SGPRUNE: SUBSPACE-GUIDED TOKEN PRUNING FOR MULTI-VIEW VISION-LANGUAGE MODELS IN AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multi-view vision-language models (VLMs) support scene understanding and reasoning in autonomous driving, but processing visual tokens from multiple cameras incurs substantial computational and memory costs. Effective pruning must preserve task-relevant content and visual diversity while allocating a limited token budget across views. We propose **SGPrune**, a training-free framework for subspace-guided visual token pruning in multi-view VLMs. SGPrune combines three stages: Attention-Based Token Selection, PCA Subspace Construction, and Adaptive Multi-View Token Pruning. The first two stages run offline, using text-to-vision attention to select question-relevant visual tokens from a subset of the training data and summarize them into view-specific PCA subspaces. The third stage operates online during inference using the stored subspaces to estimate token relevance and guide diversity-aware selection before the multimodal projector. This selection is performed jointly across views under a shared budget, yielding input-adaptive view allocations. Experiments on autonomous driving benchmarks show that SGPrune achieves state-of-the-art performance among recent pruning methods, improving the DriveLMM-o1 Overall Reasoning score by **1.13 points** and the MAPLM average score by **2.53 points** over state-of-the-art baselines at about 90% token reduction. Under the same setting, SGPrune achieves a $\mathbf{2.53}\times$ prefill speed on DriveLMM-o1 and a $\mathbf{5.68}\times$ speed on DriveMM compared to the vanilla models, offering a superior balance between performance and efficiency.

1 INTRODUCTION

Vision-Language Models (VLMs) (Bai et al., 2023; Zhu et al., 2025; Chen et al., 2024b) extend the reasoning capabilities of Large Language Models (LLMs) (Grattafiori et al., 2024; DeepSeek-AI, 2026; Achiam et al., 2023) to visual inputs by transforming images into visual tokens through vision encoders and modality alignment modules. By jointly processing visual and textual tokens, VLMs support scene understanding and reasoning over complex traffic situations in autonomous driving (AD). However, these capabilities introduce substantial computational and memory costs due to the large number of visual tokens. In InternVL3 (Zhu et al., 2025), each image tile is represented by 256 visual tokens, and processing multiple surrounding camera views further increases the visual token count, posing a challenge for latency-sensitive driving scenarios. Prior work has observed that visual tokens often contain more redundant information than textual tokens (Chen et al., 2024a). This observation motivates visual token pruning that balances task *relevance* and token *diversity*, preserving informative visual content while reducing redundancy and computational overhead.

Existing training-free token pruning methods commonly employ two selection strategies: (i) attention-based selection, which leverages text-to-vision attention to identify visual tokens relevant to the input question (Zhang et al., 2024; Chen et al., 2024a; Zhu et al., 2026). However, selecting tokens solely by their attention scores may favor salient objects without adequately preserving diverse contextual information. As illustrated in Fig. 1 (a), attention-based selection retains tokens covering prominent objects relevant to the question, but provides limited coverage of lane markings that offer complementary context for understanding the driving scene.

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

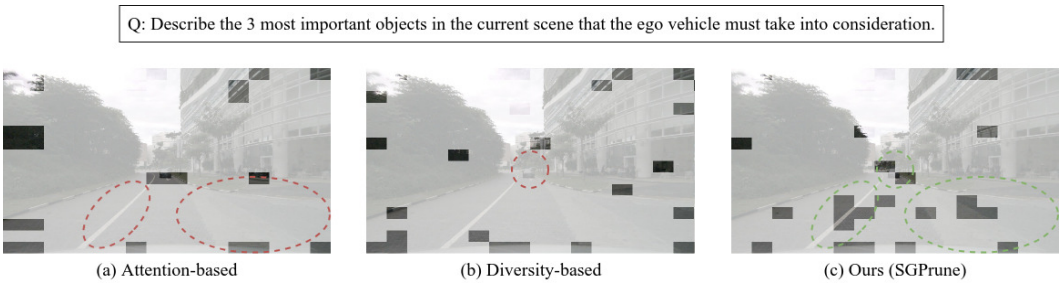


Figure 1: Qualitative comparison of visual token selection in a driving scene. (a) Attention-based selection preserves question-relevant objects but provides limited coverage of contextual information such as lane markings. (b) Diversity-based selection retains varied visual content but provides insufficient coverage of the question-relevant vehicle. (c) SGPrune balances relevance and diversity, preserving both question-relevant objects and complementary scene context. All methods use the same token budget.

(ii) Diversity-based selection instead uses token similarity to retain complementary visual tokens and suppress redundancy, sometimes incorporating additional importance estimates (Xiong et al., 2026; Fang et al., 2026; Kim et al., 2026). Depending on the method, diversity-based pruning is performed either before the LLM or within its early layers, using token similarity rather than decoder attention scores as the diversity criterion. Nevertheless, token diversity alone does not ensure textual relevance, as visually distinctive tokens may contain information unrelated to the input question. Fig. 1 (b) illustrates this limitation: the highlighted vehicle region receives insufficient token coverage despite its relevance to the question.

Multi-view pruning additionally requires allocating a shared token budget across cameras. View-specific ratios optimized at the dataset level do not adapt to scene-dependent changes in individual inputs (Xiong et al., 2026), motivating input-adaptive token allocation.

To address these limitations, we propose **SGPrune**, a training-free framework for subspace-guided visual token pruning in multi-view VLMs. SGPrune combines a text-informed relevance prior with diversity-aware token selection. Specifically, we first collect visual tokens using text-to-vision attention on a subset of the training data and construct a subspace for each view offline using principal component analysis (PCA) (Abdi & Williams, 2010). During inference, alignment with these stored subspaces provides relevance weights, which guide cosine-distance-based diversity selection. This design reuses text-guided information without requiring decoder attention extraction or subspace reconstruction during online pruning, while keeping all model parameters frozen. As illustrated in Fig. 1 (c), SGPrune retains visual tokens covering both question-relevant objects and complementary context, including lane markings.

Furthermore, SGPrune jointly selects tokens across views under a shared budget, allowing view-specific pruning ratios to emerge adaptively from the selected tokens. Qualitative comparisons of fixed and adaptive view allocation are provided in Appendix A.1.

2 Related Work

2.1 Multi-View VLMs in Autonomous Driving

Vision-language models have emerged as a promising approach to autonomous driving, connecting visual scene understanding with language-based reasoning and decision-making. DriveLM (Sima et al., 2024) introduces graph-structured visual question answering, connecting perception, prediction, and planning through interdependent question–answer pairs. Senna (Jiang et al., 2024; Song et al., 2026) integrates a vision-language model with an end-to-end driving model, using high-level planning decisions expressed in natural language to guide trajectory prediction. OmniDrive (Wang et al., 2025) introduces a vision-language dataset emphasizing 3D situational understanding and counterfactual reasoning, together with agent architectures that connect visual representations with driving decisions and planning. DriveMM (Huang et al., 2024) develops an all-in-one multimodal

model that unifies perception, prediction, and planning for autonomous driving, supporting diverse visual inputs, including multi-view images and videos. DriveLMM-o1 (Ishaq et al., 2025) introduces a reasoning-oriented dataset and model to support step-by-step understanding of driving scenarios.

While multi-view inputs provide complementary observations of the surrounding environment, they also increase the number of visual tokens processed by the language model. The resulting computational burden motivates efficient token selection for driving VLMs.

2.2 Visual Token Pruning for VLMs

Visual token pruning reduces the computational cost of VLMs by retaining a compact subset of visual tokens. Attention-based methods estimate token importance using attention scores. FastV (Chen et al., 2024a) removes low-attention visual tokens after early decoder layers. SparseVLM (Zhang et al., 2024) identifies visually relevant text tokens and uses their attention to guide visual token sparsification. HAWK (Zhu et al., 2026) scores visual tokens by the pre-RoPE attention from instruction tokens to visual tokens in the first decoder layer's query/key projections, weighted by offline-estimated per-head importance, and prunes before the full LLM forward pass.

