Poutine: Vision-Language-Trajectory Pre-Training and Reinforcement Learning Post-Training Enable Robust End-to-End Autonomous Driving

Luke Rowe*[1,2], Rodrigue de Schaetzen*[1,2], Roger Girgis[1,3], Christopher Pal[1,2,3,4], Liam Paull[1,2,4]
[1]Mila - Quebec AI Institute, [2]Université de Montréal, [3]Polytechnique Montréal, [4]CIFAR AI Chair

Abstract

We present Poutine, a 3B-parameter vision-language model (VLM) tailored for end-to-end autonomous driving in long-tail driving scenarios. Poutine is trained in two stages. To obtain strong base driving capabilities, we train Poutine-Base in a self-supervised vision–language–trajectory (VLT) next-token prediction fashion on 83 hours of CoVLA nominal driving and 11 hours of Waymo long-tail driving. Accompanying language annotations are auto-generated with a 72B-parameter VLM. Poutine is obtained by fine-tuning Poutine-Base with Group Relative Policy Optimization (GRPO) using less than 500 human preference-labeled frames from the Waymo validation set. We show that both VLT pretraining and RL fine-tuning are critical to attain strong driving performance in the long-tail. Poutine-Base achieves a rater-feedback score (RFS) of 8.12 on the validation set, nearly matching Waymo's expert ground-truth RFS. The final Poutine model achieves an RFS of 7.99 on the official Waymo test set, placing 1st in the 2025 Waymo Vision-Based End-to-End Driving Challenge by a significant margin. These results highlight the promise of scalable VLT pre-training and lightweight RL fine-tuning to enable robust and generalizable autonomy.

1. Introduction

Vision–language models (VLMs) have emerged as a powerful means of coupling visual perception with the world knowledge and common sense reasoning acquired from internet-scale pre-training [2, 5, 7, 12, 15]. For autonomous vehicles, such multimodal reasoning is most valuable in long-tail situations—rare but safety-critical events that dominate operational risk and have limited coverage in conventional driving corpora. Nevertheless, the empirical study of VLMs for driving has so far been restricted largely to nominal driving benchmarks such as nuScenes [4], where high-level semantic reasoning is seldom required and the benefits of language grounding remain unclear [3, 8–11, 13, 14, 16, 19–25]. The Waymo Vision-Based End-to-End Driving (WOD-E2E) dataset, along with its accompanying 2025 challenge, presents a new opportunity to evaluate VLMs in more challenging, curated long-tail scenarios. These characteristics make WOD-E2E a better testbed to investigate whether the knowledge embedded in VLMs can translate into safer and more reliable driving policies.

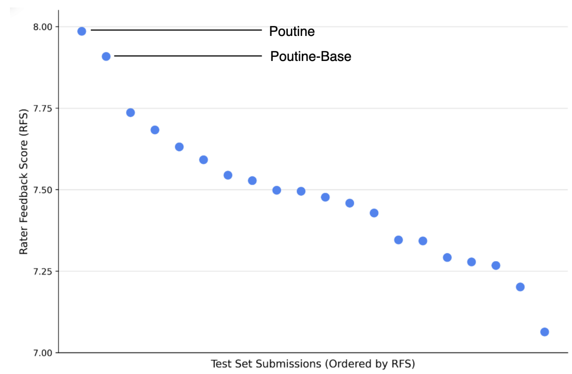

Figure 1. At the end of the submission deadline, Poutine ranked first on the 2025 Waymo Vision-Based E2E Driving Challenge by a considerable margin. Its pre-reinforcement learning variant, Poutine-Base, also outperformed all other submitted entries.

