals, such approaches mitigate common failure modes of single-criterion pruning, such as attention collapse or semantic bias toward dominant visual regions.

3.2 LLM-side Token Pruning

Since vision-side token pruning reduces the number of visual tokens *before the LLM generation process*, it directly lowers the computational cost at the prefill stage. In contrast, LLM-side pruning methods do not modify the visual encoder; instead, they leverage language-model behavior during the prefill or decoding stages, typically by exploiting prompt–visual interactions to identify and drop less informative visual tokens.

3.2.1 One-shot Pruning

This line of work in LLM-side token pruning make the pruning decision for one time.
Several methods exploit token similarity to identify redundancy. Specifically, they identify and prune similar tokens while retaining a diverse subset to preserve information for generation. For example, AdaptPrune reframes token pruning as an adaptive non-maximum suppression process that jointly considers attention, spatial distance, and token similarity, pruning visual tokens for one time in the early layer of LLM (Luan et al., 2025). In contrast, DART reframes the problem from token importance to token duplication. Rather than relying on attention scores, DART removes redundant visual tokens in one early layer by measuring embedding similarity to a small set of pivot tokens, ensuring diverse token retention (Wen et al., 2025). Libra-Merging is a hybrid pruning and merge method, also resolving the importance–redundancy dilemma by selecting representative tokens from spatial intervals and performing similarity-aware grouped merging with compensation tokens (Yang et al., 2025a). GreedyPrune formulates token selection as a combinatorial optimization problem that jointly optimizes semantic saliency and visual diversity, solving it via an efficient greedy strategy and finish the token pruning after the first layer for only one time (Pei et al., 2025).

3.2.2 Progressive Pruning

Progressive pruning reduces computation by exploiting the fact that token utility evolves during inference, allowing different pruning operations to be applied progressively rather than in a single step.
Attention-based. In shallow layers (layers 1-2), attention in relatively balances across all token types, with image tokens actively aggregating visual information through self-attention, while in deeper layers, attention becomes extremely imbalances – system prompts receive 472× higher attention efficiency than image tokens, capturing 85% of total attention (Chen et al., 2024).

Based on this observation, FastV removes visual tokens with persistently low cross-modal attention after layer 2, as it noticed in deeper LVLM layers the visual tokens receives much less attention

than system prompts in deep layers, significantly reducing computation with minimal accuracy loss (Chen et al., 2024). PyramidDrop extended the idea by exploiting the layer-wise growth of visual redundancy by retaining all tokens in shallow layers and progressively dropping less informative tokens in deeper layers (Xing et al., 2025). To eliminate the positional bias within LLM layers, FEATHER uses RoPE-free attention from the last text token as the primary criteria, ensembles it with uniform sampling criteria in early layers for coverage, and applies aggressive pruning with the refined attention criteria in later layers (Endo et al., 2025). While FastV successfully identifies inefficient visual attention in deep layers, its pruning strategy remains fundamentally text-agnostic, SparseVLM addresses this limitation through text-aware guidance, arguing that pruning should be question-adaptive (Zhang et al., 2025e).

Both FastV and SparseVLM posit that visual tokens become progressively less important in deeper layers, justifying increasingly aggressive pruning. However, this assumption of monotonic attention decline lacks empirical validation across diverse LVLM architectures. PLPHP discovers the Vision Token Re-attention phenomenon where visual attention resurges in deep layers and further introduces per-layer, per-head retention rates, enabling more adaptive pruning strategy (Meng et al., 2025). FitPrune is also a progressive pruning method, which take the self-attention score and cross-attention score as criterion and prunes the visual tokens during every layer during decoding time (Ye et al., 2025a).

Besides using attention score itself as an importance metric, there are also some variant based on attention. TransPrune evaluates token importance via Token Transition Variation through layers of token's self attention, capturing both magnitude and directional changes across layers, and enables training-free, multi-stage pruning guided by representation dynamics (Li et al., 2025a). Related method such as ATP-LLaVA Leverages dual criteria of redundancy scores (averaged self-modal and cross-modal attention) and spatial scores (2D RoPE-enhanced uniform sampling) with learnable thresholds to achieve instance-wise and layer-wise adaptive pruning (Ye et al., 2025b).

Multi-factor. Progressive pruning can also be lead by different pruning criteria through the stages with different rentention goals. Multi-factor methods jointly consider multiple pruning goals such as diversity and importance, and use different computation algorithms to balance between criteria. V2Drop adopts a variation-aware criterion, measuring token importance by representation changes between consecutive transformer layers. Tokens exhibiting low variation are considered redundant and progressively removed by three times during inference, enabling pruning without relying on attention statistics (Chen et al., 2025b). Ctp2Fic combines shallow-layer text-guided pruning with deep-layer semantic clustering in a coarse-to-fine manner (Lei et al., 2025). MoB formulates visual token pruning as a bi-objective covering problem and theoretically characterizes the trade-off between visual preservation and prompt alignment under fixed budgets, which first choose visual tokens nearest to prompts tokens and then choose the farthest tokens to the chosen tokens to increase the diversity (Li et al., 2025d). LVLM_CSP adopts a three-stage progressive pruning framework across LLM decoder layers with first the clustering stage using Seg-First criteria or `[cls]` attention scores, scattering stage reactivates all tokens for fine-grained reasoning, and finally retains top tokens ranked by `[SEG]` token's attention scores (Chen et al., 2025a).

In video tasks, FrameFusion performs similarity-based token merging at shallow layers, where visual redundancy across adjacent frames is most pronounced, and permanently removes merged tokens. At deeper layers, where semantic importance becomes more discriminative, FrameFusion applies importance-based pruning using cumulative self-attention scores to further satisfy computational budgets (Fu et al., 2025d).

Besides the training free methods, PAR prunes visual tokens across all 32 LLM layers using a meta-router and simultaneously skips redundant layers in the last 16 layers based on learnable layer controller embeddings' importance scores, optimized via self-supervised DPO by minimizing KL divergence between pruned and original outputs. (Suo et al., 2025).

3.3 Hybrid Vision–LLM Pruning

Hybrid vision–LLM pruning methods determine visual token importance by *jointly leveraging visual structure and language semantics*, explicitly coupling visual token reduction with linguistic relevance signals. The pruning strategy may happen on both vision-side and LLM-side with different guidance. By integrating cues from both modalities, these methods aim to improve task alignment

and robustness under aggressive pruning budgets, at the cost of increased complexity or additional supervision. These methods adopt a multi-stage pruning strategy across both vision-side and LLM-side, leveraging information from both modalities.

MustDrop is a typical hybrid pruning method with three pruning stages with different goals: it removes spatially redundant tokens and retain key tokens across vision encoder, utilizes text-to-image attention score to guide pruning of text-irrelevant tokens during prefilling stage, and removes output-irrelevant tokens during decoding stage (Liu et al., 2024c). STAR performs two-stage pruning—early conservative visual self-attention pruning and later aggressive cross-modal attention pruning—balancing feature preservation with task relevance (Guo et al., 2025). ToDRE also adopts a two-stage strategy that first retains a diverse subset of visual tokens via greedy k-center selection before LLM decoder and then removes all remaining visual tokens once cross-modal attention becomes negligible in deeper layers inside LLM layers(Li et al., 2025b). IVTP is another two-stage pruning methods which stabilizes token importance using group-wise attention rollout on vision-side, and then pruning the text-irrelevant visual tokens again inside the LLM (Huang et al., 2024). CATP targets multimodal in-context learning by pruning image tokens based on cross-example and query-conditioned relevance. It adopts a two-stage pruning strategy: the first stage happen after projector and before decoder, maximizing text alignment and diversity; and the second stage progressively prunes tokens based on variation of token attention and query relevance (Li et al., 2025e).

