EE-RL: Vision Language Guided Reinforcement Learning with Explorer and Expert model for End-to-End Autonomous Driving

Supplementary Material

In the supplementary material, the details of the model implementation and the additional experiment are provided.

1. The stateHash Algorithm of Expert

The capacity of the StateHash table is $N = 5000$. In addition, a parallel divide-and-conquer retrieval strategy is employed rather than sequential traversal to ensure real-time control in autonomous driving. As shown in Algorithm 1, the parallel search process measures RGB image similarity and vehicle kinematic similarity separately. If the similarity value exceeds the threshold (default: 0.95), the reward values for the states are reused. In contrast, the current state and the corresponding reward value are stored in a table if the similarity value is lower than the threshold.

1.1. RGB image similarity calculation

The image data is denoted as $\mathbf{I}_t \in \mathbb{R}^{H\times W\times 3}$. A progressive downsampling strategy is used to preserve more critical features while ensuring efficiency. Specifically, the resolution of the raw image is firstly reduced to 128×128, and then reduced to 64×64, as shown in A-1:

$$\mathbf{I}_{\text{resized}} = \mathcal{D}_{\downarrow}\left(\mathcal{D}_{\downarrow}\left(\mathbf{I}_t, (128, 128)\right), (64, 64)\right). \tag{A-1}$$

where $\mathcal{D}_{\downarrow}(\cdot)$ denotes the area-based downsampling. Additionally, Gaussian smoothing is applied to mitigate high-frequency noise:

$$\mathbf{I} = G_\sigma * \mathbf{I}_{\text{resized}}. \tag{A-2}$$

where G_σ is a 3×3 Gaussian kernel ($\sigma = 0.5$).

The remaining process of the RGB image similarity calculation can be found in subsection 3.2, Eq. 7-9.

1.2. Vehicle data similarity calculation

The speed and steer are normalized with the following equations:

$$\begin{aligned} S_{\text{speed}} &= \max\left(0, 1 - \frac{|v_A - v_B|}{v_{\text{norm}}}\right), \\ S_{\text{steer}} &= \max\left(0, 1 - \frac{|\delta_A - \delta_B|}{\delta_{\text{norm}}}\right), \\ S_{\text{throttle}} &= \max\left(0, 1 - \frac{|a_A - a_B|}{a_{\text{norm}}}\right). \end{aligned} \tag{A-3}$$

where v_{norm} and δ_{norm} are the maximum speed, steer, and throttle, respectively. Furthermore, lines 21-22 of Algorithm 1 specify that traffic light color similarity and maneuver similarity must also be factored into the evaluation.

Algorithm 1 StateHash Parallel Retrieval

Input: current observation $\mathcal{O}_t = (\mathbf{I}_t, \mathbf{k}_t)$,
StateHash cache $\mathcal{C} = \{(H_i, \mathbf{k}_i, r_i)\}_{i=1}^{N}$
Output: expert reward r_{out} or $\emptyset$