To address this question, we introduce *Poutine*, a 3B-parameter VLM for E2E driving, trained via a simple two-stage pipeline. In Stage 1, *Poutine-Base* learns base driving capabilities by training via next-token prediction over vision, language, and future-trajectory (VLT) tokens. The training corpus comprises 83 hours of nominal Japanese driving from the public CoVLA dataset [1] and 11 hours of long-tail driving from WOD-E2E. All language annotations are generated automatically by a 72B-parameter VLM, removing the need for manual labeling and yielding a fully self-supervised pre-training procedure that is straightforward to scale. In Stage 2, we refine Poutine-Base with Group Relative Policy Optimization (GRPO) using less than 500 human preference-labeled frames from the WOD-E2E validation set. Despite the modest supervision budget, this lightweight reinforcement-learning (RL) step materially improves policy performance in the long-tail.

The trained Poutine-Base model achieves a rater-

*Equal contribution.

feedback-score (RFS) of 8.12 on the WOD-E2E validation set, nearly matching the 8.13 RFS of Waymo's expert ground-truth trajectories. Moreover, a variant trained solely on the CoVLA Japanese driving dataset generalizes zero-shot to U.S. driving data, achieving an RFS of 7.74 on the WOD-E2E validation set despite never encountering Waymo data during pre-training. This result highlights the potential of collecting diverse driving data from various geographical regions to train a unified driving policy via large-scale next-token prediction VLT pre-training. After GRPO fine-tuning, the final Poutine model scores 7.99 RFS on the official WOD-E2E test set, achieving first place in the 2025 Waymo Vision-Based End-to-End Driving Challenge by a large margin (see Figure 1). Collectively, our results indicate that scalable VLT pre-training, coupled with lightweight preference-based RL, constitutes a practical recipe for robust and generalizable autonomy in challenging long-tail driving scenarios.

2. Poutine

2.1. Language Annotation Dataset Collection

Language annotations were automatically generated on the WOD-E2E and CoVLA driving datasets using a pretrained `Qwen2.5-VL 72B Instruct` model[1]. This procedure required no manual labelling or post-hoc verification, thereby eliminating human effort and improving the scalability of our approach. Each prompt supplied the VLM with three camera frames over the last second, the high-level driving command (*intent*) at the current frame, and the 4-second past trajectory. In contrast to most driving datasets with language and action data, we also conditioned on the ground-truth 5-second future trajectory of the ego vehicle. The resulting captions therefore explain why the vehicle executes the given future trajectory rather than predict where the ego should go based on the past context. We found that this significantly improves the consistency between the language annotations and ground-truth future trajectories.

We divided the language annotation task into three components: 1) identifying the relevant critical objects from a predefined list, 2) composing a short description explaining why the expert trajectory was executed, and 3) identifying the meta behavior from a predefined list of possible speed and command actions. Preliminary testing showed that the annotation quality dramatically improves when the critical object detection task is structured as yes or no questions compared to more open-ended descriptions of the task. The system prompt is shown in Figure 3. We generated 240.5 K high-quality captions for the WOD-E2E (training set) and CoVLA datasets (sampled at 1 Hz), using 12 A100 GPUs over 24 h. To accelerate VLM inference, front-view images from the current frame were resized to 532×476 px, while all other images were limited to at most 256×256 px. Figure 2 shows three examples of our automatically generated annotations.

2.2. Vision-Language-Trajectory Pre-Training

The training of Poutine-Base was completed in two stages: VLT pre-training on CoVLA followed by VLT pre-training on the WOD-E2E training data. Both stages are trained via standard next-token prediction. We adopt the `Qwen2.5-VL 3B Instruct` model without modification, operating exclusively on language and image tokens. Several design decisions were made to facilitate alignment between the image, language, and trajectory modalities and to effectively predict the trajectory autoregressively.

For pre-training on CoVLA, we preserved the annotation configuration of three consecutive frames from the preceding second, whereas on WOD-E2E we restricted visual input to the three front-facing cameras of the current frame; although omitting side and rear views reduces contextual coverage, we found the front images are sufficient for nearly all scenarios and their smaller token count enabled substantially more optimization steps within our compute budget. To discourage shortcut learning, we also removed the four-second past trajectory and intent during CoVLA pre-training, forcing the model to derive useful representations from the higher-dimensional image–caption signal rather than relying on these lower-dimensional cues.

