

# Enhanced Motion Forecasting with Plug-and-Play Multimodal Large Language Models

Katie Luo<sup>†</sup> Jingwei Ji<sup>\*‡</sup> Tong He<sup>‡</sup> Runsheng Xu<sup>‡</sup> Yichen Xie<sup>§</sup>  
Dragomir Anguelov<sup>‡</sup> Mingxing Tan<sup>‡</sup>

**Abstract**— Current autonomous driving systems rely on specialized models for perceiving and predicting motion, which demonstrate reliable performance in standard conditions. However, generalizing cost-effectively to diverse real-world scenarios remains a significant challenge. To address this, we propose Plug-and-Forecast (PnF), a plug-and-play approach that augments existing motion forecasting models with multimodal large language models (MLLMs). PnF builds on the insight that natural language provides a more effective way to describe and handle complex scenarios, enabling quick adaptation to targeted behaviors. We design prompts to extract structured scene understanding from MLLMs and distill this information into learnable embeddings to augment existing behavior prediction models. Our method leverages the zero-shot reasoning capabilities of MLLMs to achieve significant improvements in motion prediction performance, while requiring no fine-tuning—making it practical to adopt. We validate our approach on two state-of-the-art motion forecasting models using the Waymo Open Motion Dataset and the nuScenes Dataset, demonstrating consistent performance improvements across both benchmarks.

## I. INTRODUCTION

Since the first DARPA challenge, autonomous driving has progressed significantly, evolving into robo-taxis and driver assistance systems for mass consumer use [1], [2]. Today, many autonomous driving systems are built as modular frameworks, where each module is responsible for a particular driving function and is trained in a supervised manner.

The decomposition into subtasks significantly simplifies the collection of supervised training signals for each component, with the design of the individual modules inspired by how humans drive. They commonly include: perception of the scene and objects of interest [3], [4], motion forecasting/prediction of road agents surrounding the autonomous vehicle [5], [6], mapping the lanes in which the vehicle is on [7], [8], and planning a path for the autonomous vehicle, given all of the above information [9], [10]. Such systems have a lot of merit, proving fast, interpretable, and naturally yielding measurable performance at each step [10], [11]. Building upon these successes, the modular approach has enabled the progress and *democratization* of autonomous driving systems today into commercial robo-taxis in major cities and sophisticated driver assistance features enhancing everyday vehicle safety.

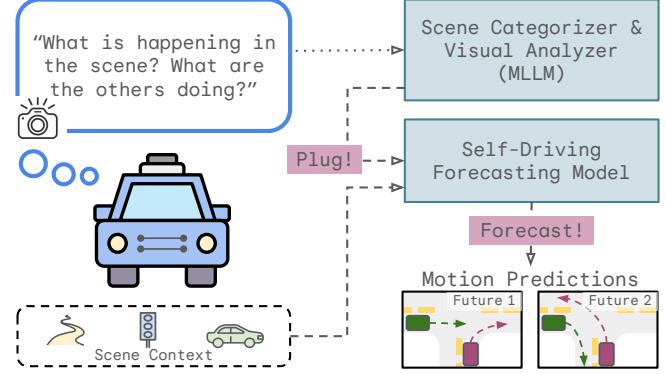


Fig. 1: **Plug-and-Forecast design.** We enhance traditional autonomous driving systems by integrating MLLMs into motion forecasting models that previously relied solely on scene context from perception modules, providing comprehensive scene understanding in a zero-shot manner.

Indeed, the frontier has now moved toward making these systems safer and more adaptable to the diverse challenges of real-world driving environments. Since autonomous vehicles operate in highly diverse real-world settings with a heterogeneous user base, they inevitably encounter cases absent from their training data. This distribution shift poses a fundamental challenge: ensuring robust performance across the long-tail of rare driving situations. One solution is to continuously collect large volumes of data and labels to expand coverage across different situations in order to continually retrain and refine the system. While there has been an increasing number of released self-driving datasets and challenges [12], [13], [1], this approach may be impractical at massive scale, as the costs of data collection and model development could be prohibitive. This raises the question: *Can we generalize the autonomous driving system to diverse, real-world scenarios in a principled and cost-efficient way?*

One avenue of generalizability arises from the recent progress in multimodal large language models (MLLM) [14], [15]. MLLMs have shown strong generalist capabilities that can capture reasoning behavior, adapting to new scenarios not explicitly represented in their training data [16]. Thus, these MLLMs have the promise to augment modular driving systems with reasoning beyond their driving-specific data even in a zero-shot manner. In this work, we explore a method to augment an existing driving system with an MLLM, showing that information prompted from it can further improve the motion prediction driving task performance (Fig. 1).

Our key insight is that language is often a better descriptor

\* Corresponding author.

† Computer and Information Sciences Department, Cornell University, {kz16}@cornell.edu. Work was done at Waymo.

‡ Waymo LLC, {jingweij, simpleig, runshengxu, dragomir, tanmingxing}@waymo.com

§ UC Berkeley, {yichen\_xie}@berkeley.edu. Work was done at Waymo.

for specific challenging cases and provides a powerful handle—beyond engineering features—to deal with targeted behaviors quickly. For example, edge cases such as the presence of emergency vehicles may be sparse within training data, and typical modular driving systems may produce less reliable predictions; however, a prompt can be quickly specified to explicitly target such scenarios (Fig. 6). This collection of targeted prompts can be queried into the MLLM for answers about the scene that goes *beyond* the limitations of its training data, and in turn, can be distilled into learnable embeddings consumable by modular driving systems. Specifically, we design prompts to extract information from the scene via the MLLM into a structured text format. This structured text is then parsed into a set of pre-defined answers, which are queried into a trained embedding space for the corresponding features. These MLLM-extracted features are then provided as an additional input to the prediction model, augmenting it with visual reasoning information relevant to the scene.

In this work, we explore the use of both scene level information as well as road-agent specific information from these MLLMs. We validate our approach using both types of MLLM queried features on two state-of-the-art motion behavior prediction models to demonstrate the effectiveness of such a plug-and-play approach. Our method, Plug-and-Forecast (PnF), consistently and significantly boosts behavior prediction performance while requiring only MLLM zero-shot inference access (*i.e.* no architecture and checkpoint access or fine-tuning needed), thereby preserving MLLMs’ generalist capabilities and their ability to handle long-tailed cases [17]. Specifically, our contributions are as follows:

- We analyze an additional source of information, language understanding, to augment our motion prediction models.
- Propose a plug-and-play MLLM augmented method for the autonomous vehicle motion prediction task.
- Empirically demonstrate a significant and consistent improvement of performance on motion prediction task.

## II. RELATED WORKS

**Motion forecasting.** Among the various tasks in autonomous driving, motion forecasting/prediction is a crucial area of research, as it models the behavior of both the autonomous vehicle and other agents on the road. Early approaches to motion predictions [18], [19], [20], [21] involve rasterizing scenes into 2D images, which are then processed by convolutional neural networks (CNNs). As the field advances, research moves forward representing road elements leveraging sparsity – such as bounding boxes, road graphs, and traffic lights – as graph nodes, which can then be processed using graph neural networks (GNNs) [22], [23] or recurrent neural networks (RNNs) [24], [25]. Some methods adopt causal models with autoregressive trajectory prediction [26], [6], allowing for sequential updates that refine predictions based on prior outcomes. In this work, we integrate MLLMs with two latest motion prediction models, Wayformer [27] and MotionLM [6], merging the strengths of well-established motion prediction frameworks with the broad capabilities of a knowledgeable generalist model.

**Multimodal Large Language Models in Autonomous Driving.** The field of Multimodal Large Language Models (MLLMs) [14], [28], [15] has rapidly progressed, addressing the need for integrated understanding across diverse data modalities, including text, vision, and audio. MLLMs show promising integration potential in autonomous driving systems. Several approaches enhance decision-making: DriveGPT4 [29] uses iterative Q&A for explaining actions and predicting controls from structures data as text prompts to GPT-4 [28]; DriveLM [30] and DriveVLM [31] apply MLLMs to graph-based VQA and chain-of-thought reasoning respectively. Powered by Gemini [15], Waymo’s EMMA model [32] directly maps raw camera sensor data into various driving-specific outputs, including planner trajectories, perception objects, and road graph elements, in a unified language space. Unlike prior work, we demonstrate that state-of-the-art motion forecasting performance can benefit from MLLMs in a zero-shot learning manner, without fine-tuning MLLMs with additional data or human labels.

