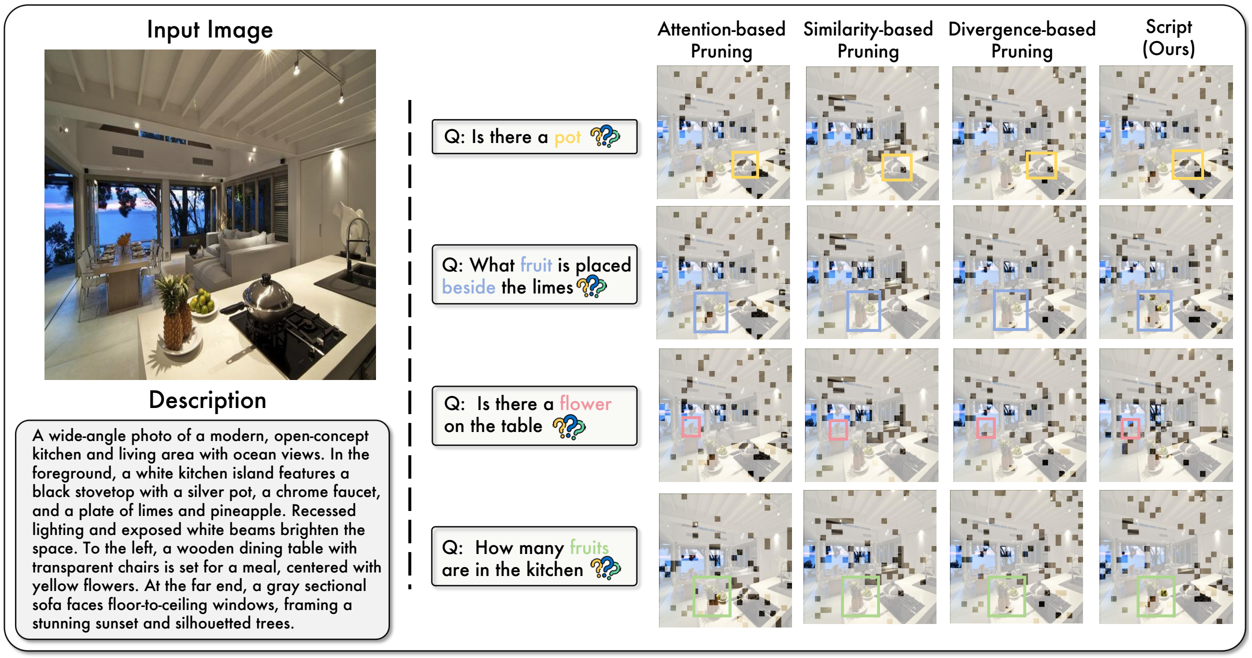
Script: Graph-Structured and Query-Conditioned Semantic Token Pruning for Multimodal Large Language Models

Anonymous authors
Paper under double-blind review



A wide-angle photo of a modern, open-concept kitchen and living area with ocean views. In the foreground, a white kitchen island features a black stovetop with a silver pot, a chrome faucet, and a plate of limes and pineapple. Recessed lighting and exposed white beams brighten the space. To the left, a wooden dining table with transparent chairs is set for a meal, centered with yellow flowers. At the far end, a gray sectional sofa faces floor-to-ceiling windows, framing a stunning sunset and silhouetted trees.

Figure 1: **Comparison of different token pruning methods.** Attention-based and similarity-based methods prune tokens using attention scores and similarity scores, respectively. In contrast, divergence-based methods detect changes in model performance and retain tokens that cause minimal impact. **Script** (Graph-**S**tructured and Que**R**y-**C**ond**I**tioned **T**oken **P**runing) combines graph-structured reduction of visual redundancy and query-conditioned semantic token selection to enable efficient pruning in MLLMs. In this example, Script successfully preserves key visual cues, such as the **silver pot** on the stove, the **pineapple** beside the **limes**, and the **flowers** on the table. Other methods fail to retain consistently.

Abstract

The rapid growth of visual tokens in multimodal large language models (MLLMs) leads to excessive memory consumption and inference latency, especially when handling high-resolution images and videos. Token pruning is a technique used to mitigate this issue by removing redundancy, but existing methods often ignore relevance to the user query or suffer from the limitations of attention mechanisms, reducing their adaptability and effectiveness. To address these challenges, we propose Script, a plug-and-play pruning method that requires no retraining and generalizes across diverse MLLMs. Script comprises two modules: a graph-structured pruning module that removes visually redundant tokens, and a query-conditioned semantic pruning module that preserves query-relevant visual information. Together, they enhance performance on multimodal tasks. Experiments on fourteen benchmarks across image and video understanding tasks show that Script consistently achieves higher model efficiency and predictive accuracy compared to existing pruning methods. On LLaVA-NeXT-7B, it achieves up to 6.8× prefill speedup and 10× FLOP reduction, while retaining 96.88% of the original performance. Code will be made publicly available upon acceptance.

1 Introduction

Recent advances in large language models (LLMs) (Touvron et al., 2023a;b; Bai et al., 2023; Yang et al., 2024) have substantially enhanced language understanding and reasoning, forming the backbone of general-purpose multimodal systems, including vision-language models and embodied agents operating in real-world scenarios. Building on this progress, multimodal large language models (MLLMs) (Li et al., 2024a; Chen et al., 2024d;c; Zhang et al., 2025) extend LLMs by integrating vision encoders, enabling joint reasoning over both visual and textual modalities. This integration empowers MLLMs to excel at diverse vision-language tasks such as visual question answering and image captioning.

However, deploying MLLMs in practical scenarios such as mobile agents or interactive assistants is constrained by the high computational cost of handling high-resolution or temporally extended visual inputs. Patch-based vision encoders typically tokenize each visual input into hundreds or even thousands of tokens (Luo et al., 2024; Guo et al., 2024; Zhang et al., 2023; Chen et al., 2024b; Li et al., 2025a; Liu et al., 2025; Bai et al., 2025). In contrast to compact textual inputs, visual representations often contain hundreds or thousands of tokens, which increases memory usage and inference latency. The explosion of vision tokens, compounded by the quadratic complexity of attention mechanisms (Vaswani et al., 2017), creates a significant bottleneck for latency- and memory-sensitive scenarios such as mobile deployment, online inference, and edge-based vision-language applications. To mitigate this inefficiency, visual token pruning has emerged as a promising strategy. An effective pruning method should minimize computational cost while preserving task performance within resource-constrained environments.

Existing visual token pruning methods can be broadly categorized into three main paradigms: (1) attention-based methods, which retain tokens with high model-assigned importance, commonly referred to as attention scores (Chen et al., 2024a; Xing et al., 2025; Zhang et al., 2024d); (2) similarity-based methods, which identify and eliminate redundant visual tokens based on feature similarity (Bolya et al., 2023; Zhang et al., 2024c; Wen et al., 2025b; Jeddi et al., 2025); and (3) divergence-minimization methods, which prune tokens by minimizing the change in the model's output (Alvar et al., 2025; Ye et al., 2025).

However, existing pruning methods face two fundamental challenges in query-conditioned scenarios. **First**, attention-based approaches depend on raw attention scores, which are susceptible to the *attention sink* issue (Barbero et al., 2025), often overlooking critical tokens. Moreover, assigning similar scores to adjacent or semantically similar tokens can reduce pruning efficiency (Alvar et al., 2025). As shown in Figure 1, such methods fail to preserve the token representing the flower, despite its clear relevance to the query. **Second**, similarity- and divergence-based methods lack explicit query conditioning. Although they address visual redundancy or output stability, they generate fixed token subsets regardless of the input query, which can lead to the omission of query-referenced objects, such as the pineapple and lime shown in Figure 1.

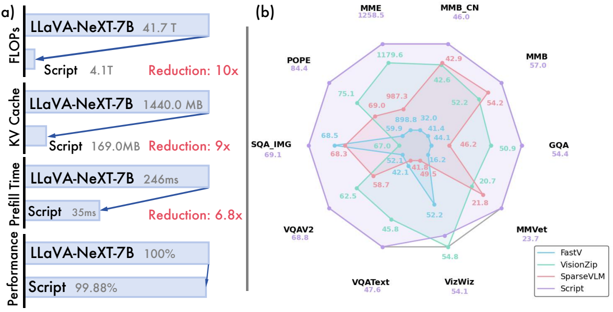


Figure 2: (a) Efficiency Analysis on LLaVA-NeXT-7B under 88.9% reduction. (b) Comparison with other baselines on LLaVA-1.5-7B under 94.4% reduction.

To address these limitations, we propose **Script**, a training-free and architecture-agnostic token pruning approach that combines graph-structured pruning (GSP) and query-conditioned semantic pruning (QCSP). On one hand, we build a bipartite graph to structure visual token redundancy, effectively identifying redundant tokens with lower computational cost. On the other hand, to eliminate reliance on attention scores and avoid issues such as attention sink, Script explicitly models interactions between the query and visual tokens, identifying query-relevant tokens using Determinantal Point Processes (DPP), a probabilistic model that favors diverse and semantically meaningful subsets. Together, these two modules enable effective token pruning by jointly considering visual redundancy and input query relevance. As illustrated in Figure 2, Script trims 88.9% tokens yet preserves 99.88% accuracy on POPE, and consistently outperforms other baselines on 10 benchmarks. In summary, our contributions are as follows:

- We identify the core limitations of existing pruning paradigms in query-conditioned scenarios, highlighting their inability to adaptively preserve task-relevant content.
- We propose Script, a training-free pruning method that combines graph-based redundancy reduction with query-aware token selection via DPP.
- Experiments across fourteen benchmarks demonstrate that Script consistently overperforms other baselines, and achieves up to **10**× speedup while retaining **96.88%** of original performance.

2 Related Work

Scalability Challenges in MLLMs. MLLMs combine visual and textual modalities to support general-purpose perception and reasoning (Alayrac et al., 2022; Zhu et al., 2024; Chen et al., 2023; Yu et al., 2024; Liu et al., 2024b; Cai et al., 2024). However, visual inputs introduce substantial token overhead due to spatial redundancy and high dimensionality, exacerbating the quadratic complexity of Transformer attention. For example, LLaVA-1.5 (Liu et al., 2023) generates 576 tokens from a 336×336 image, which is already 4–5× longer than typical text-only prompts. High-resolution models like LLaVA-NeXT (Liu et al., 2024a) and video models like LongVA (Zhang et al., 2024a) push this further, producing tens or even hundreds of thousands of tokens per input, leading to untenable memory and compute demands for many downstream applications. Consequently, such models often encounter memory bottlenecks, latency spikes, or even inference failures, especially under real-time or edge deployment scenarios (Papa et al., 2023; Li et al., 2025b), where computational resources are inherently limited. These challenges have motivated a growing body of work on token pruning, aiming to reduce input size while preserving task-relevant semantics and ideally without retraining or architectural changes.

Token Pruning Strategies. Efforts to mitigate visual token overhead in MLLMs can be broadly categorized into four approaches, each with distinct assumptions and limitations. (1) Pre-fusion compression, which downsamples or selects tokens before vision-language fusion (Li et al., 2024b; Hu et al., 2024), often requires model retraining or structural modifications, hindering plug-and-play usage. (2) Attention-based pruning, which selects tokens using attention scores (Chen et al., 2024a; Ye et al., 2025), suffers from attention drift (Wen et al., 2025a), misguiding token importance, and limiting compatibility with optimized backends such as FlashAttention (Dao, 2024). (3) Pre-language fusion pruning, which drops tokens before language-level alignment (Shang et al., 2024; Song et al., 2025), is often tightly coupled to specific vision backbones, limiting transferability across architectures. (4) Similarity-based pruning, which eliminates redundant tokens using intra-image feature similarity (Wen et al., 2025b; Alvar et al., 2025), is generally efficient and model-agnostic, but often ignores query semantics, leading to the loss of task-relevant content.

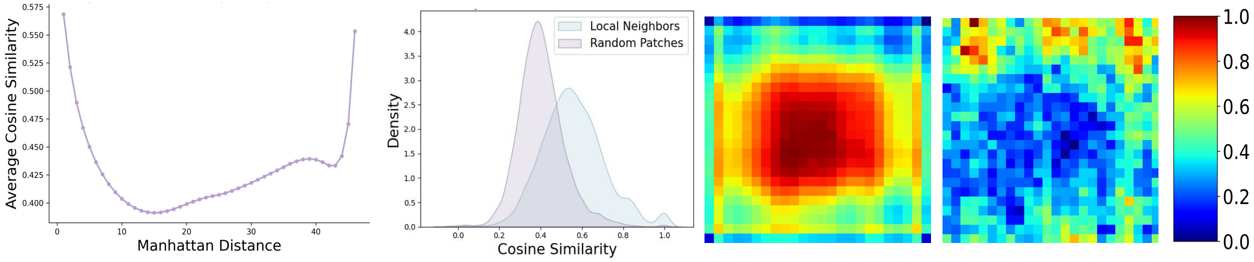
While existing approaches partially alleviate visual redundancy, few simultaneously achieve the trifecta of query relevance, architectural generality, and training-free deployment. In contrast, our proposed **Script** is explicitly query-aware, visual diversity, entirely model-agnostic, and operates without any additional training.

3 Preliminary

In this section, within the context of the task Text Visual Question Answering (TextVQA), we empirically investigate two key aspects of visual tokens in MLLMs: visual redundancy and query-conditioned relevance. In addition, we present a theoretical formulation of efficient query-conditioned token selection. The insights from both empirical and theoretical analysis inform the design of our **Script**.

3.1 Research Objective

Existing MLLMs (Liu et al., 2023; 2025; Xu et al., 2025) typically comprise three components: a vision encoder f_v, a multimodal projector g, and a large language model f_ϕ. After f_v processes an input image $\boldsymbol{X}_v$, the projector g maps the features into a visual-token sequence $\boldsymbol{H}_v \in \mathbb{R}^{n \times d}$, where n is the token count and d is the token dimension. The language model f_ϕ processes the visual tokens $\boldsymbol{H}_v$ and the tokenized query $\boldsymbol{H}_q$ to generate an answer grounded in the image. Typically, $n \gg |\boldsymbol{H}_q|$, i.e., the visual sequence is much longer



(a)Average Cosine Similarity vs. Spatial Distance

(b)Similarity Distribution: Local vs Random

(c) All Images Entropy Heatmap

(d) All Images Similarity Heatmap

Figure 3: **Token redundancy visualized via similarity and entropy on 10,000 COCO images.**

than the textual query. Equation 1 defines vision-token pruning by seeking the subset $\tilde{\boldsymbol{H}}_v^*$ that preserves model performance while using the fewest tokens.

$$\tilde{\boldsymbol{H}}_v^* = \arg \min_{\tilde{\boldsymbol{H}}_v \subseteq \boldsymbol{H}_v,\, |\tilde{\boldsymbol{H}}_v| = m} \mathcal{L}\left(f_\phi([\tilde{\boldsymbol{H}}_v; \boldsymbol{H}_q]), f_\phi([\boldsymbol{H}_v; \boldsymbol{H}_q])\right). \tag{1}$$

Here, $\mathcal{L}$ measures the discrepancy between model outputs before and after pruning, serving as a proxy for performance loss. The candidate subset $\tilde{\boldsymbol{H}}_v \subset \boldsymbol{H}_v$ contains m tokens, with $m < n$. To ensure query-aware pruning, it is essential to explicitly capture the semantic interactions between visual tokens $\boldsymbol{H}_v$ and their corresponding query tokens $\boldsymbol{H}_q$. These query-specific interactions are crucial for identifying the most query-relevant subset $\tilde{\boldsymbol{H}}_v$. At the same time, assessing the visual redundancy between $\tilde{\boldsymbol{H}}_v$ and the full set $\boldsymbol{H}_v$ is vital for promoting diversity in token selection, which is important for generalizing across a wide range of tasks. Together, these two aspects contribute to forming an informative and compact token subset.

3.2 Rethinking Visual Redundancy

In natural images, spatially adjacent tokens often encode highly overlapping content, resulting in what we refer to as local redundancy. Carefully identifying and removing such redundancy can significantly reduce computation without compromising accuracy.

Table 1: Comparison of redundancy estimation methods with a 90% pruning ratio on the POPE Benchmark. **Sim. (%)**: Percentage of similarity pairs considered. **Time**: Inference time in milliseconds. **IOU@K**: Intersection-over-Union with top-K selections. **Acc.**: Classification accuracy (%). **Speedup**: Relative time speedup compared to Full Graph.

Method	**Sim. (%)**	**Time**	**IOU@K**	**Acc.**	**Speedup**
Exhaustive	100	100	1.00	85.14	1.00x
Bipartite	∼50	35	0.93	84.49	2.81x
Random	0	34	0.52	78.63	2.82x

To investigate local redundancy, we adopt the approach from (Wang et al., 2024) and compute Manhattan distances and cosine similarities between vision-token pairs. We group similarity values by distance and calculate the average for each group. As shown in Figure 4(a), cosine similarity is highest for nearby tokens and decreases until a distance of roughly 15. Beyond this, we observe that similarity increases again at longer distances, indicating that long-range token redundancy also exists and deserves attention. Furthermore, we visualize the similarity distributions for neighboring tokens ($hop = 1$) and randomly sampled neighbor token pairs ($hop > 1$). As shown in Figure 4(b), the number of neighboring tokens with similarity greater than 0.5 is consistently higher than that of random token pairs. These results indicate that neighboring (i.e., local) tokens encode more similar visual features, but long-range redundancy should also be taken into account, as it clearly exists.

Insight 1: Redundancy Exists Beyond Local Neighborhoods

Spatially adjacent tokens show high similarity (local redundancy), yet long-range pairs still share appreciable similarity, revealing long-range redundancy. Effective pruning must therefore model both local and long-range redundancy to avoid information loss.

