RDPRUNE: LEARNING VISUAL TOKEN PRUNING VIA RATE-DISTORTION OPTIMIZATION FOR LARGE VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large vision-language models (LVLMs) incur substantial inference overhead due to long visual token sequences. Existing visual token pruning methods typically operate under predefined budgets, overlooking that inputs of different difficulty may require different amounts of visual information. We formulate visual token pruning as lossy compression, where the retained token ratio defines the rate and the behavioral discrepancy induced by pruning defines the distortion. Joint rate-distortion optimization over the data distribution enables adaptive allocation of a shared token budget across inputs according to their difficulty. We instantiate this formulation with RDPrune, a lightweight coarse-to-fine framework consisting of an early selector after vision encoder and a mid-layer selector inside the LLM. Both selectors learn binary keep-or-drop decisions to progressively compress the visual tokens. Crucially, we explicitly model pruning-induced distortion at three complementary levels: task distortion via cross-entropy, prediction distortion via output distribution KL divergence, and representation distortion via hidden states discrepancy. This directly quantifies how token removal affects downstream model behavior, rather than relying solely on pre-pruning token importance proxies. Across seven benchmarks and three LVLM backbones, RDPrune retains 97.7% of the original performance with only 11.1% of the visual tokens, achieving 2.8× inference speedup.

1 INTRODUCTION

Large vision-language models (LVLMs) have achieved strong performance across visual question answering, document understanding, visual perception, and multimodal reasoning (Liu et al., 2023; Bai et al., 2025). However, high-resolution images, multi-image inputs, and videos can produce hundreds or thousands of visual tokens, leading to substantial computation and memory overhead. Reducing redundant visual tokens is therefore important for efficient LVLM inference.

Most existing methods prune or merge visual tokens based on estimated importance or redundancy. Common criteria include attention and text-to-vision relevance (Chen et al., 2024; Zhang et al., 2025c; Sun et al., 2025; Yang et al., 2025), feature diversity (Alvar et al., 2025; Zhang et al., 2025b), token merging and adaptive pruning ratios (Shang et al., 2025; Ye et al., 2025a; Baek et al., 2026), as well as information flow and representation transition (Sun et al., 2026; Li et al., 2026). Despite these differences, most methods still focus on deciding which tokens to retain under a predefined or heuristically adjusted budget. The token cost and the resulting downstream degradation are not jointly optimized. Moreover, a shared fixed budget cannot account for differences in input difficulty. This limitation becomes particularly evident under aggressive pruning, where performance often drops sharply as the token budget decreases, as shown in Figure 1. We formulate visual token pruning as a lossy data compression problem optimized for the rate-distortion trade-off. The retained-token ratio defines the rate, while the behavioral discrepancy induced by pruning defines the distortion. Here, the rate directly reflects the inference cost of interest. Under this formulation, the key question is not simply whether a token is important, but whether its contribution to downstream performance justifies its token cost. Joint R-D optimization therefore allows the model to determine both how many tokens to retain and which tokens to retain, while adaptively allocating a shared token budget across inputs of different difficulty.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

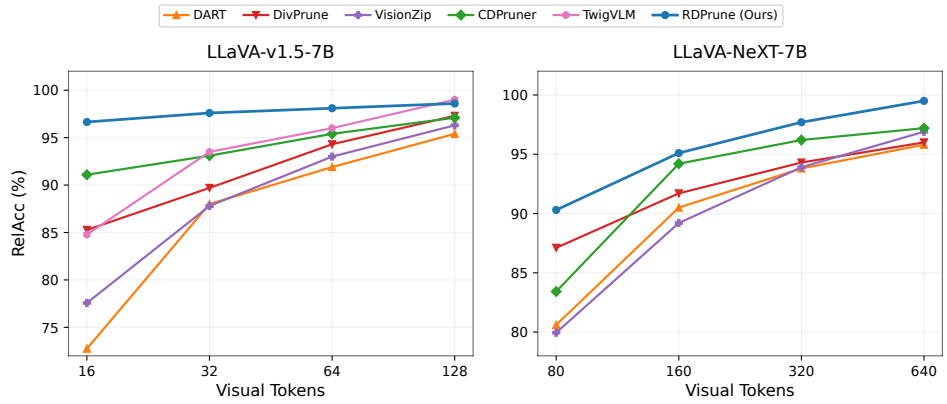

Figure 1: Performance comparison under different visual token budgets on LLaVA-v1.5-7B (left) and LLaVA-NeXT-7B (right).

The remaining challenge is how to measure pruning-induced distortion. Existing methods usually estimate token importance before pruning using proxy signals such as attention, saliency, or feature similarity. These signals indicate which tokens may be important, but do not directly measure how token removal changes the downstream model. In contrast, we define distortion by comparing model behavior before and after pruning, which more closely corresponds to the notion of distortion in lossy compression.. Specifically, we measure pruning-induced changes at three complementary levels: task performance, output distributions, and hidden representations. This provides a direct measure of how pruning affects downstream model behavior rather than relying solely on pre-pruning importance proxies.

Based on this formulation, we introduce **RDPrune**, a lightweight coarse-to-fine visual token compression framework. RDPrune first removes redundant visual tokens after the vision encoder and then performs further pruning at an intermediate layer of the LLM, where the representations have become more task-aware. At each stage, learnable binary existence indicators determine which tokens are retained. The retained-token ratio defines the rate, while task loss, output-distribution KL divergence, and hidden-state discrepancy jointly model the distortion. By optimizing rate and distortion together, RDPrune progressively compresses visual tokens while preserving downstream model behavior.

Our contributions are summarized as follows:

- **We formulate visual token pruning as a lossy data compression problem optimized for the rate-distortion trade-off.** By jointly optimizing the retained-token rate and pruning-induced behavioral discrepancy, the model can adaptively determine both how many tokens to retain and which tokens to retain across inputs of different difficulty.
- **We introduce a lightweight coarse-to-fine token compression framework.** Visual tokens are progressively pruned after the vision encoder and within the LLM, allowing further compression as the representations become increasingly task-aware.
- **We introduce a downstream-aware distortion measure based on model behavior before and after pruning.** Distortion is modeled at the task, prediction, and representation levels, directly measuring the effect of token removal rather than relying on pre-pruning importance proxies.

2 Related Work

2.1 Training-Free Visual Token Reduction

Training-free visual token reduction improves LVLM efficiency without additional optimization and can be directly applied to pretrained models. Existing methods mainly reduce tokens based on attention, redundancy, and diversity. Attention-based approaches estimate token importance from model attention patterns. FastV observes that visual attention becomes concentrated in intermediate

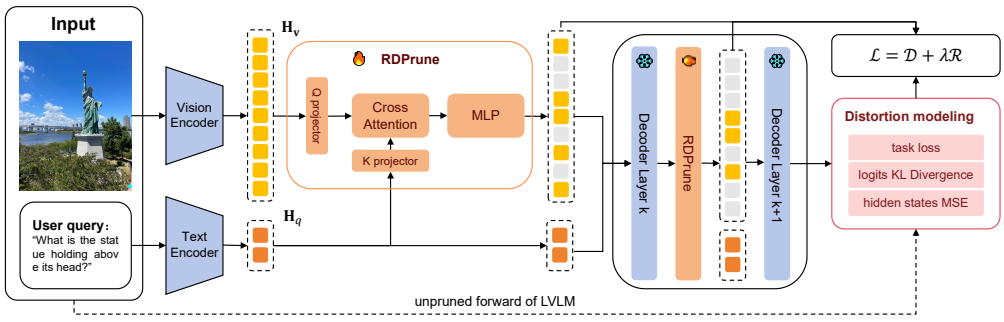

Figure 2: Overview of **RDPrune**. RDPrune performs visual token selection in two stages: early coarse pruning removes task-irrelevant tokens before LLM processing, while mid-layer refinement further eliminates redundant visual representations based on contextualized multimodal features.

