CoViPAL: Layer-wise Contextualized Visual Token Pruning for Large Vision-Language Models

Anonymous ACL submission

Abstract

Large Vision-Language Models (LVLMs) process multimodal inputs consisting of text tokens and vision tokens extracted from images or videos. Due to the rich visual information, a single image can generate thousands of vision tokens, leading to high computational costs during the prefilling stage and significant memory overhead during decoding. Existing methods attempt to prune redundant vision tokens, revealing substantial redundancy in visual representations. However, these methods often struggle in shallow layers due to the lack of sufficient contextual information. We argue that many visual tokens are inherently redundant even in shallow layers and can be safely and effectively pruned with appropriate contextual signals. In this work, we propose CoViPAL, a layer-wise contextualized visual token pruning method that employs a Plug-and-Play Pruning Module (PPM) to predict and remove redundant vision tokens before they are processed by the LVLM. The PPM is lightweight, model-agnostic, and operates independently of the LVLM architecture, ensuring seamless integration with various models. Extensive experiments on multiple benchmarks demonstrate that CoViPAL outperforms training-free pruning methods under equal token budgets and surpasses training-based methods with comparable supervision. CoViPAL offers a scalable and efficient solution to improve inference efficiency in LVLMs without compromising accuracy.

1 Introduction

Large Vision-Language Models (LVLMs, Chiang et al., 2023; Anil et al., 2023; Bai et al., 2023; Liu et al., 2023) have recently demonstrated remarkable capabilities in understanding and generating content grounded in visual inputs, including both images and videos. To effectively capture the rich spatial and semantic details inherent in visual signals, these models often rely on generating hundreds or even thousands of visual tokens per image or video. For instance, LLaVA-OneVision (Li et al., 2024a) explicitly allocates up to 7,290 visual tokens per image, leveraging a large corpus of high-quality images to maximize visual comprehension.

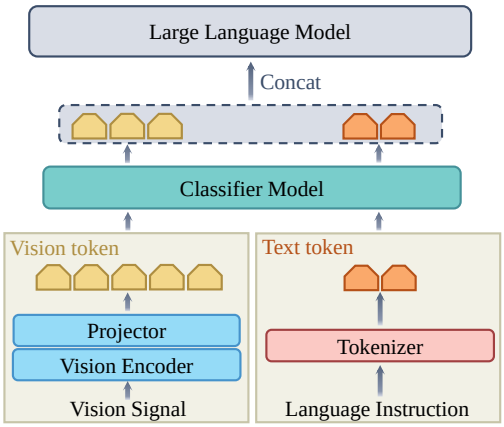
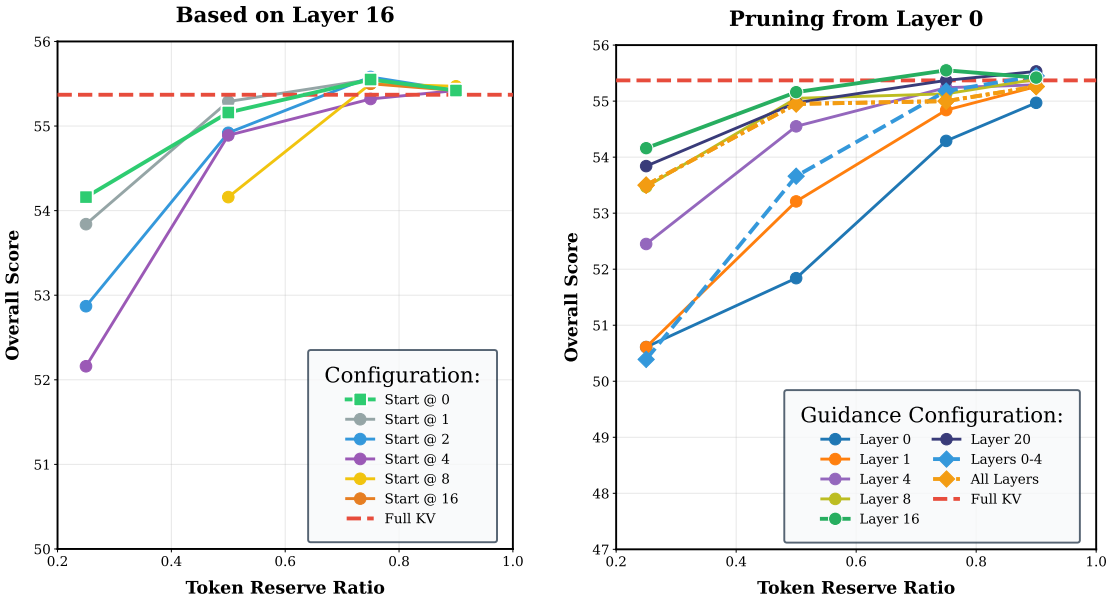


Figure 1: Illustration for CoViPAL at inference stage.

Although dense visual token representations enhance the model's capacity to understand fine-grained visual content, they come at the cost of substantial computational and memory overhead (Zhang et al., 2025). This leads to reduced inference efficiency and makes it difficult to apply LVLMs in scenarios where resources are limited or real-time performance is required.

To address this issue, prior work has explored reducing the number of visual tokens or compressing their corresponding key-value (KV) cache (Bolya et al., 2022; Shang et al., 2024), highlighting the substantial redundancy present in visual representations. Token eviction methods discard less informative tokens based on importance scores (Chen et al., 2024b; Lin et al., 2025), while token merging approaches group similar tokens and consolidate them to reduce token number (Chen et al., 2024a; Shi et al., 2023). Empirical observations suggest that pruning visual tokens in shallow layers can sig-



(a) Pruning based on attention of layer 16.

(b) Pruning guided by different attention weight.

Figure 2: Prune tokens in different layers and based on different attention weights.

nificantly hurt performance and every visual token matters in these layers (Xing et al., 2024). Despite their effectiveness to some extent, these methods largely fail to prune tokens in the shallow layers.