To further validate local redundancy and its estimation via cosine similarity, we introduce local information entropy as a distribution-level measure of neighborhood compactness and compare it with cosine similarity. For each token, we extract its 3×3 Moore neighbourhood $\mathcal{N}(v) = \{\mathbf{h}_i \mid \|\text{pos}(i) - \text{pos}(j)\|_1 \leq 1\}$ (hop=1). We then project all vectors in the neighborhood onto their first principal component and discretize the resulting scalar values into 20 equal-width bins to estimate the probability distribution $\{p_i\}$. The local information entropy is defined as:

$$\mathcal{H}_v = -\sum_{i=1}^{n} p_i \log , \tag{2}$$

where $p_i = p_i + \epsilon$ and $\epsilon = 10^{-8}$. A compact neighborhood typically exhibits low $\mathcal{H}_v$, as its token representations are similar and concentrated in a few bins. In contrast, a diverse neighborhood naturally yields higher $\mathcal{H}_v$, reflecting greater variation among neighboring tokens. As shown in Figures 3(c) and (d), regions with low local similarity typically exhibit high entropy, confirming local redundancy and validating cosine similarity as an effective proxy.

Insight 2: Cosine Similarity is a Valid Redundancy Proxy

Local information entropy inversely correlates with cosine similarity. Low-entropy (predictable) regions coincide with high neighbor similarity.

Motivated by the validated role of cosine similarity in capturing visual redundancy and the need for computational efficiency, we design a lightweight bipartite graph-based estimator that approximates both local and long-range redundancy using cosine similarity alone. This estimator forms the core of the graph-structured pruning module detailed in Section 4.1.

Insight 3: Lightweight Bipartite Graph Yields Scalable Redundancy Estimation

By weighting edges with cosine similarity, the even–odd bipartite graph achieves redundancy estimation comparable to exhaustive computation, but with significantly lower computational cost.

Tokens are represented as nodes, divided into even- and odd-indexed sets; each node in one set connects to all nodes in the other, with edge weights defined by cosine similarity. These weights enable efficient redundancy estimation and guide token removal. As shown in Table 1, empirical results confirm the effectiveness of the bipartite graph-based approach. On the POPE Benchmark with a 90% pruning ratio, it achieves an accuracy of 84.49%, which is close to that of the exhaustive approach (85.14%), while being nearly three times faster. Meanwhile, it retains 93% IOU agreement with the exhaustive approach, indicating that most key tokens are preserved. In contrast, the random approach is equally fast but drops to 78.63% accuracy and only 0.52 IOU, failing to preserve semantically relevant tokens.

3.3 Rethinking Query Relevance

Recent studies (Zhang et al., 2024b; Yang et al., 2025) address visual token redundancy using text-agnostic pruning strategies that retain tokens with high [`CLS`] attention scores from the vision encoder's final layer. However, these methods often fail to incorporate the query explicitly. They are also limited by inherent issues in attention mechanisms, such as the attention sink effect, where a disproportionate amount of attention is assigned to the first token in a sequence, regardless of its semantic relevance. As a result, these methods tend to preserve tokens from visually salient or high-attention regions. Meanwhile, they may discard less prominent areas, such as backgrounds or textures, which can still be crucial to answering the query.

This observation raises a critical question: *Is output-layer [CLS] attention sufficient to capture all query-relevant visual information?* Empirical observations suggest such attention-based strategies often overemphasize dominant foreground objects while neglecting background elements relevant to the query. As shown in Figure 1, attention focuses on the *pot* and *fruit* in the foreground but overlooks the actual target, the *flower* in the background. This results in incorrect outputs. Moreover, both divergence-based and similarity-based approaches also fail to adapt to the user query in this example.

Insight 4: CLS Attention Misses Query-Relevant Background Tokens

Attention-based relevance measures tend to focus on visually salient foregrounds and can miss query-relevant background tokens, underscoring the need for explicitly query-aware relevance scoring.

3.4 Rethinking Query-Relevant Visual Token Selection

Let the query tokens be $\mathbf{H}_q = \{\mathbf{h}_j^{(q)}\}$, and define their mean embedding as:

$$\mathbf{h}_\mu^{(q)} = \frac{1}{\ell} \sum_{j=1}^{\ell} \mathbf{h}_j^{(q)}. \tag{3}$$

The relevance score q_i of each visual token $\mathbf{h}_i$ is computed as the cosine similarity between $\mathbf{h}_i$ and the mean query embedding $\mathbf{h}_\mu^{(q)}$:

$$r_i = \frac{\mathbf{h}_i^\top \mathbf{h}_\mu^{(q)}}{\|\mathbf{h}_i\| \|\mathbf{h}_\mu^{(q)}\|}, \quad \text{for } i = 1, \dots, n. \tag{4}$$

We construct a diagonal matrix $\boldsymbol{Q} = \operatorname{diag}(r_1, \dots, r_n) \in \mathbb{R}^{n \times n}$ to encode the query relevance of each visual token. To encourage diversity among selected tokens, we define a similarity matrix $\boldsymbol{S}^{(v)} \in \mathbb{R}^{n \times n}$, where $\boldsymbol{S}_{ij}^{(v)}$ measures the pairwise similarity between vision tokens $\mathbf{h}_i$ and $\mathbf{h}_j$. We assume $\boldsymbol{S}^{(v)}$ is symmetric and positive semidefinite, which approximately holds when using cosine similarity among normalized features. This formulation naturally aligns with modeling token selection as a k-DPP (Determinantal Point Process), which promotes subsets that balance individual relevance with collective diversity. We therefore construct the following k-DPP kernel (note that $\boldsymbol{L}$ is a positive-semidefinite kernel, **not** a loss function) to integrate both query relevance and feature diversity:

$$\boldsymbol{L} = \boldsymbol{Q}^{1/2} \boldsymbol{S}^{(v)} \boldsymbol{Q}^{1/2}, \tag{5}$$

where $\boldsymbol{Q}$ explicitly encodes query alignment and $\boldsymbol{S}^{(v)}$ implicitly encourages feature diversity.

Insight 5: $\mathrm{S}^{(v)}$ Promotes Diversity

For any subset $\mathcal{I}$, $\det(\boldsymbol{L}_\mathcal{I})$ equals the squared volume of the parallelotope spanned by the selected token vectors (see Appendix B). If $\boldsymbol{S}^{(v)}$ were the identity matrix, this volume reduces to $\prod_{i \in \mathcal{I}} q_i$, i.e., relevance only. Introducing off-diagonal similarities lowers the determinant when two tokens are similar, thus maximizing $\det(\boldsymbol{L}_\mathcal{I})$ encourages mutually orthogonal (diverse) token subsets.

A k-DPP assigns to each subset $\mathcal{I} \subseteq [n]$ of fixed size k a probability proportional to $\det(\boldsymbol{L}_\mathcal{I})$, the determinant of the principal submatrix of $\boldsymbol{L}$ indexed by $\mathcal{I}$. This determinant reflects the volume spanned by the selected vectors, and is larger when the selected tokens are both informative and mutually distinct. Expanding the determinant yields:

$$\begin{aligned} \det(\boldsymbol{L}_\mathcal{I}) &= \det(\boldsymbol{Q}_\mathcal{I}^{1/2} \boldsymbol{S}_\mathcal{I}^{(v)} \boldsymbol{Q}_\mathcal{I}^{1/2}) \\ &= \det(\boldsymbol{Q}_\mathcal{I}^{1/2})^2 \cdot \det(\boldsymbol{S}_\mathcal{I}^{(v)}) \\ &= \left(\prod_{i \in \mathcal{I}} q_i \right) \cdot \det(\boldsymbol{S}_\mathcal{I}^{(v)}). \end{aligned} \tag{6}$$

This decomposition highlights the trade-off between query relevance (via $\prod_i q_i$) and visual diversity (via $\det(\boldsymbol{S}_\mathcal{I}^{(v)})$). Based on this, we define the final surrogate objective for selecting an informative token subset:

$$\mathcal{I}^* = \arg \max_{\mathcal{I} \subseteq [n],\, |\mathcal{I}| = m} \prod_{i \in \mathcal{I}} q_i \cdot \det(\boldsymbol{S}_\mathcal{I}^{(v)}). \tag{7}$$

Insight 6: k-DPP Balances Relevance and Diversity via Geometry

The k-DPP determinant can be analogized to the feature-space volume spanned by selected tokens, thus optimizing it jointly maximizes query relevance and inter-token diversity in a single objective.

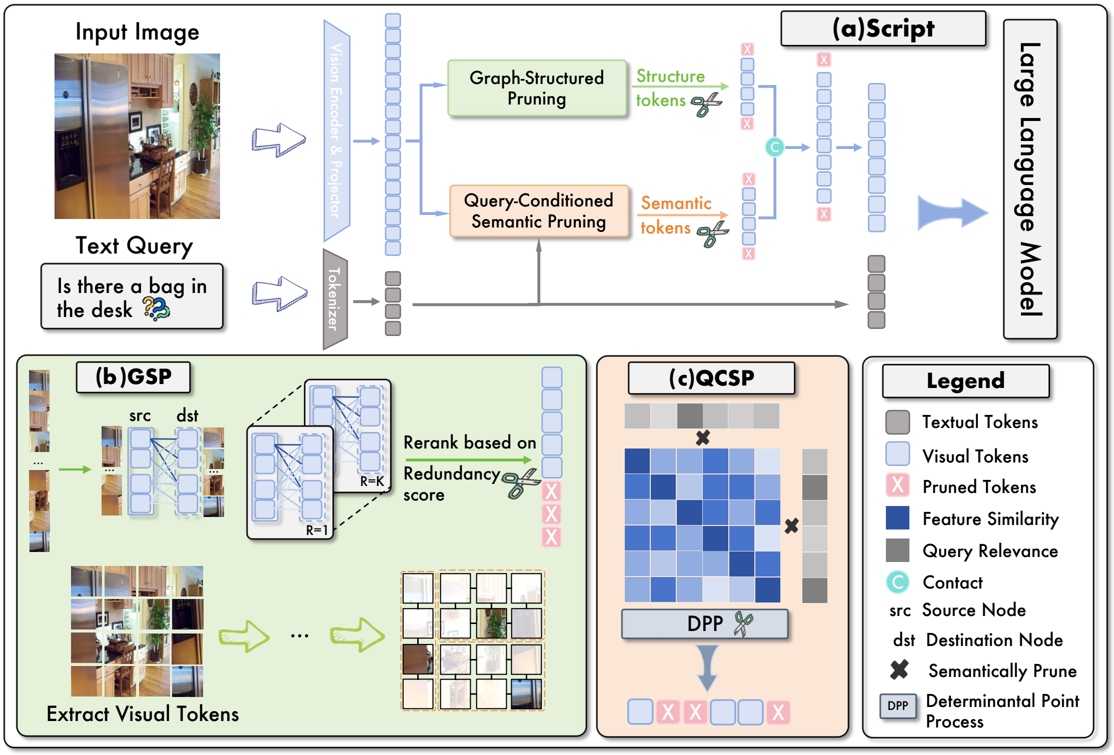


Figure 4: **Overview of Script**, a three-stage pruning framework: (a) overall architecture; (b) Query-Conditioned Semantic Pruning (QCSP); (c) Graph-Structured Pruning (GSP). Together, these modules remove semantically irrelevant and visually redundant tokens through a joint selection process.

4 Method

Motivated by the empirical analysis in Section 3, we introduce Script, as illustrated in Figure 4. Script progressively refines token selection through two main modules: (1) Graph-Structured Pruning (GSP), which removes visually redundant tokens using a bipartite similarity graph; and (2) Query-Conditioned Semantic Pruning (QCSP), which consists of two steps: Query-Conditioned Relevance Scoring (QCRS), computing the semantic relevance of each token with respect to the input query, and Diversity-Preserving Selection via DPP, selecting a compact, diverse subset of relevant tokens. The outputs from GSP and QCSP are then intersected to obtain the final token subset $\tilde{\boldsymbol{H}}_v^*$.

4.1 Graph-Structured Pruning

Transformer-based visual encoders often generate dense token sequences with substantial visual redundancy across both local and long-range contexts, as identified in Section 3.2. To mitigate this, we propose an inference-time pruning strategy based on bipartite similarity graphs without relying on model parameters.

Bipartite Graph Construction. Given visual token embeddings $\boldsymbol{H}_v = [\mathbf{h}_1, \ldots, \mathbf{h}_n] \in \mathbb{R}^{n \times d}$, we construct a bipartite graph by partitioning tokens into two disjoint node sets V_{src} and V_{dst} via alternating index assignment (e.g., even indices to V_{src}, odd-indexed to V_{dst}) (Buchholz, 2024). The bipartite graph is defined as $G = (V_{\text{src}}, V_{\text{dst}}, \boldsymbol{S}^{(v)})$, where each node in V_{src} is fully connected to every node in V_{dst}, and the edge weight $\boldsymbol{S}_{ij}^{(v)} \in \boldsymbol{S}^{(v)}$ represents the cosine similarity between tokens t_i and t_j:

$$\boldsymbol{S}_{ij}^{(v)} = \frac{\bar{\mathbf{h}}_i^{\text{src}} \cdot \bar{\mathbf{h}}_j^{\text{dst}}}{\|\bar{\mathbf{h}}_i^{\text{src}}\| \|\bar{\mathbf{h}}_j^{\text{dst}}\|}. \tag{8}$$

In this way, G explicitly structures token redundancy, with similarity scores providing insights into both local and global redundancy. Notably, this bipartite graph design reduces computational cost by up to 75% compared to conventional similarity-based pruning methods. As empirically validated in Section 3.2, most redundancy lies in local neighborhoods and is thus well-captured by the bipartite approximation effectively. Overall, V_{src} and V_{dst} jointly offer a structured, low-cost representation of $\boldsymbol{H}_v$ for redundancy modeling.

Table 2: **Performance comparisons on LLaVA-1.5-7B (Liu et al., 2023) across 10 image understanding benchmarks**. The best results in each setting are **bolded**, and the second-best are underlined.

Method	Venue	GQA	MMB	MMB$^{\text{CN}}$	MME	POPE	SQA$^{\text{IMG}}$	VQA$^{\text{V2}}$	VQA$^{\text{Text}}$	VizWiz	MMVet	Acc.	Relative
Upper Bound, 576 Tokens (100%), 3.817 TFLOPs													
LLaVA-1.5-7B (Liu et al., 2023)	*Nips'23*	61.94	64.09	58.10	1507.06	86.96	69.41	78.50	58.20	50.32	31.82	63.47	100%
Retain 192 Tokens in Average (↓ 66.7%), ∼1.253 TFLOPs													
FastV (Chen et al., 2024a)	*ECCV'24*	58.09	63.40	54.04	1490.65	82.30	68.96	72.88	54.09	54.97	29.59	61.29	96.56%
TRIM (Song et al., 2025)	*COLING'25*	58.18	59.19	51.02	1405.29	78.09	67.87	74.86	54.29	48.23	30.82	59.28	93.41%
VisionZip (Yang et al., 2025)	*CVPR'25*	59.24	63.75	53.52	1445.67	86.63	68.37	75.44	54.89	53.97	30.74	59.28	93.40%
DivPrune (Alvar et al., 2025)	*CVPR'25*	59.91	62.54	43.00	1436.39	**87.56**	69.31	75.10	53.38	**55.33**	30.96	60.89	95.94%
SparseVLM (Zhang et al., 2024d)	*ICML'25*	59.54	63.46	55.20	1450.07	86.49	68.67	75.61	56.93	51.23	31.24	62.08	97.82%
Script (Ours)	***Proposed***	**60.82**	**63.85**	**57.55**	**1493.87**	87.42	**69.49**	**77.62**	**57.81**	53.80	**31.40**	**63.45**	**100.00%**
Retain 128 Tokens in Average (↓ 77.8%), ∼0.833 TFLOPs													
FastV (Chen et al., 2024a)	*ECCV'24*	56.85	62.46	53.61	**1440.19**	80.24	68.67	71.68	52.21	55.51	28.07	60.13	94.74%
TRIM (Song et al., 2025)	*COLING'25*	56.85	58.42	48.66	1359.86	77.95	68.47	74.42	53.74	47.62	29.35	58.35	91.93%
VisionZip (Yang et al., 2025)	*CVPR'25*	57.87	62.20	53.18	1429.75	85.20	68.22	74.62	54.12	54.39	27.39	60.87	95.90%
DivPrune (Alvar et al., 2025)	*CVPR'25*	59.16	61.86	42.16	1412.97	**87.36**	**69.01**	74.19	51.92	**56.21**	28.30	60.08	94.66%
SparseVLM (Zhang et al., 2024d)	*ICML'25*	58.41	62.75	53.48	1429.41	86.22	68.62	73.86	56.29	51.83	29.12	61.21	96.43%
Script (Ours)	***Proposed***	**60.27**	**63.10**	**55.85**	1431.24	87.32	68.79	**76.55**	**56.45**	53.97	**31.09**	**62.51**	**98.49%**
Retain 64 Tokens in Average (↓ 88.9%), ∼0.415 TFLOPs													
FastV (Chen et al., 2024a)	*ECCV'24*	53.59	59.62	50.08	1366.33	75.34	68.11	58.71	51.62	54.74	27.06	56.72	89.36%
TRIM (Song et al., 2025)	*COLING'25*	56.35	55.68	46.39	1288.31	77.80	68.22	73.10	51.69	43.84	28.29	56.58	89.14%
VisionZip (Yang et al., 2025)	*CVPR'25*	55.47	60.82	51.29	1374.67	81.68	68.96	71.59	52.72	54.81	27.13	59.32	93.46%
DivPrune (Alvar et al., 2025)	*CVPR'25*	57.74	59.28	39.20	1368.28	86.51	68.22	72.35	54.51	**57.55**	27.39	59.12	93.14%
SparseVLM (Zhang et al., 2024d)	*ICML'25*	53.80	60.14	50.68	1295.16	80.92	**69.46**	68.24	53.67	50.11	24.92	57.67	90.86%
Script (Ours)	***Proposed***	**59.28**	**61.90**	**52.93**	**1412.08**	**86.95**	68.65	**75.08**	**55.20**	54.31	**29.96**	**61.49**	**96.88%**
Retain 32 Tokens in Average (↓ 94.5%), ∼0.208 TFLOPs													
FastV (Chen et al., 2024a)	*ECCV'24*	49.61	54.64	43.99	1108.35	68.59	68.26	56.48	49.85	54.22	24.50	52.56	82.80%
TRIM (Song et al., 2025)	*COLING'25*	54.35	53.31	43.27	1215.42	77.54	68.22	69.54	49.12	38.41	25.92	54.05	85.15%
VisionZip (Yang et al., 2025)	*CVPR'25*	53.32	58.85	49.31	1306.97	78.18	68.62	67.51	50.76	55.47	24.95	57.23	90.17%
DivPrune (Alvar et al., 2025)	*CVPR'25*	55.11	58.93	35.99	1303.26	83.93	68.57	69.41	51.46	**57.26**	25.73	57.16	90.05%
SparseVLM (Zhang et al., 2024d)	*ICML'25*	51.74	59.64	48.64	1181.05	77.27	69.35	63.25	51.28	49.62	23.38	55.32	87.16%
Script (Ours)	***Proposed***	**57.58**	**61.28**	**50.38**	**1338.27**	**86.77**	**68.75**	**73.25**	**53.10**	53.77	**27.57**	**59.93**	**94.42%**
Retain 16 Tokens in Average (↓ 97.3%), ∼0.103 TFLOPs													
FastV (Chen et al., 2024a)	*ECCV'24*	44.08	41.41	32.04	898.83	59.89	68.51	52.06	42.15	52.16	16.24	45.35	71.45%
TRIM (Song et al., 2025)	*COLING'25*	50.81	45.28	38.46	1085.66	81.10	66.98	61.48	39.80	35.26	20.49	49.39	77.82%
VisionZip (Yang et al., 2025)	*CVPR'25*	50.88	52.23	42.61	1179.56	75.11	66.98	62.48	45.77	54.78	20.69	53.05	83.58%
DivPrune (Alvar et al., 2025)	*CVPR'25*	51.10	53.09	31.76	1235.71	75.98	**69.21**	63.89	41.15	**55.04**	19.86	52.29	82.38%
SparseVLM (Zhang et al., 2024d)	*ICML'25*	46.16	54.16	42.93	987.32	69.00	68.26	58.72	41.78	49.47	21.84	50.17	79.04%
Script (Ours)	***Proposed***	**54.42**	**57.02**	**45.98**	**1258.55**	**84.44**	69.12	**68.77**	**47.65**	54.09	**23.71**	**56.81**	**89.51%**

