SGDrive: Scene-to-Goal Hierarchical World Cognition for Autonomous Driving

Supplementary Material

7. Discussion

D1. Research Motivation.

Existing VLM-based autonomous driving methods rely on the model's general reasoning capabilities to interpret the entire scene. However, as language-oriented models, VLMs lack inherent spatial perception and deep understanding of the driving environment. Consequently, while they can describe driving behaviors well, their planning performance often lags behind traditional end-to-end models. Inspired by human driving behavior, we observe that drivers naturally filter out background information irrelevant to the current driving task. Their attention focuses not on static entities or exhaustive scene details, but on dynamic and functional information that directly influences their actions. Importantly, human attention is anticipatory: cognitive resources are concentrated on upcoming regions to assess potential state distributions and feasible actions, guiding subsequent decisions and short-term goals. Motivated by this, we propose a scene-agent-goal framework SGDrive that enables the VLM to focus on driving-relevant driving knowledge and predict its future evolution, enabling safer and more efficient driving.

D2. Key contributions of our SGDrive.

SGDrive introduces several novel mechanisms to enhance trajectory planning via hierarchical driving-world knowledge: (i) specialized ⟨world⟩ query tokens that represent distinct levels of driving knowledge, including scene geometry, safety-critical agents, and future goal points. These hierarchical supervisory objectives that activate the model's ability to predict both current and future world states; (iii) a structured attention mask that disentangles interactions between different knowledge levels, preventing information leakage and cross-level interference; and (iv) a diffusion transformer whose trajectory generation is modulated by these hierarchical representations. Collectively, these contributions enable the model to leverage rich, predictive driving-world knowledge for safer and more effective planning.

D3. Limitations and future work.

Our work primarily focuses on leveraging hierarchical driving knowledge to activate a VLM's capability for world knowledge prediction, thereby enhancing driving safety. However, as shown in the Table 1, our framework is currently limited to supervised fine-tuning (SFT), with limited exploration of reinforcement learning fine-tuning (RFT). Although our method can be seamlessly integrated with existing reinforcement learning (RL) strategies and achieve impressive performance, the absence of an RL scheme tailored to our hierarchical design still constrains driving efficiency. We plan to explore integrating reinforcement learning with our SGDrive framework in future work to further enhance driving performance. In addition, due to token budget constraints, our current system operates on front-view inputs only. Future work will extend our hierarchical world knowledge to multi-view settings to further improve robustness and driving safety.

D4. Why does applying the formula in Eq. 8 to the averaged sub-metrics in Table 1 not reproduce the reported PDMS value?

This discrepancy arises from the official NAVSIM evaluation protocol. As clarified in the NAVSIM [9] and ReCogDrive [30] documentation, the PDMS score is computed at the scene level: for each scene, all sub-metrics are first combined using Eq. 8, and the resulting PDMS values are then averaged across all scenes. In contrast, applying Eq. 8 to the globally averaged submetrics in Table 1 corresponds to a different computation combining metrics after averaging so the resulting value does not match the official PDMS score. Note that the same reasoning also applies to Eq. 9 and Table 5.

8. More experiments

8.1. Results with reinforce learning on NVASIM benchmark

Although the core objective of our method is to learn and forecast hierarchical driving-world knowledge to enhance

Table 5. Performance comparison on Navtest Benchmark with extended metrics.

Method	NC↑	DAC↑	EP↑	TTC↑	C↑	TL↑	DDC↑	LK↑	EC↑	EPDMS↑
Transfuser [43]	97.7	92.8	79.2	92.8	**100**	99.9	98.3	67.6	95.3	77.8
VADv2 [6]	97.3	91.7	77.6	92.7	**100**	99.9	98.2	66.0	97.4	76.6
Hydra-MDP [32]	97.5	96.3	80.1	93.0	**100**	99.9	98.3	65.5	97.4	79.8
Hydra-MDP++ [32]	97.9	**96.5**	79.2	93.4	**100**	**100.0**	98.9	67.2	**97.7**	80.6
ARTEMIS [11]	98.3	95.1	81.5	97.4	**100**	99.8	98.6	96.5	98.3	83.1
ReCogDrive-8B [30]	98.3	95.2	**87.1**	97.5	98.3	99.8	**99.5**	**96.6**	86.5	83.6
SGDrive-2B (ours)	**98.6**	94.3	86.0	**97.9**	98.3	99.9	**99.5**	96.1	85.9	**86.2**

driving safety, it can also be seamlessly integrated with existing RL frameworks. Under the same RL training configuration as RecogDrive [30], our approach achieves substantially better results, as shown in Table 1. Simply incorporating our structured world knowledge features into the RL pipeline yields a PDMS of 91.1, outperforming all existing methods, including those using Lidar-signal inputs.

