Think Once, Reuse Smartly: Bio-Inspired Memory for Efficient Vision-Language Reasoning in Autonomous Driving

Anonymous ACL submission

Abstract

Vision-Language Models (VLMs) are increasingly vital for robust decision-making in autonomous driving, yet their deep reasoning creates a critical latency bottleneck, making them impractical for real-world deployment. Current approaches accelerate inference by pruning input data, but they overlook the primary source of inefficiency: the constant, wasteful re-computation of reasoning that remains valid across consecutive frames. We introduce MEMO-VLM, a memory-driven framework inspired by human cognition that eliminates this redundant reasoning. Instead of regenerating its entire reasoning, MEMO-VLM treats previous conclusions as a hypothesis to be validated against new visual evidence, intelligently reusing what remains true and surgically updating only what has changed. This is achieved with a plug-and-play, two-stage approach that requires no VLM retraining, making it a broadly applicable solution. Experiments demonstrate that MEMO-VLM accelerates inference by up to 4.3×. By bridging bio-inspired memory with computational efficiency, our work offers a practical path to deploying the advanced reasoning of VLMs in safety-critical autonomous systems.

1 Introduction

Autonomous driving has emerged as a transformative technology, promising safer and more efficient transportation(Hu et al., 2023; Ye et al., 2022). Modern approaches often adopt an end-to-end (E2E) solution that directly maps raw sensor inputs to driving actions, integrating perception, decision-making, and control into a unified model(Chen et al., 2024c; Hu et al., 2023; Jiang et al., 2023; Sun et al., 2024). Vision-Language Models (VLMs) excel at understanding complex or uncommon driving scenarios through reasoning by combining visual perception with contextual knowledge(Chen et al., 2024b; Sima et al., 2024; Xu et al., 2024; Wang et al., 2024b).

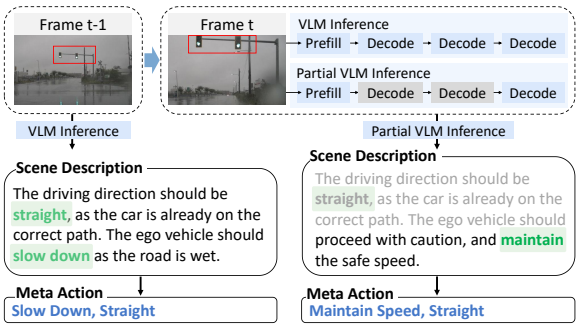


Figure 1: Partial inference accelerates temporal reasoning by reusing prior reasoning outputs (gray) and skipping redundant decodes, avoiding the high cost of standard inference.

Yet, their model inference time, often exceeding seconds per frame, makes them impractical for dynamic environments. Current acceleration methods mainly focus on spatial(Zhang et al., 2024b; Rao et al., 2021; Chen et al., 2023; Yang et al., 2024) or temporal(Vasu et al., 2024; Xu et al., 2025; Dutson et al., 2023; Chen et al., 2024e) redundancies in the input (e.g., pruning image tokens or skipping frames based on pixel similarity). While useful, these approaches overlook a more critical bottleneck: the reasoning processes itself. Recomputing reasoning outputs for near-identical consecutive frames wastes resources, a process akin to re-reading an unchanging paragraph. This is not how humans operate. We instinctively retain a stable context of a scene and update only what changes. Inspired by this cognitive efficiency, we propose MEMO-VLM, a framework that accelerates VLM inference by reusing valid reasoning across frames. Our key insight is that the most significant temporal redundancy lies not in input pixels, but in the semantic reasoning output.

MEMO-VLM is built on a simple yet powerful approach: it treats a model's previous reasoning output not as a one-time calculation, but as a hypothesis to be challenged by new visual evi-

dence. At the heart of our work is the insight that a VLM's innate ability to predict next tokens can be repurposed to evaluate the consistency between a new visual frame and a cached textual description. By looking into the model's confidence in its own prior conclusions, MEMO-VLM dynamically determines which parts of its reasoning remain valid and which require re-evaluation. This strategy allows the system to maintain a stable understanding of scenes and surgically update only the portions that have changed, effectively bypassing redundant computational steps.

We validated this approach through extensive experiments on two autonomous driving benchmarks, across two hardware platforms, and with six state-of-the-art VLMs. To prove its readiness for real road conditions, we deployed our system in a Baidu Apollo 2.0 autonomous vehicle for two weeks of continuous operation in live urban traffic. Our method running in a shadow mode demonstrated it could make real-time, human-comparable driving decisions in complex scenarios, from sudden pedestrian appearances to interactions with oncoming traffic. Our framework achieves a substantial inference speed up of up to 4.3× with near-zero accuracy loss of critical meta-actions. Moreover, our training-free solution bridges the critical gap between bio-inspired memory concepts and computational efficiency, offering a practical path toward deploying more human-like reasoning in real-time autonomous systems.

2 Related Work

2.1 VLMs for Autonomous Driving

The integration of Vision-Language Models (VLMs) is rapidly becoming a cornerstone of modern autonomous driving systems, enhancing their ability to reason about complex and rare scenarios(Xu et al., 2024). A significant body of research(Wen et al., 2023; Zheng et al., 2024; Wang et al., 2023, 2024b; Jiang et al., 2024; Feng et al., 2025) leverages VLMs for high-level decision-making, where the model interprets visual data and generates an interpretable plan or meta-action. Other works have explored hybrid frameworks that combine the semantic reasoning of VLMs with the precision of traditional controllers(Long et al., 2024; Chen et al., 2024d; Tian et al., 2024; Zhang et al., 2025) or distill VLM knowledge into more compact planning modules(Zhou et al., 2025b). While these approaches have pushed the boundaries of scene understanding and interpretability, the substantial computational cost and high latency of their reasoning processes remain a critical barrier to real-world deployment.

2.2 Efficiency in Vision-Language Models

Efforts to accelerate VLM inference have predominantly focused on exploiting redundancy in the input data, which can be broadly categorized into spatial and temporal approaches. One line of work targets spatial redundancy within a single image frame. These methods typically prune(Rao et al., 2021; Chen et al., 2023; Zhang et al., 2024a; Chen et al., 2024a; Guo et al., 2025) or merge(Bolya et al., 2022; Huang et al., 2025; Zhang et al., 2024b; Zhou et al., 2025a) visually repetitive or task-irrelevant tokens to reduce the computational load on the transformer backbone. Techniques range from using lightweight predictors to hierarchically prune tokens , leveraging cross-attention to identify and discard redundant information , to merging tokens based on feature or spatial similarity. While effective at compressing the visual input, these methods do not address the computational bottleneck of generating the textual reasoning output itself. Another line of work addresses temporal redundancy in sequential data like video(Li et al., 2021). These approaches typically reuse computations for static background regions(Dutson et al., 2023; Vasu et al., 2024) or cache visual tokens from unchanged areas of the scene across consecutive frames(Sun et al., 2025). Some methods extend this to the action space in robotics, reusing token-aware actions when visual input is stable(Xu et al., 2025; Tan et al., 2025; Qian et al., 2025).