Redundancy Scoring. To identify redundant tokens, we retain all similarity edges above a threshold τ, forming a pruned subgraph G_τ, where each node is connected to highly similar neighbors. For a token $t_i \in V_{\text{src}}$, we define its degree as: $d(t_i) = \sum_{j \in V_{\text{dst}}} \mathbb{I}\{\boldsymbol{S}_{ij}^{(v)} \geq \tau\}$, where $\mathbb{I}$ denotes the indicator function, returning 1 if the condition is true and 0 otherwise. Nodes with high degrees in G_τ are identified as structurally redundant, typically indicating that tokens represent repeated visual content in either local or long-range contexts. To further distinguish tokens with similar degrees, we use the average similarity $\mu(t_i)$ between token t_i and its neighbors in G_τ, which captures their typical similarity. For cases where t_i is an isolated node in G_τ, we define a fallback similarity $\tilde{\mu}(t_i)$ based on the mean similarity between t_i and all tokens in V_{dst} or V_{src} under the full graph G. The final redundancy score is then defined as:

$$\text{score}(t_i) = \begin{cases} d(t_i) \cdot \exp\left(\gamma(\mu(t_i) - \tau)\right), & d(t_i) > 0 \text{ in } G_\tau, \\ \tilde{\mu}(t_i) \text{ in } G, & \text{otherwise}, \end{cases} \tag{9}$$

where γ is a tunable scaling factor. The score, computed using $d(t_i)$ and $\mu(t_i)$ with exponential weighting, is designed to improve ranking contrast between highly and moderately redundant tokens.

Pruning Rate and Token Removal. Based on the computed redundancy scores, we rank all visual tokens in descending order and discard the top p most redundant ones, where p denotes the pruning ratio. This yields a structurally pruned candidate set $\tilde{\boldsymbol{H}}_v$, which is then combined with the token set semantically aligned with the query, produced in the next stage, to jointly form $\tilde{\boldsymbol{H}}_v^*$.

4.2 Query-Conditioned Semantic Pruning

While the GSP module eliminates visually redundant tokens, it does not guarantee semantic alignment with the user query. To address this limitation, we introduce the QCSP module, which comprises QCRS for each token and diversity-preserving selection via DPP based on the relevance scores.

Query-Conditioned Relevance Scoring. We begin by computing an average query embedding $\mathbf{h}_\mu^{(q)}$ via mean pooling over $\boldsymbol{H}_q$. Each visual token $\mathbf{h}_i$ is then scored by cosine similarity to $\mathbf{h}_\mu^{(q)}$, yielding a raw query relevance vector $\mathbf{r}_{\text{raw}} = [r_1 \dots r_n]$ (see Eq. 4). To ensure consistency across samples, we apply min-max normalization to obtain $\mathbf{r}_{\text{norm}} \in [0, 1]^n$.

Table 3: **Performance comparisons on** LLaVA-NeXT-7B **(Liu et al., 2024a) across 9 image understanding benchmarks**. The best results in each setting are **bolded**, and the second-best are underlined.

Method	**Venue**	**GQA**	**MMB**	**MMB$^{\text{CN}}$**	**MME**	**POPE**	**SQA$^{\text{IMG}}$**	**VQA$^{\text{V2}}$**	**VQA$^{\text{Text}}$**	**VizWiz**	**MMVet**	**Acc**	**Relative**
Upper Bound, 2,880 Tokens (100%), ∼20.825 TFLOPs													
LLaVA-NeXT-7B (Liu et al., 2024a)	*CVPR'24*	62.83	65.81	57.65	1504.72	86.92	67.59	81.20	60.38	55.65	41.11	65.43	100%
Retain 640 Tokens in Average (↓ 77.8%), ∼ 4.627 TFLOPs													
FastV (Chen et al., 2024a)	*ECCV'24*	58.93	63.14	53.88	1412.86	79.96	67.32	77.06	58.15	53.94	39.17	62.22	95.09%
TRIM (Song et al., 2025)	*COLING'25*	61.13	65.83	55.79	1473.62	86.92	66.71	78.36	54.85	54.12	37.16	63.46	96.98%
VisionZip (Yang et al., 2025)	*CVPR'25*	60.84	64.52	56.28	1408.67	85.86	66.36	79.13	**59.90**	55.46	38.79	63.76	97.44%
DivPrune (Alvar et al., 2025)	*CVPR'25*	60.58	65.77	57.31	1457.48	86.69	66.76	79.26	57.04	55.62	38.16	64.01	97.82%
SparseVLM (Zhang et al., 2024d)	*ICML'25*	60.32	65.66	56.81	1426.83	85.98	67.27	77.11	59.64	55.58	39.48	63.92	97.69%
Script (Ours)	***Proposed***	**62.64**	**65.98**	**57.76**	**1481.98**	**87.65**	**67.43**	**80.47**	58.92	**55.71**	**41.90**	**65.35**	**99.88%**
Retain 320 Tokens in Average (↓ 88.9%), ∼2.314 TFLOPs													
FastV (Chen et al., 2024a)	*ECCV'24*	52.17	53.87	44.51	1178.61	72.54	65.67	66.52	52.30	51.33	26.58	54.44	83.21%
TRIM (Song et al., 2025)	*COLING'25*	59.94	63.58	51.01	1426.49	86.17	66.22	74.27	51.02	53.94	32.77	61.02	93.27%
VisionZip (Yang et al., 2025)	*CVPR'25*	59.11	62.89	53.39	1360.61	84.44	65.71	76.41	**58.53**	55.27	36.25	62.00	94.76%
DivPrune (Alvar et al., 2025)	*CVPR'25*	59.24	64.03	55.58	1418.46	84.92	67.11	77.24	56.17	54.97	35.72	62.59	95.66%
SparseVLM (Zhang et al., 2024d)	*ICML'25*	58.75	64.18	54.86	1399.77	85.16	66.14	75.62	56.55	55.22	37.94	62.44	95.43%
Script (Ours)	***Proposed***	**61.36**	**65.39**	**55.82**	**1452.85**	**87.22**	**67.80**	**78.45**	57.24	**55.51**	**38.12**	**63.95**	**97.78%**
Retain 160 Tokens in Average (↓ 94.4%), ∼1.156 TFLOPs													
FastV (Chen et al., 2024a)	*ECCV'24*	48.19	47.96	39.77	1083.20	68.81	64.07	61.42	48.92	48.04	22.04	50.34	76.93%
TRIM (Song et al., 2025)	*COLING'25*	57.36	61.53	47.79	1279.44	84.58	65.51	71.04	45.94	52.92	29.82	58.05	88.71%
VisionZip (Yang et al., 2025)	*CVPR'25*	56.28	59.72	51.33	1342.84	84.06	65.89	73.49	54.32	53.68	34.59	60.05	91.78%
DivPrune (Alvar et al., 2025)	*CVPR'25*	55.62	62.97	53.51	1359.82	81.05	66.28	74.48	54.18	54.87	32.38	60.33	92.21%
SparseVLM (Zhang et al., 2024d)	*ICML'25*	53.27	62.93	53.45	1362.86	84.83	66.08	71.83	53.85	54.61	34.81	60.38	92.28%
Script (Ours)	***Proposed***	**60.73**	**64.20**	**53.67**	**1423.89**	**87.19**	**67.44**	**76.71**	**55.28**	**55.19**	**36.38**	**62.80**	**95.98%**

Query-Conditioned Kernel Construction. To support diversity-preserving selection of query-relevant tokens, we construct a query-aware DPP kernel that integrates token relevance with visual similarity. Importantly, visual similarity in this context helps refine query-relevant token selection rather than explicitly promote visual diversity, which is addressed by the GSP module.

Specifically, let $\boldsymbol{S}'^{(v)} \in \mathbb{R}^{n\times n}$ be the cosine similarity matrix of ℓ_2-normalized visual token embeddings:

$$\boldsymbol{S}'^{(v)}_{ij} = \left\langle \frac{\mathbf{h}_i}{\|\mathbf{h}_i\|}, \frac{\mathbf{h}_j}{\|\mathbf{h}_j\|} \right\rangle. \tag{10}$$

Then, we define a query-conditioned kernel $\tilde{\boldsymbol{L}} \in \mathbb{R}^{n\times n}$ by reweighting the pairwise visual similarity matrix $\boldsymbol{S}'^{(v)}$ with normalized token-to-query relevance scores $\mathbf{r}_{\text{norm}}$: $\tilde{\boldsymbol{L}} = \text{diag}(\mathbf{r}_{\text{norm}}) \cdot \boldsymbol{S}'^{(v)} \cdot \text{diag}(\mathbf{r}_{\text{norm}})$. Intuitively, $\tilde{\boldsymbol{L}}$ captures diversity within token relevance, promoting token selections that are both query-aligned and diverse. Its symmetric and positive semi-definite structure makes it well-suited for k-DPP-based selection.

Greedy Token Selection via Kernel Decomposition To select a compact yet informative subset of k visual tokens, we perform approximate MAP (maximum a posteriori) inference under a k-DPP defined by the query-conditioned kernel $\tilde{\boldsymbol{L}}$. This procedure allows us to find tokens that are not only individually relevant to the query but also collectively diverse in semantic content, reducing redundancy while ensuring broad query coverage. The objective is to select a subset S of k tokens such that the corresponding submatrix $\tilde{\boldsymbol{L}}_S$ maximizes $\det(\tilde{\boldsymbol{L}}_S)$, $\max_{|S|=k} \det\left(\tilde{\boldsymbol{L}}_S\right)$. The determinant reflects both semantic relevance and diversity. A higher value indicates greater diversity in the selected subset.

To efficiently approximate the MAP solution[1] for DPP, we adopt the Cholesky-based greedy algorithm (Chen et al., 2018). The algorithm starts with an empty set and incrementally adds tokens. At each step, it selects the token that yields the highest gain, where the gain is defined as the increase in the determinant of the kernel submatrix after adding the token. The algorithm is initialized as follows: $v_i^2 = \tilde{\boldsymbol{L}}_{ii}, \quad \mathbf{u}_i = \mathbf{0} \in \mathbb{R}^k \quad$ for all $i \in \{1, \ldots, n\}$, where v_i^2 represents the importance of token i for selection, and $\mathbf{u}_i$ is a vector that stores the projection of token i onto the subspace spanned by the previously selected tokens.

At each iteration, the token with the largest v^2 is selected: $j = \arg\max_{i \in Z \setminus S} v_i^2$. After the selection, for each $i \notin (S \cup \{j\})$, we compute:

$$e_i = \frac{\tilde{\boldsymbol{L}}_{ji} - \langle \mathbf{u}_j, \mathbf{u}_i \rangle}{\sqrt{v_j^2 + \epsilon}}, \quad u_i \leftarrow u_i + e_i \mathbf{e}_j, \quad v_i^2 \leftarrow v_i^2 - e_i^2, \tag{11}$$

where e_i represents the projection of token i onto the direction defined by token j, $\mathbf{e}_j$ is the j-th standard basis vector in $\mathbb{R}^n$, and ϵ is a small positive constant (e.g., 10^{-6}) introduced for numerical stability. The

[1] MAP for DPP is an NP-hard problem.

Table 4: **Performance comparison of different pruning methods on Video-LLaVA-7B with 64 frames per video.** The best results in each setting are **bolded**, and the second-best are underlined.

Method	**Venue**	**MLVU**	**MVBench**	**LongVideoBench**			**Video-MME**				**Acc.**	**Relative**
Metric		**m-avg**	**test**	**val**	perception	relation	**w/o sub**	short	medium	long		
Upper Bound, All 64 × 169 Tokens (**100%**)												
LLaVA-Video-7B	-	67.75	58.15	59.07	65.03	53.64	63.66	76.46	61.21	53.19	62.19	100%
Retain 64 × 64 Tokens (↓ **62.1%**)												
FastV Chen et al. (2024a)	*ECCV'24*	63.79	55.81	56.21	60.60	52.15	61.89	73.46	59.45	**52.71**	59.56	95.77%
DivPrune Alvar et al. (2025)	*CVPR'25*	64.01	54.82	57.97	**64.32**	**53.75**	61.31	72.49	59.43	51.02	59.90	96.32%
SparseVLM Zhang et al. (2024d)	*ICML'25*	65.33	56.68	56.30	61.20	51.57	60.94	73.20	58.78	51.12	59.45	95.59%
Script (Ours)	***Purpose***	**66.13**	**57.74**	**58.27**	64.25	53.37	**62.36**	**74.67**	**60.31**	52.59	**61.08**	**98.22%**
Retain 64 × 32 Tokens (↓ **81.1%**)												
FastV Chen et al. (2024a)	*ECCV'24*	58.75	52.27	52.64	56.88	48.25	56.00	63.38	55.79	48.34	54.70	87.95%
DivPrune Alvar et al. (2025)	*CVPR'25*	61.44	53.37	**56.64**	**62.41**	51.34	59.73	69.69	57.82	49.88	58.04	93.32%
SparseVLM Zhang et al. (2024d)	*ICML'25*	60.37	54.13	53.47	58.51	49.79	59.09	69.18	56.49	50.38	56.82	91.37%
Script (Ours)	***Purpose***	**62.77**	**55.37**	56.35	61.40	**52.97**	**60.35**	**72.11**	**58.46**	**51.20**	**58.99**	**95.00%**
Retain 64 × 16 Tokens (↓ **90.5%**)												
FastV Chen et al. (2024a)	*ECCV'24*	52.58	46.57	46.61	48.68	44.79	49.88	54.92	50.30	45.28	48.85	78.54%
DivPrune Alvar et al. (2025)	*CVPR'25*	58.60	51.13	52.27	**57.56**	47.29	56.37	**67.67**	53.34	48.11	54.72	88.98%
SparseVLM Zhang et al. (2024d)	*ICML'25*	51.97	48.87	47.42	53.02	42.89	49.58	53.81	49.45	46.22	49.25	79.19%
Script (Ours)	***Purpose***	**58.77**	**53.45**	**52.91**	57.13	**48.66**	**57.20**	65.97	**56.08**	**49.46**	**55.52**	**89.30%**

update $v_i^2 \leftarrow v_i^2 - e_i^2$ encourages selecting tokens with small projections e_i. A smaller e_i indicates that the token is less similar, that is, closer to being orthogonal to the already selected tokens, thereby enhancing the diversity of the subset.

4.3 Final Token Selection

GSP and QCSP each address complementary aspects of token selection. GSP explicitly targets visual redundancy, while QCSP captures query relevance. However, neither is sufficient on its own for robust selection. We therefore take their intersection to ensure that the final subset is both visually compact and semantically aligned with the query. If the intersection contains fewer tokens than the required number, we supplement it with additional tokens from QCSP by iteratively increasing k to retrieve more tokens.