Diversity-based approaches exploit token relationships to reduce redundancy and preserve complementary visual information. DivPrune (Alvar et al., 2025) formulates token selection as a max-min diversity problem. Importance-aware extensions further guide redundancy reduction using token-level scores. PruneSID (Fang et al., 2026) employs Principal Semantic Components Analysis to group visual tokens and estimate their importance, followed by intra-group non-maximum suppression. It also introduces an information-aware mechanism for adapting token budgets to image complexity. ZOO-Prune (Kim et al., 2026) estimates token sensitivity through zeroth-order perturbations at the multimodal projector and combines these scores with diversity-aware selection. For multi-view autonomous driving, Prune2Drive (Xiong et al., 2026) combines token-wise farthest-point sampling with dataset-level optimization of view-specific pruning ratios.

3 Method

3.1 Overview

Consider a multi-view input with V images. A frozen vision encoder produces N_v pre-projector visual tokens for view v, represented by $\mathbf{F}^v = [\mathbf{f}_1^v, \ldots, \mathbf{f}_{N_v}^v]^\top \in \mathbb{R}^{N_v \times d}$, where $\mathbf{f}_i^v \in \mathbb{R}^d$ denotes the vector representation of the i-th visual token in view $v \in \{1, \ldots, V\}$. Given a total visual-token budget $K \ll \sum_{v=1}^{V} N_v$, our goal is to retain task-relevant visual information while reducing redundancy and the computational cost of subsequent multimodal processing.

As illustrated in Fig. 2, SGPrune follows a three-stage pipeline: (a) attention-based token selection, (b) PCA subspace construction, and (c) adaptive multi-view token pruning. The first two stages are performed offline using D_{sub}, a fixed calibration subset drawn from the training data. Each example $x \in D_{sub}$ consists of multi-view images and the corresponding question, including answer options when provided. We collect visual tokens using text-to-vision attention and construct a PCA subspace for each view from the selected tokens. During inference, the third stage uses the stored subspaces to derive relevance weights for diversity-aware token selection before the multimodal projector. Tokens are jointly selected across views under a shared budget, yielding input-dependent view allocations. All model parameters remain frozen throughout the pipeline.

3.2 Attention-Based Token Selection

As illustrated in Fig. 2 (a), this stage collects question-relevant visual tokens from D_{sub} for subsequent subspace construction. For each example, we first identify visually relevant text tokens and attention heads once over the whole multi-view sequence, then use their text-to-vision attention to select visual tokens within each view and collect the corresponding pre-projector tokens.

Text token selection. Let C denote the set of candidate text-token indices in the question and its answer options, if provided, and let $P = \{(v, i) : 1 \le v \le V,\ 1 \le i \le N_v\}$ index all visual tokens. Let $\mathbf{z}_i^v \in \mathbb{R}^{d_{\text{LM}}}$ denote the projected embedding of visual token i in view v, and $\mathbf{t}_j \in \mathbb{R}^{d_{\text{LM}}}$ the input

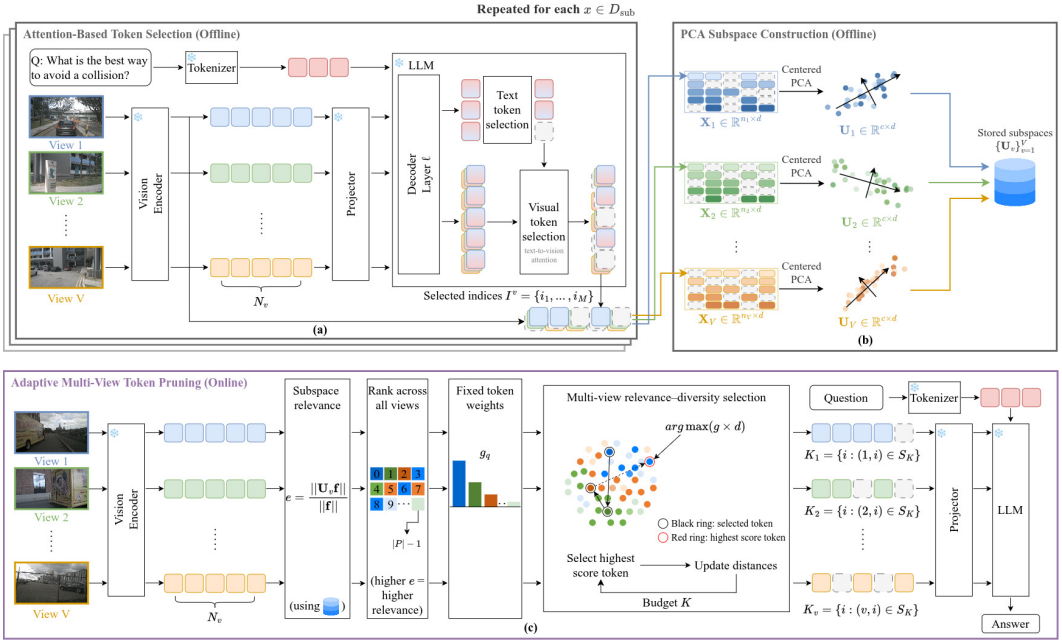


Figure 2: **Overview of SGPrune.** (a) For each example in the training subset D_{sub}, text-to-vision attention guides the selection and collection of relevant pre-projector visual tokens. (b) The collected tokens are aggregated separately for each view, and centered PCA constructs reusable subspace bases. (c) During inference, subspace relevance scores are ranked across all views to obtain fixed token weights. These weights guide multi-view relevance–diversity selection under a shared token budget, yielding input-adaptive view allocations.

embedding of candidate text token j. We compute a visual relevance score for each candidate over all views:

$$a_j = \frac{1}{|P|} \sum_{(v,i) \in P} \frac{\exp\left((\mathbf{z}_i^v)^\top \mathbf{t}_j\right)}{\sum_{k \in C} \exp\left((\mathbf{z}_i^v)^\top \mathbf{t}_k\right)}. \tag{1}$$

We retain candidates whose scores exceed the mean over candidate text tokens:

$$R = \left\{ j \in C : a_j > \frac{1}{|C|} \sum_{k \in C} a_k \right\}. \tag{2}$$

The selected candidate text-token set R is shared across all views; only the subsequent top-M selection is performed within each view.

Visual token selection. We extract text-to-vision attention at an intermediate decoder layer ℓ. Let $\rho(r)$ denote the position of text token r in the multimodal sequence and $p(v, i)$ the position of visual token i of view v, and let $A^{(\ell)}_{h,p,p'}$ be the attention weight from position p to position p' in head h at layer ℓ. Raw attention carries a positional bias that favours visual tokens close to the text tokens, so we first remove its content-independent part:

$$\tilde{A}^{(\ell)}_{h,p,p'} = \frac{A^{(\ell)}_{h,p,p'}}{\Pi_{p,p'}}, \qquad \Pi_{p,p'} = \frac{\langle \mathrm{RoPE}_p(\mathbf{1}),\, \mathrm{RoPE}_{p'}(\mathbf{1}) \rangle}{\sqrt{d_h}}, \tag{3}$$

where $\mathrm{RoPE}_p(\cdot)$ is the rotary embedding at position p, $\mathbf{1} \in \mathbb{R}^{d_h}$ the all-ones vector and d_h the head dimension. We then rank attention heads by their total corrected attention from R to all visual tokens in P and retain the top-H_s heads as H:

$$H = \mathrm{TopK}_h \left(\sum_{r \in R} \sum_{(v,i) \in P} \tilde{A}^{(\ell)}_{h,\rho(r),p(v,i)},\ H_s \right). \tag{4}$$

Each visual token is scored by averaging the corrected attention over the selected heads and text tokens in R:

$$s_i^v = \frac{1}{|H||R|} \sum_{h \in H} \sum_{r \in R} \tilde{A}^{(\ell)}_{h,\rho(r),p(v,i)}. \tag{5}$$

For each training example $x \in D_{sub}$, let $\mathbf{s}^v(x) = [s_1^v(x), \ldots, s_{N_v}^v(x)]^\top$ denote the visual-token score vector. We select the top-M visual-token indices within each view:

$$I^v(x) = \mathrm{TopK}\left(\mathbf{s}^v(x), M\right), \tag{6}$$

where TopK returns the set of indices of the highest-scoring elements. We then collect the corresponding pre-projector tokens:

$$E^v(x) = \left\{\mathbf{f}_i^v(x) : i \in I^v(x)\right\}. \tag{7}$$

Thus, attention determines which visual tokens to select, while we collect the corresponding pre-projector tokens. We aggregate these tokens across D_{sub} separately for each view to construct its PCA subspace.

