MVPruner: Dynamic Token Pruning for Accelerating Multi-view Vision-Language Models in Autonomous Driving

Nan Yang[1], Zhanwen Liu[1*], Linfeng Zhang[2], Shangyu Xie[1], Yang Wang[1], Wenzhuo Zhou[1], and Xiangmo Zhao[1]

[1] School of Information Engineering, Chang'an University, China
[2] School of Artificial Intelligence, Shanghai Jiaotong University, China
{ynan,zwliu,ywang120}@chd.edu.cn zhanglinfeng@sjtu.edu.cn

Abstract. Vision-Language Models (VLMs) improve generalization and interpretability in autonomous driving but suffer from efficiency issues due to long visual token sequences, particularly in standard multi-view settings. Existing token pruning methods employ fixed pruning rate allocation and static importance metrics, ignoring dynamic inter-view importance differences and the evolving information importance during inference. Our analysis reveals that multi-view VLMs inherently encode task-related view priors in deeper layers and exhibit dynamic information requirements. Motivated by these findings, we propose MVPruner, a two-stage adaptive token pruning method that aligns pruning behavior with the model's dynamic information requirements. The first stage allocates pruning budgets based on the information diversity of each view, and retains tokens with consistent contribution across stages, ensuring semantic representational capacity. The second stage allocates budgets and selects tokens guided by instruction text to guarantee task alignment. Experimental results on four benchmarks demonstrate the superior performance of our method. For example, DriveMM equipped with MVPruner achieves 87.3% reduction in FLOPs, 4.97× speedup in prefilling phase while retaining 98.5% accuracy on DriveLM benchmark.

Keywords: VLMs · Token pruning · Autonomous driving

1 Introduction

Vision-Language Models (VLMs) [3, 18, 20] have shown promise for achieving interpretable and generalizable perception and decision-making in autonomous driving [12, 15, 22, 28, 29, 36]. However, to ensure comprehensive environmental perception, autonomous driving systems typically rely on multi-view camera inputs [4,13,21,26,27]. The high-resolution multi-view images inevitably generate excessive visual token sequences, leading to substantial computational overhead and inference latency [7, 35], which severely hinders the deployment in latency-sensitive driving scenarios [33].

* Corresponding author.

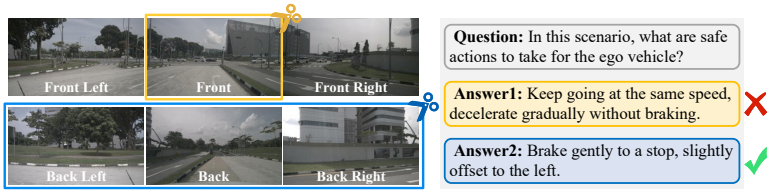

Fig. 1: Removing three unimportant views does not affect the decision, while removing only the critical front view leads to an incorrect decision, highlighting the inconsistent contributions across views.

To improve inference efficiency, visual token pruning has emerged as an effective strategy for eliminating redundant information [25,30,34]. Although pruning techniques for VLMs have achieved preliminary progress, existing methods still face several critical challenges in multi-view autonomous driving scenarios: (1) Most existing methods are designed for single-view settings and apply uniform pruning ratios across views when handling multi-view inputs, ignoring the varying contributions of different views [2,7,35]. Fig. 1 illustrates this effect: in braking scenarios triggered by a red light ahead, removing rear views does not affect the decision, whereas pruning the front view containing the key red light cue leads to incorrect and unsafe behavior. Although recent work (*e.g.*, Prune2Drive [33]) allocates view-specific pruning budgets via offline hyperparameter search, such static strategies are sensitive to scene variations and cannot adapt to varying driving tasks and instructions. (2) Widely used token pruning methods rely on static importance evaluation metrics or single-stage pruning strategies, failing to account for the model's dynamic information requirements and consequently resulting in suboptimal preservation of multi-view information.

To address these challenges, we first design a task-related view recognition experiment to quantitatively evaluate the ability of multi-view VLMs to recognize critical views across different layers. As shown in Fig. 2(a), the accuracy is low in shallow layers, increases substantially in intermediate layers where it reaches its peak. This result indicates that **VLMs inherently encode task-related view priors during deeper inference**. Furthermore, we investigate the underlying mechanisms behind this layer-wise discrepancy. As shown in Figs. 2(b) and (c), **VLMs exhibit hierarchical attention patterns** that shift from attending to broad visual context for global scene understanding to focusing on task-relevant local regions, and **dynamic information requirements** that transition from semantic diversity-driven processing to task relevance-driven processing.

The above findings provide principled guidance on *which stages to prune, how to allocate pruning budgets, and which tokens to retain*. Based on these insights, we propose **MVPruner**, a two-stage adaptive token pruning framework for multi-view VLMs that aligns budget allocation and token selection with the model's dynamic information requirements. In the first stage, inspired by the model's broad attention to all visual information in shallow layers, we design the Diversity-aware Ratio Allocation strategy that measures intra-view feature di-

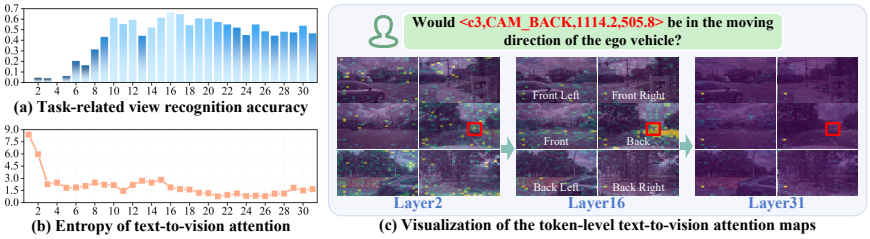

Fig. 2: (a) Task-related view recognition accuracy across layers. Higher accuracy indicates clearer identification of task-related views at this layer. **(b) Entropy of text-to-vision attention.** High entropy reflects diffuse global attention, while low entropy indicates concentrated local focus. **(c) Visualization of text-to-vision attention maps across layers.** Details are provided in Sec. 3.2.

versity to assess view importance and allocate larger budgets to information-rich views. Meanwhile, a Cross-stage Contribution-aware Token Selection mechanism is proposed to retain tokens that consistently contribute across different layers, satisfying the dynamic information requirements during inference. In the second stage, the textual instructions can effectively guide view-importance estimation in deeper layers. Therefore, we introduce the Instruction-aware Ratio Allocation strategy that measures semantic relevance between visual and instruction tokens, assigning larger budgets to task-related views. Finally, we select tokens guided by instruction text to enhance task-alignment capabilities. Experiments on the image-based multi-view autonomous driving benchmarks: DriveLMM-o1 [13], DriveLM [26], and MAPLM [5], as well as video-based benchmark STSnu [10] demonstrate that our method can consistently accelerate inference while effectively maintaining reasoning accuracy, exhibiting strong generalization in autonomous driving. Our contribution can be summarized as follows:

(1) Our analysis reveals that multi-view VLMs allocate more pronounced attention to task-related views in deeper layers and exhibit dynamic information requirements, providing insights to guide method design.