5 Experiments

5.1 Experimental Settings

We conduct comprehensive experiments to evaluate Script's performance across diverse MLLMs and benchmarks[2]. In all tables, the "Relative" column reports performance relative to the unpruned model.

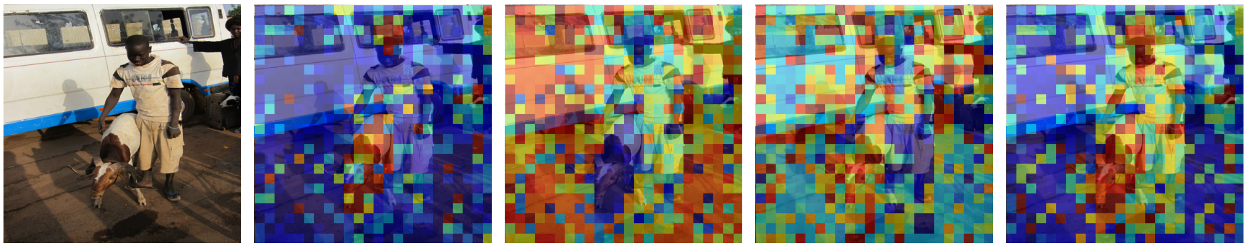
Models and Baselines. We select four representative MLLMs: LLaVA 1.5 7B (Liu et al., 2023), LLaVA NeXT 7B (Liu et al., 2024a), Video LLaVA 7B (Lin et al., 2024), and Intern VL3 (Zhu et al., 2025). These models cover both image and video modalities with diverse architectures, providing a comprehensive testbed for evaluating Script's generalizability. In addition, we compare with five state-of-the-art pruning baselines, including FastV (Chen et al., 2024a), TRIM (Song et al., 2025), SparseVLM (Zhang et al., 2024d), DivPrune (Alvar et al., 2025), and VisionZip (Yang et al., 2025). Notably, most of these baselines are either query-agnostic or tightly bound to specific model architectures. Some of these architectures are included in our evaluation, enabling fair and direct comparisons.

Benchmarks. We evaluate Script across fourteen widely adopted benchmarks spanning both image and video understanding tasks. **Image understanding tasks:** GQA (Hudson & Manning, 2019), ScienceQA (Lu et al., 2022), VQAv2 (Goyal et al., 2017), TextVQA (Singh et al., 2019b), VizWiz (Gurari et al., 2018), MMVet (Yu et al., 2023), MMBench (Liu et al., 2024c), MMBench-CN (Liu et al., 2024c), MME (Fu et al., 2023), and POPE (Li et al., 2023b). **Video reasoning tasks:** LongVideoBench (Wu et al., 2024), VideoMME (Fu et al., 2025), MLVU (Zhou et al., 2025), and MVBench (Li et al., 2023a).

5.2 Results and Discussions

LLaVA-1.5-7B. Table 2 compares visual token pruning methods on LLaVA-1.5-7B across different token budgets. Our proposed method, Script, consistently outperforms baselines such as FastV, DivPrune, and

[2] See the appendix for additional experiments, model settings, limitations, and broader impact.



Input Image Attention-based Divergence-based Similarity-based DPP-based

Figure 5: **Visualizations of Pruning Preferences.** We compute four different pruning scores for a sample from the POPE benchmark using LLaVA-1.5-7B, with the query: *"Is there a person in the image?"* **Red** indicates high preference, while **blue** indicates low.

SparseVLM, demonstrating strong adaptability and robustness across pruning levels and diverse tasks. At the highest token budget (192 tokens), Script achieves perfect accuracy (100.00%), outperforming the next-best method, DivPrune, by over 4 percentage points (95.94%). When reducing tokens to an intermediate level (64 tokens), Script retains exceptional accuracy at 96.86%, indicating a minimal performance drop despite significant token reduction. This outperforms SparseVLM by around 6 percentage points, underscoring Script's effective token selection strategy. Even under the most aggressive pruning (16 tokens, corresponding to 97.2% pruning), Script still retains high accuracy (89.51%), significantly outperforming all baselines, highlighting its strong ability to preserve essential semantic and visual information, making it suitable for resource-constrained deployment.

LLaVA-NeXT-7B. Table 3 presents a detailed comparison of visual token pruning methods applied to LLaVA-NeXT-7B across different token budgets. Our method, Script, consistently delivers the best performance across pruning levels, outperforming all baseline methods. Under mild pruning (640 tokens, 77.8% reduction), Script achieves the highest average accuracy (65.1%) and the best relative performance (99.88%), indicating minimal information loss despite significant pruning. Under moderate pruning conditions (320 tokens, 88.9% reduction), Script maintains robust performance, achieving 63.97% accuracy and 97.09% relative performance, outperforming the next-best baseline by 2.12% points. Even at the most aggressive pruning level (160 tokens, 94.4% reduction), Script still maintains superior results, reaching 62.80% accuracy and 95.98% relative performance. These results demonstrate Script's strong ability to retain essential visual and semantic cues under tight computational budgets.

Video-LLaVA-7B. Table 4 reports the performance of SOTA pruning methods on Video-LLaVA-7B across multiple token budgets. Script consistently outperforms all baselines across pruning levels, demonstrating robust performance under growing compression ratios. Under mild pruning (64×64 tokens, 62.1% reduction), Script achieves 98.22% average accuracy, demonstrating effective preservation of informative video content essential for accurate multimodal reasoning. Furthermore, under moderate pruning (64×32 tokens, 81.1% reduction), Script maintains 95.00% accuracy, outperforming FastV by roughly 7 percentage points, which demonstrates its effectiveness even under tighter token constraints. Even under aggressive pruning (64×16 tokens, 90.5% reduction), Script retains strong performance with 89.30% average accuracy, significantly surpassing other baselines at similar compression levels. These results highlight Script's ability to preserve spatio-temporal semantics, supporting its applicability in real-world video-based multimodal tasks.

5.3 Case study

To analyze how different pruning strategies handle query-relevant vision tokens, we compare four approaches: attention-based (via [`CLS`]), divergence-based, similarity-based, and DPP-based. Figure 5 visualizes the normalized scores from each method on a representative POPE benchmark. The scores reveal how each method prioritizes tokens based on distinct selection criteria.

As shown in Figure 5, the four strategies exhibit clearly different token selection behaviors. The attention-based method emphasizes globally salient regions, such as central objects, but often overlooks contextually relevant backgrounds like human bodies. In contrast, divergence-based and similarity-based methods struggle to prioritize tokens that align with the input query. The similarity-based method, in particular, ignores query information and produces relatively uniform token scores that primarily reflect visual diversity. While such diversity helps preserve overall image content, it fails to capture query-specific relevance. The DPP-based method assigns the highest relevance to human faces and moderately high scores to body-related tokens, demonstrating strong alignment with query semantics. However, similar to attention-based approaches, it lacks diversity in the selected tokens. These observations highlight the importance of combining GSP and QCSP for more balanced and effective token pruning.

5.4 Ablation study

To analyze the individual impact of each module, we conduct an ablation study to evaluate the contributions of the GSP, QCRS, and DPP modules. All experiments are conducted under stringent constraints, retaining only 16 tokens on average (97.3% pruning), with a compute cost of 0.103 TFLOPs. As shown in Table 5, naively pruning tokens causes a sharp drop in relative accuracy (average: 63.73%), underscoring the need for informed token selection strategies. Incorporating GSP alone raises relative accuracy to 66.58%, demonstrating the benefit of using a bipartite graph to model redundancy. Integrating QCSP (QCRS + DPP) alone further improves performance, increasing relative accuracy to 91.19%. Notably, the full configuration achieves 92.19% of the original accuracy while retaining only 2.7% of tokens. These results further reflect the distinct yet complementary contributions of each module.

Table 5: Ablation on LLava-1.5-7B evaluating the impact of GSP, QCRS, and DPP.

GSP	QCRS	DPP	POPE	MME	GQA	SQA^{IMG}	Acc.	Relative
LLaVA-1.5-7B, Retain 576 Tokens (100%)								
			86.96	1507.06	61.94	69.41	73.41	100%
Retain 16 Tokens in Average (↓ 97.3%), ∼0.103 TFLOPs								
✘	✘	✘	49.92	695.70	38.69	63.76	46.78	63.73%
✔	✘	✘	52.85	758.23	40.86	63.86	48.87	66.58%
✘	✔	✘	77.20	1225.27	45.82	68.67	63.23	86.14%
✔	✔	✘	77.75	1111.92	51.11	68.57	63.25	86.17%
✘	✔	✔	83.38	1252.28	53.28	68.51	66.95	91.19%
✔	✔	✔	**84.44**	**1258.55**	**54.42**	**69.12**	**67.73**	**92.19%**

5.5 Efficiency Analysis

We comprehensively evaluate recent token pruning methods on the POPE benchmark, focusing on computation cost, inference latency, memory usage, and accuracy. As shown in Table 6, the uncompressed LLaVA-NeXT-7B requires 41.7T FLOPs, with 246ms prefill and 29ms decode latency per token. It consumes 1440MB KV cache and 16.7GB peak GPU memory, achieving an F1 score of 86.8. In contrast, Script reduces FLOPs to 4.1T (10× reduction), prefill latency to 35ms (6.8× faster), and decode latency to 22ms. It also lowers KV cache to 160MB and memory to 13.5GB, while maintaining an F1 score of 86.7. Without DPP, Script maintains similar efficiency and achieves a slightly lower F1 score of 86.5. Compared to other baselines, Script achieves the lowest prefill/decode latency, the smallest KV cache, and the best overall F1 score. For instance, FastV and PDrop suffer major accuracy loss (F1: 49.5 and 60.8), while VisionZip and DivPrune perform better (F1: 82.3 and 84.7) but at higher computational and memory cost. Overall, Script achieves the best balance between efficiency and accuracy, outperforming all baselines across nearly all metrics.

Table 6: **Efficiency comparisons on the POPE benchmark**. We report the theoretical FLOPs, actual runtime, KV cache compression rate (%), and the achieved accuracy.

Method	# Token	FLOPs (T)	Prefill Time (ms/token)	Decode Time (ms/token)	KV Cache (MB)	GPU Memory (GB)	F1 Score (F1)
LLaVA-NeXT-7B	2880	41.7	246	29	1440.0	16.7	86.8
FastV(ECCV24)	320	4.4 (×9.5)	54 (×4.6)	23 (×1.2)	160.3	15.6	49.5
PDrop(CVPR25)	320	4.5 (×9.3)	55 (×4.5)	24 (×1.2)	160.2	15.6	60.8
SparseVLM(ICML25)	320	4.5 (×9.3)	71 (×3.5)	25 (×1.1)	161.2	18.6	76.9
VisionZip(CVPR25)	320	4.2 (×9.9)	38 (×6.6)	**22 (×1.3)**	**160.0**	14.8	82.3
DivPrune(CVPR25)	320	4.2 (×9.9)	38 (×6.6)	**22 (×1.3)**	**160.0**	13.9	84.7
Script(Ours)	320	**4.1 (×10)**	**35 (×6.8)**	**22 (×1.3)**	**160.0**	**13.5**	**86.7**
w/o DPP	320	**4.1 (×10)**	38 (×6.6)	**22 (×1.3)**	**160.0**	14.0	86.5

6 Conclusion

In this paper, we introduce **Script** which reduces inference cost by selecting a compact yet semantically meaningful subset of visual tokens, conditioned on the user query. Script combines visual redundancy and query relevance perspectives through four stages: (1) graph-structured filtering to remove visually redundant tokens, (2) query-conditioned relevance scoring, (3) diversity-promoting subset selection based on relevance scores using DPP, and (4) intersection-based fusion to retain tokens that are both visually compact and semantically aligned with the query. Its architecture-agnostic, plug-and-play design enables seamless integration into existing MLLMs without requiring retraining or architectural changes. Extensive experiments on fourteen vision-language benchmarks show that Script consistently reduces computational overhead and latency while preserving, or even improving, task performance under aggressive pruning settings. Future work includes extending Script to additional modalities such as audio and video in spatial-temporal contexts, and exploring adaptive pruning schedules that dynamically adjust to input complexity.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikoł aj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23716–23736, 2022.

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.

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Federico Barbero, Alvaro Arroyo, Xiangming Gu, Christos Perivolaropoulos, Michael Bronstein, Petar Veličković, and Razvan Pascanu. Why do llms attend to the first token? *arXiv preprint arXiv:2504.02732*, 2025.

Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. Token merging: Your ViT but faster. In *International Conference on Learning Representations*, 2023.

Simon Buchholz. Learning partitions from context. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL `https://openreview.net/forum?id=pRSgf5VdD0`.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, et al. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*, 2024.

Jun Chen, Deyao Zhu, Xiaoqian Shen, Xiang Li, Zechun Liu, Pengchuan Zhang, Raghuraman Krishnamoorthi, Vikas Chandra, Yunyang Xiong, and Mohamed Elhoseiny. Minigpt-v2: large language model as a unified interface for vision-language multi-task learning. *arXiv preprint arXiv:2310.09478*, 2023.

Laming Chen, Guoxin Zhang, and Eric Zhou. Fast greedy map inference for determinantal point process to improve recommendation diversity. *Advances in Neural Information Processing Systems*, 31, 2018.

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, 2024a.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, et al. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*, 2024b.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024c.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024d.

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. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality, March 2023. URL `https://lmsys.org/blog/2023-03-30-vicuna/`.

Tri Dao. FlashAttention-2: Faster attention with better parallelism and work partitioning. In *International Conference on Learning Representations (ICLR)*, 2024.

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. Mme: A comprehensive evaluation benchmark for multimodal large language models. *arXiv preprint arXiv:2306.13394*, 2023.

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

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*, pp. 6904–6913, 2017.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, et al. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14375–14385, 2024.

Zonghao Guo, Ruyi Xu, Yuan Yao, Junbo Cui, Zanlin Ni, Chunjiang Ge, Tat-Seng Chua, Zhiyuan Liu, and Gao Huang. Llava-uhd: an lmm perceiving any aspect ratio and high-resolution images. In *European Conference on Computer Vision*, pp. 390–406. Springer, 2024.

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

Wenbo Hu, Zi-Yi Dou, Liunian Li, Amita Kamath, Nanyun Peng, and Kai-Wei Chang. Matryoshka query transformer for large vision-language models. *Advances in Neural Information Processing Systems*, 37: 50168–50188, 2024.

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*, pp. 6700–6709, 2019.

Ahmadreza Jeddi, Negin Baghbanzadeh, Elham Dolatabadi, and Babak Taati. Similarity-aware token pruning: Your vlm but faster. *arXiv preprint arXiv:2503.11549*, 2025.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pp. 235–251. Springer, 2016.

Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Haija, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from https://github.com/openimages*, 2017.

Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International journal of computer vision*, 123:32–73, 2017.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024a.

Kunchang Li, Yali Wang, Yinan He, Yizhuo Li, Yi Wang, Yi Liu, Zun Wang, Jilan Xu, Guo Chen, Ping Luo, Limin Wang, and Yu Qiao. Mvbench: A comprehensive multi-modal video understanding benchmark. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22195–22206, 2023a. URL `https://api.semanticscholar.org/CorpusID:265466214`.

Yadong Li, Jun Liu, Tao Zhang, Song Chen, Tianpeng Li, Zehuan Li, Lijun Liu, Lingfeng Ming, Guosheng Dong, Da Pan, et al. Baichuan-omni-1.5 technical report. *arXiv preprint arXiv:2501.15368*, 2025a.

Yanwei Li, Chengyao Wang, and Jiaya Jia. Llama-vid: An image is worth 2 tokens in large language models. In *European Conference on Computer Vision*, pp. 323–340. Springer, 2024b.

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

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. Prompt compression for large language models: A survey. In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7182–7195, Albuquerque, New Mexico, April 2025b. Association for Computational Linguistics. ISBN 979-8-89176-189-6. URL `https://aclanthology.org/2025.naacl-long.368/`.

Bin Lin, Yang Ye, Bin Zhu, Jiaxi Cui, Munan Ning, Peng Jin, and Li Yuan. Video-llava: Learning united visual representation by alignment before projection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 5971–5984, 2024.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755. Springer, 2014.

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

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024a. URL `https://llava-vl.github.io/blog/2024-01-30-llava-next/`.

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*, 2024b.

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, 2024c.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. Ocrbench: on the hidden mystery of ocr in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024d.

Zuyan Liu, Yuhao Dong, Jiahui Wang, Ziwei Liu, Winston Hu, Jiwen Lu, and Yongming Rao. Ola: Pushing the frontiers of omni-modal language model with progressive modality alignment. *arXiv preprint arXiv:2502.04328*, 2025.

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. *Advances in Neural Information Processing Systems*, 35:2507–2521, 2022.

Gen Luo, Yiyi Zhou, Yuxin Zhang, Xiawu Zheng, Xiaoshuai Sun, and Rongrong Ji. Feast your eyes: Mixture-of-resolution adaptation for multimodal large language models. *arXiv preprint arXiv:2403.03003*, 2024.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022.

Lorenzo Papa, Paolo Russo, Irene Amerini, and Luping Zhou. A survey on efficient vision transformers: Algorithms, techniques, and performance benchmarking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46:7682–7700, 2023. URL `https://api.semanticscholar.org/CorpusID:261531260`.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PmLR, 2021.

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

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, 2019a.

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, 2019b.

