TOP-RL: Task-Optimized Progressive Token Pruning with Reinforcement Learning for Large Vision Language Models

Hengyi Wang[1], Weiying Xie*[1], Hui Jiang[1], Yaotao Wei[2], Kai Jiang[1], Mingxiang Cao[1], Chenhe Hao[1], Leyuan Fang[3]

[1]State Key Laboratory of Integrated Services Networks, Xidian University,
[2]Beijing Institute of Technology,
[3]College of Electric and Information Engineering, Hunan University,
{hyw, jhui, mingxiangcao, hch}@stu.xidian.edu.cn, wyxie@xidian.edu.cn,
ytw@bit.edu.cn, xdjiangkai@foxmail.com, fangleyuan@gmail.com,

Abstract

In recent years, Large Vision-Language Models (LVLMs) have significantly advanced multimodal tasks. However, their inference requires intensive processing of numerous visual tokens and incurs substantial computational overhead. Existing methods typically compress visual tokens either at the input stage or in early model layers, ignoring variations across tasks and depths. To address these limitations, we introduce **TOP-RL**, a **Task-Optimized Progressive** token pruning framework based on **Reinforcement Learning**. TOP-RL formulates visual token pruning as a multi-stage Markov Decision Process (MDP). It employs an agent trained with dense and fine-grained reward signals to progressively generate differentiable binary masks. This enables TOP-RL to adaptively select crucial visual tokens tailored to each task, effectively balancing accuracy and computational efficiency. Extensive experiments on leading multimodal datasets and advanced LVLMs validate that TOP-RL effectively learns task-optimized pruning policies, significantly boosting inference efficiency while preserving robust performance. For instance, LLaVA-NeXT equipped with TOP-RL achieves a **1.9**× speedup in inference time and a **9.3**× reduction in FLOPs, with **96%** performance preserved.

Introduction

Large Vision-Language Models (LVLMs) (Bai et al. 2025; Liu et al. 2023, 2024a; Chen et al. 2024b) have achieved remarkable success across various multimodal tasks, including image captioning, visual question answering (VQA), and visual grounding (Liu et al. 2024d). However, this success comes at a significant computational cost. On one hand, LVLMs inherit large parameter sizes from Large Language Models (LLMs). On the other hand, a single image often maps to hundreds or even thousands of visual tokens, significantly increasing inference overhead. Therefore, reducing computational costs during inference has become a critical challenge for practical LVLM deployment.

To tackle this challenge, recent approaches have introduced **token pruning** with promising results. Existing methods generally fall into two classes: (1) compressing all visual tokens once before feeding into the LLM (Lan et al. 2024; Yang et al. 2025), and (2) dropping a fixed subset of tokens at one shallow layer using preset thresholds (Chen et al. 2024a). These approaches ignore how visual redundancy grows across deeper layers. In response, several works (Zhang et al. 2024c; Xing et al. 2025) propose layer-wise progressive pruning that reduces tokens at multiple stages based on token importance. Nevertheless, these methods are largely guided by attention patterns and remain agnostic to downstream task objectives, limiting their adaptability and overall effectiveness.

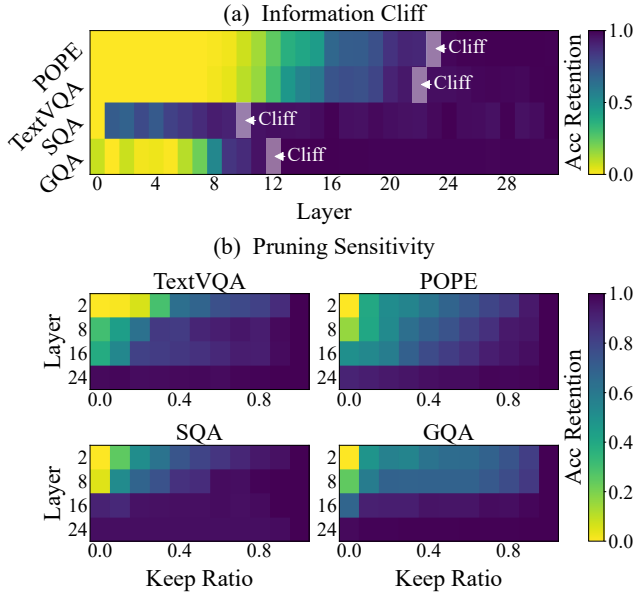

Figure 1: (a) Information Cliff. Accuracy retention when all visual tokens are pruned at each layer. The cliff appears at shallower layers for reasoning tasks and deeper layers for perception tasks. (b) Pruning Sensitivity. Accuracy retention for different token keep ratios at various layers, showing that pruning sensitivity depends on both layer and task. See Section Motivation for details.

Recent studies (Zhang et al. 2024b) reveal a phenomenon called the **Information Cliff**, the model layer beyond which removing visual tokens has little effect on performance. As

*Corresponding author.

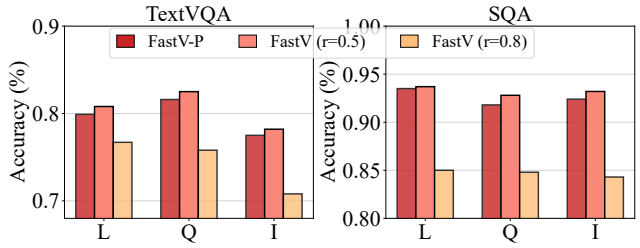

Figure 2: Compare progressive FastV (FastV-P) and vanilla FastV. For FastV, we applied pruning ratios $r = 0.5, 0.8$ at layer 8, whereas FastV-P employed pruning ratios $r = 0.5, 0.75, 0.875$ at layers 8, 16, and 24, respectively. L, Q and I mean LLaVA-OV, Qwen2.5-VL and InternVL3.

shown in Figure 1(a), this cliff appears much earlier for reasoning tasks (e.g., ScienceQA, GQA) than for perception tasks (e.g., TextVQA, POPE), indicating that visual information is retained and utilized in a highly task- and layer-dependent manner. Despite this, most existing methods either rely on explicit prompts or fail to capture such dynamic information flow. Our sensitivity analysis in Figure 1(b) further demonstrates that pruning sensitivity also varies considerably across tasks and layers. Our findings indicate that truly effective token pruning must be **coordinated with the hierarchical and task-dependent dynamics of information flow within the LVLM**, ensuring that token selection evolves in concert with the internal semantic processing.

