AdaDrive: Self-Adaptive Slow-Fast System for Language-Grounded Autonomous Driving

Ruifei Zhang[1,2], Junlin Xie[1,2], Wei Zhang[4†], Weikai Chen[†‡], Xiao Tan[4], Xiang Wan[2], Guanbin Li[3,5†]
[1] The Chinese University of Hong Kong, Shenzhen [2] Shenzhen Research Institute of Big Data
[3] Sun Yat-sen University [4] Baidu Inc.
[5]Guangdong Key Laboratory of Big Data Analysis and Processing

Abstract

Effectively integrating Large Language Models (LLMs) into autonomous driving requires a balance between leveraging high-level reasoning and maintaining real-time efficiency. Existing approaches either activate LLMs too frequently, causing excessive computational overhead, or use fixed schedules, failing to adapt to dynamic driving conditions. To address these challenges, we propose AdaDrive, an adaptively collaborative slow-fast framework that optimally determines when and how LLMs contribute to decision-making. (1) ***When*** *to activate the LLM: AdaDrive employs a novel adaptive activation loss that dynamically determines LLM invocation based on a comparative learning mechanism, ensuring activation only in complex or critical scenarios. (2)* ***How*** *to integrate LLM assistance: Instead of rigid binary activation, AdaDrive introduces an adaptive fusion strategy that modulates a continuous, scaled LLM influence based on scene complexity and prediction confidence, ensuring seamless collaboration with conventional planners. Through these strategies, AdaDrive provides a flexible, context-aware framework that maximizes decision accuracy without compromising real-time performance. Extensive experiments on language-grounded autonomous driving benchmarks demonstrate that AdaDrive state-of-the-art performance in terms of both driving accuracy and computational efficiency. Code is available at* https://github.com/ReaFly/AdaDrive.

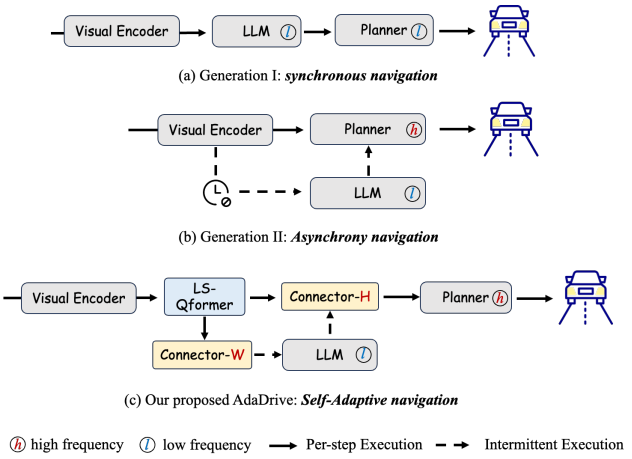

Figure 1. (a) The first generation of LLM-enhanced autonomous driving approaches [21, 38] employ a synchronous structure, where both LLM and planner operate at each driving step. (b) Generation II methods implement asynchronous processing paradigms, utilizing distinct but predetermined activation frequencies for the LLM and planner. (c) Our proposed AdaDrive also employs an asynchronous architecture but features two novel adaptive connectors: **Connector-W** for adaptively determining **when** to activate the LLM, and **Connector-H** for controlling **how** to integrate the LLM in driving tasks. This design enables enhanced flexibility in handling uncertain or emergency situations. Besides, we also incorporate LS-Qformer for efficient processing of continuous streaming data.

1. Introduction

Autonomous driving has long been a focal point in both academia and industry [1, 2, 5, 9, 10, 12, 20, 22, 26, 32, 34, 36, 37]. With the emergence of large language models (LLMs) and their multimodal extensions (MLLMs), researchers have begun integrating LLMs into autonomous driving systems to enhance cognitive reasoning and decision making [7, 8, 15–17, 24, 30]. Early approaches, such as LMDrive [21] and AD-H [38], adopt synchronous and highly-entangled sequential architectures where LLMs continuously influence the driving process at every step (see Figure 1). While these models improve driving intelligence, they introduce substantial memory overhead and latency, making real-time deployment challenging, particularly in

†Co-corresponding authors: zhangwei99@baidu.com, chenwk891@gmail.com, liguanbin@mail.sysu.edu.cn.
‡This paper solely reflects the author's personal research and is not associated with the author's affiliated institution.

high-speed, dynamic environments. To address this issue, subsequent research [4, 25] has explored asynchronous strategies, where LLM activation occurs at pre-defined intervals to balance performance and efficiency. However, these fixed schedules greatly limit model adaptability, as the need for LLM intervention varies significantly across different driving scenarios. For instances, in safety-critical situations, LLMs may not be invoked when they are needed most. Conversely, in simple scenarios, activating LLMs may be unnecessary, leading to suboptimal resource utilization.

Given these limitations, an ideal LLM-enhanced autonomous driving framework should be able to: 1) ***Dynamically decide when to activate the LLM***, ensuring that LLMs contribute only in scenarios where they are beneficial while avoiding unnecessary computational overhead; 2) ***Adaptively control the degree of LLM influence***, as our key insight reveals that while LLM engagement consistently enhances performance, a binary on/off activation with full weight (e.g. 1.0) can often be suboptimal compared to a continuous, scaled integration with a lower adaptive weight (e.g. 0.7) (see results in Table 3: ID #3 vs. ID #4).

To address these challenges, we introduce **AdaDrive**, a next-generation self-adaptive LLM-integration framework for autonomous driving. In particular, AdaDrive leverages a slow-fast system paradigm to balance *high-frequency low-latency tasks* (a lightweight planner without invoking the LLM, referred to as, the *fast path*) and *low-frequency high-reasoning tasks* (where the LLM is activated as a cognitive agent, also known as, the *slow path*).

