LightEMMA: Lightweight End-to-End Multimodal Model for Autonomous Driving

Zhijie Qiao[1,†], Haowei Li[1,†], Zhong Cao[1], Henry X. Liu[1,2,*]

***Abstract*— Vision-Language Models (VLMs) have demonstrated significant potential for end-to-end autonomous driving. However, fully exploiting their capabilities for safe and reliable vehicle control remains an open research challenge. To systematically examine advances and limitations of VLMs in driving tasks, we introduce LightEMMA, a Lightweight End-to-End Multimodal Model for Autonomous driving. LightEMMA provides a unified, VLM-based autonomous driving framework without ad hoc customizations, enabling easy integration and evaluation of evolving state-of-the-art commercial and open-source models. We construct twelve autonomous driving agents using various VLMs and evaluate their performance on the nuScenes prediction task, comprehensively assessing metrics such as inference time, computational cost, and predictive accuracy. Illustrative examples highlight that, despite their strong scenario interpretation capabilities, VLMs' practical performance in autonomous driving tasks remains concerning, emphasizing the need for further improvements. The code is available at https://github.com/michigan-traffic-lab/LightEMMA.**

I. Introduction

Autonomous vehicles (AVs) have seen tremendous advancements over the years, improving safety, comfort, and reliability. Traditional approaches rely on modular designs, rule-based systems, and predefined heuristics [1], [2]. While this structured methodology ensures interpretable and predictable behavior, it limits the ability to interpret complex scenes and make flexible, human-like decisions.

A more recent approach is learning-based end-to-end driving, which maps raw sensor inputs—along with HD maps and environmental context—directly to a driving trajectory [3]–[8]. Unlike modular pipelines, end-to-end models aim to learn a unified representation from data, enabling more holistic and potentially efficient driving decisions. However, they are often black boxes with limited interpretability, raising safety concerns in critical scenarios [9], and they require vast, diverse data, making them vulnerable to data imbalance and the curse of rarity [10].

An emerging approach with promise in addressing these challenges is the advancement of Vision-Language Models (VLMs). Trained on extensive datasets containing text, images, and videos, VLMs exhibit robust reasoning capabilities reminiscent of human-like cognition. Recent research has investigated end-to-end autonomous driving systems based on VLMs, with a comprehensive survey provided in [11]. However, existing studies primarily emphasize VLMs' scene understanding capabilities in driving contexts without fully evaluating their strengths and limitations. Additionally, many applications involve commercial vehicle deployments without accessible source code or detailed implementations, limiting their availability for broader research and collaboration.

Inspired by the recent advancements in EMMA [12] and an open-source implementation effort, OpenEMMA [13], we introduce LightEMMA—a lightweight, end-to-end multimodal framework for autonomous driving. LightEMMA adopts a zero-shot approach and fully harnesses the capabilities of existing VLMs. Our key contributions are as follows:

1) We provide an open-source baseline workflow for end-to-end autonomous driving planning tasks, designed to seamlessly integrate with the latest VLMs, enabling rapid prototyping while minimizing computational and transfer overhead.
2) We conduct a comprehensive evaluation of twelve state-of-the-art commercial and open-source VLMs using 150 test scenarios from the nuScenes prediction task. Our analysis highlights the practical strengths and limitations of current VLM-based driving strategies, providing a detailed discussion of their capabilities and potential areas for future improvement.

II. Related Work

EMMA [12], built on Gemini [14], directly maps camera data to driving outputs by uniformly representing inputs and outputs in natural language, achieving state-of-the-art motion planning. OpenEMMA [13] extends this by introducing an open-source framework using VLMs enhanced by Chain-of-Thought (CoT) reasoning, improving performance and generalizability. DriveGPT4 [15], a LLaMA2-based VLM trained on the BDD-X dataset and fine-tuned with ChatGPT data, supports multi-frame video understanding, textual queries, and vehicle control predictions. DOLPHINS [16] uses instruction tuning for in-context learning, adaptation, and error recovery. DriveMLM [17] incorporates VLM into behavior planning by integrating driving rules, user inputs, and sensor data, evaluated in CARLA's Town05 [18].