3.3 PCA Subspace Construction

As illustrated in Fig. 2 (b), we construct a compact subspace per view from the visual tokens selected in the previous stage. For each view v, we stack the selected tokens in $E^v(x)$ over all examples $x \in D_{sub}$ as rows of $\mathbf{X}_v \in \mathbb{R}^{n_v \times d}$, where n_v is the total number of collected visual tokens for that view. Let $\boldsymbol{\mu}_v \in \mathbb{R}^d$ denote the mean token vector and $\mathbf{1}_{n_v} \in \mathbb{R}^{n_v}$ the vector of all ones. We center the matrix and compute its singular value decomposition:

$$\mathbf{X}_v - \mathbf{1}_{n_v}\boldsymbol{\mu}_v^\top = \mathbf{L}_v \boldsymbol{\Sigma}_v \mathbf{R}_v^\top. \tag{8}$$

With singular values ordered in descending order, let $\mathbf{r}_{v,j}$ denote the j-th column of $\mathbf{R}_v$. We retain the leading c right singular vectors to form the basis matrix:

$$\mathbf{U}_v = [\mathbf{r}_{v,1}, \ldots, \mathbf{r}_{v,c}]^\top \in \mathbb{R}^{c \times d}. \tag{9}$$

The rows of $\mathbf{U}_v$ are orthonormal and span the subspace

$$U_v = \mathrm{span}\left\{\mathbf{r}_{v,1}, \ldots, \mathbf{r}_{v,c}\right\} \subseteq \mathbb{R}^d. \tag{10}$$

This subspace captures the principal variation in the selected visual tokens. We construct and store the view-specific bases offline and keep them fixed during inference, where they serve as relevance priors for visual token selection.

3.4 Adaptive Multi-View Token Pruning

As illustrated in Fig. 2 (c), we use the stored subspaces to guide joint visual token selection across views under a shared token budget.

Subspace relevance. During online token pruning, SGPrune estimates the relevance of each visual token using the stored basis matrix $\mathbf{U}_v$. We compute the normalized projection magnitude:

$$e_i^v = \frac{\|\mathbf{U}_v \mathbf{f}_i^v\|_2}{\max\left(\|\mathbf{f}_i^v\|_2, \epsilon\right)}, \tag{11}$$

where $\epsilon > 0$ ensures numerical stability. This score estimates the relevance of each visual token using the stored subspace U_v and serves as a prior for diversity-aware selection.

Although relevance scores are computed using view-specific subspaces, they are ranked jointly over all candidates in P. We convert these global ranks into positive relevance weights:

$$g_i^v = \exp\left[\beta\left(\frac{\mathrm{rank}_P(e_i^v)}{|P| - 1} - \frac{1}{2}\right)\right], \tag{12}$$

where $|P| = \sum_{v=1}^V N_v$ and rank_P assigns ranks from 0 to $|P| - 1$ in ascending score order across all views. The parameter $\beta > 0$ controls the strength of relevance weighting.

Multi-view relevance–diversity selection. Selecting tokens solely by subspace relevance may retain redundant visual information. We therefore combine the relevance weights with cosine-distance farthest-point sampling over the pooled candidate set P under a shared token budget K.

Let $q = (v, i)$ index a candidate token, with $\mathbf{f}_q = \mathbf{f}_i^v$ and $g_q = g_i^v$, and let $S_t \subseteq P$ denote the set of tokens selected after t iterations. We initialize the selected index set with one of the highest-weight tokens:

$$q_1 \in \arg\max_{q \in P} g_q, \qquad S_1 = \{q_1\}. \tag{13}$$

At each subsequent iteration $t = 2, \ldots, K$, we compute each remaining candidate's minimum cosine distance to the selected tokens:

$$d_q^{(t)} = \min_{q' \in S_{t-1}} \left(1 - \cos(\mathbf{f}_q, \mathbf{f}_{q'})\right), \tag{14}$$

where $\cos(\mathbf{f}_q, \mathbf{f}_{q'})$ denotes cosine similarity between the two token vectors. We select the candidate with the highest relevance-weighted diversity score:

$$q_t \in \arg\max_{q \in P \setminus S_{t-1}} g_q d_q^{(t)}, \qquad S_t = S_{t-1} \cup \{q_t\}. \tag{15}$$

For each input, relevance weights remain fixed, while minimum distances are updated after each selected token. This global selection accounts for token redundancy both within and across views.

Selection continues until K tokens are retained. The resulting allocation to each view is

$$K_v = \left|\{i : (v, i) \in S_K\}\right|, \qquad \sum_{v=1}^{V} K_v = K. \tag{16}$$

Thus, SGPrune jointly selects visual tokens and determines input-dependent view budgets.

Finally, we restore the original token order within each view and pass the selected tokens to the multimodal projector, preserving view ordering and image delimiters. Online pruning uses only the current visual tokens and stored subspaces, without extracting decoder attention.

4 Experiments

Following Prune2Drive (Xiong et al., 2026), we evaluate the effectiveness of SGPrune on two autonomous driving models, DriveLMM-o1 and DriveMM. We evaluate DriveMM on MAPLM and DriveLMM-o1 on its driving reasoning benchmark. To assess applicability beyond autonomous driving, we additionally evaluate InternVL3-8B on seven general vision-language benchmarks using LMMs-Eval (Zhang et al., 2025): GQA (Hudson & Manning, 2019), POPE (Li et al., 2023), ChartQA (Masry et al., 2022), DocVQA (Mathew et al., 2021), AI2D (Kembhavi et al., 2016), InfoVQA (Mathew et al., 2022), and ScienceQA (Lu et al., 2022). We compare SGPrune with representative visual token pruning methods, including SparseVLM (Zhang et al., 2024), HAWK (Zhu et al., 2026), PruneSID (Fang et al., 2026), ZOO-Prune (Kim et al., 2026), and Prune2Drive (Xiong et al., 2026). All methods are evaluated under matched total visual token budgets, corresponding to approximately 75% and 90% token reduction. All experiments are conducted on NVIDIA H200 GPU with 141 GB of memory.

Unless stated otherwise, SGPrune uses decoder layer $\ell = 15$ with the top $H_s = 16$ of 32 attention heads, $c = 64$, and $\beta = 2$ for the driving models ($M = 50$ for DriveLMM-o1 and $M = 75$ for DriveMM), and $M = 25$, $c = 128$, and $\beta = 0.5$ for InternVL3-8B, where all heads are used and R contains all question tokens; the norm floor is $\epsilon = 10^{-6}$ throughout.

4.1 Results on Autonomous Driving Benchmarks

Results on DriveLMM-o1. DriveLMM-o1 is fine-tuned from InternVL2.5-8B (Chen et al., 2024b) and inherits its InternViT-300M-448px-V2_5 vision encoder; input images are resized to 448×448 and, after pixel shuffle, yield 256 visual tokens per view. Following Prune2Drive (Xiong et al., 2026), we evaluate pruning budgets of 64 and 25 retained tokens per image on average, corresponding to token reductions of 75.0% and 90.2%, respectively.

Table 1: **Performance comparison on DriveLMM-o1.** Token budgets denote the average number of retained visual tokens per image. The best and second-best results within each pruning budget are bolded and underlined, respectively.