3 Background and Motivation

While VLMs offer sophisticated reasoning for autonomous driving, their autoregressive nature creates a crippling latency bottleneck. On a typical in-vehicle platform like the NVIDIA AGX Orin, processing a single frame can take over 5.5 seconds, shown in Fig. 2, rendering them unsafe for real-time deployment. Our work is motivated by the source of this inefficiency: massive redundancy in the reasoning process itself. Our analysis on the NuScenes dataset reveals that up to 78.6% of reasoning tokens, shown in Fig. 3, are identical across consecutive frames. This waste is not limited to general-purpose models. Even a state-of-the-art VLM like Senna(Jiang et al., 2024), fine-tuned

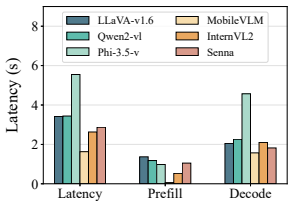


Figure 2: Inference latency of end-to-end planning with reasoning of VLMs on NVIDIA Orin.

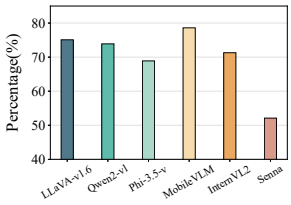


Figure 3: Redundancy ratios (%) of VLMs on NuScenes dataset.

specifically to excel at autonomous driving, still regenerates 52.1% of its reasoning unnecessarily. Current systems waste the majority of their computation regenerating known information. Inspired by cognition efficiency, we aim to replace stateless regeneration with a stateful update model, directly exploiting this semantic redundancy to achieve real-time performance.

4 Method

Our framework introduces a training-free, output-driven reuse strategy that exploits semantic redundancy in the model's own reasoning. Instead of pruning input data or shallow features, we treat model's previous chain-of-thought as a hypothesis to be validated against new visual evidence. Our method dynamically preserves stable context and updates only parts of reasoning that have changed, directly bypassing redundant computation.

4.1 Temporal Reasoning Formulation

At each time step t, the system observes a visual input $I_t \in \mathcal{I}$ and must produce a meta-action $A_t \in \mathcal{A}$, high-level driving decisions such as turning or slowing down. VLMs achieve this by first generating an internal **belief state** (including traffic conditions, object behaviors, etc.) $\xi_t \in \Xi$, which serves as an explicit reasoning for the subsequent action.

The core of our method is to exploit temporal redundancy by generating ξ_t from a **dynamic memory**, $\mathcal{M}_t$, which preserves the stable portion of the prior belief state ξ_{t-1}. This is formalized as a three-stage generative process governed by a policy π_θ:

$$\begin{aligned} \mathcal{M}_t = \mathcal{U}(I_t, \xi_{t-1}), \xi_t \sim \pi_\theta(\cdot|I_t, \mathcal{M}_t), \\ A_t \sim \pi_\theta(\cdot|I_t, \xi_t) \end{aligned} \quad (1)$$

The memory update operator, $\mathcal{U}$, identifies the maximal valid prefix of ξ_{t-1} given the new observation I_t. It computes a "divergence point" k—the first token index at which the cached reasoning no longer aligns with the current reality—and defines the memory as the prefix $\xi_{t-1}^{<k}$.

$$k := \min\{j \in [1, |\xi_{t-1}|] \mid \Phi(\xi_{t-1}^{j} | I_t, \xi_{t-1}^{<j}) < \tau_j\} \quad (2)$$

Here, Φ is a semantic consistency scoring function, and τ_j is a validity threshold. This approach is motivated by the **temporal sparsity hypothesis**: in continuous driving scenarios, the belief state ξ_t evolves sparsely. Our framework leverages this by reusing the computationally expensive, stable prefix $\mathcal{M}_t$ and regenerating only the divergent suffix, thus minimizing redundant computation.

4.2 Coarse-to-Fine Memory Reuse

Our framework employs a two-stage validation process to efficiently determine the reusable portion of prior belief state, ξ_{t-1}. This cascade is designed to first perform a fast, holistic check and only proceed to a more granular analysis if necessary.

Global Semantic Validation. The initial stage is a global test for semantic consistency. We evaluate whether the entire prior belief state ξ_{t-1} remains valid under the new visual observation I_t. This is framed as a hypothesis test where the null hypothesis is that the semantic content of the scene has not diverged significantly. To perform this test, we compute the probability of every token of the cached belief state $\xi_{t-1} = (\omega_1, \ldots, \omega_L)$ when conditioned on the new image I_t. This is calculated during a single forward pass (i.e., the prefill stage) without any token generation: $P(\omega_j|I_t) = P(\omega_j|I_t, \omega_{<j})$.

The core principle is that the model's own predictive distribution, $P(\cdot|I_t, \cdot)$, acts as an intrinsic semantic consistency function. A high probability indicates that ξ_{t-1} is a plausible description of the scene depicted by I_t. We accept ξ_{t-1} in its entirety if the probability of every token ω_i in ξ_{t-1} exceeds a calibrated per token threshold $\tau'_{\omega_i,t}$:

$$\text{Accept if } \bigwedge_{i=1}^{L} P(\omega_i|I_t, \omega_{<i}) > \tau'_{\omega_i,t} \quad (3)$$

If this condition holds, we bypass the fine-grained check and the generation process entirely, achieving maximal computational savings. This approach elegantly repurposes the model's internal representations for self-validation without any external or auxiliary validation networks.