Compared with other RL-based approaches, our model achieves best performance on NC and DAC, demonstrating that the learned driving-world knowledge effectively reduces collision risk and ensures compliance with drivable regions—both essential for safe autonomous driving. In future work, we plan to explore RL algorithms specifically tailored to our hierarchical world knowledge forecast framework to further improve driving efficiency and smoothness.

8.2. Results on the NAVSIM benchmark with extended metrics

To comprehensively evaluate our approach, we follow prior work [32] and adopt the Extended PDMS metric on the NAVSIM [9] benchmark. As shown in Table 5, SGDrive achieves the best overall performance with an EPDMS of 86.2, outperforming the previous state-of-the-art ReCogDrive-8B by 2.6 points. Our method also delivers the strongest results on the safety-critical NC and TTC metrics, while maintaining competitive performance on the newly introduced TL, LK, and EC metrics. These results collectively demonstrate the effectiveness and robustness of SGDrive in modeling driving-relevant world knowledge under the extended evaluation protocol.

8.3. Results on the Bench2Drive benchmark

As shown in Table 6, our SGDrive achieves the best performance on both Driving Score (75.47) and Success Rate (51.36) among all compared approaches in the CARLA Bench2Drive closed-loop benchmark. It outperforms the second-best method ReCogDrive by 4.11 points in Driving Score and 5.91 percentage points in Success Rate, demonstrating superior driving reliability and task completion capability. Compared with earlier methods such as TCP and VAD, SGDrive exhibits substantial improvements in both core metrics, further validating its effectiveness in complex autonomous driving scenarios.

Table 6. Close-loop results in CARLA Bench2Drive Leaderboard.

Method	Closed-loop	
	Driving Score ↑	Success Rate (%) ↑
TCP [60]	40.70	15.00
TCP-ctrl [60]	30.47	7.27
TCP-traj [60]	59.90	30.00
ThinkTwice [19]	62.44	31.23
DriveAdapter [18]	64.12	33.08
VAD [21]	42.35	15.00
UniAD-Tiny [14]	40.73	13.18
UniAD-Base [14]	45.81	16.36
ReCogDrive [30]	71.36	45.45
SGDrive (Ours)	**75.47**	**51.36**

Table 7. Comparison of hidden state fusion methods in diffusion planner.

Exp.	NC↑	TTC↑	EP↑	PDMS↑
(a)	98.2	95.0	80.6	87.1
(b)	98.1	95.1	79.7	86.9
(c)	**98.6**	**95.4**	**81.2**	**87.4**

8.4. Comparison of hidden state fusion methods in diffusion planner

As shown in Table 7, we compare several strategies for fusing hidden states within the diffusion planner. Exp. (a) incrementally injects the hidden states of different subqueries across successive cross-attention layers. Exp. (b) assigns distinct cross-attention layers to different subqueries. Exp. (c), which corresponds to our proposed design, concatenates all subquery hidden states and enables interaction at every cross-attention layer. All fusion strategies achieve strong performance, confirming that our subqueries encode rich driving-world knowledge and can effectively guide the trajectory generation process.

8.5. Implementation and metric details

Implementation. In the first stage, we fine-tune the VLM on the aggregated driving QA dataset for three epochs using a batch size of 1024. During this stage, we keep the vision encoder frozen and update only the language and cross-modal fusion modules. We use the AdamW optimizer with a base learning rate of 4×10^{-5}, a weight decay of 0.05, and a cosine learning rate schedule preceded by a 10% linear warmup.

In the second stage, we train the diffusion-based planner using behavior cloning for 220 epochs with a batch size of 512. We again use AdamW, warming the learning rate up to 1×10^{-4} during the first 1.5% of training steps and then decaying it to 1×10^{-6} following a cosine schedule. Throughout this stage, we apply a weight decay of 1×10^{-4}. We determine the number of subqueries based on the perceptual field and the minimum capacity required for reliable reasoning. Accordingly, we allocate 625 queries for geometric scene layout, 50 queries for safety-critical agent detection, and 1 query for driving-goal forecasting.