To address this issue, we propose a novel visual token pruning framework named **T**ask-**O**ptimized **P**rogressive pruning via **R**einforcement **L**earning (**TOP-RL**). Unlike existing methods relying on static attention from either vision or text, we innovatively formulate visual token pruning as a dynamic multi-stage binary decision process modeled by a **Markov Decision Process** (MDP). Specifically, we introduce a reinforcement learning agent trained with dense and fine-grained reward signals. These signals combine task performance metrics (e.g., BLEU, Accuracy) with pruning ratios. This setup enables the agent to interact with the LVLM in real-time, incrementally optimizing pruning decisions at each layer based on the feedback of LVLM. At every pruning stage, the agent generates a differentiable binary mask, selectively retaining essential visual tokens while discarding redundant ones. This reinforcement-driven decision process spans all decoder layers, dynamically balancing the trade-off between maintaining task accuracy and maximizing compression. Through this adaptive and progressive pruning strategy, TOP-RL effectively captures task-specific variations across layers. Consequently, it tailors efficient, precise, and generalizable pruning policies for diverse vision-language tasks, substantially enhancing inference efficiency. Notably, TOP-RL is thus fully compatible with **FlashAttention.** (Dao et al. 2022). **Our contributions are summarized as follows:**

- Through systematic analysis, we reveal for the first time that the distribution of visual token importance across layers is both task-dependent and dynamically varying, indicating the necessity of a task-optimized and progressive pruning strategy.
- We propose TOP-RL, a novel visual token pruning framework. Specifically, TOP-RL employs reinforcement learning with dense reward signals derived from task performance and pruning ratios, enabling the agent to adaptively optimize pruning policies at each stage.
- Extensive experiments across multiple mainstream multimodal datasets and LVLMs demonstrate that TOP-RL achieves the best trade-off between inference efficiency and task performance, consistently surpassing prior methods in overall effectiveness.

Related Work

Visual Token Compression in LVLMs. Reducing visual tokens in LVLMs is a well-established method for accelerating inference. Existing approaches can be grouped into three categories. The first dynamically prunes visual tokens during LLM inference, often relying on visual-text attention to assess token importance, like FastV (Chen et al. 2024a). While effective, these methods typically perform one-shot pruning at shallow layers, neglecting redundancy in deeper layers, and may be incompatible with efficient attention mechanisms like FlashAttention (Dao et al. 2022). The second category compresses or merges tokens in a preprocessing stage before the LLM. Methods such as LLaVA-Mini (Zhang et al. 2025), and FasterVLM (Zhang et al. 2024a) project or select visual features early, but lack dynamic adaptation and task guidance. A third line of work adopts progressive and layer-wise pruning, such as Pyramid-Drop (Xing et al. 2025), ATP-LLaVA (Ye et al. 2025), and SparseVLM (Zhang et al. 2024c), which adaptively reduce tokens across LLM layers based on content or instance difficulty. However, these methods optimize pruning for general information redundancy, not for downstream task objectives.

In contrast, our TOP-RL framework incorporates both task awareness and progressive token pruning, dynamically modeling visual information flow and optimizing the efficiency-accuracy trade-off for specific tasks.

Reinforcement Learning for Efficient Inference. RL has proven effective in model compression, with notable applications like AMC (He et al. 2018), which used RL to explore architectural design spaces for better compression. BlockDrop (Wu et al. 2018) used RL for adaptive structural compression in ResNet. Recent works like MCMC (Li et al. 2024b) and Markov-PQ (Li et al. 2024c) employed RL to discover optimal pruning and quantization strategies under constraints. Building on these approaches, our TOP-RL formulates visual token pruning as an MDP, dynamically optimizing pruning across layers with task-aware reward signals, effectively modeling visual information flow.

Proposed Methods

Motivation

We briefly introduced our insights in the introduction section, and now we present more detailed observations. First, we revisit the theory of Information Flow and Information