We optimize this slow-fast framework to achieve an optimal balance between decision accuracy and computational efficiency with two key innovations. 1) **Adaptive LLM Activations**. Instead of relying on fixed activation intervals, AdaDrive learns when to engage the LLM dynamically through a novel adaptive activation loss. By comparing LLM-assisted and LLM-free prediction during training, our model automatically identifies high-risk or complex situations where LLM intervention is most beneficial, ensuring a real on-demand activation. 2) **Dynamic LLM Contribution Scaling**. Unlike prior methods that treat LLM engagement as a binary decision, AdaDrive introduces a confidence-driven fusion strategy that adjusts the weights of LLM involvement dynamically. Our key insight is that while LLM assistance consistently improves performance, treating its activation as a binary decision with full weighting can be suboptimal -— adaptive scaling of LLM contributions often yields better results than an all-or-nothing approach.(see results in Table 3: ID #3 vs. ID #4). To counter this, AdaDrive modulates the strength of LLM influence based on the confidence of the LLM output and scene complexity, ensuring that its contributions are optimally balanced with conventional planning modules.

In addition, we propose Long-Short Q-former (LS-Qformer) to enhance visual modeling by integrating short-term precision with long-term contextual retention, ensuring consistent trajectory predictions in streaming autonomous driving. We also introduce Propagative Memory Fusion (PMF) mechanism to further optimize memory efficiency by merging evicted frame features into adjacent frames, preserving critical historical context while maintaining a compact representation. Experimental results demonstrate that AdaDrive sets a new state of art in language-grounded autonomous driving. We summarize our contributions as follows:

- We introduce AdaDrive, the first self-adaptive slow-fast architecture for LLM-enhanced autonomous driving, enabling dynamic LLM activation based on real-time driving contexts.
- We propose a novel adaptive integration mechanism, which automatically (i) learns when to activate the LLM for maximum performance gains while minimizing computational overhead, and (ii) determines how much the LLM should contribute based on model confidence and scene complexity.
- We develop LS-Qformer and PMF mechanism to enhance temporal feature aggregation and preserve critical historical context through efficient memory retention.
- We achieve state-of-the-art performance on standard language-grounded autonomous driving benchmarks in terms of both accuracy and computational efficiency.

2. Related Work

2.1. End-to-End Autonomous Driving

Imitation learning [5, 9, 32] and reinforcement learning [2, 12, 26, 37] are two primary approaches for end-to-end autonomous driving. Significant advancements have been made in both directions in recent years. InterFuser [22] enhances driving safety by effectively leveraging multi-modal, multi-view sensor data and utilizing intermediate interpretable features to constrain actions within a safe set. ReasonNet [23] focuses on global information comprehension and temporal context reasoning, substantially improving the prediction accuracy of object behaviors and enhancing system robustness in challenging scenarios. UniAD [10] proposes a novel modular end-to-end framework that unifies full-stack driving tasks within a single network, enhancing inter-module collaboration for optimal planning performance.

2.2. LLMs for Autonomous Driving

Recently, with the emergence of LLMs, their impressive logical reasoning capabilities have catalyzed the integration with autonomous driving systems [4, 21, 24, 25, 29, 30]. As a pioneering effort, LMDrive [21] utilizes

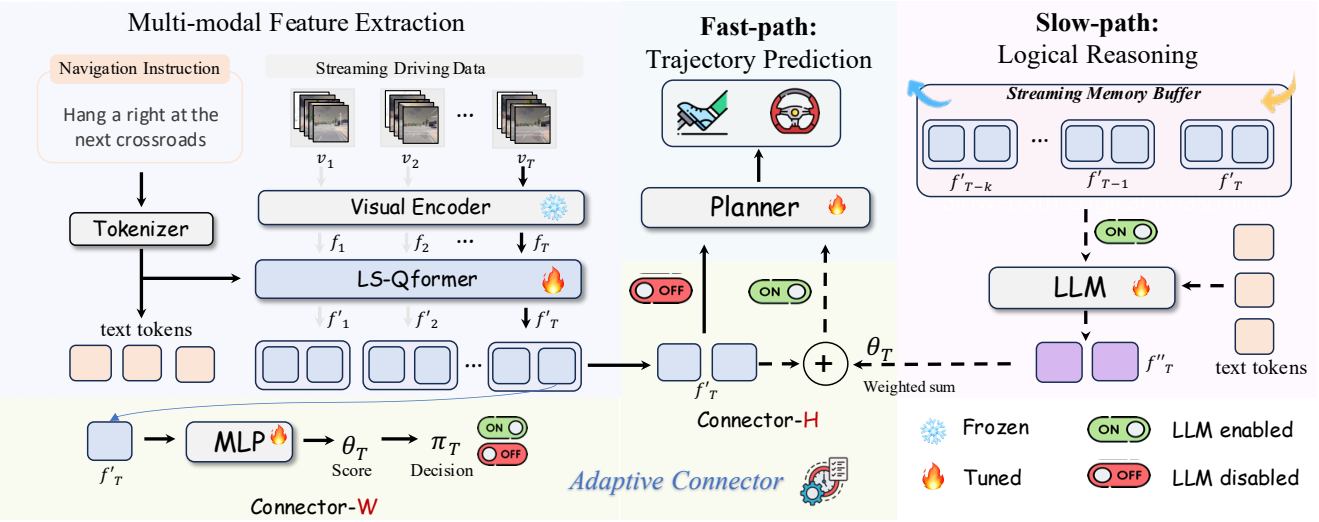

Figure 2. An overview of AdaDrive framework, comprising generic multi-modal feature extraction and parallel slow-fast paths dedicated to logical reasoning and trajectory prediction. The two paths are adaptively integrated through our proposed Connector-W and Connector-H components, determining **when** to activate the LLM and **how** to integrate the LLM for trajectory prediction, respectively. Dashed lines indicate intermittent execution, which only occurs when LLM is enabled.

