nature communications

Article

https://doi.org/10.1038/s41467-026-72030-6

Autonomous driving system based on dual process theory and deliberate practice theory

Received: 1 January 2025

Accepted: 2 April 2026

Published online: 22 April 2026

Check for updates

Xiao Zhang[1], Tianyu Hu[1] ✉, Juntao Lyu[1], Qiankun Liu[1], Qianchuan Zhao[2], Mingyi Li[1], Kang Li[1], Songde Han[1], Jiansheng Chen[1] & Huimin Ma[1] ✉

Autonomous driving, despite significant progress, is still not widely applied in open, unconstrained environments, primarily owing to deficiencies in hazard perception, few-shot generalization, corner case generalization, and evaluation metrics, resulting in reliability concerns. To address these challenges, we propose CogniDrive, a framework based on dual-process and deliberate practice theories, leveraging contextual reasoning of the Large Language Model (LLM) to enhance driving systems robustness and generalization. Inspired by dual-process theory, CogniDrive comprises two cognition modes: InstinctNav for rapid, intuitive decision-making and ReflectPlan for reflective reasoning. Enhanced by a thought model and experience embedding for LLM, InstinctNav combines behavioral cloning and retrieval augmented generation to enhance few-shot learning efficiency based on deliberate practice theory. ReflectPlan processes and internalizes reward signals embedded in language tokens within the prompt, derived from a self-reflection mechanism, to enable continuous improvement and generalization. To detect hazards in corner cases precluded by limited training data distribution, a vision language model is integrated for comprehensive environmental understanding through multimodal self-reflection. We further propose an evaluation framework that complements traditional metrics by emphasizing safety, comfort, and energy efficiency, and demonstrate state-of-the-art performance through extensive open-loop and closed-loop experiments.

Autonomous driving (AD) is a pivotal advancement in the evolution of transportation, potentially increasing safety, efficiency, and convenience. Current AD approaches include modular[1–3] and end-to-end methods[4–6], focusing primarily on mapping visual features to control signals[7]. Despite their efficiency in controlled settings, these methods fail to generalize to open-world scenarios because of their inability to handle long-tailed distribution outside the training data[8,9]. Unlike humans, who can adapt to new and unseen situations, these systems lack the requisite generalizability and flexibility. This discrepancy underscores the critical need for a new paradigm that more closely mimics the human capacity for reasoning, adjusting the level of effort in reasoning based on the complexity of the task, thereby enhancing the adaptability of AD systems to dynamic situations[10].

Recent studies[11,12] revealed that Large Language Models (LLMs) exhibit cognitive abilities previously considered unique to humans. Studies[13,14] incorporating LLMs, such as GPT-4[15] into AD systems have shown that LLMs can integrate human-like common sense and reasoning into trajectory planning. This allows LLMs to comprehend driving tasks and effectively reason in dynamic environments, thereby enhancing the robustness of autonomous vehicles[16,17].

[1]School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing, China. [2]Department of Automation, Tsinghua University, Beijing, China. ✉e-mail: tianyu@ustb.edu.cn; mhmpub@ustb.edu.cn

However, these methodologies remain confined to specific urban settings or simpler contexts such as highways. This constraint is primarily owing to the challenges raised by the following four inefficiencies inherent in current research paradigms.

Challenge 1: Inefficiency in corner cases generalization. Existing LLM-based methods primarily map queries to responses without human cognition or reflection, which limits their adaptability and generalization to corner cases where continuous adaptation is essential.

Drawing on dual-process theory[18], we developed the CogniDrive framework. This theory differentiates between two modes of cognition: System 1, fast and intuitive, is ideal for immediate responses, and System 2, which is methodical, is better suited for intricate tasks. We proposed InstinctNav, which leverages LLM as a trajectory planner, effectively mitigating data distribution constraints that limit traditional approaches. Since both existing studies and InstinctNav fall into System 1, they may frequently encounter challenges in long-tail scenarios. For example, a human driver might slow down upon seeing a ball roll onto the street, anticipating a child might follow, a nuance often missed by existing AD[14]. In these situations, the thorough analysis provided by System 2 has emerged as a key factor.

Thus, we introduce ReflectPlan for System 2 to critically evaluate the initial planning provided by InstinctNav. Specifically, we extend the Actor-Critic (AC) framework[19] by advancing it to the prompt level, where the Actor (InstinctNav) generates actions, and the Critic (ReflectPlan) evaluates and refines the former. In this setup, the LLM within ReflectPlan functions as the Critic, processing and internalizing reward signals generated by its self-reflection mechanism[20,21], which is embedded in language tokens as part of the prompt. The introduction of ReflectPlan improves InstinctNav's initial planning and enhances its generalization ability. Furthermore, CogniDrive utilizes multimodal data sources to dynamically toggle between InstinctNav and ReflectPlan. This toggling mechanism optimizes responses by adapting cognition modes based on task difficulty, thereby promoting the coexistence and synergistic operation of both modes, which enhances CogniDrive's efficiency and generalization to corner cases.

Challenge 2: Inefficiency in few-shot generalization. Deliberate practice theory[22] suggests that humans can rapidly master skills through minimal exposure based on organized reasoning patterns and experiences. Conversely, current methods rely heavily on vast amounts of training data and are constrained by the data distribution[23,24].

Building on the above theory, we improved InstinctNav through fine-tuning using an exquisite thought model, whereby text instructions guide the LLM's reasoning process[25–28], and accumulated driving experiences. We employed behavioral cloning (BC)[29] to explicitly construct a thought model for trajectory planning, enabling InstinctNav to emulate human-like reasoning through our innovative two-stage fine-tuning. Additionally, we integrated driving experience into InstinctNav with retrieval augmented generation (RAG)[30] to bolster LLM's responses by dynamically retrieving and incorporating relevant experiences, thereby enhancing context awareness and overall response quality.

Challenge 3: Inefficiency in hazard perception. The perception module within CogniDrive, derived from conventional approaches with limited training data distribution, cannot fully bridge the perceptual gap to the oracle agent[1,5]. For instance, both traditional and LLM-based methods might fail to correctly recognize a tipping truck, leading to significant data omissions that compromise perception accuracy[31]. Currently, the perception modules in LLM-based methods are still derived from traditional approaches[5,6]. To address this, we integrated a vision-language model (VLM)[32–34] as an Evaluator to identify critical environmental elements, with a particular focus on detecting hazards in corner cases. Leveraging its built-in LLM, the VLM-based Evaluator extends beyond perception to reasoning, enabling it to infer scene conditions (e.g., slippery roads due to rain), compensate for perceptual omissions (e.g., a truck overturned on the side). Feedback from both the Critic and Evaluator informs the multimodal reward value for InstinctNav, facilitating effective self-adjustment upon the initial planning.

Challenge 4: Inefficiency in the comprehensive evaluation. Conventional metrics such as L2 error and collision rate fail to capture the full spectrum of intelligent driving demands, focusing primarily on safety and precision while neglecting aspects of passenger comfort and energy efficiency[35,36]. As autonomous vehicles become more integral to daily transportation, the criteria for successful navigation evolve to include these broader considerations. Personalized metrics that assess comfort and energy efficiency are crucial, as they reflect a vehicle's overall performance, enhancing passenger satisfaction and promoting sustainable practices[37,38]. To this end, within the CogniDrive framework, we calculate both jerk and energy from the Bézier curves[39] of the planned trajectories to assess comfort and energy efficiency.

In response to the outlined challenges in AD, we have developed the CogniDrive framework rooted in dual-process theory, which harmoniously integrates the proposed innovative InstinctNav and ReflectPlan. InstinctNav, drawing on deliberate practice theory, swiftly endows the LLM with autonomous navigation capabilities, refining trajectory planning by synthesizing reasoning patterns and accumulated experiences. ReflectPlan enhances this by enabling self-assessment and iterative refinement, thereby ensuring a seamless transition between intuitive and analytical thinking. The integration of VLM significantly enhances hazard perception, which in turn boosts data reliability and augments reasoning refinement. Additionally, we propose a comprehensive evaluation system that extends beyond traditional safety measures to include passenger comfort and energy efficiency, thus meeting the expansive, real-world demands of AD systems. Rigorous experiments on the nuScenes[40], Waymo Open Motion (Waymo)[41], and CODA[42] datasets under open-loop settings, together with closed-loop tests in the CARLA[43] simulator, confirm the superior performance of the proposed methods in both routine and complex scenarios. These results demonstrate the robustness and adaptability of our dual-process approach in advancing autonomous vehicle navigation.

Results

CogniDrive was evaluated under both open-loop and closed-loop settings. In the open-loop evaluation, large-scale real-world autonomous driving datasets, including nuScenes and Waymo, were used to assess planning quality. Compared with previous state-of-the-art (SOTA) methods, CogniDrive demonstrates consistent improvements across multiple metrics. Specifically, the collision rate achieves relative improvements of 21.43% and 13.33%, respectively, while trajectory jerk is reduced by absolute margins of 5.75% and 4.99%, indicating safer and smoother driving behaviors. In addition, energy-related consumption shows absolute reductions of 2824.8J and 2943.27J. These results establish new SOTA performance on the evaluated open-loop benchmarks. To further evaluate robustness under interactive environment feedback, closed-loop experiments were conducted in the CARLA simulator, where CogniDrive achieves a relative improvement of 10.54% in route completion success rate compared with previous methods. Ablation studies are also conducted to verify the effectiveness of the individual components within the CogniDrive framework.