Method	Risk Assessment Accuracy	Traffic Rule Adherence	Scene Awareness and Object Und.	Relevance	Missing Details	Overall Reasoning
Upper Bound, 256 Tokens per image (100%)						
DriveLMM-o1	74.54	83.18	76.63	80.94	75.96	76.57
Retain 64 Tokens per image (↓75.0%)						
SparseVLM (ICML 2025)	73.47	82.42	75.58	79.83	74.86	75.53
HAWK (CVPR 2026)	74.06	82.88	76.24	80.41	75.41	76.07
PruneSID (ICLR 2026)	<u>74.40</u>	<u>83.12</u>	<u>76.74</u>	<u>80.84</u>	75.75	<u>76.46</u>
ZOO-Prune (CVPR 2026)	74.07	82.98	76.57	80.64	75.49	76.24
Prune2Drive (CVPR 2026)	74.15	83.03	76.31	80.72	<u>75.89</u>	76.27
SGPrune	**74.81**	**83.86**	**77.28**	**81.72**	**76.52**	**77.11**
Retain 25 Tokens per image (↓90.2%)						
SparseVLM (ICML 2025)	66.52	78.05	72.12	74.93	70.91	71.46
HAWK (CVPR 2026)	71.92	81.34	73.98	78.09	72.80	73.96
PruneSID (ICLR 2026)	73.31	<u>82.51</u>	75.83	<u>80.00</u>	74.97	75.63
ZOO-Prune (CVPR 2026)	<u>73.40</u>	82.41	<u>75.84</u>	79.94	74.73	75.57
Prune2Drive (CVPR 2026)	73.03	82.33	75.83	79.90	<u>75.04</u>	<u>75.64</u>
SGPrune	**74.09**	**83.58**	**76.83**	**81.33**	**76.24**	**76.77**

As shown in Table 1, SGPrune achieves the highest scores among the compared pruning methods across all six evaluation dimensions at both budgets. At 75.0% token reduction, it achieves an Overall Reasoning score of 77.11, outperforming PruneSID by 0.65 points. At 90.2% reduction, it achieves 76.77, exceeding Prune2Drive by 1.13 points. Both scores slightly exceed the unpruned model's score of 76.57, indicating strong performance preservation under aggressive visual token reduction.

Results on DriveMM. DriveMM resizes each input image to 384×384 and uses the SigLIP (Zhai et al., 2023) vision encoder to produce 729 visual tokens per view. We evaluate pruning methods on MAPLM with average budgets of 180 and 72 retained tokens per image, corresponding to token reductions of 75.3% and 90.1%, respectively.

As shown in Table 2, SGPrune achieves the highest FRM, QNS, and average scores among the compared pruning methods at both budgets. With a 180-token budget, SGPrune achieves an average score of 69.41, outperforming the strongest baseline, PruneSID, by 1.46 points.

The advantage over the strongest baseline increases at the more restrictive 72-token budget. SGPrune achieves an average score of 67.41, exceeding Prune2Drive by 2.53 points, with improvements of 3.60 and 1.47 points on FRM and QNS, respectively. Notably, SGPrune with a 72-token budget outperforms Prune2Drive with a 180-token budget in average score (67.41 vs. 66.31), demonstrating effective performance preservation with substantially fewer visual tokens.

Table 2: **Performance comparison on MAPLM using DriveMM.** FRM and QNS denote frame-overall accuracy and question-overall accuracy, respectively. Average denotes the mean of the two scores.

Method	FRM	QNS	Average
Upper Bound, 729 Tokens per image (100%)			
DriveMM	59.07	96.15	77.61
Retain 180 Tokens per image (↓75.3%)			
SparseVLM (ICML 2025)	48.73	82.05	65.39
HAWK (CVPR 2026)	34.73	76.25	55.49
PruneSID (ICLR 2026)	<u>52.73</u>	<u>83.17</u>	<u>67.95</u>
ZOO-Prune (CVPR 2026)	52.60	83.08	67.84
Prune2Drive (CVPR 2026)	50.53	82.08	66.31
SGPrune	**54.80**	**84.02**	**69.41**
Retain 72 Tokens per image (↓90.1%)			
SparseVLM (ICML 2025)	27.13	74.02	50.58
HAWK (CVPR 2026)	26.27	72.68	49.48
PruneSID (ICLR 2026)	46.33	80.58	63.46
ZOO-Prune (CVPR 2026)	46.80	80.85	63.83
Prune2Drive (CVPR 2026)	<u>48.47</u>	<u>81.28</u>	<u>64.88</u>
SGPrune	**52.07**	**82.75**	**67.41**

Table 3: **Performance comparison on general vision-language benchmarks using InternVL3-8B.** Avg. denotes the mean of per-benchmark score ratios relative to the unpruned model, expressed as a percentage.

Method	GQA	POPE	ChartQA	DocVQA	AI2D	InfoVQA	ScienceQA	Avg.
Upper Bound, 256 Tokens (100%)								
InternVL3-8B	62.22	89.82	78.20	65.73	82.80	39.83	97.82	100%
Retain 64 Tokens (↓75.0%)								
SparseVLM (ICML 2025)	53.79	86.21	27.64	23.59	75.36	24.53	91.82	71.4%
PruneSID (ICLR 2026)	**59.50**	88.88	**56.32**	**34.88**	77.72	**30.19**	**95.19**	**83.8%**
Prune2Drive (CVPR 2026)	59.27	88.22	47.32	31.58	77.36	28.66	91.92	80.2%
SGPrune	59.32	**89.37**	53.32	34.73	**78.69**	29.94	92.91	83.0%
Retain 25 Tokens (↓90.2%)								
SparseVLM (ICML 2025)	42.19	65.32	13.56	10.99	66.68	23.97	81.85	57.0%
PruneSID (ICLR 2026)	**55.26**	**85.63**	32.36	18.85	69.62	26.21	**86.86**	70.4%
Prune2Drive (CVPR 2026)	54.44	83.45	26.08	18.52	69.85	25.02	86.51	68.2%
SGPrune	55.09	85.46	**33.60**	**19.69**	**70.66**	**26.92**	86.71	**71.2%**

4.2 Results on General Vision-Language Benchmarks

We evaluate SGPrune on InternVL3-8B across seven general vision-language benchmarks with 64 and 25 retained tokens per image. As shown in Table 3, SGPrune retains 83.0% of the vanilla model's performance on average at the 64-token budget, compared with 83.8% for PruneSID. SGPrune outperforms Prune2Drive on all seven benchmarks and achieves the highest scores among the compared pruning methods on POPE and AI2D.

Under the more restrictive 25-token budget, SGPrune achieves the highest average relative performance of 71.2%, exceeding PruneSID and Prune2Drive by 0.8 and 3.0 percentage points, respectively. It achieves the highest scores on ChartQA, DocVQA, AI2D, and InfoVQA and outperforms Prune2Drive on all seven benchmarks. These results demonstrate competitive performance beyond autonomous driving, with a stronger relative advantage under aggressive token pruning.

4.3 Ablation Studies

Effect of relevance and diversity. We compare SGPrune with relevance-only selection, which retains tokens with the highest relevance weights, and diversity-only selection, which uses cosine-distance farthest-point sampling without relevance weighting. All variants operate on pooled multi-view candidates under matched total token budgets. As shown in Table 4, combining both criteria improves performance across all evaluated settings. At 90.2% token reduction on DriveLMM-o1, SGPrune outperforms relevance-only and diversity-only selection by 1.16 and 1.22 points, respectively. On DriveMM, the corresponding gains increase from 3.66 and 5.94 points at 75.3% reduction to 8.46 and 8.03 points at 90.1% reduction; at the tighter budget, either criterion alone loses more than 8 points, so neither relevance nor diversity by itself accounts for the result. These results support the complementary roles of relevance and diversity under limited token budgets.

Table 4: **Component ablation of SGPrune.** We report Overall Reasoning for DriveLMM-o1 and the mean of FRM and QNS on MAPLM for DriveMM under matched total token budgets. The best results are bolded.

Variant	DriveLMM-o1		DriveMM	
Token reduction	*75.0%*	*90.2%*	*75.3%*	*90.1%*
Selection criterion				
Relevance only	76.98	75.61	65.75	58.95
Diversity only	76.18	75.55	63.47	59.38
Offline token selection				
Random instead of attention-based	76.34	75.31	63.81	60.58
SGPrune	**77.11**	**76.77**	**69.41**	**67.41**

Effect of attention-based token selection. We replace tokens selected by attention-based token selection with randomly sampled visual tokens for offline PCA construction, keeping the remaining configuration unchanged. As shown in Table 4, attention-based token selection improves Overall Reasoning on DriveLMM-o1 by 0.77 and 1.46 points at 75.0% and 90.2% token reduction, respectively. On DriveMM, the corresponding gains are 5.60 and 6.83 points at 75.3% and 90.1% reduction. The larger gains under more aggressive pruning on both models highlight the value of attention-based token selection for constructing effective subspace relevance priors.