LLMs to comprehend natural language navigation instructions and predict future waypoints, successfully achieving language-grounded closed-loop autonomous driving. DriveMLM [29] establishes a novel interface between LLMs and autonomous driving systems through semantic mapping of language model reasoning to planners' decision state space. Despite these advancements, the high computational cost and inference latency associated with LLMs limit their practical applications. To tackle this issue, AsyncDriver [4] presents an asynchronous architecture in which the LLM maintains periodic activations to enhance a traditional planner's capabilities. DriveVLM [25] incorporates the MLLM and traditional planner, where low-frequency activated MLLM provides the reference trajectories to a high-frequency activated planner for trajectory refinement. However, these approaches employ fixed frequencies and invocation intervals for LLMs, severely limiting the collaborative operation of the two systems. In contrast, our framework empowers the planner with autonomous LLM activation capabilities, facilitating dynamic model collaboration while maintaining an optimal balance between task performance and computational resources.

2.3. MLLMs for Streaming Understanding

The rapid development of MLLM has showcased versatile capabilities in vision-language comprehension, spatial perception, and video understanding. However, these MLLMs are limited to processing fixed-length images or short clips in an offline manner, constraining their applicability in practical streaming scenarios. Recently, GPT-4o [18] has demonstrated voice-driven online response capabilities. In parallel, a series of studies [3, 19, 31, 40] have made notable strides in streaming video understanding, further pushing the boundaries of practical applications. VideoLLM-online [3] pioneers the extension of offline models into online contexts by introducing a novel training objective with a specialized EOS token, prompting the model to remain silent when responses are unnecessary. Flash-VStream [31] focuses on designing human-like memory modeling to store and process long-term video information while maintaining low inference latency. In contrast to video understanding, which focuses on high-level content comprehension and dialogue, autonomous driving in streaming scenarios emphasizes low-level, high-frequency trajectory prediction. This fundamental distinction motivates us to explore a novel paradigm that optimally balances driving performance with inference latency.

3. Method

3.1. Overview

Problem Definition: Given a sequence of streaming video clip data $\mathbf{V}_T = [v_1, v_2, ..., v_T]$ and corresponding navigation instructions $\mathbf{I}_T$, where T is the current timestamp. This work aims to establish an efficient autonomous driving system $\mathcal{S}$ to generate the instruction-following trajectory prediction:

$$W_T = \mathcal{S}(\mathbf{V}_T, \mathbf{I}_T). \tag{1}$$

Here, W_T represents the predicted waypoints for timestamp T, which is subsequently converted by PID controllers into lateral steering and longitudinal acceleration actions.

System Architecture: As shown in Fig. 2, unlike conventional designs where instruction comprehension and trajectory prediction are entangled within LLMs, our proposed **AdaDrive** decouples these two processes, running them in parallel with distinct activation frequencies. The lightweight planner operates as a low-level trajectory predictor for each frame (*fast path*), while the LLM functions as a central cognitive unit, maintaining low-frequency activation to provide essential assistance to the planner in critical situations (*slow path*). The two paths are adaptively integrated through our proposed Connector-W and Connector-H components, determining when to activate the LLM and how much the LLM should contribute to trajectory prediction, respectively.

3.2. Slow-Fast Systems

Multi-modal Feature Extraction: Given a sequence of streaming video clip data $\mathbf{V}_T = [v_1, v_2, ..., v_T]$, where each frame data comprises multi-view camera images and point cloud data. We employ a pretrained visual encoder [21] $Vis(\cdot)$ to extract and fuse these multi-modal visual features of each frame: $f_t = Vis(v_t)$, thus constructing $\mathbf{F}_T = [f_1, f_2, ..., f_T]$. The subsequent Long-Short Q-former further aggregates the feature tokens in consideration of both long-range and current frame information, denoted as $\mathbf{F}'_T = [f'_1, f'_2, ..., f'_T]$, $f'_t \in \mathbb{R}^{N\times C}$, where N is number of tokens and C is the feature dimension. (section 3.3).

Fast-path Trajectory Prediction: The lightweight planner $\mathcal{P}$ maintains high-frequency activation for each timestamp to generate the waypoint only relying on the current frame information: $W_T = \mathcal{P}(f'_T)$.

Slow-path Logical Reasoning: In contrast to the planner, we endow the LLM with access to long-range context information to fully leverage its instruction comprehension and reasoning capabilities. To prevent unbounded growth in memory usage and computational complexity, making it well-suited for streaming scenarios, we build upon $\mathbf{F}'_T$ by maintaining a streaming memory buffer to manage the features input for LLM (section 3.4). This feature buffer maintains a fixed capacity k and we denote the stored features in the buffer as $\mathbf{B}'_T = [f'_{T-k}, f'_{T-k+1}, ..., f'_T]$. Subsequently, the LLM processes the k-frame contextual information and outputs the integrated features f''_T for current timestamp T as follows:

$$f''_T = \mathcal{LLM}(\mathbf{I}_T, \mathbf{B}'_T) \tag{2}$$

Adaptive Connector: Our framework enhances the slow-fast architecture through adaptive scheduling via two specialized connectors: Connector-W and Connector-H, which orchestrate the interaction between the LLM and the planner. Specifically, Connector-W determines adaptive LLM activations, while Connector-H controls the dynamic scaling of LLM contributions.