LLM layers and removes tokens with low attention scores (Chen et al., 2024). SparseVLM exploits text-guided attention for adaptive sparsification (Zhang et al., 2025c), while FasterVLM and HiRED perform vision-side pruning before language-model computation (Zhang et al., 2024; Arif et al., 2025).

Another category focuses on token redundancy and diversity. DivPrune preserves representative tokens by encouraging feature diversity (Alvar et al., 2025), DART removes duplicated visual information (Wen et al., 2025), and VisionZip exploits dominant tokens and contextual redundancy for aggressive compression (Yang et al., 2025). Other methods further explore progressive or adaptive pruning strategies, such as PyramidDrop, MustDrop, and PruneSID (Xing et al., 2025; Liu et al., 2024a; Fang et al., 2026). Although effective, these methods mainly rely on surrogate signals measured near the pruning stage, such as attention or feature similarity, without explicitly modeling the downstream functional distortion caused by token removal.

2.2 Training-Based Visual Token Reduction

Training-based methods introduce learnable token selectors or compression modules to adapt token reduction to specific models and data distributions. ATP-LLaVA learns adaptive token retention policies conditioned on inputs and layers (Ye et al., 2025a), while LearnPruner and GlimpsePrune employ lightweight modules for multi-stage or instance-aware pruning (Takezoe et al., 2026; Zeng et al., 2026). CoViPAL and VisionSelector further optimize token selection through downstream supervision (Tang et al., 2025; Zhu et al., 2025).

Beyond token selection, several approaches learn compact visual representations or adaptive compression policies. TokenPacker and VoCo-LLaMA compress visual information through learned token aggregation or compression tokens (Li et al., 2025; Ye et al., 2025b), while DUET-VLM and VisPCO explore stage-wise compression and efficiency-aware optimization (Singh et al., 2026; Ji et al., 2026). RUTA further formulates visual token allocation from a rate-distortion perspective (Zou et al., 2026). However, existing training-based methods mainly optimize task loss, importance estimation, or compression constraints, leaving the intricate relationship between token reduction and downstream functional deviation an important direction for further investigation.

3 Method

3.1 Model Overview

We introduce **RDPrune**, a lightweight coarse-to-fine visual token pruning framework jointly optimized under a rate-distortion objective, as illustrated in Figure 2. Given an image and a user instruction, the visual encoder first produces a sequence of visual tokens, while the instruction is represented as textual tokens. RDPrune progressively reduces the visual sequence length before and between the LLM layers. The first pruning stage is placed immediately after the visual encoder and predicts a binary retention decision for each visual token using both visual and instruction representations. The remaining tokens are then processed together with the textual instructions by the LLM.

At the intermediate decoder layer of LLM, a second selector further removes redundant visual tokens. Instead of prescribing token importance using a fixed heuristic, RDPrune learns these pruning decisions from downstream supervision and optimizes them jointly through rate and distortion.

3.2 COARSE-TO-FINE SELECTION

RDPrune parameterizes the visual token rate through two structurally identical selectors, which progressively compress visual tokens. Each selector contains a mask predictor and a selection operator. The mask predictor estimates whether each visual token should be retained, while the selection operator removes tokens according to the predicted mask. The coarse and fine selectors share the same architecture and differ only in their learnable parameters, θ_1 and θ_2. This unified design allows the selector to be applied at different stages of an LVLM without changing its formulation.

Unified selector. Formally, let $\mathbf{H}^{(l)} = [\mathbf{H}_s^{(l)}; \mathbf{H}_v^{(l)}, \mathbf{H}_q^{(l)}]$ denote the hidden states at the decoder layer l, which $\mathbf{H}_s^{(l)}$, $\mathbf{H}_v^{(l)}$, $\mathbf{H}_q^{(l)}$ denote the system prompt tokens, visual tokens and user instruction tokens, respectively. With $\mathbf{H}_v^{(l)}$ and $\mathbf{H}_q^{(l)}$ as input, the mask predictor $\mathcal{G}(\mathbf{H}_v^{(l)}, \mathbf{H}_q^{(l)}; \theta)$ at the decoder layer l outputs a predicted degree of retention for each visual token

$$\mathbf{P} = \mathcal{G}\left(\mathbf{H}_v^{(l)}, \mathbf{H}_q^{(l)}; \theta\right) \in \mathbb{R}^{N_v}, \quad N_v = |\mathbf{H}_v^{(l)}|, \tag{1}$$

We obtain the binary retention mask by quantizing the predicted degree of retention:

$$m_i = \mathrm{B}\left(p_i\right) \tag{2}$$

where $\mathrm{B}(\cdot)$ denotes a binary quantizer and $m_i \in \{0, 1\}$.

After obtaining mask $\mathbf{m}$, the selection operator π produces a subset of visual tokens.

$$\pi(\mathbf{H}_v^{(l)}, \mathbf{m}) = \left\{ h_{v,i}^{(l)} | m_i = 1 \right\}. \tag{3}$$

At inference time, π directly removes tokens with $m_i = 0$, so subsequent layers process only the retained subset. During training, the same selection behavior is implemented by masked attention to preserve gradient propagation to the mask predictor. The differentiable implementation of $\mathrm{B}(\cdot)$ during training and the mechanism for applying the learned masks to suppress pruned visual tokens in the LLM forward process are provided in the Appendix A.1.

Stage I: coarse selection. The first selector operates before the first LLM layer. Given the original input $\mathbf{H}_v^{(0)}$, $\mathbf{H}_q^{(0)}$, it produces the mask and then applies the selection operation in Equations 4, 5.

$$\boldsymbol{m}^{(\mathrm{coarse})} = \mathrm{B}\left(\mathcal{G}\left(\mathbf{H}_v^{(0)}, \mathbf{H}_q^{(0)}; \theta_1\right)\right). \tag{4}$$

$$\mathbf{H}_v^{(\mathrm{coarse})} = \pi\left(\mathbf{H}_v^{(0)}, \mathbf{m}^{(\mathrm{coarse})}\right). \tag{5}$$

The selected visual tokens $\mathbf{H}_v^{(\mathrm{coarse})}$ and instruction tokens $\mathbf{H}_q^{(0)}$ are then processed by the first k layers of the LLM, producing intermediate representations $\mathbf{H}_v^{(\mathrm{coarse}),(k)}$ and $\mathbf{H}_q^{(k)}$.

Stage II: fine selection. The second selector has the same structure but uses independent parameters θ_2 and operates on the intermediate representations after the first k LLM layers. The selected visual token in fine grain is applied as

$$\mathbf{H}_v^{(\mathrm{fine})} = \pi\left(\mathbf{H}_v^{(\mathrm{coarse}),(k)}, \boldsymbol{m}^{(\mathrm{fine})}\right). \tag{6}$$

where $\boldsymbol{m}^{(\mathrm{fine})} = \mathrm{B}\left(\mathcal{G}\left(\mathbf{H}_v^{(\mathrm{coarse}),(k)}, \mathbf{H}_q^{(k)}; \theta_2\right)\right)$. Since the intermediate visual representations have already interacted with the instruction tokens and accumulated richer task context, Stage II can make more task-aware retention decisions than Stage I. The remaining LLM layers process $\mathbf{H}_v^{(\mathrm{fine}),(k)}$ together with $\mathbf{H}_q^{(k)}$ to produce the final output.

Overall, RDPrune forms a coarse-to-fine compression hierarchy, regardless of the layer index of the LLM:

$$\mathbf{H}_v \xrightarrow{\text{Stage I}} \mathbf{H}_v^{(\mathrm{coarse})} \xrightarrow{\text{Stage II}} \mathbf{H}_v^{(\mathrm{fine})}, \tag{7}$$

where Stage I removes early visual redundancy and Stage II performs further task-aware compression.