An additional comparison of shared and view-specific subspaces is provided in Appendix A.2.

4.4 Efficiency Analysis

Table 5: **Efficiency comparison on DriveLMM-o1 and DriveMM.** Pre./Dec. denote prefill/decoding speedups.

Method	Tokens	Speedup ↑		FLOPs (%) ↓	KV Cache (MB) ↓	Score ↑
		Pre.	Dec.			
Vanilla DriveLMM-o1	1536	1.00×	1.00×	100.0	232.4	76.57
+ SparseVLM (ICML 2025)	150	2.30×	1.00×	21.5	**50.7**	71.46
+ HAWK (CVPR 2026)	150	**2.54**×	1.00×	**21.3**	**50.7**	73.96
+ PruneSID (ICLR 2026)	150	2.11×	0.99×	**21.3**	**50.7**	75.63
+ ZOO-Prune (CVPR 2026)	150	1.70×	1.00×	73.4	**50.7**	75.57
+ Prune2Drive (CVPR 2026)	150	2.26×	1.00×	23.7	56.4	75.64
+ **SGPrune**	150	2.53×	0.99×	**21.3**	**50.7**	**76.77**
Vanilla DriveMM	2916	1.00×	1.00×	100.0	410.4	77.61
+ SparseVLM (ICML 2025)	288	4.36×	1.02×	15.7	**65.9**	50.58
+ HAWK (CVPR 2026)	288	**6.09**×	1.01×	**15.3**	**65.9**	49.48
+ PruneSID (ICLR 2026)	288	3.85×	1.02×	**15.3**	**65.9**	63.46
+ ZOO-Prune (CVPR 2026)	288	2.88×	1.02×	50.4	**65.9**	63.83
+ Prune2Drive (CVPR 2026)	288	5.31×	1.02×	18.0	76.8	64.88
+ **SGPrune**	288	5.68×	1.01×	**15.3**	**65.9**	**67.41**

We evaluate the computational cost and KV cache memory usage of SGPrune on DriveLMM-o1 and DriveMM using an NVIDIA H200 GPU with 141 GB of memory. As shown in Table 5, reducing the total visual-token count from 1536 to 150 on DriveLMM-o1 achieves a $2.53\times$ prefill speedup, reduces FLOPs to 21.3% of the vanilla model, and decreases KV cache memory from 232.4 to 50.7 MB. On DriveMM, reducing the token count from 2916 to 288 yields a $5.68\times$ prefill speedup, with FLOPs reduced to 15.3% of the vanilla model and KV cache memory decreasing from 410.4 to 65.9 MB. SGPrune achieves the near-best prefill speedup and the highest task score among the compared pruning methods on both models.

5 Conclusion

In this work, we present SGPrune, a training-free framework for efficient visual token pruning in multi-view VLMs for autonomous driving. By constructing reusable PCA subspaces from visual tokens selected via text-to-vision attention, SGPrune combines a text-informed relevance prior with diversity-aware token selection without extracting decoder attention during inference. Experiments demonstrate that SGPrune achieves state-of-the-art performance among the compared pruning methods on driving benchmarks, improving DriveLMM-o1 Overall Reasoning by 1.13 points and the MAPLM average score by 2.53 points over the strongest baselines at approximately 90% token reduction. It also achieves prefill speedups of $2.53\times$ on DriveLMM-o1 at 90.2% reduction and $5.68\times$ on DriveMM at 90.1% reduction, alongside competitive performance on seven general vision-language benchmarks. These findings demonstrate that reusable subspace priors can effectively guide relevance–diversity selection, enabling more efficient multi-view VLM inference.

AI USE STATEMENT

Ideation. No AI tools were used for research ideation. The research problem, the SGPrune method, and the experimental design were conceived entirely by the authors.

Writing. An AI assistant was used to produce first drafts of some sections (in particular, parts of the appendix) and to translate passages drafted by the authors. All drafts and translations were then rewritten, elaborated, and finalized by the authors, who checked every statement against the experiments; the final text, claims, and wording are the authors' own.

Experiments. An AI coding assistant was used to write and debug the code for running the benchmarks; all such code was reviewed and verified by the authors, who take responsibility for every reported number. For evaluation, the DriveLMM-o1 reasoning scores are produced by an LLM judge (GPT-4o-mini), as required by the benchmark's official protocol.

We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work proposes a training-free inference-time method and does not involve human subjects, new data collection, or model training. All experiments use publicly released benchmarks (DriveLMM-o1, which builds on nuScenes; MAPLM; and seven general vision-language benchmarks) and publicly released model checkpoints under their respective licenses; nuScenes images are distributed with faces and license plates anonymized, and we release no images or derived data beyond a list of benchmark item identifiers and PCA bases that contain no personal information. The DriveLMM-o1 reasoning score follows the benchmark's official protocol, which sends questions, reference answers, and model outputs to a third-party API (GPT-4o-mini); this data consists solely of public benchmark text and model generations. Because the method is intended for autonomous-driving assistants, we note that any token-pruning scheme discards visual information and could omit safety-relevant details in rare cases; our evaluation is offline on benchmark data, and we do not claim readiness for deployment, which would require system-level safety validation beyond benchmark accuracy. Finally, by reducing the visual tokens processed per query, the method lowers the inference compute and energy of multi-view VLMs. The authors declare no competing interests.

REPRODUCIBILITY STATEMENT

We provide a complete description of SGPrune in Section 3 and Algorithms 1 and 2. The appendix details the **implementation environment**, **software versions** (Appendix A.3 and Table 13), **public model checkpoints and datasets** (Appendix A.3), **calibration-subset construction** (Appendix A.3), and **hyperparameter settings** (Section 4) and **sensitivity analyses** (Appendix A.2). All experiments use frozen public checkpoints and publicly available datasets, with links provided in the implementation details. The reported token budgets, evaluation protocols, and efficiency measurements are specified in Sections 4.1, 4.2, and 4.4, enabling reproduction of the experimental setup and results. An **anonymized implementation**, including scripts for calibration, subspace construction, inference, and evaluation, is provided as supplementary material.

REFERENCES

Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

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

Saeed Ranjbar Alvar, Gursimran Singh, Mohammad Akbari, and Yong Zhang. Divprune: Diversity-based visual token pruning for large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9392–9401. IEEE, 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*, 1(2):3, 2023.

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.

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*, 2024b.

DeepSeek-AI. Deepseek-v4.1-flash: Pushing the limits of kv cache compression, 2026.

Zhengyao Fang, Pengyuan Lyu, Chengquan Zhang, Guangming Lu, Jun Yu, and Wenjie Pei. Prune redundancy, preserve essence: Vision token compression in vlms via synergistic importance-diversity. *arXiv preprint arXiv:2603.09480*, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Zhijian Huang, Chengjian Fen, Feng Yan, Baihui Xiao, Zequn Jie, Yujie Zhong, Xiaodan Liang, and Lin Ma. Drivemm: All-in-one large multimodal model for autonomous driving. *arXiv preprint arXiv:2412.07689*, 2024.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702. IEEE, 2019.

Ayesha Ishaq, Jean Lahoud, Ketan More, Omkar Thawakar, Ritesh Thawkar, Dinura Dissanayake, Noor Ahsan, Yuhao Li, Fahad Shahbaz Khan, Hisham Cholakkal, et al. Drivelmm-o1: A step-by-step reasoning dataset and large multimodal model for driving scenario understanding. In *2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 20501–20508. IEEE, 2025.

Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. 2024. URL https://arxiv.org/abs/2410.22313.

Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *European conference on computer vision*, pp. 235–251. Springer, 2016.

Youngeun Kim, Youjia Zhang, Huiling Liu, Aecheon Jung, Sunwoo Lee, and Sungeun Hong. Zooprune: Training-free token pruning via zeroth-order gradient estimation in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39572–39582, 2026.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 conference on empirical methods in natural language processing*, pp. 292–305, 2023.

