IRL-VLA: Vision-Language-Action Training via Reward World Model

Supplementary Material

1. Appendix

We organize the supplementary material as follows. First, in Sec.2 we discuss IRL-VLA's main contribution and broader impacts. In Sec.3, we provides details of the evaluation metrics used on NAVSIM and describes IRL-VLA's implementation much more details for image backbone, semantic reasoning and 3d reasoning, including all key hyperparameters. Sec.4 we presents additional qualitative results, including extensive visualizations on NAVSIM and Bench2Drive. In Sec.5 we discuss the limitation of the IRL-VLA.

2. Main Contribution and Broader Impacts

The framework of VLA. We propose a VLA architecture that simultaneously incorporates the scenario generalization capability of LLM-based methods [3, 14] and the strong 3D understanding capacity [1, 2, 11, 12] of E2E models. Compared to existing end-to-end (E2E) solutions on NavSim[2, 11, 12] , our work extends a more robust and comprehensive BEV representation[5] in engineering practice. This is achieved by simultaneously integrating learning tasks such as semantic segmentation, vectorized object detection, and high-precision vector map reconstruction.

Why use RL? The VLA[3] and end-to-end[2, 12] approaches for imitation learning primarily focus on minimizing the discrepancy between predicted trajectories and ground truth trajectories. However, these methods generally exhibit suboptimal performance in closed-loop evaluation. Among existing closed-loop evaluation approaches on NavSim, state-of-the-art algorithms are predominantly scorer-based methods[10, 11]. These approaches predict multiple trajectories along with their corresponding scores and employ rule-based selection mechanisms to choose the optimal trajectory based on these scores. To achieve higher scores, these methods typically need to predict a larger number of trajectories, thereby expanding the selection space. In contrast, our proposed IRL-VLA leverages reinforcement learning to enable the model to directly learn the optimal policy, rather than merely imitating historical trajectories or pursuing high-scoring trajectories. This allows our method to achieve superior performance during inference with only a minimal number of candidate trajectories.

Why use Reward World Model? When employing reinforcement learning, it is necessary to compute the reward for trajectories predicted by the policy in real time. Existing approaches typically feed the predicted trajectories directly into a simulator, which then calculates the corresponding rewards—a methodology known as RLVR (Reinforcement Learning with Verifiable Rewards)[13]. However, this simulation-based reward computation requires significant time overhead and necessitates a heterogeneous computing pipeline to balance the time costs between RL sampling and simulator execution. This complexity in pipeline design leads to challenges in cluster management for large-scale applications. Our proposed Reward World Model offers a distinct solution by adopting the RLAIF (Reinforcement Learning with AI Feedback) paradigm[7]. This approach involves training a reward world model on large-scale datasets, enabling the direct use of this model to obtain rewards during the reinforcement learning process. By decoupling the complex environment setup from the reinforcement learning framework, our method achieves greater simplicity and efficiency in large-scale applications.

In summary, our work successfully streamlines a complex autonomous driving reinforcement learning architecture—which originally relied on simulators and heterogeneous distributed computing—into an elegant and unified reinforcement learning framework consisting of a policy action model and a reward world model.

3. Implementation Details

3.1. Image backbone

We use an energy efficient backbone for real-time VoVNetv2-99[8, 9] for our image backbone. The input camera is set to a resolution ofof 256×704.

3.2. Semantic reasoning

To enhance the processing and fusion efficiency of multimodal information in autonomous driving scenarios, we propose the VLM Command Guidance module (incorporating multi-class speed-planning and path-planning commands) to optimize the traffic efficiency of autonomous driving models. This module is built upon the Senna-VLM[6] framework, which employs a multi-image encoding strategy and multi-view prompting mechanisms to achieve efficient and comprehensive scene understanding. The details of VLM Command are shown below:

Speed Plan Command: Emergency Decel, Full Stop, Creeping, Regenerative Braking, Controlled Decel, Mild Decel Linear, Mild Decel, Mild Accel, Mild Accel NonLinear, Mild Accel Linear, Aggressive Accel, Constant Speed Loose, Constant Speed Strict

Path Plan Command: Swerving, Sharp Left Turn, Slight Left Turn, Sharp Right Turn, Slight Right Turn, Straight Strict, Left LaneChange, Lane Micro Adjust, Right LaneChange

3.3. 3D reasoning

To enhance 3D reasoning in autonomous driving scenarios, we train 3d object detection, hd map construction and bev segmentation in BEV representation. And use this BEV feature, 3d object embedding and hd map embedding as conditions in diffusion process. The BEV grid size is set to 128×128, covering a perception range of 64×64 meters along the x and y directions in the ego coordinate space. We aggregate features from 30 agents and one ego vehicle to provide implicit guidance for the subsequent trajectory diffusion process. Additionally, the explicit outputs of objects and maps enable the planner to perform collision detection and drivable area checks, enhancing the planning process beyond feature-based trajectory selection alone.