Several open-source datasets are available for training and evaluating autonomous driving systems, notably the Waymo Open Dataset [19] and nuScenes [20]. Extended benchmarks like nuScenes-QA [21], nuPrompt [22], LingoQA [23], and Reason2Drive [24] further support evaluation of language and reasoning capabilities.

This research was partially funded by the DARPA TIAMAT Challenge (HR0011-24-9-0429).

[1]Z. Qiao, H. Li, Z. Cao, and H. X. Liu are with the Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI 48109, USA.

[2]H. X. Liu is also with University of Michigan Transportation Research Institute, Ann Arbor, MI 48109, USA.

†These authors contributed equally to this work.

*Corresponding author: Henry X. Liu (henryliu@umich.edu).

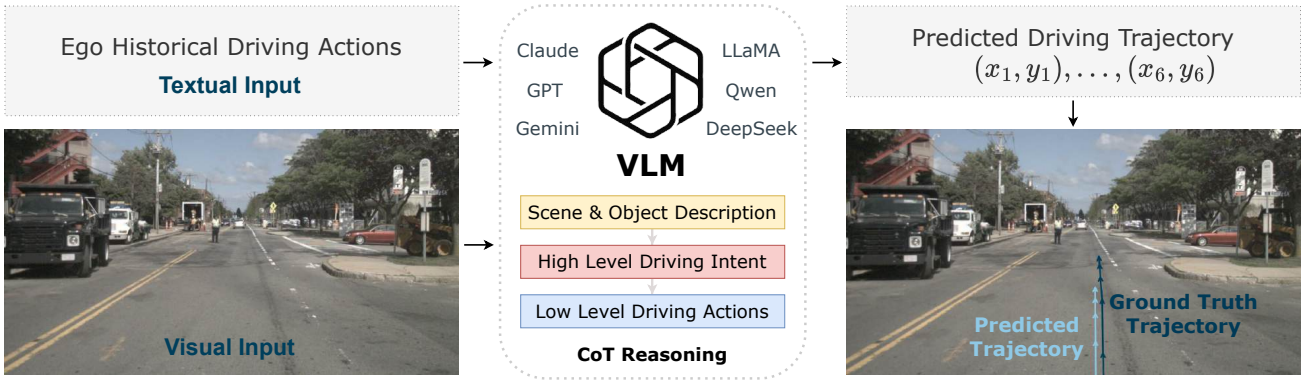

Fig. 1. LightEMMA Architecture.

III. Methodology

An overview of the LightEMMA architecture is shown in Fig. 1. A brief workflow is outlined below, with detailed descriptions provided in subsequent subsections.

For each inference cycle, the current front-view camera image and historical vehicle driving data are input into the VLM. To enhance interpretability and facilitate structured reasoning, we employ a Chain-of-Thought (CoT) prompting strategy, whose final stage explicitly outputs a sequence of predicted control actions. These actions are numerically integrated to produce the predicted trajectory, which is subsequently compared against the ground truth. All VLMs are evaluated uniformly, using consistent prompting and evaluation procedures without model-specific adaptations.

A. VLM Selection

We select state-of-the-art VLMs from both open-source and commercial offerings, covering 6 model types with a total of 12 models. For each model type, we evaluate two variants: a basic version and an advanced version. All models used are the latest publicly available versions (as of the end of this project) that support both textual and image inputs. This setup enables comprehensive performance comparisons both across different models and between variants within the same model family. The selected models are: GPT-4o, GPT-4.1 [25], Gemini-2.0-Flash, Gemini-2.5-Pro [14], Claude-3.5-Sonnet, Claude-3.7-Sonnet [26], DeepSeek-VL2-16B, DeepSeek-VL2-28B [27], LLaMA-3.2-11B-Vision-Instruct, LLaMA-3.2-90B-Vision-Instruct [28], Qwen2.5-VL-7B-Instruct, and Qwen2.5-VL-72B-Instruct [29].

For commercial models, we access them via paid APIs. This approach simplifies deployment by eliminating the need to manage local hardware, software updates, and scalability, as these tasks are handled directly by the providers.