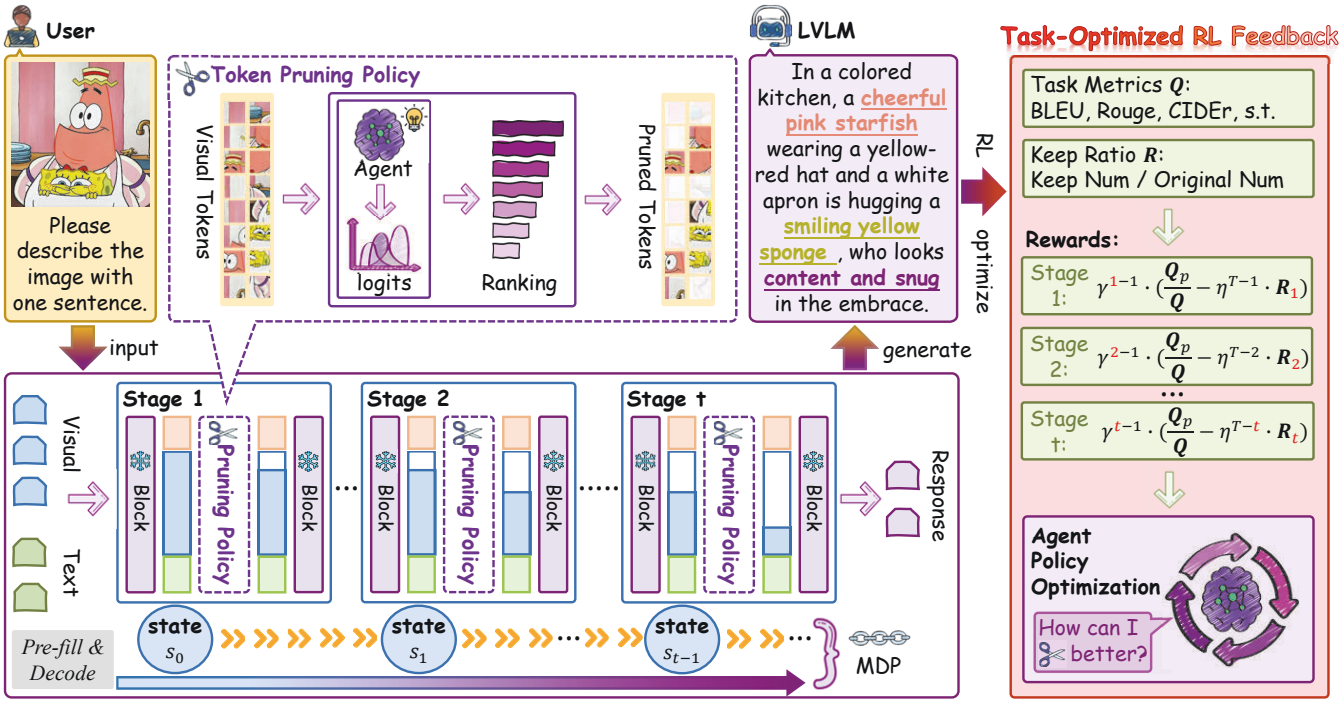

Figure 3: The architecture of TOP-RL. We begin by feeding images and prompts to obtain the state $\mathbf{S}$, action $\mathbf{A}$, and reward $\mathbf{R}$. During the Pre-fill stage, LVLM performs multiple stages of progressive pruning. After collecting sufficient samples, we train the agent (**Actor** and **Critic**) with reinforcement learning. TOP-RL can thus dynamically track the attention to visual information across different layers and adaptively learn optimal pruning strategies for each layer, conditioned on the specific task and input. This enables TOP-RL to achieve task-specific and fine-grained pruning, substantially improving inference efficiency and generalization while maintaining high performance.

Cliff. The method (Zhang et al. 2024b) defines Information Flow as the degree of attention that visual tokens receive when generating responses. Information Cliff refers to a specific layer beyond which the LVLM no longer attends to visual tokens, meaning these tokens cease to contribute to response generation; we thus designate this particular layer as the Information Cliff. Specifically, we conducted 32 experiments, each time completely removing all visual tokens at a particular fixed layer to identify the position of the Information Cliff. As illustrated in Figure 1(a), it is evident that for simpler tasks like POPE and TextVQA, the cliff emerges relatively late. This occurs because these tasks primarily require the LVLM to perform perceptual-level reasoning, thus demanding fewer layers for deeper semantic integration. In contrast, ScienceQA and GQA exhibit early-occurring Information Cliffs due to their complexity, as the LVLM needs extensive reasoning and deeper semantic processing after initially extracting visual information, thereby ceasing attention to visual tokens at shallower layers.

Insight 1
The importance of visual information flows **differently** across layers, **depending on the specific task**.

While the observation of Information Cliff is insightful, the experiment described above is relatively coarse. Therefore, we further conducted a detailed experiment examining the sensitivity of visual tokens during LVLM inference, as shown in Figure 1(b). Specifically, we experimented at layers 2, 8, 16, and 24 using 11 pruning ratios ranging from 0 to 1, following the pruning method adopted by FastV. Each value in the heatmap represents the retained accuracy ratio after pruning at a certain layer compared to the original accuracy on the dataset. As indicated by the heatmap, POPE and TextVQA exhibit notable pruning sensitivity until deeper layers, whereas ScienceQA and GQA demonstrate low pruning sensitivity even at shallow layers.

Insight 2
Visual token pruning should **adapt to the contribution** of visual information toward generating responses.

To further verify this idea succinctly, we modified the FastV method by applying multiple-stage pruning during inference, examining whether this aligns with the actual behavior of LVLMs. As shown in Figure 2, at a pruning ratio of $r = 0.5$, FastV-P significantly reduces computational cost compared to vanilla FastV, while incurring minimal accuracy loss; when the pruning ratio increases to $r = 0.8$, despite similar computational costs, FastV-P maintains a clear

advantage. These results demonstrate the effectiveness of progressive pruning in achieving a better trade-off between accuracy and efficiency compared to vanilla FastV. In summary, our findings indicate that optimal visual token pruning in LVLMs necessitates strategies that are both **progressive across layers** and **adaptive to task-specific requirements**.

Problem Formulation

We consider an LVLM that takes an image X^v and text prompt X^t as input. The image is encoded by a vision encoder E into visual features $Z^v \in \mathbb{R}^{L\times d^v}$, which are projected via an adapter p to match the LLM input dimension, yielding embeddings $H^v \in \mathbb{R}^{L\times d^t}$. These are concatenated with text embeddings $H^t = \text{embed}(X^t)$ and fed into the LLM to produce an output: $y = \mathbf{LLM}([H^v; H^t])$.