Dingjie Song, Wenjun Wang, Shunian Chen, Xidong Wang, Michael X. Guan, and Benyou Wang. Less is more: A simple yet effective token reduction method for efficient multi-modal LLMs. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 7614–7623, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL `https://aclanthology.org/2025.coling-main.508/`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Hongjie Wang, Bhishma Dedhia, and Niraj K. Jha. Zero-tprune: Zero-shot token pruning through leveraging of the attention graph in pre-trained transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

Zichen Wen, Yifeng Gao, Weijia Li, Conghui He, and Linfeng Zhang. Token pruning in multimodal large language models: Are we solving the right problem? *arXiv preprint arXiv:2502.11501*, 2025a.

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. *arXiv preprint arXiv:2502.11494*, 2025b.

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

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, et al. Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

Jin Xu, Zhifang Guo, Jinzheng He, Hangrui Hu, Ting He, Shuai Bai, Keqin Chen, Jialin Wang, Yang Fan, Kai Dang, Bin Zhang, Xiong Wang, Yunfei Chu, and Junyang Lin. Qwen2.5-omni technical report. *arXiv preprint arXiv:2503.20215*, 2025.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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*, 2025.

Weihao Ye, Qiong Wu, Wenhao Lin, and Yiyi Zhou. 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, pp. 22128–22136, 2025.

Tianyu Yu, Haoye Zhang, Yuan Yao, Yunkai Dang, Da Chen, Xiaoman Lu, Ganqu Cui, Taiwen He, Zhiyuan Liu, Tat-Seng Chua, et al. Rlaif-v: Aligning mllms through open-source ai feedback for super gpt-4v trustworthiness. *arXiv preprint arXiv:2405.17220*, 2024.

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

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Hang Zhang, Xin Li, and Lidong Bing. Video-llama: An instruction-tuned audio-visual language model for video understanding. *arXiv preprint arXiv:2306.02858*, 2023.

Peiyuan Zhang, Kaichen Zhang, Bo Li, Guangtao Zeng, Jingkang Yang, Yuanhan Zhang, Ziyue Wang, Haoran Tan, Chunyuan Li, and Ziwei Liu. Long context transfer from language to vision. *arXiv preprint arXiv:2406.16852*, 2024a.

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*, 2024b.

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*, 2024c.

Shaolei Zhang, Qingkai Fang, Zhe Yang, and Yang Feng. Llava-mini: Efficient image and video large multimodal models with one vision token. *arXiv preprint arXiv:2501.03895*, 2025.

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. *arXiv preprint arXiv:2410.04417*, 2024d.

Yuanhan Zhang, Jinming Wu, Wei Li, Bo Li, Zejun Ma, Ziwei Liu, and Chunyuan Li. Video instruction tuning with synthetic data. *arXiv preprint arXiv:2410.02713*, 2024e.

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

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL `https://openreview.net/forum?id=1tZbq88f27`.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Script: Graph-Structured and Query-Conditioned Semantic Token Pruning for Multimodal Large Language Models

Appendix

A Notation Overview

To facilitate clarity and improve readability, we provide a detailed glossary of all mathematical symbols used throughout this paper, as shown in Table 7. The table serves as a centralized reference point, systematically organizing every notation involved in the core components of the Script framework, including equations, algorithmic modules, and architectural definitions. The glossary covers all key components of the proposed framework, including visual encoding, query representation, relevance scoring, similarity computation, DPP-based token selection, structural pruning, and auxiliary constants. This table is intended to improve the clarity of mathematical derivations, reduce cross-referencing overhead, and support accurate reproduction and further development of the method.

Table 7: Detailed symbol glossary used throughout the *Script* framework.

Symbol	**Type / Dim**	**Definition**
$\boldsymbol{X_v}$	Image / video input	Raw visual input passed into the vision encoder f_v.
f_v	Function	Vision encoder extracting patch-level features from $\boldsymbol{X_v}$.
g	Function	Multimodal projector mapping visual features to token embeddings.
n, d, ℓ, m	Integers	#tokens (input, dim, query length, retained) respectively.
$\boldsymbol{H}_v$	Matrix	Sequence of n visual tokens, each d-dimensional, before pruning.
$\boldsymbol{H}_q$	Sequence of vectors	Tokenized user query of length ℓ.
$\tilde{\boldsymbol{H}}_v$	Matrix ($m \times d$)	Visual token subset after pruning ($m < n$).
f_φ	Function (LLM)	Frozen language model consuming $[\tilde{\boldsymbol{H}}_v; \boldsymbol{H_q}]$.
$L(\cdot, \cdot)$	Scalar loss	Task loss comparing model outputs before and after pruning.
$\mathbf{h}_i$	Vector ($\in \mathbb{R}^d$)	i-th visual token embedding in $\boldsymbol{H_v}$.
$\mathbf{h}_\mu^{(q)}$	Vector ($\in \mathbb{R}^d$)	Mean pooled embedding of the query tokens.
q_i	Scalar	Query-conditioned relevance score of token i.
$\boldsymbol{Q} = \mathrm{diag}(q_1, \ldots, q_n)$	Diagonal matrix	Relevance weighting matrix for visual tokens.
S_{ij}	Scalar	Cosine similarity between tokens i and j.
$\boldsymbol{S}$	Matrix ($n \times n$)	Symmetric similarity matrix among visual tokens.
$\boldsymbol{L} = \boldsymbol{Q}^{1/2}\boldsymbol{S}\boldsymbol{Q}^{1/2}$	Matrix ($n \times n$)	DPP kernel combining relevance and diversity.
$\mathcal{I},\ \lvert\mathcal{I}\rvert = m$	Index set	Token indices selected for retention.
$\mathcal{I}^*$	Index set	Optimal index subset maximizing log-det of DPP kernel.
$\tilde{\boldsymbol{L}}$	Matrix	Query-conditioned DPP kernel with temperature scaling.
$\mathbf{r}, \tilde{\mathbf{r}}$	Vectors ($\in [0,1]^n$)	Raw and temperature-scaled relevance scores.
$V_{\mathrm{src}}, V_{\mathrm{dst}}$	Token subsets	Bipartite partitions used in graph-based redundancy pruning.
$G = (V_{\mathrm{src}}, V_{\mathrm{dst}}, \boldsymbol{S})$	Graph	Token similarity graph for structural filtering.
d_i^2	Scalar	Residual variance of token i under low-rank projection.
u_i	Vector ($\in \mathbb{R}^T$)	Coefficient vector for token i in basis space.
e_i	Scalar	Normalizer in Cholesky-style update.
$\boldsymbol{C} \in \mathbb{R}^n \times T$	Matrix	Basis matrix for low-rank approximation of DPP kernel.
T	Integer	Target rank or DPP subset size.
τ	Threshold (scalar)	Redundancy threshold for similarity pruning.
ϵ	Constant (10^-6)	Numerical stability term in projection update.
$\odot$	Operator	Element-wise (Hadamard) product.
$\langle \cdot, \cdot \rangle$	Operator	Inner product used for cosine similarity.
$\lVert \cdot \rVert$	Operator	Euclidean norm used in normalization.

B Determinantal Diversity: Equivalence, Bounds, and Optimality

B.1 Notation.

To facilitate clear and consistent exposition, we first define the key symbols and assumptions used throughout the paper and illustrate each with a simple example or explanation.

- $\|\cdot\|_2$ **(Euclidean norm):** The standard ℓ_2 norm for vectors in $\mathbb{R}^d$.
  Example: $\|(3,4)\|_2 = \sqrt{3^2+4^2} = 5$.
- $\lambda_i(A)$ **(i-th largest eigenvalue of A):** For a symmetric matrix $A \in \mathbb{R}^{k\times k}$, we order its eigenvalues as $\lambda_1(A) \geq \lambda_2(A) \geq \cdots \geq \lambda_k(A)$. Since all our kernels will be positive semi-definite (PSD), all $\lambda_i \geq 0$.
- $\mathbf{1}_k$ **(all-ones vector):** A k-dimensional column vector with all entries equal to one, i.e., $(1,\dots,1)^\top \in \mathbb{R}^k$.

Let $\mathcal{T} = v_1,\dots,v_n \subset \mathbb{R}^d$ denote a collection of feature vectors. We assume that each v_i has unit norm: $\|v_i\|_2 = 1$ for all i. This means that each vector lies on the unit hypersphere in $\mathbb{R}^d$.

Example: The vector $v_i = (1,0,\dots,0)^\top$ has $\|v_i\|_2 = 1$, and thus is a valid member of $\mathcal{T}$.

We adopt this unit-norm convention to ensure that inner products $v_i^\top v_j$ correspond exactly to cosine similarities, simplifying the interpretation of the similarity kernel introduced next. Meanwhile, this normalization remains fixed throughout the paper.

Similarity Kernel. Given a collection of unit-norm vectors $\mathcal{T} = [v_1,\dots,v_n] \subset \mathbb{R}^d$ as introduced above, we define a similarity kernel matrix $L \in \mathbb{R}^{n\times n}$ whose (i,j)-entry captures the cosine similarity between v_i and v_j:

$$L_{ij} = v_i^\top v_j, \qquad \text{for all } 1 \leq i,j \leq n. \tag{12}$$

Since all v_i lie on the unit hypersphere, we have $v_i^\top v_j \in [-1,1]$. In particular, $L_{ii} = v_i^\top v_i = 1$, and the more similar two vectors are, the closer L_{ij} is to 1.

We can express the full kernel compactly using matrix notation. Let $V = [v_1 \ v_2 \ \cdots \ v_n] \in \mathbb{R}^{d\times n}$ be the matrix whose columns are the feature vectors.

Then:

$$L = V^\top V. \tag{13}$$

This representation immediately implies that L is symmetric and positive semi-definite (PSD), because for any $x \in \mathbb{R}^n$, we have:

$$x^\top L x = x^\top V^\top V x = \|Vx\|_2^2 \geq 0. \tag{14}$$

Therefore, L satisfies the standard spectral properties of Gram matrices: all eigenvalues of L are real and non-negative, and L admits an orthonormal eigendecomposition. These properties are critical for the DPP model discussed next.

B.2 Determinant as a Diversity Objective

We now connect the determinant of a kernel submatrix with geometric diversity. Specifically, we show that for any subset of k unit vectors, the determinant of their Gram matrix equals the square of the volume of the parallelotope they span.

Proposition 1 (Determinant–Volume Equivalence)**.** *Let $S = [i_1,\dots,i_k] \subseteq [n]$ be an index subset of size k. Define $V_S = [v_{i_1},\dots,v_{i_k}] \in \mathbb{R}^{d\times k}$ as the submatrix of feature vectors indexed by S, and $L_S = V_S^\top V_S \in \mathbb{R}^{k\times k}$ as their Gram matrix. Then,*

$$\det L_S = \big(\det V_S\big)^2 = \big(\mathrm{Vol}_k(V_S)\big)^2, \tag{15}$$

where $\mathrm{Vol}_k(V_S)$ is the k-dimensional volume of the parallelotope spanned by the column vectors of V_S.

Proof. Let $V_S = QR$ be the QR decomposition of V_S, where $Q \in \mathbb{R}^{d\times k}$ has orthonormal columns and $R \in \mathbb{R}^{k\times k}$ is upper triangular. Then,

$$L_S = V_S^\top V_S = R^\top Q^\top QR = R^\top R, \tag{16}$$

using the fact that $Q^\top Q = I_k$. Taking determinants on both sides gives:

$$\det L_S = \det(R^\top R) = (\det R)^2. \tag{17}$$

Moreover, from $V_S = QR$, we have $\det V_S = \det R$ because Q has orthonormal columns and thus $\det Q$ is ± 1 (or undefined if $d < k$). Therefore,

$$\det L_S = (\det V_S)^2. \tag{18}$$

Geometrically, $|\det V_S|$ gives the volume of the k-dimensional parallelotope formed by the columns of V_S (a known fact from multilinear algebra). Thus,

$$\mathrm{Vol}_k(V_S) = |\det V_S|, \quad \text{so} \quad \det L_S = (\mathrm{Vol}_k(V_S))^2. \tag{19}$$

□

This equivalence forms the foundation for interpreting $\det L_S$ as a natural diversity objective: it assigns higher values to sets of vectors that are more 'spread out' or 'orthogonal' in space.

B.3 Diversity Upper Bound via Hadamard Inequality

The determinant of any Gram matrix L_S formed from unit-norm vectors is upper bounded by 1. This follows from the classical Hadamard inequality, which gives a constraint on the determinant of a positive semi-definite matrix in terms of its diagonal entries.

Corollary 1 (Hadamard Bound)**.** *Let L_S be the Gram matrix defined above, formed from k unit vectors. Then:*

$$\det L_S \le 1, \qquad \det L_S = 1 \iff v_{i_p}^\top v_{i_q} = 0 \textit{ for all } p \ne q. \tag{20}$$

Proof. Hadamard's inequality states that for any positive semi-definite matrix $A \in \mathbb{R}^{k\times k}$ with diagonal entries a_{ii},

$$\det A \le a_{11}a_{22}\cdots a_{kk},$$

with equality if and only if the columns of A are orthogonal.

In our case, $L_S = V_S^\top V_S$ is PSD and satisfies $\mathrm{diag}(L_S) = (1, 1, \ldots, 1)$ since each $\|v_i\|_2 = 1$.

Therefore,

$$\det L_S \le 1, \tag{21}$$

with equality if and only if the vectors $v_{i_1}, \ldots, v_{i_k}$ are mutually orthogonal. □

The determinant $\det L_S$ quantifies how linearly independent or "spread out" the vectors in S are. The maximum value 1 occurs when the vectors are exactly orthogonal. Therefore, maximizing $\det L_S$ promotes diversity by encouraging the selection of vectors that are as close to orthogonal as possible.

B.4 Determinantal Point Processes and Diversity Maximization

We now formally introduce Determinantal Point Processes (DPPs), which are probability distributions over subsets that inherently promote diversity. In the fixed-size or k-DPP setting, each subset S of size k is assigned a probability proportional to the determinant of its associated submatrix L_S:

$$\mathbb{P}_L(S) \propto \det L_S, \qquad \text{for } |S| = k. \tag{22}$$

Specifically, we formulate this as:

$$\mathbb{P}_L(S) = \frac{\det L_S}{\sum\limits_{\substack{T \subseteq [n] \\ |T| = k}} \det L_T}. \tag{23}$$

where the denominator sums over all k-subsets of $[n]$, ensuring a valid probability distribution.

This probabilistic model encourages the selection of sets S whose vectors are geometrically diverse, since higher determinant values correspond to higher volume (as shown previously). In practice, one often seeks the *most probable* subset under this model, known as the maximum a posteriori (MAP) estimate:

$$S_{\text{MAP}} = \arg\max_{|S|=k} \det L_S. \tag{24}$$

This optimization problem aims to find the subset of k vectors that span the largest volume parallelotope in $\mathbb{R}^d$, or equivalently, the subset exhibiting maximal geometric diversity. The DPP framework thus provides both a probabilistic model and a concrete objective—$\det L_S$—for achieving diverse subset selection.

B.5 Redundancy Metrics and Determinant Bounds

While the determinant $\det L_S$ captures diversity, we can also assess subset quality via *redundancy* metrics. These measure how similar the vectors in S are to each other, either in the worst-case or on average. Two commonly used metrics are:

$$\rho_{\max}(S) = \max_{i \neq j \in S} L_{ij}, \qquad \rho_{\text{avg}}(S) = \frac{2}{k(k-1)} \sum_{i<j \in S} L_{ij}. \tag{25}$$

Here, $L_{ij} = v_i^\top v_j$ is the cosine similarity between unit vectors v_i and v_j. The value $\rho_{\max}(S)$ measures the most redundant pair (i.e., most similar), while $\rho_{\text{avg}}(S)$ captures the overall similarity level.

We now relate these redundancy measures to $\det L_S$ using spectral bounds. The first bound is derived using Gershgorin's circle theorem.

Lemma 1 (Gershgorin-Based Lower Bound). *For any subset S of size k, we have:*

$$\lambda_{\min}(L_S) \geq 1 - \rho_{\max}(S), \qquad \det L_S \geq (1 - \rho_{\max}(S))^k. \tag{26}$$

Proof. Each row of L_S has diagonal entry 1, and all off-diagonal entries satisfy $|L_{ij}| \leq \rho_{\max}(S)$ for $i \neq j$. By Gershgorin's circle theorem, every eigenvalue of L_S lies in at least one disk centered at 1 with radius $\rho_{\max}(S)$. Hence:

$$\lambda_{\min}(L_S) \geq 1 - \rho_{\max}(S).$$

Since L_S is symmetric, all its eigenvalues are real. Therefore, Gershgorin's circle theorem applies directly and guarantees that all eigenvalues lie within disks centered at the diagonal entries with radius equal to the sum of the absolute off-diagonal entries in each row. Therefore, L_S is symmetric PSD, its determinant is the product of eigenvalues:

$$\det L_S = \prod_{i=1}^{k} \lambda_i(L_S) \geq \lambda_{\min}(L_S)^k \geq (1 - \rho_{\max}(S))^k.$$

□

Next, we upper bound $\det L_S$ in terms of average similarity, using the arithmetic mean-geometric mean inequality.

Lemma 2 (AM–GM Upper Bound). *Let $\lambda_1, \ldots, \lambda_k$ be the eigenvalues of L_S. Then:*

$$\det L_S \leq \left(1 + \rho_{\text{avg}}(S)\right)^{k-1}. \tag{27}$$

This lemma provides a simple upper bound on the determinant of a Gram matrix in terms of average pairwise similarity. While useful for theoretical analysis, this AM–GM-based inequality is often loose in practice, especially when $\rho_{\text{avg}}(S)$ is non-negligible. The bound does not account for the detailed spectral structure of L_S, and thus can significantly overestimate the true value of $\det L_S$. To address this, we next derive a refined upper bound that is spectrally informed and provably tighter.