Hierarchical and Adaptive Threshold Calibration. To be robust, a validity threshold must account for both global domain shifts and local token

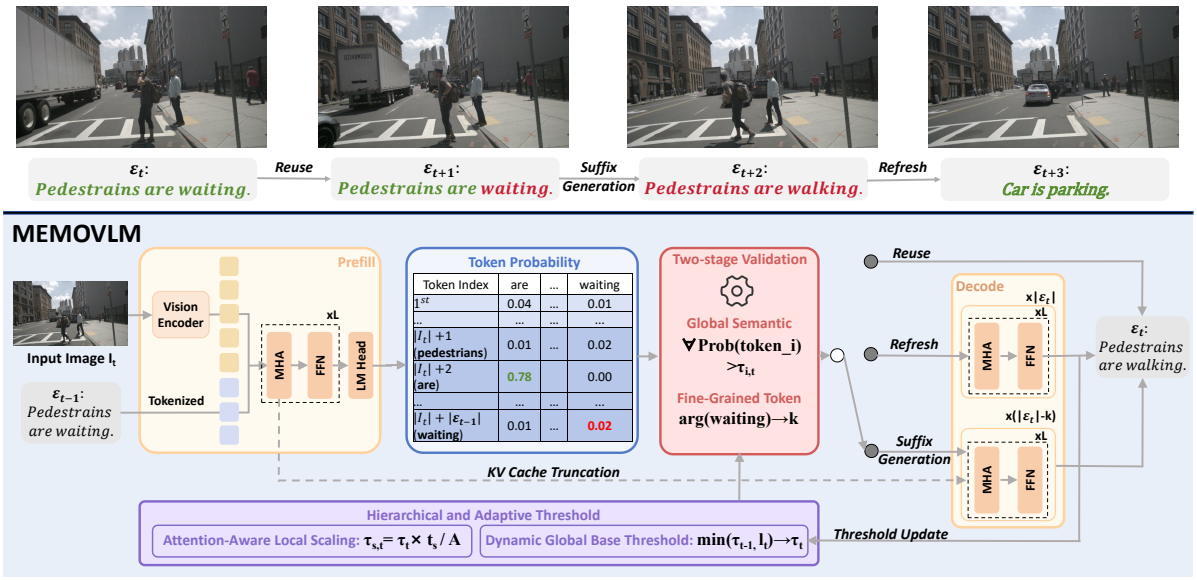


Figure 4: Overview of the proposed memory-driven framework for accelerating VLM inference in autonomous driving: Instead of regenerating all the states at each time step, the framework retains and reuses past reasoning outputs based on semantic consistency across consecutive frames, reducing unnecessary computation.

importance. We achieve this with a hierarchical calibration strategy that defines a dynamic, per-token threshold $\tau'_{s,t}$.

1. Dynamic Global Base Threshold (τ_t). It provides initial stability and adapts to long-term environmental changes.

- **Offline Initialization:** The initial threshold τ_0 is set to the minimum token probability for all valid reasoning outputs from the entire NuScenes dataset. For each token s_i in the output R_t, we record its probability $P(s_i|I_t, R_{<i})$. The threshold τ_0 is defined as:

$$\tau_0 = \min_{t \in \mathcal{F}} \min_i P(s_i|I_t, R_t^{<i}) \tag{4}$$

  where $\mathcal{F}$ is the set of frames.

- **Online Adaptation:** The threshold adapts conservatively over time. This update is triggered only during a memory refresh event which will be discussed in Section 4.4, using the lowest probability $l_{\min,t}$ from the newly regenerated sequence: $\tau_{t+1} \leftarrow \min(\tau_t, l_{\min,t})$.

2. Attention-Aware Local Scaling. This static component adjusts the base threshold based on a token's semantic role. Our analysis of the NuScenes dataset reveals that tokens corresponding to critical entities (e.g., "car", "pedestrian") receive 3–5× higher attention than less informative tokens. To compute this, we first consider the standard scaled dot-product attention weights. For each layer ℓ and head h, the weight from a language token s to an image token j is:

$$w_{s\to j}^{\ell,h} = \text{softmax}\Big(\frac{Q_s^{(\ell,h)} K_j^{(\ell,h)\mathsf{T}}}{\sqrt{d_k}}\Big) \tag{5}$$

The image-focus score $\bar{t}_s$ is the average of these weights over all L layers, H heads, and all image tokens:

$$\bar{t}_s = \frac{1}{LH} \sum_{\ell=1}^{L} \sum_{h=1}^{H} \sum_{j \in \mathcal{I}} w_{s\to j}^{\ell,h} \tag{6}$$

where $\mathcal{I}$ indexes the image tokens. This ratio $\bar{t}_s$ therefore directly measures how much attention token s places on image tokens. Over an offline "representative scene" token set $\mathcal{S}$, we compute the global calibration constant:

$$\bar{A} = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \bar{t}_s \tag{7}$$

The final, operational threshold for a token s at time t combines the dynamic global base with the static, attention-based scaling factor. This ensures that semantically critical tokens, which have a higher image-focus, face a proportionally stricter validity check: $\tau'_{s,t} = \tau_t \cdot \frac{\bar{t}_s}{\bar{A}}$.

Fine-Grained Token Validation. If the global check fails, we examine the memory at a finer granularity. Rather than discarding the entire memory, we prune only parts that no longer fit the current scene. We find the first token whose predicted probability falls below τ_t and truncate at that point, removing that token and everything after it in that segment. This "divergence point" t_w marks where the cached reasoning becomes outdated:

$$k = \arg \min_{i \in \{1,\dots,L\}} \left\{ i \mid P(\omega_i | I_t, \xi_{t-1}^{<i}) \leq \tau'_{\omega_i,t} \right\} \tag{8}$$

For example, if R_{t-1} contained *"A pedestrian is waiting at the crosswalk,"* and *"waiting"* now has low probability (perhaps the pedestrian started moving), we truncate to *"A pedestrian is..."*. This preserves what's still valid while removing outdated information. By avoiding recomputation of unchanged descriptions, we reduce the transformer's decoding computation from $O(T^2)$ to roughly $O((1-\alpha)T^2)$, where α is the fraction of tokens pruned.

4.3 Memory-Driven Autoregressive Generation

Once the validation process identifies the divergence point k, and thus the valid memory prefix $\mathcal{M}_t = \xi_{t-1}^{<k}$, the system begins an efficient, memory-driven generation process to produce the new belief state ξ_t. It leverages the KV cache mechanism intrinsic to the Transformer architecture.

Prefill with Memory Context. Instead of starting from an empty context, the inference process is initialized with a prefill step that computes the Key-Value (KV) cache for the combined input of the current image I_t and the valid memory prefix $\mathcal{M}_t$. This single, parallel forward pass produces the prefix KV cache:

$$(K_{\text{prefix}}, V_{\text{prefix}}) = \text{ComputeKV}(I_t, \mathcal{M}_t) \tag{9}$$

where $K_{\text{prefix}}, V_{\text{prefix}} \in \mathbb{R}^{L \times H \times (|\mathcal{I}_t| + k) \times d_k}$, with L, H, d_k being the number of layers, heads, and key dimension, respectively. This step encodes the entire stable context.