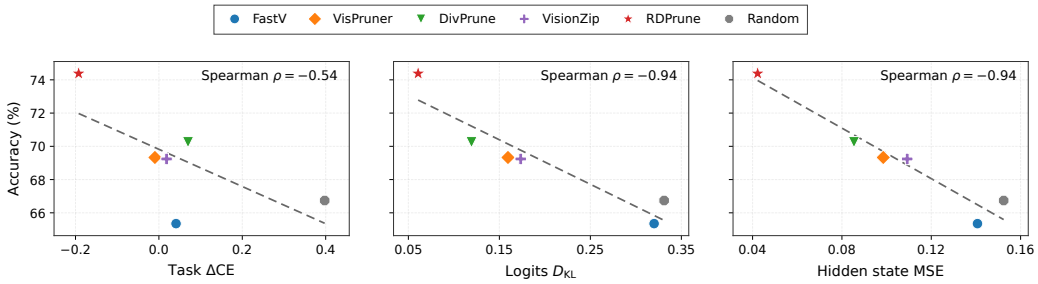

Figure 3: Relationship between task distortion, output-distribution distortion, hidden-representation distortion, and downstream performance under different visual-token pruning configurations.

3.3 Rate Modeling

The binary retention masks determine the visual token rate in the rate-distortion objective, allowing RDPrune to jointly learn which tokens to retain and how many tokens are worth retaining. We model the coarse-to-fine rate from the perspective of the total number of active visual tokens across LLM layers. Suppose LLM contains L decoder layers and Stage II is applied after layer k. Therefore, the number of remaining visual tokens each decoder layer is shown in Equation 8.

$$N_v^{(l)} = \begin{cases} |\mathbf{H}_v^{(\mathrm{I}),(l)}|, & 1 \le l \le k \\ |\mathbf{H}_v^{(\mathrm{II}),(l)}|, & k < l \le L \end{cases}. \tag{8}$$

We normalize this quantity by the number of original visual tokens and define the resulting measure as the final rate $\mathcal{R}$ of visual token sequence:

$$\boxed{\mathcal{R} = \frac{\sum_{l=1}^{L} N_v^{(l)}}{LN_v} = \frac{k\sum_{i=1}^{N_v} m_i^{(\mathrm{coarse})} + (L-k)\sum_{i=1}^{N_v} m_i^{(\mathrm{coarse})} m_i^{(\mathrm{fine})}}{LN_v}}. \tag{9}$$

3.4 Downstream Functional Distortion Modeling

Most existing methods estimate pruning distortion using proxy signals, such as attention, text-to-vision relevance, or feature diversity. However, these proxies measure token importance from intermediate representations rather than the downstream behavioral changes caused by token removal.

We therefore characterize pruning distortion through downstream functional deviations. As shown in Figure 3, we evaluated different methods on several benchmarks and measured the difference between downstream behavior and the original model during evaluation. These observations illustrate that larger deviations in prediction distributions and hidden states correlate with greater accuracy degradation, while task loss provides a complementary signal, motivating three complementary distortion terms: task fidelity, output distribution consistency and hidden state consistency. Let $\mathbf{z}_t^F$ and $\mathbf{H}_t^{F,(L)}$ denote the logits and final layer hidden states of the full unpruned model, and $\mathbf{z}_t^P$ and $\mathbf{H}_t^{P,(L)}$ denote those of the pruned model, where $t \in \mathcal{T}$ indexes supervised answer tokens.

Task fidelity. We directly use the autoregressive loss of the pruned model:

$$\mathcal{L}_{\mathrm{task}} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} \mathrm{CE}\left(\mathbf{z}_t^P, y_t\right), \tag{10}$$

where y_t is the ground-truth answer token, and $\mathrm{CE}(\cdot)$ denotes the cross-entropy function.

Output distribution consistency. To preserve the predictive distribution of the reference model, we minimize:

$$\mathcal{L}_{\mathrm{out}} = \frac{T^2}{|\mathcal{T}|}\sum_{t\in\mathcal{T}} D_{\mathrm{KL}}\left(\mathrm{softmax}\left(\frac{\mathbf{z}_t^F}{T}\right) \middle\| \mathrm{softmax}\left(\frac{\mathbf{z}_t^P}{T}\right)\right), \tag{11}$$

where T is the temperature coefficient. $\mathcal{L}_{\mathrm{out}}$ constrains the complete predictive distribution and therefore provides a denser signal for measuring pruning-induced behavioral changes.

Hidden states representation consistency. We further preserve the internal representation supporting downstream prediction:

$$\mathcal{L}_{\text{hid}} = \frac{1}{|\mathcal{T}|d} \sum_{t \in \mathcal{T}} \left\| \mathbf{H}_t^{F,(L)} - \mathbf{H}_t^{P,(L)} \right\|_2^2, \tag{12}$$

where d denotes the hidden-state dimension. $\mathcal{L}_{\text{hid}}$ encourages the compressed model to make deeper downstream representations closer to the unpruned model.

The overall functional distortion can be written as:

$$\boxed{\mathcal{D} = \beta_{\text{task}} \mathcal{L}_{\text{task}} + \beta_{\text{out}} \mathcal{L}_{\text{out}} + \beta_{\text{hid}} \mathcal{L}_{\text{hid}}}, \tag{13}$$

where β_{task}, β_{out} and β_{hid} are weighting coefficients.

3.5 Joint Rate-Distortion Optimization

Given visual tokens $\mathbf{H_v}$ and instruction tokens $\mathbf{H_q}$, most existing visual token pruning methods operate under an externally specified token budget K. Their objective can be abstracted as selecting the best subset of exactly K tokens according to a proxy utility as shown in Equation 14.

$$S_K = \underset{S \subseteq \mathbf{H_v}, |S|=K}{\operatorname{argmax}} \; u\left(S; \mathbf{H_v}, \mathbf{H_q}\right), \tag{14}$$

where $u(S; \mathbf{H_v}, \mathbf{H_q})$ measures token utility using a predefined proxy. Under this formulation, K is fixed independently of the resulting downstream distortion. Different samples or tasks are therefore forced to operate under the same prescribed budget, even though they may require different amounts of visual information.

RDPrune removes this fixed-budget constraint and jointly optimizes the visual-token rate and pruning-induced distortion. The backbone LVLM is kept frozen, and only the selector parameters $\theta = \theta_1, \theta_2$ are optimized. Given a training set $\mathcal{S}$ consisting of input-label pairs $(\mathbf{x}, y)$, where $\mathbf{x} = (\mathbf{H_v}, \mathbf{H_q})$ comprises the visual and user instruction tokens, we optimize the selector parameters as:

$$\boxed{\theta^* = \underset{\theta}{\operatorname{argmin}} \; \mathbb{E}_{(\mathbf{x},y)\sim\mathcal{S}} \left[\mathcal{D}\left(x, y; \theta\right) + \lambda \mathcal{R}\left(x; \theta\right)\right]}, \tag{15}$$

where $\mathcal{D}$ measures the downstream behavioral distortion caused by pruning, $\mathcal{R}$ measures the retained visual-token rate, and λ controls their trade-off.

Unlike fixed-budget pruning, this objective does not prescribe how many tokens each sample must retain. Instead, the shared rate constraint is optimized over the data distribution, allowing token resources to adapt across samples and tasks: inputs that require richer visual evidence can retain more tokens, while easier inputs can be represented with fewer. Through the differentiable retention masks, gradients from both $\mathcal{D}$ and $\mathcal{R}$ are propagated to the selectors. RDPrune therefore learns jointly which tokens to retain and how many are worth retaining according to their downstream utility under an explicit rate-distortion trade-off.

4 Experiments

We evaluate RDPrune on multiple visual question answering and multimodal understanding benchmarks to investigate its trade-off between task performance and inference efficiency. We compare against existing vision token pruning methods under different retention ratios and report both benchmark performance and practical inference costs.