## III. PROBLEM SETUP

In this work, we focus on the task of motion prediction. Let  $\mathbf{s}_t$  be the states of all agents in a scenario at time  $t$ . Let  $\mathbf{m} = (\mathbf{r}, \tau_{1:T})$  represent the map-traffic elements, consisting of static elements (*e.g.* lane geometries, road boundaries, crosswalks)  $\mathbf{r}$  and dynamic elements (*e.g.* traffic lights, temporary road works)  $\tau_t$ . In addition, we assume access to passively collected ego vehicle sensor states  $\xi_t$ , *e.g.* camera measurements. We wish to predict all *future* agent states  $\mathbf{s}_{t_0+1:T}$  for the current time  $t_0$  given past agent states  $\mathbf{s}_{1:t_0}$ , map  $\mathbf{m}$ , and sensor measurements  $\xi_{1:t_0}$ :

**Agent States** The state of each agent  $i$  at time  $t$  is represented as  $s_t^i \in \mathbb{R}^d$ , encoding position, velocity, heading, and agent type. The joint state  $\mathbf{s}_t = \{s_t^1, \dots, s_t^N\}$  consists of the states of all  $N$  agents in the scene. In a modular driving system, the past and current agent states are usually provided by upstream perception modules.

**Map Road Graph States** The map-traffic representation  $\mathbf{m}$  consists of static road elements  $\mathbf{r}$  encoding permanent infrastructure and dynamic traffic elements  $\tau_t$  that update with traffic conditions. Road segments  $\mathbf{r}$  is represented as a collection of road-element poly-segments.  $\tau_t$  is the traffic light states at timestep  $t$ .

**Sensor States** The collected sensor measurements  $\xi_t$  provide additional scene context through ego vehicle observations, particularly from camera data. These measurements are used in complement with the past agent states to extract context-dependent visual prompts for the analysis module in PnF.

## IV. PLUG-AND-FORECAST

Our goal is to design a system that can leverage the reasoning and analysis capabilities of an MLLM for the modular AV stack, particularly for the task of motion prediction. Our method, Plug-and-Forecast (PnF) consists of three main components: A language-based Visual Semantic Analyzer (Section IV-A) to extract fine-grain semantic features

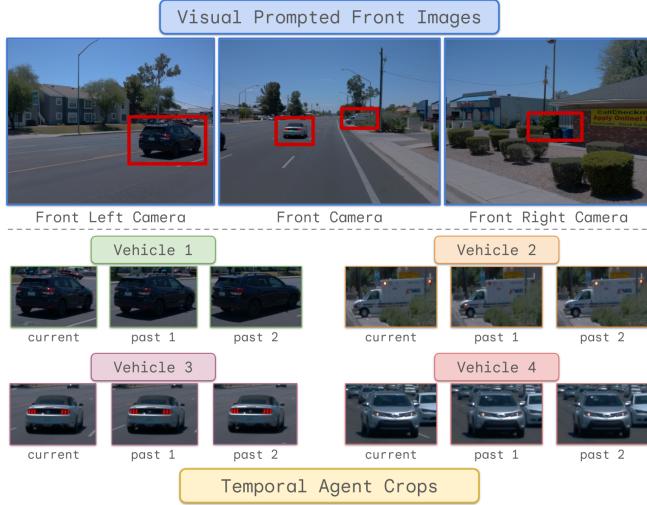


Fig. 2: Visual Prompting Example.

such as agent behaviors, intentions, or visual variants. A driving Scene Categorizer (Section IV-B) to parse out scene-level driving characteristics such as road type or weather. And a Transformer-based architecture augmentation (Section IV-C) to incorporate the text into a modular AV stack. Our language-based components take advantage of large, pre-trained MLLMs and do not require fine-tuning, thus making them applicable for *any* modular AV task. For this work, we focus on the motion prediction task.

#### A. Visual Semantic Analyzer

The Visual Semantic Analyzer (VSA) aims to extract agent-specific semantics useful for the downstream task of motion prediction. It takes as inputs the sensor inputs  $\xi_{0:t_0}$ —the rolling window of past information—and an automatically generated text prompt. It outputs a structured text representation  $\mathbf{x}_{1:N}$  of all  $N$  agents in the scene, together. Below, we describe the details of the component, and explain how  $\mathbf{x}_{1:N}$  is produced using MLLMs.

**Multimodal Prompting.** Recent progress in MLLM has extended the advances in language models into domains of images and cross-modal reasoning [33], [15]. Our work leverages the power of MLLMs in a zero-shot fashion, and does not require finetuning. Specifically, the VSA takes a multimodal prompt, consisting of images of the current time and focused agent-following temporal crops extracted from  $\xi_{0:t_0}$ . We annotate the camera images from the current time using red bounding boxes around the agents-of-interest as a visual prompt. This has shown to improve their focus onto localized regions, which they otherwise may struggle to do [34]. In addition, the VSA component is provided with additional visual prompts consisting of focused crops of these agents, as well as 2 past crops for additional, temporal reasoning. Because we have access to agent states from the past  $\mathbf{s}_{1:t_0}$ , we can quickly and cheaply construct a rolling memory of the camera context from the past for each agent. To reduce noisy or incomplete information getting to the MLLM, we apply filtering for object crops that are occluded or too far away. Such information is provided in addition to

the text-based prompting; an example of the visual prompt engineering can be seen in Fig. 2.

**Language Analysis of Agent Semantics.** We wish to prompt the MLLM to obtain agent semantics useful for motion prediction. Thus, the full multimodal prompt is a mixture of both image-based visual prompting, as well as semantics specific text prompting. To capture the variations of semantics that may be important across different agent class types (*e.g.* between vehicle and pedestrians), we design class-specific, focused prompts. By leveraging the class type from the agent states,  $\mathbf{s}_{t_0}$ , we are able to construct a type dependent query, with the corresponding image prompts, for the scene. In this work, we consider the vehicle and pedestrian class: For the vehicle class, we query for the presence of emergency vehicles, vehicle type, signals, and potential action intention in the next 3 seconds (keep forward, slow down, turn, stop, or parked). For the pedestrian class, we query for the presence of a micromobility and action intention in the next 3 seconds (jaywalk, continue on sidewalk, cross, turn, or stop/wait). In both classes, we also query for if the agent is heavily occluded. To further improve the quality of the generation, we leverage generation examples and Chain-of-Thought prompting [35]. The final output consists of per-agent chain-of-thought explanation, followed by a markdown formatted table of final answers, with an agent’s feature per row. See Fig. 3 for an example of the VSA component’s input and output for the vehicle class. The full prompt and complete examples are provided in the supplementary. While we experiment with a defined set of semantic tasks for MLLMs to answer, the promising capabilities of generalist models allow for easy expansion of the question set within the same proposed framework.

**Leveraging Agent-Level Structured Text.** The generated text, while prompted to be structured for each agent and to answer specific questions, are not immediately consumable by a modular AV model. To bridge this gap, we begin by extracting the relevant row for each agent  $i$ , which is parsed to get an answer per question to get a list of text-based agent semantics. We further introduce a smaller set of “vocabulary” that are pre-specified answers to the questions; in this way, we can discretize the outputs without relying on the underlying model embeddings of the generations. Because the generation can be incomplete or prone to minor issues, we include a default answer for all values that do not fit in the pre-defined structure. The final output is a multi-hot feature  $\mathbf{x}_i$ , corresponding to the answer set, for each agent  $i$ , that captures individual visual features and behavior intentions that is ready to be consumed by downstream models. There are major benefits to this design: structured outputs can be handled very easily by LLMs/MLLMs, these outputs are easy to interpret, and models can directly consume the final results as input features with minimal processing.

#### B. Driving Scene Categorizer

In order to capture scene level understanding that may be present beyond individual agent-level, we add in a Scene

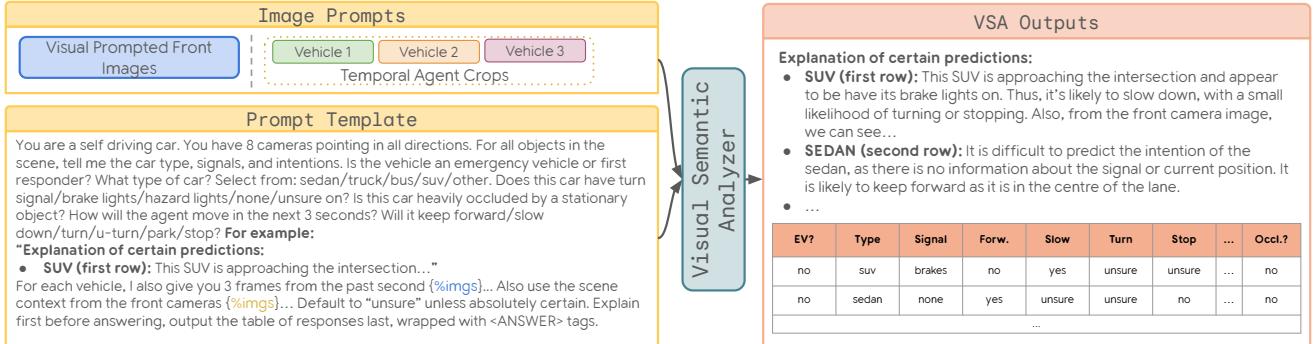


Fig. 3: Example Visual Semantic Analyzer (VSA) Input and Outputs.