To ensure model fairness, we adopted Qwen2.5-1.5B-Instruct[44] as the unified backbone across all LLM-based methods. Each baseline was re-trained from this checkpoint with the same data budget and hardware, using its original training protocol (optimization schedule, loss functions, prompts) and inference configuration to preserve behavior specific to each method. To ensure dataset fairness, we employ a unified split strategy in which train, validation, and test sets are strictly

Table 1 | Safety, comfort, and energy efficiency metrics in common scenarios

Dataset	Method	Collision rate (%) ↓				Jerk (%) ↓	Energy (J) ↓
		1s	2s	3s	Avg		
nuScenes	ST-P3[4]	0.23	0.62	1.27	0.71	8.24	15480.86
	UniAD[5]	0.06	0.12	0.52	0.23	7.32	9312.12
	VAD[6]	0.07	0.10	0.24	0.14	7.76	10472.43
	DriveVLM[75]	0.04	0.07	0.54	0.22	8.62	10341.85
	DiMA[76]	0.05	0.08	0.39	0.17	10.13	11897.32
	GPT-Driver[13]	0.04	0.12	0.34	0.17	8.88	12002.35
	Agent-Driver[14]	0.07	0.10	0.63	0.26	8.37	10389.30
	Ours	**0.03**	**0.07**	**0.23**	**0.11**	**1.57**	**6487.32**
	Human[40]	—	—	—	—	6.86	9648.98
Waymo	MT[48]	0.17	0.49	0.95	0.54	6.47	9372.57
	OpenEMMA[49]	**0.04**	0.14	0.29	0.15	6.78	10295.72
	GPT-Driver[13]	0.09	0.17	0.32	0.19	9.40	11792.12
	Ours	0.06	**0.11**	**0.22**	**0.13**	**1.48**	**6429.30**
	Human[41]	—	—	—	—	5.89	8429.59

The best-performing results are highlighted in bold. The symbol ↓ indicates that lower values of the metric correspond to better performance of the method. The same notation is used for all tables.

separated without overlap, and evaluation is conducted exclusively on the test set.

Regarding evaluation metrics, we align their selection with the experimental objectives. Specifically, L2 error is reported when evaluating CogniDrive's learning ability, while in other cases a comprehensive evaluation system (or its relevant subsets) is employed, complemented by task-specific metrics where appropriate. In addition, baseline selection is determined by the experimental setting and objective, ensuring that each evaluation is aligned with the aspect of CogniDrive's capability under investigation. The detailed reasons for the above choices can be found in the Supplementary Information, Note 5.

Efficiency in open-world generalization

One of the advantages of the proposed CogniDrive based on dual-process theory is that different reasoning modes can be adopted flexibly depending on the complexity of the task. To verify the validity of the proposed framework, we classified the nuScenes samples D into common scenarios $D_g = \{(i_t, x_t) \in D \mid i_t \notin I\}$ that require only simple numerical prediction capabilities, and corner cases $D_c = \{(i_t, x_t) \in D \mid i_t \in I\}$ that require sophisticated reasoning capabilities, according to the index I in CODA. The x_t denotes a query pertaining to the current driving scenario. The i_t is the index of the sample in D and CODA.

Comparison in common scenarios. Most AD methods primarily focus on common scenarios that do not require complex reasoning capabilities. We verified the performance of the proposed method on D_g and compared it with the baselines. Table 1 (nuScenes) shows that our method outperforms all baselines in terms of safety. The proposed method achieved a 21.42% lower average collision rate than the second-best method. The results confirmed CogniDrive's ability to handle general driving tasks aimed at safety using preset prompts. Table 1 (nuScenes) shows that the proposed method (Jerk = 1.57%, Energy = 6487.32J) outperforms the human driver (Jerk = 6.86%, Energy = 9648.98J) and baselines. This suggests that, when presented with related experience prompts, the proposed method surpasses baselines in comprehending personalized measures beyond merely aligning with the oracle agent.

Comparison in corner cases. Corner cases are difficult to address using conventional methods because of a lack of cognition and reflection regarding the environment[45–47]. As shown in Table 2, the proposed method is far superior to the baselines, achieving the lowest collision rate of 2.05%. This demonstrates the effectiveness of the proposed approach for managing complex scenarios. The primary reasons for this improvement are twofold: the method leverages LLM as a Critic to enhance planning through reflection reward value, and it utilizes VLM to address perception omissions.

Table 2 | Collision performance in corner cases

Method	Collision rate (%)			
	1s	2s	3s	Avg
ST-P3[4]	6.67	7.50	12.50	8.89
UniAD[5]	4.17	6.67	10.00	6.95
VAD[6]	2.50	5.01	9.17	5.56
DriveVLM[75]	1.03	2.72	5.83	3.19
DiMA[76]	0.95	2.36	5.09	2.13
GPT-Driver[13]	5.83	7.50	11.67	7.94
Agent-Driver[13]	1.67	4.17	7.50	4.45
Ours	**0.91**	**1.82**	**3.43**	**2.05**

The best-performing results are highlighted in bold.

CogniDrive outperforms existing LLM-based methods in handling corner cases. The latter, despite their reasoning capabilities, struggle with complex scenarios when limited to text-based environmental descriptions. In contrast, our method excels by leveraging the unstructured scenario understanding of VLM, thereby effectively addressing unpredictable, long-tailed scenarios such as trucks tipping over. Additionally, our method's superior performance compared to Agent-Driver likely stems from the effectiveness of the textual rewards in guiding the LLM rather than structured information supplementation in Agent-Driver.

Generalization in open-loop testing. Experiments on the nuScenes and CODA datasets demonstrate the effectiveness of our approach under diverse urban driving conditions. To further demonstrate generalization, we evaluate our method on the Waymo dataset. For the Waymo evaluation, we applied both Motion Transformer (MT)[48] and OpenEMMA[49], and additionally included GPT-Driver, which is robust to sensor adaptation. Other approaches under nuScenes were excluded from this dataset due to sensor adaptation issues.

As shown in Table 1 (Waymo), our method consistently outperforms existing baselines, achieving both the lowest Collision rate (0.13%) and the lowest Jerk (1.48%). Notably, our total Energy (6429.30J) is significantly lower than all compared methods, highlighting superior efficiency. These results demonstrate that our framework not only generalizes well across datasets but also delivers robust performance.

Comparison in closed-loop testing. The experiments on the nuScenes, Waymo, and CODA datasets provide valuable insights into model capabilities in controlled and repeatable settings. However, such open-loop evaluations cannot capture error accumulation or dynamic interactions with the environment. To address these limitations, we conducted closed-loop testing in CARLA[43], placing the model in a real-time feedback loop to enable a more realistic assessment of stability, safety, and adaptability. We compare our approach against two Reinforcement Learning (RL)[50] methods (SA-TD3[51] and MMI-DQN[52]), two mainstream end-to-end pipelines (UniAD and VAD), and three representative LLM/VLM-based approaches (GPT-Driver, Agent-Driver, DriveVLM, and DriverGPT4-V2).

As shown in Table 3, our method achieves the highest success rate (27.37%), the lowest Jerk (8.20%), and the lowest Energy (8173.62J), consistently outperforming all baselines across safety, comfort, and efficiency. The CARLA experiments cover diverse settings, including variations in geography, weather, and illumination, with full configuration details given in Supplementary Information, Note 1.

Table 3 | Performance in closed-loop testing on CARLA

Method	Succ. (%) ↑	Jerk (%) ↓	Energy (J) ↓
SA-TD3[51]	9.42	16.32	13256.12
MMI-DQN[52]	8.18	14.27	11392.40
UniAD[5]	18.29	11.55	10210.32
VAD[6]	17.34	11.24	9455.71
DriveVLM[75]	20.90	9.70	9725.80
DriverGPT4-V2[74]	24.76	11.37	10692.32
GPT-Driver[13]	18.01	14.80	11195.34
Agent-Driver[14]	21.16	12.18	10080.28
Ours	**27.37**	**8.20**	**8173.62**

Succ. indicates the completion success rate of pre-defined routes in CARLA. The best-performing results are highlighted in bold. The symbol ↓ indicates that lower values correspond to better performance. The symbol ↑ indicates that higher values correspond to better performance.

Interpretability. CogniDrive differs from conventional methods[2–6] by relying on black-box artificial neural networks[53] to map visual features to signals. The inference of the proposed method is a sequence-to-sequence (seq2seq) task[54] realized by chain-of-thought (CoT)[25] that allows the LLM to generate trajectory planning in textual form, token-by-token, which makes the proposed method interpretable. Fig. 1 presents the output of the proposed approach, showing the trajectory planning, reflection, and re-planning, all visualized in textual form.

Reasoning mode switching. The frequency of ReflectPlan activation differs markedly across scenarios, occurring in 5.32% of common scenarios but rising to 36.42% in corner cases. This substantial disparity indicates that the mechanism is selectively triggered in complex environments that demand enhanced reasoning. These results further demonstrate the effectiveness of the Evaluator in enabling adaptive and strategic mode switching within the CogniDrive framework.