The collected annotation dataset (Section 2.1) allows us to recast trajectory planning as a four-stage chain-of-thought (CoT) reasoning sequence: (i) detection of critical objects and conditions, (ii) generation of a natural-language explanation, (iii) meta-behaviour selection, and (iv) prediction of the future path. The model executes four successive question–answer turns, each stage conditioning on the previous response. To reduce compounding error, we predict only five 1 Hz waypoints and then upsample them to 4 Hz using cubic-spline interpolation, rather than predicting the full 5 s trajectory at 4 Hz directly. Figure 4 shows our system prompt for VLT pre-training. To fully exploit the available data for training, frames without captions are still used for the trajectory prediction task, and the training set is subsampled so that captions accompany half of the selected frames. In total, we used ≈10% of CoVLA frames and 20 % of WOD-E2E frames for model training, uniformly subsampled across the scenarios.

2.3. Reinforcement Learning Post-Training

The final stage of Poutine fine-tunes Poutine-Base with RL on the 479 human preference-labeled frames in the WOD-E2E validation set. In this stage, we fine-tune only on the future trajectory prediction task and omit the structured CoT reasoning tasks. This significantly improves infer-

[1] The original CoVLA captions lacked semantic diversity which prompted us to generate new captions.

The expert driver plans a left turn, navigating around **nearby vehicles** parked along the street. A **conflicting vehicle** approaches from the left, requiring the driver to adjust the trajectory to avoid collision. The **traffic element**, a stop sign, necessitates a cautious approach. The driver's future trajectory carefully maneuvers to **yield** to the oncoming vehicle while ensuring a safe **left turn**.

The expert driver plans to **stop** due to a **nearby vehicle** with an open door, a **conflicting vehicle** approaching from the left, and a stop sign. The trajectory **halts** to ensure safety, avoiding potential collisions with the **door-opening vehicle** and respecting the **traffic element**.

The expert driver plans a **straight** trajectory while avoiding a **cyclist** on the right side of the road. The cyclist's presence necessitates maintaining a safe distance, ensuring the vehicle does not encroach on their path. The driver's steady course ahead ensures safety and compliance with the road layout.

Figure 2. **Generated annotations on WOD-E2E data.** The red dots depict the 5-second future trajectory. Only two views from the current frame and the explanation of the annotation are shown. Bold text highlights the objects and meta behavior selected by the model.

```
You are an expert labeller of driving scenarios.
Input:
- 3 frames of multi-view images collected from the ego-vehicle over the last 1 second
- Current high-level intent (string)
- 4-second past trajectory (16 steps at 4 Hz)
- Expert 5-second future trajectory (20 steps at 4 Hz)
Task:
1. Inspect the input and decide, for each object class below, whether at least one critical
instance of that class is present (i.e., it materially affects the ego-vehicle's future trajectory
). A vehicle can be a car, bus, truck, motorcyclist, scooter, etc. traffic_element includes
traffic signs and traffic lights. road_hazard may include hazardous road conditions, road debris,
obstacles, etc. A conflicting_vehicle is a vehicle that may potentially conflict with the ego's
future path.
   Object classes to audit:
     - nearby_vehicle
     - pedestrian
     - cyclist
     - construction
     - traffic_element
     - weather_condition
     - road_hazard
     - emergency_vehicle
     - animal
     - special_vehicle
     - conflicting_vehicle
     - door_opening_vehicle
2. Output "yes" or "no" for every class (no omissions).
3. Compose a concise natural-language description explaining why the expert safe driver plans the
given future trajectory.
     - Mention only the classes you marked "yes"
     - Describe how each of those critical objects or conditions influences the trajectory.
     - Do not invent objects or conditions not present in the input.
4. From the expert's 5-second future trajectory, assign exactly one category from each list:
     - speed ∈ { keep, accelerate, decelerate }
     - command ∈ { straight, yield, left_turn, right_turn, lane_follow, lane_change_left,
lane_change_right, reverse }
   Choose the label that best summarises the overall behaviour of the expert future trajectory.
     - If none fits, use 'other', but do this sparingly.
Output format (strict JSON, no extra keys, no commentary):
{
  "critical_objects": {
    "nearby_vehicle": "yes | no",
    "pedestrian": "yes | no",
    "cyclist": "yes | no",
    "construction": "yes | no",
    "traffic_element": "yes | no",
    "weather_condition": "yes | no",
    "road_hazard": "yes | no",
    "emergency_vehicle": "yes | no",
    "animal": "yes | no",
    "special_vehicle": "yes | no",
    "conflicting_vehicle": "yes | no",
    "door_opening_vehicle": "yes | no"
  },
  "explanation": "100-word description that references only the classes marked 'yes'",
  "meta_behaviour": {
    "speed": "keep | accelerate | decelerate | other",
    "command": "straight | yield | left_turn | right_turn | lane_follow | lane_change_left |
lane_change_right | reverse | other"
  }
}
```