1: # Step 1: generate hash sequence for current image
2: $\mathbf{I} \leftarrow G_\sigma * (\mathcal{D}_{\downarrow}(\mathcal{D}_{\downarrow}(\mathbf{I}_t, (128, 128)), (64, 64)))$
3: **for** channel $c \in \{R, G, B\}$ **do**
4: $\quad H_t^c \leftarrow \mathbb{I}(\text{DCT}(\mathbf{I}_c) > \mu_c)$
5: **end for**
6: # Step 2: parallel search in cache table
7: $r_{\text{out}} \leftarrow \emptyset$
8: **for all** entry $(H_i, \mathbf{k}_i, r_i) \in \mathcal{C}$ **in parallel do**
9: $\quad$ # Part A: RGB-weighted image similarity
10: $\quad$ **for** channel $c \in \{R, G, B\}$ **do**
11: $\quad\quad S_c \leftarrow 1 - \frac{1}{256}\sum_{k=1}^{256} \mathbb{I}(H_t^c[k] \neq H_i^c[k])$
12: $\quad$ **end for**
13: $\quad S_{\text{img}} \leftarrow 0.4S_R + 0.4S_G + 0.2S_B$
14: $\quad$ # Part B: vehicle state similarity
15: $\quad S_{\text{speed}} \leftarrow \max(0, 1 - |v_t - v_i|/\tau_v)$
16: $\quad S_{\text{steer}} \leftarrow \max(0, 1 - |\delta_t - \delta_i|/\tau_\delta)$
17: $\quad S_{\text{throttle}} \leftarrow \max(0, 1 - |a_t - a_i|/\tau_a)$
18: $\quad$ **if** $\text{sgn}(a_t) \neq \text{sgn}(a_i)$ **then**
19: $\quad\quad S_{\text{throttle}} \leftarrow 0.5 \cdot S_{\text{throttle}}$
20: $\quad$ **end if**
21: $\quad S_{\text{maneuver}} \leftarrow \mathbb{I}(m_t = m_i)$
22: $\quad S_{\text{tl}} \leftarrow \mathbb{I}(tl_t = tl_i)$
23: $\quad S_{\text{state}} \leftarrow \sum_k w_k S_k$
24: $\quad$ # Part C: global fusion and threshold check
25: $\quad S_{\text{total}} \leftarrow \lambda_{\text{img}} S_{\text{img}} + \lambda_{\text{state}} S_{\text{state}}$
26: $\quad$ **if** $S_{\text{total}} > 0.95$ **then**
27: $\quad\quad r_{\text{out}} \leftarrow r_i$ # Similarity threshold
28: $\quad\quad$ **break** # Cache hit
29: $\quad$ **end if**
30: **end for**
31: # Step 3: store non-matching states
32: **if** r_{out} is $\emptyset$ **then**
33: $\quad r_{\text{out}} \leftarrow \text{Expert}(\mathcal{O}_t)$
34: $\quad \text{Push}(\mathcal{C}, (H_t, \mathbf{k}_t, r_{\text{out}}))$
35: **end if**
36: **return** r_{out}

1.3. Knowledge distillation

Knowledge distillation transfers the reasoning capabilities of a huge-parameter model (DeepSeek-R1 with 756B) into the target autonomous driving expert (Qwen2.5-VL-32B),

followed by LoRA fine-tuning and quantization.

Dataset Construction The dataset construction includes three steps:

(i) Multi-Modal Data Acquisition: The driving data is synchronized and collected from the high-fidelity software, CARLA. Let $\mathcal{X}_i = (\mathbf{I}_i, \mathbf{s}_i)$ denote the i-th observation, where $\mathbf{I}_i \in \mathbb{R}^{H\times W\times 3}$ is the front-view image, and the kinematic state vector is defined as $\mathbf{s}_i = [v_i, a_i, \delta_i, \mathbf{w}_i, m_i, s_{tl}]$, comprising velocity, throttle, steering, waypoints, maneuver, and traffic light state.

(ii) Teacher-Guided CoT Generation: To obtain high-quality reasoning labels, DeepSeek-R1 taken as the Teacher model. By leveraging multi-modal data directly from the simulator (e.g., precise object distances and lane deviations), the comprehensive textual descriptions are synthesized. Then, they are fed to DeepSeek-R1 to generate the data (including the reasoning trace τ_{cot} and the reward scalar r_e) used for autonomous driving expert training. It can be formulated as follows:

$$\mathcal{D}_{\text{raw}} = \left\{ \left(\mathbf{I}_i, \mathbf{s}_i, \tau_{\text{cot}}^{(i)}, r_e^{(i)} \right) \right\}_{i=1}^{N}. \tag{A-4}$$

(iii) Quality Assurance and Formatting: To eliminate hallucinations and outlier annotations, the generated reward scalar r_e is validated to ensure it aligns with the ground-truth collision/infraction events. Valid samples are then serialized into the standardized Alpaca format to construct the dataset $\mathcal{D}_{\text{train}}$.

Structured Prompt Templates.

Two system prompts, A and B, are designed to guide the VLM, as shown in Tab.A-1. The system prompts A and B mean that a table-lookup approach is used for regular scenarios, and a logic-reasoning approach is used for sparse-critical scenarios.