Some methods jointly perform token pruning and token merging. We summarize them as a complementary category of hybrid approaches, since they involve pruning and operate across both the vision- and LLM-side. AIM combines token merging with token pruning, employing cosine similarity between embeddings as merging criteria before the LLM, then uses PageRank scores computed from self-attention weights as pruning criteria for progressive layer-wise token reduction within the LLM (Zhong et al., 2025). VisionDrop addresses cross-modal misalignment in LLMs by performing training-free, visual-only token pruning across multiple stages (in both visual encoder and LLM decoder), combining dominant token selection with contextual merging to preserve fine-grained visual information (Xu et al., 2025). VScan demonstrates that the effectiveness of visual token pruning critically depends on when pruning is performed along the LVLM pipeline, rather than solely on how tokens are scored. By revealing distinct roles of shallow vision layers and middle LLM layers, VScan reframes token reduction as a stage-aware optimization problem (Zhang et al., 2025a).

4 Comparison with Related Paradigms

Visual token pruning in LVLMs is closely related to and sometimes cooperates with several efficiency-oriented paradigms that also aim to reduce computation or memory cost. However, these paradigms differ fundamentally from token pruning in terms of *what* is reduced, *where* the reduction is applied, and *when* the reduction takes effect. We summarize the most relevant paradigms below and clarify their conceptual differences from visual token pruning.

Token Merging and Token Compression. Token merging and compression methods reduce the effective token count by *aggregating or replacing* tokens rather than explicitly discarding them. A representative example is ToMe, which introduces a training-free token merging mechanism that progressively fuses similar tokens via fast bipartite matching and proportional attention (Bolya et al., 2023). iLLaVA extends token merging to large vision–language models by performing attention-guided token merging in both the visual encoder and the language model (Hu et al., 2024). Related approaches such as VisionZip (Yang et al., 2025b), FiCoCo (Han et al., 2025), LaCo (Liu et al., 2025), LightVLM (Hu et al., 2025), and Fwd2Bot (Bulat et al., 2025) further compress dense visual tokens into compact semantic representations through clustering, pooling, or learned summarization. Unlike token pruning, these methods preserve information through aggregation rather than removal, typically offering higher stability but more limited aggressiveness under extreme compression budgets.

KV-cache Management and Adaptive Attention. KV-cache based methods reduce *decoding-time* memory footprint and attention computation by selectively retaining or computing key–value states, without modifying the input token sequence. Twilight generalizes sparse attention by replacing fixed-budget top-k selection with adaptive top-p retention (Lin et al., 2025). A-VL proposes a plug-and-play adaptive attention mechanism that separately manages visual and textual KV caches, dynamically retaining only critical visual states and a small local

text window (Zhang et al., 2025b). These methods primarily accelerate the decoding stage and are orthogonal to token pruning, which focuses on reducing prefilling cost via token selection.

Structural Pruning. Structural pruning reduces computation by removing or compressing *model structures* such as weights, modules, attention heads, or transformer layers. Representative examples include EfficientLLaVA (Liang et al., 2025a) and UKMP (Wu et al., 2025), which perform parameter-level pruning with learned importance metrics, and Short-LVLM (Ma et al., 2025a), which removes redundant transformer layers in a training-free manner. While structural pruning provides consistent speedups across both prefilling and decoding stages, it lacks the input adaptivity and instance-level flexibility offered by token pruning.

Overall, these paradigms are complementary rather than competing. Token pruning focuses on dynamic, input-adaptive token selection before or during language interaction, while token merging, KV-cache management, patch merging, structural pruning, and semantic compensation address efficiency from orthogonal dimensions. In practice, these techniques can be combined to further improve the efficiency of large vision–language models.

5 Benchmarks and Experimental Protocols

Benchmarks. We introduce several benchmarks to which most of the selected papers adapt. The detailed information of each benchmarks are presented in Table 1. Notably, these tasks differ in token redundancy, reasoning depth, and dependency on fine-grained features. For instance, text-oriented VQA and detailed visual reasoning typically demand high token fidelity, whereas global captioning may tolerate higher sparsity. Consequently, this diverse benchmark suite is essential for evaluating the generalization of pruning methods across varying sensitivities to information loss, ensuring that efficiency gains do not come at the cost of failing specific task distributions.

Experimental Protocols. To ensure fair comparison and reproducibility, rigorous protocols are required beyond simple metric reporting. Standard evaluations in the papers typically adhere to three key aspects:

- **Backbone Consistency**: Comparisons are strictly conducted on identical LVLM architectures (e.g., LLaVA (Liu et al., 2023b), BLIP-2 (Li et al., 2023a)) to isolate the efficacy of the pruning algorithm from the underlying model capability.
- **Pruning Paradigms**: Distinctions are explicitly drawn between training-free (zero-shot) methods and fine-tuning-based approaches, as they operate under fundamentally different computational budgets.
- **Inference Constraints**: Critical hyperparameters, including input resolution, batch size, and maximum generation length, are fixed to standardize the evaluation. This control is vital when comparing static pruning ratios against dynamic token budgets.

To assess the efficiency of token pruning methods, evaluations typically measure the trade-off between model sparsity and downstream task performance. Researchers primarily examine the capabilities of LVLMs across varying pruning ratios (i.e., the number or proportion of preserved tokens). The fundamental performance metrics include absolute Accuracy and the Performance Retention Rate, which quantifies the percentage of performance maintained relative to the original, unpruned baseline.

In terms of computational efficiency, quantitative comparisons rely on a multi-dimensional suite of metrics covering theoretical complexity, temporal latency, and spatial memory footprint:

- **FLOPs** (Floating Point Operations) serve as a standard proxy for evaluating the theoretical computational complexity of the model.
- **CUDA Latency** measures the actual wall-clock time required for kernel launches and data transfers. This metric is frequently decomposed into two phases to isolate stage-specific benefits: (1) Prefill Time, the duration required to process all input tokens and compute the embedding for the first generated token; and (2) Decode Time, the latency incurred during the subsequent autoregressive generation process.
- **KV Cache and GPU Memory** are utilized to evaluate the spatial efficiency gains. These metrics assess the reduction in video random access memory (VRAM) usage and Key-Value cache overhead, highlighting the method's ability to alleviate hardware bottlenecks during inference.