(2) We propose MVPruner, which dynamically adjusts pruning ratios across different views in an online manner and adaptively retains important tokens according to the model's dynamic information requirements.

(3) Experimental results demonstrate that our method achieves a superior efficiency–performance trade-off. With 10% of visual tokens retained, our method maintains accuracy of 98.0%, 98.5%, and 94.5%, outperforming the second-best method by 2.6%, 0.8% and 3.5%, achieving 3.80×, 4.97× and 3.95× speedup in the prefilling stage on DriveLMM-o1, DriveLM and MAPLM, respectively.

2 Related Work

2.1 VLMs for Autonomous Driving

By jointly modeling visual inputs and language instructions within a shared embedding space, VLMs enhance reasoning, interpretability, and generalization

in autonomous driving, supporting complex driving tasks [6, 14, 16, 17, 24, 36]. Early efforts including GPT-Driver [22], demonstrate that frozen VLMs can process driving tasks while simultaneously producing human-readable rationales. Recently, DriveLM [26] and DriveLMM-o1 [13] have respectively introduced a graph-based visual question answering dataset and a step-by-step reasoning dataset to enhance the driving-scene understanding of VLMs. DriveMM [11] designs a general VLM to process diverse data inputs and perform a broad spectrum of tasks. However, the long visual token sequences, especially for multi-view inputs, lead to significant inference latency and limit their practical applications.

2.2 Visual Token Pruning for VLMs

Recently, token pruning methods have attracted increasing attention as an effective acceleration strategy for VLMs [23, 30]. These methods are generally categorized into two groups [25]: (1) attention-based methods, which prune unimportant tokens via attention scores [7, 19, 32, 35]; (2) similarity-based methods [1, 2, 31], which either merge similar tokens or maximize diversity of retained tokens. However, they mainly focus on single-view inputs and overlook the varying redundancy across views. Although recent work, Prune2Drive [33], allocates pruning budgets across views through offline hyperparameter search, it incurs additional computational overhead and exhibits limited generalization capability. Moreover, these methods rely on static importance metrics or single-stage pruning, overlooking the model's cross-layer dynamic information requirements.

3 Method

3.1 Multi-view Vision-Language Models

A multi-view VLM [13, 26] typically processes a pair of inputs, denoted as (V, T), where V is the multi-view image input and T is the text input which is composed of system text and instruction text. The visual input $V \in \mathbb{R}^{P\times H\times W\times 3}$ is converted to visual tokens $X^v \in \mathbb{R}^{P\times M\times D}$ using a vision encoder and a projector layer, where P is the number of views, M is the number of tokens within each view and D is the dimension of the feature. The text input is mapped to textual tokens $X^t \in \mathbb{R}^{N\times D}$ using a text encoder, where N is the number of textual tokens. Then, they are fed to an LLM to generate the prediction.

3.2 Key Observations

In autonomous driving tasks, textual instructions often exhibit explicit spatial specificity (*e.g.*, "Would the object in the back-view camera be in the moving direction?"). These samples enable us to quantitatively analyze whether multi-view VLMs can inherently identify instruction-referenced views. To investigate this, we design a task-related view recognition experiment. Specifically, we evaluate two representative models, DriveMM [11] and DriveLMM-o1 [13], by extracting

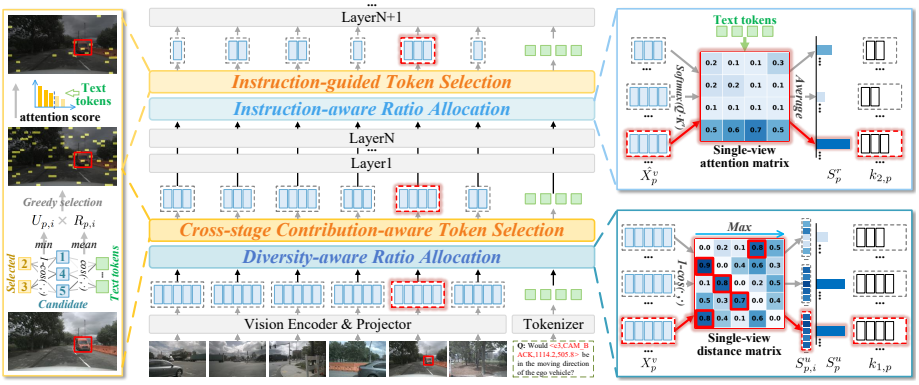

Fig. 3: Overall framework of MVPruner. In the shallow layer, MVPruner adjusts budget based on intra-view information diversity and selects tokens that consistently contribute across different layers. In the deeper layer, MVPruner allocates the budget via the semantic similarity between each view and the instruction text, and retains tokens with high cross-modal attention scores.

view-specific subsets from the DriveLM [26] and DriveLMM-o1 datasets. For each decoder layer, we compute the cross-modal attention scores between the instruction and visual tokens of each view. An identification is deemed correct if the view yielding the maximum score aligns with the instruction-referenced view. As shown in Fig. 2(a), as the layers deepen, the accuracy gradually increases and reaches its peak in the middle layers. This finding indicates that **cross-modal attention patterns in the deeper layers of VLMs effectively identify task-relevant views**.

Furthermore, to examine layer-wise differences in recognition accuracy, we analyze the cross-layer evolution of attention patterns between visual and textual modalities. Specifically, we compute the entropy of text-to-vision attention at each layer to measure the distribution of attended visual information. As shown in Fig. 2(b), entropy is initially high, decreases rapidly with fluctuations, and further declines in deep layers. This trend reflects a shift in the model's attention patterns, consistent with the attention maps in Fig. 2(c). **In early layers**, textual semantics interact with broad, global visual context to establish initial scene understanding and coarse cross-modal alignment. **In intermediate layers**, the model progressively concentrates on task-relevant visual regions. **In deep layers**, visual information is further aggregated and exhibits consistently high attention at fixed spatial locations within each view. This finding suggests that **reasoning in VLMs exhibits dynamic information requirements that transition from semantic diversity-driven to task relevance-driven.**

3.3 Framework Overview

Inspired by the above findings, we propose MVPruner, a two-stage pruning framework that transitions from structural summarization to semantic focus-

ing, aligning with the model's dynamic information requirements, as shown in Fig. 3. Specifically, in the first stage, the Diversity-aware Ratio Allocation strategy is employed in the shallow layer to assign pruning rates based on differences in information diversity across views. Meanwhile, the Cross-stage Contribution-aware Token Selection mechanism retains tokens that consistently contribute across different stages. In the second stage, the Instruction-aware Ratio Allocation strategy is applied in the deeper layer, where pruning rates are determined by the task relevance of each view. Finally, redundant tokens are further removed guided by instruction.