Efficiency in few-shot generalization

We embedded the cognitive principles of deliberate practice theory into our AD system through a two-stage fine-tuning process combined with experience accumulation and retrieval mechanisms. To verify its

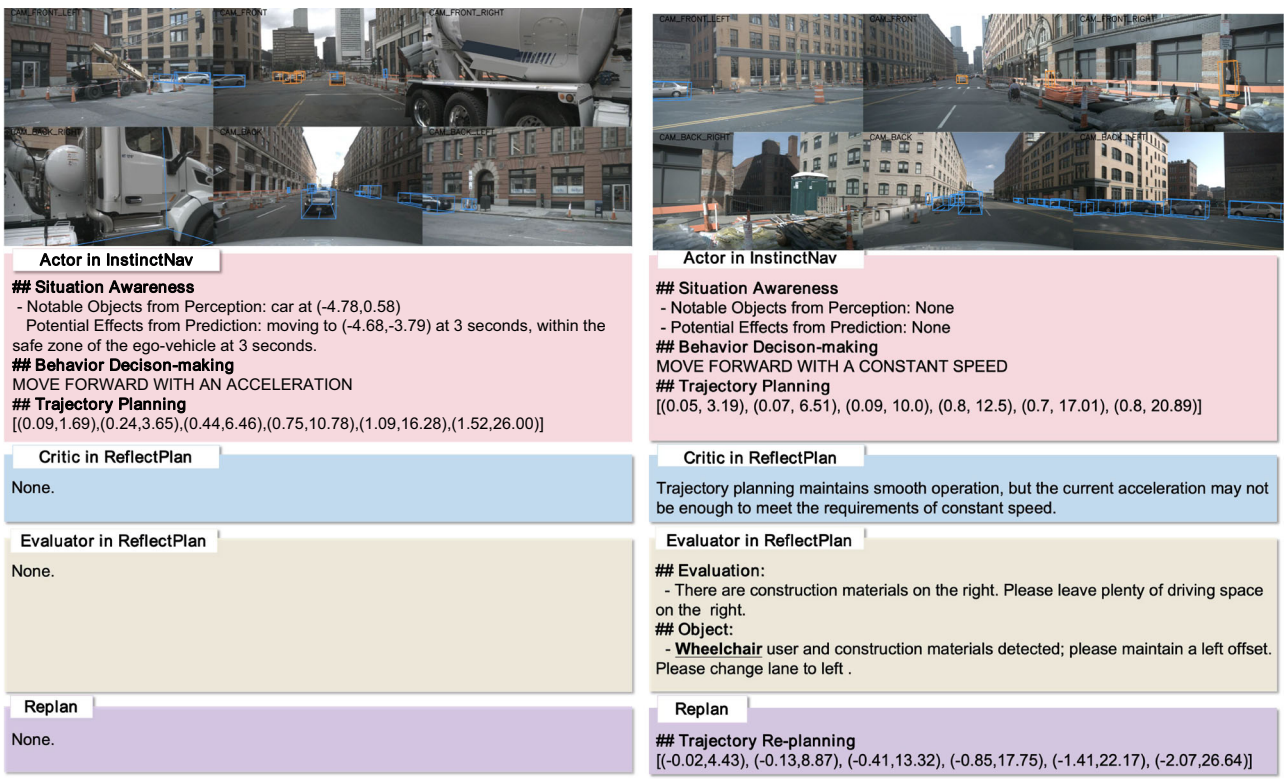

Fig. 1 | Instance of a corner case from CODA and the corresponding reasoning process and planning results. Left Instance of a common scenario and the corresponding reasoning process. **Right** Instance of a corner case from CODA accompanied by corresponding reasoning processes.

https://doi.org/10.1038/s41467-026-72030-6

Table 4 | Ablation studies on efficiency and robustness

(a) Two-stage fine-tuning.				
Method	**L2 (m) ↓**			
	1s	**2s**	**3s**	**Avg**
ST-P3[4]	1.14	1.46	2.59	1.73
UniAD[5]	0.17	0.34	0.64	0.38
VAD[6]	0.15	0.31	0.60	0.35
DriveVLM[75]	0.16	0.37	0.68	0.40
GPT-Driver[13]	0.16	0.36	0.66	0.39
Agent-Driver[13]	0.19	0.32	0.70	0.40
CogniDrive w/o	1.89	2.17	4.50	2.86
+FL	0.15	0.36	0.61	0.37
+FL&VL	**0.13**	**0.27**	**0.56**	**0.32**

(b) Experience accumulation.			
Setting	**Collision (%)**	**Jerk (%)**	**Energy (J)**
Zero-shot w/o	1.07	11.32	13220.01
Fine-tuning w/o	0.13	3.78	8132.73
I	**0.11**	**1.57**	**6432.77**
II	**0.11**	1.82	6838.19
III	0.16	5.47	7832.01
IV	0.22	5.17	9518.47
V	0.20	6.32	9996.82

(c) Perception module and VLM understanding.				
Perception Module		**VLM**	**Avg Col (%)**	**Avg L2 (m)**
Detection	**Prediction**			
UniAD	UniAD	×	0.14	0.33
VAD	VAD	×	0.15	0.31
UniAD	UniAD	?	**0.11**	**0.32**

(a) Effectiveness of the proposed two-stage fine-tuning strategy. The notation w/o indicates settings without fine-tuning. **(b)** Effect of experience on Collision (%), Jerk (%), and Energy (J) scores in common scenarios. The notation w/o indicates settings without experience, whereas I-V denotes the number of experiences in the process of RAG. **(c)** Effect of the environment perception module. Avg Col denotes average collision rate, and Avg L2 denotes the average L2 trajectory error. The same notation applies to all panels. The best-performing results are highlighted in bold. The symbol ↓ indicates that lower values correspond to better performance.

effectiveness, ablation experiments were conducted on various components of deliberate practice theory.

Two-stage fine-tuning. Fine-tuning is a crucial component of deliberate practice theory. We achieve this through two stages: format learning (FL) and value learning (VL). The L2 errors between the planned trajectories of all planning methods and ground truth are listed in Table 4a. The results show that the proposed method with FL approaches an average L2 error of 0.37. In contrast, our method, after two-stage fine-tuning, achieves state-of-the-art (SOTA) performance with a 0.32 L2 error. Table 4a confirms the effectiveness of the two-stage fine-tuning with FL + VL, which validates the effectiveness of the proposed approach in mimicking the process of human learning in deliberate practice theory.

Experience base. Experience accumulation is another crucial component of deliberate practice theory. To distinguish between the experience and improvement for corner cases using ReflectPlan, we performed ablation experiments on common scenarios. The results in Table 4b show that our method achieves the lowest average collision rate of 0.11% when searching for one experiences in a experience base, demonstrating its effectiveness. However, contrary to our initial expectation that more experience would enhance performance, the results indicate the opposite. Inspired by ref. 55, we observed that

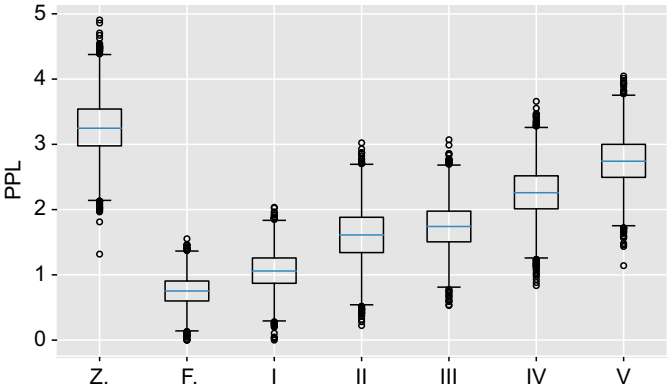

Fig. 2 | Box plots of pperplexity scores under different experience settings. PPL denotes perplexity. Lower PPL reflects reduced uncertainty in LLM generation as more experience is incorporated. Z. and F. denote the Zero-shot w/o and Fine-tuning w/o settings, respectively. In the box plots, the center line represents the median, the box indicates the interquartile range, the whiskers extend to 1.5 × the interquartile range, and individual points correspond to outliers. Source data are provided as a Source Data file.

increasing the number of low-relevance experiences raised the Perplexity (PPL) score of LLM. $PPL = \exp(-\frac{1}{T}\sum_{i=1}^{T}\log p(x_i))$[56] is a measure of model uncertainty. This is evidenced by the results presented in Fig. 2, which show that as irrelevant experiences increase, the uncertainty in LLM increases, adversely affecting its reasoning accuracy. This is because the model relies on extracting relevant features and patterns from extensive data. When inundated with irrelevant information, the attention mechanism of the model is disrupted, leading to higher uncertainty and decreased prediction accuracy.

Few-shot generalization. Humans possess the remarkable ability to master complex tasks with minimal exposure, a capability known as few-shot generalization[57]. To assess this ability of LLM within the CogniDrive framework, we conducted targeted experiments. Traditional methods FF[2] and EO[3] were included as baselines to highlight the proposed method's superior performance in few-shot setting. For comparison, we adopted ST-P3 with 100% of the training samples as the baseline. Figure 3a, b illustrate the few-shot generalization performance in terms of average collision rate and L2 error, respectively, demonstrating the effectiveness of the proposed method in few-shot generalization.

As shown in Fig. 3a, CogniDrive surpassed the baseline in average collision rate when using only 10% of the training data. Furthermore, Fig. 3b shows that CogniDrive outperforms the baseline with just 5% of the training samples. Figure 3c indicates that with minimal adjustments to the training data, CogniDrive can rapidly learn control command formats for machinery. Overall, these findings confirm the effectiveness of the proposed method in achieving excellent performance in few-shot generalization, thereby improving learning efficiency.