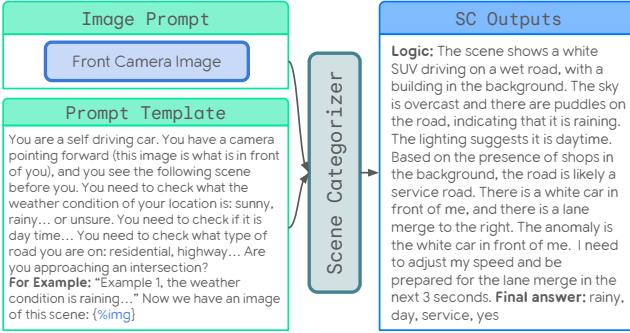


Fig. 4: Scene Categorizer (SC) Input and Output.

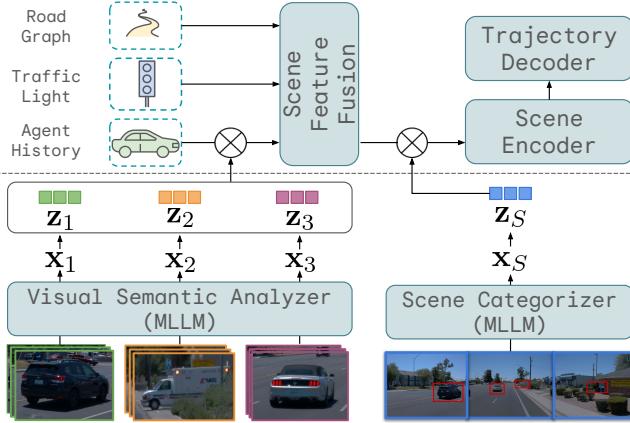


Fig. 5: Architecture of PnF method. We augment a Transformer-based modular AV stack, allowing us to learn to incorporate text inputs into the stack for motion prediction.  $\otimes$  stands for learned information gain operation.

Categorizer (SC) component that categorizes scene-level information relevant for agent behavior. The SC component not only provides scene-level context, but also helps with motion prediction in cases where agents are heavily occluded and may not provide good visual cues. Specifically, given camera inputs of the current time,  $\xi_{t_0}$ , it outputs another structured text representation  $x_S$ , capturing holistic scene context for driving behavior.

**MLLM for Holistic Scene Understanding.** To extract such information, we leverage the MLLM along with  $\xi_{t_0}$  and query for the current scene information important to driving, *i.e.* weather condition of the location, time of day, road type (*e.g.* residential or highway), and if the ego-vehicle is approaching an intersection. We query the MLLM using a scene-level

prompt, and leverage chain-of-thought reasoning to improve efficacy of the generation. The final output consists of a high-level scene reasoning, followed by the answers for each question (Fig. 4). The full SC prompt is provided in the supplementary.

**Scene-Level Structured Text Representation.** Similar to the outputs of the VSA component, downstream modular stacks are unable to consume the text input directly. Thus, we map the final answers to the scene-level question set onto a pre-defined answer-set ‘vocabulary’. Similarly, we define a default ‘unsure’ value for capturing text that falls outside of this vocabulary. From this, we obtain for each scene a multi-hot vocabulary vector  $x_s$ , that captures the scene-level categorizations necessary for driving behavior prediction.

### C. Plugging MLLMs into Modular AV Stacks

To leverage the rich contextual information from agent-level analysis and scene categorizations from the MLLM components into the modular AV stacks, we propose an augmentation to existing Transformer-based motion prediction architectures, which are currently the predominant paradigm [27], [36], [37]. Our approach processes structured text inputs—agent-specific descriptions  $x_{1:N}$  from the VSA component and scene-level descriptions  $x_S$  from the SC component—by embedding them into a learned representation space. PnF then incorporates them into the Transformer prediction model via a learned information gain, applied at both the agent-feature level and to the whole scene-feature level, respectively. Finally, it uses the augmented features to predict future agent states,  $\hat{s}_{t_0+1:T}$ . We visualize PnF’s augmented architecture in Fig. 5.

**Structured Text Embedding.** Our PnF framework augments a modular AV motion prediction model via a learned incorporation of the text information from the prior components. To facilitate this, we base our modular AV model’s encoder on the architecture first introduced by prior work [27], which has a similar Transformer base architecture that is used by many current state-of-the-art prediction models, including [38], [6]. Different from them, we learn a structured language embedding that is added onto the agent-features and subsequently scene-features, similar to how a learned positional embedding is added to improve performance. Specifically, given  $x_{1:N}$  and  $x_S$  outputs from the VSA and SC components, we learn  $z_{1:N}$  and  $z_S$  embedding mappings

**Explanation of certain predictions:**

- Pedestrian 1:** The pedestrian is running in the middle of a road, and there is no crosswalk nearby, indicating jaywalking. The pedestrian is not on a micromobility device. Despite being in the road, the pedestrian is likely going to continue running in the same direction and not change course in the next 3 seconds.
- Object 1:** An ambulance with hazard lights turned on. It is parked and unlikely to move in the next 3 seconds.

**Explanation of certain predictions:**

- Pedestrian on Scooter (first row):** The pedestrian is on an electric scooter, and is legally crossing at an intersection. The road markings appear to be a crosswalk for the pedestrian. We should drive carefully and yield. Because the crossing is at a crosswalk, it is legal, and therefore not jaywalking. Finally, because it is at the crosswalk, they'll likely cross in the next 3 seconds.

Fig. 6: **Output samples of the VSA component.** We visualize the image of the actor corresponding to the generation. Observe that the MLLM can reason about extreme cases.

into the embedding space used in the prediction model:

$$\forall i, \mathbf{z}_i = \text{emb}_a(\mathbf{x}_i) \in \mathbb{R}^{d_a} \quad (1)$$

$$\mathbf{z}_S = \text{emb}_S(\mathbf{x}_S) \in \mathbb{R}^{d_S}, \quad (2)$$

where  $d_a$  is the feature dimension of the agent features and  $d_S$  is the dimension of the encoded scene features. We use a learned linear embedding layer  $\text{emb}_a$  and  $\text{emb}_S$  for the VSA and SC structured text representation, respectively. This allows the direct incorporation of the PnF outputs from MLLM to be used by the prediction model.

**Learned Information Gain.** In order to allow the model to selectively incorporate the sparse VSA outputs, we learn to predict an information gain —a scalar information bottleneck clamped between  $(-1, 1)$ — that is multiplied to the embedding before added into the prediction model features. Specifically, given structured text embeddings  $\mathbf{z}_{1:N}$  and  $\mathbf{z}_S$ , we train a small MLP  $f_{\theta_a}$  to predict the scalar gains  $\alpha_{1:N}$  for each of the  $N$  agents, conditioned on VSA semantics, and augment the individual agent features:

$$\forall i, \alpha_i = \tanh(f_{\theta_a}(\mathbf{z}_i)) \quad (3)$$

$$\mathbf{f}'_i = \mathbf{f}_i + \alpha_i \cdot \mathbf{z}_i \quad (4)$$

where  $\mathbf{f}_i$  is agent  $i$ 's feature and  $\mathbf{f}'_i$  is the input to the encoder. These agent features are then early-fused into the other scene features via the Transformer encoder [27]. Similarly, we train another MLP  $f_{\theta_S}$  to predict the scene-level gain  $\alpha_S$  and augment the aggregated scene feature:

$$\alpha_S = \tanh(f_{\theta_S}(\mathbf{z}_S)) \quad (5)$$

$$\mathbf{f}'_S = \mathbf{f}_S + \alpha_S \cdot \mathbf{z}_S \quad (6)$$

where  $\mathbf{f}_S$  is the aggregated scene feature and  $\mathbf{f}'_S$  is the input to the decoder. Recall that the scene-level feature is the output of the encoder, thus contains information from the agent past states as well as map-traffic information  $\mathbf{m}$ .