Suffix Generation via Cross-Attention. The model generates the remainder of the belief state, the suffix $\xi_t^{\geq k}$, autoregressively. At each decoding step $j > k$, the model computes the attention context vector c_j for each head by attending to the concatenation of stable prefix cache and the KV entries from the newly generated suffix tokens ($\xi_t^{<j}$):

$$c_j = \text{Attention}(Q_j, [K_{\text{prefix}}; K_{\text{suffix},<j}], [V_{\text{prefix}}; V_{\text{suffix},<j}]) \tag{10}$$

Here, $[\cdot;\cdot]$ denotes concatenation along the sequence dimension. This mechanism ensures that the generation of new tokens is fully conditioned on both the fresh visual information and the preserved reasoning from the past, maintaining spatiotemporal continuity. The next token ω_j is then sampled from the distribution produced by the model's output layer: $\omega_j \sim \pi_\theta(\cdot | c_j)$. This process continues until an end-of-sequence token is generated. This approach significantly reduces decoding latency, as the number of sequential generation steps is reduced from $|\xi_t|$ to $|\xi_t| - k$.

Final Belief State Update. Once the suffix generation is complete, the final belief state ξ_t is formed by concatenating the reused memory prefix and the newly generated suffix. This complete state ξ_t becomes the input for the action prediction A_t and serves as the candidate belief state, ξ_t, for the next time step, thus completing the inference cycle.

4.4 Robustness via Bounded Error Propagation

While a single probability drop leads to memory truncation, such an event might be an isolated anomaly rather than a systemic failure. To distinguish between transient noise and actual semantic divergence, we introduce a parameter-free refresh strategy based on token correlation.

The theoretical basis of this strategy is that in an VLM, the probability of a token is strongly dependent on its correlation with the immediate predecessor. Therefore, if an error occurs in the memory at a specific token, but the subsequent token returns to a high probability, it implies the error did not affect the semantic integrity of the sequence. However, if there are consecutive tokens with probabilities below their calibration threshold, it indicates that the error has accumulated and propagated to the next step. Thus, a full memory refresh is triggered by the event $\mathcal{R}_t$, defined as the occurrence of any two consecutive validation failures:

$$\mathcal{R}_t \iff \exists j \text{ s.t. } D_j \wedge D_{j+1} \tag{11}$$

where D_j is the validation failure event for token ω_j (i.e., $P(\omega_j | I_t, \omega_{<j}) < \tau'_{\omega_j,t}$). Upon this trigger, the entire memory $\mathcal{M}_t$ is discarded ($\mathcal{M}_t \leftarrow \emptyset$), forcing a complete regeneration of the belief state

Model	Method	Latency (s) ↓		Accuracy (F1) ↑			BLEU-4 ↑	CIDEr↑	Challenge Set Succ.(%)↑
		Orin	3090	All	Speed	Direction			
llava-v1.6-vicuna-7b	Vanilla	3.43	2.68	57.75	71.20	79.73	51.13	2.38	51.48
	SparseVLM	2.65	1.80	54.83	67.64	75.74	48.57	2.26	48.31
	VLA-Cache	2.52	1.73	57.68	71.13	79.65	51.05	2.35	51.42
	FlashVLA	2.07	1.35	57.70	71.15	79.68	51.08	2.37	51.45
	Ours	1.79	1.13	57.44	70.63	79.33	50.74	2.36	51.45
Qwen2-vl-2B-Instruct	Vanilla	3.44	2.82	55.65	70.59	75.41	49.32	2.21	53.13
	SparseVLM	2.86	2.32	52.85	67.06	71.64	46.85	2.10	50.25
	VLA-Cache	2.98	2.12	55.60	70.55	74.35	48.14	2.19	53.08
	FlashVLA	2.24	1.79	54.58	70.52	75.33	49.28	2.20	51.97
	Ours	1.64	1.39	55.25	70.11	75.01	48.97	2.21	53.13
Phi-3.5-vision	Vanilla	5.55	2.04	56.45	70.11	79.94	50.80	2.48	53.05
	SparseVLM	5.06	1.78	53.62	66.60	75.94	48.25	2.35	49.87
	VLA-Cache	4.67	1.88	56.38	70.03	79.85	50.15	2.47	53.05
	FlashVLA	3.28	1.35	56.40	70.11	79.90	50.75	2.47	52.86
	Ours	1.92	0.89	56.23	69.74	79,49	50.41	2.46	52.96
InternVL2-4B	Vanilla	1.64	1.50	55.43	69.23	76.38	47.31	2.17	47.45
	SparseVLM	1.60	1.48	52.66	65.77	72.56	44.94	2.06	45.52
	VLA-Cache	1.48	1.31	55.40	69.15	75.92	47.24	2.15	47.41
	FlashVLA	0.97	0.77	55.12	69.20	76.32	47.28	2.16	46.35
	Ours	0.40	0.35	55.11	69.12	76.29	46.97	2.16	47.45
MobileVLM-3B	Vanilla	2.63	1.95	53.41	67.64	74.31	45.51	2.11	48.23
	SparseVLM	2.37	1.70	50.73	64.26	70.59	43.23	2.00	46.19
	VLA-Cache	2.39	1.67	52.35	67.57	72.25	44.43	2.09	48.13
	FlashVLA	1.78	1.33	53.38	67.60	74.28	45.48	2.10	48.19
	Ours	0.96	0.80	53.05	67.26	73.91	45.18	2.10	48.23
Senna	Vanilla	2.87	2.06	71.28	78.98	91.43	55.37	3.41	65.04
	SparseVLM	2.25	1.41	59.86	70.15	88.75	51.45	2.89	60.33
	VLA-Cache	2.36	1.53	70.35	77.86	91.38	52.12	3.15	65.02
	FlashVLA	1.73	1.26	70.20	77.90	91.16	55.30	3.40	65.01
	Ours	1.33	0.90	70.97	78.32	91.33	51.97	3.15	65.01

Table 1: Evaluation of inference latency and accuracy metrics across different VLMs and methods on NVIDIA AGX Orin and RTX 3090 hardware platforms. The proposed method achieves significant speedup with minimal accuracy degradation, demonstrating its effectiveness in balancing efficiency and reasoning quality.

from scratch based on the current observation I_t. The occurrence of $D_j \wedge D_{j+1}$ serves as a robust indicator of semantic drift, ensuring that the system resets only when the reasoning chain has demonstrably broken down.

5 Experiments

5.1 Setup and Implementation Details

Datasets. We conduct experiments on two autonomous driving benchmarks. DriveLM(Sima et al., 2024) is a manually annotated dataset based on nuScenes(Caesar et al., 2020), includes 5k+ multi-view camera sequences paired with human-verified scene descriptions and meta-actions, 91.4 QA pairs per frame on average. OmniDrive(Wang et al., 2024b) is a synthetic dataset generated using GPT-4, which extends nuScenes(Caesar et al., 2020) with diverse driving scenarios and language-based reasoning tasks.