Efficiency in environment and hazard perception

For perception ablation, we selected UniAD and VAD as they are widely adopted and have demonstrated stable performance. Table 4c presents the results of our ablation experiment on environment perception. These results demonstrate the effectiveness of the perception module and VLM in our method. First, we integrated the traditional perception module from UniAD to construct LLM's queries, significantly enhancing our approach. Additionally, as shown in Table 4c our method outperforms conventional methodologies, primarily due to the superior sensing and reasoning capabilities of VLM. This

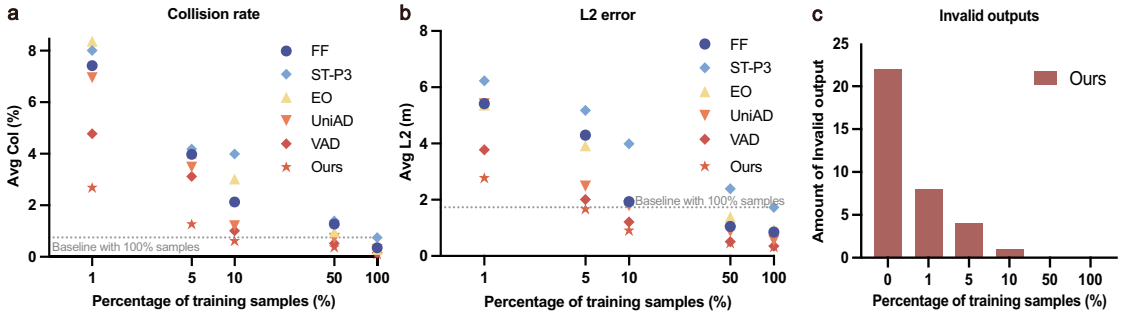

Fig. 3 | Few-shot generalization performance of the proposed method. a Few-shot generalization performance comparison on the Avg Col (%). Avg Col denotes average collision rate. **b** Few-shot generalization performance comparison on the Avg L2 (m). Avg L2 denotes the average L2 trajectory error. **c** CogniDrive's number of misformatted results as training samples increase. Source data are provided as a Source Data file.

demonstrates the effectiveness of our approach in enhancing the perception capabilities of AD systems.

Discussion

Overall, our approach exhibits excellent performance in both common scenarios, where previous approaches excel, and in complex scenarios, where previous approaches encounter challenges. In addition, the black-box nature of the LLM makes this approach extendable to other LLMs, such as Llama-3[58], Claude-3[59], GPT-4o[60], and OpenAI o1-preview[61]. This suggests that the current results have not reached the full potential of the proposed method and that the proposed approach will continue to offer opportunities for optimization as LLM develops.

The results presented in Tables 1–4 and Figs. 2–3 provide quantifiable indicators of performance. However, these results do not fully capture the advantages of the proposed approach. Drawing on dual-process theory, our method effectively handles long-tail scenarios by leveraging the common sense and reasoning abilities of LLM, addressing perceptual omissions using VLM. These capabilities allow our approach to mimic the perception, cognition, reasoning, and decision cycles of human drivers, underscoring the method's suitability for open-world environments where human-like understanding and reasoning are critical. This process is illustrated in Fig. 1. Compared with purely VLM-based approaches, integrating LLM reasoning with VLM-based reflection enables structured multi-step decision reasoning while avoiding repeated heavy visual inference, thereby improving both reasoning flexibility and overall computational efficiency.

A practical limitation of our framework is real-time performance: the proposed framework relies on a large LLM ($\geq 0.5B$ parameters[62]), resulting in non-negligible forward-pass latency during decision making. To meet real-time requirements, we systematically investigate acceleration techniques, such as pruning and quantization. A more detailed discussion of these considerations is provided in Supplementary Information, Note 2. In addition, LLM may also inherit pre-training biases, potentially resulting in unfair risk allocation in AD, which we do not explicitly address. More broadly, LLM-based approaches for safety-critical decision-making remain an emerging research direction, and their interpretability, safety guarantees, and real-world effectiveness have not yet reached full consensus. While our framework demonstrates encouraging results in both open-loop datasets and closed-loop simulation, further validation in real-world driving environments will be necessary to fully assess its deployment potential. As future work, we will encode fairness and ethical rules as first-order logic under explicit legal frameworks to bound inference and enable auditability, and we will evaluate their implications for fairness, interpretability, and accountability. An extended discussion of these considerations is provided in Supplementary Information, Note 3.

In summary, the development of the CogniDrive framework represents a significant advancement in AD technology, incorporating human-like adaptability and sophisticated decision-making. By leveraging a dual-system approach, the LLM in CogniDrive enhances the ability of autonomous vehicles to generalize and navigate in open-world environments. Our approach uses two-stage fine-tuning to construct a thought model and incorporates experience accumulation to greatly enhance its learning efficiency by encoding deliberate practice theory. Additionally, the integration of the VLM expands CogniDrive's perception capabilities, providing more reliable environmental insights. VLM's reasoning ability further contributes to trajectory planning by generating reward signals in text-based environmental representations that the LLM can interpret. To meet the broader, real-world demands of AD systems, we propose a comprehensive evaluation system extending beyond traditional safety measures to include passenger comfort and energy efficiency. All data and code related to our method are available in the Fine-tuning details section. The progress of our approach is expected to encourage domain experts to explore new directions in advancing the field.

Methods

General schema and task design

Conventional methods formulate the AD task by generating the corresponding driving action $\mathcal{A}$ based on scene information $\mathcal{S}$ encompassing ego, observed state, and other information, such as historical trajectory $\mathcal{H}$. These actions serve as intermediate features to formulate an N-waypoint trajectory $\mathcal{T} = \{(u_t, v_t)\}_{t=1}^{N} \in \mathcal{R}^{N\times 2}$, where N represents the total number of planned time steps.

CogniDrive framework is based on LLM to realize dual-process and deliberate practice theories, as shown in Fig. 4a. To leverage the human-like reasoning abilities of LLMs, we redefined the trajectory-planning task as a seq2seq task, as shown in Fig. 4b. In this setup, the states $\mathcal{S}$ and histories $\mathcal{H}$ serve as inputs, and the LLM generates vehicle control trajectories based on the instruction $\mathcal{L}_g$. CogniDrive operates in two cognition modes.

InstinctNav: in the fast and intuitive planning mode, CogniDrive instructs the LLM, with the instruction $\mathcal{L}_g$, to sequentially infer tokens of objects $\mathcal{O}$, behaviors $\mathcal{A}$ (e.g., slowing down, turning left), and trajectories $\mathcal{T}$ through CoT. This is achieved by mapping the concatenated query prompt $\{\mathcal{L}_g, \mathcal{S}, \mathcal{H}\}$ to the response $\{\mathcal{O}, \mathcal{A}, \mathcal{T}\}$, utilizing the accumulated experience base $\mathcal{M}$, as detailed in Eq. (1).

ReflectPlan: in the slow but analytical planning mode, CogniDrive instructs the LLM through instruction $\mathcal{L}_r$ to sequentially infer reflections $\mathcal{R}$ to obtain an improved trajectory $\mathcal{T}^*$ using a CoT reasoning process. This process involves assessing the need for reflective analysis, where reflections $\mathcal{R}$ are generated through a self-reflection mechanism to optimize $\{\mathcal{O}, \mathcal{A}, \mathcal{T}\}$, as shown in Eq. (2). During the training stage, these reflections are further accumulated into the