This design allows the model to selectively incorporate the agent-level information since the per-agent VSA features

may not be always present. Note that when feeding 0-valued embeddings  $\mathbf{z}$ , the scalar gain  $\alpha$  will be 0, too. Additionally, this allows the model to learn a sparsification that regularizes the occasional additional features. As a bonus, the learned gain also helps to handle noisy outputs from the VSA and SG modules, thereby reducing the noise from faulty MLLM generations or hallucinations.

**Motion Prediction Training.** To train the lightweight embedding layers, we use end-to-end training on the prediction task as supervision. From the features of the augmented prediction model encoder, the modular AV model now has an encoded representation that has context both from the traditional perception task  $(\mathbf{s}_{1:t_0}, \mathbf{m})$  as well as context from the VSA ( $\mathbf{x}_{1:N}$ ) and SC ( $\mathbf{x}_S$ ) reasoning components. We leverage the query-based decoder head from [27], which outputs a time-series extension to a mixture-of-Gaussians, as well as a “valid” mask to handle variable number of predictions. Finally, the whole model, including the transformer-based prediction model and embedding layers for adapting MLLM outputs, is trained end-to-end on the classification and regression loss from [39], [27] to the poses in the ground truth observed future predictions  $\mathbf{s}_{t_0+1:T}$ . From this, we construct future state per agent  $\hat{\mathbf{s}}_{t_0+1:T}$  as our final prediction via trajectory aggregation on the output GMM following [39], [40]. Note that such a design for the decoder head is simply due to the performance and ease of training, and indeed any prediction head can be swapped in without loss of generality, such as autoregressive predictors [6], [41].

## V. EXPERIMENTAL RESULTS

**Datasets.** We evaluate our method on two large-scale autonomous driving datasets: the Waymo Open Motion Dataset (WOMD) [43] and the nuScenes Dataset [13]. The Waymo Open Motion Dataset contains over 570 hours of real-world driving scenarios across diverse locations and conditions, providing rich multi-agent trajectory data sampled at 10Hz. Each scene spans approximately 9 seconds, where we use 1 second of history and predict 8 seconds for future state prediction. The dataset includes detailed semantic map information and agent interactions across urban and suburban environments. We additionally associate the frames with their corresponding camera images to obtain inputs for the MLLM. The nuScenes prediction dataset comprises almost 1000 driving scenes of 20 seconds each, consisting of images and trajectory data sampled at 2Hz, collected from Boston and Singapore. We follow the standard experiment protocol of using 2 seconds of history to predict 6 seconds of future trajectories. Both datasets provide HD maps with semantic elements like lanes and crosswalks, while WOMD additionally provides traffic signals.

**Evaluation.** We adopt the standard metrics used in trajectory forecasting literature. For the WOMD, we report the minimum average displacement error (minADE), minimum final displacement error (minFDE), and miss rate (MR), averaged at timesteps of 3, 5, and 8 seconds, following the Waymo Open Motion Dataset [43] leaderboard convention. On WOMD,

Method	Reference	minADE ( $\downarrow$ )	minFDE ( $\downarrow$ )	Miss Rate ( $\downarrow$ )	Overlap ( $\downarrow$ )	mAP ( $\uparrow$ )	soft-mAP ( $\uparrow$ )
MultiPath++ [39]	ICRA 2022	0.978	2.305	0.440	-	-	-
MTR [37]	NeurIPS 2022	0.605	1.225	0.137	-	0.416	-
Wayformer [27]	ICRA 2023	0.551	1.160	0.121	-	0.410	0.425
MoST + Cam. [36]	CVPR 2024	0.539	1.110	0.117	-	0.420	0.440
MotionLM <sup>†</sup> [6]	<i>Reproduced</i>	0.574	1.189	0.139	0.129	0.382	0.403
+ PnF	Ours	<b>0.565</b>	<b>1.166</b>	<b>0.132</b>	<b>0.129</b>	<b>0.390</b>	<b>0.413</b>
Wayformer <sup>†</sup>	<i>Reproduced</i>	0.539	1.111	0.119	0.128	0.425	0.446
+ PnF	Ours	<b>0.528</b>	<b>1.084</b>	<b>0.113</b>	<b>0.127</b>	<b>0.437</b>	<b>0.457</b>

TABLE I: WOMD validation set performance.  $\dagger$  marks our reproduced baselines. We add PnF on top for fair comparisons.

Method	MLLM	$K = 1$			$K = 5$		
		minADE ( $\downarrow$ )	minFDE ( $\downarrow$ )	Miss Rate ( $\downarrow$ )	minADE ( $\downarrow$ )	minFDE ( $\downarrow$ )	Miss Rate ( $\downarrow$ )
Wayformer [27]	-	2.471	6.587	0.830	1.137	2.701	0.542
	Qwen2-VL-7B [42]	2.465	6.571	0.828	1.117	2.647	<u>0.537</u>
	Gem. 1.5 Flash [15]	<b>2.444</b>	<u>6.551</u>	<b>0.825</b>	<u>1.102</u>	<u>2.591</u>	0.540
	Gem. 1.5 Pro [15]	<u>2.455</u>	<b>6.540</b>	0.826	<b>1.093</b>	<b>2.574</b>	<b>0.535</b>
MotionLM [6]	-	2.641	6.960	0.847	1.247	2.972	<u>0.518</u>
	Gem. 1.5 Flash [15]	<u>2.598</u>	<u>6.936</u>	<u>0.836</u>	<u>1.179</u>	<u>2.801</u>	0.526
	Gem. 1.5 Pro [15]	<b>2.508</b>	<b>6.695</b>	<b>0.830</b>	<b>1.132</b>	<b>2.706</b>	<b>0.504</b>

TABLE II: NuScenes validation set performance. We report the prediction metrics from nuScenes at  $K = 1$  and  $K = 5$ .

Method	minADE ( $\downarrow$ )	minFDE ( $\downarrow$ )	soft-mAP ( $\uparrow$ )
Wayformer [27]	0.819	1.737	0.305
+ PnF	0.753	1.589	0.330
$\Delta$	<b>-7.99%</b>	<b>-8.52%</b>	<b>8.37%</b>

TABLE III: WOMD hardest subset performance.

Signal	Intent	Type	Scene	minADE ( $\downarrow$ )	minFDE ( $\downarrow$ )	soft-mAP ( $\uparrow$ )
		-		0.539	1.111	0.446
✓	✓			0.533	1.094	0.453
	✓	✓		0.529	1.085	0.449
✓		✓		<b>0.528</b>	1.094	0.457
✓	✓	✓	✓	0.530	1.086	0.453
	✓	✓	✓	0.529	<b>1.084</b>	<b>0.457</b>

TABLE IV: Performance over different reasoning types.

we additionally report mean average precision (mAP) and soft mAP, which evaluate prediction confidence scores across different motion categories (*e.g.*, straight, turns). For the nuScenes dataset, we report the average displacement error (ADE), final displacement error (FDE), and miss rate (MR) at 2.0m threshold. Following standard practice, we report these metrics for the top  $K = 1$  and  $K = 5$  predictions, where  $K$  represents the number of trajectory modes.

**Implementation Details.** We leverage Gemini 1.5 [15] for its relatively cheap cost, high performance, and —most importantly— long context window. For all experiments going forward, assume that queries to MLLM made by the VSA and SC components are to Gemini 1.5 Flash, unless otherwise specified. Accesses are made via public endpoints. We leverage the encoder-decoder framework for the modular prediction model from both [27] and [6] to demonstrate the generalizability of our framework.