3.4 Diversity-aware Ratio Allocation

In shallow layers, attention is relatively uniformly distributed across all views, reflecting the reasoning process driven by semantic diversity. This observation motivates us to start from the information richness within each view, preserving more information for views containing more diverse content, supplying the deeper layers with a more comprehensive visual context. To this end, we propose the Diversity-aware Ratio Allocation (DRA) strategy that dynamically assigns the pruning budget based on the measured feature diversity of each view, maximizing the diversity and completeness of visual representations during compression.

Firstly, we compute the semantic distances between all token pairs $\left(X^v_{p,i}, X^v_{p,j}\right)$ within each view p, $(i,j) \in M, i \neq j$ to estimate feature diversity. The distance $d_p(\cdot,\cdot)$ is defined as:

$$d_p\left(X^v_{p,i}, X^v_{p,j}\right) = 1 - \frac{X^v_{p,i} \cdot X^v_{p,j}}{\left\|X^v_{p,i}\right\| \left\|X^v_{p,j}\right\|}. \tag{1}$$

For each token, the minimum pairwise distance is used as its uniqueness score $S^u_{p,i}$, where a higher score indicates that the token is more unique relative to others in the same view:

$$S^u_{p,i} = min\left(d_p(X^v_{p,i}, X^v_{p,1:M})\right). \tag{2}$$

Subsequently, the diversity score of a view $S^u_p = \frac{1}{M}\sum_{i=1}^{M} S^u_{p,i}$ is obtained by averaging the uniqueness scores of all its tokens, with a higher score indicating a greater feature density compared to other views. The retention ratio $r_{1,p}$ and number $k_{1,p}$ for each view is determined based on the normalized S^u_p:

$$r_{1,p} = r_1 \times P \times S^u_p, \quad k_{1,p} = r_{1,p} \times M, \tag{3}$$

where r_1 is the average retention ratio for the first-stage pruning.

3.5 Cross-stage Contribution-aware Token Selection

Tokens retained in shallow layers should support task-agnostic global semantic modeling while providing a foundation for task-relevant representation extraction in deeper layers. To this end, we propose the Cross-stage Contribution-aware

Algorithm 1 Cross-stage Contribution-aware Token Selection

Require: Visual tokens $X_p^v \in \mathbb{R}^{M \times D}$ of view p, candidate index set $\mathcal{C} \subseteq \{1, \ldots, M\}$, target token number $\mathcal{K} = k_{1,p}$
Ensure: Selected index set $\mathcal{S}$, with $|\mathcal{S}| = \mathcal{K}$
Phase 1: Initialization
Compute distance matrix: $D_{p,i,j} \leftarrow d_p\left(X_{p,i}^v, X_{p,j}^v\right)$
Compute semantic relevance: $R_{p,i} \leftarrow cosine\left(X_{p,i}^v, X_I^t\right)$
Select token with the highest semantic relevance: $i^\star \leftarrow \arg\max(R_{p,i})$
Initialize selected token list: $\mathcal{S} \leftarrow \{i^\star\}$
Phase 2: Iteratively add the subsequent tokens
while $|\mathcal{S}| < \mathcal{K}$ **do**
  for each $j \in \mathcal{C} \setminus \mathcal{S}$ **do**
    Uniqueness score: $U_{p,j} \leftarrow \min_{i \in \mathcal{S}} D_{p,i,j}$
    Cross-stage contribution: $Imp_{p,j} \leftarrow U_{p,j} \times R_{p,j}$
  end for
  $j^\star \leftarrow \arg\max_{j \in \mathcal{C} \setminus \mathcal{S}}(Imp_{p,j})$
  $\mathcal{S} \leftarrow \mathcal{S} \cup \{j^\star\}$
end while
return $\mathcal{S}$

Token Selection (CCTS) mechanism that jointly evaluates a token's immediate contribution to the current stage and its potential contribution to subsequent stages, aligning selection with the model's dynamic information requirements.

Specifically, we formulate the token selection in shallow layers as an optimization problem aimed at maximizing cross-stage contribution. In this formulation, immediate contribution is quantified by semantic uniqueness within visual space, while future contribution is estimated through task relevance which is computed as the cosine similarity between visual and instruction tokens. An iterative greedy selection strategy is then employed to efficiently search for the optimal token subset.

Firstly, the token with the highest task relevance is selected to initialize the set of selected tokens. Subsequently, the minimum semantic distance between a candidate token and the selected set is computed using Eq. (2) and used as the semantic uniqueness metric, denoted by $U_{p,i}$. This metric is further weighted by the task relevance score $R_{p,i}$, enabling a comprehensive assessment of the token's cross-stage contribution $Imp_{p,i}$:

$$Imp_{p,i} = U_{p,i} \times R_{p,i}. \tag{4}$$

After that, the token with the highest cross-stage contribution is appended to the selected set. By iterating this process until the target token budget is reached, this mechanism maximizes cross-stage information utility. The algorithm flow is detailed in Algorithm 1.

3.6 Instruction-aware Ratio Allocation

In deeper layers, attention is focused on local task-related regions, and the model can distinguish the task relevance of different views, reflecting the reasoning process driven by task relevance. Motivated by this, we design the Instruction-aware Ratio Allocation (IRA) strategy for the second stage, which adaptively allocates pruning budget based on semantic relevance between visual and instruction tokens, retaining more critical information for task-related views.

Specifically, we extract the attention matrix between the visual tokens and textual instruction tokens, and compute the mean attention score of all tokens within each view as its semantic relevance score S_p^r. A higher score indicates that the view exhibits greater task relevance than other views:

$$S_{p,i}^r = \frac{1}{\tilde{N}} \sum_{j=1}^{\tilde{N}} A_{i,j}^p, \quad S_p^r = \frac{1}{k_{1,p}} \sum_{i=1}^{k_{1,p}} S_{p,i}^r, \tag{5}$$

where $\tilde{N}$ is number of instruction tokens, $S_{p,i}^r$ is attention score of i-th token in view p, A^p is attention matrix between view p and instruction tokens.