Our goal is to prune the visual tokens $H^v = [h_1^v, ..., h_L^v]$ to reduce sequence length and accelerate inference. Specifically, we learn a binary mask $m \in \{0,1\}^L$, where each entry indicates whether a token is kept (1) or dropped (0). The challenge is to select tokens that preserve task performance while improving efficiency.

Task-Optimized RL Analysis

Therefore, we formulate the token pruning during LVLM pre-fill stage as a binary decision optimization problem. Specifically, we define the policy as a trainable π_θ with parameters θ, which is a lightweight MLP. Given a sequence of visual tokens, the agent maps this input to a pruning action that determines which tokens to retain. In this way, the pruning procedure can be naturally modeled as a **Markov Decision Process** (MDP), where each step corresponds to a structured decision over the token sequence. MDP is defined as $\mathcal{M} = \langle \mathbf{S}, \mathbf{A}, \mathbf{P}, \mathbf{R}, \gamma \rangle$. Here, the episode length T is fixed and equals the total number of pruning steps. Concretely, let $H \in \mathbb{R}^{L\times d}$ be the input visual tokens of length L. We define two key quantities over the data distribution $\mathcal{D}$ at stage t:

$$L_t(\theta) = \mathbb{E}_{H\sim\mathcal{D}}\Big[\frac{L_t}{L}\Big], \qquad Q(\theta) = \mathbb{E}_{H\sim\mathcal{D}}\Big[\frac{Q_p}{Q}\Big], \tag{1}$$

where L_t is the number of current visual tokens retained, Q and Q_p denote the metric without and with pruning. Note that Q_p and L_p are computed only after all pruning steps are completed. We then train π_θ to maximize the combined objective at the pruning stage t:

$$\begin{aligned} J_t(\theta) &= Q(\theta) - L_t(\theta), \\ &= \mathbb{E}_{H\sim\mathcal{D}}\Big[\frac{Q_p}{Q} - \frac{L_t}{L}\Big]. \end{aligned} \tag{2}$$

In this manner, the policy π_θ adaptively strikes an optimal balance between computational efficiency and model performance tailored to each task objective.

State (S). At each stage $t \in \{1, \ldots, T\}$, the state $s_t = [\, t, H_t^v \,]$, where $H_t^v \in \mathbb{R}^{L_t\times d}$ denotes the visual token representations that aligned to the text token embedding space.

Action (A). The action is a binary mask $m_t \in \{0,1\}^{L_t}$, sampled from $\pi_\theta(\cdot \mid s_t)$.

State Transition (P). The next state s_{t+1} is deterministically given by:

$$s_{t+1} = p(s_t, m_t) = \mathbf{Transformer\ Layers}\big(m_t, s_t\big). \tag{3}$$

Reward (R). Our goal is to train a policy π_θ that **maximizes** the expected cumulative reward:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{t=1}^{T} \gamma^{t-1} r_t\right], \tag{4}$$

where r_t is the reward assigned at each pruning stage, and γ is the discount factor (with $\gamma = 0.9$ in our experiments). For each episode, we define the dense and step-wise reward as:

$$r_t = r^Q - \alpha_t r_t^L, \tag{5}$$

where $r^Q = \frac{Q_p}{Q}$ and $r_t^L = \frac{L_t}{L}$ are computed after all pruning steps, and are assigned to each step t.

To encourage early-stage pruning, we introduce a temporal scaling factor $\alpha_t = \eta^{T-t}$ for the pruning ratio reward, which increases the incentive for pruning at earlier steps. Typically, we use $\eta = 0.95$ in our experiments. Therefore, γ_t governs the smoothing of long-term returns, while α_t specifically modulates the retention ratio term.

Training. The pipeline is shown in Figure 3. For training, each image-question pair is fed into the LVLM to generate answers. The image is encoded by the vision encoder E into a sequence of visual tokens, which are projected into embeddings and concatenated with text tokens for LLM inference.

At each pruning stage t, the current visual tokens serve as the input state s_t of the agent. The $\mathbf{Actor}$ encodes s_t and, via policy $\pi_\theta(s_t)$, outputs logits for Gumbel-Softmax sampling, yielding a differentiable binary mask m_t. The pruning action a_t is then defined by masking the tokens, with pruned tokens passed through subsequent Transformer layers to yield the next state s_{t+1}. This process generates a sequence of $\langle \mathbf{S}, \mathbf{A}, \mathbf{R} \rangle$ tuples per inference.

Rewards are computed after generation: the task metric Q is evaluated, and together with the pruning ratio, a dense reward is assigned for each pruning action as in Equation (5).

We train the reinforcement learning agent using Proximal Policy Optimization (PPO) on collected trajectories. To address the incompatibility between GAE and variable-length token pruning, we adopt the attention masking strategy from (Rao et al. 2021), which preserves the number of tokens while neutralizing pruned tokens impact in self-attention. Specifically, for binary mask $\hat{m}$ and total token count N, the attention mask $\mathbf{G}$ is:

$$G_{ij} = \begin{cases} 1, & i = j \\ \hat{m}_j, & i \neq j \end{cases} \tag{6}$$

and the masked attention is computed as

$$\tilde{A}_{ij} = \frac{\exp(P_{ij}) G_{ij}}{\sum_{k=1}^{N} \exp(P_{ik}) G_{ik}}, \tag{7}$$

with $P_{ij} = (QK^T)_{ij}/\sqrt{C}$. This ensures pruned tokens do not influence attention, while the attention map remains $N \times N$ for parallel computation and stable GAE.