Safety-critical Subsets. To test the system under pressure, we filter full datasets to create the challenge subsets containing only safety-critical frames where the vehicle's meta-action changes, such as responding to a sudden obstacle or a new traffic signal. Performance on this set directly measures the system's ability to handle the abrupt, safety-critical events that define real-world driving.

Models. Our evaluation includes a set of VLMs to show the broad applicability of our method: LLaVA-v1.6-vicuna-7B(Liu et al., 2023), Qwen2-VL-2B(Wang et al., 2024a), Phi-3.5-Vision(Abdin et al., 2024), InternVL2-4B(Chen et al., 2024f), and MobileVLM-V2-3B(Chu et al., 2024). We also conducted experiments on Senna(Jiang et al., 2024), a VLM specifically fine-tuned for autonomous driving, to show our approach can further optimize even specialized models.

Hardware. Experiments are conducted on NVIDIA Jetson AGX Orin (64GB) designed for

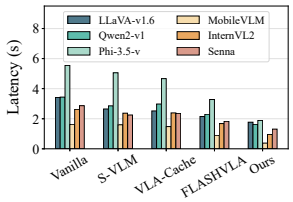


Figure 5: Inference latency comparison of methods on NVIDIA Jetson Orin.

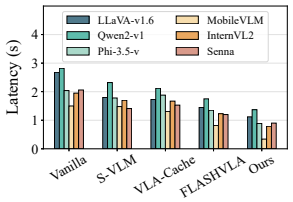


Figure 6: Inference latency comparison of methods on NVIDIA RTX 3090.

in-vehicle computing and NVIDIA RTX 3090 for server-grade performance.

Metrics. We evaluate performance on two primary fronts: efficiency, measured as end-to-end inference time per frame and decision-making accuracy. We evaluate accuracy using the F1 score, reported for direction, speed (fast, normal, slow, stop) and their combination. The quality of the underlying reasoning is evaluated using standard metrics: BLEU-4(Papineni et al., 2002) for lexical overlap and CIDEr(Vedantam et al., 2015) for semantic relevance.

5.2 Overall Performance

We compare our method against three baselines. Vanilla refers to the standard, unmodified VLM, fine-tuned on the respective datasets. We apply SmoothQuant(Xiao et al., 2023) to these models to simulate a realistic, performance-optimized deployment. SparseVLM(Zhang et al., 2024b) prunes visual tokens based on textual guidance to reduce computational load. VLA-Cache(Xu et al., 2025) caches static tokens across sequential data, avoiding recomputation for unchanged regions. FLASHVLA(Tan et al., 2025) reuses token-aware actions when the visual input remains stable.

Model	Type	Accuracy (F1)↑ All	Speed	Direction
llava-v1.6-vicuna-7b	full	57.44	70.63	79.33
	w/o AC	56.75	71.43	75.67
	w/ DT	40.14	51.59	54.75
	w/o MR	44.76	53.41	61.73
Phi-3.5-vision	full	56.23	69.74	79,49
	w/o AC	54.99	67.35	78.98
	w/ DT	45.24	61.84	65.48
	w/o MR	42.92	52.75	55.13

Table 2: Ablation study of the proposed framework's submodules. The table reports accuracy metrics to demonstrate the impact of each module on reasoning quality. "AC" denotes attention-aware calibration. "DT" denotes discontinuous KV cache truncation."MR" denotes memory refresh.

Inference Speedup. MEMO-VLM delivers a significant inference speedup, ranging from 1.7× to 4.3× across all tested models and hardware (Fig. 5, 6). This advantage holds even for highly optimized models like Senna(Jiang et al., 2024), which we accelerate by 2.3×. Notably, on an edge device (AGX Orin), our method transforms the InternVL2-4B model from an impractical 1.64 seconds per frame to a real-time capable 0.40 seconds. This performance consistently surpasses input-level optimizations like SparseVLM (1.5×-4.2×) and VLA-Cache (1.4×-3.9×). Unlike FLASHVLA that discards entire reasoning chains, our fine-grained validation selectively preserves valid tokens while only removing outdated ones, highlighting the clear superiority of reusing semantic reasoning.

Reasoning Quality. The speedups achieved by MEMO-VLM do not come at the expense of reasoning quality or safety. While other acceleration techniques often incur significant performance penalties, Table 1 shows our approach induces a degradation of less than 1% compared to vanilla inference. This confirms that our semantic reuse avoids the harmful truncation of critical tokens.

Failure Case Analysis. Our method exhibits remarkable reliability on the safety-critical challenge set, with **near-zero** accuracy loss.A closer analysis of the rare failure cases reveals they are conservative when it chooses to "maintain speed" rather than accelerate into an open road. This conservative bias in ambiguous contexts ensures these nuanced discrepancies do not compromise safety.

5.3 Ablation Study

Submodule Validity. As shown in Table 2, removing the attention-aware calibration module reduces meta-action F1 accuracy by 1.2–2.3 points, underscoring its critical role in prioritizing safety-relevant tokens (e.g., "pedestrian") over syntactical or low-attention elements. Furthermore, we ex-

Model	Threshold	Latency (s) ↓ Orin	3090	Accuracy (F1) ↑ Direction	Speed
llava-v1.6-vicuna-7b	$1.1 \times \tau_t$	1.94	1.34	70.66	79.33
	τ_t	1.79	1.13	70.63	79.33
	$0.9 \times \tau_t$	1.67	1.05	69.76	78.21
Phi-3.5-vision	$1.1 \times \tau_t$	2.13	1.05	69.76	79.46
	τ_t	1.92	0.89	69.74	79,49
	$0.9 \times \tau_t$	1.75	0.85	68.71	77.90

Table 3: Sensitivity analysis of threshold.

periment with discontinuous KV cache truncation, where invalid tokens are removed mid-sentence without preserving the preceding valid prefix. This approach leads to significant performance degradation of more than 15%, as tokens following the false memory absorb previously invalid information, introducing noise and confusion into the model's reasoning process. These results validate the effectiveness of our prefix-preserving truncation strategy, which maintains grammatical coherence and semantic consistency by retaining valid token sequences while discarding only the invalid suffix. Disabling memory refresh mechanism led to a substantial drop of more than 10%, in both scene understanding and action accuracy, demonstrating that our system for bounding error propagation is crucial for maintaining high performance over time.