For open-source models, we download them from HuggingFace and deploy them locally using H100 GPUs. Most models require only a single H100 GPU, although larger models may require more; we report the minimum number of GPUs needed in Table I. To facilitate multi-GPU deployments, we leverage PyTorch's automatic device mapping for efficient GPU utilization.

B. Camera Input

When the front-view camera image is input into the VLM, we do not employ any visual encoders such as CLIP [30], nor do we apply preprocessing techniques to modify the image before feeding it to the model. Our findings indicate that VLMs effectively describe the scene and accurately identify objects directly from raw visual inputs, demonstrating robustness in handling unprocessed visual data.

In line with this design approach, we also choose to use only the current driving scene image as input, rather than concatenating multiple past frames as performed in previous studies [13], [15]. Our preliminary experiments indicate that incorporating additional frames does not yield noticeable performance gains. Instead, the model tends to redundantly extract identical features across multiple frames rather than capturing meaningful spatiotemporal dynamics. Additionally, adding more frames results in a roughly linear increase in processing time and computational cost without clear performance benefits.

Alternatively, models such as VideoBERT [31] and VideoMAE [32] support video inputs through specialized temporal encodings rather than simply treating videos as frame sequences. Such models inherently adopt different architectures and could potentially capture richer temporal information. However, exploring these specialized video-oriented models falls outside the scope of our current study.

C. Driving History Input

Our work represents vehicle actions using speed and curvature, an interpretable format in which speed captures longitudinal motion and curvature describes lateral movement. We adopt this representation instead of explicit trajectory points because VLMs often struggle to reason effectively about coordinates that involve implicit physical constraints.

D. VLM Promoting

We adopt a straightforward CoT approach to guide the VLM in scene understanding and action generation, where the output from each stage is integrated into the subsequent stage along with additional prompt:

TABLE I
COMPARISON OF MODEL EFFICIENCY AND COMPUTATIONAL COST

Model	Infer Time (s)	Infer Cost (¢)	Input Tokens	Output Tokens	H100 GPUs
GPT-4o	12.1	1.4	4402	341	-
GPT-4.1	15.1	1.24	4495	425	-
Claude-3.5-Sonnet	13.4	0.65	5929	452	-
Claude-3.7-Sonnet	14.8	2.47	5946	461	-
Gemini-2.0-Flash	4.5	0.07	6398	272	-
Gemini-2.5-Pro	36.5	3.7†	1896†	3276†	-
DeepSeek-VL2-16B	10.0	-	6042	255	1
DeepSeek-VL2-28B	13.9	-	6398	277	1
LLaMA-3.2-11B-Vision-Instruct	7.4	-	1037†	311	1
LLaMA-3.2-90B-Vision-Instruct	40.8	-	1077†	356	3
Qwen-2.5-VL-7B-Instruct	8.8	-	6554	318	1
Qwen-2.5-VL-72B-Instruct	32.3	-	6630	369	2

1) **Scenario Description:** The VLM receives an image as input and is prompted to interpret the overall scene, including lane markings, traffic lights, vehicles, pedestrian activities, and other pertinent objects.
2) **High-Level Driving Intent:** The generated scene description is integrated with the ego vehicle's historical driving actions, allowing the VLM to interpret past behaviors within the current scene context and predict the next high-level driving action.
3) **Low-Level Driving Commands:** The scene description and generated high-level command are used to prompt the VLM to output a structured list of low-level driving actions in the specified format as [(v1, c1), (v2, c2), (v3, c3), (v4, c4), (v5, c5), (v6, c6)], without additional text or explanations.

E. Example Workflow

In this section, we demonstrate an example workflow and results from the CoT process using the images shown in Fig. 1. The responses and the trajectory are generated by Claude-3.7-Sonnet. Due to space limitations, we preserve the original wording but omitted some less critical details.

Scenario Description: The image shows an urban street scene with active road construction or maintenance work...A police officer is standing in the middle of the road directing traffic...There are several construction vehicles visible including a black dump truck parked on the left side of the road along with other work vehicles...Traffic appears to be controlled through this construction zone with the presence of the officer and possible lane restrictions.