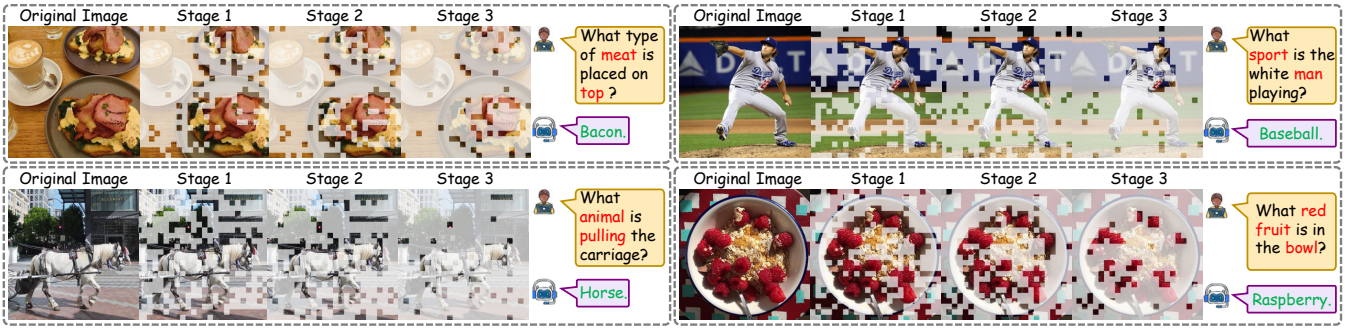

Figure 4: Visualization of TOP-RL on different VQA prompts. From left to right: the original input image, the visual tokens after the first pruning stage, the visual tokens after the second pruning stage, and the visual tokens after the third pruning stage.

Computing Cost Estimation

Following the complexity estimation in (Chen et al. 2024a; Xing et al. 2025), we consider the computational cost associated with the Multi-Head Attention (MHA) and Feed-Forward Network (FFN) modules within Transformer layers. Specifically, for a single Transformer layer with an input sequence consisting of n visual tokens, hidden dimension d, and FFN intermediate dimension m, the FLOPs related to visual tokens can be approximated as

$$4nd^2 + 2n^2d + 3ndm. \tag{8}$$

In our proposed multi-stage pruning framework, the Transformer comprises L_T layers in total. We perform pruning at layers $k_1, k_2, \ldots, k_t$, with pruning ratios denoted by $r_1, r_2, \ldots, r_t$. The t denotes the total prune stage number. Then the total FLOPs related to visual token across the entire model is formulated as:

$$\sum_{i=1}^{t} L_i\big(4n_i d^2 + 2n_i^2 d + 3n_i dm\big), \tag{9}$$

$$\text{s.t. } n_i = r_i n, \quad L_i = k_i - k_{i-1}, \quad \sum_{i=1}^{t} L_i = L_T. \tag{10}$$

Furthermore, we introduce a lightweight **Actor** model for dynamically selecting visual tokens. For visual tokens input to the **Actor** at pruning stage t, the FLOPs is given by:

$$2(d_1 d_2 + 2d_2^2)\, n_t, \tag{11}$$

where d_1 and d_2 is input and hidden feature dimension. Thus, the total FLOPs of TOP-RL in LLaVA-NeXT for the visual token computations across the model is formulated as:

$$\text{FLOPs} = \sum_{i=1}^{t} \Big[L_i\big(4n_i d^2 + 2n_i^2 d + 3n_i dm\big) + 2(d_1 d_2 + 2d_2^2)\, n_t \Big]. \tag{12}$$

Experiments

Experimental Settings

Models. To demonstrate the versatility of TOP-RL, we conduct extensive experiments on multiple LVLMs. Our evaluation spans various architectures, including the LLaVA family: LLaVA-1.5 (Liu et al. 2023), LLaVA-NeXT (Liu et al. 2024b), Video-LLaVA (Lin et al. 2023) and LLaVA-OV (Li et al. 2024a), alongside the current advanced models such as Qwen2.5-VL (Bai et al. 2025) and InternVL-3 (Zhu et al. 2025).

Benchmarks. To comprehensively evaluate the effectiveness and generalizability of TOP-RL across tasks with varying complexity, we follow the multi-level taxonomy introduced by (Yao et al. 2025; Liu et al. 2024c). Specifically, we categorize mainstream benchmarks into two levels according to task difficulty: perception and reasoning. For each level, we select representative benchmark for systematic evaluation. The perception level includes POPE, VizWiz, Flickr30k, RefCOCO and TextVQA; and the reasoning level encompasses GQA, AI2D, MathVista and ScienceQA (SQA). Refer to Appendix A for more details.

Baselines. For fair comprehensive evaluation, we compare TOP-RL with general token pruning methods FastV (Chen et al. 2024a), VisionZip (Yang et al. 2025), and DivPrune (Alvar et al. 2025)) and progressive methods (PyramidDrop (Xing et al. 2025) and SparseVLM (Zhang et al. 2024c)).

RL Settings. Owing to space constraints, RL training details and the complete architectures of **Actor** and **Critic** models are provided in the Appendix A. All results are averaged over three runs with different random seeds.

Implementation Details. Following the common practice in (Rao et al. 2021), we utilize Gumbel-Softmax sampling during training for differentiable mask learning, while adopting deterministic top-k token selection at inference. Furthermore, we anneal the temperature parameter τ of Gumbel-Softmax from 1.0 to 0.05 over the course of training, encouraging early-stage exploration and late-stage exploitation. Refer to Appendix A for more details.