Figure 3. **System prompt for generating language annotations.** We changed 'driving scenarios' to '*left-hand-side* driving scenarios' and 'multi-view images' to '*front-view* images' for CoVLA.

ence latency without significantly impacting performance, as shown in Section 3. We use the preference-labeled trajectories provided in the WOD-E2E validation data along with their corresponding rater feedback labels to compute the RFS as a reward signal. As such, our reward functions consist of the normalized RFS reward and a format reward for outputting the future trajectory in the correct format. We fine-tune Poutine-Base with GRPO [17], selected for its recent success in fine-tuning LLMs and VLMs [6, 18].

```
You are an expert driver.
Input:
- 1 frame of multi-view images collected from the ego-vehicle at the present timestep
- Current high-level intent (string)
- 4-second past trajectory (16 steps at 4 Hz)
Task 1: Critical Objects and Conditions Detection
Decide whether at least one critical instance of each class could influence the ego-vehicle's
future path (no omissions). A vehicle can be a car, bus, truck, motorcyclist, scooter, etc.
traffic_element includes traffic signs and traffic lights. road_hazard may include hazardous road
conditions, road debris, obstacles, etc. A conflicting_vehicle is a vehicle that may potentially
conflict with the ego's future path. Output "yes" or "no" for every class (no omissions).
   Object classes to audit:
     - nearby_vehicle
     - pedestrian
     - cyclist
     - construction
     - traffic_element
     - weather_condition
     - road_hazard
     - emergency_vehicle
     - animal
     - special_vehicle
     - conflicting_vehicle
     - door_opening_vehicle
Output format (strict JSON, no extra keys, no commentary):
{
  "critical_objects": {
    "nearby_vehicle": "yes | no",
    "pedestrian": "yes | no",
    "cyclist": "yes | no",
    "construction": "yes | no",
    "traffic_element": "yes | no",
    "weather_condition": "yes | no",
    "road_hazard": "yes | no",
    "emergency_vehicle": "yes | no",
    "animal": "yes | no",
    "special_vehicle": "yes | no",
    "conflicting_vehicle": "yes | no",
    "door_opening_vehicle": "yes | no"
  }
}
Task 2: Natural Language Explanation
Compose a concise natural-language description of the optimal future 5-second trajectory for the
ego vehicle that the expert driver (you) plans and explain why the expert driver plans to execute
this trajectory.
    - Mention only the classes you marked "yes" in the previous task.
    - Describe how each of those critical objects or conditions influences the optimal trajectory.
    - Do not invent objects or conditions not present in the input.
Output format (strict JSON, no extra keys, no commentary):
{
  "explanation": "100-word description that references only the classes marked 'yes'"
}
Task 3: Meta-Behaviour Selection
Assign exactly one category from each list. Choose the label that best summarises the overall
behaviour of the optimal future trajectory:
     - speed ∈ { keep, accelerate, decelerate }
     - command ∈ { straight, yield, left_turn, right_turn, lane_follow, lane_change_left,
lane_change_right, reverse }
     - If none fits, use 'other', but do this sparingly.
Output format (strict JSON, no extra keys, no commentary):
{
  "meta_behaviour": {
    "speed": "keep | accelerate | decelerate | other",
    "command": "straight | yield | left_turn | right_turn | lane_follow | lane_change_left |
lane_change_right | reverse | other"
  }
}
Task 4: Future Trajectory Prediction
Given the input, critical objects/conditions, natural language explanation, and meta-behaviour,
predict the optimal 5-second future trajectory (5 steps at 1 Hz) of the ego vehicle.
Output format (raw text, not markdown or LaTeX):
[x_1, y_1], [x_2, y_2], [x_3, y_3], [x_4, y_4], [x_5, y_5]
```