Sensitivity Analysis of Threshold τ_t. We evaluate the trade-off between latency and accuracy by changing τ_t. As shown in Table 3, a stricter threshold (higher τ_t) will marginally improve the accuracy rate, but it will increase the delay because the correct description is recognized as outdated. Our choice of τ_t as the minimum token probability establishes it as the lower bound of the optimal interval. A looser threshold (lower τ_t) will reduce latency but lower accuracy, which proves the reliability of our method in the selection of τ_t. This suggests the method becomes too tolerant of semantic shifts, potentially missing subtle but crucial changes in the driving scene.

Cross-dataset Generalizability. To evaluate cross-dataset generalizability, we conducted experiments using threshold calibrated on NuScenes and apply it to the BDD-X dataset with our automatically threshold finetune mechanism. BDD-X contains diverse driving scenarios (e.g., day/night, highway/city) and includes 8.4M frames with annotated actions and explanations. We first calibrated our threshold on 5k frames from nuScenes. We then applied resulting thresholds directly to a 5k-frame test set from BDD-X. The results in Table 4 demonstrate minimal degradation compared to baseline. Thus, our automatically threshold finetune mechanism generalize well under distribution shifts.

Model	Type	Accuracy (F1)↑ All	Direction	Speed
llava-v1.6-vicuna-7b	Vanilla	53.05	61.53	73.64
	Trans	52.21	61.19	73.11
Phi-3.5-vision	Vanilla	51.32	60.75	71.34
	Trans	50.45	60.35	70.56

Table 4: Evaluation of accuracy metrics across different VLMs and methods. "Trans" stands for threshold calibrated on NuScenes and applied it to the BDD-X dataset with automatically threshold finetune mechanism.

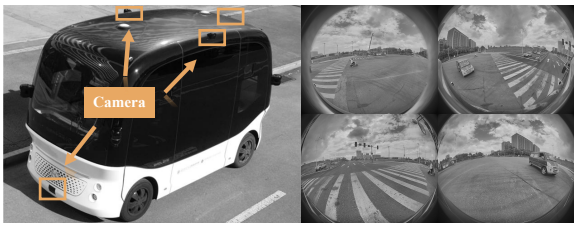


Figure 7: Evaluation on Baidu Apollo2 bus.

5.4 Real World Deployment

To validate real-world applicability, we deployed InternVL2-4B on a Baidu Apollo 2.0 vehicle equipped with an NVIDIA RTX 3090Ti GPU and Intel(R) Core(TM) i7-12700H CPU, running in a shadow mode for two weeks in complex urban environments (Fig. 7). MEMO-VLM proved crucial, enabling real-time control at 0.35 seconds per frame and achieving 68.18% meta-action accuracy. A qualitative review of failures revealed a consistent pattern of safe, conservative actions; the model's primary errors were not dangerous decisions but rather an overcautious failure to accelerate where a human would. This tendency towards caution, rather than recklessness, supports the method's viability for real-world deployment. To correct this over-caution, future work will focus on improving the reward function to better balance safety with driving efficiency. We will also use imitation learning, training the model on data from skilled drivers to help it learn how to make more confident yet safe driving decisions.

6 Conclusion

We present a memory-driven framework that accelerates vision-language model (VLM) inference in autonomous driving by addressing temporal redundancy in consecutive scene reasoning. Experiments demonstrate 1.7×-4.3× speedup on edge hardware with less than 1% accuracy degradation in meta-action prediction, reducing redundant computations by 70% in stable scenarios. This work bridges bio-inspired memory mechanisms with computational efficiency, offering a practical solution for real-time autonomous systems .

7 Ethical Considerations

Our research focuses on accelerating Vision-Language Models (VLMs) for autonomous driving, a domain with safety and societal implications. The primary ethical consideration of this work centers on the safety of deploying approximated reasoning in safety-critical systems. By reusing past reasoning states to reduce latency, MEMO-VLM introduces a trade-off between computational efficiency and the potential risk of missing sudden environmental changes. To mitigate this, we designed our framework with bounded error propagation (as detailed in Section 4.4), ensuring that the system defaults to a full refresh of reasoning when confidence drops, rather than hallucinating safety. In our real-world deployment, we utilized a "shadow mode" setup where the system ran passively alongside a human driver, ensuring no physical risk to public safety during testing.

Our experiments rely on publicly available, standard benchmarks (NuScenes, DriveLM, OmniDrive, BDD-X). We adhere to the usage licenses of these datasets, which are designed to anonymize personally identifiable information such as license plates and faces. Furthermore, our approach contributes to "Green AI" initiatives. By reducing redundant computation in VLMs, MEMO-VLM significantly lowers the energy consumption and carbon footprint associated with running large-scale models on edge devices, making advanced AI more environmentally sustainable for widespread deployment.

Limitations

MEMO-VLM exhibits a tendency toward overcaution. As observed in our real-world deployment (Section 5.4), the model occasionally fails to accelerate into open roads due to strict consistency thresholds. While this conservative behavior ensures safety, it may lead to unnatural driving patterns that could impede traffic flow or differ from human-like fluidity. Future work is required to better balance safety with driving efficiency via improved reward functions.

Furthurmore, we observed diminishing returns when combining our temporal reuse strategy with aggressive spatial pruning methods. As detailed in Section 5.2 and Table 5, integrating MEMO-VLM with spatial redundancy techniques like SparseVLM led to a notable degradation in meta-action accuracy (dropping by over 10 points in some cases). This indicates that simultaneously reducing information in both spatial and temporal dimensions can erode the semantic context necessary for complex reasoning, limiting the composability of our framework with other extreme compression techniques.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, and 1 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*.

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

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631.

Hanning Chen, Yang Ni, Wenjun Huang, Yezi Liu, SungHeon Jeong, Fei Wen, Nathaniel Bastian, Hugo Latapie, and Mohsen Imani. 2024a. Vltp: Vision-language guided token pruning for task-oriented segmentation. *arXiv preprint arXiv:2409.08464*.

Jiao Chen, Suyan Dai, Fangfang Chen, Zuohong Lv, and Jianhua Tang. 2024b. Edge-cloud collaborative motion planning for autonomous driving with large language models. *arXiv preprint arXiv:2408.09972*.

Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. 2024c. End-to-end autonomous driving: Challenges and frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Xuanyao Chen, Zhijian Liu, Haotian Tang, Li Yi, Hang Zhao, and Song Han. 2023. Sparsevit: Revisiting activation sparsity for efficient high-resolution vision transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2061–2070.

Yuan Chen, Zi-han Ding, Ziqin Wang, Yan Wang, Lijun Zhang, and Si Liu. 2024d. Asynchronous large language model enhanced planner for autonomous driving. In *European Conference on Computer Vision*, pages 22–38. Springer.