4.1 Experimental Setup

We train RDPrune on 20,000 randomly sampled examples from GQA (Hudson & Manning, 2019), while keeping the backbone VLM frozen and optimizing only the two pruning selectors. Training is performed on four NVIDIA A100 80GB GPUs and completes within one hour. The two selectors are inserted after the vision encoder and at the LLM layer 12, respectively. During training, pruning is simulated without physically removing tokens, whereas hard token removal is applied at inference. To facilitate comparison with other methods, we set stage II to fixed top-K pruning during evaluation, while stage I allowed the model to determine the retention rate. More implementation details are shown in A.2

Method	GQA	SQA^I	VQA^T	POPE	MME	VQA^{v2}	MMB	MMB^{CN}	Avg
Upper Bound, 576 Tokens									
Vanilla	61.9	69.5	58.2	85.9	1862	78.5	64.7	58.3	100.0%
Retain Averaged 128 Tokens									
FastV	49.6	60.2	50.6	59.6	1490	61.8	56.1	56.4	83.2%
DART	57.9	69.1	56.3	80.4	1721	74.7	60.7	57.3	95.4%
DivPrune	59.3	69.0	56.1	86.7	1718	76.0	62.0	54.8	96.4%
VisPruner	58.2	69.1	57.0	84.6	1794	75.8	62.7	57.3	97.2%
VisionZip	57.6	68.9	56.8	83.2	1762	75.6	62.0	56.7	96.3%
CDPruner	59.9	68.6	56.2	87.4	1745	76.6	63.1	55.0	97.1%
TwigVLM	60.6	69.5	57.8	86.6	1818	77.9	63.5	55.6	98.6%
LearnPruner	60.3	68.5	57.3	86.7	1820	77.3	63.8	56.8	98.5%
RDPrune	61.5	68.4	57.6	85.8	1824	78.0	63.9	56.1	98.6%
Retain Averaged 64 Tokens									
FastV	46.1	51.1	47.8	48.0	1256	55.0	48.0	52.7	73.5%
DART	54.7	69.3	54.7	73.9	1705	71.3	59.5	54.0	91.9%
DivPrune	57.8	68.2	54.7	85.6	1674	74.1	59.3	52.3	93.9%
VisPruner	55.4	69.1	55.8	80.4	1689	72.7	61.3	55.1	93.9%
VisionZip	55.1	69.0	55.5	77.0	1690	72.4	60.1	55.4	93.0%
CDPruner	58.6	68.1	55.3	87.1	1717	75.4	61.1	53.2	95.4%
TwigVLM	58.8	70.0	55.8	82.7	1760	75.6	60.4	53.1	95.4%
LearnPruner	58.9	68.3	56.6	86.8	1750	76.0	62.6	55.7	96.9%
RDPrune	61.0	68.4	57.3	85.9	1818	76.9	63.8	56.0	98.2%
Retain Averaged 32 Tokens									
FastV	41.5	42.6	42.5	32.5	1090	43.4	37.8	33.2	58.5%
DART	52.9	69.3	52.2	69.1	1615	67.1	58.5	50.0	88.0%
DivPrune	54.9	68.6	52.9	81.5	1611	71.2	57.6	49.1	90.5%
VisPruner	52.2	69.2	53.9	72.7	1567	67.7	58.4	52.7	89.0%
VisionZip	51.8	68.8	53.1	68.7	1536	67.1	57.7	50.3	87.2%
CDPruner	57.0	69.0	53.2	87.4	1657	73.6	59.6	49.6	93.1%
LearnPruner	57.2	68.2	56.1	84.5	1672	74.0	60.8	55.5	94.8%
RDPrune	59.6	68.5	56.4	86.1	1826	75.4	63.6	56.0	97.6%

Table 1: Performance comparisons of different pruning methods on LLaVA-v1.5-7B. 'Avg' denotes the average performance on all benchmarks retained relative to the unpruned model (set to 100%). MME is reported in its original score units.

4.2 Main Results

Results on LLaVA-v1.5-7B. Table 1 compares RDPrune with representative training-free pruning methods (Chen et al., 2024; Wen et al., 2025; Alvar et al., 2025; Zhang et al., 2025a; Yang et al., 2025; Zhang et al., 2025b) and training-based pruning methods (Shao et al., 2025; Takezoe et al., 2026). RDPrune advantage becomes more pronounced as the budget tightens: at 64 and 32 tokens, RDPrune retains 98.1% and 97.6% of the original performance. These results highlight the effectiveness of RDPrune in the low-budget regime, where preserving downstream functionality with a small retained subset becomes particularly challenging. More importantly, RDPrune exhibits a substantially flatter budget-performance profile. Reducing the token budget fourfold, from 128 to 32, decreases its average relative performance by only 1.0 percentage point. This cross budget advantage indicates that RDPrune tolerates more aggressive compression before incurring rapid performance degradation.

Results on LLaVA-NEXT-7B. In Table 2, we further evaluate RDPrune on LLaVA-NeXT-7B, which contains up to 2880 visual tokens. RDPrune consistently maintains strong performance under different compression settings, preserving 95.1% of the original performance with only 160 tokens.

Results on Qwen3-VL-8B. In Table 3, we further evaluate RDPrune on Qwen3-VL-8B under three visual token retention rates. RDPrune consistently maintains strong performance across different token budgets. As the retention rate decreases from 33.3% to 11.1%, the average performance of FastV and VisionZip drops by 18.4 and 17.0 percentage points, respectively, while RDPrune drops by only 11.0 points, demonstrating better robustness under aggressive compression.

Method	$\mathbf{VQA^{v2}}$	$\mathbf{VQA^{T}}$	GQA	$\mathbf{SQA^{I}}$	MME	MMB	Avg
Upper Bound, 2880 Tokens							
Vanilla	81.2	60.3	64.2	70.2	1842	67.9	100%
Retain Averaged 640 Tokens							
DART	78.3	59.5	61.3	68.2	1721	64.9	96.1%
VisionZip	79.1	59.9	61.2	68.1	1787	65.8	97.2%
DivPrune	79.3	57.0	61.9	67.8	1760	65.8	96.3%
CDPruner	79.9	58.4	62.6	67.9	1800	66.3	97.5%
LearnPruner	80.5	60.5	63.8	70.0	1837	67.5	99.6%
RDPrune	81.4	61.0	64.0	69.1	1796	67.7	99.5%
Retain Averaged 320 Tokens							
DART	75.7	57.6	59.5	67.5	1705	64.2	94.1%
VisionZip	76.2	58.8	58.9	67.5	1702	63.3	94.1%
DivPrune	77.2	56.2	61.1	67.7	1724	63.9	94.6%
CDPruner	78.4	57.4	61.6	67.8	1807	65.5	96.5%
LearnPruner	78.3	58.4	62.2	68.6	1845	66.8	97.7%
RDPrune	80.1	60.1	62.8	68.3	1761	66.1	97.7%
Retain Averaged 160 Tokens							
DART	72.5	54.9	56.8	67.8	1615	62.0	90.7%
VisionZip	71.4	55.0	55.2	67.9	1630	58.6	89.4%
DivPrune	75.0	54.1	59.3	67.1	1643	62.9	92.0%
CDPruner	76.7	55.4	60.8	67.5	1749	64.2	94.3%
LearnPruner	76.2	55.0	58.7	67.6	1784	65.3	94.0%
RDPrune	77.5	58.2	60.6	68.4	1689	64.6	95.1%

Table 2: Performance comparisons of different pruning methods on LLaVA-NeXT-7B.

Method	$\mathbf{VQA^{v2}_{lite}}$	$\mathbf{VQA^{T}}$	GQA	$\mathbf{SQA^{I}}$	MME	MMB	Avg
Upper Bound, 1280 Tokens							
Vanilla	79.6	82.3	61.5	94.5	2373	83.4	100%
Retain Averaged 426 Tokens							
FastV	73.5	71.4	55.8	86.5	1977	77.0	89.5%
VisionZip	75.3	77.6	59.1	89.8	2106	79.7	94.1%
RDPrune	77.4	76.8	60.4	91.6	2189	80.4	95.7%
Retain Averaged 284 Tokens							
FastV	66.5	63.0	52.8	82.4	1887	71.1	83.0%
VisionZip	70.2	70.9	56.7	88.5	1957	76.5	89.1%
RDPrune	76.1	71.5	59.5	89.2	2093	77.8	92.5%
Retain Averaged 142 Tokens							
FastV	57.2	47.2	47.2	77.1	1639	58.4	71.1%
VisionZip	60.9	46.5	52.6	83.0	1772	68.1	77.1%
RDPrune	68.2	57.3	57.1	85.9	1933	73.1	84.7%

Table 3: Performance comparisons of different pruning methods on Qwen3-VL-8B. $\mathbf{VQA^{v2}_{lite}}$ is a subset of VQA-v2 consisting of 500 samples.