After that, we reallocate the remaining token budget across views using a compensatory normalization scheme:

$$r_{2,p} = r_2 \times \left(1 + S_p^r - S_p^u\right), \quad k_{2,p} = r_{2,p} \times k_{1,p}, \tag{6}$$

where r_2 denotes the average retention ratio for the second-stage pruning, and $S_p^r - S_p^u$ represents the relative deviation from the diversity score. $r_{2,p}$ and $k_{2,p}$ are the retention ratio and number for each view.

Finally, the top $k_{2,p}$ tokens with the highest attention score $S_{p,i}^r$ to instruction tokens are selected to further enhance the task-alignment capability.

4 Experiments

This section first outlines the experimental setup, followed by a comparative evaluation against state-of-the-art (SOTA) methods. We then conduct ablation studies to validate the effectiveness of the proposed method. Finally, visualization results are presented to demonstrate its adaptability.

4.1 Experimental Setup

Models. We verify the effectiveness of the proposed method by experiment on two specialist autonomous driving visual-language models: DriveMM [11] and DriveLMM-o1 [13]. Specifically, DriveMM is built based on LLaVA-OneVision-7B [18] and fine-tuned across various autonomous driving datasets, can process diverse data inputs and driving tasks. DriveLMM-o1 is built based on InternVL2.5-8B [8] and fine-tuned in the DriveLMM-o1 dataset to enhance step-by-step reasoning ability in driving scenarios.

Table 1: Comparison with SOTA methods on DriveLMM-o1 using DriveLMM-o1 model.

Methods	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	73.81	82.26	75.64	79.51	71.38	74.37
Retain 64 Tokens per image (↓ 75%)						
FastV (ECCV24)	71.35	80.58	72.85	76.74	69.25	72.41 (97.4%)
DART (EMNLP25)	71.50	78.21	72.80	77.33	69.63	71.46 (96.1%)
SparseVLM (ICML25)	70.08	79.59	71.87	75.66	67.43	71.23 (95.8%)
PACT (CVPR25)	71.20	78.11	71.44	77.21	68.95	70.95 (95.4%)
DivPrune (CVPR25)	71.66	78.39	73.04	77.45	69.92	71.67 (96.4%)
Prune2Drive (CVPR26)	71.31	80.74	73.07	77.48	69.30	72.69 (97.7%)
Ours	**72.91**	**81.96**	**74.85**	**78.55**	**71.58**	**73.79 (99.2%)**
Retain 25 Tokens per image (↓ 90%)						
FastV (ECCV24)	67.74	78.07	69.37	73.24	66.02	69.56 (93.5%)
DART (EMNLP25)	68.60	75.81	69.97	74.94	67.60	69.01 (92.8%)
SparseVLM (ICML25)	68.18	77.81	69.87	73.98	66.20	69.40 (93.3%)
PACT (CVPR25)	68.20	76.23	69.11	74.14	66.10	68.38 (91.9%)
DivPrune (CVPR25)	69.32	76.47	70.78	75.53	68.22	69.65 (93.7%)
Prune2Drive (CVPR26)	70.76	77.66	72.29	77.01	68.94	70.31 (94.5%)
Ours	**72.05**	**81.81**	**73.80**	**77.61**	**70.25**	**72.91 (98.0%)**

Benchmarks. We evaluate our method on three large-scale image-based autonomous driving VQA benchmarks: DriveLM [26], DriveLMM-o1 [13], and MAPLM [5], as well as the video-based benchmark STSnu [10]. DriveLM is annotated on the NuScenes [4] dataset, including QA pairs of perception, prediction, and planning tasks with six-view images. DriveLMM-o1 is also annotated on NuScenes to advance step-by-step visual reasoning for autonomous driving with six-view images as input. MAPLM is tailored for advanced map and traffic scene understanding, supporting tasks such as scene classification, lane counting, and point cloud quality analysis based on three-view images and synchronized LiDAR point cloud maps. STSnu is proposed to evaluate the spatio-temporal reasoning capabilities of VLMs based on six-view videos. We follow the common metrics in each benchmark for fair comparison. Details are in the supplementary materials.

Evaluation Metrics. We follow the common metrics in each benchmark for fair comparison: (1) DriveLM [26] implements four evaluation metrics: accuracy, GPT score, language-based evaluation (BLEU-4, Rouge, and CIDEr), and match score. (2) DriveLMM-o1 [13] adopts five driving-specific reasoning evaluation metrics: Risk Assessment Accuracy, Traffic Rule Adherence, Scene Awareness and Object Understanding, Relevance, and Missing Details, as well as Overall Reasoning. (3) MAPLM [5] uses FRM (Frame-overall-accuracy) and QNS (Question-overall-accuracy). (4) STSnu [10] categorizes questions into four interaction types and evaluates them separately: (i) Ego: Which of the following options best describes the ego vehicle's driving maneuver? (ii) Agent: Which of the following options best describes the driving behavior of the <reference to agent>? (iii) Ego-to-agent: Which of the following options best describes the ego vehicle's driving behavior with respect to the <reference to agent>? (iv) Agent-to-agent: Which of the following options best describes <reference to agent 1>'s maneuver with respect to <reference to agent 2>?

Table 2: Comparison with SOTA methods on DriveLM using DriveMM model.

Model	Accuracy	Chatgpt	BLEU-4	Rouge	CIDEr	Match	Average
Upper Bound, 729 Tokens per image (100%)							
DriveMM	0.81	65.44	0.61	0.74	0.19	33.9	59.1
Retain 180 Tokens per image (↓ 75%)							
FastV	0.77	63.95	0.54	0.72	0.15	32.8	56.6(95.8%)
DART	0.78	64.02	0.56	0.73	0.18	33.6	57.2(96.8%)
SparseVLM	0.79	63.74	0.57	0.73	0.19	33.4	57.4(97.1%)
PACT	0.78	64.24	0.56	0.73	0.19	33.3	57.3(97.0%)
DivPrune	0.79	64.67	0.59	0.74	0.21	33.7	57.9(98.0%)
Prune2Drive	**0.80**	64.92	0.60	**0.75**	0.20	34.0	58.3(98.6%)
Ours	0.79	**65.52**	**0.61**	**0.75**	**0.22**	**34.3**	**58.7(99.3%)**
Retain 72 Tokens per image (↓ 90%)							
FastV	0.68	63.52	0.49	0.71	0.08	32.3	54.1(91.5%)
DART	0.69	64.20	0.51	0.71	0.12	32.0	54.7(92.6%)
SparseVLM	0.75	63.38	0.52	0.72	0.15	32.9	55.9(94.6%)
PACT	0.76	64.76	0.54	0.73	0.15	32.9	56.8(96.1%)
DivPrune	0.78	64.16	0.55	0.73	**0.17**	32.7	57.0(96.4%)
Prune2Drive	0.78	64.52	0.56	**0.74**	0.16	33.4	57.4(97.1%)
Ours	**0.79**	**65.41**	**0.58**	**0.74**	**0.17**	**34.0**	**58.2(98.5%)**

Comparison Methods. We compare our method with several recent representative token pruning methods, including FastV [7], DART [31], SparseVLM [35], PACT [9], and DivPrune [1]; as well as Prune2Drive [33] which is specifically designed for multi-view VLMs. To ensure a fair comparison, we optimized the pruning configuration for Prune2Drive on each benchmark using its official open-source implementation. To assess pruning robustness, we evaluate all methods under two widely adopted pruning settings, where on average 75% and 90% of visual tokens are pruned. The pruning ratio is computed as the average token reduction across all layers of the LLM, following prior works [1,7,9,31,33,35]. Additional experiments under more pruning settings are provided in the supplementary material.