Connector-W: Given the current driving context feature f'_T extracted by LS-Qformer, we predict a confidence score that determines the LLM's activation utilizing an MLP function:

$$\theta_T = \mathcal{MLP}(f'_T) \tag{3}$$

The continuous probability distribution θ_T is transformed into a discrete **binary decision** $\pi_T \in \{0, 1\}$ through the Gumbel-Softmax reparameterization, which ensures end-to-end differentiability by maintaining the gradient flow:

$$\pi_T = \text{Gumbel-Softmax}(\theta_T) \tag{4}$$

However, the optimization of π_T presents significant challenges due to the absence of gold standards or ground-truth supervision signals for optimal activation timing. In our work, we propose a novel comparative learning based adaptive activation loss, to address these issues. Specifically, in the training stage, We perform two forward passes for trajectory prediction: one with LLM assistance yielding $W_T^{LLM} = \mathcal{P}(f'_T + f''_T)$, and another without, producing $W_T = \mathcal{P}(f'_T)$. Subsequently, we calculate the trajectory loss (L1 loss) for W_T^{LLM} and W_T, denoted as $\mathcal{L}_T^{LLM}$ and $\mathcal{L}_T$, respectively. Following a warmup phase where both losses converge to stable values, their comparative difference reflects the magnitude of LLM's contribution to trajectory prediction at the current timestep. Thus, we link the binary decision π_T with the trajectory losses to construct a novel **adaptive activation loss**:

$$\mathcal{L}_{ada} = \pi_T * \mathcal{L}_T^{LLM} + (1 - \pi_T) * \mathcal{L}_T \tag{5}$$

Optimizing this objective function naturally induces $\pi_T = 1$ when $\mathcal{L}_T^{LLM} < \mathcal{L}_T$ and 0 otherwise, thereby enabling the model to learn optimal LLM activation conditions. Further, to achieve optimal performance gains while minimizing computational overhead, we introduce a penalty term γ into the LLM-assisted trajectory loss to control the frequency of LLM activations, ensuring LLM is only activated when $\mathcal{L}_T^{LLM}$ is significantly lower than $\mathcal{L}_T$ by a predetermined margin d:

$$\mathcal{L}_{ada} = \pi_T * (\mathcal{L}_T^{LLM} + \gamma) + (1 - \pi_T) * \mathcal{L}_T \tag{6}$$

$$\gamma = max(d - (L_T - L_T^{LLM}), 0) \tag{7}$$

Connector-H: Through our proposed adaptive activation loss, the model (Connector-W) learns to determine optimal LLM activation timing. However, binary fusion (all-or-nothing) may not be the optimal strategy for seamless integration with conventional planners. To enable dynamic LLM contribution scaling, Connector-H leverages

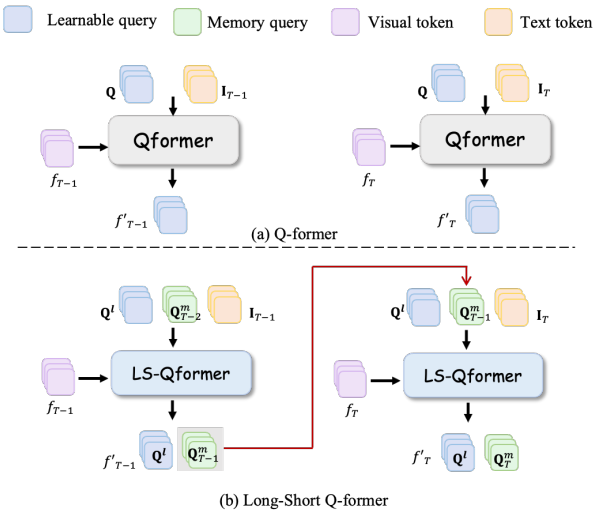

Figure 3. Comparisons between the Q-former and our proposed Long-Short Q-former (LS-Qformer).

the predicted confidence score θ_T as a fusion coefficient for weighted feature integration, generating a third trajectory prediction $W_T^{Fuse} = \mathcal{P}(f'_T + \theta_T * f''_T)$. The trajectory loss computed for W_T^{Fuse} inherently guides the model to learn optimal contribution scaling.

Inference Stage: Connector-W predicts the confidence score θ_T and corresponding binary decision π_T for LLM's activation. Upon LLM activation, Connector-H modulates the contribution of LLM features f''_T to base features f'_T by leveraging the prediction confidence θ_T as a dynamic weighting coefficient. Specifically, the trajectory prediction can be uniformly formulated as follows:

$$W_T = \begin{cases} \mathcal{P}(f'_T), & \text{if LLM is not activated ,} \\ \mathcal{P}(f'_T + \theta_T * f''_T), & \text{if LLM is activated.} \end{cases} \tag{8}$$

3.3. Long-Short Feature Modeling

As a common connector to bridge visual encoder and LLM, Q-former has been applied in many MLLMs. The vanilla Q-former can be formulated as follows:

$$f'_T = \text{Q-former}(\mathbf{Q}, f_T, \mathbf{I}_T) \tag{9}$$

where $\mathbf{Q}$ is additional introduced learnable tokens for feature aggregation. However, this module processes each frame separately while ignoring the long-range temporary information. To tackle this issue, we propose a Long-Short Q-former. Inspired by the group mechanism [6], we partition the learnable tokens into two groups, denoted as $\mathbf{Q}^m$ and $\mathbf{Q}^l$. $\mathbf{Q}^m$ is propagated into the next frame for aggregating long-range information while $\mathbf{Q}^l$ are similar to the vanilla Q-former focusing on the current frame:

$$f'_T = [\mathbf{Q}^l; \mathbf{Q}^m_T] = \text{Q-former}(\mathbf{Q}^l, \mathbf{Q}^m_{T-1}, f_T, \mathbf{I}_T) \tag{10}$$

Through this mechanism, LS-Qformer simultaneously extracts critical features from current frames and models temporal feature evolution, yielding richer visual representations.

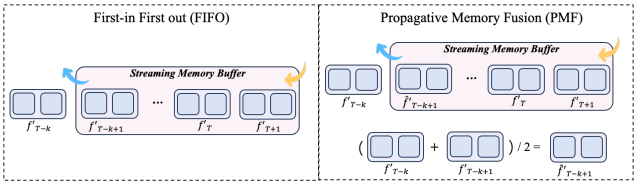

Figure 4. Illustration of FIFO and our proposed PMF. Unlike FIFO, PMF maintains a compact buffer while enabling forward information flow by merging features from to-be-evicted frames into their preceding frames.