Results on different rate coefficients Different models are obtained by training with different λ values. Instead of manually specifying the number of retained visual tokens, RDPrune automatically determines the token rate according to the rate-distortion objective. As shown in Figure 4, increasing λ encourages stronger visual token compression and produces different rate-distortion operating points. The resulting models consistently maintain competitive performance across various token budgets, demonstrating the flexibility of RDPrune for adaptive visual token reduction.

4.3 Efficiency Analysis

We evaluate the computational efficiency of RDPrune on LLaVA-v1.5-7B and LLaVA-NeXT-7B using a single NVIDIA A100 80GB GPU. As shown in Table 4, retaining only 11.1% of the original visual tokens substantially reduces computation and KV-cache usage. On LLaVA-v1.5-7B, RDPrune reduces KV cache from 382.0MB to 146.8MB, while achieving $2.0\times$ faster prefill and $1.6\times$ faster end-to-end inference. The gains are more pronounced for LLaVA-NeXT-7B with longer visual sequences, reaching $3.7\times$ prefill and $2.8\times$ total speedups, together with a 75.3% reduction in KV cache memory.

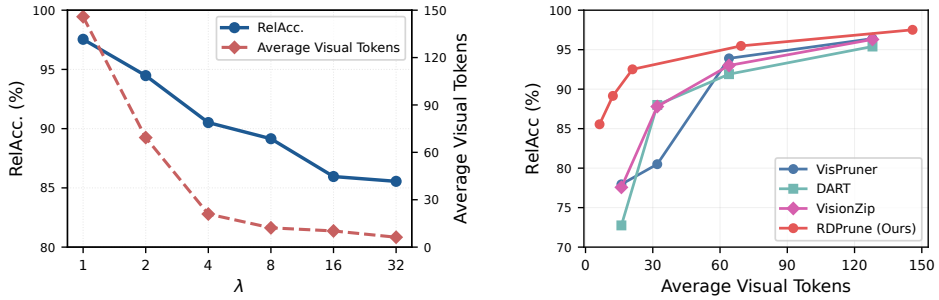

Figure 4: **Left** :Relationship between Rate and model performance and the average retained number of visual tokens under different λ. **Right** :Performance comparison of RD models trained with different λ and various comparison methods.

Method	Tokens	TFLOPs	GPU Memory (GB)	KV Cache (MB)	Prefill Time	Total Time
LLaVA-v1.5-7B	576	10.6	14.2	382.0	55.9 (1.0×)	78.1 (1.0×)
+ RDPrune	64	4.3	13.7	146.8	28.0 (2.0×)	49.2 (1.6×)
LLaVA-NeXT-7B	2880	36.3	16.3	1209.4	226.8 (1.0×)	238.1 (1.0×)
+ RDPrune	320	10.0	14.1	298.5	61.7 (3.7×)	84.2 (2.8×)

Table 4: Efficiency analysis of RDPrune on LLaVA-v1.5-7B and LLaVA-NeXT-7B

4.4 Ablation Studies

We conduct ablation studies to examine the key design choices of RDPrune. In addition to the distortion components analyzed below, further ablations on the Stage-II pruning layer and the coarse-to-fine pruning design are provided in Appendix A.3.

$\mathcal{L}_{task}$	$\mathcal{L}_{out}$	$\mathcal{L}_{hid}$	Avg
LLaVA-v1.5-7B			100%
✓	×	×	92.8
×	✓	×	95.0
×	×	✓	96.1
✓	✓	×	95.1
✓	×	✓	96.3
×	✓	✓	96.3
✓	✓	✓	97.6

Table 5: Ablation study of different components of distortion function.

The ablation of distortion components. The effectiveness of the proposed distortion modeling is crucial to the success of RDPrune. We design a comparative experiment to verify the role of each Distortion metric in the training process. We train the model multiple times on LLaVA-v1.5-7B, disabling one or two distortion components during training. Then, we evaluate the model on benchmarks, retaining 32 visual tokens at the end, and observe the changes in model performance. The experimental results are shown in Table 5. All three distortion components are effective. Among individual distortion terms, $\mathcal{L}_{hid}$ achieves the strongest performance, indicating that preserving hidden representations provides an effective signal for maintaining downstream functionality.

5 Conclusion

In this work, we formulate visual token pruning as a rate-distortion optimization problem, where token rate and downstream distortion are jointly considered under a unified framework. Based on this formulation, we propose RDPrune, a coarse-to-fine visual token compression framework that effectively balances computational efficiency and model performance. Extensive experiments demonstrate that RDPrune achieves favorable rate-distortion trade-offs across various benchmarks and compression levels. However, RDPrune still requires lightweight distortion-aware training, which limits its applicability compared with fully training-free pruning methods. Moreover, the current rate formulation mainly relies on token ratio as a proxy for computational cost. Future work will explore more comprehensive hardware-aware rate modeling that better reflects practical inference efficiency.

AI USE STATEMENT

In this work, we used generative AI tools to assist with manuscript polishing, experimental and evaluation script development, and table and figure preparation. We did not use generative AI to generate experimental data, measurements, or reported results. All AI-assisted text, code, and visual materials were reviewed and verified by the authors, and all generated or modified code was executed and tested before use. All numerical results reported in this paper were obtained from our own experiments and checked against the corresponding outputs and logs. We take responsibility for the final content of this work, including all text, claims, code, figures, and other artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies efficient visual token pruning for large vision-language models and involves no human subjects, collection of personal data, or sensitive attributes. All experiments use publicly available models and established vision-language benchmarks (Goyal et al., 2017; Singh et al., 2019; Hudson & Manning, 2019; Lu et al., 2022; Li et al., 2023; Fu et al., 2025; Liu et al., 2024b) under their respective licenses and terms of use. Baseline methods (Yang et al., 2025; Chen et al., 2024; Alvar et al., 2025; Zhang et al., 2025a; Takezoe et al., 2026; Shao et al., 2025; Zhang et al., 2025b; Wen et al., 2025) are evaluated using their publicly released implementations whenever available, following the corresponding licenses and recommended settings. We are aware of no additional privacy, discriminatory, or dual-use concerns.

REPRODUCIBILITY STATEMENT

We provide the information necessary to reproduce our experiments throughout the main paper and appendix. The section 3 specifies coarse-to-fine pruning procedure and the jointly rate-distortion optimization. The section 4 reports the evaluated baseline models, the comparison methods, the training datasets, benchmarks and pruning budgets. Additional implementation details are provided, including selector architectures, pruning locations, optimization settings, loss coefficients, and training hyperparameters in the Appendix A.2, as well as the key steps to achieve differentiability of the training process in the Appendix A.1. We will release the model training code.

REFERENCES

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

Kazi Hasan Ibn Arif, JinYi Yoon, Dimitrios S. Nikolopoulos, Hans Vandierendonck, Deepu John, and Bo Ji. Hired: Attention-guided token dropping for efficient inference of high-resolution vision-language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(2): 1773–1781, 2025. doi: 10.1609/aaai.v39i2.32171.

Changwoo Baek, Jouwon Song, Sohyeon Kim, and Kyeongbo Kong. Agilepruner: An empirical study of attention and diversity for adaptive visual token pruning in large vision-language models. In *The Fourteenth International Conference on Learning Representations*, 2026.