#### A. Prediction Performance

We demonstrate the results of our method, PnF, as applied to the motion prediction task on the WOMD in Tab. I. We benchmark our work against a series of motion forecasting

Gain Ablation	minADE ( $\downarrow$ )	minFDE ( $\downarrow$ )	soft-mAP ( $\uparrow$ )
None (added)	0.531	1.090	0.454
Constant	0.531	1.085	0.449
Learned (Ours)	<b>0.529</b>	<b>1.084</b>	<b>0.457</b>

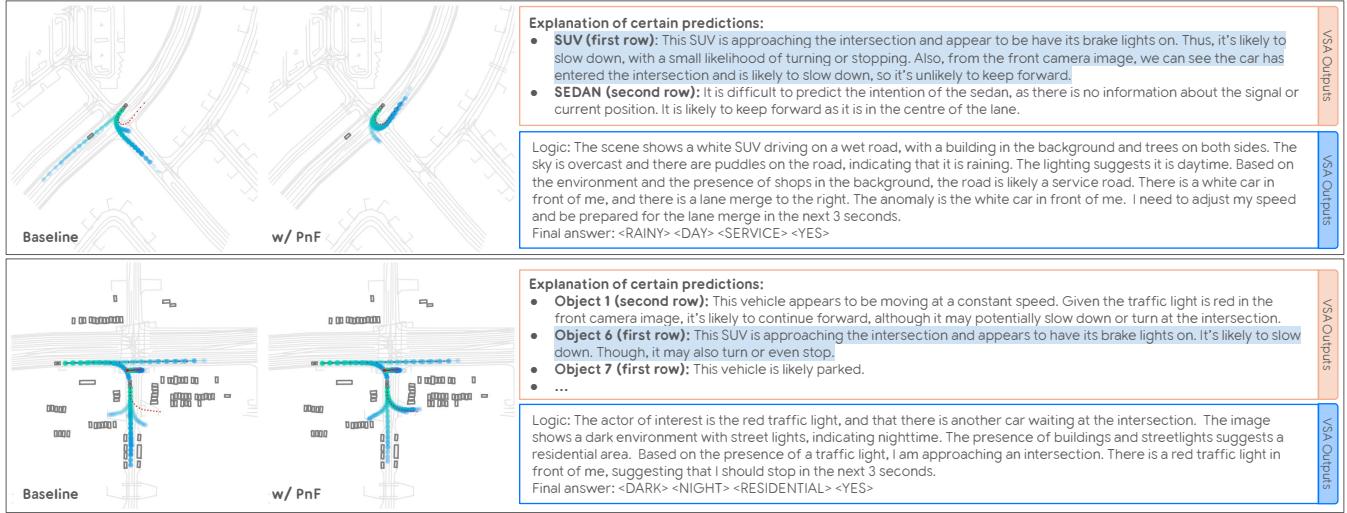
TABLE V: Ablation on gain values used.

works. Our method is applied to both the Wayformer model and the MotionLM model; under both settings, incorporating PnF consistently improves performance. Indeed, applying our method onto Wayformer beats all baselines, including MoST [36], which receives inputs from camera images but assumes accessibility of a powerful vision encoder. Our prediction performance results on the nuScenes prediction task are reported in Tab. II. Similarly, we observe performance gains over the baseline when PnF is applied. Because the nuScenes dataset is smaller, we are able to run our method using MLLM at three model sizes, Gemini 1.5 Flash and Pro [15] as well as an open source 7-billion parameter model Qwen2-VL-7B [42], to demonstrate the performance across MLLM capabilities. Observe that regardless of MLLM capacity, PnF improves performance, particularly at larger MLLM for larger sample counts, suggesting that the VSA and SC components are able to capture different modes of motion futures.

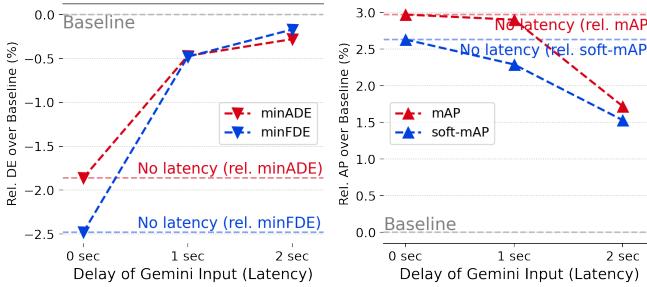
#### B. Method Analysis

**Qualitative Results.** We visualize some qualitative results of our method compared to the baseline in Fig. 7. For ease of reference, we additionally provide the natural language text outputs from the VSA and the SC components corresponding to the scene prediction. Observe that in the first example, the mode of the prediction is corrected and fixed, thanks to guidance from the analysis provided by PnF. In the second examples, the reasoning highlighted in text shows that the method is able to use the context provided to better calibrate the uncertainty between different possible action choices.

**Results on Hardest Split.** To better understand the gains



**Fig. 7: Qualitative Analysis.** We visualize predictions from the Wayformer model without and with PnF applied. Predictions are color-coded temporally: green indicates the near future, and blue the farthest. We include outputs from the VSA and SC components corresponding to the predictions (highlighted). Our predictions better model behavior by leveraging text logic.



**Fig. 8: Method performance with latency** values at current time (0 sec), 1 sec, and 2 sec of delay on the MLLM features. We show the relative displacement errors (lower is better) on the left, and relative average precision (higher is better) relative to the baseline in grey.

from PnF, we conduct an analysis on the top 10% hardest scenarios in Tab. III, and see that a majority of our gains (more than 3× as compared to evaluation over the whole set) is obtained there. The top 10% hardest set is defined as the scenarios in which the baseline method achieves the highest minADE. This aligns with our intuition that our method is able to improve on the tail-end cases, thanks to the MLLMs generalizability.

**Ablations.** To validate our design choices, we ablate our method on both the query design of the VSA and SC components in Tab. IV, and the architecture design choice of our information bottleneck in the input gain in Tab. V. Observe that with the inclusion of all the query information, the performance is the best. Removing type information reduces all metrics. Removing vehicle signal information drastically reduces the average precision for track prediction, suggesting that it is giving context into trajectory mode intentions. Including scene only information improves slightly over the baseline, but worse than including VSA features.

**Latency Analysis.** One key question that we aimed to answer is the effects on delayed outputs from the VSA and SC

components, since currently MLLMs are unable to run in real-time due to the autoregressive nature of the generation. While there have been ample works to mitigate this, such as [44], we analyze the effect of latency of structured text features on our method’s performance. To simulate this for the prediction task, we run an analysis on using delayed outputs from generations for PnF on the relative performance improvement over the baseline in Fig. 8. Specifically, we supply the modular AV stacks at time  $t$  with outputs from the VSA and SC at time  $t - k$  for  $k = \{0, 1, 2\}$  seconds. Observe that, while indeed the latency adversely affects the performance, we see positive gains even at 2 seconds of delay. Indeed, classification (*i.e.*, mAP) still performs well with little decline in performance with up to 1 second of delay, suggesting that our method is decently robust to latency.

## VI. DISCUSSION AND FUTURE WORKS

This work proposed PnF, a method that effectively augments autonomous driving systems through zero-shot MLLM inference and consistently improves performance across multiple state-of-the-art motion forecasting models. Our results demonstrate one of the potential uses of MLLMs to enhance modular autonomous driving systems. Several promising directions remain for future exploration. First, investigating real-time integration of MLLM outputs into driving systems could address latency concerns for deployment. Additionally, expanding the range of safety-critical scenarios and testing with a broader set of driving models would further validate the generalizability of our approach. As we look into the future of autonomous vehicle advancements, focus should turn to leveraging information across modalities to generalize into real-world environments.

## ACKNOWLEDGMENT

We thank the Perception Research team at Waymo for their insightful discussion and valuable feedback. Katie Luo is support in part by the AAUW Dissertation Fellowship.