Although many token pruning methods report substantial FLOPs reductions, the realized wall-

Task Category	Description	Benchmarks
VQA	Answer questions based on images	VQAv2(Goyal et al., 2016), GQA(Hudson and Manning, 2019), VizWiz(Gurari et al., 2018), ScienceQA-IMG(Lu et al., 2022), HallBench(Guan et al., 2024), POPE(Li et al., 2023b), MME(Fu et al., 2025a), MMBench(Liu et al., 2024d), MMBench-CN(Liu et al., 2024d), MM-Vet(Yu et al., 2023)
Text-oriented VQA	VQA requiring text recognition in images	TextVQA(Singh et al., 2019), ChartQA(Masry et al., 2022), AI2D(Kembhavi et al., 2016), OCRBench(Liu et al., 2023c)
Video Understanding	Answer questions based on videos	MLVU(Zhou et al., 2025), MVBench(Li et al., 2024), LongVideoBench(Wu et al., 2024), Video-MME(Fu et al., 2025b)
Visual Captioning	Generate descriptive captions for images	COCO Caption(Lin et al., 2015), Flickr30k(Young et al., 2014)
Document & OCR	Extract and understand textual information	DocVQA(Mathew et al., 2020), IIIT5K(Mishra et al., 2012), ICDAR(Pfitzmann et al., 2022)

Table 1: Evaluation Benchmarks for Different Vision-Language Tasks

clock speedup in practical deployments is often significantly smaller. This discrepancy arises because most pruning strategies introduce irregular or input-dependent sparsity patterns that current GPU kernels cannot efficiently exploit (Dehghani et al., 2022; Hooker, 2020). Standard CUDA kernels are optimized for dense, regularly shaped tensors; dynamic token removal produces ragged sequences that require costly gather–scatter operations and break Tensor Core alignment (e.g., dense 16×16 tiles). Recent optimized kernels such as FlashInfer (Ye et al., 2025c) and FSA (Yan et al., 2025) attempt to mitigate this by repacking sparse tokens into dense SRAM-based tile layouts, thereby maintaining Tensor Core compatibility. In addition, block sparsity (e.g., 64×64) has been increasingly adopted to better align with the 128-byte cache-line granularity of modern GPUs. Emerging hardware features such as the Tensor Memory Accelerator (TMA) in NVIDIA Hopper further help mask irregular addressing overhead during both prefill and decoding. From a systems perspective, methods that apply token reduction *prior to* attention computation, maintain structured token layouts, or enable static-shape execution are more likely to achieve consistent real-world acceleration. We therefore recommend that future work report both FLOPs and wall-clock latency to provide a complete picture of practical efficiency.

These metrics provide a comprehensive view of the efficiency gains achieved by token pruning from both temporal (operation time) and spatial (memory footprint) perspectives.

6 Future Directions and Conclusion

Previous token pruning methods has achieved good balancing efficiency and performance. Here we outline some practical limitations and open challenges in token pruning methods.

Vision-side pruning reduces visual redundancy using intra-visual signals such as similarity and attention offering efficient acceleration since the token number is reduced before the LLM decoding. However, its task-agnostic nature and lack of language awareness might fundamentally limit its effectiveness in more complex multimodal reasoning tasks. LLM-side pruning introduces query-relevance token selection by leveraging cross-modal signals during decoding. While effective for suppressing visually redundant but task-irrelevant information, its reliance on noisy attention cues and late-stage pruning fundamentally limits efficiency gains and exposes it to cross-modal misalignment risks (Zhang et al., 2025c). Hybrid pruning methods combines both vision- and LLM-side methods by multi-stage or multi-factor pruning strategy, offering a principled compromise between efficiency and task relevance.

Task-agnostic Pruning. Recent studies suggest that aggressive visual token pruning may introduce failure modes beyond accuracy degradation, including spatial misalignment and hallucination amplification. GAP demonstrates that preserving positional consistency during pruning is crucial for maintaining grounding performance, indicating that future pruning methods should explicitly account for geometric and spatial constraints (Chien et al., 2025). Similarly, VASparse reveals that naive sparsification strategies can exacerbate visual hallucinations, motivating hallucination-aware pruning objectives that go beyond attention-based importance estimation (Zhuang et al., 2025). These findings highlight the need for behavior-aware pruning frameworks that optimize not only efficiency but

also grounding fidelity and hallucination robustness. Also, while LVLMs are widely used in a wide range of tasks, today's LVLMs pruning methods mainly limit on VQA tasks. How to extend the pruning strategy and cooperate with a wider range of practical phenomena is a trending research area. Researchers may explore the different ways for token importance evaluation and criteria for token retaining for different practical tasks.

Hallucination or Pruning. Along with the hallucination perspective, it's also crucial to create pruning task specific benchmarks. For now the evaluation for pruning topic still builds on the existed tasks, researchers performs original and pruned models on the same benchmarks and compare the accuracy drop as well as efficiency gains. An ideal pruning strategy should maintain the informational and sufficient tokens to answer the questions. However, there still lacks of discussion on the hidden mechanisms of whether the model can answer the questions with remaining tokens or the insignificant accuracy drop comes from model hallucination, especially when the pruning rates are high. Li et.al (Li et al., 2025d) investigated the prompt-visual coupling effect across different benchmarks and found that budget allocation should differ by the coupling rate of tasks. This also highlighted the necessity of investigating whether the pruning strategy is aligned with the tasks. In the future, introducing pruning specific tasks and benchmarks is important.

Practical Decision Guidance. Beyond cataloging existing methods, we distill a three-step decision framework to help practitioners select an appropriate pruning strategy for their deployment scenario.

Step 1: Identify the primary bottleneck. If prefill latency dominates, vision-side or hybrid pruning is preferred, since reducing tokens before the LLM yields the largest efficiency gains. If KV-cache memory or decode throughput is the bottleneck, LLM-side pruning is more effective. When both prefill and decode costs matter, hybrid pruning is recommended—applying vision-side reduction first and LLM-side eviction during decoding.

Step 2: Match the task requirements. General VQA and captioning tasks tolerate high pruning ratios (70–80%), as global semantic content is preserved even under aggressive token removal. OCR, document QA, and grounding tasks require preservation of spatial diversity and fine-grained local details; conservative ratios ($\leq$40%) are recommended, and merging-based approaches (e.g., ALTP, HoloV) are preferred over hard pruning. Video understanding tasks require temporal-aware strategies (e.g., MMG-Vid, FrameFusion) in addition to spatial pruning.

Step 3: Apply deployment constraints. If no training is allowed, methods marked Train?=✓ in Table 2 should be excluded (e.g., PAR, ATP-LLaVA, Glimpse). If no additional parameters are permitted, methods introducing new learnable modules should be avoided. Methods relying on `[CLS]` token (e.g., LLaVA-PruMerge, HiRED, FoPru) assume ViT-based encoders and cannot be directly applied to models such as Qwen2.5-VL that use different visual encoding schemes.

Temporal Redundancy in Video. Video inputs introduce temporal redundancy on top of spatial redundancy, yet the treatment of video-specific pruning remains underdeveloped relative to its importance. Current strategies can be broadly divided into two categories: (1) frame-level approaches that measure inter-frame similarity and uniformly prune redundant frames, and (2) temporal-aware approaches that exploit motion cues or segment-level budget allocation. MMG-Vid exemplifies the latter by segmenting videos by frame similarity, dynamically allocating budgets to maximize marginal gain, and prioritizing tokens novel to the selection history (Ma et al., 2025b). FrameFusion performs similarity-based token merging at shallow layers for temporal deduplication and importance-based pruning at deeper layers (Fu et al., 2025d). Despite these advances, several challenges remain open: motion-aware token selection guided by optical flow, principled handling of long-range temporal dependencies in hour-long videos, etc.