3.4. NAVSIMV2 Metrics

The NAVSIMV2 Navhard benchmark provides a non-reactive simulation environment and employs the Extend Predictive Driver Model Score (EPDMS) as its close-loop planning metric:

$$\text{EPDMS} = \underbrace{\prod_{m \in \mathcal{M}_{\text{pen}}} \text{filter}_m(\text{agent}, \text{human})}_{\text{penalty terms}} \cdot \underbrace{\frac{\sum_{m \in \mathcal{M}_{\text{avg}}} w_m \cdot \text{filter}_m(\text{agent}, \text{human})}{\sum_{m \in \mathcal{M}_{\text{avg}}} w_m}}_{\text{weighted average terms}} \quad (1)$$

where EPDMS integrates two sub-metrics group: $\mathcal{M}_{\text{pen}} = \{\text{NC}, \text{DAC}, \text{DDC}, \text{TLC}\}$ and $\mathcal{M}_{\text{avg}} = \{\text{TTC}, \text{EP}, \text{HC}, \text{LK}, \text{EC}\}$. No At-Fault Collision (NC), Drivable Area Compliance (DAC), Driving Direction Compliance (DDC), Lane Keeping(LK),Time-to-Collision (TTC), History Comfort (HC), Extended Comfort(EC), Traffic Light Compl. (TLC) and Ego Progress (EP) to produce a comprehensive closed-loop planning score.

3.5. Bench2Drive Metrics

We summarize here the evaluation metrics used in the Bench2Drive benchmark[4], which are adopted in our experiments. These metrics are designed for closed-loop end-to-end autonomous driving and capture task success, rule compliance, efficiency, and ride comfort.

Success Rate measures whether the agent can reach the destination within the time limit while strictly following traffic rules. A route is counted as successful only if no infractions occur and the full mission is completed.

Driving Score extends the official CARLA driving score by jointly considering route completion and penalties for different types of infractions. It reflects the agent's ability to progress toward the goal while maintaining safe and lawful behavior.

Driving Efficiency evaluates whether the ego vehicle maintains an appropriate speed relative to surrounding traffic. The score is computed at multiple checkpoints distributed along the route and reflects the agent's capability to drive efficiently rather than overly conservatively.

Comfortness assesses driving smoothness based on acceleration, jerk, yaw dynamics, and other indicators of ride quality. Bench2Drive evaluates smoothness on short trajectory segments, ensuring stability against isolated outlier frames. This metric reflects passenger comfort and the naturalness of the agent's motion.

4. More Qualitative Results

4.1. Compare with other SOTA

In the Figure.1, we present different stages of the Navsimv2 navhard dataset: stage 1 contains real-world data, while stage 2 consists of synthetic scenario data. Overall, we compared with IRL-VLA-IL, the reinforcement-learning–driven IRL-VLA achieves higher traffic efficiency and improved safety.

Specifically, in stage 1, Figures 2 and 4, the trajectory of IRL-VLA-IL nearly comes to a stop, whereas IRL-VLA successfully passes through. In stage 1, Figures 1 and 3, IRL-VLA produces trajectories that are more comfortable and efficient than those generated by imitation-learning–based methods.

For the synthetic scenarios, in stage 2, Figures 1, 2, and 4, IRL-VLA again demonstrates higher traffic efficiency than the imitation learning baselines. In stage 2, Figure 3, the lead vehicle is stationary; the trajectory from IRL-VLA-IL leads to a collision (due to similar non-synthetic scenarios in which the lead vehicle is moving), while IRL-VLA safely avoids the situation.

4.2. Failure Cases

In the Figure.2, we also present IRL-VLA failure cases on Navsim benchmark. In the failure cases, we still observe several problematic situations, such as unsafe driving or violations of traffic rules. In Figure 1, IRL-VLA-IL drives in the wrong direction, while IRL-VLA runs onto the curb. In other images, the ego vehicle is positioned in the oncoming lane in the synthetic data, and none of the trajectories return successfully to the correct lane.

5. Limitation

Although IRL-VLA demonstrates SOTA performance on the NAVSIMv2 navhard benchmark, several limitations should be acknowledged for completeness. 1) Although the RWM removes the need for a high-fidelity simulator, it relies on rule-based EPDMS metrics as supervision signals. These metrics reflect human-designed heuristics rather than real human preferences, which may limit alignment