3.4. Streaming Memory Buffer

Long-range contextual information is crucial for predicting objects' potential behaviors and trajectories, thereby enabling safer autonomous driving. However, storing and processing continuous streaming data inevitably leads to exponential growth in computational overhead and potential memory overflow. To address these challenges, we propose a fixed-size streaming memory buffer with a Propagative Memory Fusion (PMF) strategy for managing historical driving data (illustrated in Fig. 4). Compared to First-in-First-out (FIFO) which only retains fixed-length features, our PMF mechanism preserves information by merging features of the to-be-evicted frame into its preceding frame, maintaining a compact buffer while enabling forward information propagation:

$$\hat{f}'_{T-k+1} = (f'_{T-k} + f'_{T-k+1})/2 \tag{11}$$

Subsequently, the memory buffer is updated to $\mathbf{B}'_T = [\hat{f}'_{T-k+1}, f'_{T-k+2}, ..., f'_{T+1}]$, where $\hat{f}'_{T-k+1}$ represents the fused features.

4. Experiments

4.1. Experimental Setup

Dataset: We train the AdaDrive on the standard LangAuto dataset [21], a comprehensive multi-modal collection comprising 64K instruction-following sequences. Each sequence encapsulates synchronized multi-view camera images and LiDAR point clouds, providing rich spatiotemporal context for autonomous navigation.

Benchmarks and Metrics: We conduct closed-loop autonomous driving evaluations on the LangAuto benchmark

Table 1. Performance comparison of our method with state-of-the-art approaches on the LangAuto-Tiny and LangAuto-Short benchmarks.

Method	LLM (#Params)	LangAuto-Tiny			LangAuto-Short		
		DS ↑	RC ↑	IS ↑	DS ↑	RC ↑	IS ↑
LMDrive [21]	LLaMA2 (7B) [27]	56.1	64.2	0.87	44.8	53.5	0.84
	Vicuna-v1.5 (7B) [39]	59.0	69.9	0.84	47.0	56.5	0.83
	LLaVA-v1.5 (7B) [13]	66.5	77.9	0.85	50.6	60.0	0.84
	TinyLLaMA (1.1B) [33]	64.1	75.0	0.86	46.2	59.7	0.79
AD-H [38]	Mipha (3B) [41] + OPT (350M) [35]	68.0	74.4	0.87	54.3	61.8	**0.86**
AdaDrive	TinyLLaMA (1.1B) [33] + Planner (3M)	**80.9**	**87.6**	**0.90**	**70.6**	**85.3**	0.81

Table 2. Performance comparison of our method with state-of-the-art approaches on the LangAuto benchmark.

Method	LLM (#Params)	LangAuto			Mem ↓ (G)	Inf. Time ↓ (ms)
		DS ↑	RC ↑	IS ↑		
LMDrive [21]	LLaMA2 (7B) [27]	32.8	40.1	0.81		
	Vicuna-v1.5 (7B) [39]	34.0	39.0	0.85	26.91	526
	LLaVA-v1.5 (7B) [13]	36.2	46.5	0.81		
	TinyLLaMA (1.1B) [33]	25.2	38.6	0.71	16.29	445
AD-H [38]	Mipha (3B) [41] + OPT (350M) [35]	41.1	48.5	**0.86**	-	-
AdaDrive	TinyLLaMA (1.1B) [33] + Planner (3M)	**42.9**	**53.4**	0.82	**6.79**	**189**

within the CARLA simulation environment, where the benchmark is structured into three distinct subtasks based on driving distances: LangAuto-Tiny, LangAuto-Short, and LangAuto. Route completion (RC), infraction score (IS), and driving score (DS) are three widely adopted evaluation metrics. Specifically, RC denotes the ratio of successfully traversed distance by an agent to the total planned route length. IS aggregates multiple categories of traffic violations through geometric progression, initializing at 1.0 and decaying multiplicatively with each infraction occurrence. DS synthesizes route completion and infraction penalties through multiplication, serving as the principal evaluation criterion and providing a comprehensive assessment of autonomous driving performance.

Model Configuration: Our framework employs a pre-trained visual encoder from [21], which remains frozen during training. For language modeling, we adopt TinyLLaMA [33], a lightweight language model, to reduce computational overhead and parameter count. The planner adopts a 4-layer Transformer [28] architecture. We adopt 20 learnable tokens and 20 memory tokens in LS-Qformer and set the capacity k of streaming memory buffer to 10.

Implementation Details: We employ an AdamW optimizer with a cosine learning rate scheduler. The initial learning rate is set to 1×10^{-5} with training spanning 15 epochs. In the loss function, we set the hyper-parameter margin d to 0.3 to constrain the LLM activation.

4.2. Main Results

Closed-loop Driving Performance: We conduct comprehensive experiments to evaluate our method on the LangAuto benchmarks [21], comparing against state-of-the-art approaches including LMDrive [21] and AD-H [38]. It is worth noting that AD-H employs additional mid-level language commands to train its hierarchical multi-agent driving system. The experimental results are presented in Tables 1 and 2. Our proposed AdaDrive demonstrates superior performance across all distance-based sub-tracks, particularly excelling in tiny and short route scenarios. Specifically, AdaDrive achieves driving scores of 80.9% and 70.6% on the LangAuto-Tiny and LangAuto-Short benchmarks, surpassing the second-best method AD-H by significant margins of 12.9% and 16.3%, respectively. These results validate the effectiveness of our self-adaptive slow-fast driving system.

Table 3. Ablation of Connector and LS-Qformer components on the LangAuto-Tiny benchmark.