Agentic Reasoning and Long-form Generation. While most existing pruning methods are evaluated on VQA tasks, emerging applications such as agentic reasoning and long-form generation pose distinct challenges. Agentic tasks involve multi-step reasoning where early token removal may cause cascading errors, as downstream reasoning steps depend on visual details that were pruned in earlier stages. Long-form generation shifts the computational bottleneck from prefill to decode, requiring different pruning priorities: preserving tokens that sustain coherence over extended generation horizons rather than those most relevant to a single-turn answer. Exploring pruning strategies tailored to these settings represents a promising direction for future work.

7 Limitations

In this survey we reviewed visual token pruning method for LVLMs since 2024. Despite our effort to provide a comprehensive overview of token pruning in LVLMs, this survey has some limitations. First of all, as efficiency LVLMs nowadays are having great progress and there are more methods other than token pruning as well as methods combining token pruning with other compression technics. Secondly, most methods reviewed in this survey are developed for ViT-based visual encoders and late-fusion architectures. Emerging unified or early-fusion LVLMs may exhibit different redundancy patterns, which could limit the direct applicability of existing pruning principles and require rethinking pruning criteria and evaluation methodologies.

Acknowledgements

This work is supported in part by the U.S. Office of Naval Research Award under Grant Number N00014-24-1-2668, the National Science Foundation under Grants IIS-2316306 and CNS-2330215, the National Institutes of Health (NIH) under Grant R01EB293388, and Merck BARDS Academic Collaboration Grant.

References

Mohammad Mahdi Abootorabi, Amirhosein Zobeiri, Mahdi Dehghani, Mohammadali Mohammadkhani, Bardia Mohammadi, Omid Ghahroodi, Mahdieh Soleymani Baghshah, and Ehsaneddin Asgari. 2025. Ask in any modality: A comprehensive survey on multimodal retrieval-augmented generation. *Preprint*, arXiv:2502.08826.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. 2025. Divprune: Diversity-based visual token pruning for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9392–9401.

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. 2025. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 1773–1781.

Bizhe Bai, Jianjian Cao, Yadan Luo, and Tao Chen. 2025. Local information matters: Inference acceleration for grounded conversation generation models through adaptive local-aware token pruning. *Preprint*, arXiv:2503.23959.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2023. Token merging: Your vit but faster. *Preprint*, arXiv:2210.09461.

Adrian Bulat, Yassine Ouali, and Georgios Tzimiropoulos. 2025. Fwd2bot: Lvlm visual token compression with double forward bottleneck. *Preprint*, arXiv:2503.21757.

Hanning Chen, Yang Ni, Wenjun Huang, Hyunwoo Oh, Yezi Liu, Tamoghno Das, and Mohsen Imani. 2025a. Lvlm_csp: Accelerating large vision language models via clustering, scattering, and pruning for reasoning segmentation. *Preprint*, arXiv:2504.10854.

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

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

Jian Cheng, Haidong Kang, Yuxin Shao, Nan Li, Pengjun Chen, Rui Wang, Saiqin Long, Xiaochun Yang, and Lianbo Ma. 2025. Survey on efficient large language models: Principles, algorithms, applications, and open issues. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21.

Tzu-Chun Chien, Chieh-Kai Lin, Shiang-Feng Tsai, Ruei-Chi Lai, Hung-Jen Chen, and Min Sun. 2025. Grounding-aware token pruning: Recovering from drastic performance drops in visual grounding caused by pruning. *Preprint*, arXiv:2506.21873.

Mostafa Dehghani, Anurag Arnab, Lucas Beyer, Ashish Vaswani, and Yi Tay. 2022. The efficiency misnomer. *Preprint*, arXiv:2110.12894.

Mark Endo, Xiaohan Wang, and Serena Yeung-Levy. 2025. Feather the throttle: Revisiting visual token pruning for vision-language model acceleration. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22826–22835.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and 1 others. 2025a. Mme: A comprehensive evaluation benchmark for multimodal large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Chaoyou Fu, Yuhan Dai, Yongdong Luo, Lei Li, Shuhuai Ren, Renrui Zhang, Zihan Wang, Chenyu Zhou, Yunhang Shen, Mengdan Zhang, and 1 others. 2025b. Video-mme: The first-ever comprehensive evaluation benchmark of multi-modal llms in video analysis. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24108–24118.

Mingyu Fu, Wei Suo, Ji Ma, Lin Yuanbo Wu, Peng Wang, and Yanning Zhang. 2025c. Mitigating information loss under high pruning rates for efficient large vision language models. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 4156–4165.

Tianyu Fu, Tengxuan Liu, Qinghao Han, Guohao Dai, Shengen Yan, Huazhong Yang, Xuefei Ning, and Yu Wang. 2025d. Framefusion: Combining similarity and importance for video token reduction on large vision language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22654–22663.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. 2016. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. *CoRR*, abs/1612.00837.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. 2024. Hallusionbench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *CVPR*.

Yichen Guo, Hanze Li, Zonghao Zhang, Jinhao You, Kai Tang, and Xiande Huang. 2025. Star: Stage-wise attention-guided token reduction for efficient large vision-language models inference. *Preprint*, arXiv:2505.12359.

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

Yuhang Han, Xuyang Liu, Zihan Zhang, Pengxiang Ding, Junjie Chen, Donglin Wang, Honggang Chen, Qingsen Yan, and Siteng Huang. 2025. Filter, correlate, compress: Training-free token reduction for mllm acceleration. *Preprint*, arXiv:2411.17686.

Sara Hooker. 2020. The hardware lottery. *Preprint*, arXiv:2009.06489.

Lianyu Hu, Fanhua Shang, Wei Feng, and Liang Wan. 2025. Lightvlm: Acceleraing large multimodal models with pyramid token merging and kv cache compression. *Preprint*, arXiv:2509.00419.

Lianyu Hu, Fanhua Shang, Liang Wan, and Wei Feng. 2024. illava: An image is worth fewer than 1/3 input tokens in large multimodal models. *Preprint*, arXiv:2412.06263.

Kai Huang, Hao Zou, Ye Xi, BoChen Wang, Zhen Xie, and Liang Yu. 2024. Ivtp: Instruction-guided visual token pruning for large vision-language models. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XVII*, page 214–230, Berlin, Heidelberg. Springer-Verlag.

Drew A. Hudson and Christopher D. Manning. 2019. GQA: a new dataset for compositional question answering over real-world images. *CoRR*, abs/1902.09506.

Lei Jiang, Weizhe Huang, Tongxuan Liu, Yuting Zeng, Jing Li, Lechao Cheng, and Xiaohua Xu. 2024. Fopru: Focal pruning for efficient large vision-language models. *CoRR*.

Lei Jiang, Zixun Zhang, Yuting Zeng, Chunzhao Xie, Tongxuan Liu, Zhen Li, Lechao Cheng, and Xiaohua Xu. 2025. DCP: Dual-cue pruning for efficient large vision-language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 21202–21215, Suzhou, China. Association for Computational Linguistics.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. 2016. A diagram is worth a dozen images. In *Computer Vision – ECCV 2016*, pages 235–251, Cham. Springer International Publishing.