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.

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

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. Docvqa: A dataset for vqa on document images, 2021. URL https://arxiv.org/abs/2007.00398.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and CV Jawahar. Infographicvqa. In *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 2582–2591. IEEE, 2022.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *European conference on computer vision*, pp. 256–274. Springer, 2024.

Yuehao Song, Shaoyu Chen, Hao Gao, Yifan Zhu, Weixiang Yue, Jialv Zou, Bo Jiang, Zihao Lu, Yu Wang, Qian Zhang, et al. Senna-2: Aligning vlm and end-to-end driving policy for consistent decision making and planning. *arXiv preprint arXiv:2603.11219*, 2026.

Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. Omnidrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22442–22452. IEEE, 2025.

Minhao Xiong, Zichen Wen, Zhuangcheng Gu, Xuyang Liu, Rui Zhang, Hengrui Kang, Jiabing Yang, Junyuan Zhang, Weijia Li, Conghui He, et al. Prune2drive: A plug-and-play framework for accelerating vision-language models in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 25215–25224, 2026.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. Sigmoid loss for language image pre-training. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 11941–11952. IEEE, 2023.

Kaichen Zhang, Bo Li, Peiyuan Zhang, Fanyi Pu, Joshua Adrian Cahyono, Kairui Hu, Shuai Liu, Yuanhan Zhang, Jingkang Yang, Chunyuan Li, et al. Lmms-eval: Reality check on the evaluation of large multimodal models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 881–916, 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*, 2024.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, 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.

Qihui Zhu, Tao Zhang, Yuchen Wang, Zijian Wen, Mengjie Zhang, Shuangwu Chen, Xiaobin Tan, Jian Yang, Yang Liu, Zhenhua Dong, et al. Hawk: Head importance-aware visual token pruning in multimodal models. *arXiv preprint arXiv:2604.07812*, 2026.

A Appendix

A.1 Qualitative Analysis of View Allocation

Fig. 3 compares fixed and input-adaptive view allocation under the same total visual token budget. A fixed allocation assigns predetermined budgets to camera views regardless of the current scene. In Fig. 3 (a), the red-bordered views contain few prominent objects relevant to the question, yet receive substantial token budgets. This illustrates how a fixed allocation can devote a large share of the limited budget to views with relatively little question-relevant object information.

In contrast, SGPrune jointly selects tokens across views using relevance weights and feature diversity. The number of retained tokens in each view is determined by this joint selection rather than prescribed beforehand. As illustrated in Fig. 3 (b), SGPrune reallocates tokens toward views containing relevant objects and complementary scene context. This example illustrates how input-adaptive allocation can better match the distribution of useful visual information across views.



Figure 3: **Qualitative comparison of view-wise token allocation** under the same total visual token budget. (a) Fixed allocation assigns tokens to views using predefined ratios. (b) SGPrune jointly selects tokens across views, producing input-adaptive view allocation.

A.2 Additional Ablation Studies

Effect of view-specific subspaces. We compare the view-specific subspaces used in SGPrune with a single subspace shared across all views, fit on the pooled tokens of every view. As shown in Table 6, view-specific subspaces improve scores by 0.37 and 0.42 points on DriveLMM-o1 at 75.0% and 90.2% token reduction, and by 1.32 and 0.89 points on DriveMM at 75.3% and 90.1% reduction, respectively. The larger gap on DriveMM is consistent with its more heterogeneous views: three camera images and a BEV point-cloud rendering, whose token distributions a single shared subspace must cover at once, whereas the six views of DriveLMM-o1 are all camera images. These results support constructing separate subspaces to account for view-dependent token distributions.

Table 6: **Effect of view-specific subspaces.** We report Overall Reasoning for DriveLMM-o1 and the mean of FRM and QNS on MAPLM for DriveMM under matched total token budgets. The best results are bolded.

Variant	**DriveLMM-o1**		**DriveMM**	
Token reduction	*75.0%*	*90.2%*	*75.3%*	*90.1%*
Shared basis across views	76.74	76.35	68.09	66.52
SGPrune	**77.11**	**76.77**	**69.41**	**67.41**

Sensitivity to subspace construction hyperparameters. We examine the effects of the number of selected tokens per view and example, M, the total number of collected tokens per subspace, n_v, and the subspace dimension, c. We vary one parameter at a time while keeping the remaining parameters fixed. We use $\beta = 2$ for DriveLMM-o1 and DriveMM, and $\beta = 0.5$ for InternVL3-8B.

For DriveLMM-o1, Table 7 shows that $M = 50$, $n_v = 100{,}000$, and $c = 64$ achieve the highest mean score across token budgets in all three comparisons. Although $M = 25$ and $n_v = 150{,}000$ yield slightly higher scores at 90.2% reduction, their gains are only 0.03 and 0.09 points, respectively, with lower scores at 75.0% reduction. Subspace dimensions of 32 and 64 produce similar results, whereas increasing c to 128 reduces performance at both budgets.

For DriveMM, Table 8 shows that $M = 75$, $n_v = 100{,}000$, and $c = 64$ achieve the highest scores at both token budgets in all three comparisons. Increasing M from 50 to 75 yields modest improvements, but further increasing it to 100 reduces performance. Likewise, increasing n_v from 100,000 to 150,000 provides no additional gains. A subspace dimension of $c = 64$ outperforms both $c = 32$ and $c = 128$, indicating that retaining more principal components does not necessarily improve token selection.

For InternVL3-8B, Table 9 shows that $M = 25$ and $n_v = 100{,}000$ achieve the highest scores at both token budgets when $c = 128$. The preferred subspace dimension depends on the pruning budget: $c = 256$ performs best at 75.0% reduction, whereas $c = 128$ performs best at 90.2% reduction. The

Table 7: **Sensitivity to subspace construction hyperparameters on DriveLMM-o1.** We report Overall Reasoning with $\beta = 2$. Mean denotes the average across the two token budgets. Within each group, the best results are bolded and configurations with the highest Mean are shaded.

M	n_v	c	**Token reduction**		**Mean**
			75.0%	*90.2%*	
Varying M					
25	100,000	64	76.91	**76.80**	76.86
50	100,000	64	**77.11**	76.77	**76.94**
75	100,000	64	76.94	76.49	76.72
Varying n_v					
50	50,000	64	76.96	76.73	76.85
50	100,000	64	**77.11**	76.77	**76.94**
50	150,000	64	76.99	**76.86**	76.93
Varying c					
50	100,000	32	77.09	76.74	76.92
50	100,000	64	**77.11**	**76.77**	**76.94**
50	100,000	128	76.58	76.30	76.44

Table 8: **Sensitivity to subspace construction hyperparameters on MAPLM using DriveMM.** We report the mean of FRM and QNS with $\beta = 2$. Mean denotes the average across the two token budgets. Within each group, the best results are bolded and configurations with the highest Mean are shaded.

M	n_v	c	**Token reduction**		**Mean**
			75.3%	*90.1%*	
Varying M					
50	100,000	64	69.32	67.27	68.30
75	100,000	64	**69.41**	**67.41**	**68.41**
100	100,000	64	68.83	66.12	67.48
Varying n_v					
75	50,000	64	68.87	67.04	67.96
75	100,000	64	**69.41**	**67.41**	**68.41**
75	150,000	64	68.94	66.73	67.84
Varying c					
75	100,000	32	68.19	66.13	67.16
75	100,000	64	**69.41**	**67.41**	**68.41**
75	100,000	128	68.86	66.83	67.85

two dimensions yield the same mean score at the reported precision, while $c = 128$ uses a smaller basis.

Across the three models, increasing the number of collected tokens or the subspace dimension does not consistently improve performance. These results highlight the importance of evaluating subspace construction choices across different token budgets.

Sensitivity to relevance-weight strength. We examine the effect of β, which controls the spread of the rank-based relevance weights. Larger values increase the weight differences between high- and low-ranked tokens, placing greater emphasis on subspace relevance during diversity-aware selection. We fix the subspace construction settings for each model and vary only β.