Implementation Details. Our two-stage pruning is applied before layers 0 and 16, with average pruning ratios of r and r^2, causing an exponential reduction.

4.2 Main Results

This section presents comparative results between our method and SOTA methods on DriveLMM-o1, DriveLM, MAPLM and STSnu benchmarks.

DriveLMM-o1. The results are presented in Tab. 1. Our method retains 98.0% and 99.2% of the original performance using only 10% and 25% of the visual tokens, respectively, outperforming SOTA methods. Notably, under the 90% pruning setting, our method outperforms the second-best method Prune2Drive by a substantial margin of 2.6 on the Overall Reasoning metric, without any offline hyperparameter search.

DriveLM. The results in Tab. 2 show that our method retains near-original performance with only 10% and 25% tokens, achieving 98.5% and 99.3% of the original model, respectively. Notably, it even surpasses the original model on

Table 3: Comparison with SOTA methods on MAPLM using DriveMM model.

Model	FRM	QNS	Average	FRM	QNS	Average
Upper Bound, 729 Tokens per image (100%)						
DriveMM	58.60	85.87	72.24	58.60	85.87	72.24
	Retain 180 Tokens per image (↓ 75%)			*Retain 72 Tokens per image* (↓ 90%)		
FastV	47.67	81.70	64.69(89.5%)	33.07	76.15	54.61(75.6%)
DART	51.27	83.27	67.27(93.1%)	40.27	79.23	59.75(82.7%)
SparseVLM	49.80	82.52	66.16(91.6%)	35.67	77.13	56.40(78.1%)
PACT	50.87	82.50	66.69(92.3%)	30.87	74.78	52.83(73.1%)
DivPrune	53.46	83.10	68.28(94.5%)	48.80	80.75	64.78(89.7%)
Prune2Drive	53.47	83.10	68.29(94.5%)	48.33	81.28	64.81(89.7%)
Ours	**56.07**	**84.55**	**70.31(97.3%)**	**53.27**	**83.33**	**68.30(94.5%)**

Table 4: Comparison with SOTA methods on STSnu using DriveMM model.

Methods	Ego	Ego-to-Agent	Agent	Agent-to-Agent	Average
Upper Bound, 196 Tokens per image (100%)					
DriveMM	43.16	59.33	43.70	30.67	44.21
Retain 49 Tokens per image (↓ 75%)					
FastV	40.02	**63.08**	42.72	26.81	43.16(97.6%)
DART	38.70	58.32	**44.24**	28.79	42.51(93.9%)
SparseVLM	31.04	59.62	43.29	**32.89**	41.71(94.3%)
PACT	31.28	59.98	42.79	31.63	41.42(93.7%)
DivPrune	36.65	58.16	42.60	29.82	41.81(94.6%)
Prune2Drive	36.65	58.52	43.04	30.51	42.18(95.4%)
Ours	**44.59**	58.34	43.08	30.99	**44.25(100%)**
Retain 19 Tokens per image (↓ 90%)					
FastV	19.55	61.90	38.33	28.19	36.99(83.7%)
DART	15.41	37.76	36.08	26.35	28.90(65.4%)
SparseVLM	26.06	69.18	40.30	29.16	41.18(93.1%)
PACT	15.86	65.96	39.00	27.79	37.15(84.0%)
DivPrune	30.75	59.50	**43.79**	30.95	41.25(93.3%)
Prune2Drive	31.89	**72.76**	40.20	27.84	**43.17(97.6%)**
Ours	**39.33**	57.75	43.49	**31.46**	43.00(97.3%)

Chatgpt, Rouge, CIDEr and Match under the 75% pruning setting. Compared to Prune2Drive, our method yields consistent improvements of 0.8 and 0.4.

MAPLM. Results in Tab. 3 show that under the more challenging multi-sensor input setting, our method exhibits a substantially smaller accuracy drop than SOTA methods. With 90% pruning, FastV, SparseVLM, and Prune2Drive degrade by 17.63, 15.84, and 7.43, respectively, whereas our method drops by only 3.94, remaining close to the original model. This indicates that our method can effectively identify critical information across diverse inputs and exhibits stability and strong generalization capability across tasks.

STSnu. On the STSnu benchmark, we apply our MVPruner and other methods independently to each frame of the six-view video sequences. The results are presented in Tab. 4. Under the 75% pruning setting, our method even surpasses the original model by 0.04, outperforming SOTA methods and demonstrating its effectiveness in removing spatiotemporal redundancy.

Table 5: Efficiency comparison on DriveLMM-o1, DriveLM, and MAPLM benchmarks under the 90% pruning setting.

	Methods	Speedup (ms) ↓ (Prefilling)	(Decoding)	FLOPs ↓	GPU Peak ↓ Memory (GB)	Throughput ↓ (samples/s)	Performance ↑
DriveLMM-o1	DriveLMM-o1	591	51.9	100%	17.31	11.84	74.37
	+ FastV	128.5(↑4.60×)	45.9(↑1.13×)	20.3%	16.61	10.57(↑1.12×)	69.56
	+ SparseVLM	164.6(↑3.59×)	47.6(↑1.09×)	20.7%	16.65	9.55(↑1.24×)	69.40
	+ Prune2Drive	**127.1(↑4.65×)**	47.2(↑1.10×)	20.3%	16.58	**8.58(↑1.38×)**	70.31
	+ **Ours**	155.5(↑3.80×)	**42.2(↑1.23×)**	**20.0%**	**16.08**	8.60(↑**1.38**×)	**72.91**
DriveLM	DriveMM	1968	65.3	100%	16.47	8.12	59.1
	+ FastV	340.5(↑5.78×)	63.4(↑1.03×)	13.3%	15.99	5.56(↑1.46×)	55.4
	+ SparseVLM	484.7(↑4.06×)	64.0(↑1.02×)	14.4%	16.00	5.88(↑1.38×)	55.9
	+ Prune2Drive	**307.5(↑6.40×)**	59.9(↑1.09×)	13.4%	16.00	**5.49(↑1.48×)**	57.4
	+ **Ours**	395.6(↑4.97×)	**51.8(↑1.26×)**	**12.7%**	**15.93**	5.58(↑1.45×)	**58.2**
MAPLM	DriveMM	1182	61.7	100%	16.24	1.72	72.24
	+ FastV	286.9(↑4.12×)	58.2(↑1.06×)	15.6%	15.91	0.82(↑2.10×)	54.61
	+ SparseVLM	353.9(↑3.34×)	59.9(↑1.03×)	15.8%	15.93	1.11(↑1.55×)	56.40
	+ Prune2Drive	**274.9(↑4.30×)**	55.6(↑1.11×)	15.6%	15.91	0.69(↑2.49×)	64.81
	+ **Ours**	299.1(↑3.95×)	**51.6(↑1.20×)**	**15.2%**	**15.88**	**0.64(↑2.69×)**	**68.30**