Yulong Lei, Zishuo Wang, Jinglin Xu, and Yuxin Peng. 2025. Ctp2fic: From coarse-grained token pruning to fine-grained token clustering for lvlm inference acceleration (chinamm 2025). *Available at SSRN 5545751*.

Ao Li, Yuxiang Duan, Jinghui Zhang, Congbo Ma, Yutong Xie, Gustavo Carneiro, Mohammad Yaqub, and Hu Wang. 2025a. Transprune: Token transition pruning for efficient large vision-language model. *Preprint*, arXiv:2507.20630.

Duo Li, Zuhao Yang, Xiaoqin Zhang, Ling Shao, and Shijian Lu. 2025b. Todre: Effective visual token pruning via token diversity and task relevance. *Preprint*, arXiv:2505.18757.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *Preprint*, arXiv:2301.12597.

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

Kaiyuan Li, Xiaoyue Chen, Chen Gao, Yong Li, and Xinlei Chen. 2025c. Balanced token pruning: Accelerating vision language models beyond local optimization. *Preprint*, arXiv:2505.22038.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, and 1 others. 2024. Mvbench: A comprehensive multi-modal video understanding benchmark. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22195–22206.

Yangfu Li, Hongjian Zhan, Tianyi Chen, Qi Liu, and Yue Lu. 2025d. Why 1 + 1 < 1 in visual token pruning: Beyond naive integration via multi-objective balanced covering. *Preprint*, arXiv:2505.10118.

Yanshu Li, Jianjiang Yang, Zhennan Shen, Ligong Han, Haoyan Xu, and Ruixiang Tang. 2025e. Catp: Contextually adaptive token pruning for efficient and enhanced multimodal in-context learning. *Preprint*, arXiv:2508.07871.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. 2023b. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305, Singapore. Association for Computational Linguistics.

Yinan Liang, Ziwei Wang, Xiuwei Xu, Jie Zhou, and Jiwen Lu. 2025a. Efficientllava: Generalizable auto-pruning for large vision-language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 9445–9454.

Yuxuan Liang, Xu Li, Xiaolei Chen, Yi Zheng, Haotian Chen, Bin Li, and Xiangyang Xue. 2025b. Pyramid token pruning for high-resolution large vision-language models via region, token, and instruction-guided importance. *Preprint*, arXiv:2509.15704.

Chaofan Lin, Jiaming Tang, Shuo Yang, Hanshuo Wang, Tian Tang, Boyu Tian, Ion Stoica, Song Han, and Mingyu Gao. 2025. Twilight: Adaptive attention sparsity with hierarchical top-p pruning. *Preprint*, arXiv:2502.02770.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. 2015. Microsoft coco: Common objects in context. *Preprint*, arXiv:1405.0312.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2024a. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 26296–26306.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023a. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Juntao Liu, Liqiang Niu, Wenchao Chen, Jie Zhou, and Fandong Meng. 2025. Laco: Efficient layer-wise compression of visual tokens for multimodal large language models. *Preprint*, arXiv:2507.02279.

Shilong Liu, Hao Cheng, Haotian Liu, Hao Zhang, Feng Li, Tianhe Ren, Xueyan Zou, Jianwei Yang, Hang Su, Jun Zhu, and 1 others. 2024b. Llava-plus: Learning to use tools for creating multimodal agents. In *European conference on computer vision*, pages 126–142. Springer.

Ting Liu, Liangtao Shi, Richang Hong, Yue Hu, Quanjun Yin, and Linfeng Zhang. 2024c. Multi-stage vision token dropping: Towards efficient multimodal large language model. *Preprint*, arXiv:2411.10803.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. 2024d. Mmbench: Is your multi-modal model an all-around player? In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VI*, page 216–233, Berlin, Heidelberg. Springer-Verlag.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xucheng Yin, Cheng lin Liu, Lianwen Jin, and Xiang Bai. 2023c. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in Neural Information Processing Systems*, 35:2507–2521.

Bozhi Luan, Wengang Zhou, Hao Feng, Zhe Wang, Xiaosong Li, and Houqiang Li. 2025. Multi-cue adaptive visual token pruning for large vision-language models. *Preprint*, arXiv:2503.08019.

Deng Luo, Dongyang Zhang, Qiuhao Xie, Cencen Liu, Qiang Dong, and Xiurui Xie. 2025. Rethinking attention cues: Multi-factor guided token pruning for efficient vision-language understanding. *Available at SSRN 5615684*.

Ji Ma, Wei Suo, Peng Wang, and Yanning Zhang. 2025a. Short-lvlm: Compressing and accelerating large vision-language models by pruning redundant layers. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 3575–3584.

Junpeng Ma, Qizhe Zhang, Ming Lu, Zhibin Wang, Qiang Zhou, Jun Song, and Shanghang Zhang. 2025b. Mmg-vid: Maximizing marginal gains at segment-level and token-level for efficient video llms. *Preprint*, arXiv:2508.21044.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279.

Minesh Mathew, Dimosthenis Karatzas, R Manmatha, and CV Jawahar. 2020. Docvqa: A dataset for vqa on document images. corr abs/2007.00398 (2020). *arXiv preprint arXiv:2007.00398*.

Yu Meng, Kaiyuan Li, Chenran Huang, Chen Gao, Xinlei Chen, Yong Li, and Xiaoping Zhang. 2025. Plphp: Per-layer per-head vision token pruning for efficient large vision-language models. *Preprint*, arXiv:2502.14504.

A. Mishra, K. Alahari, and C. V. Jawahar. 2012. Scene text recognition using higher order language priors. In *BMVC*.

Ruiguang Pei, Weiqing Sun, Zhihui Fu, and Jun Wang. 2025. Greedyprune: Retenting critical visual token set for large vision language models. *Preprint*, arXiv:2506.13166.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S Nassar, and Peter W J Staar. 2022. Doclaynet: A large human-annotated dataset for document-layout segmentation. page 3743–3751.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. *Preprint*, arXiv:2103.00020.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2025. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22857–22867.

Kele Shao, Keda Tao, Kejia Zhang, Sicheng Feng, Mu Cai, Yuzhang Shang, Haoxuan You, Can Qin, Yang Sui, and Huan Wang. 2025. When tokens talk too much: A survey of multimodal long-context token compression across images, videos, and audios. *arXiv preprint arXiv:2507.20198*.

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

Yi-Lin Sung, Jaehong Yoon, and Mohit Bansal. 2024. Ecoflap: Efficient coarse-to-fine layer-wise pruning for vision-language models. *Preprint*, arXiv:2310.02998.

Wei Suo, Ji Ma, Mengyang Sun, Lin Yuanbo Wu, Peng Wang, and Yanning Zhang. 2025. Pruning all-rounder: Rethinking and improving inference efficiency for large vision language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20247–20256.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Tiannan Wang, Wangchunshu Zhou, Yan Zeng, and Xinsong Zhang. 2023. Efficientvlm: Fast and accurate vision-language models via knowledge distillation and modal-adaptive pruning. In *Findings of the association for computational linguistics: ACL 2023*, pages 13899–13913.