Visual token reduction is less effective in shallow layers, primarily because tokens in these layers interact with fewer transformer decoder layers, resulting in limited contextual information. This makes it challenging to identify unimportant tokens, leading to significant performance degradation when attempting to prune visual tokens at these stages. However, we observe that some visual tokens are inherently redundant and can be effectively and safely pruned when guided by appropriate contextual information. Based on this insight, we propose CoViPAL, a contextualized visual token pruning method that operates across all layers. CoViPAL implements the PPM module using small classifier trained on limited data to identify and remove less important tokens before they are passed to the base model of LVLM, thereby reducing the number of visual tokens while maintaining model performance.

We conducted experiments on two models: LLaVA-OneVision and LLaVA-Video. For LLaVA-OneVision, we trained the classifier using only 0.46% of the pretraining dataset, while for LLaVA-Video, we extended its capabilities to handle video inputs using just 7.4% of the video instruction-following dataset. Additionally, we performed extensive experiments on a variety of image and video benchmarks. The results demonstrate that our method reduces the prefilling time by up to 60% compared to the original model, with only minimal performance degradation when pruning 75% of the visual tokens. Furthermore, our approach outperforms both training-free methods, FastV and SparseVLM, and the training-based method PyramidDrop, when maintaining the same percentage visual tokens.

2 Related Works

2.1 Token Pruning

Token pruning methods aim to remove tokens with low attention or feature similarity after early or intermediate layers (Chen et al., 2024b; Lin et al., 2025; Xing et al., 2024), or optimize pruning schedules using small inference batches to meet FLOPs budgets (Ye et al., 2025). These methods generally prioritize the preservation of early tokens to avoid information loss.

2.2 Token Merging

Alternatively, similarity-based merging techniques fuse redundant tokens either spatially or cross-modally to reduce token count while maintaining semantic integrity and accuracy (Chen et al., 2024a; Shi et al., 2023). These methods achieve compression without compromising downstream performance. They typically leave the tokens in shallow

layers unmerged to maintain overall performance.

2.3 Hybrid Methods

Recent methods combine pruning and merging by ranking tokens based on attention, pruning low-importance tokens, and merging redundant ones to recycle information (Zhong et al., 2024; Shang et al., 2024; Zhang et al., 2025). For instance, LOOK-M (Wan et al., 2024) addresses long-context inference by compressing the KV cache through text-guided merging of similar key-value pairs, thereby reducing memory usage and improving decoding speed.

These approaches generally retain visual tokens in shallow layers to minimize significant performance degradation. In contrast, our method demonstrates that visual token redundancy exists across all layers and can be safely pruned using a lightweight classifier trained on a small dataset. This approach facilitates earlier and more efficient pruning without sacrificing critical information.

3 Preliminary

3.1 Notations

In LVLMs, a vision encoder is typically employed to extract visual features, while a projector is used to map these features into the word embedding space. We denote the vision encoder and projector as $g(\cdot)$, so the visual tokens are represented as $\mathbf{H}_\mathsf{v} = g(\mathbf{X}_\mathsf{v})$, where $\mathbf{X}_\mathsf{v}$ is the visual input. The textual input is represented by the text tokens $\mathbf{H}_\mathsf{t}$, which are concatenated with the visual tokens, forming the input to the LLM as $f(\cdot)$.

For token pruning, we assign an importance score $\mathbb{S}$ to each visual token. This score serves as the guiding criterion for the pruning process, directly determining the relevance of each token. Based on this score, we select the most important tokens to retain, while pruning those deemed less relevant, thereby reducing the overall number of visual tokens in the input.

3.2 Preliminary Experiment

We conduct a preliminary study using LLaVA-OneVision-7b-chat (Li et al., 2024a) on the MVBench dataset (Li et al., 2024c), where token pruning is applied at decoder layer L_p, guided by attention weights from an earlier layer L_g.

As shown in Figure 2, the choice of guidance layer L_g has a stronger impact on pruning effectiveness than the pruning layer L_p itself. This underscores the importance of selecting a semantically rich guidance layer. In particular, the 16th layer in LLaVA-OneVision proves to be a strong candidate for generating token importance scores.

Prior work often assumes $L_p = L_g$, attributing pruning performance to the pruning layer rather than the quality of the guidance (Zhong et al., 2024; Zhang et al., 2025; Lin et al., 2025). Our results challenge this assumption, showing that such coupling may lead to suboptimal pruning.

We observe that many visual tokens are inherently redundant and can be pruned with minimal performance loss when guided effectively. However, using deeper layers for guidance ($L_g > L_p$) introduces a trade-off: the model must prefill up to L_g to compute attention scores A_g, then reprocess from L_p after pruning. This two-step procedure adds significant inference overhead.

4 Method

4.1 Inference

Our observations indicate that some visual tokens are inherently redundant across layers, while the attention weights in shallow layers are not sufficiently effective at guiding the pruning. To address this, we employ a plug-and-play pruning classifier (referred to as the *classifier*) to capture the inherent redundancy of the visual features for pruning.

We denote the classifier as $p_\theta(\cdot)$. It is positioned just before the LLM $f(\cdot)$. During inference, we compute the importance score for each visual token with the classifier as follows:

$$\mathbb{S} = p_{\boldsymbol{\theta}}(\mathbf{H}_\mathsf{v}, \mathbf{H}_t), \tag{1}$$