High-Level Intent: Given the current scene with active construction and a police officer directing traffic in the middle of the road, the ego vehicle should significantly decelerate by approximately $2-3m/s^2$ to prepare for possible stopping, reducing speed to around $3-4m/s$ in the next 3 seconds. The vehicle should continue following the lane while being prepared to come to a complete stop if directed by the officer.

Low-Level Commands: [(6.0, -0.001), (5.0, -0.001), (4.0, 0.0), (3.5, 0.0), (3.0, 0.0), (3.0, 0.0)].

IV. EXPERIMENTS

Using the proposed methods and models, we evaluate performance on the nuScenes prediction task across a total of 3,908 frames drawn from 150 test scenarios. The evaluation concentrates on two aspects: computational efficiency of the models and the accuracy of their trajectory predictions.

A. Inference Time

A summary of inference times is presented in Table I, showing the average processing time per image frame. Gemini-2.0-Flash achieves the fastest inference at only 4.5 seconds per frame, while LLaMA-3.2-90b is the slowest at 40.8 seconds. Qwen-2.5-72B and Gemini-2.5-Pro also exhibit relatively slow performance, each requiring over 30 seconds per frame. The remaining models typically operate around 10 seconds per frame, with basic versions generally running faster than their advanced counterparts.

Note that even the fastest model, Gemini-2.0-Flash, has a processing time significantly slower than the real-time update frequency. To be genuinely effective for real-world deployment, these models would need to operate one to two orders of magnitude faster. Additionally, API-based commercial models rely on stable internet connections, which might be unreliable in a moving vehicle. Conversely, local deployment faces constraints from limited computing power and energy consumption, further restricting their practicality.

B. Input and Output Tokens

We compute the average number of input and output tokens per frame using the official instructions provided by each model. As shown in Table I, the number of input tokens is significantly higher than the output tokens, typically around 6000 input tokens compared to roughly 300 output tokens. This aligns with expectations, as inputs include image data while outputs are purely textual.

However, there are some exceptions. The LLaMA models report only about 1000 input tokens per frame. Upon further investigation, we discovered that the official LLaMA token counting method excludes image tokens, counting only text.

TABLE II
PERFORMANCE COMPARISON ON NUSCENES PREDICTION TASK

Model	Response Error (%)	L2 1s (m)	L2 2s (m)	L2 3s (m)	L2 avg (m)
GPT-4o	7.8	**0.28**	**0.93**	**2.02**	**1.07**
GPT-4.1	28.9	-	-	-	-
Claude-3.5-Sonnet	0.4	0.29	0.98	2.12	1.13
Claude-3.7-Sonnet	0.0	0.28	0.94	2.04	1.09
Gemini-2.0-Flash	0.7	0.31	1.08	2.36	1.25
Gemini-2.5-Pro	0.0	0.37	1.35	2.96	1.56
DeepSeek-VL2-16B	0.9	0.66	1.68	2.92	1.75
DeepSeek-VL2-28B	0.0	0.66	1.71	3.01	1.79
LLaMA-3.2-11B-Vision-Instruct	0.7	0.52	1.42	2.68	1.54
LLaMA-3.2-90B-Vision-Instruct	0.0	0.34	1.14	2.45	1.31
Qwen-2.5-VL-7B-Instruct	0.0	0.46	1.33	2.55	1.45
Qwen-2.5-VL-72B-Instruct	62.9	-	-	-	-
Simple Baseline	-	0.29	0.96	2.06	1.10

Despite extensive efforts, we could not identify a reliable approach to accurately estimate image-related tokens; therefore, we present these results as provided by the official method.

Additionally, the token counts for Gemini-2.5-Pro clearly contain errors in both input and output token calculations, as they significantly deviate from the results of comparable models. Notably, Gemini-2.0-Flash, calculated using the identical token counting setup, produced consistent and reasonable results, indicating an issue specific to Gemini-2.5-Pro that needs correction.