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

Haoning Wu, Dongxu Li, Bei Chen, and Junnan Li. 2024. Longvideobench: A benchmark for long-context interleaved video-language understanding. *Advances in Neural Information Processing Systems*, 37:28828–28857.

Zimeng Wu, Jiaxin Chen, and Yunhong Wang. 2025. Unified knowledge maintenance pruning and progressive recovery with weight recalling for large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 8550–8558.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. 2025. Pyramidrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *Preprint*, arXiv:2410.17247.

Rui Xu, Yunke Wang, Yong Luo, and Bo Du. 2025. Rethinking visual token reduction in lvlms under cross-modal misalignment. *Preprint*, arXiv:2506.22283.

Ran Yan, Youhe Jiang, Zhuoming Chen, Haohui Mai, Beidi Chen, and Binhang Yuan. 2025. Fsa: An alternative efficient implementation of native sparse attention kernel. *Preprint*, arXiv:2508.18224.

Longrong Yang, Dong Shen, Chaoxiang Cai, Kaibing Chen, Fan Yang, Tingting Gao, Di Zhang, and Xi Li. 2025a. Libra-merging: Importance-redundancy and pruning-merging trade-off for acceleration plug-in in large vision-language model. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9402–9412.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2025b. Visionzip: Longer is better but not necessary in vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19792–19802.

Huanjin Yao, Ruifei Zhang, Jiaxing Huang, Jingyi Zhang, Yibo Wang, Bo Fang, Ruolin Zhu, Yongcheng Jing, Shunyu Liu, Guanbin Li, and Dacheng Tao. 2025. A survey on agentic multimodal large language models. *Preprint*, arXiv:2510.10991.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. 2025a. Fit and prune: fast and training-free visual token pruning for multi-modal large language models. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'25/IAAI'25/EAAI'25. AAAI Press.

Xubing Ye, Yukang Gan, Yixiao Ge, Xiao-Ping Zhang, and Yansong Tang. 2025b. Atp-llava: Adaptive token pruning for large vision language models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24972–24982.

Zihao Ye, Lequn Chen, Ruihang Lai, Wuwei Lin, Yineng Zhang, Stephanie Wang, Tianqi Chen, Baris Kasikci, Vinod Grover, Arvind Krishnamurthy, and Luis Ceze. 2025c. Flashinfer: Efficient and customizable attention engine for llm inference serving. *Preprint*, arXiv:2501.01005.

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

Weihao Yu, Zhengyuan Yang, Linjie Li, Jianfeng Wang, Kevin Lin, Zicheng Liu, Xinchao Wang, and Lijuan Wang. 2023. Mm-vet: Evaluating large multimodal models for integrated capabilities. *arXiv preprint arXiv:2308.02490*.

Quan-Sheng Zeng, Yunheng Li, Qilong Wang, Peng-Tao Jiang, Zuxuan Wu, Ming-Ming Cheng, and Qibin Hou. 2025. A glimpse to compress: Dynamic visual token pruning for large vision-language models. *Preprint*, arXiv:2508.01548.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. *Preprint*, arXiv:2303.15343.

Ce Zhang, Kaixin Ma, Tianqing Fang, Wenhao Yu, Hongming Zhang, Zhisong Zhang, Yaqi Xie, Katia Sycara, Haitao Mi, and Dong Yu. 2025a. Vscan: Rethinking visual token reduction for efficient large vision-language models. *Preprint*, arXiv:2505.22654.

Junyang Zhang, Mu Yuan, Ruiguang Zhong, Puhan Luo, Huiyou Zhan, Ningkang Zhang, Chengchen Hu, and Xiang-Yang Li. 2025b. A-vl: Adaptive attention for large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22461–22469.

Qizhe Zhang, Aosong Cheng, Ming Lu, Renrui Zhang, Zhiyong Zhuo, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. 2025c. Beyond text-visual attention: Exploiting visual cues for effective token pruning in vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20857–20867.

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. 2025d. Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. *Preprint*, arXiv:2506.10967.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. 2025e. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *Preprint*, arXiv:2410.04417.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, and 1 others. 2023. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.

Yiwu Zhong, Zhuoming Liu, Yin Li, and Liwei Wang. 2025. Aim: Adaptive inference of multi-modal llms via token merging and pruning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 20180–20192.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, and 1 others. 2025. Mlvu: Benchmarking multi-task long video understanding. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 13691–13701.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577.

Xianwei Zhuang, Zhihong Zhu, Yuxin Xie, Liming Liang, and Yuexian Zou. 2025. Vasparse: Towards efficient visual hallucination mitigation via visual-aware token sparsification. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 4189–4199.

Xin Zou, Di Lu, Yizhou Wang, Yibo Yan, Yuanhuiyi Lyu, Xu Zheng, Linfeng Zhang, and Xuming Hu. 2025. Don't just chase "highlighted tokens" in mllms: Revisiting visual holistic context retention. *Preprint*, arXiv:2510.02912.

A Appendix.

A.1 Tables.

To organize the rapidly growing body of work on LVLM token pruning, we provide a systematic literature review structured as follows in this appendix. Table 2 provides a detailed comparison of vision token pruning methods, including their key innovations, training requirements, and pruning granularity. Table 3 complements this by surveying related compression techniques—such as token merging, knowledge distillation, and quantization—that address similar efficiency goals but through different mechanisms.

Table 2: Survey of Vision Token Pruning Methods for LVLMs. **Arch. Dep.**: architecture dependency—**H**igh (requires specific encoder features such as `[CLS]` token), **M**edium (assumes specific cross-modal attention layout), **L**ow (largely backbone-agnostic). **Task Suit.**: validated task categories—**G**eneral VQA, **T**ext-oriented VQA / OCR, **V**ideo Understanding, **R**eferring / Grounding.