Table 6: Latency analysis of each pruning module on DriveLM benchmark under the 90% pruning setting.

DRA	CCTS (/view)	IRA	ITS (/view)	Total
3.55ms	26.48ms	1.11ms	0.06ms	163.9ms

By preserving information aligned with the dynamic requirements of the model's reasoning process, our method achieves consistent performance gains across diverse benchmarks and exhibits strong generalization.

Inference Efficiency. We analyze the efficiency of our method using speedup in the prefilling and decoding stages, FLOPs, GPU peak memory usage and throughput, under the 90% pruning setting. The results are shown on Tab. 5, our method achieves the fastest decoding time, the lowest FLOPs, and GPU peak memory usage. Despite achieving efficiency comparable to Prune2Drive, our method preserves performance more effectively and can be directly applied to diverse models and scenarios without additional adjustments or costly hyper-parameter tuning, offering a more practical and scalable solution.

We further profile each component using DriveMM on DriveLM with six views and 729 tokens/view under the 90% pruning setting. As shown in Tab. 6, CCTS dominates the overhead due to pairwise similarity computation, taking 26.48 ms/view. The total pruning overhead is 163.9 ms, only 2.9% of the full model's end-to-end latency (5580 ms). The detailed computing budget of our method is detailed in the supplementary materials.

4.3 Ablation study

In this section, we conduct ablation studies on DriveLM and MAPLM under the 90% pruning setting to evaluate the key design choices of our method.

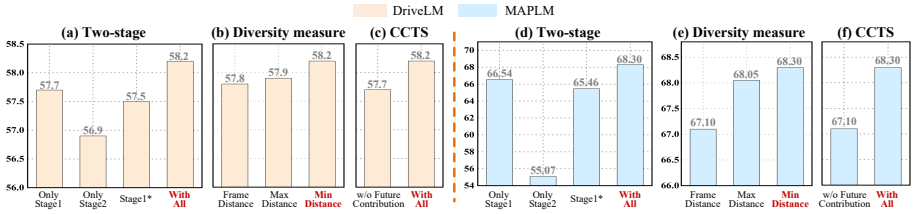

Fig. 4: Ablation studies on DriveLM and MAPLM benchmarks using DriveMM model.

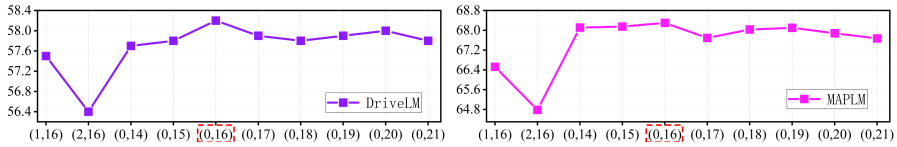

Fig. 5: Performance of different pruning layer selection strategies on DriveLM and MAPLM benchmarks using DriveMM model.

Two-stage Strategy. We validate the proposed two-stage pruning strategy. Specifically, each stage is independently applied to the shallow layers to meet the target pruning ratio. Additionally, a variant in which stage 2 adopts the same strategy as stage 1 is evaluated, denoted as Stage1*, the results are shown in Figs. 4(a, d). Applying stage 1 alone effectively removes redundant information while maximizing the semantic coverage of the retained token subset, satisfying the diversity-driven information requirements of shallow layers and outperforming SOTA methods even when used independently. In contrast, applying stage 2 independently yields inferior results, as early layers attend broadly to visual content and have not yet focused on task-relevant regions. Stage1* exhibits accuracy decline because the textual and visual modalities have already been aligned in deeper layers, and the model needs task-relevant information. These results confirm the necessity of the two-stage design, which aligns with the dynamic information requirements of the model during the reasoning process.

Diversity Measure in DRA. We compare different diversity metrics for visual tokens in DRA, including distance to the global frame feature of the view, maximum pairwise distance, and minimum pairwise distance which we adopted. As shown in Figs. 4(b, e), the choice of diversity metrics has only a minor impact on overall performance. However, the minimum pairwise distance consistently provides the most effective measure of token uniqueness.

Metrics in CCTS. We evaluate the impact of the future contribution metric on the CCTS. As shown in Figs. 4(c, f), removing the future contribution metric causes the premature loss of potentially task-related information and degrades accuracy. By assessing each token's contribution throughout the entire reasoning process, our method maximizes the representational capacity of the retained tokens in the shallow layer.

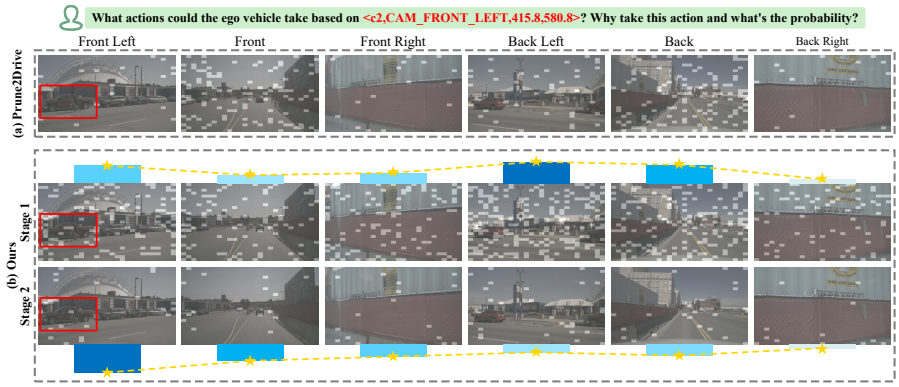

Fig. 6: Quantitative comparison of kept tokens. Taller and darker bars indicate view importance, where MVPruner allocates more tokens to information-rich views in stage 1 and to task-relevant views in stage 2.