Result Analysis

Main Result. We compare TOP-RL with leading token pruning baselines under varying compression ratios. As shown in Table 1, when retaining 192 tokens (66.7% reduction), TOP-RL achieves the highest overall accuracy

Methods	Perception					Reasoning				Avg
	POPE	VizWiz	Flickr30k	RefCOCO	TextVQA	GQA	AI2D	MathVista	SQA	
Upper Bound, 576 Tokens (100%)										
LLaVA-1.5-7B	85.9	59.1	67.9	60.8	58.4	61.8	54.3	24.6	69.8	100%
Retain Average 192 Tokens (66.7%↓)										
FastV	80.6 93.9%	54.2 91.7%	63.6 93.7%	38.7 63.6%	52.7 90.2%	52.6 85.1%	50.4 92.8%	16.8 68.3%	66.2 94.8%	87.7%
SparseVLM	84.4 98.3%	54.8 92.6%	52.8 77.8%	56.2 92.4%	57.8 99.0%	57.2 92.6%	50.8 93.6%	19.4 78.9%	65.3 93.6%	91.3%
PDrop	82.3 95.8%	55.7 94.2%	62.5 92.0%	46.8 77.0%	55.6 95.2%	55.7 90.1%	51.6 95.0%	17.6 71.5%	69.2 99.1%	91.2%
VisionZip	85.1 99.1%	56.3 95.2%	63.8 94.0%	53.4 87.8%	57.8 99.0%	59.4 96.1%	51.7 95.1%	22.3 90.7%	58.9 84.4%	94.3%
DivPrune	84.8 98.7%	56.1 94.9%	55.2 81.3%	**56.3** **92.6%**	56.3 96.4%	58.7 95.1%	51.4 94.6%	21.6 87.8%	68.6 98.3%	93.9%
TOP-RL	**85.6** **99.7%**	**57.9** **98.0%**	**67.1** **98.8%**	56.1 92.2%	**58.2** **99.7%**	**60.9** **98.5%**	**52.3** **96.1%**	**23.0** **93.5%**	**69.6** **99.7%**	**97.6%**
Retain Average 64 Tokens (88.9%↓)										
FastV	52.5 61.1%	50.3 85.0%	12.6 18.6%	29.8 49.0%	47.8 81.9%	38.6 62.5%	43.6 80.3%	9.8 39.8%	67.7 97.1%	66.5%
SparseVLM	75.5 87.9%	52.3 88.5%	32.7 48.2%	50.7 83.4%	53.4 91.4%	53.8 87.1%	42.1 77.5%	12.2 49.6%	65.2 93.4%	78.0%
PDrop	55.9 65.1%	51.8 87.7%	50.8 74.8%	37.4 61.5%	55.9 95.7%	41.9 67.8%	43.9 80.8%	11.4 46.3%	68.0 97.4%	76.0%
VisionZip	78.6 91.5%	55.4 93.8%	60.3 88.8%	50.1 82.4%	56.5 96.8%	57.0 92.2%	47.8 88.1%	19.5 79.3%	58.9 84.4%	87.9%
DivPrune	80.2 93.4%	53.7 90.9%	53.6 78.9%	53.1 87.2%	57.2 98.0%	55.3 89.5%	46.5 85.7%	18.9 76.8%	67.8 97.1%	87.1%
TOP-RL	**81.9** **95.4%**	**56.4** **95.4%**	**62.5** **92.0%**	**53.3** **87.7%**	**57.8** **99.0%**	**57.2** **92.6%**	**48.0** **88.4%**	**20.1** **81.7%**	**69.2** **99.1%**	**91.0%**

Table 1: Comparison of TOP-RL and baselines at different visual tokens. Tasks are grouped into Perception and Reasoning. Each entry reports both the main metric and its retention ratio (i.e., the percentage relative to the original unpruned result). Across all settings, TOP-RL achieves the best performance and strong robustness, especially under high compression. TOP-RL is applied as an inference-only method.

(97.6%) across perception, understanding, and reasoning tasks, with clear gains on challenging reasoning datasets (e.g., MathVista, ScienceQA). Even under extreme compression (64 tokens, 88.9% reduction), TOP-RL maintains robust performance, outperforming FastV and SparseVLM by 24.5% and 13.0%, respectively. These results highlight the strong efficiency-accuracy trade-off of TOP-RL.

On video question answering (Table 2), TOP-RL compresses 2048 tokens down to 194 while still achieving the highest average accuracy (93.9%). It consistently outperforms baselines and excels on the most challenging datasets, demonstrating its ability to preserve critical information in both image and video understanding.

Computational Efficiency. We conduct a comprehensive evaluation of inference efficiency and computational cost on LLaVA-NeXT-7B, as shown in Table 3. All methods are benchmarked under identical conditions with the average number of visual tokens fixed at 320. Compared to the unpruned baseline, TOP-RL achieves a substantial reduction in both total inference time (**1.91**× speedup) and theoretical FLOPs (**9.32**× reduction), while maintaining competitive task accuracy. Although some methods such as VisionZip achieve marginally higher FLOPs reduction, TOP-RL delivers the **best overall trade-off** between efficiency and accuracy. Specifically, our method achieves an F1 score of 87.2, outperforming all other baselines at the same compression level.

Qualitative Visualization. Figure 4 illustrates the token pruning decisions made by TOP-RL on various VQA tasks. We show the pruning visualization at successive stages of the process. As the stages advance, the proportion of pruned tokens increases. Even in the final stage, when the vast ma-

Method	MSVD	MSRVTT	ActivityNet	Avg
Video-LLaVA	70.5 100.0%	51.3 100.0%	43.2 100.0%	100.0%
FastV	47.2 67.0%	42.4 82.6%	38.5 89.1%	71.7%
SparseVLM	65.7 93.2%	45.4 88.5%	41.5 96.1%	90.7%
VisionZip	64.3 91.2%	46.8 91.2%	42.2 97.7%	93.4%
TOP-RL	**66.4** **94.2%**	**47.1** **91.8%**	**42.7** **98.8%**	**93.9%**

Table 2: Results of Video-LLaVA with TOP-RL on video question answering benchmarks. The original number of video tokens is 2048, while our experiment collectively prunes it down to an average of 194 tokens.