## REFERENCES

- [1] Waymo, "Waymo one is now open to everyone in SF," June 2024.
- [2] Tesla, "Autopilot and Full Self-Driving (Supervised)," Tesla, 2024.
- [3] S. Shi, X. Wang, and H. Li, "Pointrenn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019.
- [4] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," *arXiv preprint arXiv:2203.17270*, 2022.
- [5] B. Ivanovic and M. Pavone, "The trajectron: Probabilistic multi-agent trajectory modeling with dynamic spatiotemporal graphs," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 2375–2384.
- [6] A. Seff, B. Cera, D. Chen, M. Ng, A. Zhou, N. Nayakanti, K. S. Refaat, R. Al-Rfou, and B. Sapp, "Motionlm: Multi-agent motion forecasting as language modeling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 8579–8590.
- [7] B. Liao, S. Chen, X. Wang, T. Cheng, Q. Zhang, W. Liu, and C. Huang, "Maptr: Structured modeling and learning for online vectorized hd map construction," *arXiv preprint arXiv:2208.14437*, 2022.
- [8] K. Z. Luo, X. Weng, Y. Wang, S. Wu, J. Li, K. Q. Weinberger, Y. Wang, and M. Pavone, "Augmenting lane perception and topology understanding with standard definition navigation maps," *arXiv preprint arXiv:2311.04079*, 2023.
- [9] S. Casas, A. Sadat, and R. Urtasun, "Mp3: A unified model to map, perceive, predict and plan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 403–14 412.
- [10] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, et al., "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 17 853–17 862.
- [11] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun, "End-to-end interpretable neural motion planner," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8660–8669.
- [12] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [13] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," *arXiv preprint arXiv:1903.11027*, 2019.
- [14] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, A. Zisserman, and K. Simonyan, "Flamingo: a visual language model for few-shot learning," 2022.
- [15] G. G. Team, "Gemini: A family of highly capable multimodal models," 2024. [Online]. Available: <https://arxiv.org/abs/2312.11805>
- [16] S. Addepalli, A. R. Asokan, L. Sharma, and R. V. Babu, "Leveraging vision-language models for improving domain generalization in image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 23 922–23 932.
- [17] Y. Zhai, S. Tong, X. Li, M. Cai, Q. Qu, Y. J. Lee, and Y. Ma, "Investigating the catastrophic forgetting in multimodal large language model fine-tuning," in *Conference on Parsimony and Learning*. PMLR, 2024, pp. 202–227.
- [18] S. Casas, W. Luo, and R. Urtasun, "Intentnet: Learning to predict intention from raw sensor data," in *CoRL*, 2018.
- [19] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," in *CoRL*, 2019.
- [20] J. Hong, B. Sapp, and J. Philbin, "Rules of the road: Predicting driving behavior with a convolutional model of semantic interactions," in *CVPR*, 2019.
- [21] Y. Biktairov, M. Stebelev, I. Rudenko, O. Shliazhko, and B. Yangel, "Prank: motion prediction based on ranking," *Advances in neural information processing systems*, vol. 33, pp. 2553–2563, 2020.
- [22] S. Casas, C. Gulino, R. Liao, and R. Urtasun, "Spagnn: Spatially-aware graph neural networks for relational behavior forecasting from sensor data," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 9491–9497.
- [23] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *ECCV*. Springer, 2020, pp. 541–556.
- [24] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2255–2264.
- [25] C. Tang and R. R. Salakhutdinov, "Multiple futures prediction," *Advances in neural information processing systems*, vol. 32, 2019.
- [26] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine, "Precog: Prediction conditioned on goals in visual multi-agent settings," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2821–2830.
- [27] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp, "Wayformer: Motion forecasting via simple & efficient attention networks," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2980–2987.
- [28] O. G.- Team, "Gpt-4 technical report," 2024.
- [29] Z. Xu, Y. Zhang, E. Xie, Z. Zhao, Y. Guo, K.-Y. K. Wong, Z. Li, and H. Zhao, "Drivegpt4: Interpretable end-to-end autonomous driving via large language model," *RA-L*, 2024.
- [30] C. Sima, K. Renz, K. Chitta, L. Chen, H. Zhang, C. Xie, P. Luo, A. Geiger, and H. Li, "Drivelm: Driving with graph visual question answering," in *ECCV*, 2024.
- [31] X. Tian, J. Gu, B. Li, Y. Liu, C. Hu, Y. Wang, K. Zhan, P. Jia, X. Lang, and H. Zhao, "Drivevlm: The convergence of autonomous driving and large vision-language models," in *CoRL*, 2024.
- [32] J.-J. Hwang, R. Xu, H. Lin, W.-C. Hung, J. Ji, K. Choi, D. Huang, T. He, P. Covington, B. Sapp, J. Guo, D. Anguelov, and M. Tan, "Emma: End-to-end multimodal model for autonomous driving," 2024. [Online]. Available: <https://arxiv.org/abs/2410.23262>
- [33] H. Huang, O. Zheng, D. Wang, J. Yin, Z. Wang, S. Ding, H. Yin, C. Xu, R. Yang, Q. Zheng, et al., "Chatgpt for shaping the future of dentistry: the potential of multi-modal large language model," *International Journal of Oral Science*, vol. 15, no. 1, p. 29, 2023.
- [34] A. Shvedritskii, C. Rupprecht, and A. Vedaldi, "What does clip know about a red circle? visual prompt engineering for vlms," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 11 987–11 997.
- [35] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou, et al., "Chain-of-thought prompting elicits reasoning in large language models," *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022.
- [36] N. Mu, J. Ji, Z. Yang, N. Harada, H. Tang, K. Chen, C. R. Qi, R. Ge, K. Goel, Z. Yang, et al., "Most: Multi-modality scene tokenization for motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 14 988–14 999.
- [37] S. Shi, L. Jiang, D. Dai, and B. Schiele, "Motion transf. with global intention localization and local movement refinement," *NeurIPS*, 2022.
- [38] L. Feng, M. Bahari, K. M. B. Amor, É. Zablocki, M. Cord, and A. Alahi, "Unitraj: A unified framework for scalable vehicle trajectory prediction," *arXiv preprint arXiv:2403.15098*, 2024.
- [39] Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," *arXiv preprint arXiv:1910.05449*, 2019.
- [40] B. Varadarajan, A. Hefny, A. Srivastava, K. S. Refaat, N. Nayakanti, A. Cornman, K. Chen, B. Douillard, C. P. Lam, D. Anguelov, et al., "Multipath++: Efficient information fusion and trajectory aggregation for behavior prediction," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 7814–7821.
- [41] J. Philion, X. B. Peng, and S. Fidler, "Trajeglish: Learning the language of driving scenarios," *arXiv e-prints*, pp. arXiv–2312, 2023.
- [42] P. Wang, S. Bai, S. Tan, S. Wang, Z. Fan, J. Bai, K. Chen, X. Liu, J. Wang, W. Ge, Y. Fan, K. Fang, M. Du, X. Ren, R. Men, D. Liu, C. Zhou, J. Zhou, and J. Lin, "Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution," *arXiv preprint arXiv:2409.12191*, 2024.
- [43] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, et al., "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 9710–9719.
- [44] T. Dao, D. Fu, S. Ermon, A. Rudra, and C. Ré, "Flashattention: Fast and memory-efficient exact attention with io-awareness," *NeurIPS*, vol. 35, pp. 16 344–16 359, 2022.

# Enhanced Motion Forecasting with Plug-and-Play Multimodal Large Language Models

## Appendix

### VII. MLLM PROMPTING DETAILS

We include the exact prompts used for the Visual Semantic Analyzer (VSA) and Scene Categorizer (SC) components in the following sections.

#### A. VSA Prompting

For the VSA component, we break our prompts down by vehicle class and pedestrian class, to best target challenging cases and behaviors. We present our prompt for the vehicle class in Fig. 9, targeting difficult cases including emergency vehicles (EV), and getting additional information such as vehicle type, signaling behavior, and high-level intention approximation. Similarly, we present the prompt for the pedestrian class in Fig. 10, where we tackle edge cases such as micromobility (scooters, skateboards, *etc.*), jaywalking case, and high-level intention approximation. In both prompts, we use chain-of-thought and examples in the prompt to encourage correct behavior. We include multi-modal prompting, inserting image crops and scene images, shown in Fig. 2 of the main text.

#### B. SC Prompting

We provide the prompt used for the SC component in Fig. 11. We query for prediction dependent fields such as weather, time of day, and type of road. Likewise, we use chain-of-thought and examples for improved prompt performance.

### VIII. ADDITIONAL QUALITATIVE RESULTS

#### A. Full Ablations Tables

We additionally include all metrics for tables in the main text where columns may have truncated due to space. In Tab. VI, we report results corresponding to Tab. 4 of the main text, where we ablate the inclusion of different reasoning types of the VSA and SC components on final prediction performance. We include all metrics for the performance on the hardest 10% split in Tab. VII. Similarly, we report all results for the ablation on how the gain controls incorporation of information from the MLLM components in Tab. VIII.

#### B. Uncertainty Estimates

We report the standard error in Tab. IX. Standard error quantifies the variability or uncertainty in a sample statistic as an estimate of the corresponding population parameter, reflecting how much the statistic is expected to fluctuate across different samples. We estimate the standard error (Std. Err.) over the entire validation set, to measure how much our method varies across different inputs. Observe that standard error is low across all metrics, and that performance gains obtained by applying PnF are statistically significant over the baselines.

### IX. QUALITATIVE RESULTS

#### A. Additional Qualitative Results

We visualize additional qualitative results in Fig. 12, with an additional focus on the pedestrian class. Observe that performance gains obtained by adding PnF is consistent across class, and the MLLM queried features helps to identify modes of behaviors.

#### B. VSA and SC Outputs

We visualize qualitative outputs of the MLLM components, VSA and SC. We include an example on the “Vehicle” class in Fig. 13 and an example on the “Pedestrian” class in Fig. 14.

### X. ADDITIONAL DETAILS

#### A. Parameter Count Analysis

Our method, PnF, consists of a learnable embedding of dimension 128 for each of the structured text outputs. Our learnable gain is a 2-layer MLP on top of the inputs. In total, this increases our parameter count from approximately 7.32 million parameters to approximately 7.33 million parameters, for an approximate 0.15% increase on trainable parameters for the Wayformer model. We consider this a very lightweight addition, thereby increasing the practicality in adopting our method to use off-the-shelf MLLM’s for improving prediction performance.