Pruned Layers Selection. We provide a detailed analysis of the pruning layer selection strategies, the results are shown in Fig. 5. The placement of stage 1 has an impact on the model, exhibiting a trend of performance degradation as the layer depth increases. One possible explanation is that, as the layer deepens, visual features progressively align with the text modality, resulting in the loss of fine-grained visual information and consequently reducing the reliability of the diversity metric. In contrast, the placement of stage 2 has only a minor effect on accuracy, indicating that the model in deeper layers can recognize task-related regions. Notably, the layer at which stage 2 achieves peak accuracy aligns with the layer exhibiting the highest task-related view recognition accuracy in Sec. 3.2, indicating that our prior analysis accurately captures the model's ability to identify task-relevant information.

4.4 Qualitative Analysis

The qualitative comparison results between our method and Prune2Drive are shown in Fig. 6. Since most samples in the calibration set focus on front and back views, Prune2Drive allocates fixed, larger token budgets to them. Consequently, it fails to preserve relevant information when the model must attend to objects in the front-left view. In contrast, our method dynamically allocates the token budget across different reasoning stages according to the scene and task instructions. It maximizes the semantic representational capacity of retained tokens in shallow layers to provide comprehensive contextual information for subsequent layers, and precisely preserves task-relevant tokens in deeper layers.

Fig. 7 shows samples with a fixed scene and varying instructions. Our method captures the resulting shifts in view importance and adaptively reallocates budgets, demonstrating strong generalization capability. More visualizations are in the supplementary materials.

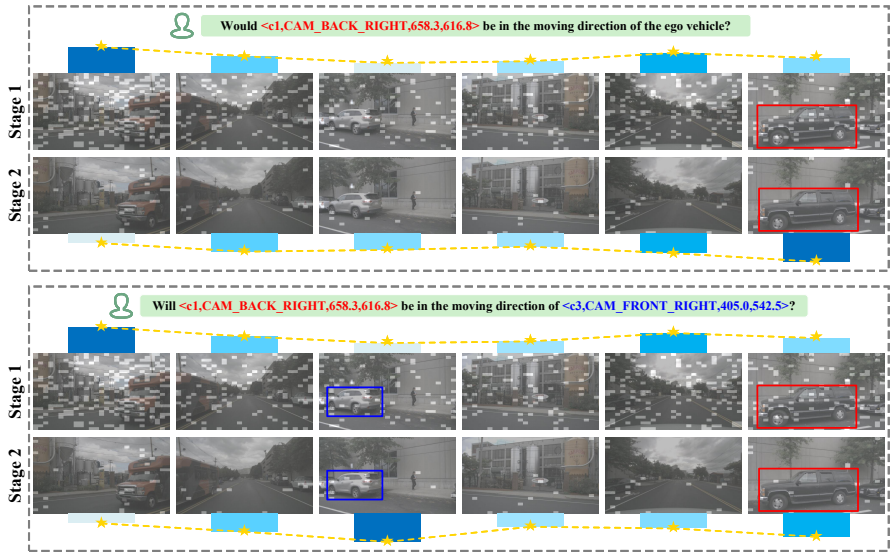

Fig. 7: Visualization of kept tokens. Taller and darker bars indicate view importance. Our method adaptively adjusts pruning budget allocation in response to variations in the instruction.

5 Conclusion

In this paper, we analyze the ability of multi-view VLMs to identify important views and the underlying mechanisms. Results reveal that attention scores can effectively indicate task-related views in deeper layers and the reasoning process of VLMs exhibits dynamic information requirements. Guided by these findings, we propose MVPruner, which dynamically adjusts pruning ratios across views and retains the most critical tokens at each stage, aligning with the model's dynamic information requirements. Extensive experiments demonstrate that our method achieves a superior efficiency–performance trade-off across diverse benchmarks and exhibits strong generalization.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China under Grant U24B20127, in part by the National Key Research and Development Program of China under Grant 2024YFB4303102, and in part by the Chang'an University Graduate Student Research Innovation and Practice Program under Grant 300103267062.

References

1. Alvar, S.R., Singh, G., Akbari, M., Zhang, Y.: Divprune: Diversity-based visual token pruning for large multimodal models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 9392–9401 (2025) 4, 10
2. Bolya, D., Fu, C.Y., Dai, X., Zhang, P., Feichtenhofer, C., Hoffman, J.: Token merging: Your vit but faster. arXiv preprint arXiv:2210.09461 (2022) 2, 4
3. Bordes, F., Pang, R.Y., Ajay, A., Li, A.C., Bardes, A., Petryk, S., Mañas, O., Lin, Z., Mahmoud, A., Jayaraman, B., et al.: An introduction to vision-language modeling. arXiv preprint arXiv:2405.17247 (2024) 1
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020) 1, 9
5. Cao, X., Zhou, T., Ma, Y., Ye, W., Cui, C., Tang, K., Cao, Z., Liang, K., Wang, Z., Rehg, J.M., et al.: Maplm: A real-world large-scale vision-language benchmark for map and traffic scene understanding. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 21819–21830 (2024) 3, 9
6. Chen, L., Wu, P., Chitta, K., Jaeger, B., Geiger, A., Li, H.: End-to-end autonomous driving: Challenges and frontiers. IEEE Transactions on Pattern Analysis and Machine Intelligence (2024) 4
7. Chen, L., Zhao, H., Liu, T., Bai, S., Lin, J., Zhou, C., Chang, B.: An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In: European Conference on Computer Vision. pp. 19–35. Springer (2024) 1, 2, 4, 10
8. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Zhong, M., Zhang, Q., Zhu, X., Lu, L., 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 (2024) 8
9. Dhouib, M., Buscaldi, D., Vanier, S., Shabou, A.: Pact: Pruning and clustering-based token reduction for faster visual language models. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 14582–14592 (2025) 10
10. Fruhwirth-Reisinger, C., Malić, D., Lin, W., Schinagl, D., Schulter, S., Possegger, H.: Stsbench: A spatio-temporal scenario benchmark for multi-modal large language models in autonomous driving. arXiv preprint arXiv:2506.06218 (2025) 3, 9
11. Huang, Z., Feng, C., Yan, F., Xiao, B., Jie, Z., Zhong, Y., Liang, X., Ma, L.: Drivemm: All-in-one large multimodal model for autonomous driving. arXiv preprint arXiv:2412.07689 (2024) 4, 8
12. Huang, Z., Sheng, Z., Qu, Y., You, J., Chen, S.: Vlm-rl: A unified vision language models and reinforcement learning framework for safe autonomous driving. Transportation Research Part C: Emerging Technologies **180**, 105321 (2025) 1
13. Ishaq, A., Lahoud, J., More, K., Thawakar, O., Thawkar, R., Dissanayake, D., Ahsan, N., Li, Y., Khan, F.S., Cholakkal, H., et al.: Drivelmm-o1: A step-by-step reasoning dataset and large multimodal model for driving scenario understanding. arXiv preprint arXiv:2503.10621 (2025) 1, 3, 4, 8, 9
14. Jiang, B., Chen, S., Liao, B., Zhang, X., Yin, W., Zhang, Q., Huang, C., Liu, W., Wang, X.: Senna: Bridging large vision-language models and end-to-end autonomous driving. arXiv preprint arXiv:2410.22313 (2024) 4