Proof. By the AM–GM inequality, for any non-negative real numbers $\lambda_1, \ldots, \lambda_k$, we have:

$$\det L_S = \prod_{i=1}^{k} \lambda_i \leq \left(\frac{1}{k} \sum_{i=1}^{k} \lambda_i \right)^k = \left(\frac{\operatorname{tr}(L_S)}{k} \right)^k = 1^k = 1.$$

To relate this to $\rho_{\text{avg}}(S)$, we consider the Rayleigh quotient of the all-ones vector $\mathbf{1}_k$:

$$\frac{\mathbf{1}_k^\top L_S \mathbf{1}_k}{\mathbf{1}_k^\top \mathbf{1}_k} = \frac{\sum_{i,j \in S} L_{ij}}{k^2} = \frac{k + k(k-1) \cdot \rho_{\text{avg}}(S)}{k^2} = 1 + \frac{k-1}{k} \rho_{\text{avg}}(S).$$

This quantity upper bounds the largest eigenvalue $\lambda_{\max}(L_S)$ since the Rayleigh quotient provides an upper bound over all unit vectors.

Now, to obtain an upper bound on the determinant, we consider a worst-case distribution of eigenvalues under the trace constraint. Specifically, we assume that one eigenvalue is large and the remaining $(k-1)$ are equal and small. This skewed distribution is known to maximize the determinant under fixed trace and maximum eigenvalue constraints.

Let $\lambda_{\max}$ be the largest eigenvalue, and assume the remaining $(k-1)$ eigenvalues are equal to λ_{rest}. Since the total trace is k, we have:

$$\lambda_{\max} + (k-1)\lambda_{\text{rest}} = k.$$

Solving for λ_{rest}, we get:

$$\lambda_{\text{rest}} = \frac{k - \lambda_{\max}}{k-1}.$$

The determinant is then:

$$\det L_S = \lambda_{\max} \cdot \left(\frac{k - \lambda_{\max}}{k-1} \right)^{k-1}.$$

Substituting the upper bound on $\lambda_{\max}$ from above:

$$\lambda_{\max} \leq 1 + \frac{k-1}{k} \rho_{\text{avg}}(S),$$

and simplifying this expression yields the claimed upper bound:

$$\det L_S \leq (1 + \rho_{\text{avg}}(S))^{k-1}.$$

This bound is general but often loose, especially when the off-diagonal similarities in L_S are uniformly high. In such cases, we next show that a refined bound based on structured matrices can be significantly tighter. □

Motivation for a Refined Bound. The AM–GM bound above provides a worst-case envelope that does not incorporate any spectral structure of L_S beyond its average similarity. However, when all off-diagonal entries of L_S are equal to ρ_{avg}, the matrix exhibits an extremal spectral configuration that yields the smallest determinant under fixed average similarity. By explicitly analyzing this case, we obtain a refined upper bound that better captures the worst-case decay of $\det L_S$ with increasing redundancy.

Lemma 3 (Refined Upper Bound via Spectral Construction)**.** *Let L_S be the Gram matrix of k unit vectors with average off-diagonal similarity $\rho_{\text{avg}}(S) \in [0, 1)$. Then,*

$$\det L_S \leq (1 + (k-1)\rho_{\text{avg}}(S)) \cdot (1 - \rho_{\text{avg}}(S))^{k-1}. \tag{28}$$

Proof. Consider the class of symmetric PSD matrices L_S where all off-diagonal entries are equal to ρ_{avg}, and all diagonal entries are 1:

$$(L_S)_{ij} = \begin{cases} 1 & \text{if } i = j \\ \rho_{\text{avg}} & \text{if } i \neq j. \end{cases}$$

This matrix has a known spectral decomposition: one eigenvalue equals $1 + (k-1)\rho_{\text{avg}}$, and the remaining $(k-1)$ eigenvalues are $1 - \rho_{\text{avg}}$.

Hence, the determinant is:

$$\det L_S = (1 + (k-1)\rho_{\text{avg}}) \cdot (1 - \rho_{\text{avg}})^{k-1}.$$

By Schur-convexity of the determinant function over the eigenvalue simplex, this configuration gives the maximal determinant among all matrices with the same average off-diagonal similarity. Thus, the bound holds. □

While this refined bound is not a universal upper bound over all positive semi-definite matrices with a given average off-diagonal similarity, it tightly characterizes the worst-case configuration when all off-diagonal entries are equal. Among such uniformly redundant matrices, the bound is achieved exactly, making it a sharp envelope for assessing diversity degradation under homogeneously correlated subsets.

Compared to Lemma 2, this refined bound is strictly tighter whenever $\rho_{\text{avg}}(S) > 0$, as it directly models the spectral behavior of uniformly redundant configurations. The bound thus provides a sharper envelope for analyzing diversity degradation in highly correlated subsets.

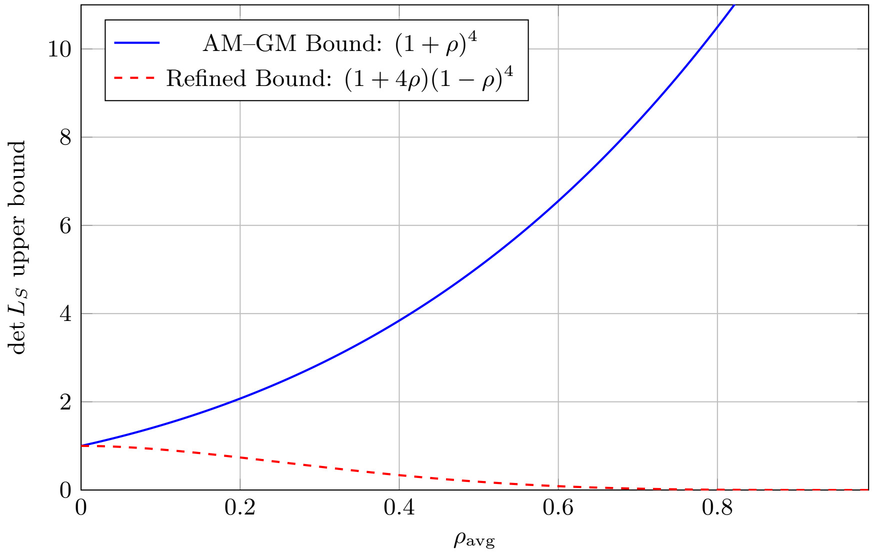


Figure 6: Upper bounds on $\det L_S$ vs. average similarity ρ_{avg} for $k = 5$. The refined bound tracks the worst-case determinant decay more tightly than the AM–GM bound.

B.6 Global Optimality of the MAP Subset

We now show that the subset S^* that maximizes $\det L_S$ among all k-subsets also minimizes both redundancy metrics $\rho_{\max}$ and ρ_{avg}. This establishes that the most diverse subset under the DPP determinant objective is simultaneously the least redundant.

Theorem 1 (Determinant-maximizing subset minimizes redundancy)**.** *Let* $S^* = \arg\max_{|S|=k} \det L_S$ *be the MAP solution of the k-DPP. Then for any other k-subset S, we have:*

$$\rho_{\max}(S^*) \leq \rho_{\max}(S), \qquad \rho_{\text{avg}}(S^*) \leq \rho_{\text{avg}}(S). \tag{29}$$

Proof. (i) *Worst-case similarity.* From Lemma 1, we have for any subset S:

$$\rho_{\max}(S) \geq 1 - \det(L_S)^{1/k}, \tag{30}$$

which follows by rearranging Eq. equation 26. Since $x \mapsto 1 - x^{1/k}$ is strictly decreasing on $(0, 1]$, it attains its minimum when $\det L_S$ is maximized. Hence:

$$\rho_{\max}(S^*) \leq \rho_{\max}(S).$$

(ii) *Average similarity.* Similarly, from Lemma 2, we rearrange Eq. equation 27:

$$\rho_{\text{avg}}(S) \geq \det(L_S)^{1/(k-1)} - 1. \tag{31}$$

Again, this is a strictly decreasing function of $\det L_S$, so the maximizing subset S^* achieves minimal average redundancy:

$$\rho_{\text{avg}}(S^*) \leq \rho_{\text{avg}}(S).$$

□

Corollary 2 (MAP subset is globally most diverse)**.** *The k-DPP MAP subset* S^* *simultaneously maximizes geometric diversity (via* $\det L_S$*) and minimizes both* $\rho_{\max}$ *and* ρ_{avg} *among all size-k subsets.*

B.7 Approximation via Greedy MAP Inference

Exact MAP inference for a k-DPP is NP-hard, as it requires evaluating $\binom{n}{k}$ determinants. However, the objective $\log \det L_S$ is known to be monotone and submodular when L is positive semi-definite, which allows for a natural greedy approximation algorithm. The function $\log \det L_S$ is known to be monotone and submodular when L is PSD, as it arises from the entropy of Gaussian variables or from spectral submodular theory (see [ref]).

Let $\widehat{S}$ denote the greedy solution obtained by iteratively selecting vectors that offer the greatest marginal increase in $\log \det L_S$. Then, by the classical Nemhauser–Wolsey theorem for submodular maximization:

$$\log \det L_{\widehat{S}} \geq (1 - 1/e)\, \log \det L_{S^*}, \tag{32}$$

where S^* is the exact MAP subset from Eq. equation 24.

Exponentiating both sides gives a multiplicative guarantee on the determinant:

$$\det L_{\widehat{S}} \geq (\det L_{S^*})^{1-1/e}. \tag{33}$$

We now translate this guarantee into approximate bounds on redundancy. Applying Lemma 1, we get:

$$\rho_{\max}(\widehat{S}) \leq 1 - (\det L_{\widehat{S}})^{1/k} \leq 1 - (\det L_{S^*})^{(1-1/e)/k}. \tag{34}$$

Similarly, applying Lemma 2 in reverse:

$$\rho_{\text{avg}}(\widehat{S}) \leq (\det L_{\widehat{S}})^{1/(k-1)} - 1 \leq (\det L_{S^*})^{(1-1/e)/(k-1)} - 1. \tag{35}$$

Hence, the greedy subset $\widehat{S}$ enjoys constant-factor approximation guarantees on both diversity and redundancy metrics, with approximation factors of $(\det L_{S^*})^{1-1/e}$ for diversity, and redundancy bounds that degrade gracefully with $1/k$ and $1/(k-1)$ exponents.

B.8 Summary.

This section has established a formal connection between geometric diversity, as quantified by the determinant of a Gram submatrix, and redundancy metrics such as $\rho_{\max}$ and ρ_{avg}. We showed that:

- Maximizing $\det L_S$ encourages the selection of nearly orthogonal vectors, which minimizes both redundancy measures.
- The MAP solution to the k-DPP yields the globally most diverse and least redundant subset.
- Even when exact MAP inference is intractable, greedy algorithms yield approximate solutions with provable guarantees, not only on $\det L_S$, but also on redundancy.

Together, these results justify the determinant objective as a principled and effective diversity metric, both theoretically and algorithmically.

Application to Script. The diversity metric and DPP framework established in this section provide the theoretical foundation for the Script method proposed in the main text. While Script operates in a practical, query-conditioned multimodal setting, its core selection strategy is grounded in the same principle of maximizing the determinant of a positive semi-definite kernel submatrix.

In Script, the DPP kernel is constructed as:

$$L = \operatorname{diag}(r) \cdot S \cdot \operatorname{diag}(r), \tag{36}$$

where S is a symmetric matrix capturing token-token similarity (e.g., cosine similarity), and r contains query-conditioned relevance scores. Since S is symmetric PSD and $\operatorname{diag}(r)$ is diagonal with non-negative entries, the resulting matrix L remains PSD by congruence transformation. This formulation modulates pairwise similarities by semantic alignment with the user query, but retains the PSD structure necessary for DPP-based selection.

The subset selection objective in Script thus becomes:

$$S^* = \arg\max_{|S|=k} \det(L_S), \tag{37}$$

which is a direct instance of the MAP inference problem for k-DPPs. As shown earlier, maximizing $\det(L_S)$ promotes geometric diversity and minimizes redundancy. Script extends this by incorporating semantic relevance via r, enabling informed token pruning aligned with both diversity and task relevance.

This shows that even when the kernel matrix is modulated by non-uniform relevance scores—as encoded by $\operatorname{diag}(r)$—the determinant-based objective retains its ability to balance geometric diversity with task-conditioned selection. Such a formulation is particularly well-suited for transformer compression and attention summarization, where preserving non-redundant yet relevant tokens is essential.

Therefore, the theoretical results in this section not only justify the determinant-based diversity objective used in Script, but also explain its effectiveness in minimizing redundancy and preserving essential information under aggressive token reduction.

Empirically, we find that maximizing $\det(L_S)$ with query-conditioned L enables Script to select token subsets that are both semantically relevant and minimally redundant, yielding improved performance in summarization and retrieval tasks (see Section 5).

C Details of experimental setup

C.1 Model Architectures

LLaVA-1.5 (Liu et al., 2023) The LLaVA series represents a foundational line of open-source vision-language models (VLMs), recognized for their simple design, low training cost, and strong performance. The original LLaVA architecture integrates a pretrained CLIP (Radford et al., 2021) as the visual encoder and Vicuna (Chiang et al., 2023) as the language model, connected via a linear projection layer. This enables the LLM to accept image grid features as input. Through visual instruction tuning, LLaVA gains the ability to handle multimodal tasks. LLaVA-1.5 enhances this framework by replacing the linear connector with a multi-layer perceptron (MLP), increasing input image resolution, and utilizing a broader and more diverse set of instruction tuning data. These modifications lead to substantial performance improvements. The model processes images at a resolution of 336×336, resulting in 576 visual tokens per image.

LLaVA-NeXT (Liu et al., 2024a) LLaVA-NeXT (also referred to as LLaVA-1.6) builds upon LLaVA-1.5 by introducing a dynamic resolution mechanism aimed at improving visual perception. Instead of using a fixed resolution, the model selects an optimal aspect ratio based on the original image and increases its resolution by up to 4×. Importantly, the visual encoder remains unchanged. To handle the higher-resolution inputs, the image is divided into multiple sub-images of the original size. Each sub-image is encoded independently, and the resulting visual tokens are concatenated and passed to the language model. This approach enhances the model's performance in tasks such as visual reasoning, optical character recognition (OCR), and knowledge-intensive questions. For consistency and fair comparison, we fix the resolution to 672×672 (4× the original), generating 2,880 visual tokens per image.

LLaVA-Video (Zhang et al., 2024e) LLaVA-Video is a variant of the LLaVA family designed specifically for video understanding. It introduces the SlowFast frame sampling strategy to balance the number of frames and the density of visual tokens. The model utilizes SigLIP (Zhai et al., 2023) as the visual encoder and processes video frames at 384×384 resolution, encoding each frame into 729 visual tokens. To reduce computational load, a 2×2 average pooling operation is applied to the grid features, effectively reducing the number of visual tokens by a factor of 4. During evaluation, we uniformly sample 64 frames per video, resulting in a total of 10,816 visual tokens. This design allows LLaVA-Video to efficiently model both spatial and temporal aspects of visual input.

InternVL3 (Zhu et al., 2025) One of the most advanced open-source MLLMs at present. Building upon its predecessor, InternVL2.5, it retains the ViT-MLP-LLM architecture, integrating a Vision Transformer with a large language model through an MLP connector. InternVL3 features a native multimodal pre-training paradigm, jointly acquiring linguistic and multimodal capabilities in a single stage. It incorporates Variable Visual Position Encoding to handle extended multimodal contexts and employs advanced training techniques like supervised fine-tuning and mixed preference optimization. InternVL3 demonstrates superior performance across a wide range of multimodal tasks, including tool usage, GUI agents, industrial image analysis, and 3D vision perception.

C.2 Evaluation Benchmarks

C.2.1 General Image Benchmarks

VQAv2 (Goyal et al., 2017) An open-ended visual question answering benchmark that evaluates a model's ability to understand images, natural language, and commonsense knowledge. It contains 265,016 images from the COCO dataset (Lin et al., 2014) and abstract scenes, with each image paired with an average of 5.4 questions. Each question is annotated with 10 ground truth answers and 3 plausible alternatives. We use the test-dev split for evaluation.

GQA (Hudson & Manning, 2019) A large-scale VQA benchmark built on real-world images from the Visual Genome dataset (Krishna et al., 2017), designed to test compositional reasoning and fine-grained visual understanding. It provides over 22 million balanced question-answer pairs, with each image accompanied by a detailed scene graph describing objects, attributes, and relationships. We evaluate on the test-dev balanced split.

VizWiz (Gurari et al., 2018) A real-world VQA benchmark created from images taken by blind users, paired with spoken questions and 10 crowd-annotated answers each. It introduces two main challenges: answering questions and detecting unanswerable ones, due to issues like poor image quality and ambiguous content. We use the test split for evaluation.

ScienceQA (Lu et al., 2022) A multimodal, multiple-choice QA benchmark covering diverse scientific domains. It includes 21,208 questions categorized across 26 topics, 127 categories, and 379 skills. Nearly half of the questions include image or text context, and the majority are supplemented with grounded lectures (83.9%) and detailed explanations (90.5%). We evaluate using the test split that includes image context, referred to as ScienceQA-IMG.

POPE (Li et al., 2023b) A benchmark focused on evaluating object hallucination in vision-language models. Using images from COCO, it formulates binary questions regarding the presence of specific objects in the scene. Precision, recall, and F1 score are used to quantify hallucination. We use the test split for evaluation.

MME (Fu et al., 2023) A broad benchmark assessing both perceptual and cognitive abilities of multimodal large language models. It comprises 14 subtasks across perception (e.g., object recognition, counting, color, position, OCR) and cognition (e.g., commonsense reasoning, numerical calculation, translation, code understanding). All binary instruction-answer pairs are manually constructed to avoid data leakage, ensuring rigorous evaluation.