Side	Paper	Category	Train?	Arch. Dep.	Granularity	Task Suit.	Key Innovation
Vision-side	LLaVA-PruMerge (Shang et al., 2025)	Similarity-based	×	H	Before projector	G	`[cls]` attention sparsity + key-similarity clustering
	DivPrune (Alvar et al., 2025)	Similarity-based	×	L	After projector	G	Diversity-driven: maximize minimum pairwise distance
	CDPruner (Zhang et al., 2025d)	Semantic-aware	×	L	After projector	G, T	Token similarity + textual relevance (DPP-based)
	MFPruner (Luo et al., 2025)	Semantic-aware	×	H	Before projector	G, T	Fuse `[CLS]` attn, similarity, instruction relevance via voting
	MMG-Vid (Ma et al., 2025b)	Semantic-aware	×	L	After projector	V	Two-level marginal gain maximization; adaptive temporal budget
	ALTP (Bai et al., 2025)	Semantic-aware	×	L	After projector	G, R	Region partitioning + density-adaptive token allocation
	FoPru (Jiang et al., 2024)	Attention-based	×	H	Before projector	G	Variance-aware `[CLS]`-attention with global/local selection
	PTP (Liang et al., 2025b)	Attention-based	×	H	After projector	G, T	Region-level + token-level `[CLS]` attention
	HoloV (Zou et al., 2025)	Attention-based	×	H	Before projector	G	Crop-wise variance + `[CLS]` attention importance
	VisPruner (Zhang et al., 2025d)	Attention-based	×	H	Before projector	G	`[CLS]` attn importance + redundancy removal via similarity
	HiRED (Arif et al., 2025)	Attention-based	×	H	Before projector	G	Early `[CLS]`-attn for budget; final `[CLS]`-attn for importance
LLM-side	AdaptPrune (Luan et al., 2025)	One-shot	×	M	During decoding	G	Cross-attn + spatial distance + diverse tokens
	DART (Wen et al., 2025)	One-shot	×	L	During decoding	G	Pivot tokens + duplicate dropping
	Libra-Merging (Yang et al., 2025a)	One-shot	×	M	During decoding	G	Pruning–merging trade-off
	GreedyPrune (Pei et al., 2025)	One-shot	×	M	Between layer 1–2	G	Greedy subset selection for critical token retention
	FastV (Chen et al., 2024)	Attention-based	×	M	After layer 2	G	Drop low-attention visual tokens
	PyramidDrop (Xing et al., 2025)	Attention-based	×	M	Stage-wise	G	Progressive pruning ratio + instruction-token attention
	FEATHER (Endo et al., 2025)	Attention-based	×	M	Early–mid + deep	G	RoPE-free attention scores
	SparseVLM (Zhang et al., 2025e)	Attention-based	×	M	All layers	G	Text–visual cross-attention sparsity
	PLPHP (Meng et al., 2025)	Attention-based	×	M	Per-layer, per-head	G	Layer/head-wise adaptive retention
	FitPrune (Ye et al., 2025a)	Attention-based	×	M	All layers	G	Minimize attention distribution divergence
	TransPrune (Li et al., 2025a)	Attention-based	×	L	Shallow–mid layers	G	Token Transition Variation for layer-wise selection
	ATP-LLaVA (Ye et al., 2025b)	Attention-based	✓	M	Between any two layers	G	Learnable adaptive token pruning
	V2Drop (Chen et al., 2025b)	Multi-factor	×	L	Predefined layers	G	Variation-based token importance
	Ctp2Fic (Lei et al., 2025)	Multi-factor	×	M	Layer 7 + Layer 22	G	Text-guided pruning + LSH-based clustering
	PAR (Suo et al., 2025)	Multi-factor	✓	M	Adaptive	G	Meta-router trained via DPO

Continued on next page

Table 2 – continued from previous page

Side	Paper	Category	Train	Arch. Dep.	Granularity	Task Suit.	Key Innovation
	FrameFusion (Fu et al., 2025d)	Multi-factor	×	M	Deep decoder layers	V	Shallow merging (similarity) + deep pruning (importance)
	LVLM_CSP (Chen et al., 2025a)	Multi-factor	×	M	Deep decoder layers	G	Clustering–Scattering–Pruning via [SEG] attention
	MoB (Li et al., 2025c)	Multi-factor	×	L	Early decoder layer	G	Balance prompt alignment + visual preservation
Hybrid	CATP (Li et al., 2025e)	Multi-stage	×	M	Pre-decoder + shallow decoder	G	Text–image alignment + diversity; inter-layer attn changes
	MustDrop (Liu et al., 2024c)	Multi-stage	×	M	Encode + prefill + decode	G	Lifecycle-aware token dropping
	VisionDrop (Xu et al., 2025)	Multi-stage + merging	×	M	Vision encoder + decoder	G	Visual-only scoring + stage-wise dominant selection
	IVTP (Huang et al., 2024)	Two-stage	×	M	Vision encoder + early decoder	G, T	Group-wise attention rollout + instruction-aware retaining
	AIM (Zhong et al., 2025)	Merging + pruning	×	M	Vision merging + LLM pruning	G	Cosine similarity merging + PageRank-based pruning
	Glimpse (Zeng et al., 2025)	Attention-based	✓	M	After layer K	G	Learnable glimpse token + predictor
	STAR (Guo et al., 2025)	Multi-stage	×	M	Vision encoder + LLM layer	G	Early self-attention + later cross-modal guidance
	ToDRE (Li et al., 2025b)	Multi-stage	×	L	After encoder + late decoder	G	Diverse token selection with relevance awareness
	VScan (Zhang et al., 2025a)	Merging + pruning	×	M	Vision merging + mid decoder pruning	G, T, R	Stage-aware optimization; global–local scanning

Year	Paper	Category	Train?	Granularity	Key Innovation
2022	EfficientVLM(Wang et al., 2023)	Model compression	✓	Layer-/module-level	Distill-then-prune with modal-adaptive pruning that learns task-specific importance of vision vs. language encoders
2023	ToMe(Bolya et al., 2023)	Structural compression	×	Encoder merging	Training-free token merging for ViTs
2024	ECOFLAP(Sung et al., 2024)	Weight Pruning	×	Layer-wise (weight)	Coarse-to-fine layer-wise pruning with global importance estimated via zeroth-order gradients
2024	iLLaVA(Hu et al., 2024)	Visual token merging	×	Encoder + LLM	Attention-guided one-step token merging with information recycling across both image encoder and LLM
2024	VisionZip(Yang et al., 2025b)	Structural compression	✓	Encoder dominant-token + merge	Attention concentration + similarity merging
2025	ACCM(Fu et al., 2025c)	Visual token pruning	✓	Encoder + Decoder Pruning	Train a caption model to mitigate information loss / retain key information
2025	A-VL(Zhang et al., 2025b)	KV-cache optimization	×	Decoder-side	Modality-aware adaptive attention: hierarchical vision KV selection + multi-scale text cache
2025	EfficientLLaVA(Liang et al., 2025a)	Weight Pruning	✓	Layer-wise	Structural risk minimization: search layer-wise pruning ratios using few proxy samples and evolve the search space by optimizing the vision–language projector
2025	FiCoCo-V / -L (Han et al., 2025)	Token compression	×	Filter–Correlate–Compress	Three-phase design; V/L variants (training-free)
2025	Fwd2Bot(Bulat et al., 2025)	Visual token compression	✓	Decoder-side	Condense visual tokens into task-agnostic summary tokens, jointly optimized with autoregressive + contrastive losses
2025	short-LVLM (Ma et al., 2025a)	Layer pruning	×	Decoder-side layer-level	Token-aware layer localization (Token Importance Scores) + Subspace-Compensated Pruning
2025	LightVLM (Hu et al., 2025)	Token merging + KV cache compression	×	Encoder-side merging + Decoder-side KV cache	Pyramid token merging across LLM layers to hierarchically condense visual tokens + attention-guided KV cache compression
2025	UKMP(Wu et al., 2025)	Weight pruning	✓	MHA and FFN across vision + language	Balance modality- and block-wise importance + distillation
2025	DCP(Jiang et al., 2025)	Structured pruning	×	Dependency-aware channel pruning	Dependency-consistent pruning for efficiency
2025	Twilight (Lin et al., 2025)	Dynamic top-p	×	Decoder attention sparsity	Top-p instead of top-k for adaptive retention

Table 3: Other Efficiency Methods mentioned in this survey

Table 4: Performance comparison at ≈192 retained tokens (↓66.7% pruning rate) on LLaVA-1.5-7B. Green = best method at this budget. "—" = not reported.