15. Jiang, S., Huang, Z., Qian, K., Luo, Z., Zhu, T., Zhong, Y., Tang, Y., Kong, M., Wang, Y., Jiao, S., et al.: A survey on vision-language-action models for autonomous driving. arXiv preprint arXiv:2506.24044 (2025) 1
16. Kim, J., Rohrbach, A., Darrell, T., Canny, J., Akata, Z.: Textual explanations for self-driving vehicles. In: Proceedings of the European conference on computer vision (ECCV). pp. 563–578 (2018) 4
17. Kim, M.J., Pertsch, K., Karamcheti, S., Xiao, T., Balakrishna, A., Nair, S., Rafailov, R., Foster, E., Lam, G., Sanketi, P., et al.: Openvla: An open-source vision-language-action model. arXiv preprint arXiv:2406.09246 (2024) 4
18. Li, B., Zhang, Y., Guo, D., Zhang, R., Li, F., Zhang, H., Zhang, K., Zhang, P., Li, Y., Liu, Z., et al.: Llava-onevision: Easy visual task transfer. arXiv preprint arXiv:2408.03326 (2024) 1, 8
19. Liang, Y., Ge, C., Tong, Z., Song, Y., Wang, J., Xie, P.: Not all patches are what you need: Expediting vision transformers via token reorganizations. arXiv preprint arXiv:2202.07800 (2022) 4
20. Liu, H., Li, C., Wu, Q., Lee, Y.J.: Visual instruction tuning. Advances in neural information processing systems **36**, 34892–34916 (2023) 1
21. Liu, Z., Cheng, J., Fan, J., Lin, S., Wang, Y., Zhao, X.: Multi-modal fusion based on depth adaptive mechanism for 3d object detection. IEEE Transactions on Multimedia **27**, 707–717 (2023) 1
22. Mao, J., Qian, Y., Ye, J., Zhao, H., Wang, Y.: Gpt-driver: Learning to drive with gpt. arXiv preprint arXiv:2310.01415 (2023) 1, 4
23. Rao, Y., Zhao, W., Liu, B., Lu, J., Zhou, J., Hsieh, C.J.: Dynamicvit: Efficient vision transformers with dynamic token sparsification. Advances in neural information processing systems **34**, 13937–13949 (2021) 4
24. Shao, H., Hu, Y., Wang, L., Song, G., Waslander, S.L., Liu, Y., Li, H.: Lmdrive: Closed-loop end-to-end driving with large language models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15120–15130 (2024) 4
25. Shao, K., Tao, K., Zhang, K., Feng, S., Cai, M., Shang, Y., You, H., Qin, C., Sui, Y., Wang, H.: When tokens talk too much: A survey of multimodal long-context token compression across images, videos, and audios. arXiv preprint arXiv:2507.20198 (2025) 2, 4
26. Sima, C., Renz, K., Chitta, K., Chen, L., Zhang, H., Xie, C., Beißwenger, J., Luo, P., Geiger, A., Li, H.: Drivelm: Driving with graph visual question answering. In: European conference on computer vision. pp. 256–274. Springer (2024) 1, 3, 4, 5, 9
27. Sun, P., Kretzschmar, H., Dotiwalla, X., Chouard, A., Patnaik, V., Tsui, P., Guo, J., Zhou, Y., Chai, Y., Caine, B., et al.: Scalability in perception for autonomous driving: Waymo open dataset. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2446–2454 (2020) 1
28. Wang, S., Yu, Z., Jiang, X., Lan, S., Shi, M., Chang, N., Kautz, J., Li, Y., Alvarez, J.M.: Omnidrive: A holistic vision-language dataset for autonomous driving with counterfactual reasoning. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 22442–22452 (2025) 1
29. Wei, H., Wang, R., Hu, H., Sun, S., Song, X., Feng, M., Guo, K., Huang, Y., Cui, H., Akhtar, N.: Monocular multi-object 3d visual language tracking. IEEE Transactions on Image Processing **35**, 2050 – 2065 (2026) 1
30. Wen, Z., Gao, Y., Li, W., He, C., Zhang, L.: Token pruning in multimodal large language models: Are we solving the right problem? arXiv preprint arXiv:2502.11501 (2025) 2, 4

31. Wen, Z., Gao, Y., Wang, S., Zhang, J., Zhang, Q., Li, W., He, C., Zhang, L.: Stop looking for important tokens in multimodal language models: Duplication matters more. arXiv preprint arXiv:2502.11494 (2025) 4, 10
32. Xing, L., Huang, Q., Dong, X., Lu, J., Zhang, P., Zang, Y., Cao, Y., He, C., Wang, J., Wu, F., et al.: Pyramiddrop: Accelerating your large vision-language models via pyramid visual redundancy reduction. arXiv preprint arXiv:2410.17247 (2024) 4
33. Xiong, M., Wen, Z., Gu, Z., Liu, X., Zhang, R., Kang, H., Yang, J., Zhang, J., Li, W., He, C., et al.: Prune2drive: A plug-and-play framework for accelerating vision-language models in autonomous driving. arXiv preprint arXiv:2508.13305 (2025) 1, 2, 4, 10
34. Yang, N., Wang, Y., Liu, Z., Li, M., An, Y., Zhao, X.: Smamba: Sparse mamba for event-based object detection. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 9229–9237 (2025) 2
35. Zhang, Y., Fan, C.K., Ma, J., Zheng, W., Huang, T., Cheng, K., Gudovskiy, D., Okuno, T., Nakata, Y., Keutzer, K., et al.: Sparsevlm: Visual token sparsification for efficient vision-language model inference. arXiv preprint arXiv:2410.04417 (2024) 1, 2, 4, 10
36. Zhou, X., Liu, M., Yurtsever, E., Zagar, B.L., Zimmer, W., Cao, H., Knoll, A.C.: Vision language models in autonomous driving: A survey and outlook. IEEE Transactions on Intelligent Vehicles (2024) 1, 4