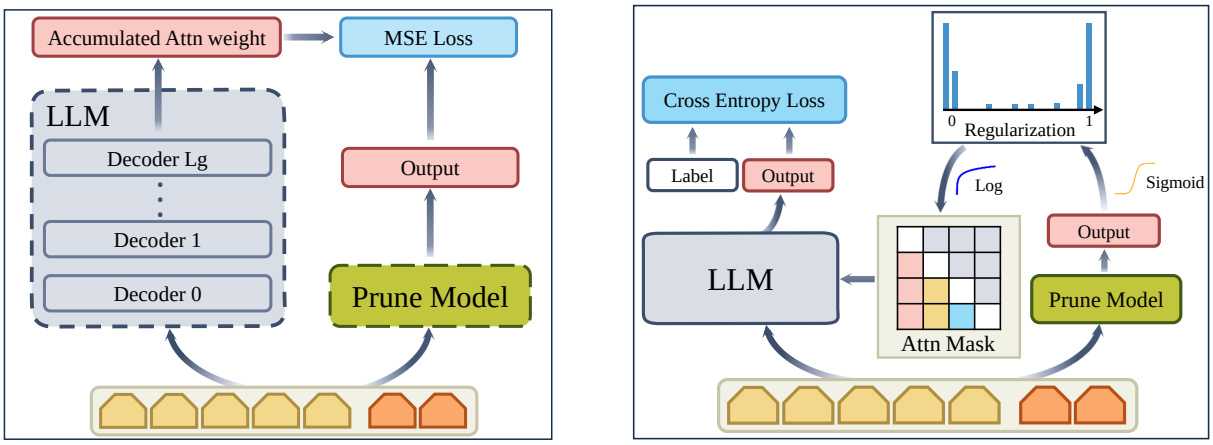
where $\mathbb{S}$ represents the importance scores. Based on these scores, we perform pruning with a given reserve ratio r. The indices of the visual tokens to be retained are determined by:

$$\mathbb{I} = \text{TopK}(\mathbb{S}, r \times n_v), \tag{2}$$

where n_v is the total number of visual tokens.

Note that $\mathbb{S}$ is computed over visual and text tokens, but only the scores for visual tokens. This is because visual token redundancy depends not only on visual features but also on textual context, which guides the model in identifying more relevant visual tokens (Sun et al., 2025; Yang et al., 2025).

A detailed discussion of the model architecture can be found in Section 5.



(a) The strategy of train stage 1.

(b) The strategy of train stage 2.

Figure 3: Two stage training strategy.

0	-inf	-inf	-inf	-inf	-inf	-inf
M_1	0	-inf	-inf	-inf	-inf	-inf
M_1	M_2	0	-inf	-inf	-inf	-inf
M_1	M_2	M_3	0	-inf	-inf	-inf
M_1	M_2	M_3	M_4	0	-inf	-inf
M_1	M_2	M_3	M_4	M_5	0	-inf
M_1	M_2	M_3	M_4	M_5	M_6	0

Figure 4: Attention mask at the training stage2.

4.2 Training Stage 1

Our observations indicate that attention weights in deeper layers of the LLM effectively guide the pruning process. The attention weights in these layers contain significant contextual information, which highlights the tokens that need to be attended by the attention mechanism. Therefore, we leverage the information to train the classifier, as illustrated in Figure 3a.

We denote the guiding attention weights as A_g, and the specific layer from which these weights are derived as l_g. For training, we use the accumulated attention weights, denoted as A_l, as the target labels. The label for the k-th token is computed by accumulating the attention weights over the relevant layers:

$$A_{l,k} = \sum_{i=1}^{h} \sum_{j=n-n_v}^{n} A_{g,i,j,k}, \tag{3}$$

where h represents the number of attention heads, n is the number of tokens.

The classifier outputs $\hat{\mathbb{S}}$ represents the predicted importance scores for the visual tokens. To train the classifier, we optimize the model using the mean squared error (MSE) loss function, aiming to minimize the discrepancy between the predicted scores $\hat{\mathbb{S}}$ and the accumulated attention labels A_l. The MSE loss is computed as follows:

$$\mathcal{L}_{\text{mse}} = \frac{1}{n_v} \sum_{k=1}^{n_v} \left(\hat{\mathbb{S}}_k - A_{l,k} \right)^2, \tag{4}$$

where $\hat{\mathbb{S}}_k$ and $A_{l,k}$ are the predicted and true importance scores, respectively, for the k-th visual token.

The objective is to train the classifier to output scores that align with the accumulated attention weights, which will then guide the pruning operation effectively.

This training is efficient, as only the small classifier $p_{\boldsymbol{\theta}}(\cdot)$ is optimized. The LLM parameters before layer l_g remain fixed, avoiding gradient computation, while those after layer l_g are dropped during training, greatly reducing both computation and memory overhead.

4.3 Training Stage 2

To further improve the model's capacity to capture contextual information and accurately identify important visual tokens, we introduce an end-to-end training phase, as illustrated in Figure 3b, which incorporates a differentiable approximation of the pruning operation.

Direct pruning of less-relevant visual tokens during training using hard indexing (e.g., $\mathbf{H}_{\text{v}}[\mathbb{I}]$) is non-differentiable and thus breaks the backpropa-

gation process. To address this, we simulate the pruning effect by modifying the attention mechanism through a soft attention mask.

We apply a sigmoid activation to the classifier outputs $\hat{\mathbb{S}}$ to normalize the predicted importance scores into the range $[0, 1]$:

$$\mathbb{P} = \sigma(\hat{\mathbb{S}}). \tag{5}$$

Here, $\mathbb{P}_i$ can be interpreted as the retention probability for the i-th visual token. To simulate pruning, we convert the normalized importance scores into attention biases using a logarithmic transformation:

$$B = \log(\mathbb{P}). \tag{6}$$

This transformation ensures that tokens with low importance scores receive large negative biases, thus masking them during attention computation.

We then construct the final attention mask 4 by adding the attention bias B_j to the standard causal attention mask:

$$M_{i,j} = M^{\text{causal}}_{i,j} + B_j, \quad \text{for } i > j, \tag{7}$$

where M^{causal} is the standard causal mask.

The model outputs predictions $\hat{y}$, and the ground truth labels are denoted as y. We define the training objective as a combination of the cross-entropy loss and a regularization term:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{ce}}(y, \hat{y}) + k \times \mathcal{L}_{\text{reg}}(\mathbb{P}), \tag{8}$$

where $\mathcal{L}_{\text{reg}}(\mathbb{P})$ enforces the model to retain a predefined ratio r of visual tokens and encourages the model to approximate the token pruning patterns during inference. The parameter k controls the weight of $\mathcal{L}_{\text{reg}}(\mathbb{P})$ in the overall loss function.