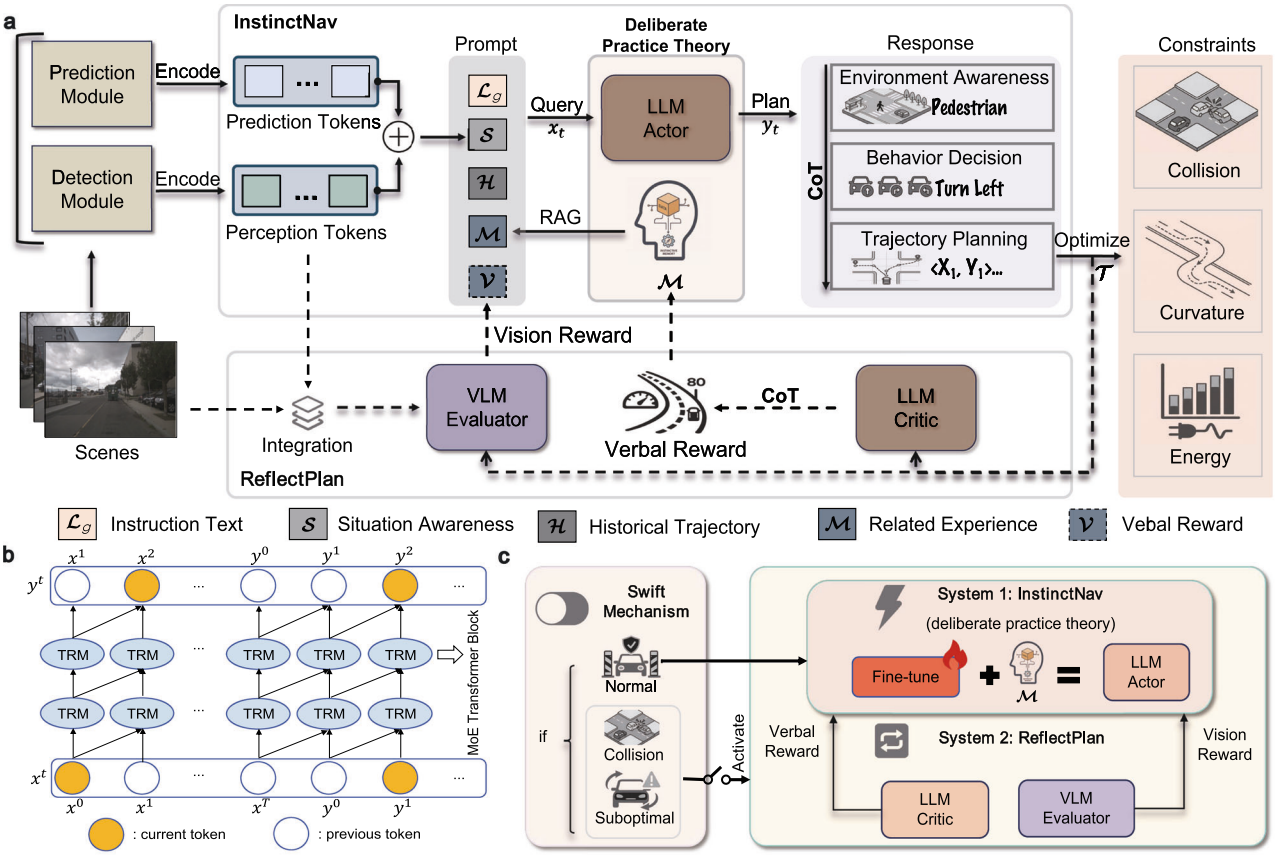

Fig. 4 | CogniDrive is an AD domain-specific reasoning framework built dual-process theory and deliberate practice theory. a CogniDrive diagram framework. **b** A sketch of the seq2seq task in CogniDrive. Each TRM unit functions as a specialized expert within the MoE[78] Transformer block. These blocks selectively activate experts per layer to improve computational efficiency and scalability. **c** The symbiotic thought switch mechanism between InstinctNav and ReflectPlan. RAG denotes retrieval-augmented generation and CoT denotes chain-of-thought reasoning.

experience base $\mathcal{M}$ to support future scenarios.

$$\{\mathcal{O},\mathcal{A},\mathcal{T}\}=LLM(\mathcal{L}_g,\mathcal{S},\mathcal{H};\mathcal{M}) \quad (1)$$

$$\{\mathcal{R},\mathcal{O}^*,\mathcal{A}^*,\mathcal{T}^*\}=LLMs(\mathcal{L}_r,\mathcal{S},\mathcal{H},\mathcal{T};\mathcal{M}) \quad (2)$$

InstinctNav and ReflectPlan represent System 1 and System 2 in dual-process theory, respectively, as shown in Fig. 4c. CogniDrive facilitates the symbiosis between two cognition modes through a dynamic switching mechanism. A more detailed discussion on the correspondence between the cognitive theory and its computational implementation is provided in Supplementary, Note 4.

InstinctNav planning mode schema

InstinctNav, as CogniDrive's System 1, delivers rapid and intuitive driving decisions by following a fixed, human-inspired reasoning schema: Situation Awareness → Behavior Decision-making → Trajectory Planning. Inspired by deliberate practice theory, this schema is encoded into the LLM on a dialogue dataset D_d constructed from expert demonstrations, with the two-stage fine-tuning process illustrated in Fig. 5b, c, enabling iterative internalization of reasoning patterns. Leveraging predefined CoT templates with minimal branching (Fig. 5a), InstinctNav mirrors System 1 by operating quickly, pattern-driven, and with low cognitive cost. In addition, a lightweight RAG module selectively retrieves the most relevant experiences from the experience base $\mathcal{M}$ (Fig. 5d) to guide trajectory planning, narrowing the search space and ensuring immediate safety and comfort without exhaustive deliberation.

Thought model. We propose a Thought Model that methodically infers trajectory planning through stages of Situational Awareness, Behavior Decision-making, and Trajectory Planning. To support this modeling, we designed a system message $\mathcal{L}_g$ composed of Role Definition, Thought Model, and Environment Information, thereby orchestrating the LLM's functionality by defining roles, guiding task execution, and integrating contextual data to enhance decision-making accuracy and coherence. Role Definition activates the LLM's domain knowledge, while Environment Information feeds into LLM's reasoning. The details of the Thought Model illustrated in Fig. 5a are outlined in the subsequent three paragraphs.

• Situation Awareness. Situation Awareness involves identifying noteworthy objects and analyzing their underlying influences. This process requires a deep understanding of the current scenario and the application of common-sense reasoning[9]. LLM is equipped with enhanced situational awareness through CoT.

• Behavior Decision-making. This module, guided by Situational Awareness, fundamentally influences planning. This stage generates behaviors (such as maintaining a constant speed, accelerating left, or changing lanes) involved in trajectory planning. Compared with traditional methods, Behavior Decision-making in LLM restricts the search space thereby ensuring improved trajectory planning results.

• Trajectory Planning. Trajectory Planning in the language modeling task, guided by Behavior Decision-making, aims to predict N waypoints that are safe, comfortable, and energy-efficient. This process involves the LLM generating text in system message format: Trajectory: [(0.07, 4.59),..., (1.62, 26.46)]. Trajectory Planning is initiated by analyzing Situational Awareness, followed by reasoning based

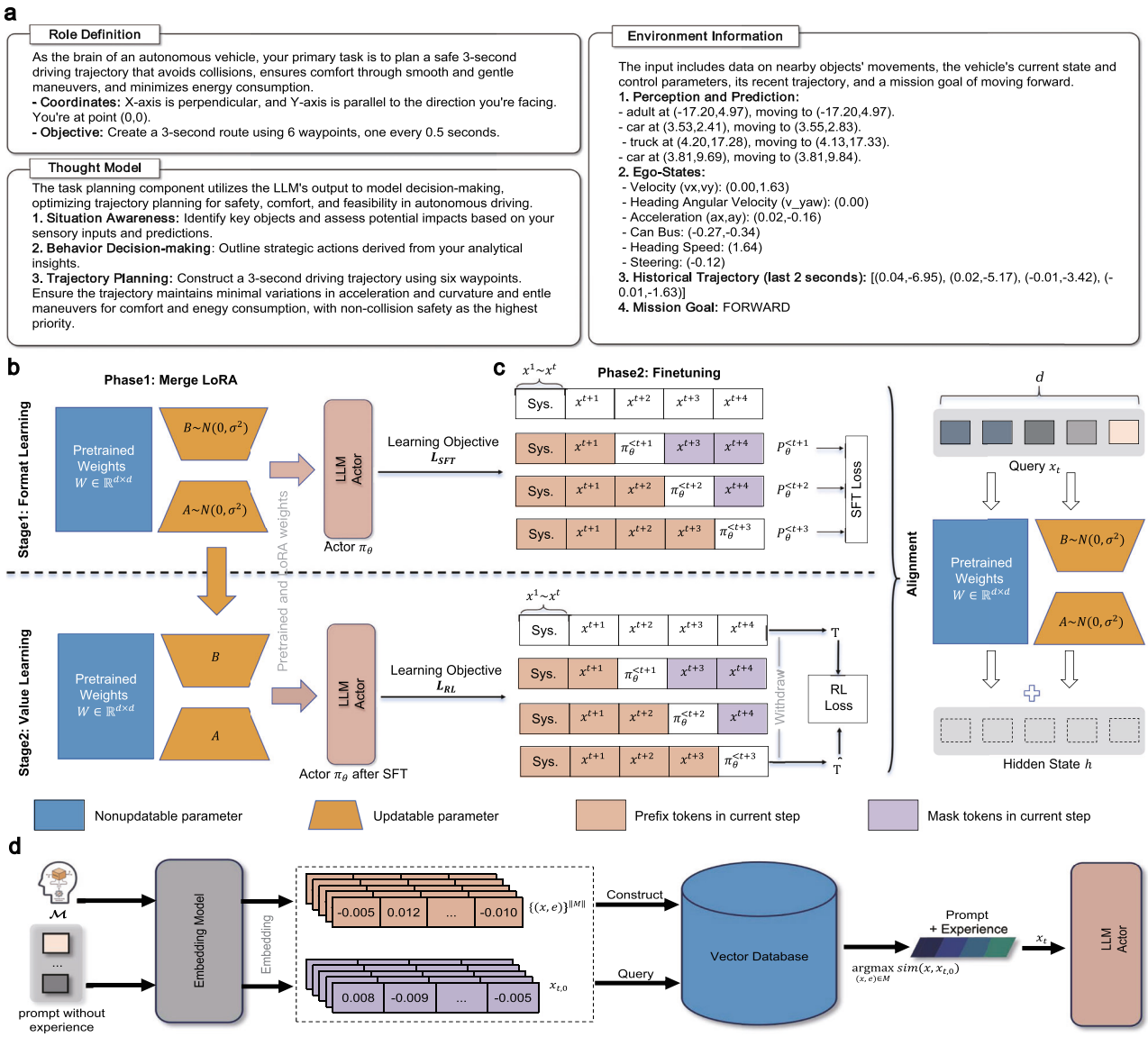

Fig. 5 | InstinctNav diagram. a The system message in the Actor's prompt includes role definition, thought model, and sampled environment information. **b** The Actor is fine-tuned in two stages: format learning and value learning. The Actor consists of a frozen LLM and an adaptable LoRA module, where the LoRA parameters in the second stage are initialized from those learned in the first stage. **c** LoRA merging process. The pretrained LLM weights correspond to the linear layers of the model. **d** Experience texts derived from historical driving scenarios are embedded into the prompt. A lightweight language model generates embeddings, and cosine similarity $sim(\cdot)$ is used for retrieval. x and $x_{t,0}$ denote the embedding indices used for similarity computation. LoRA (Low-Rank Adaptation) is a parameter-efficient fine-tuning method that adapts pre-trained models by introducing trainable low-rank matrices.

on Behavior Decision-making, leading to a synthesis of this information.