Figure 4. **System prompt used for VLT pre-training.** The prompt was adjusted accordingly for pre-training on CoVLA and for frames that did not contain language annotations.

3. Results

Datasets For VLT pre-training we leveraged the public CoVLA dataset, which contains 10,000 front-view, 30s driving videos recorded in Japan at 20 Hz with correspond-

Method	VLT Pre-Training		CoT		RFS[†] (Avg) ↑
	CoVLA	WOD-E2E	Train	Test	
Qwen 2.5-VL 3B Instruct	✗	✗	✗	✗	5.59
Qwen 2.5-VL 7B Instruct	✗	✗	✗	✗	5.40
Poutine-Base CoVLA	✓	✗	✓	✗	7.74
Poutine-Base No-CoVLA	✗	✓	✓	✗	7.95
Poutine-Base No Language	✓	✓	✗	✗	7.94
Poutine-Base	✓	✓	✓	✗	**8.12**
Poutine-Base CoT	✓	✓	✓	✓	<u>8.08</u>
Ground-Truth	–	–	–	–	8.13

Table 1. **Results on WOD-E2E validation split with different pre-training and inference strategies.** Best-performing model **bolded** and second-best <u>underlined</u>. † RFS range: 4–10.

ing ego trajectory information. Raw images (1928×1208 px) are down-sampled to a max of 512×512 px, and for each frame we store the future 5s ego trajectory sub-sampled to 4 Hz to be consistent with the WOD-E2E format. The WOD-E2E dataset provides 4021 long-tail driving scenarios of 20s each, captured at 10 Hz from 8 multi-view cameras. Each scenario belongs to one of a pre-defined set of long-tail categories, such as pedestrians, intersections, cut-ins, and special vehicles. The official split allocates 2037 videos for training, 479 for validation, and 1505 for testing. Each frame contains a 4 s ego-vehicle past trajectory and a 5 s ground-truth future trajectory, both sampled at 4 Hz.

Evaluation Metrics One frame per validation and test scenario contains three human expert-rated 5 s trajectories each scored in $[0, 10]$; these rated trajectories are used to compute the RFS, which constitutes the primary evaluation metric for the challenge. The RFS for a predicted future trajectory is computed by first constructing a trust region around each of the three rated trajectories. If the prediction lies within the trust region of its nearest rated trajectory, it is assigned that trajectory's score. Otherwise, the assigned score is exponentially lower than that of the closest rated trajectory's score, with a floor RFS of 4. The final RFS is calculated as the mean of the per-scenario-category scores.