Yukang Chen, Fuzhao Xue, Dacheng Li, Qinghao Hu, Ligeng Zhu, Xiuyu Li, Yunhao Fang, Haotian Tang, Shang Yang, Zhijian Liu, and 1 others. 2024e. Longvila: Scaling long-context visual language models for long videos. *arXiv preprint arXiv:2408.10188*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024f. 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*, pages 24185–24198.

Xiangxiang Chu, Limeng Qiao, Xinyu Zhang, Shuang Xu, Fei Wei, Yang Yang, Xiaofei Sun, Yiming Hu, Xinyang Lin, Bo Zhang, and 1 others. 2024. Mobilevlm v2: Faster and stronger baseline for vision language model. *arXiv preprint arXiv:2402.03766*.

Matthew Dutson, Yin Li, and Mohit Gupta. 2023. Eventful transformers: Leveraging temporal redundancy in vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16911–16923.

Bowen Feng, Zhiting Mei, Baiang Li, Julian Ost, Roger Girgis, Anirudha Majumdar, and Felix Heide. 2025. Verdi: Vlm-embedded reasoning for autonomous driving. *arXiv preprint arXiv:2505.15925*.

Yichen Guo, Hanze Li, Zonghao Zhang, Jinhao You, Kai Tang, and Xiande Huang. 2025. Star: Stage-wise attention-guided token reduction for efficient large vision-language models inference. *arXiv preprint arXiv:2505.12359*.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, and 1 others. 2023. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862.

Hsiang-Wei Huang, Wenhao Chai, Kuang-Ming Chen, Cheng-Yen Yang, and Jenq-Neng Hwang. 2025. Tosa: Token merging with spatial awareness. *arXiv preprint arXiv:2506.20066*.

Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. 2024. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv preprint arXiv:2410.22313*.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. 2023. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350.

Yawei Li, Babak Ehteshami Bejnordi, Bert Moons, Tijmen Blankevoort, Amirhossein Habibian, Radu Timofte, and Luc Van Gool. 2021. Spatio-temporal gated transformers for efficient video processing. In *Advances in Neural Information Processing Systems Workshops*, volume 3(7), page 8.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning.

Keke Long, Haotian Shi, Jiaxi Liu, and Xiaopeng Li. 2024. Vlm-mpc: Vision language foundation model (vlm)-guided model predictive controller (mpc) for autonomous driving. *arXiv preprint arXiv:2408.04821*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.

Kangan Qian, Sicong Jiang, Yang Zhong, Ziang Luo, Zilin Huang, Tianze Zhu, Kun Jiang, Mengmeng Yang, Zheng Fu, Jinyu Miao, and 1 others. 2025. Agentthink: A unified framework for tool-augmented chain-of-thought reasoning in vision-language models for autonomous driving. *arXiv preprint arXiv:2505.15298*.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. 2021. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *Advances in neural information processing systems*, 34:13937–13949.

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

Boyuan Sun, Jiaxing Zhao, Xihan Wei, and Qibin Hou. 2025. Llava-scissor: Token compression with semantic connected components for video llms. *arXiv preprint arXiv:2506.21862*.

Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. 2024. Sparsedrive: End-to-end autonomous driving via sparse scene representation. *arXiv preprint arXiv:2405.19620*.

Xudong Tan, Yaoxin Yang, Peng Ye, Jialin Zheng, Bizhe Bai, Xinyi Wang, Jia Hao, and Tao Chen. 2025. Think twice, act once: Token-aware compression and action reuse for efficient inference in vision-language-action models. *arXiv preprint arXiv:2505.21200*.

Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Yang Wang, Zhiyong Zhao, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. 2024. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*.

Pavan Kumar Anasosalu Vasu, Fartash Faghri, Chun-Liang Li, Cem Koc, Nate True, Albert Antony, Gokul Santhanam, James Gabriel, Peter Grasch, Oncel Tuzel, and 1 others. 2024. Fastvlm: Efficient vision encoding for vision language models. *arXiv preprint arXiv:2412.13303*.

Ramakrishna Vedantam, C Lawrence Zitnick, and Devi Parikh. 2015. Cider: Consensus-based image description evaluation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4566–4575.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024a. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Shihao Wang, Zhiding Yu, Xiaohui Jiang, Shiyi Lan, Min Shi, Nadine Chang, Jan Kautz, Ying Li, and Jose M Alvarez. 2024b. Omnidrive: A holistic llm-agent framework for autonomous driving with 3d perception, reasoning and planning. *arXiv preprint arXiv:2405.01533*.

Wenhai Wang, Jiangwei Xie, ChuanYang Hu, Haoming Zou, Jianan Fan, Wenwen Tong, Yang Wen, Silei Wu, Hanming Deng, Zhiqi Li, and 1 others. 2023. Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving. *arXiv preprint arXiv:2312.09245*.

Licheng Wen, Daocheng Fu, Xin Li, Xinyu Cai, Tao Ma, Pinlong Cai, Min Dou, Botian Shi, Liang He, and Yu Qiao. 2023. Dilu: A knowledge-driven approach to autonomous driving with large language models. *arXiv preprint arXiv:2309.16292*.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR.

Siyu Xu, Yunke Wang, Chenghao Xia, Dihao Zhu, Tao Huang, and Chang Xu. 2025. Vla-cache: Towards efficient vision-language-action model via adaptive token caching in robotic manipulation. *arXiv preprint arXiv:2502.02175*.

Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. 2024. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *IEEE Robotics and Automation Letters*.

Senqiao Yang, Yukang Chen, Zhuotao Tian, Chengyao Wang, Jingyao Li, Bei Yu, and Jiaya Jia. 2024. Visionzip: Longer is better but not necessary in vision language models. *arXiv preprint arXiv:2412.04467*.

Xiaoqing Ye, Mao Shu, Hanyu Li, Yifeng Shi, Yingying Li, Guangjie Wang, Xiao Tan, and Errui Ding. 2022. Rope3d: The roadside perception dataset for autonomous driving and monocular 3d object detection task. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21341–21350.

Qizhe Zhang, Aosong Cheng, Ming Lu, Zhiyong Zhuo, Minqi Wang, Jiajun Cao, Shaobo Guo, Qi She, and Shanghang Zhang. 2024a. [cls] attention is all you need for training-free visual token pruning: Make vlm inference faster. *arXiv preprint arXiv:2412.01818*.

Rongyu Zhang, Menghang Dong, Yuan Zhang, Liang Heng, Xiaowei Chi, Gaole Dai, Li Du, Yuan Du, and Shanghang Zhang. 2025. Mole-vla: Dynamic layer-skipping vision language action model via mixture-of-layers for efficient robot manipulation. *arXiv preprint arXiv:2503.20384*.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and 1 others. 2024b. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint arXiv:2410.04417*.