MMBench (Liu et al., 2024c) A comprehensive benchmark designed to evaluate a wide range of multimodal capabilities. It features a large and diverse set of questions, surpassing prior benchmarks in scale and coverage. A novel CircularEval strategy, powered by ChatGPT, converts open-ended responses into structured formats for consistent scoring. A Chinese version, MMBench-CN, is also provided.

MM-Vet (Yu et al., 2023) A benchmark emphasizing the integration of multimodal skills. It defines six core capabilities, recognition, OCR, knowledge, language generation, spatial reasoning, and mathematics, via 218 challenging examples. Evaluation is conducted using ChatGPT to ensure consistency across varied answer formats.

HallusionBench (Guan et al., 2024) An image-context reasoning benchmark crafted to expose two frequent failure modes of large vision–language models: language hallucination (answers driven by strong linguistic priors that contradict the image) and visual illusion (misleading visual features that produce confident yet wrong responses). Comprising carefully designed examples that remain challenging for GPT-4V and LLaVA-1.5, it enables fine-grained diagnosis of how VLMs over-trust language or under-exploit vision, offering insights for building more faithfully grounded models.

C.2.2 Text-Oriented Benchmarks

TextVQA (Singh et al., 2019a) A benchmark designed to evaluate models' ability to read and reason about text embedded in images. Sourced from the Open Images v3 dataset (Krasin et al., 2017), it includes scenes rich in textual content such as signs and packaging. The benchmark emphasizes integration of OCR with visual and linguistic reasoning. We use the validation split for evaluation.

AI2D (Kembhavi et al., 2016) A diagram-based question answering benchmark consisting of over 5,000 grade school science diagrams, annotated with more than 150,000 structured labels and ground-truth syntactic parses. It also includes over 15,000 multiple-choice questions aligned with the diagrams, enabling research on visual reasoning and diagram understanding in scientific contexts. We use the test split with mask for evaluation.

ChartQA (Masry et al., 2022) A large-scale benchmark designed for question answering over charts, focusing on complex reasoning that involves both visual interpretation and logical or arithmetic operations. It includes 9.6K human-written questions and 23.1K questions generated from chart summaries. Unlike prior template-based benchmarks, ChartQA challenges models to perform multi-step reasoning using both the visual content and underlying data tables of charts, highlighting the need for advanced multimodal understanding. We use the test split for evaluation.

OCRBench (Liu et al., 2024d) A comprehensive evaluation benchmark assessing the OCR capabilities of large multimodal models. It comprises 29 datasets across diverse text-related visual tasks, including text recognition, scene text-centric VQA, document-oriented VQA, key information extraction, and handwritten mathematical expression recognition.

Table 8: **Performance comparisons on LLaVA-1.5-13B Liu et al. (2024a) across 10 image understanding benchmarks**.

Method	Venue	**VQA$^{\text{V2}}$**	**GQA**	**VizWiz**	**SQA$^{\text{IMG}}$**	**VQA$^{\text{Text}}$**	**POPE**	**MME**	**MMB$^{\text{EN}}$**	**MMB$^{\text{CN}}$**	**MMVet**	**Acc.**	**Average**
Upper Bound, 576 Tokens (100%), 3.817 TFLOPs													
LLaVA-1.5-13B Liu et al. (2023)	*Nips'23*	80.10	63.35	53.36	72.85	61.24	86.00	1531.25	68.55	63.55	36.22	66.21	100%
Retain 128 Tokens in Average (↓ 77.8%), ∼0.833 TFLOPs													
FastV Chen et al. (2024a)	*ECCV'24*	75.53	58.13	54.46	74.32	58.76	75.55	1460.56	66.01	62.23	32.78	63.17	95.41%
TRIM Song et al. (2025)	*COLING'25*	76.34	59.14	49.87	72.14	55.08	86.38	1426.49	67.11	58.14	35.13	63.06	95.24%
VisionZip Yang et al. (2025)	*CVPR'25*	76.83	57.79	52.03	73.68	58.39	82.77	1449.42	67.14	62.95	36.01	64.01	96.67%
DivPrune Alvar et al. (2025)	*CVPR'25*	77.41	59.21	53.35	72.28	58.40	86.81	1457.97	66.13	60.74	34.41	64.16	96.90%
SparseVLM Zhang et al. (2024d)	*ICML'25*	77.61	59.46	51.74	74.23	59.93	85.02	1487.59	68.14	62.36	35.92	64.87	97.97%
Script (Ours)	***Proposed***	77.87	59.27	52.49	73.29	58.45	87.31	1498.30	67.15	61.35	36.12	**64.83**	**97.64%**
Retain 64 Tokens in Average (↓ 88.9%), ∼0.415 TFLOPs													
FastV Chen et al. (2024a)	*ECCV'24*	65.73	51.39	53.48	73.01	53.74	56.49	1246.54	59.12	55.18	26.89	55.78	84.24%
TRIM Song et al. (2025)	*COLING'25*	73.12	57.85	49.23	72.00	52.10	86.65	1406.12	65.04	52.57	27.98	60.73	91.72%
VisionZip Yang et al. (2025)	*CVPR'25*	73.75	56.23	53.12	74.29	57.41	75.67	1379.66	64.94	61.33	33.40	61.91	93.51%
DivPrune Alvar et al. (2025)	*CVPR'25*	75.20	57.91	54.49	71.67	57.54	84.35	1454.29	64.13	59.87	29.31	62.73	94.74%
SparseVLM Zhang et al. (2024d)	*ICML'25*	73.12	55.91	52.17	72.09	57.14	77.91	1374.35	65.12	60.23	32.94	61.62	93.07%
Script (Ours)	***Proposed***	76.67	59.64	53.55	72.75	57.86	87.19	1466.98	65.85	58.72	36.20	**64.20**	**96.96%**
Retain 32 Tokens in Average (↓ 94.5%), ∼0.208 TFLOPs													
FastV Chen et al. (2024a)	*ECCV'24*	61.13	48.36	51.68	72.37	50.73	53.99	1198.33	54.27	53.22	23.65	52.93	79.95%
TRIM Song et al. (2025)	*COLING'25*	69.83	55.67	48.58	70.64	49.65	85.85	1284.57	63.13	45.45	26.34	57.79	86.28%
VisionZip Yang et al. (2025)	*CVPR'25*	68.40	52.71	53.09	72.79	55.20	66.85	1257.67	61.32	55.58	29.43	57.81	87.62%
DivPrune Alvar et al. (2025)	*CVPR'25*	72.00	56.20	54.55	70.89	54.76	79.12	1405.02	61.47	57.12	27.98	60.42	91.25%
SparseVLM Zhang et al. (2024d)	*ICML'25*	71.57	54.05	51.54	70.86	53.74	77.45	1327.37	62.88	58.91	28.13	59.55	89.94%
Script (Ours)	***Proposed***	75.25	58.75	53.35	71.99	55.43	87.31	1421.10	63.79	56.36	30.77	**62.41**	**94.26%**

C.2.3 Video Benchmarks

MLVU (Zhou et al., 2025) The first large-scale benchmark for long video understanding. It features videos ranging from 3 minutes to 2 hours and spans nine tasks covering holistic, single-detail, and multi-detail understanding. Both multiple-choice and open-ended questions are included. We report performance using the M-Avg metric.

MVBench (Li et al., 2023a) A benchmark tailored for evaluating temporal reasoning in video comprehension. It includes 20 carefully designed tasks requiring dynamic understanding across multiple frames. MVBench introduces a static-to-dynamic transformation approach to systematically test temporal understanding. We use the test split for evaluation.

LongVideoBench (Wu et al., 2024) A large-scale benchmark for evaluating understanding of long-form videos. It consists of 3,763 videos (up to 1 hour long), each accompanied by subtitles and 6,678 human-written multiple-choice questions across 17 categories. A key feature is the referring reasoning task, where questions target specific video segments. We use the validation split for evaluation.

Video-MME (Fu et al., 2025) A comprehensive video benchmark featuring 900 expert-curated videos spanning 256 hours across six primary domains and 30 subfields. Videos range from 11 seconds to 1 hour in duration and include video, audio, and subtitles (not used during evaluation). It provides 2,700 expert-annotated QA pairs designed to probe complex temporal and multimodal reasoning abilities.

C.3 Comparison Methods

C.3.1 Text-based Methods

FastV (Chen et al., 2024a) The first work to identify inefficiencies in visual attention within MLLMs. Based on this observation, FastV proposes a simple, training-free acceleration method: after the second transformer layer, it prunes a portion of visual tokens with the lowest visual-text attention scores. This strategy significantly reduces computational cost during inference without retraining.

SparseVLM (Zhang et al., 2024d) Inspired by multi-stage pruning methods such as PyramidDrop, SparseVLM introduces a more fine-grained strategy that incorporates textual guidance. It observes that not all instruction tokens are equally informative for pruning visual tokens. Therefore, it first selects text tokens highly relevant to the visual content as “raters," and uses their attention distribution to guide which visual tokens should be preserved or pruned, resulting in improved model efficiency and accuracy.

Table 9: **Performance comparisons on LLaVA-Next-13B Liu et al. (2024a) across 8 image understanding benchmarks**. The best results in each setting are **bolded**, and the second-best are underlined.

Method	Venue	**VQA^{V2}**	**GQA**	**VizWiz**	**SQA^{IMG}**	**VQA^{Text}**	**POPE**	**MME**	**MMB^{EN}**	**MMB^{CN}**	**MMVet**	**Acc.**	**Average**
Upper Bound, 576 Tokens (100%), 3.817 TFLOPs													
LLaVA-NeXT-13B Liu et al. (2024a)	*Nips'23*	82.30	64.35	59.15	73.15	63.20	85.20	1539.50	68.55	61.05	45.02	67.91	100%
Retain 640 Tokens in Average (↓ 77.8%), ∼4.627 TFLOPs													
FastV Chen et al. (2024a)	*ECCV'24*	79.34	60.89	56.44	71.75	60.74	80.24	1536.7	65.25	59.39	43.68	65.45	96.38%
TRIM Song et al. (2025)	*COLING'25*	79.34	63.19	54.31	71.42	57.64	87.31	1543.36	68.75	61.21	42.34	66.26	97.57%
VisionZip Yang et al. (2025)	*CVPR'25*	79.37	62.79	56.12	70.83	61.91	85.83	1529.22	68.31	62.61	46.98	67.13	98.84%
DivPrune Alvar et al. (2025)	*CVPR'25*	80.34	63.54	56.73	72.12	59.42	86.44	1531.41	67.51	62.59	39.10	66.42	97.81%
SparseVLM Zhang et al. (2024d)	*ICML'25*	79.49	62.74	57.55	72.35	62.48	85.56	1573.74	68.85	64.10	41.35	67.31	99.11%
Script (Ours)	***Proposed***	81.14	64.22	57.02	72.08	61.40	87.31	1552.61	68.93	61.91	47.13	**67.87**	**99.95%**
Retain 320 Tokens in Average (↓ 88.9%), ∼2.314 TFLOPs													
FastV Chen et al. (2024a)	*ECCV'24*	66.98	54.36	53.33	70.11	55.34	64.10	1288.0	59.48	54.14	30.52	57.27	84.34%
TRIM Song et al. (2025)	*COLING'25*	75.94	61.13	52.12	69.59	52.18	87.42	1484.6	67.53	57.34	33.41	63.09	92.90%
VisionZip Yang et al. (2025)	*CVPR'25*	76.18	60.37	54.84	70.12	60.17	82.33	1417.13	66.45	62.53	41.21	64.50	94.96%
DivPrune Alvar et al. (2025)	*CVPR'25*	78.13	61.28	55.10	72.39	57.46	85.32	1465.01	65.91	61.59	39.42	64.98	95.68%
SparseVLM Zhang et al. (2024d)	*ICML'25*	76.75	60.91	54.75	70.94	60.01	81.35	1491.46	68.10	63.45	39.36	65.05	95.78%
Script (Ours)	***Proposed***	79.64	63.11	55.31	71.46	58.74	87.63	1501.15	66.73	61.48	42.32	**66.16**	**97.42%**
Retain 160 Tokens in Average (↓ 94.4%), ∼1.156 TFLOPs													
FastV Chen et al. (2024a)	*ECCV'24*	62.67	50.79	52.03	69.11	51.74	62.95	1229.18	56.48	53.16	27.36	54.77	80.66%
TRIM Song et al. (2025)	*COLING'25*	72.15	58.93	51.12	69.19	49.12	87.00	1392.30	65.75	51.65	27.68	60.21	88.66%
VisionZip Yang et al. (2025)	*CVPR'25*	72.45	57.81	52.35	69.47	58.69	76.38	1393.49	64.85	60.01	35.39	61.70	90.86%
DivPrune Alvar et al. (2025)	*CVPR'25*	75.64	60.30	53.15	71.44	56.43	81.29	1436.17	65.19	60.59	37.34	63.31	93.22%
SparseVLM Zhang et al. (2024d)	*ICML'25*	74.62	59.79	52.38	69.89	55.89	80.92	1429.44	65.19	59.17	36.48	62.58	92.15%
Script (Ours)	***Proposed***	77.85	62.32	53.19	71.37	56.47	88.83	1476.99	65.49	59.91	40.34	**64.96**	**95.65%**

C.3.2 Vision-based Methods

TRIM (Song et al., 2025) Pruning based solely on visual input, while ignoring textual context, may lead to suboptimal decisions. TRIM addresses this issue by utilizing CLIP-based similarity. It computes cosine similarities between image tokens (from the visual encoder) and text tokens (from the text encoder), and uses these similarity scores to rank visual tokens by importance. Low-similarity tokens are pruned to accelerate inference without significant performance loss.

VisionZip (Yang et al., 2025) A visual-only token pruning method that analyzes self-attention concentration in the visual encoder. VisionZip first selects dominant tokens with high self-attention weights. Then, it applies clustering on the remaining tokens to extract diverse contextual tokens. The union of dominant and contextual tokens is passed to the language model, aiming to preserve both saliency and diversity of visual content.

C.3.3 Similarity-based Methods

DivPrune (Alvar et al., 2025) DivPrune reformulates token pruning as a Maximum Minimum Distance Problem (MMDP), aiming to select the most diverse subset of visual tokens. Rather than relying solely on attention or similarity scores, it explicitly maximizes the minimum pairwise distance among retained tokens, ensuring the selected tokens cover a broad semantic space. This diversity-preserving strategy leads to robust performance under extreme token reduction.

C.4 Implementation Details

For image-based benchmarks, we adopt the official implementation of LLaVA[3]. The models are loaded using the released checkpoints (e.g., LLaVA-1.5 and LLaVA-NeXT), and we follow the default preprocessing pipeline for image resizing (e.g., 336×336 or 672×672 resolution), tokenization, and prompt formatting. All evaluations are conducted in a zero-shot setting unless otherwise specified. For video-based benchmarks, we follow the official implementation of LLaVA-NeXT[4], which also supports LLaVA-Video. We adopt the same architecture for multi-frame video inputs and process videos by uniformly sampling frames (e.g., 64 frames per video). Each frame is resized to 384×384, and visual tokens are pooled as described in the original LLaVA-Video paper. For evaluation, we use the `lmms-eval` toolkit[5], which provides standardized metric computation and dataset loading for long-form video understanding tasks. All experiments are run

[3] https://github.com/haotian-liu/LLaVA
[4] https://github.com/LLaVA-VL/LLaVA-NeXT
[5] https://github.com/EvolvingLMMs-Lab/lmms-eval

Table 10: **Performance comparisons on InternVL3-8B Zhu et al. (2025) across 10 image understanding benchmarks**. The best results in each setting are **bolded**, and the second-best are underlined.

Method	AI2D	TextVQA	ChartQA	OCRBench	HallBench	MME	MMB-EN	MMB-CN	Acc.	Average
Upper Bound, All 1280 Tokens (**100%**)										
InternVL3-8B	85.28	81.51	85.07	853	50.02	2393.22	83.82	82.54	84.22	100.0%
Retain 256 Tokens (↓ **80.0%**)										
FastV	82.21	74.34	70.58	632	48.45	2348,31	83.36	82.04	77.98	92.6%
DivPrune	80.88	64.70	57.51	477	38.63	2249.17	80.63	80.27	70.41	82.80%
Script	82.87	75.97	72.20	640	48.98	2334.22	83.45	81.57	**78.35**	**93.03%**
Retain 128 Tokens (↓ **90.0%**)										
FastV	77.35	63.55	46.82	426	42.75	2250.31	81.09	80.20	68.44	80.9%
DivPrune	76.45	55.44	42.58	378	37.57	2166.22	78.49	77.55	64.31	75.75%
Script	79.88	67.65	50.78	471	44.46	2282.98	82.12	80.53	**70.98**	**84.29%**

on NVIDIA H100 GPUs (80GB), using bfloat16 precision. For fair comparison across different models and pruning strategies, we keep inference batch size and decoding settings (e.g., temperature = 0.2, top-$k = 1$) consistent throughout all benchmarks. **For each benchmark, we repeat the evaluation 3 times and report the average result as the final score.**

D Additional Experimental Results

D.1 Script for Large Parameters of Model

As shown in Table 8 and Table 9, our method **Script** consistently outperforms other baselines across both 7B and 13B model settings. This demonstrates that Script is effective and scalable, maintaining strong performance regardless of model size. Our proposed method **Script** achieves top performance across all evaluated benchmarks in both 7B and 13B configurations. It demonstrates not only superior accuracy but also remarkable robustness across varying model capacities, consistently ranking first in overall average score.