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

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

Zhengyao Fang, Pengyuan Lyu, Chengquan Zhang, Guangming Lu, Jun Yu, and Wenjie Pei. Prune redundancy, preserve essence: Vision token compression in VLMs via synergistic importance-diversity. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=i36E5Ezm0H.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, Rongrong Ji, Caifeng Shan, and Ran He. MME: A comprehensive evaluation benchmark for multimodal large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4899.

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

Drew A. Hudson and Christopher D. Manning. GQA: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *International Conference on Learning Representations*, 2017.

Huawei Ji, Yuanhao Sun, Yuan Jin, Cheng Deng, Jiaxin Ding, Luoyi Fu, and Xinbing Wang. Vispco: Visual token pruning configuration optimization via budget-aware pareto-frontier learning for vision-language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 9281–9301. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.420.

Ao Li, Yuxiang Duan, Jinghui Zhang, Congbo Ma, Yutong Xie, Gustavo Carneiro, Mohammad Yaqub, and Hu Wang. Transprune: Token transition pruning for efficient large vision-language model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39529–39538, 2026.

Wentong Li, Yuqian Yuan, Jian Liu, Dongqi Tang, Song Wang, Jie Qin, Jianke Zhu, and Lei Zhang. Tokenpacker: Efficient visual projector for multimodal llm. *International Journal of Computer Vision*, 133:6794–6812, 2025. doi: 10.1007/s11263-025-02491-7.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 292–305, Singapore, 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.20.

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

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

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. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pp. 216–233. Springer, 2024b.

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

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949, 2021.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. In *ICCV*, 2025.

Zhenwei Shao, Mingyang Wang, Zhou Yu, Wenwen Pan, Yan Yang, Tao Wei, Hongyuan Zhang, Ning Mao, Wei Chen, and Jun Yu. Growing a twig to accelerate large vision-language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20064–20074, 2025.

Aditya Kumar Singh, Hitesh Kandala, Pratik Prabhanjan Brahma, Zicheng Liu, and Emad Barsoum. Duet-vlm: Dual stage unified efficient token reduction for vlm training and inference. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17651–17660, 2026.

Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8317–8326, 2019.

Guohao Sun, Yufei Wang, Sizhuo Ma, Yuege Xie, Yuting Cheng, Zhiqiang Tao, and Jian Wang. Ifprune: Information-flow guided token pruning for efficient vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3522–3531, 2026.

Yizheng Sun, Yanze Xin, Hao Li, Jingyuan Sun, Chenghua Lin, and Riza Batista-Navarro. Lvpruning: An effective yet simple language-guided vision token pruning approach for multi-modal large language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 4299–4308, 2025. doi: 10.18653/v1/2025.findings-naacl.242.

Rinyoichi Takezoe, Yaqian Li, Zihao Bo, Anzhou Hou, Mo Guang, and Kaiwen Long. Learnpruner: Rethinking attention-based token pruning in vision language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Dxb6gBJHby.

Zicong Tang, Ziyang Ma, Suqing Wang, Zuchao Li, Lefei Zhang, Hai Zhao, Yun Li, and Qianren Wang. Covipal: Layer-wise contextualized visual token pruning for large vision-language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 20701–20714. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.1127.

Zichen Wen, Yifeng Gao, Shaobo Wang, Junyuan Zhang, Qintong Zhang, Weijia Li, Conghui He, and Linfeng Zhang. Stop looking for "important tokens" in multimodal language models: Duplication matters more. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 9961–9980. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.emnlp-main.505.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. Conical visual concentration for efficient large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14593–14603, 2025. doi: 10.1109/CVPR52734.2025.01360.

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

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

Xubing Ye, Yukang Gan, Xiaoke Huang, Yixiao Ge, and Yansong Tang. Voco-llama: Towards vision compression with large language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 29836–29846, 2025b. doi: 10.1109/CVPR52734.2025.02777.

Quan-Sheng Zeng, Yunheng Li, Qilong Wang, Peng-Tao Jiang, Zuxuan Wu, Ming-Ming Cheng, and Qibin Hou. A glimpse to compress: Dynamic visual token pruning for large vision-language models. *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2026. doi: 10.1109/TCSVT.2026.3702147.

Qizhe Zhang, Aosong Cheng, Ming Lu, Zhiyong Zhuo, Minqi Wang, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. [CLS] attention is all you need for training-free visual token pruning: Make vlm inference faster. *arXiv preprint arXiv:2412.01818*, 2024.

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

Qizhe Zhang, Mengzhen Liu, Lichen Li, Ming Lu, Yuan Zhang, Junwen Pan, Qi She, and Shanghang Zhang. Beyond attention or similarity: Maximizing conditional diversity for token pruning in mllms. In *Advances in Neural Information Processing Systems*, volume 38, 2025b.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, et al. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning*, 2025c.

Jiaying Zhu, Yurui Zhu, Xin Lu, Wenrui Yan, Dong Li, Kunlin Liu, Xueyang Fu, and Zheng-Jun Zha. Visionselector: End-to-end learnable visual token compression for efficient multimodal llms. *arXiv preprint arXiv:2510.16598*, 2025. Accepted to ACM Multimedia 2026.

Jian Zou, Xiaoyu Xu, Zhihua Wang, Yilin Wang, Balu Adsumilli, and Kede Ma. Ruta: Principled visual token allocation via rate-utility optimization. *arXiv preprint arXiv:2608.04132*, 2026.

A APPENDIX

A.1 DETAILED IMPLEMENTATION OF RDPRUNE DIFFERENTIABILITY

Differentiable binary mask learning. In Section 3.2, we introduce the binary retention mask generated by the quantization function $\mathrm{B}(\cdot)$. This section provides the detailed implementation of this quantization process during training.

Formally, let $\mathbf{H_v}$ and $\mathbf{H_q}$ denote the visual tokens and instruction tokens, respectively. The mask predictor consisting of a cross-attention and an MLP block can be written as:

$$\mathbf{S} = \mathrm{MLP}\left(\mathrm{CrossAttention}\left(\mathbf{H_v}, \mathbf{H_q}\right)\right), \tag{16}$$

where $s_i \in \mathbf{S}$ represents the *keep* and *drop* scores of the visual token i, respectively. During training, these continuous scores are quantized into binary discrete retaining decisions using a differentiable categorical reparameterization called Gumbel-Softmax (Jang et al., 2017). The working principle is shown in Equations 17, 18.

$$g_{i,k} = -\log\left(-\log u_{i,k}\right), \qquad u_{i,k} \sim \mathrm{Uniform}(0,1), \tag{17}$$

$$p_{i,j} = \frac{\exp\left((a_{i,j} + g_{i,j})/\tau\right)}{\sum_{k\in\{\mathrm{keep},\mathrm{drop}\}} \exp\left((a_{i,k} + g_{i,k})/\tau\right)}, \quad j \in \{\mathrm{keep}, \mathrm{drop}\}, \tag{18}$$

where $p_i \in \mathbf{P}$ denotes the normalized *keep* and *drop* probability vector obtained by applying Gumbel-Softmax to the logits of the ith visual token, and is also the output of the function $\mathcal{G}$. τ is the temperature parameter of Gumbel-Softmax. The continuous normalized probability is quantized into discrete binary token retention mask. Gumbel-Softmax adopt a straight-through estimator (STE) to preserve the discrete decision in the forward pass while ensuring gradients can propagate backward.

$$m_i^{\mathrm{hard}} = \begin{cases} 1, & p_{i,\mathrm{keep}} \geq p_{i,\mathrm{drop}} \\ 0, & p_{i,\mathrm{keep}} < p_{i,\mathrm{drop}} \end{cases}. \tag{19}$$

$$m_i = \mathrm{sg}\left(m_i^{\mathrm{hard}} - p_{i,\mathrm{keep}}\right) + p_{i,\mathrm{keep}}, \tag{20}$$

where $\mathrm{sg}(\cdot)$ denotes the stop-gradient operator. The equations above are the specific implementation of $\mathrm{B}(\cdot)$. Additionally, during the inference stage, g_i is set to 0 while GumbelSoftMax becomes a regular softmax.