Yupeng Zheng, Zebin Xing, Qichao Zhang, Bu Jin, Pengfei Li, Yuhang Zheng, Zhongpu Xia, Kun Zhan, Xianpeng Lang, Yaran Chen, and 1 others. 2024. Planagent: A multi-modal large language agent for closed-loop vehicle motion planning. *arXiv preprint arXiv:2406.01587*.

Xirui Zhou, Lianlei Shan, and Xiaolin Gui. 2025a. Dynrsl-vlm: Enhancing autonomous driving perception with dynamic resolution vision-language models. *arXiv preprint arXiv:2503.11265*.

Zewei Zhou, Tianhui Cai, Seth Z Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. 2025b. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *arXiv preprint arXiv:2506.13757*.

A Integration with Image Token Redundancy

Our method primarily optimizes reasoning by leveraging semantic temporal redundancy, which is orthogonal to approaches that exploit the temporal and spatial redundancy of image tokens. To explore the potential synergy between these paradigms, we conduct experiments combining our framework with VLA-Cache(temporal token caching) and SparseVLM(spatial token pruning). As shown in table 5, while the fusion achieves faster inference speeds, it also leads to performance degradation, with meta-action F1 accuracy dropping by 1.68–10.4 points. This trade-off arises because autonomous driving is a highly sensitive task where excessive pruning of information, whether spatial or temporal, can significantly impair reasoning quality and decision-making reliability. These results highlight the importance of balancing efficiency with the preservation of safety-critical details in real-world autonomous systems.

B System Overhead

MEMO-VLM maintains high efficiency by imposing negligible computational and memory overhead. As shown in Fig. 8 and 9, our coarse-to-fine

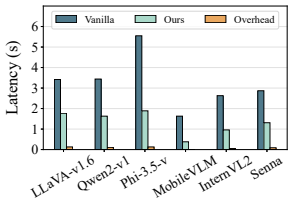


Figure 8: End-to-end overhead on Jetson Orin.

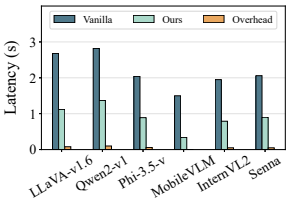


Figure 9: End-to-end overhead on RTX 3090.

redundancy detection pipeline constitutes less than 5% of the total inference latency, a result of its tight integration within the VLM's prefill stage that obviates the need for external networks. Similarly, the runtime memory footprint is minimal; its KV cache is transient, generated and discarded on a per-frame basis, retaining only a small set of memory tokens that occupy just tens of KBs.

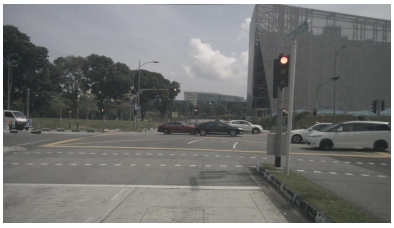
C Qualitative Results

Fig. 10 illustrates a busy intersection scenario where our framework reuses 83% of cached tokens across three consecutive frames. In our global semantic check module, we identified that the left and middle diagrams are similar, prompting the model to retain its full reasoning output. However, we detected a difference in scenes between the middle and right diagrams, as the car approached the intersection. Consequently, the model preserved part of the previous output while updating the meta-action to "stop".

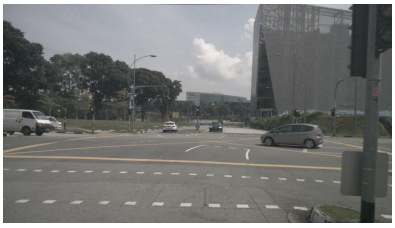
Model	Method	Latency(s) ↓		Accuracy (F1)↑			BLEU-4↑	CIDEr↑
		Orin	3090	All	Direction	Speed		
llava-v1.6-vicuna-7b	Ours	1.79	1.13	**57.44**	**70.63**	**79.33**	**50.74**	**2.36**
	Ours + SparseVLM	1.57	0.99	49.25	64.74	73.85	43.52	1.94
	Ours + VLA-Cache	**1.49**	**0.95**	55.65	70.20	75.54	49.52	2.26
Phi-3.5-vision	Ours	1.92	0.89	**56.23**	**69.74**	**79,49**	**50.41**	**2.46**
	Ours + SparseVLM	1.77	0.86	45.76	62.43	67.87	44.76	1.88
	Ours + VLA-Cache	**1.68**	**0.84**	53.51	66.21	76.65	48.22	2.15

Table 5: Comparison of inference latency and accuracy metrics when integrating our framework with spatial redundancy methods. While the fusion achieves faster inference speeds, it leads to performance degradation, indicating that excessive pruning of information can impair reasoning quality in safety-critical tasks.

The driving scene is a busy intersection with multiple traffic lights and vehicles. There are several cars and a truck in the area, some of which are stopped at the traffic lights. A few cars are moving through the intersection. There are multiple traffic lights in the scene, indicating that the intersection is well-regulated. The car's front camera is facing the **red traffic light**. The car is positioned in the middle of the intersection, and there are no pedestrians visible in the scene. Based on the image, the driving direction should be **straight**, the ego vehicle should drive **slowly** as there are other vehicles and pedestrians in the area.



The driving scene is a busy intersection with multiple traffic lights and vehicles. There are several cars and a truck in the area, some of which are stopped at the traffic lights. A few cars are moving through the intersection. There are multiple traffic lights in the scene, indicating that the intersection is well-regulated. The car's front camera is facing the **red traffic light**. The car is positioned in the middle of the intersection, and there are no pedestrians visible in the scene. Based on the image, the driving direction should be **straight**, the ego vehicle should drive **slowly** as there are other vehicles and pedestrians in the area.



The driving scene is a busy intersection with multiple traffic lights and vehicles. There are several cars and a truck in the area, some of which are stopped at the traffic lights. A few cars are also moving through the intersection. There are multiple traffic lights in the scene, indicating that the intersection is well-regulated. The **traffic lights are currently red**, indicating that the ego vehicle should **stop**. Based on the image, the ego vehicle should proceed with caution when the traffic light turns green.

Figure 10: Qualitative result of our framework in a busy intersection scenario. The left and middle diagrams show similar scenes, prompting the model to retain its full reasoning output. However, a difference is detected between the middle and right diagrams as the car approaches the intersection. Consequently, the model preserves part of the previous scene description while updating the meta-action to "stop". This illustrates how our framework adaptively reuses valid reasoning states and updates only the necessary parts of the scene description.