Our diffusion planner adopts a DiT-style architecture that follows the design used in ReCogDrive [30]. The model alternates between self-attention, which captures pairwise waypoint relations, and cross-attention, which injects our subqueries' priors into the trajectory space, enabling an effective fusion of scene understanding and trajectory optimization.

NAVSIM metric. NAVSIM [9] scores driving agents in two steps. First, subscores in range $[0, 1]$ are computed after simulation. Second, these subscores are aggregated into the PDM Score (PDMS) $\in [0, 1]$. We use the following aggregation of subscores based on the official definition:

$$\text{PDMS} = \underbrace{\left(\prod_{m \in \{\text{NC,DAC}\}} \texttt{score}_m \right)}_{\text{penalties}} \times \underbrace{\left(\frac{\sum_{w \in \{\text{EP,TTC,C}\}} \texttt{weight}_w \times \texttt{score}_w}{\sum_{w \in \{\text{EP,TTC,C}\}} \texttt{weight}_w} \right)}_{\text{weighted average}}. \tag{8}$$

Subscores are categorized by their importance as penalties or terms in a weighted average. A penalty punishes inadmissible behavior such as collisions with a factor < 1. The weighted average aggregates subscores for other objectives such as progress and comfort.

NAVSIM metric with extended PDMS. Hydra-MDP++ [32] extends the original PDMS metric by incorporating additional aspects of driving performance, including Traffic Lights Compliance (TL), Lane Keeping Ability (LK), and Extended Comfort (EC), providing a more comprehensive evaluation of a method's effectiveness. Formally, the Extended PDM Score (EPDMS) is computed as:

$$\text{EPDMS} = \underbrace{\prod_{m \in \{\text{NC,DAC,DDC,TL}\}} S^m}_{\text{penalty terms}} \times \underbrace{\frac{\sum_{w \in \{\text{EP,TTC,C,LK,EC}\}} \text{weight}_w \cdot S^w}{\sum_{w \in \{\text{EP,TTC,C,LK,EC}\}} \text{weight}_w}}_{\text{weighted average of positive indicators}}. \tag{9}$$

Here, the first term accumulates multiplicative penalties for safety-critical violations, while the second term computes a weighted average over positive performance indicators, providing a balanced assessment of driving quality and comfort.

9. Additional visualizations

9.1. Qualitative results

We provide additional qualitative results in Figure 7 and Figure 8. For both straight-driving and turning scenarios, our predicted trajectories closely follow the ground truth. We also include several failure cases.

9.2. Failure cases

As illustrated in Figure 9, when relying solely on a single front-view image, the model may exhibit slight deviations under extreme turning conditions. In such scenarios, due to the absence of corresponding viewpoints, accurately predicting long-horizon trajectories becomes challenging, sometimes leading to lane-change errors. Incorporating multi-view inputs is a promising direction to mitigate these limitations in future work.

Figure 7. Qualitative results on the Navtest benchmark.