D.2 Script for advanced open-source MLLM

In addition to LLaVA, we further apply Script to one of the most advanced open-source MLLMs to date, InternVL3. The results are shown in Table 10. Here, we fix the input resolution to 896×896, yielding 1,280 visual tokens. Notably, unlike its performance on the LLaVA series, DivPrune exhibits a significant performance drop on InternVL3, as it does not account for the relevance to user instructions during pruning. In contrast, our Script jointly considers both diversity and relevance, consistently achieving the best performance across different reduction ratios. Specifically, even when 90% of the visual tokens are removed, our method retains 83.9% of the original performance, 3% higher than the second-best FastV, demonstrating its effectiveness and adaptability in advanced MLLM architectures.

D.3 Statistical Significance Experiments

To assess the robustness and statistical reliability of our method, we conduct 20 independent runs using LLaVA-Next-7B under identical experimental settings with varying random seeds. All statistical comparisons are made against the Script. To determine whether Script is statistically better than other methods, we perform paired one-sided t-tests and Wilcoxon signed-rank tests. e evaluate the null hypothesis H_0: Script performs the same as the others, against the alternative hypothesis H_1: Script performs better. Since the resulting p-values are significantly lower than the significance threshold of

Table 11: Statistical comparison between our method and other baselines on MME benchmark with ratain 160 tokens. **StdDev** denotes standard deviation. Reported p-values correspond to the paired one-sided t-test and Wilcoxon signed-rank test. All experiments are set with the significance level of $\alpha = 5\%$.

Method	**Mean (%)**	**StdDev**	t**-test** p	**Wilcoxon** p
FastV	1076.81	2.62	1.01×10^{-43}	9.54×10^{-7}
TRIM	1316.74	3.41	8.96×10^{-35}	9.54×10^{-7}
VisionZip	1349.28	2.34	8.49×10^{-36}	9.54×10^{-7}
DivPrune	1347.72	2.04	1.90×10^{-36}	9.54×10^{-7}
SparseVLM	1369.62	2.39	7.58×10^{-35}	9.54×10^{-7}
Script(Ours)	1487.98	0.94	1.00	1.00

$\alpha = 0.05$, we reject the null hypothesis H_0. This suggests that the observed performance difference is statistically significant.

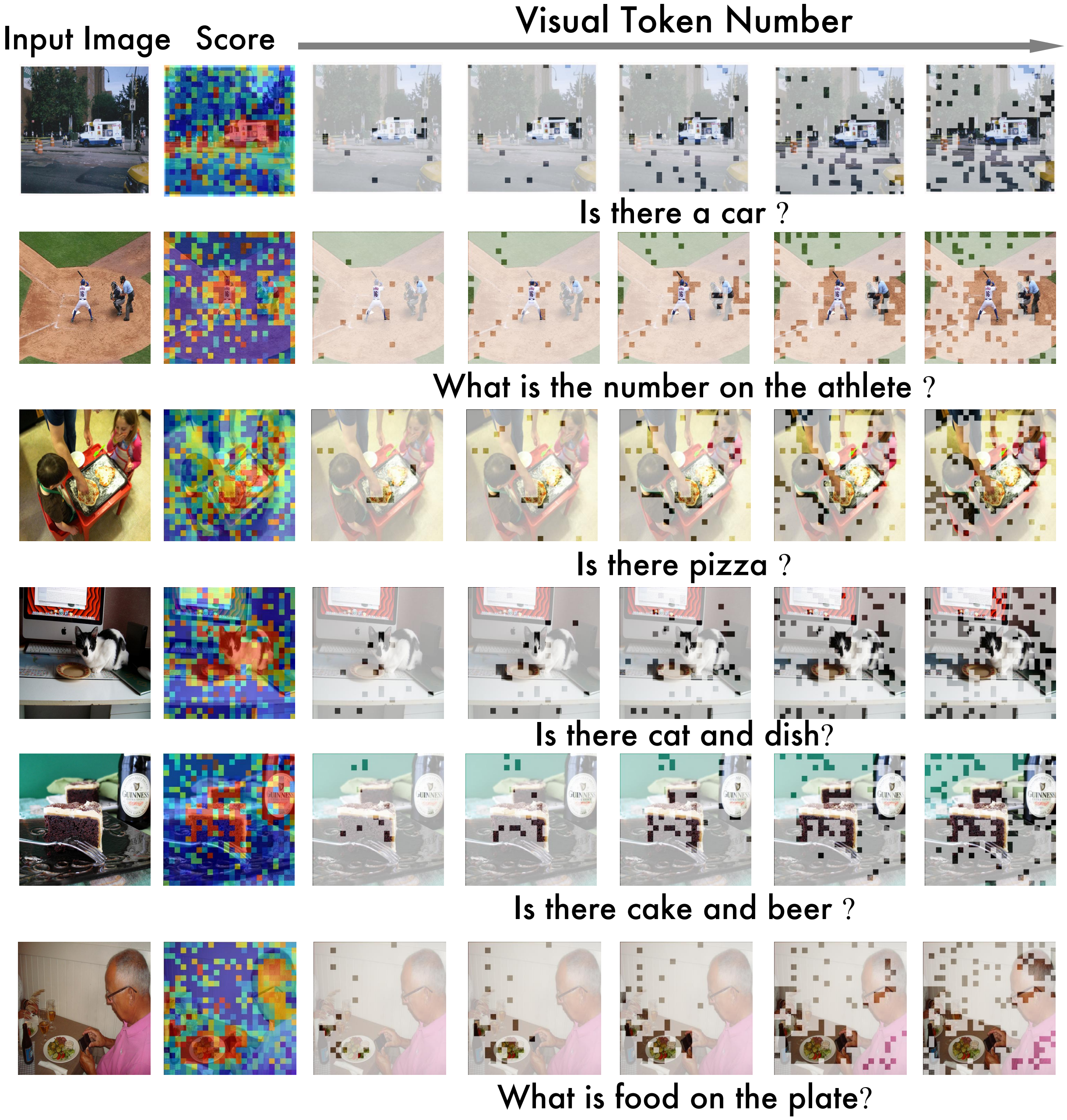


Figure 7: **Visualizations of relevance scores and retained tokens.** Each visualization illustrates the spatial attention allocated by models to regions corresponding to various textual instructions, demonstrating the capacity of pre-trained multimodal models to identify and focus on task-specific visual elements.

E Additional Visualization Results

In this section, we provide additional visualizations that comprehensively illustrate the relevance scores and retained visual tokens as shown in Figure 7 and 8. These visualizations further emphasize the strengths

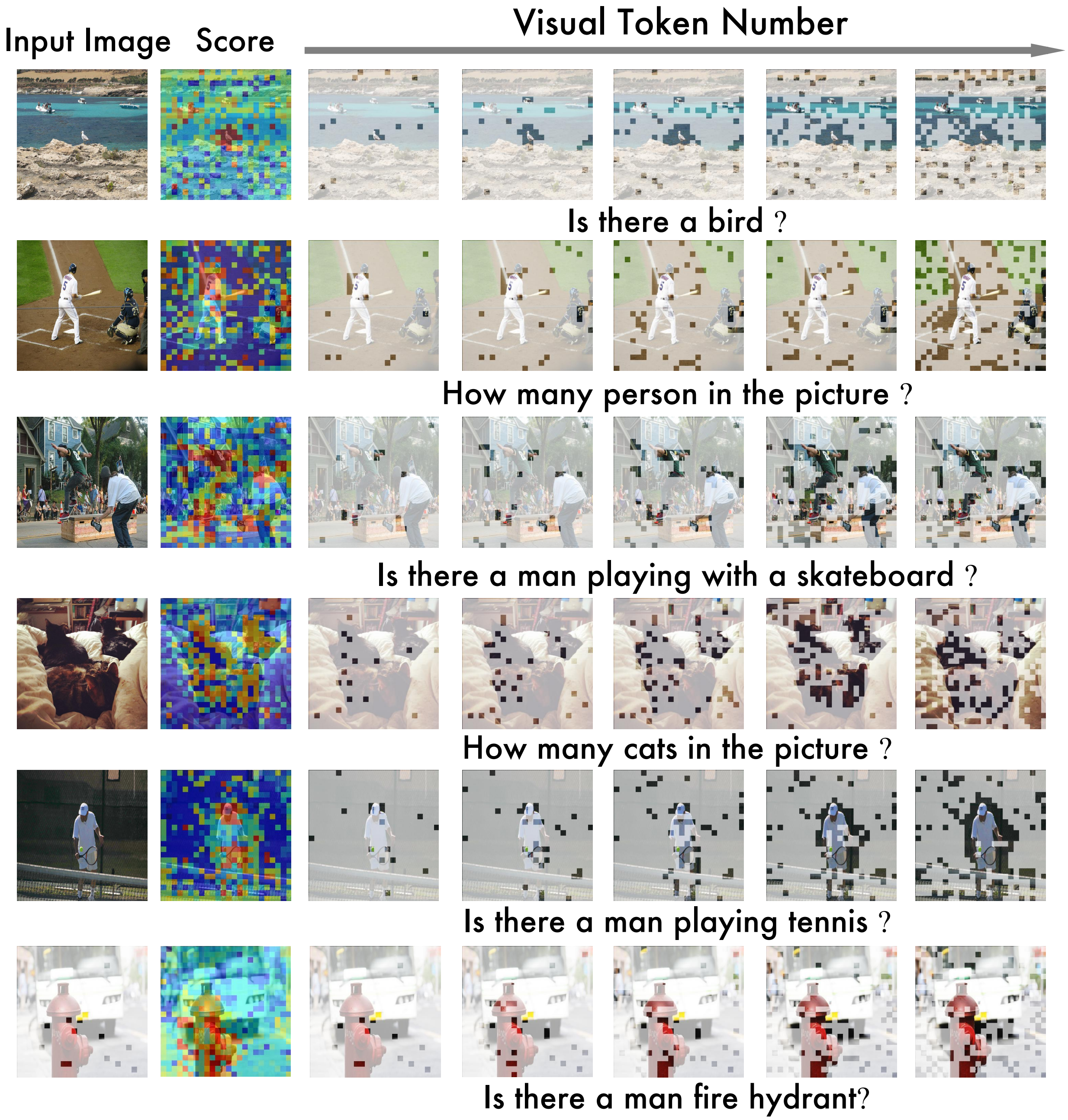

Input Image
Score
Visual Token Number
Is there a bird ?
How many person in the picture ?
Is there a man playing with a skateboard ?
How many cats in the picture ?
Is there a man playing tennis ?
Is there a man fire hydrant?

Figure 8: **Visualizations of relevance scores and retained tokens.** Each visualization illustrates the spatial attention allocated by models to regions corresponding to various textual instructions, demonstrating the capacity of pre-trained multimodal models to identify and focus on task-specific visual elements.

of models utilizing language-image pre-training, showcasing their enhanced capability to align textual instructions accurately with the corresponding visual regions. Specifically, the visualizations depict clear and intuitive correspondences: for instance, when instructions involve detecting the presence of specific objects or counting individuals or animals, the relevance maps highlight pertinent regions directly associated with these queries. Such precise spatial attention enables effective and efficient visual token pruning, significantly reducing redundancy by retaining only informative regions. This targeted retention not only improves computational efficiency but also boosts interpretability by transparently displaying the reasoning processes of multimodal large language models. Overall, these visualizations underscore the robustness and adaptability of pre-trained models in dynamically identifying and concentrating on task-specific visual details essential for accurate multimodal understanding.

To evaluate the effectiveness of our proposed method on larger language models, we apply Script to two models equipped with 13B LLMs: LLaVA-1.5-13B and LLaVA-NeXT-13B. The results are presented in Table 8 and Table 9. The larger language models lead to significant performance improvements and also make MLLMs less sensitive to visual token pruning. Among various pruning strategies, text-attention-based methods benefit the most from scaling up the language model, indicating that a larger LLM brings more accurate attention. Across different types of pruning methods, Script consistently outperforms all other approaches under various reduction ratios. With 77.8% of visual tokens removed, our method retains 98.0% and 99.9% of the original performance on LLaVA-1.5-13B and LLaVA-NeXT-13B, respectively, demonstrating its effectiveness on larger language models.

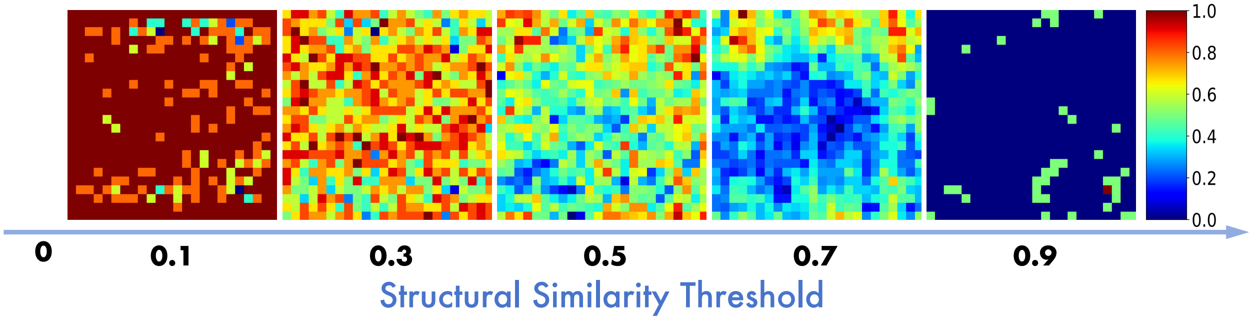


Figure 9: **Visualization of structural redundancy under different thresholds.** We show patch-level redundancy maps computed from CLIP-ViT features on images from the COCO dataset. Redundancy is defined as the average cosine similarity between each patch and its spatial neighbors. **Red** regions indicate high structural redundancy, while **blue** regions are more distinctive. As the threshold increases (from left to right: 0.1 to 0.9), the resulting selection becomes increasingly sparse, focusing on structurally salient regions.

Redundancy Visualization Across Thresholds. To better understand how structural redundancy varies across visual patches, we visualize the *redundancy score maps* under multiple threshold values ($\tau = 0.1, 0.3, 0.5, 0.7, 0.9$), as shown in Fig. 9. For each image, we compute a patch-wise redundancy score by averaging the cosine similarity of each patch with its 8-connected spatial neighbors. Binary masks are then derived by thresholding these scores at varying levels of τ.

From these visualizations, we make several concrete observations: **1. Low-threshold masks capture nearly all background regions.** At $\tau = 0.1$, the redundancy masks are overly dense, activating most of the image, including backgrounds and even parts of objects. For example, in the third row of Fig. 9, the sky and road areas are almost entirely marked as redundant, despite their varying brightness or shadows. This reflects high local structural similarity in these regions.

2. Mid-thresholds begin to separate object boundaries from background. At $\tau = 0.5$, patches corresponding to uniform regions, such as wall surfaces, grass fields, or repeated window patterns, remain marked as redundant. However, patches near object contours (e.g., bicycle spokes or human silhouettes) start to drop out of the redundancy mask, indicating a transition point in local distinctiveness.

3. High thresholds isolate semantically salient regions. At $\tau = 0.9$, the remaining non-redundant patches are sparsely located and strongly correspond to objects or parts that deviate structurally from their neighborhood. In the fourth row, only the traffic sign and vehicle wheels persist, while nearly all background and texture-repetitive areas are filtered out. This confirms that our score effectively distinguishes structurally unique regions.

4. Local redundancy clusters spatially. Across thresholds, redundant patches tend to form contiguous clusters (rather than scattered points), especially in flat surfaces like roads, skies, and building facades. This affirms our assumption that redundancy is fundamentally a local phenomenon, driven by short-range visual similarity.

These findings highlight the critical role of threshold selection in pruning. Lower thresholds retain nearly all patches and offer little compression, while overly high thresholds risk discarding important structural details. Our visualizations demonstrate that an appropriate threshold (e.g., $\tau = 0.5$) can reliably suppress locally redundant regions while preserving semantically meaningful tokens. This supports the design of our pruning strategy based on structural redundancy.

F Limitations

One key limitation of our work is that the proposed pruning method relies on direct access to the encoded visual tokens during inference, which limits its applicability to open-source Multimodal large language models. In contrast, many widely used proprietary models such as ChatGPT, Gemini, and Claude operate as closed systems, where intermediate visual features are not accessible. Although these black-box models also incur high computational costs for visual reasoning, our method cannot currently be applied in such settings. Moreover, while our method is compatible with advanced open-source MLLMs such as Qwen2.5-VL and InternVL3, we observe that these models are generally more sensitive to visual token pruning compared to the LLaVA series. Specifically, under the same pruning ratio, these models exhibit more significant performance degradation. This phenomenon is likely due to the fact that such architectures already incorporate internal visual token compression techniques, such as pixel unshuffle or token merging, which reduce redundancy prior to pruning. As a result, additional pruning may lead to excessive information loss. Adapting pruning strategies to better suit these highly optimized architectures, such as by incorporating pruning-aware training or model-specific heuristics, remains an important direction for future research.

G Broader Impacts

Multimodal large language models have demonstrated remarkable success across a wide range of domains, including education, accessibility, robotics, and content creation. Despite their capabilities, these models often incur high inference costs, particularly when processing high-resolution images or extended video sequences. Such computational demands pose substantial challenges for real-world deployment.

In this work, we introduce a simple yet effective visual token pruning strategy that enhances inference efficiency without requiring any additional training. By selectively removing redundant visual inputs, our method significantly reduces computational overhead and deployment costs. This improvement facilitates the deployment of MLLMs on resource-constrained platforms, such as mobile devices and edge computing environments, thereby promoting broader accessibility and scalability in practical applications.

However, it is important to note that improving computational efficiency does not inherently address the ethical challenges associated with MLLMs. Our method does not mitigate risks related to potential misuse, such as the generation of harmful content or the dissemination of misinformation. As such, continued research and policy development are essential to ensure the responsible and safe use of increasingly efficient multimodal language models.