C. Cost

The cost section applies exclusively to commercial APIs. To ensure accurate measurement and reporting, billing history is cross-referenced with official pricing tables, based on input and output token usage. For clarity, all results presented in Table I are shown in cents per frame.

Gemini-2.0-Flash is the cheapest at only 0.07, making its cost negligible. GPT-4o and GPT-4.1 exhibit similar costs, approximately 1.3. Claude-3.7-Sonnet is significantly more expensive than Claude-3.5-Sonnet, and notably pricier than the GPT models. Due to inaccuracies in Gemini-2.5-Pro's token calculations, an exact estimate is difficult. Thus, the value reported here is based solely on billing history after running the model.

D. Response Error

In the final model output stage, we observed diverse response formatting errors. Although we prompt the VLM to strictly return outputs in the format [(v1, c1), (v2, c2), (v3, c3), (v4, c4), (v5, c5), (v6, c6)] without additional text, we occasionally encountered deviations such as missing brackets or commas, additional explanation or punctuation, and incorrect list lengths. Specific examples are omitted here but are available in our GitHub repository.

As shown in Table II, Qwen-2.5-72B exhibits the highest error rate at 62.9%, while its basic counterpart, Qwen-2.5-7B, produced no errors. GPT-4.1 also demonstrates a problematic error rate of 28.9%, and GPT-4o shows fewer errors at 7.8%. The remaining models perform reliably, displaying either zero or less than 1% error rates.

We argue that, given identical prompting and workflow across all models, these random failures reflect inherent model limitations rather than systematic flaws in our framework. While many formatting errors could be mitigated through post-processing, additional prompts, or other enhancement techniques, our objective is to evaluate, not optimize, individual model performance. Therefore, we maintain a consistent experimental design and report observed error rates without modification.

E. Prediction Accuracy

The prediction accuracy follows the standard evaluation approach adopted in the nuScenes prediction task, reporting the L2 loss at 1s, 2s, and 3s intervals, along with their average value. Due to response errors, each model yields predictions for a different subset of the original frames. To ensure a fair comparison, we exclude any frame from evaluation for all models if any model fails to produce a valid prediction for it. Because Qwen-2.5-72B and GPT-4.1 exhibit particularly high failure rates, we exclude these two models entirely from this analysis to retain a sufficiently large set of frames. Ultimately, this filtering results in a subset of 3506 frames out of the original 3908, preserving 90 percent of the data.

The L2 loss results are summarized in Table II. For simplicity and ease of comparison, our analysis primarily focuses on the average L2 loss (in meters); Overall, GPT-4o achieves the best performance at 1.07 m, closely followed by Claude-3.5-Sonnet and Claude-3.7-Sonnet, whose results are only slightly inferior. The Gemini models perform comparatively worse; notably, Gemini-2.5-Pro exhibits substantially poorer performance than Gemini-2.0-Flash. Overall, open-source models underperform relative to commercial ones, with the two DeepSeek models demonstrating the worst performance.

F. L2 Loss Baseline

While the L2 loss provides a straightforward method to evaluate model prediction performance, it may not fully capture the complexity of driving scenarios, as discussed by [33]. To mitigate this issue, we introduce a simple yet effective baseline: extending the latest AV action unchanged for the next three seconds. Trajectories generated from these constant actions are then evaluated by computing their L2 losses against the ground truth.

Our results indicate that this trivial baseline achieves an average L2 loss of 1.10 m, closely matching the best-performing VLM results from GPT-4o (1.07 m) and Claude 3.7-Sonnet (1.09 m), and significantly outperforming many other models. This comparison highlights the current limitations of zero-shot VLM approaches in trajectory planning tasks, indicating that existing models may struggle to adequately handle driving-specific complexities. Consequently, it emphasizes the need for targeted enhancements, such as designing VLM architectures specifically for driving contexts or fine-tuning models using domain-specific driving datasets.

Despite achieving sub-optimal L2 loss results in the trajectory prediction tasks, it is important to acknowledge that VLMs frequently demonstrate meaningful driving intelligence and often behave differently compared to the simple baseline—though not necessarily better. We further explore this point in the subsequent section.