ID	Connector		LS-Qformer	DS ↑	RC↑	IS↑
	W	H				
1	✗	✗	✗	67.4	75.3	0.86
2	✗	✗	✓	71.9	82.6	0.84
3	✓	✗	✓	77.9	84.8	0.89
4	✓	✓	✓	**80.9**	**87.6**	**0.90**

Table 4. Ablation of different feature modeling methods on the LangAuto-Tiny benchmark.

Method	#Token	DS ↑	RC↑	IS ↑
Q-former [11]	40	75.8	83.4	0.88
SeqQ-Former [14]	40	77.6	83.5	0.89
Q-former&Add	40	77.4	83.5	0.89
LS-Qformer (Ours)	20+20	**80.9**	**87.6**	**0.90**

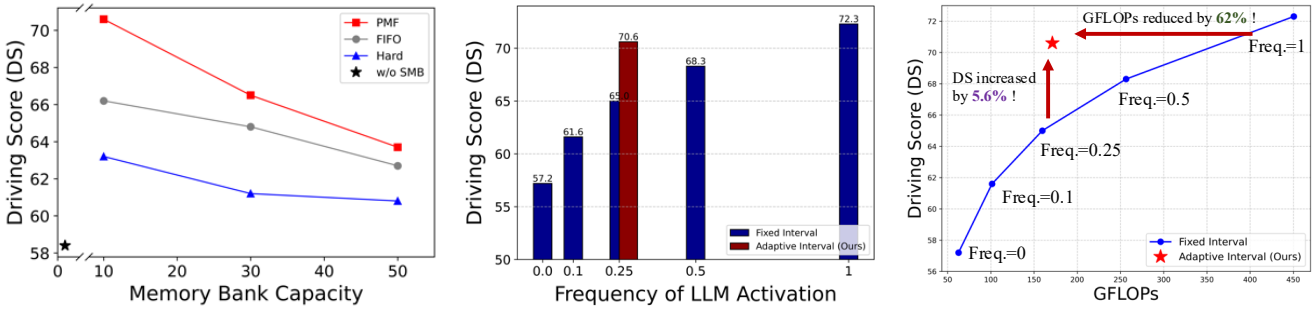

Figure 5. (a) Ablation on varying streaming memory buffer (SMB) capacities and content update mechanisms. (b) Comparison of our self-adaptive LLM activation *vs.* fixed-interval activation (freq. = 0, 0.1, 0.25, 0.5, and 1, where 0 indicates no activation and 1 indicates full activation) on driving scores. (c) Comparison of our self-adaptive LLM activation with fixed-interval LLM activation in terms of computational cost (GFLOPs) and driving scores. These analyses are performed using the LangAuto-Short benchmark.

Inference Time and Memory Cost: In addition to enhanced driving performance, our method exhibits substantial advantages in inference time and computational overhead, as showcased in Table 2. These benefits are attributed to two key architectural designs: 1) the self-adaptive slow-fast system. Unlike LMDrive and AD-H, which adopt sequential processing requiring LLM inference at every timestamp, our parallel architecture primarily relies on a lightweight planner, with the LLM activated only during emergencies as determined by the system's adaptive scheduling. Moreover, the planner only needs to process the current frame features, as the historical information has been propagated through the LS-Qformer. These architectural designs significantly reduce the system's inference latency. 2) The tailored streaming memory buffer. Existing methods lack specialized handling for streaming inputs, leading to data accumulation and increased memory overhead. In contrast, we explicitly propose a streaming memory buffer architecture that efficiently manages input data, reducing memory costs while improving inference speed.

4.3. Ablation Study

Components Effectiveness: We conduct comprehensive ablation studies to validate the effectiveness of the proposed LS-Qformer and quantify the performance gains achieved through connector-driven LLM interaction. First, we start from the baseline which implements a vanilla Q-former that independently aggregates frame-level features and a planner for trajectory prediction (ID #1). The results are presented in Table 3. Replacing the vanilla Q-former with our proposed LS-Qformer (ID #2) yields substantial performance improvements. These results demonstrate that the LS-Qformer effectively captures temporal dependencies in historical information, enabling more informed planning decisions. Furthermore, we integrate the LLM into our system through the Connector architecture. Leveraging the dynamic LLM activation mechanism governed by Connector-W (ID #3), our approach achieves significant performance gains, attaining a driving score of 77.9%. Moreover, by replacing the conventional full weighting LLM's feature fusion with our innovative Connector-H-controlled dynamic LLM contribution scaling strategy (ID #4), we observe further enhancement in the overall DS performance metrics.

Analysis of LS-Qformer: We compare our LS-Qformer against several architectural variants: 1) the standard Q-former which processes frames independently in a frame-wise manner; 2) SeqQ-Former [14], which propagates current output tokens as queries for subsequent frame feature extraction; and 3) Q-former with temporal accumulation, which incorporates historical context by additively fusing token representations from previous frames with current frame features. The results in Table 4 demonstrate that our LS-former achieves optimal driving scores by ingeniously incorporating long-range historical information with cur-

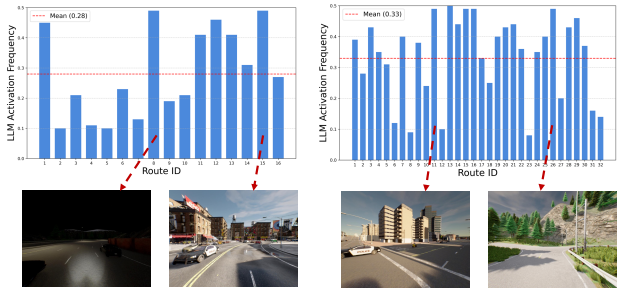

Figure 6. The distribution of LLM activation frequencies across all routes in both LangAuto-Short and LangAuto benchmarks.

rent frame content through a grouping mechanism.