A naive regularization (Nawrot et al., 2024) such as:$\mathcal{L}_{\text{reg}} = \mathcal{L}_1(r, \text{mean}(\mathbb{P}))$ enforces a global retention rate r, but tends to collapse all probabilities $\mathbb{P}_i$ to values near r, harming discriminative capacity.

To promote a clearer distinction between important and unimportant visual tokens, we introduce a contrastive style regularization objective that explicitly separates their predicted importance scores.

We first compute the indices of the top and bottom tokens based on the classifier's normalized outputs $\mathbb{P} \in [0, 1]^{n_v}$, where n_v is the number of visual tokens:

$$\begin{aligned} \mathbb{I}_{\text{high}} &= \text{TopK}(\mathbb{P}, \lfloor r \cdot n_v \rfloor), \\ \mathbb{P}_{\text{high}} &= \mathbb{P}[\mathbb{I}_{\text{high}}], \\ \mathbb{I}_{\text{low}} &= \text{DTopK}(\mathbb{P}, \lfloor (1-r) \cdot n_v \rfloor), \\ \mathbb{P}_{\text{low}} &= \mathbb{P}[\mathbb{I}_{\text{low}}]. \end{aligned} \tag{9}$$

Then we define the regularization loss as:

$$\mathcal{L}_{\text{reg}} = \mathcal{L}_1\left(1, \text{mean}(\mathbb{P}_{\text{high}} - \mathbb{P}_{\text{low}})\right). \tag{10}$$

This objective aims to maximize the average margin between the most and least important tokens. Specifically: $\text{TopK}(\cdot)$ returns the indices of the top $r \cdot n_v$ visual tokens with the highest importance scores, $\text{DTopK}(\cdot)$ returns the indices of the bottom $(1-r) \cdot n_v$ tokens, $\mathcal{L}_1(1, \cdot)$ penalizes deviation from the target margin of 1 between high and low importance scores.

This regularization guides the classifier to assign high retention scores to top-ranked tokens and low scores to less relevant ones, aligning with the inference-time selection and enabling pruning-aware learning in a fully differentiable way.

5 Experiments

5.1 Experimental Setup

Baselines We evaluate our methods with three baseline approaches: FastV (Chen et al., 2024b), SparseVLM (Zhang et al., 2025), and PyramidDrop (Xing et al., 2024), all of which performing token pruning. FastV prunes visual tokens in a specific layer using self-attention scores of that layer. PyramidDrop prunes tokens in predefined layers based on attention weights. SparseVLM also prunes tokens in predefined layers but merges part of the pruned tokens and reserve them. FastV and SparseVLM are plug-and-play methods, while PyramidDrop offers both training-free and training-based strategies.

Base Models We conduct experiments on two state-of-the-art LVLMs: LLaVA-OneVision-7b-Chat (Xiong et al., 2024) and LLaVA-Video-7b (Zhang et al., 2024). LLaVA-OneVision-7b-chat is trained on a combination 4.8M dataset of image and video. LLaVA-Video-7b is fine-tuned from LLaVA-OneVision using a joint dataset, including LLaVA-Video-178K. For evaluating image tasks, we use LLaVA-OneVision-7b-Chat, while LLaVA-Video-7b is used for video task evaluations.

Classifier Model We design a compact classifier with two projection layers and 8 encoder layers. The first projection maps LVLM embeddings to the classifier input, and the second outputs a scalar score $\mathbb{S} = 1$. The encoder comprises 8 layers, each with a hidden size of 768, intermediate size of 3072, 16 attention heads, and 4 key-value heads, resulting in a total of 71.20M parameters.

Models	GQA	MME	SEED	MMStar	AI2D	OCRVQA	TextVQA	InfoVQA	Avg(%)
LLaVA-OV-7b	61.70	1605.41	76.59	61.67	82.77	59.83	75.02	65.52	100.00%
reserve ratio = 0.5									
FastV	60.89	1586.18	74.95	57.20	80.70	58.56	71.69	49.58	94.34%
SparseVLM	59.35	1560.75	74.40	54.67	78.69	45.96	69.49	42.06	88.49%
PDrop	61.02	1590.56	75.75	59.00	80.83	57.19	74.71	**60.90**	97.29%
PDrop*	59.97	1532.82	**75.89**	58.00	80.24	**59.90**	70.64	46.83	93.56%
CoViPAL	**61.31**	**1613.37**	75.48	**59.07**	**82.12**	57.85	**74.08**	59.66	**97.48%**
reserve ratio = 0.25									
FastV	56.12	1523.37	65.88	47.23	73.25	46.29	57.29	35.18	80.54%
SparseVLM	52.85	1415.58	67.42	45.40	70.30	32.09	43.61	28.03	71.88%
PDrop	58.02	1470.22	67.50	49.80	73.06	48.83	**68.32**	41.90	84.93%
PDrop*	57.77	1531.10	70.47	49.80	74.31	**49.22**	64.95	34.47	84.12%
CoViPAL	**59.93**	**1559.29**	**73.22**	**54.33**	**79.47**	48.92	65.99	**47.28**	**89.48%**

Table 1: Results of image benchmark.

Models	MVBench	MMBVideo	$MLVU^m$	$MLVU^g$	LongVB	WorldSense	Avg(%)
LLaVA-Video-7b	58.32	1.71	62.40	4.16	52.50	38.20	100.00%
reserve ratio = 0.5							
FastV	**56.87**	**1.67**	60.60	4.89	**52.60**	37.60	101.41%
SparseVLM	55.29	1.63	59.20	4.51	50.10	37.30	97.74%
Pdrop	55.21	1.63	56.80	4.96	52.10	34.80	98.43%
Pdrop*	55.74	1.60	**61.50**	4.73	49.60	**38.90**	99.62%
CoViPAL	56.66	1.66	61.40	**4.97**	51.80	38.10	**101.75%**
reserve ratio = 0.25							
FastV	52.74	1.55	55.90	4.68	48.20	36.50	95.08%
SparseVLM	50.00	1.52	54.50	4.33	47.00	36.30	91.77%
Pdrop	50.50	1.55	53.30	4.70	48.90	33.50	92.74%
Pdrop*	53.03	1.58	**59.20**	4.72	48.30	**37.70**	97.06%
CoViPAL	**55.42**	**1.61**	55.80	**4.85**	**51.30**	37.20	**98.38%**

Table 2: Results of video benchmark.