Mask-guided token suppression in LLM forward. During training, physically removing tokens from the sequence would break the differentiable computational path between downstream supervision and the pruning selector. Following the masked-attention strategy used in DynamicViT (Rao et al., 2021), we therefore retain the original tensor structure and inject the straight-through retention mask into subsequent attention computation. Conceptually, for an attention logit matrix $\mathbf{A}$ and an extended token mask $\mathbf{m}$, where textual tokens and system prompt tokens are always assigned a value of 1, the masked attention normalization can be expressed as:

$$\mathrm{SoftmaxMask}\left(\mathbf{A}, \mathbf{m}\right)_{ij} = \frac{\exp(A_{ij})\, m_j}{\sum_k \exp(A_{ik})\, m_k + \epsilon}, \tag{21}$$

where ϵ is a small constant for numerical stability. Consequently, visual tokens assigned a zero retention mask do not contribute to subsequent token interactions, while gradients from downstream objectives can still propagate through the straight-through estimator to the selector parameters. At inference time, the binary decision mask can instead be used for physical token removal.

A.2 IMPLEMENTATION DETAILS

Both pruning stages adopt the same lightweight query-conditioned selector. Visual tokens and user instruction tokens are first normalized and projected into a 512-dimensional latent space. A multi-head cross-attention module then conditions each visual token on the textual query, followed by a residual MLP. To estimate token importance, the module integrates token-level features with globally aggregated visual context and predicts a binary keep/drop score for each visual token.

Method	Stage I	Stage II	Avg Tokens	RelAcc(%)	TFLOPS	Prefill Time(ms)
RDPrune	256	64	136	93.3	3.0	23.8
w/o Stage II	256	–	256	95.0	4.9	32.5
w/o Stage I	–	64	256	94.0	4.1	28.8
w/o Stage II*	136	–	136	77.9	3.0	24.1

Table 6: Ablation study on the coarse-to-fine stages.

During training, differentiable binary masks are obtained using Gumbel-Softmax with a straight-through estimator, and this mask is input into the LLM to mask the corresponding visual tokens. We train the selectors for one epoch using AdamW with a learning rate of 1×10^{-4}, cosine decay, a warm-up ratio of 0.1. The coefficients for the task fidelity, output distribution distortion and hidden states distortion terms are set to $\beta_{\text{task}} = 0.05$, $\beta_{\text{out}} = 0.5$, and $\beta_{\text{hid}} = 1.0$, respectively, with a distillation temperature of $T = 2$. The rate regularization coefficient is set to $\lambda = 5.0$.

A.3 More Experimental Results

Ablation study on the coarse-to-fine pruning stages. We ablate the two pruning stages in Table 6. Removing Stage II increases relative accuracy from 93.3% to 95.0%, but also raises the computational cost from 3.0 to 4.9 TFLOPs and the prefill time from 23.8ms to 32.5ms. Similarly, performing pruning only at Stage II yields 94.0% relative accuracy, while requiring 4.1 TFLOPs and 28.8ms. More critically, when the Stage-I-only variant is adjusted to match RDPrune in average token count and computation, its relative accuracy drops substantially to 77.9%, despite nearly identical prefill time. These results indicate that aggressive early pruning causes substantial information loss, whereas postponing pruning incurs higher inference cost. The coarse-to-fine design therefore provides a more favorable accuracy–efficiency trade-off.

Ablation on pruning layer k of Stage II. We investigate the effect of the pruning layer for Stage II by placing the selector at different depths of the LLM. As shown in Table 7, moving the pruning layer deeper consistently improves the average performance, but also introduces higher computational cost. In particular, placing Stage II at layer 12 achieves 97.6% average performance with only 3.8 TFLOPs, while further delaying pruning only brings marginal performance gains at noticeably higher computation. We therefore select the layer 12 as the default pruning position for Stage II, providing a favorable trade-off between computational efficiency and model performance.

Layer idx	TFLOPs	RelAcc(%)
4	3.0	93.7
8	3.4	96.4
12	3.8	97.6
16	4.3	97.9
20	4.7	98.1

Table 7: Ablation study of pruning layer location of stage II.

A.4 Qualitative Examples

We present qualitative examples selected from VQAv2 (Goyal et al., 2017) to compare different visual token pruning strategies under aggressive compression as shown in Figures 5, 6. We consider FastV (Chen et al., 2024) and DivPrune (Alvar et al., 2025) as representative attention-based and diversity-based methods, respectively. FastV ranks visual tokens according to LLM attention scores and removes tokens with low attention, while DivPrune reduces redundancy by preserving a diverse subset of visual tokens in the feature space. RDPrune adopts a coarse-to-fine pruning strategy. In the coarse stage, it preserves query-relevant regions together with most semantically meaningful contextual information. The fine stage further refines the retained subset according to downstream functional distortion. As a result, with only 32 visual tokens, RDPrune often preserves the evidence required for correct prediction, whereas attention- or diversity-based pruning may discard critical regions. Moreover, we observe three failure cases involving spatial references. As shown in Figure 6, the failures mainly arise from excessive refinement rather than coarse localization errors. Stage-I successfully preserves the question-relevant regions, whereas Stage-II may discard fine-grained attribute cues, weaken relative spatial relationships, or reduce the visual coverage required for existence reasoning. These observations motivate richer distortion modeling that explicitly accounts for spatial relations and regional evidence preservation, together with stage-adaptive rate allocation to prevent excessive information loss during fine-grained pruning.

A.5 Evaluation Benchmarks

We evaluate our method on seven widely used benchmarks covering general visual question answering, scene-text understanding, compositional reasoning, scientific reasoning, hallucination, and comprehensive multimodal understanding.

VQA-v2 (Goyal et al., 2017) The VQA-v2 benchmark is a large-scale visual question answering benchmark built on natural images from MS-COCO. It reduces language priors by pairing visually similar images with the same question but different answers, and therefore mainly evaluates general visual perception and question answering capabilities. VQA^{v2}_{lite} is a subset of VQA-v2 consisting of 500 samples.

TextVQA (Singh et al., 2019) The TextVQA benchmark evaluates the ability of a model to understand and reason about textual information embedded in natural images. Answering its questions often requires jointly recognizing scene text, understanding the surrounding visual context, and reasoning over the extracted textual content. The dataset is denoted as VQA^T in our experiments.

GQA (Hudson & Manning, 2019) The GQA benchmark is designed for real-world visual reasoning and compositional question answering. Its questions are constructed based on scene graphs and cover object attributes, spatial relations, comparisons, and multi-step compositional reasoning, making it suitable for evaluating fine-grained visual reasoning.

ScienceQA (Lu et al., 2022) The ScienceQA benchmark is a multimodal multiple-choice benchmark covering diverse science topics and requiring both knowledge understanding and multi-step reasoning. Following common LVLM evaluation protocols, we report results on the image-based subset, denoted as SQA^I, which requires visual information to answer the corresponding science questions.

POPE (Li et al., 2023) The POPE benchmark is a polling-based benchmark specifically designed to evaluate object hallucination in LVLMs. It converts object-existence verification into binary questions and measures whether a model generates responses that are faithfully grounded in the visual input.

MME (Fu et al., 2025) The MME benchmark is a comprehensive evaluation of both perception and cognition capabilities of multimodal large language models. It consists of 14 subtasks covering abilities such as object existence, counting, position, color, OCR, commonsense reasoning, numerical calculation, and text-related reasoning.

MMBench (Liu et al., 2024b) The MMBench benchmark is a comprehensive multiple-choice benchmark that evaluates multimodal models across a diverse set of fine-grained abilities, including visual perception and reasoning. It adopts a hierarchical ability taxonomy and the CircularEval protocol to provide a more robust assessment of multimodal understanding. English and Chinese versions of MMBench are denoted as MMB and MMB^{CN}, respectively.