Methods	Total Time↓	FLOPs (T)↓	GPU (GB)↓	Score (F1)↑
LLaVA-NeXT-7B	28:21 (1.00×)	24.6 (1.00×)	16.9	91.6
FastV	17:02 (1.66×)	3.89 (6.32×)	15.6	81.9
SparseVLM	18:32 (1.53×)	4.02 (6.12×)	18.6	84.7
PDrop	14:50 (1.90×)	2.64 (9.32×)	15.6	85.1
VisionZip	**13:27** (**2.11**×)	**1.88** (**13.09**×)	23.8	86.4
TOP-RL	14:48 (1.91×)	2.64 (9.32×)	**15.6**	**87.2**

Table 3: Efficiency analysis of different pruning methods on LLaVA-NeXT-7B. TOP-RL achieves a competitive balance between accuracy and efficiency.

jority of visual tokens have been removed, the tokens relevant to the question remain intact. This behavior demonstrates that TOP-RL not just adapts effectively to different tasks but also reliably captures essential visual information throughout the layer-wise pruning process.

Ablation Studies

Cross-Task Transfer. We report cross-task transfer results in Table 4, where each TOP-RL pruning agent is trained on a single task and evaluated zero-shot on all tasks. As expected, the highest accuracy consistently occurs when the agent is evaluated on its training task (diagonal entries), confirming the effectiveness of task-specific policies. Within the same cognitive domain, cross-task transfer leads to only minor accuracy drops, demonstrating strong generalization. In contrast, transferring across cognitive levels results in larger performance degradation. It still remains challenging.

Source ↓	TextVQA	POPE	SQA	GQA
TextVQA	**58.2** **100.0%**	82.7 96.6%	46.4 66.8%	38.1 62.5%
POPE	58.9 101.2%	**85.6** **100.0%**	58.6 84.5%	60.2 98.8%
SQA	55.3 95.0%	67.2 78.5%	**69.6** **100.0%**	61.5 100.8%
GQA	51.9 89.2%	72.3 84.5%	68.1 98.1%	**60.9** **100.0%**

Table 4: Compare the zero-shot accuracy of pruning agents trained on individual tasks and evaluated across different task level on LLaVA-1.5-7B.

Extended Experiments and Ablation

Due to space limitations, we defer all extended ablation studies to the Appendix. Specifically, Appendix B summarizes further generalization analysis on the MME benchmark, including the evaluation of perception- and reasoning-oriented agents individually and under mixed-question scenarios, which highlights the task-specific adaptiveness of our method. Comprehensive results on advanced LVLMs (e.g., Qwen2.5-VL) are also included in Appendix B.

Moreover, Appendix C presents extensive ablation studies covering: (1) sensitivity to the trade-off parameters γ and η; (2) the contribution of each reward component in Equation (5); (3) the influence of pruning step count T; (4) the impact of dense versus terminal-only reward assignment; and (5) the comparison between progressive multi-stage pruning and one-shot pruning under the same token budget, which further validates the effectiveness of the proposed reinforcement-learning-based progressive pruning strategy in balancing efficiency and performance. These results demonstrate the robustness and generality of TOP-RL.

Conclusion

We introduce TOP-RL, a task-optimized reinforcement learning framework for progressive visual token pruning in LVLMs. Unlike static or heuristic approaches, TOP-RL enables adaptive, layer-wise token selection by leveraging dense reward feedback. Our analyses confirm that token importance varies across layers and tasks, motivating a dynamic pruning strategy. Experiments on diverse benchmarks and models demonstrate that TOP-RL consistently achieves a strong balance between efficiency and accuracy, making it a practical solution for efficient LVLM deployment.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 62322117, 62371365, U24B20136, and U22B2014; in part by the Fundamental and Interdisciplinary Disciplines Breakthrough Plan of the Ministry of Education of China JYB2025XDXM105; in part by the Fundamental Research Funds for the Central Universities; and in part by the Innovation Fund of Xidian University.

References

Alvar, S. R.; Singh, G.; Akbari, M.; and Zhang, Y. 2025. DivPrune: Diversity-based Visual Token Pruning for Large Multimodal Models. In *Computer Vision and Pattern Recognition*, 9392–9401.

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

Chen, L.; Zhao, H.; Liu, T.; Bai, S.; Lin, J.; Zhou, C.; and Chang, B. 2024a. 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*, 19–35.

Chen, Z.; Wang, W.; Cao, Y.; Liu, Y.; Gao, Z.; Cui, E.; Zhu, J.; Ye, S.; Tian, H.; Liu, Z.; et al. 2024b. Expanding Performance Boundaries of Open-Source Multimodal Models with Model, Data, and Test-Time Scaling. *arXiv preprint arXiv:2412.05271*.

Dao, T.; Fu, D.; Ermon, S.; Rudra, A.; and Ré, C. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. In *Neural Information Processing Systems*, 16344–16359.

He, Y.; Lin, J.; Liu, Z.; Wang, H.; Li, L.-J.; and Han, S. 2018. Amc: Automl for model compression and acceleration on mobile devices. In *European Conference on Computer Vision*, 784–800.

Lan, Z.; Niu, L.; Meng, F.; Li, W.; Zhou, J.; and Su, J. 2024. AVG-LLaVA: A Large Multimodal Model with Adaptive Visual Granularity. *arXiv preprint arXiv:2410.02745*.

Li, B.; Zhang, Y.; Guo, D.; Zhang, R.; Li, F.; Zhang, H.; Zhang, K.; Zhang, P.; Li, Y.; Liu, Z.; et al. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Li, S.; Chen, J.; Liu, S.; Zhu, C.; Tian, G.; and Liu, Y. 2024b. MCMC: Multi-constrained model compression via one-stage envelope reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*.