Analysis of Streaming Memory Buffer: We investigate the impact of different memory bank capacities and content update mechanisms on trajectory prediction, as illustrated in Fig. 5(a). Several key observations emerge: 1) Thanks to our LS-Qformer's effective context aggregation, our method achieves comparable performance even when the LLM only attends to the current frame (w/o SMB). 2) Smaller memory bank capacities prove more beneficial for trajectory prediction. We hypothesize that as memory content increases, the LLM's instruction perception capability becomes diluted among the expanded context. 3) The hard update mechanism, which completely clears the current buffer upon reaching capacity limits, introduces inherent instabilities in subsequent trajectory predictions. In contrast, the PMF mechanism maintains temporal coherence while preserving more contextual information, leading to superior performance.

Analysis of Adaptive Collaboration: We compare our adaptive LLM activation strategy against fixed-interval activation at various frequencies. As illustrated in Fig 5(b), higher activation frequencies consistently yield more stable and robust driving performance. Our adaptive LLM activation mechanism enables dynamic responses to critical scenarios, achieving comparable performance to continuous LLM activation (frequency = 1.0) while maintaining an average activation frequency of only **0.28**. Fig 5(c) further demonstrates that our method strikes an optimal balance between driving performance and computational efficiency, reducing GFLOPs by 62% compared to continuous activation (frequency = 1.0) while improving driving scores by 5.6% relative to the fixed-interval scheme with a similar frequency (frequency = 0.25).

Besides, we analyze the distribution of LLM activation frequencies across all routes in both LangAuto-Short and LangAuto benchmarks, as illustrated in Fig. 6. The activation frequencies range from 0.1 to 0.5, demonstrating effective sparsity and dynamic adaptation, with average activation rates of 0.28 and 0.33 respectively. Notably, higher activation frequencies are observed in challenging routes, such as dense urban streets, nighttime conditions or mountain roads, validating our design principle of adaptive LLM engagement for complex situations. Furthermore, by analyzing the temporal distribution of LLM activations within individual routes, we identify patterns of increased LLM engagement during critical driving steps. As illustrated in Fig. 7, LLM activations are predominantly concentrated in complicated scenarios, including directional transitions, and intersection navigation. The LLM's advanced logical reasoning capabilities significantly enhance the autonomous vehicle agent's decision-making performance in these situations.

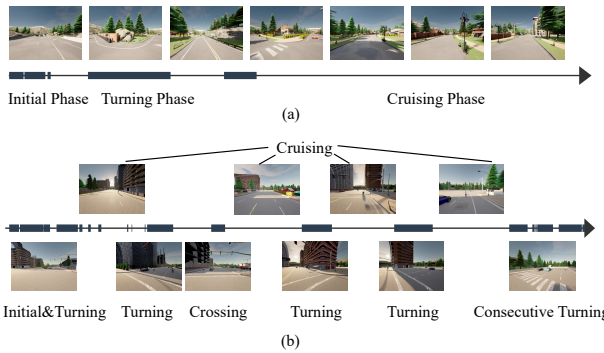

Figure 7. Temporal distribution of LLM activations for Route 7 (a) and Route 9 (b) in the LangAuto-Short benchmark. LLM activation moments, highlighted in darkblue on the timeline, demonstrate concentrated engagement during complex maneuvers such as turning and crossing, while remaining dormant during routine cruising phases.

5. Conclusion

This work explores LLM-powered language-grounded autonomous driving, focusing on two fundamental questions: optimal activation timing and effective utilization strategies of LLMs. Specifically, our approach features a self-adaptive slow-fast architecture that adaptively schedules LLM activation according to driving situations, while dynamically modulating its contribution weight based on prediction confidence scores. This strategy significantly enhances model flexibility and robustness while maintaining controlled computational overhead. Additionally, we introduce a tailored LS-Qformer for effective historical context aggregation and a streaming memory buffer with a propagative memory fusion strategy for efficient unbounded temporal data management. Extensive experiments demonstrate that our approach significantly outperforms existing methods in both effectiveness and efficiency, validating its potential for practical applications.

Acknowledgments

This work is supported in part by the National Key R&D Program of China (2024YFB3908503), in part by the National Natural Science Foundation of China (62322608), in part by the Shenzhen Longgang District Science and Technology Innovation Special Fund (No. LGKCYLWS2023018), in part by the Futian Healthcare Research Project (No.FTWS002), and in part by the Shenzhen Medical Research Fund (No. C2401036). This work is also sponsored by CIE-Tencent Robotics X Rhino-Bird Focused Research Program.

References

[1] Sergio Casas, Abbas Sadat, and Raquel Urtasun. Mp3: A unified model to map, perceive, predict and plan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14403–14412, 2021. 1

[2] Raphael Chekroun, Marin Toromanoff, Sascha Hornauer, and Fabien Moutarde. Gri: General reinforced imitation and its application to vision-based autonomous driving. *Robotics*, 12(5):127, 2023. 1, 2

[3] Joya Chen, Zhaoyang Lv, Shiwei Wu, Kevin Qinghong Lin, Chenan Song, Difei Gao, Jia-Wei Liu, Ziteng Gao, Dongxing Mao, and Mike Zheng Shou. Videollm-online: Online video large language model for streaming video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18407–18418, 2024. 3

[4] Yuan Chen, Zi-han Ding, Ziqin Wang, Yan Wang, Lijun Zhang, and Si Liu. Asynchronous large language model enhanced planner for autonomous driving. In *European Conference on Computer Vision*, pages 22–38. Springer, 2025. 2, 3

[5] Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9329–9338, 2019. 1, 2

[6] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016. 5

[7] Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, and Ziran Wang. Drive as you speak: Enabling human-like interaction with large language models in autonomous vehicles. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 902–909, 2024. 1

[8] Daocheng Fu, Xin Li, Licheng Wen, Min Dou, Pinlong Cai, Botian Shi, and Yu Qiao. Drive like a human: Rethinking autonomous driving with large language models. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 910–919, 2024. 1