Implementation Details For CoVLA VLT pre-training, we fine-tuned all modules of the `Qwen-2.5-VL-3B-Instruct` model for one epoch with an effective batch size of 64 and a learning rate of 1e-5 under a cosine decay schedule, completing in 24 h on four NVIDIA A100 GPUs. WOD-E2E VLT pre-training used identical hyper-parameters except for two epochs and a reduced batch size of 16, finishing in 10 h on the same hardware. RL post-training then optimized Poutine-base with GRPO for 2,000 steps, employing a linear-decay schedule from 1e-6, sampling temperature 0.9, $\beta = 0.04$, 8 rollouts per sample, and an effective batch size of 32. This stage required 12 h on four A100 GPUs. At inference, we used a 1e-6 temperature for the CoT and greedy decoding for trajectory prediction. We removed the intent from conditioning at inference, which we found to marginally improve performance.

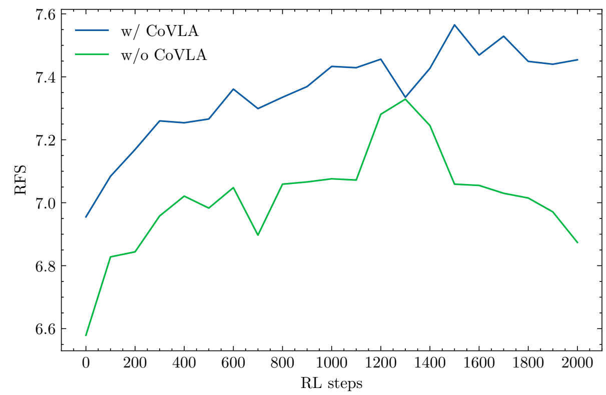

Figure 5. **GRPO Results.** Comparison between RL on a model pretrained with (blue) versus without (green) CoVLA. Both models were pre-trained on WOD-E2E. Checkpoints were evaluated on held-out test set of 63 examples from the WOD-E2E val split.

Results Figure 1 ranks the final test-set submissions to the 2025 Waymo Vision-Based E2E Driving Challenge. *Poutine* leads by a wide margin with an RFS of 7.99, while its pre-RL variant (*Poutine-Base*, 7.91) places second, underlining the value of GRPO post-training. Table 1 analyzes pre-training ablations. Models that skip VLT pre-training (i.e., the two Qwen baselines) trail all Poutine variants, confirming the necessity of domain-specific pre-training. Training on CoVLA alone (*Poutine-Base CoVLA*) transfers zero-shot to Waymo with an RFS of 7.74, demonstrating strong cross-domain generalization. Removing CoVLA data (*Poutine-Base No-CoVLA*) or the auto-generated captions (*Poutine-Base No Language*) from training degrades performance to 7.95 and 7.94, respectively, showing that both large-scale pre-training and language supervision are beneficial. Consistent with prior works [10, 19], generating CoT at inference (*Poutine-Base CoT*) does not improve over the no-CoT *Poutine-Base* variant. Further investigation is required to determine the benefit of CoT for reasoning in long-tail scenarios at inference time. The final *Poutine-Base* model achieves an 8.12 RFS, nearly matching the 8.13 RFS attained with the expert WOD-E2E trajectories.

Figure 5 shows the results of applying GRPO to the models pre-trained with and without CoVLA data for 2,000 steps on 416 of the 479 WOD-E2E preference-labeled validation examples, with the remaining 63 held-out for evaluation purposes. Both models clearly benefit from RL post-training; however, the model pre-trained with CoVLA achieves significantly higher RFS, illustrating that large-scale VLT pre-training provides a stronger foundation for subsequent RL optimization.

Conclusion Poutine combines VLT pre-training with GRPO fine-tuning to yield a 3B-parameter VLM that sets a new state-of-the-art on the Waymo Vision-Based E2E Driving benchmark. Its simple and scalable design highlights a practical path towards robust long-tail driving autonomy.

Acknowledgment

We thank Samsung, the IVADO and the Canada First Research Excellence Fund (CFREF) / Apogée Funds, the Canada CIFAR AI Chairs Program, and the NSERC Discovery Grants program for financial support. We also thank Mila - Quebec AI Institute for compute resources.