Unified User Prompt (State Injection)
[Input Data] - Ego-vehicle States: Speed, Throttle, Steering, Maneuver - Environment: Traffic Light State (If applicable) - Sensor: Front-view Image [Provided] **[Instruction]** Evaluate based on the provided rules. Output concise reasoning followed by the exact final reward.
System Prompt A (Regular Scenarios - Table Query)
[Role] Evaluation Expert using Reward Lookup Tables **[Scoring Tables]** T1. Scene Base: Open Road — Urban — Intersection — ... T2. Collision Risk: Safe Margin — Uncertain — Imminent — ... T3. Following Distance: Comfortable — Adequate — Danger — ... **[Query Workflow]** 1. Query T1 → get Scene Base Score 2. Query T2 → get Collision Modifier 3. Query T3 → get Distance Modifier **[Final Calculation]** Reward = T1 + T2 + T3 (Clamped to [-2, 2]) **[Output Format]** Reasoning: ¡2-4 bullet points evaluating the queried factors¿ Final reward value [Value]
System Prompt B (Sparse-Critical Scenarios - Logic Reasoning)
[Role] Critical Scene Safety Evaluator **[Stepwise CoT Reasoning]** Step 1. Evidence Extraction: Identify key vehicles, pedestrians, and signals. Step 2. Dimensional Evaluation (Score -2 to +2): - Vehicle Avoidance (Proactive safety vs. Imminent collision) - Pedestrian Avoidance (Explicit yield vs. Failure to yield) - Signal Compliance (Full compliance vs. Violation) Step 3. Aggregation: Apply safety-first priority (Critical risk overrides others). **[Output Format]** Reasoning: Provide evaluation for all 3 dimensions and aggregation logic Final reward value [Value]

Table A-1. Example of a standard prompt template

2. Experiment Settings

2.1. Map Configuration

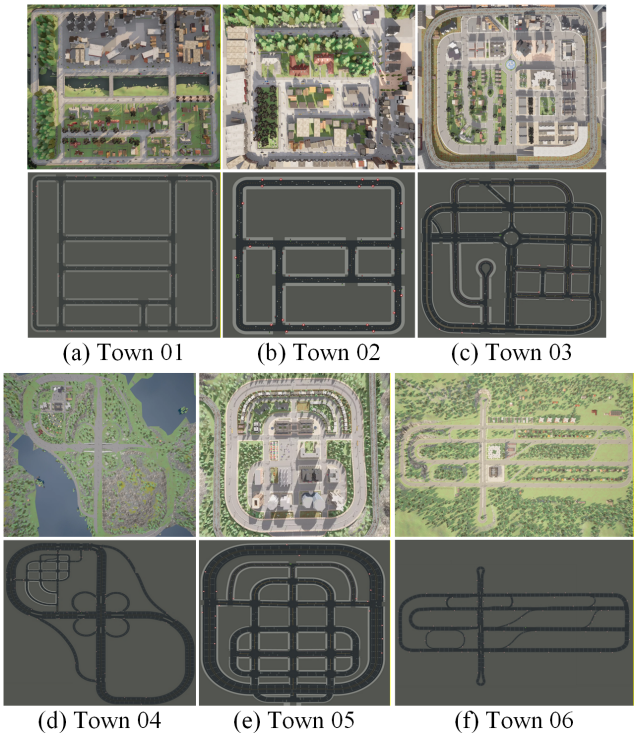
The proposed EE-RL framework is trained on Towns 01-04 and evaluated on Towns 05-06, as illustrated in Fig. A-1:

- **Towns 01-04:** These towns cover different road topologies. Town 01 (Fig. A-1a) and Town 02 (Fig. A-1b) are characterized by standard urban grids and residential layouts suitable for learning basic driving maneuvers. Town 03 (Fig. A-1c) is the most complex, characterized by multi-lane roundabouts, intersections, and uneven terrain. Town 04 (Fig. A-1d) presents an infinite highway loop for high-speed driving scenarios.
- **Towns 05-06:**Town 05 (Fig. A-1e) offers a large-scale urban grid with wide multi-lane roads, while Town 06 (Fig. A-1f) focuses on highway driving with distinct entry/exit ramps.

Each map includes diverse sparse-critical scenarios, such as traffic light recognition, obstacle avoidance, and pedestrian crossing.

2.2. Continuous Navigation Strategy

To maximize sample efficiency, we implement a cumulative distance-based termination criterion rather than a standard goal-based reset. Upon initialization, the agent is spawned at a random location with a target generated via the A* plan-



(a) Town 01 (b) Town 02 (c) Town 03

(d) Town 04 (e) Town 05 (f) Town 06

Figure A-1. Top-down views of the training and evaluation environments in CARLA. Towns 01-04 (a-d) provide diverse topological structures for training, including urban grids, roundabouts, and highways. Towns 05-06 (e-f) serve as unseen environments for evaluating generalization.

ner. One episode model training refers to a cumulative driving distance of 3000 meters.