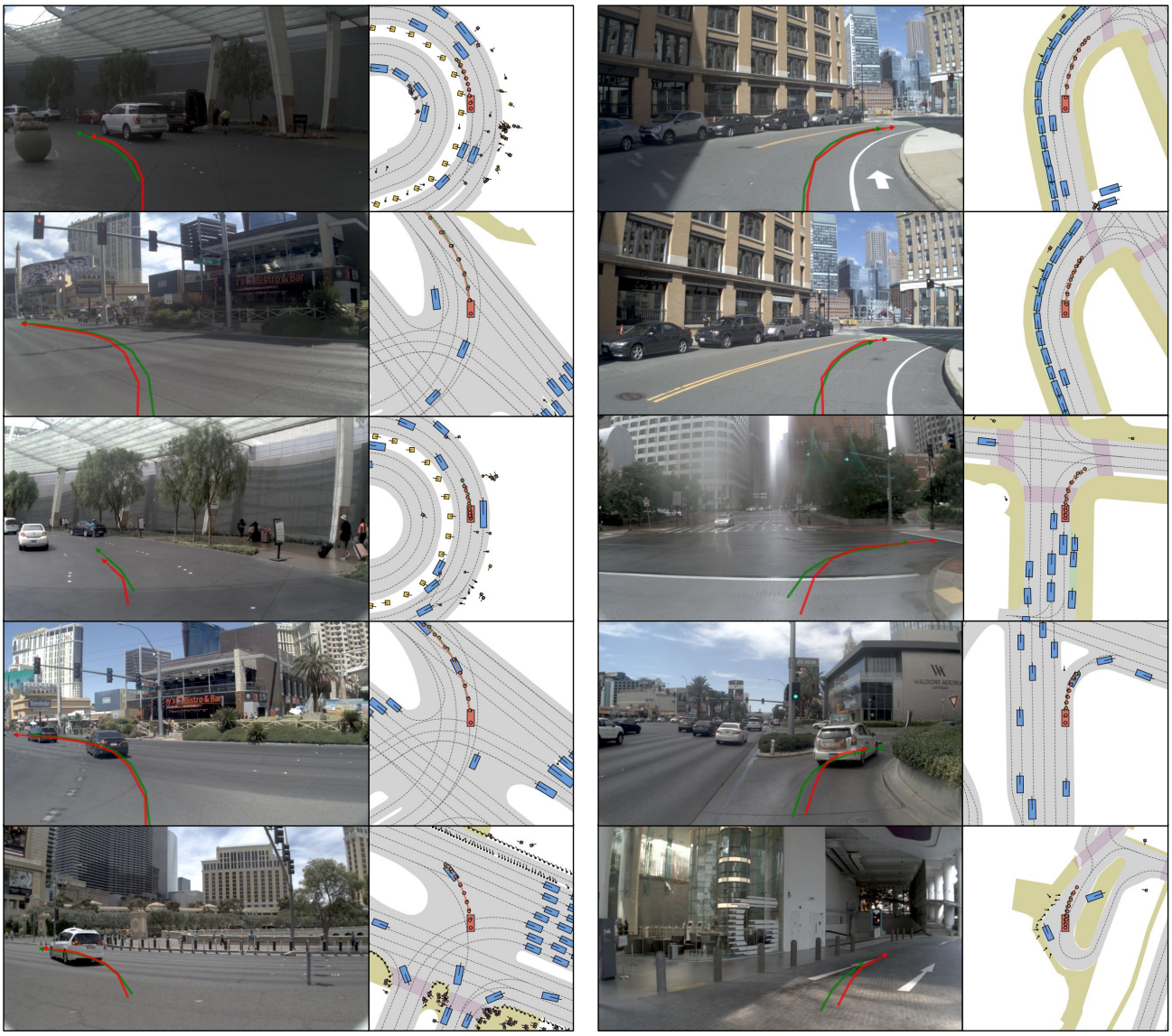
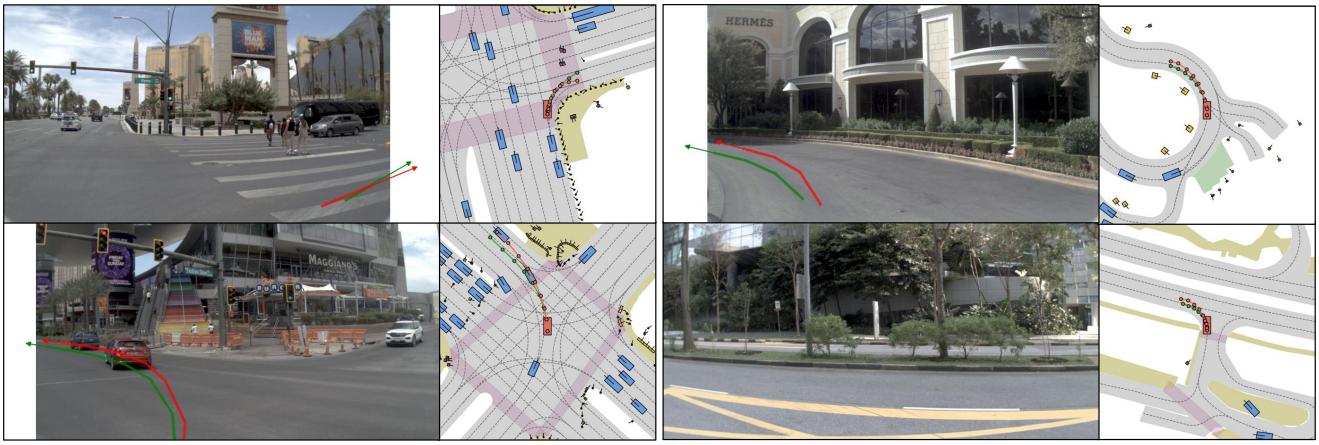


Figure 8. Qualitative results on the Navtest benchmark.



Failure cases

Figure 9. Qualitative analysis of representative failure cases on the Navtest benchmark.