Method	GQA	MMBench	MME	POPE	SQA-IMG	VQA^{v2}	VQA^{Text}	Relative %
LLaVA-1.5-7B (Full, 576t)	61.9	64.7	1862	85.9	69.5	78.5	58.2	100%
Retain 192 Tokens (↓66.7%)								
ToMe	54.3	60.5	1563	72.4	65.2	68.0	52.1	88–89%
FastV	52.7	61.2	1612	64.8	67.3	67.1	52.5	88–91%
LLaVA-PruMerge	54.3	59.6	1632	71.3	67.9	70.6	54.3	90.3%
HiRED	58.7	62.8	1737	82.8	68.4	74.9	47.4	93.6%
SparseVLM	57.6	62.5	1721	83.6	69.1	75.6	56.1	95–96%
PDrop	57.1	63.2	1766–97	82.3	68.8	75.1	56.1	96–97%
MustDrop	58.2	62.3	1787	82.6	69.2	76.0	56.5	97.2%
V^2Drop	58.5	63.7	1826	85.1	69.3	—	55.6	97.6%
FiCoCo-L	61.1	64.6	—	84.6	69.6	76.8	55.7	98.0%
DART†	60.0	63.6	**1856**	82.8	69.8	76.7	57.4	98.8%
VisionZip	59.3	63.0	1783	85.3	68.9	76.8	57.3	98.5%
VisionZip‡	60.1	63.4	1834	84.9	68.2	77.4	57.8	99.1%
VisionDrop	59.99	65.19	1801	87.23	69.06	77.28	57.81	98.76%
VScan	60.6	63.9	1806	86.2	68.6	77.8	57.7	99.0%
ATP-LLaVA* (144t)	—	—	1473.9	84.2	69.1	76.4	—	≈98%
GreedyPrune	61.4	63.3	1488	85.5	—	—	—	97.81%
TransPrune-High (≈156t)	61.4	66.0	1540	85.0	69.5	77.9	57.8	**100.0%**
MoB	**61.4**	**64.1**	**1860**	**84.8**	**70.1**	**78.3**	**58.5**	**100.6%**

Table 5: Performance comparison at ≈128 retained tokens (↓77.8% pruning rate) on LLaVA-1.5-7B.

Method	GQA	MMBench	MME	POPE	SQA-IMG	VQA^{v2}	VQA^{Text}	Relative %
LLaVA-1.5-7B (Full, 576t)	61.9	64.7	1862	85.9	69.5	78.5	58.2	100%
Retain 128 Tokens (↓77.8%)								
ToMe	52.4	53.3	1343	62.8	59.6	63.0	49.1	80–82%
FastV	49.6	56.1	1490	59.6	60.2	61.8	50.6	83–87%
SparseVLM	56.0	60.0	1696	80.5	67.1	73.8	54.9	93–94%
HiRED	57.2	61.5	1710	79.8	68.1	73.4	46.1	91.6%
PDrop	56–57	61–62	1644–61	82.3	68–69	72.9	55.1	94–96%
MustDrop	56.9	61.1	1745	78.7	68.5	74.6	56.3	95.6%
DivPrune	59.4	61.5	1405.1	87.0	61.5	76.0	55.9	97.5%
V^2Drop	56.3	61.8	1712	80.9	68.8	—	53.8	94.0%
VisionZip‡	58.9	62.6	1823	83.7	68.3	76.6	57.0	98.4%
DART	58.7	63.2	**1840**	80.1	69.1	75.9	56.4	98.0%
FasterVLM	58.34	62.54	1433.8	83.46	67.92	76.19	57.07	98.75%
ATP-LLaVA* (144t)	59.5	**66.0**	1473.9	84.2	69.1	76.4	—	98.1%
VisionDrop	58.61	64.52	1777	85.92	68.52	76.24	57.63	97.80%
CDPruner	59.9	63.1	1431.4	**87.7**	63.1	76.6	56.2	99.0%
MFPruner	59.7	62.8	1419.9	86.5	68.7	76.7	55.8	97.9%
GreedyPrune	**61.2**	62.4	1483	85.6	—	—	—	96.75%
VScan	59.8	63.0	1792	86.1	68.9	77.1	57.3	98.8%
MoB	**60.9**	**63.5**	**1845**	**82.1**	**69.6**	**77.5**	**57.8**	**99.4%**

Table 6: Performance comparison at ≈64 retained tokens (↓88.9% pruning rate) on LLaVA-1.5-7B. High-compression regime that clearly differentiates method quality.

Method	GQA	MMBench	MME	POPE	SQA-IMG	VQA^{v2}	VQA^{Text}	Relative %
LLaVA-1.5-7B (Full, 576t)	61.9	64.7	1862	85.9	69.5	78.5	58.2	100%
Retain 64 Tokens (↓88.9%)								
IVTP	60.4	66.1	72.6	85.7	67.8	77.8	58.2	–
ToMe	48.6	43.7	1138	52.5	50.0	57.1	45.3	69–71%
FastV	46.1	48.0	1256	48.0	51.1	55.0	47.8	73–77%
LLaVA-PruMerge	48.8	47.4	1201	65.3	51.9	56.2	46.1	86.5%
SparseVLM	52.7	56.2	1505	75.1	62.2	68.2	51.8	84–86%
PDrop	41.9–47	33–58	1092–1561	55.9	—	—	50.6	70–87%
V^2Drop	50.5	55.2	1470	75.1	68.9	—	51.8	86.9%
DivPrune	57.5	60.1	1334.7	85.5	60.1	74.1	54.5	94.7%
MustDrop	53.1	60.0	1612	68.0	63.4	69.3	54.2	90.1%
FasterVLM (58t)	54.91	60.57	1348.6	75.85	68.91	71.92	55.28	94.24%
DART†	55.9	60.6	**1765**	73.9	69.8	72.4	54.4	93.7%
VisionZip‡	57.0	61.5	1756	80.9	68.8	74.2	56.0	95.2%
VisionDrop	55.89	62.95	1698	81.58	69.31	73.16	55.59	95.22%
MFPruner	58.2	61.6	1405.8	85.9	69.6	75.2	55.3	96.8%
GreedyPrune	60.4	61.3	1442	84.4	—	—	—	94.22%
VScan	58.3	62.1	1698	85.0	69.1	75.4	55.6	96.7%
MoB	59.0	62.1	1806	77.2	69.8	75.5	57.0	96.4%
CDPruner	**58.6**	**61.1**	**1415.1**	**87.5**	**61.1**	**75.4**	**55.3**	**97.0%**

(a) MMBench versions (EN/CN), POPE computation, and VQAv2 splits differ across papers; absolute cross-paper comparisons should be made with caution. (b) HiRED is specialised for high-resolution LLaVA-NeXT. (c) FoPru is the only method reporting measured TTFT/TPOT latency; consult it when wall-clock inference speed matters. (d) DART† and ATP-LLaVA require training-stage integration, typically yielding +3–5 relative pp. (e) FastV and ToMe are fully plug-in (no training needed) and are recommended only as rapid-prototyping baselines. (f) PDrop shows high result variance across papers, likely due to hyperparameter sensitivity; verify against original paper settings. (g) PDrop values vary substantially across reporting papers due to hyperparameter sensitivity.