Training Implementation In training stage 1, we use 3% of LLaVA-NeXT-Data (which is 0.46% of the training data of LLaVA-OneVision-7b-Chat), totaling 22.2K samples, to train the classifier with base model LLaVA-OneVision-7b-Chat. After stage 1, we proceed to Stage 2, initializing the classifier from Stage 1. During the training stage 2, we trained two classifiers. One is trained on the same data as training stage 1 with the base model LLaVA-OneVision-7b-chat, this classifier is used for image benchmark evaluation. And another on 20% of the 0_30_s_academic_v0_1 (13.2K samples) dataset with LLaVA-Video-7b for video benchmark evaluation. For PyramidDrop, we fine-tune two models using LoRA (Hu et al., 2022): one on 10% of LLaVA-NeXT-Data with LLaVA-OneVision-7b-Chat for image evaluation, and the other on 60% of 0_30_s_academic_v0_1 with LLaVA-Video-7b for video evaluation. The larger dataset for PyramidDrop ensures consistent training time, as Stage 2 is incompatible with Flash Attention (Dao et al., 2022), which doesn't support this type of custom attention mask currently.

Training Hyperparameters Each run is trained for one epoch using Bfloat16 precision. The learning rate is set to 1e-5, except in Stage 2 where it is reduced to 0.5e-5 to preserve parameters from Stage 1. We set $k = 0.01$ in Eq. 8, and apply a cosine scheduler. For PyramidDrop, we use a LoRA rank of 32 (97.72M trainable params). The input length is capped at 3000 tokens. For image input, we use the anyres-max-2 setting, producing up to 2189 visual tokens-leaving room for text to avoid truncation. For video input, we allow up to 8 frames (max 1568 visual tokens). The reserve ratio during training is fixed at 0.25, as we believe that a training-based method should be robust enough to accommodate differences in reserve ratio between

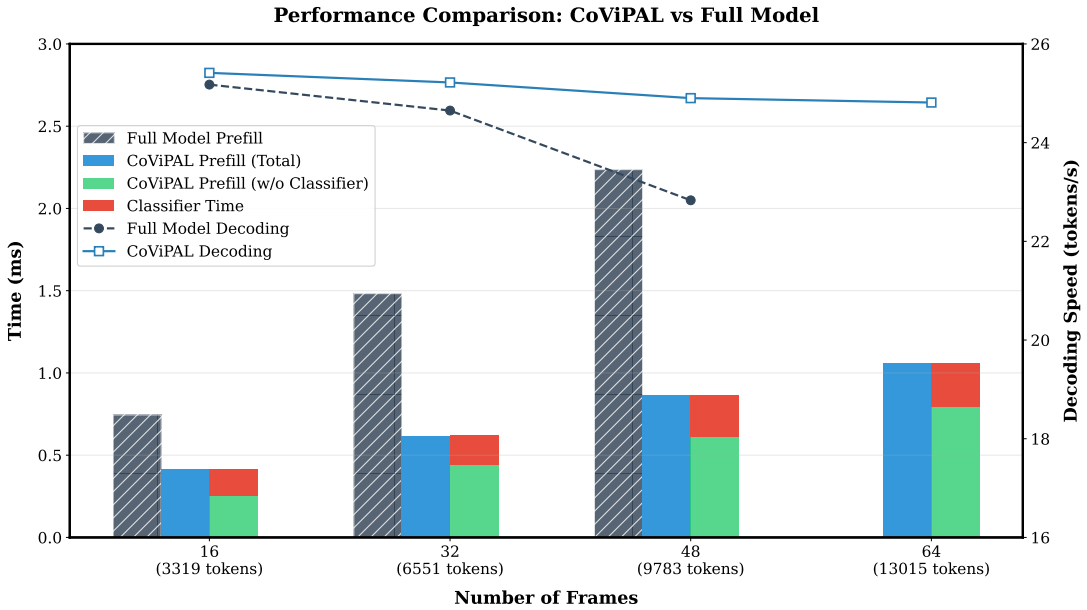


Figure 5: Efficiency Results of CoViPAL on LLaVA-OneVision-7b-Chat.

training and inference settings.

Evaluation Benchmarks We evaluate our methods on eight image and five video benchmarks spanning visual reasoning, multimodal comprehension, temporal understanding and so on. This diverse set ensures a comprehensive assessment across visual inputs. Details are provided in the Appendix B. All evaluations use VLMEvalKit (Duan et al., 2024).

5.2 Evaluation Results

Image Benchmarks We evaluate CoViPAL on eight widely used image benchmarks, with the results reported in Table 1. Our results indicate that CoViPAL effectively preserves the model's image comprehension capabilities on tasks of real-world scenarios. CoViPAL consistently surpasses the three baseline methods when retaining 50% or only 25% of the image tokens. Particularly, when the reserve ratio is set to 25%, which significantly challenges the model's token selection capability, CoViPAL demonstrates superior performance by accurately identifying and preserving the most crucial visual tokens. Additionally, results confirm the robustness of CoViPAL, as performance remains stable even when the inference reserve ratio (50%) differs from the training reserve ratio (25%).

Video Benchmarks We further evaluate CoViPAL on five widely recognized video benchmarks, with the results summarized in Table 2. The experimental results demonstrate that CoViPAL effectively eliminates redundant or less relevant visual tokens, leading to performance improvements under various conditions. CoViPAL consistently outperforms the three comparative baselines, exhibiting only a minor performance degradation of 1.62% when pruning 75% of the visual tokens. Moreover, the results suggest that videos are more information-sparse compared to images, containing a higher proportion of redundant visual tokens, thereby making video tasks inherently more robust to token pruning.