#### B. WOMD Dataset Details

We provide additional details into the Waymo Open Motion Dataset (WOMD). WOMD is a real-work self-driving dataset, consisting of both camera image inputs and annotated motion tracks. The dataset is collected with 9 surround-view cameras around the vehicle. Motion trajectory annotations are labeled at 10Hz. The prediction task uses 1 second of past track information, and evaluates on 8 seconds of future information. We additionally give camera image information, similar to the NuScenes dataset [13]. Camera information is provided up to the frame closest to the current prediction frame, at the end of 1 second past information.

### Vehicle Class Prompt (VSA)

You are a self driving car. You have 8 cameras pointing in all directions. For all objects in the scene, tell me the car type, signals, and intentions. Is the vehicle an emergency vehicle or first responder? Select from: <YES>, <NO>, <UNSURE>. What type of car? Select from: <SEDAN>, <TRUCK>, <BUS>, <SUV>, <OTHER>. Does this car have <TURN SIGNAL>, <BRAKE LIGHTS>, <HAZARD LIGHTS>, <NONE>, or <UNSURE> on? Is this car heavily occluded by a stationary object? Select from: <YES>, <NO>, <UNSURE>. How will the agent move in the next 3 seconds? Will it <KEEP FORWARD>, <SLOW DOWN>, <TURN>, <U-TURN>, <PARKED>, <STOP>. Select from: <YES>, <NO>, <UNSURE>. For example:

\*\*Explanation of certain predictions:\*\*

\* \*\*SUV (first row):\*\* This SUV is approaching the intersection and appear to be have its brake lights on. Thus, it's likely to slow down, with a small likelihood of turning or stopping.

<\ANSWER>

Emergency Vehicle?	Vehicle Type	Signal	Keep Forward	Slow Down	Turn	U-Turn	Parked	Stop	Heavy Occlusion
----- ----- ----- ----- ----- ----- ----- ----- ----- -----									
NO   SUV   BRAKE LIGHTS   NO   YES   UNSURE   NO   NO   UNSURE   NO									
NO   SEDAN   NONE   YES   UNSURE   UNSURE   NO   NO   NO   NO									
<\ANSWER>									

For each vehicle, I also give you 3 frames from the past second:<img> <img> <img> ...

Also use the scene context: the front camera took the following image around you of the scene: Front camera:<img> Front left camera:<img> Front right camera:<img>

Default to <UNSURE> unless absolutely certain. Answer <UNSURE> if you cannot see clearly. Explain first before answering, output the table of responses last, wrapped with <\ANSWER> tags.

Fig. 9: Vehicle Class Prompt (VSA)

MLLM Reasoning Over				Displ. Error		Avg. Prec.	
Signal	Intention	Type	Scene	minADE (↓)	minFDE (↓)	mAP (↑)	soft-mAP (↑)
✓	✓	✓	✓	0.529	1.084	0.437	0.457
✓	✓			0.533	1.094	0.433	0.453
	✓	✓		0.529	1.082	0.428	0.449
✓		✓		0.528	1.081	0.436	0.457
		✓		0.530	1.086	0.433	0.453

TABLE VI: Ablation table showing the different reasoning types inclusion on performance.

Method	minADE (↓)	minFDE (↓)	Miss Rate (↓)	Overlap Rate (↓)	mAP (↑)	soft-mAP (↑)
Wayformer [27]	0.819	1.737	0.252	0.138	0.294	0.305
+ PnF	0.753	1.589	0.217	0.137	0.318	0.330
Δ	-7.99%	-8.52%	-13.91%	-0.73%	7.96%	8.37%

TABLE VII: Performance on the hardest subset on the WOMD.

Gain Ablation	minADE (↓)	minFDE (↓)	Miss Rate (↓)	Overlap Rate (↓)	mAP (↑)	soft-mAP (↑)
None (added)	0.531	1.090	0.114	0.127	0.433	0.454
Constant	0.531	1.085	0.110	0.127	0.428	0.449
Learned (Ours)	0.529	1.084	0.113	0.127	0.437	0.457

TABLE VIII: Ablation on how gain values are incorporated.

### Pedestrian Class Prompt (VSA)

You are a self driving car. You have 8 cameras pointing in all directions. For all pedestrians in the scene, tell me if they are jaywalking, if they have a micromobility (scooter, skateboard), and their intentions. Is the pedestrian jaywalking? Select from: <YES>, <NO>, <UNSURE>. Is this pedestrian on a micromobility vehicle, such as a scooter or skateboard? Select from: <YES>, <NO>, <UNSURE>. How will the agent move in the next 3 seconds? Will they <WALK SIDEWALK>, <CROSS>, <TURN>, <STOP>, <WAITING>. Select from: <YES>, <NO>, <UNSURE>. Is the pedestrian occluded or not visible? Select from: <YES>, <NO>, <UNSURE>. For example:

\*\*Explanation of certain predictions:\*\*

\* \*\*Pedestrian on Scooter (first row):\*\* The pedestrian is on a electric scooter, and is legally crossing at an intersection. We should drive carefully and yield. Because the crossing is at a crosswalk, it is legal, and therefore not jaywalking. Finally, because it is at the crosswalk, they'll likely cross in the next 3 seconds.

<ANSWER>

Jay Walking?	Micromobility	Walk on Sidewalk	Cross	Turn	Stop	Waiting	Low Visibility	
----- ----- ----- ----- ----- ----- ----- ----- -----								
NO	YES	NO	YES	NO	NO	NO	NO	
NO	NO	YES	NO	YES	NO	NO	NO	

<\ANSWER>

For each pedestrian, I also give you 3 frames from the past second:<img> <img> <img> ...

Make sure to use the scene context: the front cameras took the following image around you of the scene: Front camera:<img> Front left camera:<img> Front right camera:<img>  
Use the scene context to identify where the road and sidewalks are. Default to <UNSURE> unless absolutely certain. Answer <UNSURE> if you cannot see clearly. Explain first before answering, output the table of responses last, wrapped with <ANSWER> tags.

Fig. 10: Pedestrian Class Prompt (VSA)

Method		minADE ( $\downarrow$ )	minFDE ( $\downarrow$ )	Miss Rate ( $\downarrow$ )	Overlap Rate ( $\downarrow$ )	mAP ( $\uparrow$ )	soft-mAP ( $\uparrow$ )
Wayformer <sup>†</sup>	Metric	0.539	1.111	0.119	0.128	0.425	0.446
	Std. Err.	$\pm 0.002$	$\pm 0.004$	$\pm 0.001$	$\pm 0.001$	$\pm 0.001$	$\pm 0.001$
	Metric	0.529	1.084	0.113	0.127	0.437	0.457
	Std. Err.	$\pm 0.002$	$\pm 0.004$	$\pm 0.001$	$\pm 0.001$	$\pm 0.001$	$\pm 0.001$
MotionLM <sup>†</sup>	Metric	0.574	1.189	0.139	0.129	0.382	0.403
	Std. Err.	$\pm 0.004$	$\pm 0.008$	$\pm 0.001$	$\pm 0.001$	$\pm 0.001$	$\pm 0.001$
	Metric	0.565	1.166	0.132	0.129	0.390	0.413
	Std. Err.	$\pm 0.004$	$\pm 0.007$	$\pm 0.001$	$\pm 0.001$	$\pm 0.001$	$\pm 0.001$

TABLE IX: Standard Errors on WOMD.

### Scene Prompt (SC)

You are a self driving car. You have a camera pointing forward (this image is what is in front of you), and you see the following scene before you. You need to check what the weather condition of your location is: sunny, rainy, snowy, foggy, dark, or unsure. You need to check if it is day time, evening, or night time. You need to check what type of road you are on: residential, highway, express way, service road, or other. Are you approaching an intersection? You need to check how the scene will evolve in the next 3 seconds, and what are actors of interest? What is the weather condition? Select from: <SUNNY>, <RAINY>, <SNOWY>, <FOGGY>, <DARK>, <UNSURE>. What is the time of day? Select from: <DAY>, <EVENING>, <NIGHT>. What type of road you are on? Select from: <RESIDENTIAL>, <HIGHWAY>, <EXPRESS>, <SERVICE>, <OTHER>, <UNSURE>. Are you approaching an intersection or lane merge? Select from: <YES>, <NO>, <UNSURE>.