For DriveLMM-o1, Table 10 shows that $\beta = 2$ achieves the highest scores at both token budgets. Compared with $\beta = 1$, it improves Overall Reasoning by 0.24 and 0.48 points at 75.0% and 90.2% token reduction, respectively. Increasing β further to 3 slightly reduces performance.

Table 9: **Sensitivity to subspace construction hyperparameters on InternVL3-8B.** Scores denote the mean of per-benchmark score ratios relative to the vanilla model, expressed as percentages. Mean denotes the average across the two token budgets. We fix $\beta = 0.5$. Within each group, the best results are bolded and configurations with the highest Mean are shaded.

M	n_v	c	**Token reduction**		**Mean**
			75.0%	*90.2%*	
Varying M					
10	100,000	128	82.70	70.82	76.76
25	100,000	128	**83.01**	**71.17**	**77.09**
50	100,000	128	82.72	70.46	76.59
Varying n_v					
25	50,000	128	82.60	71.15	76.88
25	100,000	128	**83.01**	**71.17**	**77.09**
25	150,000	128	82.80	70.96	76.88
Varying c					
25	100,000	64	82.26	70.51	76.39
25	100,000	128	83.01	**71.17**	**77.09**
25	100,000	256	**83.52**	70.66	**77.09**

Table 10: **Sensitivity to relevance-weight strength on DriveLMM-o1.** We report Overall Reasoning with $M = 50$, $n_v = 100{,}000$, and $c = 64$. Mean denotes the average across the two token budgets. The best results are bolded, and the row with the highest Mean is shaded.

β	**Token reduction**		**Mean**
	75.0%	*90.2%*	
1	76.87	76.29	76.58
2	**77.11**	**76.77**	**76.94**
3	77.06	76.55	76.81

For DriveMM, Table 11 shows that $\beta = 2$ also performs best at both budgets. At 90.1% token reduction, it exceeds $\beta = 1$ and $\beta = 3$ by 2.70 and 1.16 points, respectively. These differences are larger than those at 75.3% reduction, indicating greater sensitivity to relevance weighting under more aggressive pruning.

Table 11: **Sensitivity to relevance-weight strength on MAPLM using DriveMM.** We report the mean of FRM and QNS with $M = 75$, $n_v = 100{,}000$, and $c = 64$. Mean denotes the average across the two token budgets. The best results are bolded, and the row with the highest Mean is shaded.

β	**Token reduction**		**Mean**
	75.3%	*90.1%*	
1	68.13	64.71	66.42
2	**69.41**	**67.41**	**68.41**
3	68.57	66.25	67.41

For InternVL3-8B, Table 12 shows that $\beta = 0.5$ achieves the highest relative performance at both budgets. It outperforms $\beta = 0.25$ by 1.07 and 1.27 percentage points at 75.0% and 90.2% token reduction, respectively. Increasing β to 1 does not provide further improvements. Overall, the results support $\beta = 2$ for the driving models and $\beta = 0.5$ for InternVL3-8B among the evaluated settings.

Table 12: **Sensitivity to relevance-weight strength on InternVL3-8B.** Scores denote the mean of per-benchmark score ratios relative to the vanilla model, expressed as percentages. We fix $M = 25$, $n_v = 100{,}000$, and $c = 128$. Mean denotes the average across the two token budgets. The best results are bolded, and the row with the highest Mean is shaded.

β	Token reduction		Mean
	75.0%	*90.2%*	
0.25	81.94	69.90	75.92
0.5	**83.01**	**71.17**	**77.09**
1	82.91	70.64	76.78

A.3 Implementation Details

Software environment. The two driving backbones pin different PyTorch and transformers versions, so their experiments run in two Conda-managed environments; Table 13 lists the Python version and key package versions of each. The InternVL3-8B experiments use LMMs-Eval 0.7.2.

Table 13: **Software versions used for the driving experiments.**

Package	DriveLMM-o1	DriveMM
Python	3.10.20	3.10.20
PyTorch	2.13.0	2.1.2
torchvision	0.28.0	0.16.2
transformers	4.49.0	4.43.1
tokenizers	0.21.4	0.19.1
accelerate	1.14.0	1.14.0
flash-attn	2.8.3.post1	2.6.3
triton	3.7.1	2.1.0

Models. All experiments use frozen public checkpoints; no weights are trained or modified at any stage:

1. DriveLMM-o1
2. DriveMM
3. InternVL3-8B

Datasets. We construct a separate calibration subset D_{sub} for each benchmark and fit the corresponding PCA subspaces offline. For the driving benchmarks, we subsample frames at a fixed stride from the training split in its official ordering. Because the training data are ordered by recording, temporally adjacent frames show nearly the same scene, so a fixed stride covers a broader range of scenes than the same number of consecutive frames, while remaining fully deterministic. For DriveLMM-o1, we take every ninth (frame, question) pair. For DriveMM, we keep one question per frame, cycling through the four question categories, and take every fifth frame. From each pool we use, under a fixed seeded shuffle, the smallest subset that yields $n_v = 100{,}000$ vectors per view: 2,000 pairs ($M = 50$) for DriveLMM-o1 and 1,334 frames ($M = 75$) for DriveMM. For the general vision-language benchmarks, we build one basis per benchmark from up to 4,000 (image, question) pairs that do not overlap with the evaluated split.

For autonomous driving, we use DriveLMM-o1, whose multi-view images originate from nuScenes, and MAPLM. For general vision-language evaluation, we use GQA, POPE, ChartQA, DocVQA, AI2D, InfoVQA, and ScienceQA.

Algorithms. Algorithms 1 and 2 summarize the offline and online procedures of SGPrune using the notation of Section 3. The offline procedure combines attention-based token selection and PCA

subspace construction. The online procedure performs adaptive multi-view token pruning for each input using the current visual tokens and stored subspaces.

For DriveLMM-o1, the offline procedure, including attention-based token selection and PCA subspace construction for all views with $n_v = 100{,}000$ tokens per view, takes approximately 20 minutes on a single NVIDIA H200 GPU with 141 GB of memory. The resulting subspaces are reused during inference without reconstruction.

Algorithm 1 Offline attention-based token selection and PCA subspace construction

Require: Frozen VLM; calibration subset D_{sub}; decoder layer ℓ; number of selected heads H_s; tokens per view and example M; subspace dimension c

Ensure: View-specific basis matrices $\{\mathbf{U}_v\}_{v=1}^{V}$

for each view v **do**
  $\mathbf{X}_v \leftarrow$ empty matrix with d columns
end for
for each example $x \in D_{sub}$ **do**
  Obtain pre-projector visual tokens $\{\mathbf{F}^v(x)\}_{v=1}^{V}$ from the input images
  $P \leftarrow \{(v,i) : 1 \le v \le V,\ 1 \le i \le N_v\}$
  $\mathbf{z}_i^v \leftarrow$ projector output for $\mathbf{f}_i^v(x)$, for each $(v,i) \in P$
  Build the multimodal sequence containing all views, the question, and answer options if provided
  $C \leftarrow$ indices of candidate question and option tokens
  $\mathbf{t}_j \leftarrow$ input embedding of candidate text token j, for each $j \in C$
  for each candidate text token $j \in C$ **do**
    $$a_j \leftarrow \frac{1}{|P|} \sum_{(v,i)\in P} \frac{\exp((\mathbf{z}_i^v)^\top \mathbf{t}_j)}{\sum_{k\in C} \exp((\mathbf{z}_i^v)^\top \mathbf{t}_k)}$$
  end for
  $$R \leftarrow \left\{ j \in C : a_j > \tfrac{1}{|C|} \textstyle\sum_{k\in C} a_k \right\}$$
  Run the decoder through layer ℓ, extract attention weights $A^{(\ell)}$, and correct the positional bias: $\tilde{A}^{(\ell)}_{h,p,p'} \leftarrow A^{(\ell)}_{h,p,p'} / \Pi_{p,p'}$
  $H \leftarrow$ indices of the H_s heads with the largest total corrected attention from R to visual tokens in P
  for each view v **do**
    for each visual token $i = 1, \ldots, N_v$ **do**
      $$s_i^v(x) \leftarrow \frac{1}{|H||R|} \sum_{h\in H} \sum_{r\in R} \tilde{A}^{(\ell)}_{h,\rho(r),p(v,i)}$$
    end for
    $I^v(x) \leftarrow \mathrm{TopK}(\mathbf{s}^v(x), M)$
    Append $\mathbf{f}_i^v(x)^\top$ to $\mathbf{X}_v$ for each $i \in I^v(x)$
  end for
end for
for each view v **do**
  $n_v \leftarrow$ number of rows in $\mathbf{X}_v$
  $\boldsymbol{\mu}_v \leftarrow$ mean token vector of $\mathbf{X}_v$
  $(\mathbf{L}_v, \mathbf{\Sigma}_v, \mathbf{R}_v^\top) \leftarrow \mathrm{SVD}(\mathbf{X}_v - \mathbf{1}_{n_v} \boldsymbol{\mu}_v^\top)$
  $\mathbf{U}_v \leftarrow [\mathbf{r}_{v,1}, \ldots, \mathbf{r}_{v,c}]^\top$ ▷ Leading c right singular vectors
end for
return $\{\mathbf{U}_v\}_{v=1}^{V}$