Efficiency Results We evaluate the efficiency of CoViPAL on LLaVA-OneVision-7b-Chat with video input on a single RTX 3090 GPU. The sample frame size ranges from 16 to 64, resulting in input tokens ranging from 3k to 13k. With a reserve ratio of 0.25, we measure prefilling time, decoding speed for generating 1k tokens, and the classifier's overhead during prefilling. Results are shown in Figure 5.

CoViPAL significantly reduces prefilling time and accelerates decoding. For 48-frame input, it cuts prefilling time by over 60% and enables 64-frame inference on a 24G GPU, where the original model fails due to memory limits.

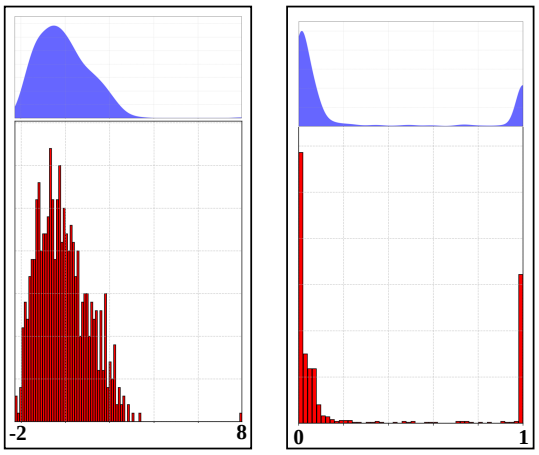
5.3 Ablation Study

5.4 Similarity Based Token Replacement

Model Structure for Contextual Information Capture We compare two classifier models: a multi-layer encoder with 71.2M parameters ($p_{\boldsymbol{\theta}}^{8}$) and a single-layer encoder with 165.18M parame-

Models	GQA	MME	SEED	MMStar	AI2D	OCRVQA	TextVQA	InfoVQA
LLaVA-OV-7b	61.70	1605.41	76.59	61.67	82.77	59.83	75.02	65.52
reserve ratio = 0.5								
$p_{\boldsymbol{\theta}}^1$	60.85	1547.10	74.85	58.20	81.22	52.77	68.35	52.12
$p_{\boldsymbol{\theta}}^8$	**61.31**	**1613.37**	**75.48**	**59.07**	**82.12**	**57.85**	**74.08**	**59.66**
reserve ratio = 0.25								
$p_{\boldsymbol{\theta}}^1$	57.21	1446.89	70.67	51.20	76.91	37.43	48.21	35.55
$p_{\boldsymbol{\theta}}^8$	**59.93**	**1559.29**	**73.22**	**54.33**	**79.47**	**48.92**	**65.99**	**47.28**

Table 3: Ablation Study on model structure.



(a) $\mathbb{S}$ of classifier after training stage 1.

(b) $\mathbb{P}$ of classifier after training stage 2.

Figure 6: The distribution of classifier outputs after two training stages.

ters ($p_{\boldsymbol{\theta}}^1$), in which the letter use the same settings as LLaVA-OneVision-7b-Chat decoder architecture.

Trained with the same two-stage strategy on 3% of LLaVA-NeXT-Data, $p_{\boldsymbol{\theta}}^8$ consistently outperforms $p_{\boldsymbol{\theta}}^1$ on image tasks, as shown in Table 3. Despite its smaller size, the deeper model captures redundant token patterns more effectively, highlighting the advantage of deeper attention layers in modeling contextual information for pruning.

k for Two-Stage Training The hyperparameter k in Eq. 8 is crucial in Stage 2. A large k causes early sharp separation of retain probabilities $\mathbb{P}$ which hindering the subsequent training, while a small k keeps $\mathbb{P}$ continuous, misaligned with the discrete selection required during inference. The distribution of classifier outputs are showed in Appendix C.

We train with k values from 0.0001 to 0.1 using LLaVA-OneVision-7b-Chat and evaluate on GQA (Hudson and Manning, 2019). For $k = 0.0001$, we warm up with $k = 0.01$ to avoid continuous distribution throughout training. As shown in Table 4, $k = 0.01$ yields the best performance.

Models	$r = 50\%$	$r = 25\%$
LLaVA-OV-7b	61.70	
$k = 0.1$	61.19	59.31
$k = 0.01$	**61.31**	**59.94**
$k = 0.0001$	61.11	58.73

Table 4: Ablation study on k.

Effectiveness of the Training Strategy Training Stage 2 from a randomly initialized model led to a collapse of retain probabilities $\mathbb{P}$ to 0 throughout training, even with $k = 0.1$, as shown in Figure 7c. In contrast, initializing from the Stage 1 model Figure 6a allowed $\mathbb{P}$ to stabilize and discretize effectively Figure 6b.

These results underscore the value of the two-stage strategy: Stage 1 captures contextual attention patterns, providing a strong initialization for Stage 2 to identify redundant tokens and simulate pruning under smaller k.

6 Conclusion

We propose CoViPAL, a novel contextualized visual token pruning method that efficiently reduces the computational and memory overhead of Large Vision-Language Models by leveraging a lightweight and plug-and-play pruning module. CoViPAL identifies and removes redundant visual tokens across all layers with minimal supervision, achieving up to 50% reduction in pre-filling time and pruning 75% of visual tokens while maintaining competitive performance. Our method outperforms both training-free and training-based approaches, offering a scalable and adaptable solution for efficient multimodal inference. This work provides new insights into visual token redundancy and paves the way for deploying LVLMs in resource-constrained settings.

Limitations

While our approach has been validated on representative LVLMs, the diversity of model backbones explored so far remains limited. In future work, we plan to extend our method to a broader range of architectures, including base models from the LLaMA and Mistral families, to assess its applicability across different LVLM paradigms and better understand its architectural generality.