References

[1] Hidehisa Arai, Keita Miwa, Kento Sasaki, Kohei Watanabe, Yu Yamaguchi, Shunsuke Aoki, and Issei Yamamoto. Covla: Comprehensive vision-language-action dataset for autonomous driving. In *WACV*, 2025. 1

[2] Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv*, 2023. 1

[3] Yifan Bai, Dongming Wu, Yingfei Liu, Fan Jia, Weixin Mao, Ziheng Zhang, Yucheng Zhao, Jianbing Shen, Xing Wei, Tiancai Wang, and Xiangyu Zhang. Is a 3d-tokenized LLM the key to reliable autonomous driving? *arXiv*, 2024. 1

[4] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 1

[5] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *arXiv*, 2023. 1

[6] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv*, 2025. 3

[7] Gemini Team Google. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv*, 2024. 1

[8] Deepti Hegde, Rajeev Yasarla, Hong Cai, Shizhong Han, Apratim Bhattacharyya, Shweta Mahajan, Litian Liu, Risheek Garrepalli, Vishal M. Patel, and Fatih Porikli. Distilling multi-modal large language models for autonomous driving. *arXiv*, 2025. 1

[9] Zhijian Huang, Tao Tang, Shaoxiang Chen, Sihao Lin, Zequn Jie, Lin Ma, Guangrun Wang, and Xiaodan Liang. Making large language models better planners with reasoning-decision alignment. In *ECCV*, 2024.

[10] Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, James Guo, Dragomir Anguelov, and Mingxing Tan. EMMA: end-to-end multimodal model for autonomous driving. *arXiv*, 2024. 4

[11] Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *arXiv*, 2024. 1

[12] AI @ Meta Llama Team. The llama 3 herd of models. *arXiv*, 2024. 1

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

[14] Ming Nie, Renyuan Peng, Chunwei Wang, Xinyue Cai, Jianhua Han, Hang Xu, and Li Zhang. Reason2drive: Towards interpretable and chain-based reasoning for autonomous driving. In *ECCV*, 2024. 1

[15] OpenAI. GPT-4 technical report. *arXiv*, 2023. 1

[16] Chenbin Pan, Burhaneddin Yaman, Tommaso Nesti, Abhirup Mallik, Alessandro Gabriele Allievi, Senem Velipasalar, and Liu Ren. VLP: vision language planning for autonomous driving. In *CVPR*, 2024. 1

[17] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv*, 2024. 3

[18] Haozhan Shen, Peng Liu, Jingcheng Li, Chunxin Fang, Yibo Ma, Jiajia Liao, Qiaoli Shen, Zilun Zhang, Kangjia Zhao, Qianqian Zhang, et al. Vlm-r1: A stable and generalizable r1-style large vision-language model. *arXiv*, 2025. 3

[19] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *ECCV*, 2024. 1, 4

[20] Thomas Tian, Boyi Li, Xinshuo Weng, Yuxiao Chen, Edward Schmerling, Yue Wang, Boris Ivanovic, and Marco Pavone. Tokenize the world into object-level knowledge to address long-tail events in autonomous driving. In *CoRL*, 2024.

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

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

[23] Shuo Xing, Chengyuan Qian, Yuping Wang, Hongyuan Hua, Kexin Tian, Yang Zhou, and Zhengzhong Tu. Openemma: Open-source multimodal model for end-to-end autonomous driving. *arXiv*, 2024.

[24] Yi Xu, Yuxin Hu, Zaiwei Zhang, Gregory P. Meyer, Siva Karthik Mustikovela, Siddhartha Srinivasa, Eric M. Wolff, and Xin Huang. VLM-AD: end-to-end autonomous driving through vision-language model supervision. *arXiv*, 2024.

[25] Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, and Alois C. Knoll. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. *arXiv*, 2025. 1