The modules use CoT for both individual reasoning and mutual interconnection, with the proposed Thought Model representing an advanced CoT task composed of a sequence of standard CoT tasks, forming a layered and meticulous reasoning framework. The explicitly specified multi-layer CoT reasoning process demonstrates strong interpretability (as shown in Fig. 1). Building upon this reasoning process, the Thought Model design also predefines instructions to respond to multi-dimensional goals such as safety, comfort, and energy efficiency.

Actor. As described in the AC framework, the LLM in CogniDrive is partly instantiated as an Actor to plan trajectories. The Actor is expressed as π_θ to predict and sample tokens in its vocabulary table to generate trajectories. At each step t of an episode, the state s_t and historical trajectory h_t are taken as inputs to construct the prompt $x_t = Concat(\mathcal{L}_g, s_t, h_t)$ which feeds into π_θ to generate y_t made up of noteworthy objective o_t, driving behavior a_t, and available trajectory τ_t token-by-token, which is a process embodied as CoT. Therefore, Eq. (1) is reformulated as follows.

$$y_t = \pi_\theta(x_t; \mathcal{M}). \tag{3}$$

To enhance Actor under the lens of deliberate practice theory, we incorporate a RAG mechanism that searches the experience base $\mathcal{M}$ using the embedding of x_t to retrieve the most relevant experience e_t (Fig. 5d).

We regard π_θ as a language model (LM) that takes as input the instruction $\mathcal{L}_g = \{x_t^1, \ldots, x_t^c\}$ together with the $\{x_t^{c+1}, \ldots, x_t^m\}$, which encodes perception and ego state, denoted as query $x_t = \{x_t^1, x_t^2, \ldots, x_t^m\}$, and generates the corresponding response $y_t = \{y_t^1, y_t^2, \ldots, y_t^n\}$ at t-th step. Here, m and n denote the query and response token lengths, respectively. y_t is the sentence from

conditional probability distribution $\pi_\theta(y_t|x_t; e_t)$. From the language modeling perspective, Eq. (3) translates into:

$$p_\theta(y_t|x_t; e_t) = \prod_{i=1}^{T} p_\theta(y_t^i|x_t, y_t^{<i}; e_t), \tag{4}$$

Fine-tuning. In deliberate practice theory, organized skill learning corresponds to finetuning the LLM in Actor, where the Thought Model is encoded into the LLM to form a structured base model that supports experiential accumulation.

We introduce both format learning (FL) and value learning (VL) to master the Thought Model and optimize performance, as shown in Fig. 5b. FL ensures that the Actor can generate planning with a structured output format, facilitating straightforward trajectory extraction and stimulating the numerical prediction capabilities. Meanwhile, VL aligns the Actor outputs more closely with the oracle agents, enhancing the model's adaptability and flexibility. At each fine-tuning stage, low-rank adaptation (LoRA)[63] reduces costs and increases speed by initializing and applying low-rank decomposition, minimizing parameter updates while maintaining performance, as shown in Fig. 5c. This two-stage fine-tuning not only improves task performance but also ensures the system's robustness.

• Format learning Stage. To ensure that the generative model π_θ consistently produces trajectory planning results in a fixed format, to make the trajectories usable, we fine-tuned the Actor, specifically referring to Qwen2.5-1.5B-Instruct through Supervised Fine-Tuning (SFT)[64] that predicts the mask token y^i in turn based on the given prefix $x + y^{<i}$ on the i-th step, as in Eq. (4). The objective function is expressed as follows:

$$L_{SFT}(\theta) = -\mathbb{E}_{(x,y)\in D_d} \log p_\theta(y^i|x + y^{<i}), \tag{5}$$

where $D_d = \{\langle x_t, y_t\rangle | t \in [0, T]\}$ is the dataset consisting of text pairs of query x and response y. We concatenated the prompts and corresponding responses into complete strings to train the LLM model.

We optimize model parameters so that the FL fine-tuned model tends to generate $\hat{y}_t$ in a fixed format matching the ground truth y_t made up of noteworthy objective o_t, driving behavior a_t, and available trajectory $\boldsymbol{\tau}_t$ tokens. The fine-tuned model is constrained to generate $\hat{y}_t$ in the same structured tokenized format, ensuring one-to-one alignment with the ground truth trajectory. Prefilled context vectors x_t are gradient-masked during loss computation, guiding the model to focus on the predefined output structure.

• Value learning Stage. At this stage, the state space in step j comprises x and progressively generates tokens $\{y^i\}_{i=1}^{j-1}$. Accordingly, the action that the Actor explores and samples is the generated token y^j at the j-th step. To improve the Actor after FL with the intervention of RL, we introduced a objective function designed to better align our model's planning with human drivers and optimize the utilization of trajectory reward, as expressed in Eq. (6).

$$L_{RL}(\theta) = \mathbb{E}\left[x \sim X, \{\hat{y}_i\}_{i=1}^{G} \sim \pi_{ref}\right] \frac{1}{G}\sum_{i=1}^{G}\left\{\min\left[\frac{\pi_\theta(\hat{y}_i|x)}{\pi_{ref}(\hat{y}_i|x)}A_i, clip\left(\frac{\pi_\theta(\hat{y}_i|x)}{\pi_{ref}(\hat{y}_i|x)}, 1-\epsilon, 1+\epsilon\right)A_i\right] - \beta D_{KL}\left[\pi_\theta \parallel \pi_{ref}\right]\right\}, \tag{6}$$

$$D_{KL}[\pi_\theta \parallel \pi_{ref}] = \frac{\pi_\theta(\hat{y}|x)}{\pi_{ref}(\hat{y}|x)} - \log\frac{\pi_\theta(\hat{y}|x)}{\pi_{ref}(\hat{y}|x)} - 1, \tag{7}$$

Here, X is the query part within D_d; $D_{\mathrm{KL}}(\pi_\theta||\pi_{\mathrm{ref}})$ is used to penalize the RL strategy for generating substantial deviations from the initial model in each training batch to ensure that the model outputs reasonably coherent text; π_{ref} refers to the parameter-frozen LLM within the Actor. Eq. (7) uses the Schulman approximation ($ratio - log_{ratio} - 1$)[65] to ensure that the KL divergence is always positive.

The expectation $\mathbb{E}_{\{\hat{y}_i\}_{i=1}^{G}\sim\pi_{ref}}$ aims to average the rewards to approximate the advantage value A_i, thereby reducing GPU memory usage during the calculation of total advantage value calculated by a model of the same scale as the Actor, as expressed in Eq. (8)–(9). Aligning with the reward model and subtracting a baseline from each reward to decrease variance are sufficient to stabilize training[66].

$$A_i = \frac{R_i - avg(\mathbf{R}_{ref})}{std(\mathbf{R}_{ref})}, \tag{8}$$

$$avg(\mathbf{R}_{ref}) = \frac{\sum_{j=1}^{G} w_j R_{ref,j}}{\sum_{j=1}^{G} w_j}, \tag{9}$$

where w_j is the inverse of the PPL of LLM. A lower PPL indicates that the model is more certain about the sample. Therefore, higher weights are assigned to the corresponding reward. This negates the need for decay techniques such as generalized advantage estimation.

The computational efficiency of L_{RL} was further enhanced by employing a reward function (Eq. (10)) to estimate trajectory errors rather than using a reward model of the same scale as the Actor, thereby reducing the GPU memory demands. The performance of π_θ is evaluated by calculating the Euclidean distance between the predicted trajectory $\hat{\boldsymbol{\tau}} = [(\hat{u}_1, \hat{v}_1), \ldots, (\hat{u}_N, \hat{v}_N)]$ and the oracle agent's trajectory $\boldsymbol{\tau} = [(u_1, v_1), \ldots, (u_N, v_N)]$, as denoted by Eq. (10). Here, x and y refer to coordinates.

$$R = \sqrt{(\boldsymbol{\tau} - \hat{\boldsymbol{\tau}})^2}. \tag{10}$$

The Euclidean error not only ensures the model's alignment with human's safe driving trajectories but also, by mimicking human driving behavior, subtly inherits the multi-objective trade-offs between safety, comfort, and energy efficiency.

ReflectPlan planning mode schema

ReflectPlan serves as CogniDrive's System 2, enhancing trajectory generation through reflective optimization under textual feedback from multimodal reflection (Fig. 6a). The process unfolds within an Actor-Critic-Evaluator loop, where generated plans are iteratively refined based on verbal and perceptual feedback. The LLM-based Critic delivers verbal feedback on safety, comfort, and energy efficiency, while the VLM-based Evaluator contributes visual feedback derived from scene reasoning and hazard perception. These textual feedback from multimodal reflection are integrated into re-planning without retraining, stored for reuse to extend CoT depth and support coherent decision-making in complex scenarios.

Attention heatmap analysis (Fig. 6d) reveals that ReflectPlan exhibits a more diffuse and broadly distributed attention pattern across token dependencies compared to the localized focus of InstinctNav, thereby empirically supporting its higher complexity in probabilistic reasoning, consistent with the probabilistic nature of the LLM (Eq. (4)).