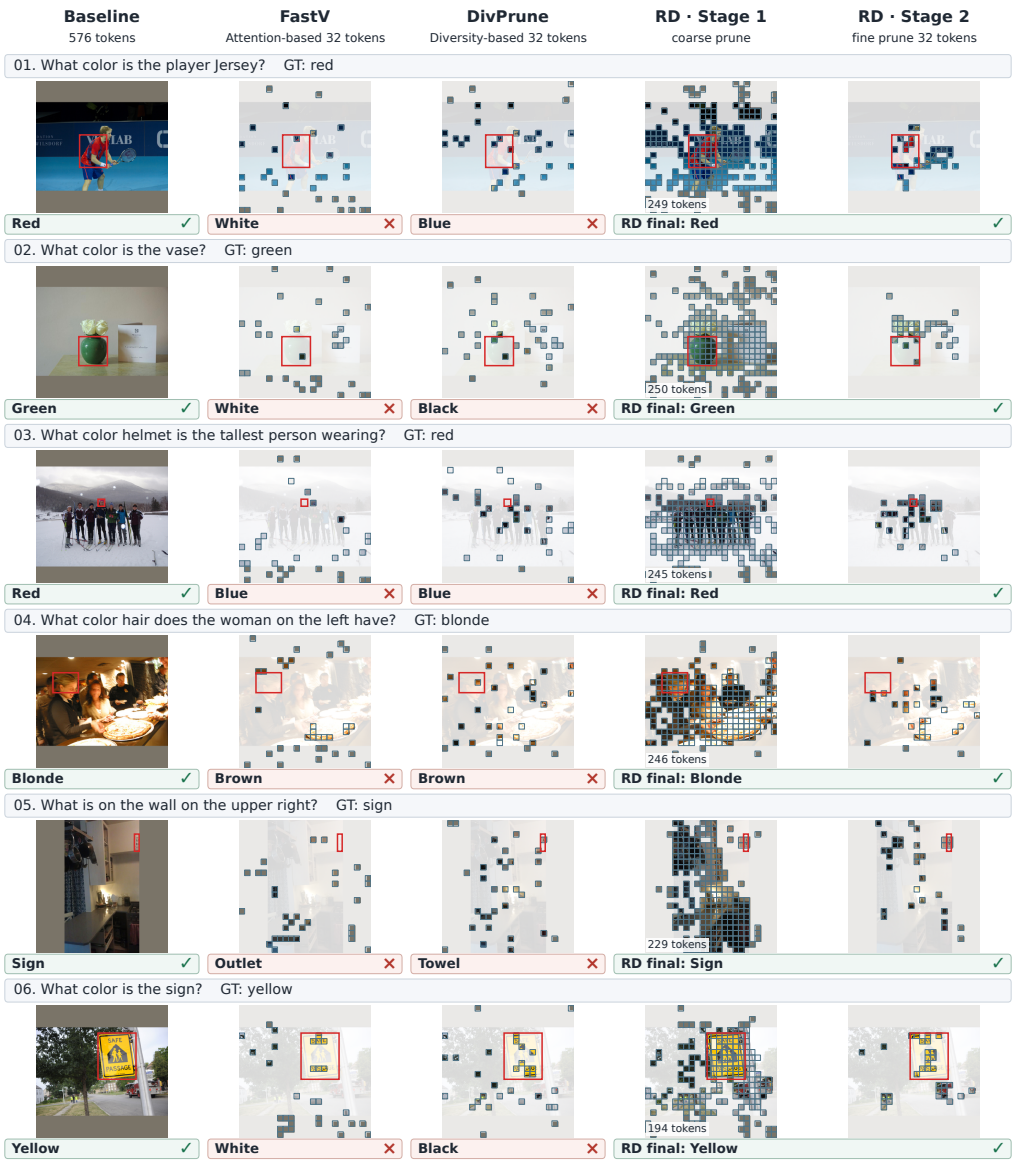

Figure 5: Visualization of the pruning results. Red bounding box represents the key region of the task.

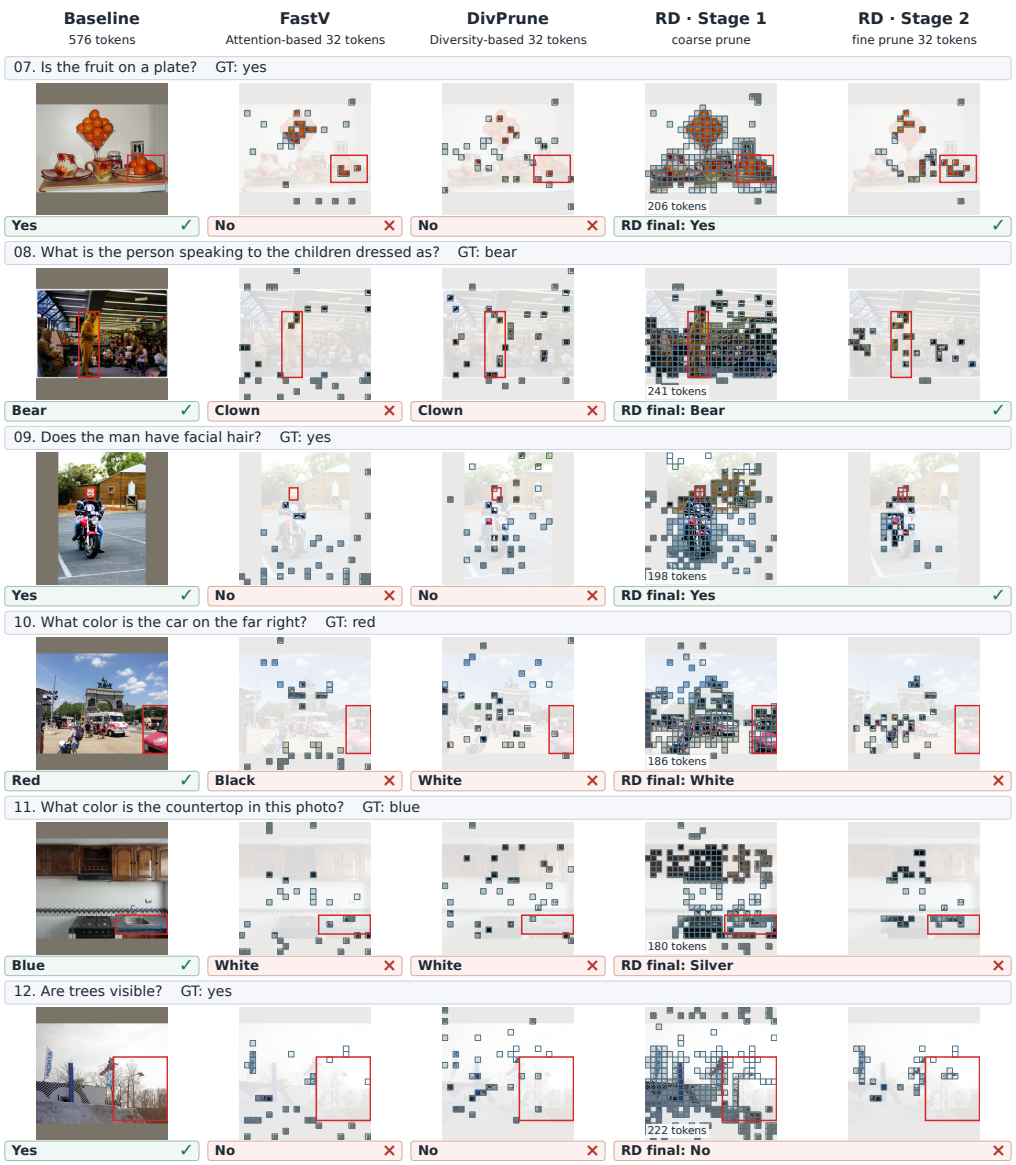

Figure 6: Visualization of the pruning results. Red bounding box represents the key region of the task.