In Algorithm 2, the running minimum m_q equals $d_q^{(t)}$ after the distance update at iteration t. Relevance weights remain fixed for each input, while minimum distances are updated after each selected token. The resulting view budgets satisfy $\sum_{v=1}^{V} K_v = K$.

Algorithm 2 Online adaptive multi-view token pruning

Require: Pre-projector visual tokens $\{\mathbf{F}^v\}_{v=1}^{V}$; basis matrices $\{\mathbf{U}_v\}_{v=1}^{V}$; relevance-weight strength β; total token budget K; numerical constant $\epsilon > 0$
Ensure: Selected token index set S_K with $|S_K| = K$

$P \leftarrow \{(v, i) : 1 \le v \le V,\ 1 \le i \le N_v\}$
for each candidate $q = (v, i) \in P$ **do**
  $\mathbf{f}_q \leftarrow \mathbf{f}_i^v$
  $e_q \leftarrow \dfrac{\|\mathbf{U}_v \mathbf{f}_q\|_2}{\max(\|\mathbf{f}_q\|_2, \epsilon)}$
end for
$r_q \leftarrow \mathrm{rank}_P(e_q)$ for each $q \in P$ ▷ Ascending ranks from 0 to $|P| - 1$
$g_q \leftarrow \exp\left[\beta \left(\frac{r_q}{|P|-1} - \frac{1}{2}\right)\right]$ for each $q \in P$
$m_q \leftarrow +\infty$ for each $q \in P$ ▷ Running minimum cosine distance
Choose $q_1 \in \arg\max_{q \in P} g_q$
$S_1 \leftarrow \{q_1\}$
for $t = 2, \ldots, K$ **do**
  for each candidate $q \in P \setminus S_{t-1}$ **do**
    $m_q \leftarrow \min\big(m_q,\ 1 - \cos(\mathbf{f}_q, \mathbf{f}_{q_{t-1}})\big)$
  end for
  Choose $q_t \in \arg\max_{q \in P \setminus S_{t-1}} g_q m_q$
  $S_t \leftarrow S_{t-1} \cup \{q_t\}$
end for
for each view v **do**
  $K_v \leftarrow |\{i : (v, i) \in S_K\}|$
  Retain the selected tokens of view v in their original order
end for
Pass retained tokens to the multimodal projector and insert them into their corresponding view slots, preserving view ordering and image delimiters
return S_K

A.4 Qualitative Visualization

To better understand the performance gains of SGPrune, we qualitatively visualize token selection results on DriveLMM-o1. We compare SGPrune with HAWK (Zhu et al., 2026), an attention-based baseline, and Prune2Drive (Xiong et al., 2026), a diversity-based baseline, under the same total token budget. To extend HAWK to the multi-view setting, we pool visual tokens from all six views and select the global top-K tokens according to its attention scores.

As shown in Fig. 4, HAWK primarily retains tokens around salient objects, but often provides limited coverage of the surrounding context. Prune2Drive preserves more diverse regions across the scene, but can allocate insufficient tokens to question-relevant objects. Moreover, its view-wise pruning ratios are fixed using the training set, which can allocate many tokens to views with limited relevant information for a given input. In contrast, SGPrune preserves both question-relevant objects and their complementary local context, as highlighted by the green boxes in Fig. 4. By jointly selecting tokens across all views under a shared budget, SGPrune also enables input-adaptive allocation of visual tokens across views.

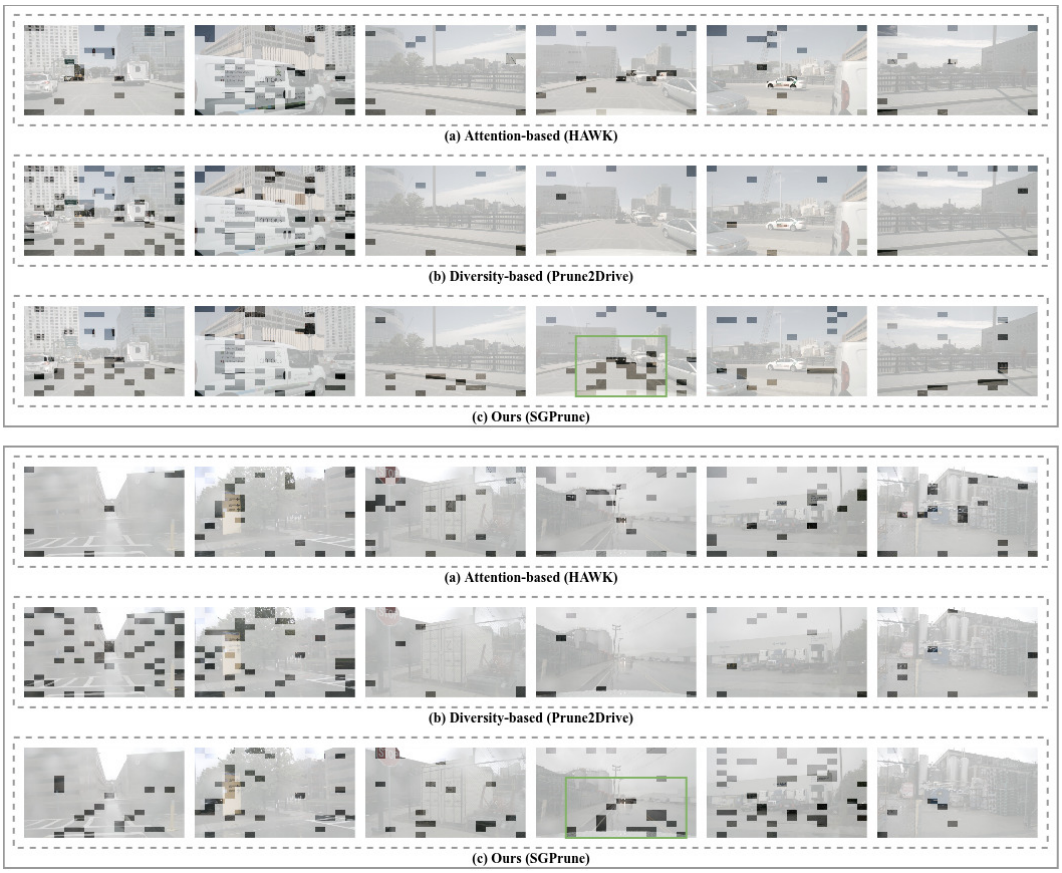


Figure 4: **Qualitative comparison of multi-view visual token selection on DriveLMM-o1.** The top and bottom groups show two representative six-view driving examples under the same total token budget. Faded regions indicate discarded visual tokens. (a) HAWK concentrates tokens on salient objects but provides limited coverage of surrounding context. (b) Prune2Drive preserves diverse visual regions but can miss question-relevant objects. (c) SGPrune jointly considers relevance and diversity, retaining both question-relevant objects and complementary scene context. Green boxes highlight question-relevant objects and their complementary local context preserved by SGPrune.