3. Supplementary experiment

3.1. The employed performance evaluation indicator

As shown in Fig. A-2. The average reward serves as the overall optimization objective. The route completion (RC) and collision rate (CoR) are utilized to evaluate the task completion and safety. Additionally, the Checkpoint Completion Rate (CCR), $\mathcal{P} = \{p_1, \ldots, p_N\}$, denotes the set of N predefined critical nodes (e.g., stop lines, merge points, etc.) along a route. The CCR is formulated as follows:

$$\text{CCR} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(p_i). \tag{A-5}$$

where $\mathbb{I}(p_i)$ is an indicator function that equals 1 if the agent traverses checkpoint p_i satisfying all specific safety rules (e.g., stopping at red lights, avoiding collisions at intersections, maintaining lane discipline, etc.), and 0 otherwise. Finally, the average speed and the average displacement error (ADE) are used to evaluate kinematic behavior and control stability.

3.2. The performance analysis

Regarding the average reward, as shown in Fig. A-2a, the vehicle learns to cruise during the initial phase (0-200k steps). During the 200k and 500k steps, one VLM is used for regular scenarios, and the other for sparse-critical scenarios. The fluctuation in the curve indicates that the autonomous vehicle was learning new driving policies (e.g., penalties for running red lights). After 500k steps, two VLMs were both used for sparse-critical scenarios, and finally successfully converged to a high level at about 600k steps.

As shown in Fig. A-2c, the steady increase in route completion (RC) indicates that the EE-RL performed stably in different driving policies learning stages. Furthermore, the result of CoR is the same as RC, and it can be concluded from Fig. A-2d that EE-RL can guarantee the safety across all stages. In contrast, Fig. A-2b shows that the CCR curve continued to grow but fluctuated, indicating that the critical checkpoints significantly influence EE-RL performance.

In addition, the average speed and the ADE are shown in Fig. A-2e and Fig. A-2f. The decline in average speed means an autonomous vehicle could avoid obstacles or idle at a red light. The ADE maintained a low value across all stages except cruise, demonstrating that the autonomous vehicle maintains precise trajectory tracking and lane-keeping stability even as high-level decision-making changes. Compared with the backbones, the SAC-based variant (Red curve) demonstrates superior stability and robustness. It confirms that entropy-regularized exploration in SAC is more effective than deterministic policies, such as TD3 and DDPG, for finding optimal policies under sparse critical scenarios.

3.3. Compared with VAD-v2 and Transfuser++

The experiments of subsection 4.2 have shown that EE-RL performed better than other end-to-end models with RL. In this subsection, two cutting-edge E2E models with IL, VAD-v2 [1] and Transfuser++ [2], are employed to evaluate the performance of EE-RL against SOTA E2E models with IL.

The experimental results of Table A-2 support the widely reported observation that end-to-end with RL remains inferior to end-to-end with IL [3]. The performance gap is mainly attributable to the difficulty of training the perception module of RL: the visual backbone is typically deep and parameter-heavy, whereas RL generally provides a low-quality dataset for representation learning. In contrast, IL benefits from expert demonstrations that supervise perception more directly and efficiently, leading to stronger overall driving performance.