Li, Y.; Zhang, X.; Xie, W.; Zhang, J.; Fang, L.; and Du, J. 2024c. Markov-PQ: Joint Pruning-Quantization via Learnable Markov Chain. *IEEE Transactions on Circuits and Systems for Video Technology*.

Lin, B.; Zhu, B.; Ye, Y.; Ning, M.; Jin, P.; and Yuan, L. 2023. Video-LLaVA: Learning United Visual Representation by Alignment Before Projection. *arXiv preprint arXiv:2311.10122*.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024a. Improved baselines with visual instruction tuning. In *Computer Vision and Pattern Recognition*, 26296–26306.

Liu, H.; Li, C.; Li, Y.; Li, B.; Zhang, Y.; Shen, S.; and Lee, Y. J. 2024b. LLaVA-NeXT: Improved reasoning, OCR, and world knowledge.

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2023. Visual Instruction Tuning. In *Neural Information Processing Systems*.

Liu, Y.; Duan, H.; Zhang, Y.; Li, B.; Zhang, S.; Zhao, W.; Yuan, Y.; Wang, J.; He, C.; Liu, Z.; Chen, K.; and Lin, D. 2024c. MMBench: Is Your Multi-modal Model an All-around Player? *arXiv preprint arXiv:2307.06281*.

Liu, Y.; Duan, H.; Zhang, Y.; Li, B.; Zhang, S.; Zhao, W.; Yuan, Y.; Wang, J.; He, C.; Liu, Z.; et al. 2024d. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, 216–233.

Rao, Y.; Zhao, W.; Liu, B.; Lu, J.; Zhou, J.; and Hsieh, C.-J. 2021. DynamicViT: Efficient Vision Transformers with Dynamic Token Sparsification. In *Neural Information Processing Systems*.

Wu, Z.; Nagarajan, T.; Kumar, A.; Rennie, S.; Davis, L. S.; Grauman, K.; and Feris, R. 2018. BlockDrop: Dynamic Inference Paths in Residual Networks. In *Computer Vision and Pattern Recognition*.

Xing, L.; Huang, Q.; Dong, X.; Lu, J.; Zhang, P.; Zang, Y.; Cao, Y.; He, C.; Wang, J.; Wu, F.; and Lin, D. 2025. PyramidDrop: Accelerating Your Large Vision-Language Models via Pyramid Visual Redundancy Reduction. arXiv:2410.17247.

Yang, S.; Chen, Y.; Tian, Z.; Wang, C.; Li, J.; Yu, B.; and Jia, J. 2025. Visionzip: Longer is better but not necessary in vision language models. In *Computer Vision and Pattern Recognition*, 19792–19802.

Yao, R.; Zhang, B.; Huang, J.; Long, X.; Zhang, Y.; Zou, T.; Wu, Y.; Su, S.; Xu, Y.; Zeng, W.; Yang, Z.; Li, G.; Zhang, S.; Li, Z.; Chen, Y.; Xiong, S.; Xu, P.; Zhang, J.; Zhou, B.; Clifton, D.; and Gool, L. V. 2025. LENS: Multi-level Evaluation of Multimodal Reasoning with Large Language Models. *arXiv preprint arXiv:2505.15616*.

Ye, X.; Gan, Y.; Ge, Y.; Zhang, X.-P.; and Tang, Y. 2025. ATP-LLaVA: Adaptive Token Pruning for Large Vision Language Models. In *Computer Vision and Pattern Recognition*, 24972–24982.

Zhang, Q.; Cheng, A.; Lu, M.; Zhuo, Z.; Wang, M.; Cao, J.; Guo, S.; She, Q.; and Zhang, S. 2024a. [CLS] Attention is All You Need for Training-Free Visual Token Pruning: Make VLM Inference Faster. *arXiv preprint arXiv:2412.01818*.

Zhang, S.; Fang, Q.; Yang, Z.; and Feng, Y. 2025. LLaVA-Mini: Efficient Image and Video Large Multimodal Models with One Vision Token. In *International Conference on Learning Representations*.

Zhang, X.; Quan, Y.; Shen, C.; Yuan, X.; Yan, S.; Xie, L.; Wang, W.; Gu, C.; Tang, H.; and Ye, J. 2024b. From redundancy to relevance: Information flow in lvlms across reasoning tasks. *arXiv preprint arXiv:2406.06579*.

Zhang, Y.; Fan, C.-K.; Ma, J.; Zheng, W.; Huang, T.; Cheng, K.; Gudovskiy, D.; Okuno, T.; Nakata, Y.; Keutzer, K.; et al. 2024c. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*.

Zhu, J.; Wang, W.; Chen, Z.; Liu, Z.; Ye, S.; Gu, L.; Tian, H.; Duan, Y.; Su, W.; Shao, J.; Gao, Z.; Cui, E.; Wang, X.; Cao, Y.; Liu, Y.; Wei, X.; Zhang, H.; Wang, H.; Xu, W.; Li, H.; Wang, J.; Deng, N.; Li, S.; He, Y.; Jiang, T.; Luo, J.; Wang, Y.; He, C.; Shi, B.; Zhang, X.; Shao, W.; He, J.; Xiong, Y.; Qu, W.; Sun, P.; Jiao, P.; Lv, H.; Wu, L.; Zhang, K.; Deng,

H.; Ge, J.; Chen, K.; Wang, L.; Dou, M.; Lu, L.; Zhu, X.; Lu, T.; Lin, D.; Qiao, Y.; Dai, J.; and Wang, W. 2025. InternVL3: Exploring Advanced Training and Test-Time Recipes for Open-Source Multimodal Models. arXiv:2504.10479.