V. Case Examples

This section discusses six representative scenarios illustrated in Fig. 2. Due to the extensive number of available frames, these examples were carefully selected to highlight typical behaviors rather than providing an exhaustive analysis. Each figure compares trajectories predicted by the VLMs with ground-truth trajectories, serving as illustrative examples rather than exact model outputs. The detailed reasoning processes, trajectory generation rationales, and identified failure modes are presented below.

Case 1: Trajectory Bias from Historical Actions

Fig. 2.1 illustrates a scenario in which the ground-truth trajectory involves driving straight, yet the predicted trajectory erroneously suggests a strong right turn, failing to recognize an obstacle positioned on the right. Although initially counterintuitive, this behavior is consistently observed across all models. It occurs because the AV had just completed a right turn at an intersection immediately preceding this frame. Consequently, the historical actions reflect a pronounced curvature to the right. However, the VLMs struggle to identify the updated road conditions based solely on the current front-view image, mistakenly projecting the preceding turning behavior forward. Notably, shortly after the vehicle resumes a straight path, the models correctly adjust and begin predicting straight trajectories again. Such errors are prevalent among models and occur frequently, not only with right turns but similarly with left turns.

Case 2: Insufficient Context from Visual Cues

Fig. 2.2 demonstrates another scenario in which all models consistently fail. In this case, the ground-truth trajectory involves turning left, yet all models incorrectly predict continuing straight. Although this scenario is inherently challenging—given the absence of explicit left-turn markings on the pavement or dedicated traffic lights—there are still implicit indicators available. For instance, the AV occupies the leftmost lane, whereas vehicles in the adjacent lane to the right are positioned to continue straight. To reliably overcome this issue, models could incorporate additional contextual information, such as explicit navigation instructions clearly indicating a left turn at the intersection.

Case 3 & 4: Divergent Responses to Stop Signals

Fig. 2.3 illustrates a scenario highlighting notable divergences in VLM responses. In this case, the AV gradually approaches a stopped vehicle at an intersection controlled by a red traffic signal. The ground-truth trajectory demonstrates the AV smoothly and progressively decelerating until it reaches a complete stop behind the leading vehicle. However, the VLM predictions diverge into two distinct categories, neither accurately replicating the ground-truth behavior.

The first category typically includes models with relatively lower L2 loss. These models correctly identify the presence of the red traffic signal and the stopped vehicle ahead, as reflected in their scenario descriptions. Nevertheless, they predict an immediate and abrupt braking action instead of the controlled, gradual deceleration observed in reality. This behavior indicates that while these models effectively recognize critical visual cues and associate them with appropriate driving actions, they lack nuanced spatial reasoning based solely on visual input. Consequently, their responses appear event-triggered—instantly reacting to visual signals such as a red light—rather than demonstrating a comprehensive understanding of the developing scenario.

The second category comprises models generally characterized by higher L2 loss. These models inaccurately predict that the AV will continue straight through the intersection without slowing or stopping, effectively ignoring both the stationary vehicle and the red traffic signal. Such predictions reveal fundamental shortcomings in the models' capability to interpret critical visual cues and link them appropriately to driving actions, underscoring significant opportunities for further improvement.

A similar pattern is observed in Fig. 2.4. Here, the ground-truth behavior again involves the AV approaching an intersection with a red traffic signal, where a pedestrian is actively crossing. VLM predictions either anticipate an abrupt emergency stop, despite ample distance available ahead, or entirely overlook the pedestrian and traffic signal, forecasting that the AV will maintain its speed and pass through without decelerating.

Case 5: Divergent Responses to Go Signals

Fig. 2.5 depicts a scenario in which the AV is initially stationary, waiting at an intersection controlled by a traffic

Fig. 2. LightEMMA nuScenes prediction task examples.

signal. Upon the traffic signal changing from red to green, the ground-truth behavior involves the AV promptly initiating acceleration and smoothly traversing the intersection. Models exhibiting lower L2 loss closely replicate this behavior, accurately recognizing the green signal as a clear indicator to proceed and consequently predicting appropriate acceleration trajectories. Conversely, models characterized by higher L2 loss remain stationary, failing to establish the crucial link between the green signal and the corresponding action of accelerating. Their responses are indistinguishable from our simplistic constant-action baseline, highlighting their inability to effectively interpret dynamic visual cues to initiate appropriate vehicle movements.