Critic. Assuming that the LLM functions as the Actor in InstinctNav, a Critic is necessary to adjust the Actor's behavior according to the AC framework. Initially, the Actor generates trajectory $\boldsymbol{\tau}_t$ using Eq. (3). The Critic then evaluates this trajectory based on x_t, producing a reflection and reward r_t as specified in Eq. (11). The Actor then generates the optimized response y_t^* by incorporating r_t into the query x_t, as shown in Eq. (12), where the optimized trajectory $\boldsymbol{\tau}_t^*$ is included in y_t^*. Eq. (11)–(12) illustrate the commonality between training and inference. During the

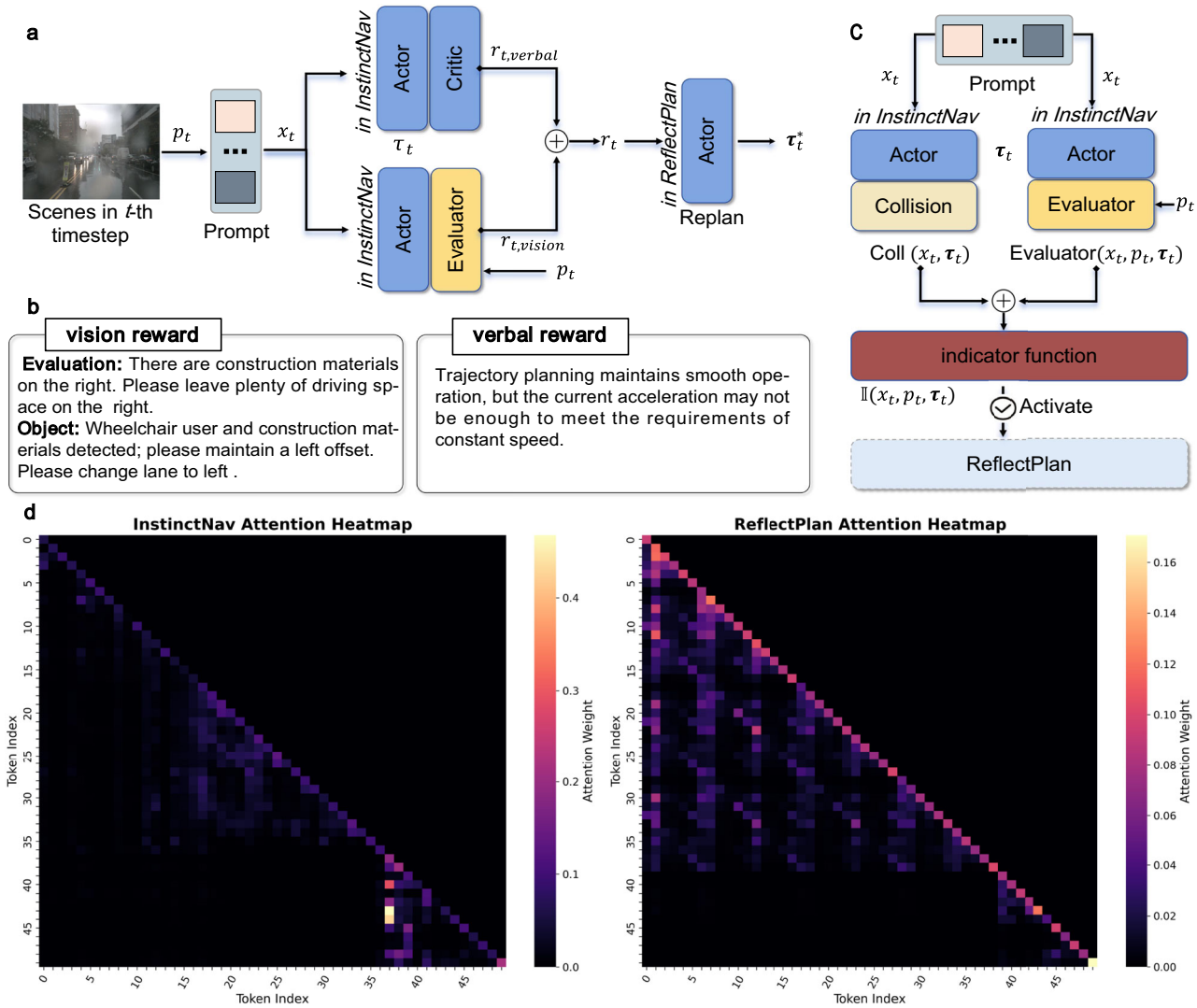

Fig. 6 | Overall flowchart of the ReflectPlan planning mode. a Overall process of ReflectPlan planning. The plan represents InstinctNav as an Actor. **b** Example of a concrete textual representation of r_t. **c** Conversion process from InstinctNav to ReflectPlan. **d** Attention heatmaps of ReflectPlan and InstinctNav, illustrating differences in focus patterns. Source data for (**d**) are provided as a Source Data file.

inference process, through the addition of visual modules and experience base, these equations are displayed in different forms. The details of these forms are introduced in the inference optimization section.

$$r_t = Critic(x_t, \boldsymbol{\tau}_t) \tag{11}$$

$$y_t^* = (o_t^*, a_t^*, \boldsymbol{\tau}_t^*) = \pi_\theta(x_t, r_t) \tag{12}$$

Experience accumulation and retrieval. Experience plays a fundamental role in refining human driving skills based on deliberate practice theory. In our framework, experience accumulation is formalized by using time-to-collision (TTC)[67] as the reflection selection criterion. For each training sample, the model performs multiple reasoning trials to generate a set of candidate reflections $E = \{r^1, r^2, ..., r^m\}$. The optimal reflection e is then selected as

$$e = \arg\max_{r \in E} \text{TTC}(\boldsymbol{\tau}^*, r). \tag{13}$$

and stored in the experience base $\mathcal{M}$. TTC($\boldsymbol{\tau}^*$, r) is the TTC with $\boldsymbol{\tau}^*$ refined by r as Eq. (12).

During inference, we employ the Faiss[68] database to efficiently retrieve the top-k most relevant experience texts based on cosine semantic similarity (Fig. 5d). A similarity-threshold filter γ_{sim} is then applied to discard low-relevance candidates (e.g., $sim < \gamma_{sim}$). The remaining experience texts are appended, in pragmatically instructive natural language, to the LLM inference prompt, thereby guiding the model to leverage past knowledge for policy transfer and planning optimization.

Inference optimization. ReflectPlan lacks a direct reward value as in the training phase of TTC for trajectory planning, which hinders learning from mistakes. To address this issue, we designed an LLM instantiation Critic for feedback through verbal self-reflection to simulate experience accumulation. However, the verbal feedback from LLM alone is insufficient for environmental perception omissions, visual feedback, and decision-making accuracy. Therefore, we propose a VLM-based Evaluator to complement sensor-based perception, introducing a semantic alignment channel[69–73] that converts multi-view images into structured textual assessments. These assessments are then fed into re-planning, enabling timely hazard perception (e.g., missed obstacles with potential collision risk) thereby mitigating potential collision risks.

We propose the multimodal reward-value expressed in Eq. (16), to emulate human multisensory integration.

Verbal Reward. The Critic shown in Eq. (14) self-assesses the generated trajectory planning performed by the Actor, using predefined metrics (e.g., safety, comfort, and energy assumption).

Vision Reward. The Evaluator shown in Eq. (15) assesses the environment using VLM to identify and describe critical elements for the Actor, such as obstacles, road conditions, and corner cases.

$$r_{t,verbal} = Critic(x_t, \boldsymbol{\tau}_t), \tag{14}$$

$$r_{t,vision} = Evaluator(x_t, p_t, \boldsymbol{\tau}_t), \tag{15}$$

$$r_t = Concat(r_{t,verbal}, r_{t,vision}). \tag{16}$$

Here, p_t is a multiview image captured by vehicle-mounted cameras of the corresponding x_t. ReflectPlan evaluates r_t using PPL, filters out low-confidence cases (e.g., PPL > γ_{ppl}), and retains only reliable reflections to enhance policy robustness.

Dual-system planning scheme

Executing ReflectPlan in all scenarios is redundant. CogniDrive can orchestrate the switch between InstinctNav and ReflectPlan based on the complexity of scenarios, as shown in Fig. 6c. To automate and accurately determine the necessity for self-reflection, we proposed a straightforward indicator function $\mathbb{I}(\cdot)$ that leverages predefined rules and the VLM's image understanding and evaluative capabilities, defined as follows:

$$\mathbb{I}(x_t, \boldsymbol{\tau}_t) = \begin{cases} 1 & if\ Coll(\boldsymbol{\tau}_t; x_t)\ or\ E_t = 1 \\ 0 & otherwise \end{cases} \tag{17}$$

$$\{E_t, r_{t,vision}\} = Evaluator(x_t, p_t, \boldsymbol{\tau}_t). \tag{18}$$

Here, Coll(·) ∈ {0, 1} is determined by the overlap between the ego vehicle's 3D bounding box and the predicted 3D bounding boxes of surrounding obstacles, where the 3D boxes are provided by the perception and prediction modules. By incorporating the Evaluator flag E_t, Eq. (15) can be reformulated as Eq. (18), where $E_t \in \{0, 1\}$ specifies whether re-planning is required. The $r_{t,vision}$ will not be generated when the scenario is tractable, as indicated by $E_t = 0$.

In CogniDrive, the switching indicator $\mathbb{I}(\cdot)$ is implemented using primarily collision prediction and a binary VLM flag. This design choice emphasizes interpretability and robustness, ensuring that the gating mechanism remains transparent and reliable in safety-critical contexts. While alternative learned gating strategies may offer additional flexibility, they are considered outside the scope of the present study. Exploring such adaptive gating mechanisms will be an important direction for future work.

Dataset

For trajectory planning, the state $\mathcal{S}$ and historical trajectory $\mathcal{H}$ provided to the LLM were derived from the nuScenes dataset, which is a comprehensive urban dataset designed for AD research. The nuScenes dataset encompasses dynamic urban scenes with vehicles, pedestrians, and other road participants captured by high-resolution LiDAR, cameras, and radar. It provides extensive annotations and detailed environmental information.