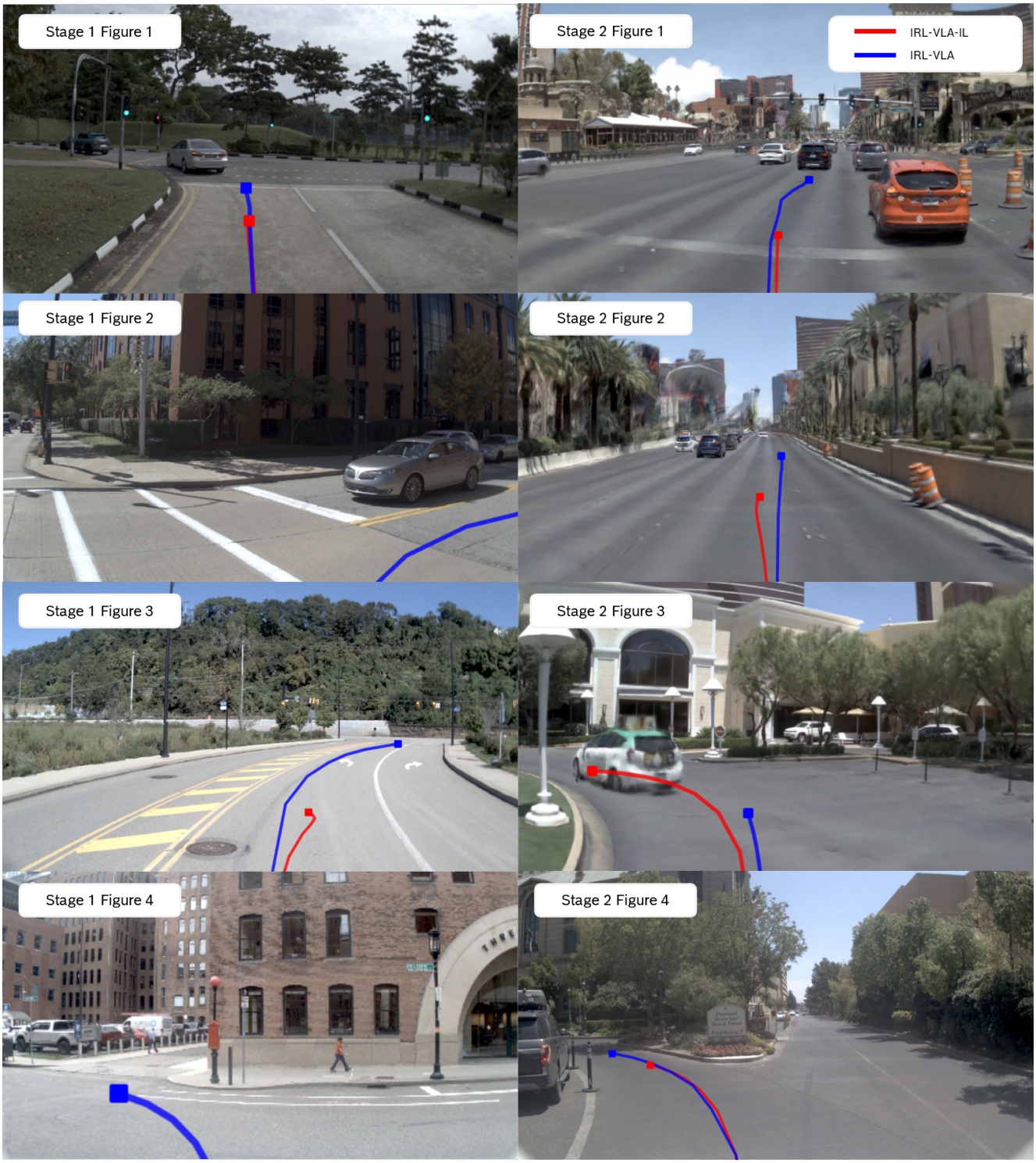

Figure 1. **Comparisons on the NAVSIMV2 Navhard benchmark.**

with human comfort and social norms. Incorporating human feedback or preference-based rewards is left for future work. 2) While simulation-free RL improves scalability, the lack of environment interaction also restricts the policy's ability to recover from compounding errors or explore novel emergent behaviors. The RWM approximates reward but does not model dynamic environment transitions, which means IRL-VLA still inherits part of imitation learning's

Figure 2. **Failure cases of IRL-VLA on the NAVSIMV2 Navhard benchmark.**

covariate shift.

References

[1] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024. 1

[2] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2022. 1

[3] Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, et al. Emma: End-to-end multimodal model for autonomous driving. *arXiv preprint arXiv:2410.23262*, 2024. 1

[4] Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. *Advances in Neural Information Processing Systems*, 37:819–844, 2024. 2

[5] Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. *arXiv preprint arXiv:2503.07656*, 2025. 1

[6] 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 preprint arXiv:2410.22313*, 2024. 1

[7] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. 2023. 1

[8] Youngwan Lee and Jongyoul Park. Centermask: Real-time anchor-free instance segmentation. 2020. 1

[9] Youngwan Lee, Joong-won Hwang, Sangrok Lee, Yuseok Bae, and Jongyoul Park. An energy and gpu-computation efficient backbone network for real-time object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 0–0, 2019. 1

[10] Zhenxin Li, Kailin Li, Shihao Wang, Shiyi Lan, Zhiding Yu, Yishen Ji, Zhiqi Li, Ziyue Zhu, Jan Kautz, Zuxuan Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024. 1

[11] Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Joshua Chen, Nadine Chang, Maying Shen, Zuxuan Wu, Shiyi Lan, and Jose M Alvarez. Generalized trajectory scoring for end-to-end multimodal planning. *arXiv preprint arXiv:2506.06664*, 2025. 1

[12] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion

model for end-to-end autonomous driving. *arXiv preprint arXiv:2411.15139*, 2024. 1

[13] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024. 1

[14] 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 preprint arXiv:2412.15208*, 2024. 1