Case 6: Conflicting Visual Cues and Model Responses

The final example, shown in Fig. 2.6, presents an intriguing scenario where even models demonstrating low L2 loss exhibit differing behaviors. Analogous to the situation in Fig. 2.5, the traffic signal has just transitioned from red to green. One set of models observes the green light and predicts immediate acceleration, disregarding the vehicle directly ahead. Conversely, another group of models accurately recognizes the conflicting cues—acknowledging that despite the green signal, the AV must remain stationary due to the obstructing vehicle. This scenario further extends the observations from Fig. 2.5, highlighting how different VLMs respond when confronted with conflicting visual information.

Furthermore, such divergent responses from VLMs underscore the inherent instability in their decision-making processes when applied to autonomous driving tasks. These inconsistencies can lead directly to hazardous situations, such as unintended acceleration or collision risks, emphasizing the necessity for robust safety mechanisms or guardrails.

VI. Conclusion

In this work, we introduced LightEMMA, a lightweight, end-to-end autonomous driving framework specifically designed for integration with state-of-the-art Vision-Language Models. Using a Chain-of-Thought prompting strategy, we illustrated that VLMs can sometimes accurately interpret complex driving scenarios and produce intelligent responses. Notably, LightEMMA mainly serves as an accessible baseline rather than optimizing performance for specific VLMs.

Systematic evaluations with the nuScenes prediction task assessed dimensions such as computational efficiency, hardware demands, and API costs. Quantitative analyses using L2 loss underscored the limitations of current VLM predictions and highlighted the inadequacy of relying solely on this metric. Qualitative analyses further identified common shortcomings, including over-reliance on historical trajectory data, limited spatial awareness, and reactive decision-making. Consequently, future research should focus on developing driving-specific models or fine-tuning existing VLMs with domain-specific datasets, recognizing the current advantage commercial VLMs hold over open-source alternatives.

References

[1] M. Buehler, K. Iagnemma, and S. Singh, *The DARPA Urban Challenge: Autonomous Vehicles in City Traffic*, vol. 56. Springer Science & Business Media, 2009.

[2] D. Yang, K. Jiang, D. Zhao, C. Yu, Z. Cao, S. Xie, Z. Xiao, X. Jiao, S. Wang, and K. Zhang, "Intelligent and connected vehicles: Current status and future perspectives," *Science China Technological Sciences*, vol. 61, no. 10, pp. 1446–1471, 2018.

[3] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, and H. Li, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[4] B. Wei, M. Ren, W. Zeng, M. Liang, B. Yang, and R. Urtasun, "Perceive, attend, and drive: Learning spatial attention for safe self-driving," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4875–4881, 2021.

[5] 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 (CVPR)*, June 2019.

[6] M. Toromanoff, E. Wirbel, and F. Moutarde, "End-to-end model-free reinforcement learning for urban driving using implicit affordances," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7151–7160, 2020.

[7] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao, "St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning," in *European Conference on Computer Vision (ECCV)*, 2022.

[8] M. Yang, K. Jiang, B. Wijaya, T. Wen, J. Miao, J. Huang, C. Zhong, W. Zhang, H. Chen, and D. Yang, "Review and challenge: High definition map technology for intelligent connected vehicle," *Fundamental Research*, 2024.

[9] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end autonomous driving: Challenges and frontiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[10] H. X. Liu and S. Feng, "Curse of rarity for autonomous vehicles," *Nature Communications*, vol. 15, p. 4808, 2024.

[11] Z. Yang, X. Jia, H. Li, and J. Yan, "Llm4drive: A survey of large language models for autonomous driving," *ArXiv*, vol. abs/2311.01043, 2023.