In addition, the current experiments are conducted on models of moderate scale. Scaling up to larger model sizes will allow us to further investigate the generalization and effectiveness of our pruning framework in high-capacity settings. These extensions will provide deeper insights into the scalability and robustness of our approach.

References

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Slav Petrov, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy P. Lillicrap, and 33 others. 2023. Gemini: A family of highly capable multimodal models. *CoRR*, abs/2312.11805.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *CoRR*, abs/2308.12966.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2022. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461*.

Jieneng Chen, Luoxin Ye, Ju He, Zhao-Yang Wang, Daniel Khashabi, and Alan Yuille. 2024a. Efficient large multi-modal models via visual context compression. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. 2024b. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *Computer Vision - ECCV 2024 - 18th European Conference, Milan, Italy, September 29-October 4, 2024, Proceedings, Part LXXXI*, volume 15139 of *Lecture Notes in Computer Science*, pages 19–35. Springer.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024c. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330*.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359.

Haodong Duan, Junming Yang, Yuxuan Qiao, Xinyu Fang, Lin Chen, Yuan Liu, Xiaoyi Dong, Yuhang Zang, Pan Zhang, Jiaqi Wang, and 1 others. 2024. Vlmevalkit: An open-source toolkit for evaluating large multi-modality models. In *Proceedings of the 32nd ACM international conference on multimedia*, pages 11198–11201.

Xinyu Fang, Kangrui Mao, Haodong Duan, Xiangyu Zhao, Yining Li, Dahua Lin, and Kai Chen. 2024. Mmbench-video: A long-form multi-shot benchmark for holistic video understanding. *Advances in Neural Information Processing Systems*, 37:89098–89124.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. 2024. Mme: A comprehensive evaluation benchmark for multimodal large language models. *Preprint*, arXiv:2306.13394.

Jack Hong, Shilin Yan, Jiayin Cai, Xiaolong Jiang, Yao Hu, and Weidi Xie. 2025. Worldsense: Evaluating real-world omnimodal understanding for multimodal llms. *arXiv preprint arXiv:2502.04326*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Drew A Hudson and Christopher D Manning. 2019. 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*, pages 6700–6709.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. 2024a. Llava-onevision: Easy visual task transfer. *CoRR*, abs/2408.03326.

Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. 2024b. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. *arXiv preprint arXiv:2404.16790*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

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

Zhihang Lin, Mingbao Lin, Luxi Lin, and Rongrong Ji. 2025. Boosting multimodal large language models with visual tokens withdrawal for rapid inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 5334–5342.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Preprint*, arXiv:2304.08485.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. 2022. Infographicvqa. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1697–1706.

Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. 2019. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, pages 947–952. IEEE.

Piotr Nawrot, Adrian Lancucki, Marcin Chochowski, David Tarjan, and Edoardo M. Ponti. 2024. Dynamic memory compression: Retrofitting llms for accelerated inference. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Min Joon Seo, Hannaneh Hajishirzi, Ali Farhadi, and Oren Etzioni. 2014. Diagram understanding in geometry questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Yuzhang Shang, Mu Cai, Bingxin Xu, Yong Jae Lee, and Yan Yan. 2024. Llava-prumerge: Adaptive token reduction for efficient large multimodal models. *arXiv preprint arXiv:2403.15388*.

Dachuan Shi, Chaofan Tao, Anyi Rao, Zhendong Yang, Chun Yuan, and Jiaqi Wang. 2023. Crossget: Cross-guided ensemble of tokens for accelerating vision-language transformers. *arXiv preprint arXiv:2305.17455*.

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

Yizheng Sun, Yanze Xin, Hao Li, Jingyuan Sun, Chenghua Lin, and Riza Batista-Navarro. 2025. Lvpruning: An effective yet simple language-guided vision token pruning approach for multi-modal large language models. *arXiv preprint arXiv:2501.13652*.

Zhongwei Wan, Ziang Wu, Che Liu, Jinfa Huang, Zhihong Zhu, Peng Jin, Longyue Wang, and Li Yuan. 2024. Look-m: Look-once optimization in kv cache for efficient multimodal long-context inference. *arXiv preprint arXiv:2406.18139*.

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

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. 2024. Pyramidrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. *CoRR*, abs/2410.17247.

Tianyi Xiong, Bo Li, Dong Guo, Huizhuo Yuan, Quanquan Gu, and Chunyuan Li. 2024. Llava-onevision-chat: Improving chat with preference learning.

Dingchen Yang, Bowen Cao, Anran Zhang, Weibo Gu, Winston Hu, and Guang Chen. 2025. Beyond intermediate states: Explaining visual redundancy through language. *arXiv preprint arXiv:2503.20540*.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. 2025. Fit and prune: Fast and training-free visual token pruning for multi-modal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22128–22136.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and 1 others. 2025. Sparsevlm: Visual token sparsification for efficient vision-language model inference. In *International Conference on Machine Learning*.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. 2024. Video instruction tuning with synthetic data. *CoRR*, abs/2410.02713.

Yiwu Zhong, Zhuoming Liu, Yin Li, and Liwei Wang. 2024. Aim: Adaptive inference of multi-modal llms via token merging and pruning. *arXiv preprint arXiv:2412.03248*.

Junjie Zhou, Yan Shu, Bo Zhao, Boya Wu, Zhengyang Liang, Shitao Xiao, Minghao Qin, Xi Yang, Yongping Xiong, Bo Zhang, Tiejun Huang, and Zheng Liu. 2025. Mlvu: Benchmarking multi-task long video understanding. *Preprint*, arXiv:2406.04264.

A Related Work

A.1 Large Vision-Language Models

Large vision-language models (LVLMs) combine vision encoders with large language models to jointly process image and text inputs. This multimodal architecture has achieved strong performance on tasks like visual question answering and captioning, with representative models including BLIP-2(Li et al., 2023), Qwen-VL (Bai et al., 2023), and the LLaVA series (Liu et al., 2024). LLaVA-OneVision (Li et al., 2024a) extends LVLMs to handle single-image, multi-image, and video inputs in a unified framework, while LLaVA-Video (Zhang et al., 2024) adapts to the video domain via instruction tuning. However, rich visual inputs often produce thousands of tokens, leading to high computational and memory costs. This bottleneck limits inference efficiency and practical deployment, highlighting the need for token compression to make LVLMs more scalable and efficient.