Example 1, the weather condition is raining, and there is a tree in front of you, You should respond with:

Logic: There are no actors of interest, as the road is empty besides me. Based on the image, it is raining, reducing the visibility. The primary concern is the tree object that is in the road. The light is still out, which suggests it is day time. I am driving on a road that is not a road with buildings on either side, suggesting that it is a residential road. I am not approaching any intersections, or merging lanes. The anomaly is the tree branch blocking the road. This is a dangerous situation and the car needs to take action to avoid it.

Final answer: <RAINY> <DAY> <RESIDENTIAL> <NO>

Example 2, there is an emergency vehicle in front of me, and cars surrounding me have started slowing down (brake lights are visible), You should respond with:

Logic: The actor of interest is the white ambulance vehicle and the grey car in front of me that is slowing down. The road ahead is straight with an ambulance with flashing lights exiting from a service road. The anomaly is the emergency vehicle that is in front of me. I see the car in front of me slowing down, suggesting an emergency situation. The image shows a clear sky and well-lit surroundings, indicating sunny weather and daytime. The presence of buildings and streetlights suggests a residential area. There is an anomaly, since there is an emergency vehicle. The ambulance, which is an emergency vehicle, is in front of me, which suggests I should be careful and yield to it in the next 3 seconds.

Final answer: <SUNNY> <DAY> <RESIDENTIAL> <NO>

Now we have an image of this scene:<img>

Fig. 11: Scene Prompt (SC)

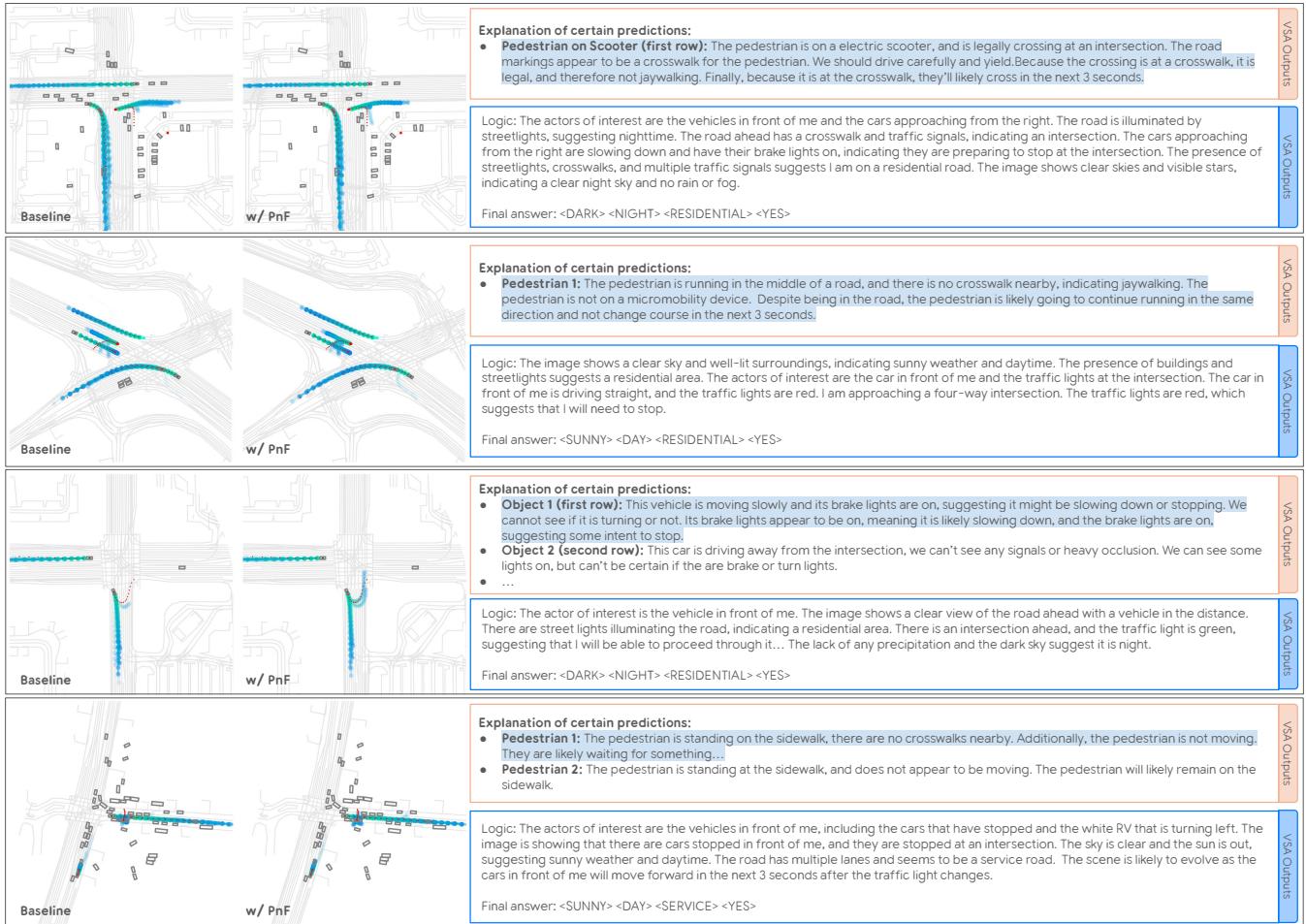


Fig. 12: **Additional Qualitative Results.** We visualize additional prediction outputs on the Wayformer baseline model with and without PnF applied. We focus on pedestrian prediction results in this selection, where the 1st, 2nd, and 4th results demonstrate our method on pedestrian prediction. We show an additional prediction result for the vehicle class in the 3rd row.



(a) Image input for visualization.

#### VSA Output

**\*\*Explanation of certain predictions:\*\***

\* **\*\*SUV (first row):\*\*** This SUV is approaching the intersection and appear to be have its brake lights on. Thus, it's likely to slow down, with a small likelihood of turning or stopping. Also, from the front camera image, we can see the car has entered the intersection and is likely to slow down, so it's unlikely to keep forward.

\* **\*\*SEDAN (second row):\*\*** It is difficult to predict the intention of the sedan, as there is no information about the signal or current position. It is likely to keep forward as it is in the centre of the lane.

<ANSWER>

```
| Emergency Vehicle? | Vehicle Type | Signal | Keep Forward | Slow Down | Turn | U-Turn | Parked | Stop | Heavy
Occlusion |
|---|---|---|---|---|---|---|---|
| NO | SUV | BRAKE LIGHTS | NO | YES | UNSURE | NO | NO | UNSURE | NO |
| NO | SEDAN | NONE | YES | UNSURE | UNSURE | NO | NO | NO | NO |
<\ANSWER>
```

(b) VSA raw output.

#### SC Output

Logic: The scene shows a white SUV driving on a wet road, with a building in the background and trees on both sides. The sky is overcast and there are puddles on the road, indicating that it is raining. The lighting suggests it is daytime. Based on the environment and the presence of shops in the background, the road is likely a service road. There is a white car in front of me, and there is a lane merge to the right. The anomaly is the white car in front of me. I need to adjust my speed and be prepared for the lane merge in the next 3 seconds.

Final answer: <RAINY> <DAY> <SERVICE> <YES>

(c) SC raw output

Fig. 13: **VSA and SC component outputs demonstrated on the vehicle class.** Observe that the final output is structured in a tagged format.



(a) Image input for visualization.

#### VSA Output

**\*\*Explanation of certain predictions:\*\***

\* **Pedestrian on Scooter (first row):** The pedestrian is on a electric scooter, and is legally crossing at an intersection. The road markings appear to be a crosswalk for the pedestrian. We should drive carefully and yield.Because the crossing is at a crosswalk, it is legal, and therefore not jaywalking. Finally, because it is at the crosswalk, they'll likely cross in the next 3 seconds.

<ANSWER>

Jay Walking?	Micromobility	Walk on Sidewalk	Cross	Turn	Stop	Waiting	Low Visibility
----- ----- ----- ----- ----- ----- ----- -----							
NO	YES	NO	YES	NO	NO	NO	NO

<\ANSWER>

(b) VSA raw output

#### SC Output

Logic: The actors of interest are the vehicles in front of me and the cars approaching from the right. The road is illuminated by streetlights, suggesting nighttime. The road ahead has a crosswalk and traffic signals, indicating an intersection. The cars approaching from the right are slowing down and have their brake lights on, indicating they are preparing to stop at the intersection. The presence of streetlights, crosswalks, and multiple traffic signals suggests I am on a residential road. The image shows clear skies and visible stars, indicating a clear night sky and no rain or fog.

Final answer: <DARK> <NIGHT> <RESIDENTIAL> <YES>

(c) SC raw output

Fig. 14: **VSA and SC component outputs demonstrated on the pedestrian class.** The answers are similarly structured and wrapped with tags for easy downstream use.