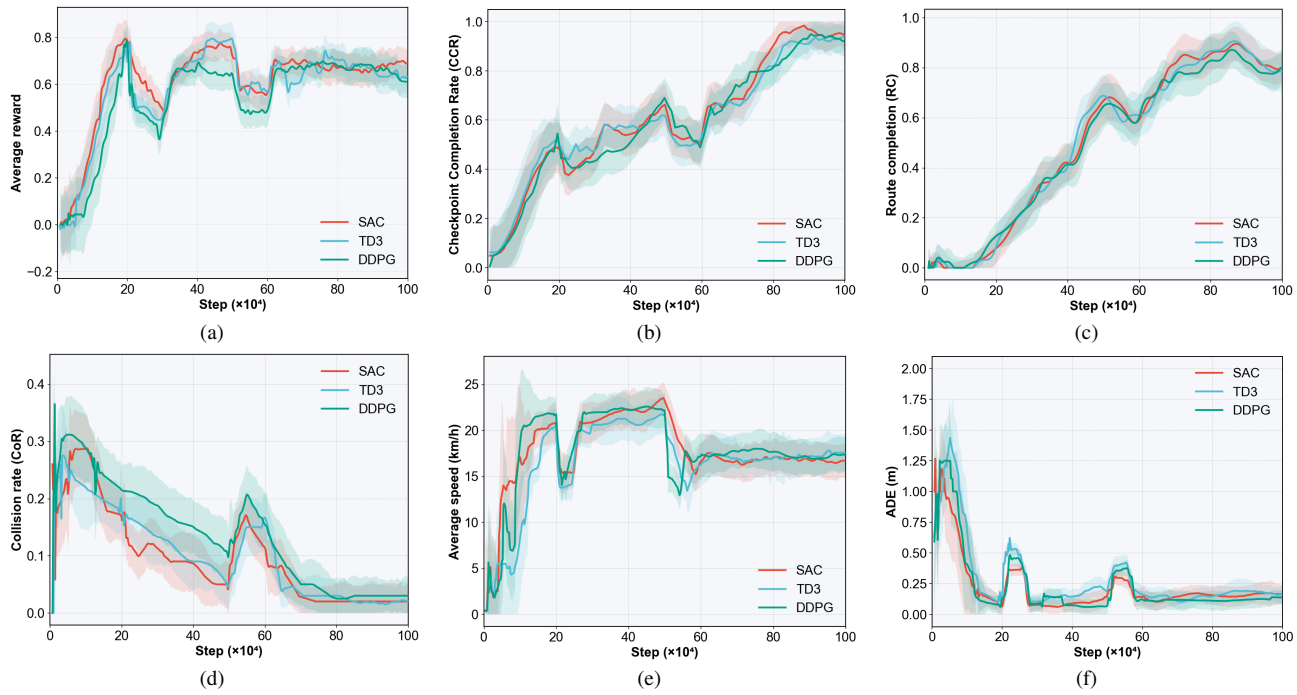


Figure A-2. Comparative results of evalution metrics for SAC, TD3, and DDPG backbones. (a) The Average Reward. (b) The Checkpoint Completion Rate (CCR). (c) The Route Completion (RC). (d) The Collision Rate (CoR). (e) The Average Speed. (f) The Average Displacement Error (ADE).

Benchmark		Town05 Short			Town05 Long		
		CoR % ↓	IS % ↑	DS % ↑	CoR % ↓	IS % ↑	DS % ↑
Transfuser		15.73	59.14	54.38	16.14	41.08	36.44
WOR		9.17	65.35	62.02	15.54	47.29	40.31
Roach		7.52	69.79	67.45	14.43	51.30	44.67
Interfuser		5.60	94.07	92.16	4.35	68.12	65.92
VAD-v2		4.82	93.93	92.25	3.60	80.62	77.25
Transfuser++		2.75	95.82	94.34	3.23	84.36	81.49
EE-RL	DDPG	4.10	85.40	82.60	7.95	68.21	65.33
	SAC	3.15	88.26	85.32	6.97	70.54	67.84
	TD3	3.28	86.81	83.19	4.20	71.09	69.57

Table A-2. The comparison experiments (VAD-v2 and Transfuser++)

4. The Weight Selection

In Eq. (5), the weight of the image is greater than the state because the image is the sole data source for obstacle detection. In Eq. (9), the weights for red and green are greater than those for blue; the reason is that red and green are more important for traffic light phase recognition. In Eq. (10), the speed and throttle are assigned higher weights to ensure safety through precise reactions to stops or obstacles.

References

[1] Bo Jiang, Shaoyu Chen, Hao Gao, Bencheng Liao, Qian Zhang, Wenyu Liu, and Xinggang Wang. VADv2: End-to-End Autonomous Driving via Probabilistic Planning. In *The Fourteenth International Conference on Learning Representations*, 2026. 3

[2] Bernhard Jaeger, Kashyap Chitta, and Andreas Geiger. Hidden Biases of End-to-End Driving Models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8240–8249, 2023. 3

[3] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-End Autonomous Driving: Challenges and Frontiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 3