[12] J.-J. Hwang, R. Xu, H. Lin, W.-C. Hung, J. Ji, K. Choi, D. Huang, T. He, P. Covington, B. Sapp, Y. Zhou, J. Guo, D. Anguelov, and M. Tan, "Emma: End-to-end multimodal model for autonomous driving," 2024.

[13] S. Xing, C. Qian, Y. Wang, H. Hua, K. Tian, Y. Zhou, and Z. Tu, "Openemma: Open-source multimodal model for end-to-end autonomous driving," 2025.

[14] Google, "Gemini: A family of highly capable multimodal models," 2024.

[15] 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," *IEEE Robotics and Automation Letters*, vol. 9, no. 10, pp. 8186–8193, 2024.

[16] Y. Ma, Y. Cao, J. Sun, M. Pavone, and C. Xiao, "Dolphins: Multimodal language model for driving," in *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XLV*, (Berlin, Heidelberg), p. 403–420, Springer-Verlag, 2024.

[17] W. Wang, J. Xie, C. Hu, H. Zou, J. Fan, W. Tong, Y. Wen, S. Wu, H. Deng, Z. Li, *et al.*, "Drivemlm: Aligning multi-modal large language models with behavioral planning states for autonomous driving," *arXiv preprint arXiv:2312.09245*, 2023.

[18] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," 2017.

[19] S. Ettinger, S. Cheng, B. Caine, C. Liu, H. Zhao, S. Pradhan, Y. Chai, B. Sapp, C. R. Qi, Y. Zhou, Z. Yang, A. Chouard, P. Sun, J. Ngiam, V. Vasudevan, A. McCauley, J. Shlens, and D. Anguelov, "Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9710–9719, October 2021.

[20] 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 ," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (Los Alamitos, CA, USA), pp. 11618–11628, IEEE Computer Society, June 2020.

[21] T. Qian, J. Chen, L. Zhuo, Y. Jiao, and Y.-G. Jiang, "Nuscenes-qa: a multi-modal visual question answering benchmark for autonomous driving scenario," in *Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence and Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence and Fourteenth Symposium on Educational Advances in Artificial Intelligence*, AAAI'24/IAAI'24/EAAI'24, AAAI Press, 2024.

[22] D. Wu, W. Han, T. Wang, Y. Liu, X. Zhang, and J. Shen, "Language prompt for autonomous driving," *arXiv preprint*, 2023.

[23] A.-M. Marcu, L. Chen, J. Hünermann, A. Karnsund, B. Hanotte, P. Chidananda, S. Nair, V. Badrinarayanan, A. Kendall, J. Shotton, and O. Sinavski, "Lingoqa: Visual question answering for autonomous driving," *arXiv preprint arXiv:2312.14115*, 2023.

[24] M. Nie, R. Peng, C. Wang, X. Cai, J. Han, H. Xu, and L. Zhang, "Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving," in *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part XXVI*, (Berlin, Heidelberg), p. 292–308, Springer-Verlag, 2024.

[25] OpenAI, "Gpt-4 technical report," 2024.

[26] Anthropic, "The claude 3 model family: Opus, sonnet, haiku," 2024.

[27] DeepSeek-AI, "Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding," 2024.

[28] Meta-AI, "Llama: Open and efficient foundation language models," 2023.

[29] Alibaba, "Qwen2 technical report," 2024.

[30] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, "Learning transferable visual models from natural language supervision," in *Proceedings of the 38th International Conference on Machine Learning* (M. Meila and T. Zhang, eds.), vol. 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763, PMLR, 18–24 Jul 2021.

[31] C. Sun, A. Myers, C. Vondrick, K. Murphy, and C. Schmid, "Videobert: A joint model for video and language representation learning," in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 7464–7473, 2019.

[32] Z. Tong, Y. Song, J. Wang, and L. Wang, "VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training," in *Advances in Neural Information Processing Systems*, 2022.

[33] J.-T. Zhai, Z. Feng, J. Du, Y. Mao, J.-J. Liu, Z. Tan, Y. Zhang, X. Ye, and J. Wang, "Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes," *arXiv preprint arXiv:2305.10430*, 2023.