[9] Jeffrey Hawke, Richard Shen, Corina Gurau, Siddharth Sharma, Daniele Reda, Nikolay Nikolov, Przemysław Mazur, Sean Micklethwaite, Nicolas Griffiths, Amar Shah, et al. Urban driving with conditional imitation learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 251–257. IEEE, 2020. 1, 2

[10] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17853–17862, 2023. 1, 2

[11] Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR, 2023. 7

[12] Xiaodan Liang, Tairui Wang, Luona Yang, and Eric Xing. Cirl: Controllable imitative reinforcement learning for vision-based self-driving. In *Proceedings of the European conference on computer vision (ECCV)*, pages 584–599, 2018. 1, 2

[13] Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *arXiv preprint arXiv:2310.03744*, 2023. 6

[14] Fan Ma, Xiaojie Jin, Heng Wang, Yuchen Xian, Jiashi Feng, and Yi Yang. Vista-llama: Reliable video narrator via equal distance to visual tokens. *arXiv preprint arXiv:2312.08870*, 2023. 7

[15] Jiageng Mao, Yuxi Qian, Hang Zhao, and Yue Wang. Gpt-driver: Learning to drive with gpt. *arXiv preprint arXiv:2310.01415*, 2023. 1

[16] Jiageng Mao, Junjie Ye, Yuxi Qian, Marco Pavone, and Yue Wang. A language agent for autonomous driving. *arXiv preprint arXiv:2311.10813*, 2023.

[17] Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving. *arXiv preprint arXiv:2312.03661*, 2023. 1

[18] OpenAI. Hello gpt-4o, 2024. 3

[19] Rui Qian, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Shuangrui Ding, Dahua Lin, and Jiaqi Wang. Streaming long video understanding with large language models. *arXiv preprint arXiv:2405.16009*, 2024. 3

[20] Abbas Sadat, Sergio Casas, Mengye Ren, Xinyu Wu, Pranaab Dhawan, and Raquel Urtasun. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIII 16*, pages 414–430. Springer, 2020. 1

[21] Hao Shao, Yuxuan Hu, Letian Wang, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. *arXiv preprint arXiv:2312.07488*, 2023. 1, 2, 4, 5, 6

[22] Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Conference on Robot Learning*, pages 726–737. PMLR, 2023. 1, 2

[23] Hao Shao, Letian Wang, Ruobing Chen, Steven L Waslander, Hongsheng Li, and Yu Liu. Reasonnet: End-to-end driving with temporal and global reasoning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13723–13733, 2023. 2

[24] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. *arXiv preprint arXiv:2312.14150*, 2023. 1, 2

[25] Xiaoyu Tian, Junru Gu, Bailin Li, Yicheng Liu, Chenxu Hu, Yang Wang, Kun Zhan, Peng Jia, Xianpeng Lang, and Hang Zhao. Drivevlm: The convergence of autonomous driving and large vision-language models. *arXiv preprint arXiv:2402.12289*, 2024. 2, 3

[26] Marin Toromanoff, Emilie Wirbel, and Fabien Moutarde. End-to-end model-free reinforcement learning for urban driving using implicit affordances. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7153–7162, 2020. 1, 2

[27] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023. 6

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 6

[29] Wenhai Wang, Jiangwei Xie, ChuanYang Hu, Haoming Zou, Jianan Fan, Wenwen Tong, Yang Wen, Silei Wu, Hanming Deng, Zhiqi Li, et al. Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving. *arXiv preprint arXiv:2312.09245*, 2023. 2, 3

[30] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kenneth KY Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *arXiv preprint arXiv:2310.01412*, 2023. 1, 2

[31] Haoji Zhang, Yiqin Wang, Yansong Tang, Yong Liu, Jiashi Feng, Jifeng Dai, and Xiaojie Jin. Flash-vstream: Memory-based real-time understanding for long video streams. *arXiv preprint arXiv:2406.08085*, 2024. 3

[32] Jiakai Zhang and Kyunghyun Cho. Query-efficient imitation learning for end-to-end simulated driving. In *Proceedings of the AAAI conference on artificial intelligence*, 2017. 1, 2

[33] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv:2401.02385*, 2024. 6

[34] Ruifei Zhang, Xiangru Lin, Wei Zhang, Jincheng Lu, Xuekuan Wang, Xiao Tan, Yingying Li, Errui Ding, Jingdong Wang, and Guanbin Li. Interactive 3d object detection with prompts. In *European Conference on Computer Vision*, pages 140–157. Springer, 2024. 1

[35] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. Opt: Open pre-trained transformer language models, 2022. *URL https://arxiv.org/abs/2205.01068*, 3:19–0, 2023. 6

[36] Wei Zhang, Jiaming Li, Meng Xia, Xu Gao, Xiao Tan, Yifeng Shi, Zhenhua Huang, and Guanbin Li. Offsetnet: Towards efficient multiple object tracking, detection, and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 1

[37] Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. End-to-end urban driving by imitating a reinforcement learning coach. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 15222–15232, 2021. 1, 2

[38] Zaibin Zhang, Shiyu Tang, Yuanhang Zhang, Talas Fu, Yifan Wang, Yang Liu, Dong Wang, Jing Shao, Lijun Wang, and Huchuan Lu. Ad-h: Autonomous driving with hierarchical agents. *arXiv preprint arXiv:2406.03474*, 2024. 1, 6

[39] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena, 2023. 6

[40] Xingyi Zhou, Anurag Arnab, Shyamal Buch, Shen Yan, Austin Myers, Xuehan Xiong, Arsha Nagrani, and Cordelia Schmid. Streaming dense video captioning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18243–18252, 2024. 3

[41] Minjie Zhu, Yichen Zhu, Xin Liu, Ning Liu, Zhiyuan Xu, Chaomin Shen, Yaxin Peng, Zhicai Ou, Feifei Feng, and Jian Tang. A comprehensive overhaul of multimodal assistant with small language models. *arXiv preprint arXiv:2403.06199*, 2024. 6