B Benchmark Detail

We evaluate our method on a diverse collection of vision-language benchmarks, covering both image and video modalities. As summarized in Table Table 5, the image-based benchmarks include GQA (Hudson and Manning, 2019), MME (Fu et al., 2024), SEED-Bench (Li et al., 2024b), MM-Star (Chen et al., 2024c), AI2D (Seo et al., 2014), OCR-VQA (Mishra et al., 2019), TextVQA (Singh et al., 2019), and InfographicVQA (Mathew et al., 2022).

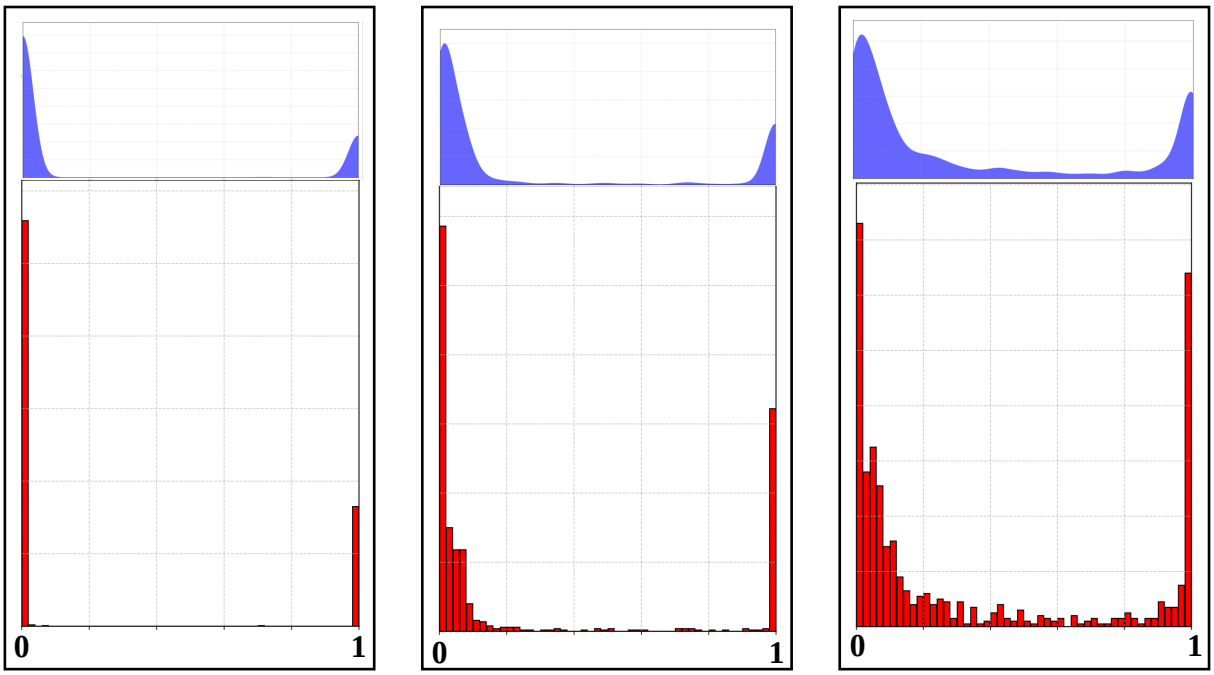
For video-based evaluation, we adopt MVBench (Li et al., 2024c), MMBench-Video (Fang et al., 2024), MLVU (Zhou et al., 2025), LongVideoBench (Wu et al., 2024), and WorldSense (Hong et al., 2025). These benchmarks collectively provide a comprehensive testbed for assessing both the effectiveness and generalizability of our proposed method.

C Classifier Output Distribution

We provide the distribution of the classifier model outputs after training stage 2, which highlights the influence of different settings for the hyperparameter k during this stage. The hyperparameter k plays a crucial role in the second stage of training. When k is set to 0.1, the retain probabilities $\mathbb{P}$ become sharply separated at the beginning of stage 2, as shown in Figure 7a, which can hinder subsequent training. On the other hand, when k is set to 0.0001, a large portion of the $\mathbb{P}$ values remain continuous, as seen in Figure 7c, which prevents the values from approximating the discrete selection patterns needed during inference. When $k = 0.1$, the distribution of the classifier's output in Figure 7b aligns with the pruning operation during the inference stage, allowing the model to gradually identify redundant tokens and simulate pruning under smaller values of k.

Modality	Benchmark	Short Name	Task Feature
Image	GQA	GQA	Visual attribute reasoning
	MME	MME	Multimodal evaluation across modalities
	SEED-Bench	SEED	Generative multimodal comprehension
	MMStar	MMStar	Vision tasks with minimal data leakage
	AI2D	AI2D	Diagram understanding
	OCR-VQA	OCRVQA	Text-based image reasoning
	TextVQA	TextVQA	Scene text understanding
	InfographicVQA	InfoVQA	Multimodal infographic reasoning
Video	MVBench	MVBench	Temporal understanding in videos
	MMBench-Video	MMBenchV	Long-form video reasoning
	MLVU	MLVU	Multi-task video understanding
	LongVideoBench	LongVB	Interleaved video-language reasoning
	WorldSense	WorldSense	Omni-modal (visual/audio/text) understanding

Table 5: Detailed Evaluation Benchmarks



(a) $\mathbb{P}$ of classifier after training stage 2 when $k = 0.1$.

(b) $\mathbb{P}$ of classifier after training stage 2 when $k = 0.01$.

(c) $\mathbb{P}$ of classifier after training stage 2,$k = 0.0001$.

Figure 7: The distribution of classifier outputs after training stages 2 when setting different k.