The dialogue data D from nuScenes, used for the LLM, was organized following the approach in ref. 13. First, we implemented algorithms for perception, detection, and movement prediction using UniAD. Subsequently, we developed a program that converts these structured perceptions and trajectory predictions into unstructured textual data, making them interpretable using LLM. A total of 28,130 manually annotated human driving data were converted into dialogue form.

Corner case. CogniDrive was originally designed to mimic the human analytical thinking described by System 2 in dual-process theory to address the difficulty encountered by traditional methods in coping with corner cases. Therefore, we selected corner cases in nuScenes using the index acquired in CODA to validate the effectiveness of our approach in complex scenarios. CODA is a dataset that organizes and annotates corner cases from AD datasets such as nuScenes, enhancing testing for AD in complex scenarios.

Evaluation criteria

The essence of CogniDrive is to learn human driving data using the cognitive models of dual-process theory and deliberate practice theory. To assess these capabilities, we evaluated CogniDrive and baselines using fundamental AD metrics, including the L2 error between the predicted trajectory and ground truth, as well as the collision rate, calculated as the proportion of coordinates in the trajectory where collisions occurred. The L2 error was calculated using Eq. (10).

To better incorporate passenger considerations into the AD system, we proposed a comprehensive evaluation system that includes fundamental metrics, comfort, and energy efficiency. CogniDrive predicts discrete coordinate points. For the precise calculation of comfort and energy efficiency, these points were converted into a Bézier curve[39] that relied on polynomial interpolation based on a set of control points to define the shape of the curve. CogniDrive enhances safety and evaluates personalized metrics, such as comfort and energy efficiency, for advanced driving tasks. Comfort was measured by the rate of change in acceleration, aiming for minimal sudden movements in any direction, as expressed in Eq. (19). The energy efficiency was determined by the work performed during acceleration to maintain power, as in Eq. (20).

$$\text{Jerk} = \frac{1}{M}\sum_{i=1}^{M} \parallel B''(t+\delta_t) - B''(t) \parallel \tag{19}$$

$$\text{Energy} = \int_{\substack{t\in[0,1] \\ B'(t)\cdot B''(t)>0}} mB'(t)\cdot B''(t)\,dt \tag{20}$$

where M represents the number of samples. We approximated the rate of change in acceleration using these M samples. The variable δ_t was employed to compute the rate of change in acceleration at each sampling point through differential calculations.

The Jerk and Energy metrics are simplified approximations for relative performance comparison under uniform simulation. The vehicle mass is normalized to unit weight, while the effects of gravity, friction, and air resistance are omitted. Although this simplification reduces physical fidelity, it ensures that performance differences primarily reflect algorithmic capabilities rather than environment-induced variability, thereby improving the consistency and efficiency of benchmarking.

Comparison baselines

The performance of our method was compared with the following conventional and LLM-based methods.

• **Conventional Methods**. FF[2] supports motion planning by using self-supervised learning to forecast future free space, enabling collision avoidance without reliance on object detection or tracking and eliminating the need for manual annotations. EO[3] uses geometric occupancy maps with differentiable raycasting to forecast future scenes by aligning occupancy predictions with future LiDAR sweeps for self-supervised learning. ST-P3[4] proposes a spatiotemporal feature learning scheme aimed at deriving a set of more representative

features for perception, prediction, and planning tasks. UniAD[5] presents a unified framework that follows a planning-oriented philosophy. Instead of a standalone modular design and multitask learning, UniAD hierarchically integrates a series of tasks, including perception, prediction, and planning. VAD[6] offers an end-to-end unified vectorized paradigm. It models the driving scene as a fully vectorized representation, eliminating the need for computationally intensive, dense, rasterized representations and hand-designed post-processing steps. MT[48] leverages a transformer-based motion forecasting framework that models trajectory prediction via global intention localization and local movement refinement.

• **RL-based Methods**. SA-TD3[51] applies a single-agent Twin Delayed Deep Deterministic Policy Gradient framework to address the challenge of safe and efficient navigation through high-density T-intersections. MMI-DQN[52] adopts the Deep Q-Network framework to tackle robust navigation under uncertainty in mixed-traffic environments with diverse vehicle types and pedestrians.

• **LLM/VLM-based Methods**. GPT-Driver[13] converts GPT-3.5 into a motion planner by treating motion planning as a language-modeling task, using language descriptions to generate trajectories and improving numerical reasoning using a prompting-reasoning-fine-tuning strategy. Agent-Driver[14] transforms AD by using LLMs as cognitive agents for decision-making, task planning, and motion planning, integrating a tool library and cognitive memory to significantly improve performance and interpretability. DriveGPT4-V2[74] formulates autonomous driving as a language-driven planning problem, leveraging large language models to infer high-level intents and constraints from scene context and generate motion plans under open-loop evaluation, demonstrating improved generalization across large-scale real-world driving datasets.

A VLM, leveraging its built-in LLM, can also achieve reasoning capabilities of inferring scene conditions (e.g., slippery roads due to rain), compensating for perceptual gaps (e.g., a truck overturned on the side). DriveVLM[75] integrates a VLM into AD, combining visual and linguistic modalities for efficient scene understanding and decision-making. OpenEMMA[49] employs a VLM to process raw camera sensor data and generate future trajectories and planning decisions in natural language. DiMA[76] distills high-level reasoning and decision-making knowledge from a powerful multimodal large language model into a compact student driving policy, enabling efficient and robust closed-loop execution under dynamic environments, particularly in long-tail driving scenarios.

Fine-tuning details

The LLM employed in this study is the instruction-tuning version of Qwen2.5 (Qwen2.5-1.5B-Instruct), a dense transformer model trained with instruction tuning and RLHF, achieving strong performance on open benchmarks[44].

The model training was configured for five epochs. The Qwen model was fine-tuned using parameter efficient fine-tuning facilitated by LoRA, with specific rank parameters set at $r = 16$, `lora_alpha` = 32, and `lora_dropout` = 0.05, with no quantization applied to maintain the integrity of the model's original performance characteristics. For model generation, a greedy decoding strategy was used with a `temperature` of 0.3, a maximum of 4096 new tokens per prompt (`max_new_tokens` = 128), and a nucleus sampling cutoff (`top_p` = 1.0), without employing a repetition penalty to promote natural variability in the generated text. Due to the high training cost of LLMs, hyperparameters were empirically chosen through validation performance rather than extensive search.

The model was fine-tuned using eight A100 (80GB) GPUs and took approximately 12 hours. The fine-tuned model weights, along with comprehensive code and detailed instructions for downloading the weights, initializing the models, and performing inference tasks, are available in the repository at URL.

Data availability

The data used in this study are publicly available from the official project websites of the nuScenes dataset https://www.nuscenes.org/[40] and the Waymo Open Dataset https://waymo.com/open/[41], subject to their respective licensing terms. Based on these datasets, we generate derived textual data for large language model prompting and evaluation, including structured descriptions, reasoning traces, and experience representations. These derived data do not contain raw sensor measurements and cannot be used to reconstruct the original datasets. The processed data generated in this study are publicly available in the archived Zenodo repository at https://doi.org/10.5281/zenodo.19179549[77]. The data supporting the findings of this study, including experimental results and evaluation statistics, are available within the paper and its Supplementary Information. Source data are provided with this paper.

Code availability

The source code used in this study is publicly available on GitHub at https://github.com/2351672564/CogniDrive. An archived version of the code is available on Zenodo at https://doi.org/10.5281/zenodo.19179549[77]. An executable reproduction package is available through Code Ocean at 10.24433/CO.5837541.v1.

References

1. Kunz, F. et al. Autonomous driving at Ulm University: A modular, robust, and sensor-independent fusion approach. In 2015 IEEE intelligent vehicles symposium, 666–673 (2015).
2. Hu, P., Huang, A., Dolan, J., Held, D., & Ramanan, D. Safe local motion planning with self-supervised freespace forecasting. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 12732–12741 (2021).
3. Khurana, T. et al. Differentiable raycasting for self-supervised occupancy forecasting. In European Conference on Computer Vision, 353–369 (2022).
4. Hu, S. et al. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In European Conference on Computer Vision, 533–549 (2022).
5. Hu, Y. et al. Planning-oriented autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 17853–17862 (2023).
6. Jiang, B. et al. Vad: Vectorized scene representation for efficient autonomous driving. In Proceedings of the IEEE/CVF International Conference on Computer Vision 8340–8350 (2023).
7. Lee, E. J., Kim, J. Y., Kim, Y. B. & Kim, S. K. Microwave-transparent metallic metamaterials for autonomous driving safety. *Nat. Commun.* **15**, 4516 (2024).
8. Fei, N. et al. Towards artificial general intelligence via a multimodal foundation model. *Nat. Commun.* **13**, 3094 (2022).
9. Zhou, Y., Tan, G., & Gou, C. Hierarchical home action understanding with implicit and explicit prior knowledge. In IEEE International Conference on Acoustics, Speech and Signal Processing, 4015–4019 (2024).
10. Lin, B. Y. et al. Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. *Adv. Neural Inf. Process. Syst.* **36**, 23813–23825 (2024).
11. Xi, Z. et al. The rise and potential of large language model based agents: A survey. *Sci. China Inf. Sci.* **68**, 121101 (2025).
12. Zhao, W. X. et al. A survey of large language models. https://arxiv.org/abs/2303.18223 (2023).
13. Mao, J., Qian, Y., Zhao, H., & Wang, Y. Gpt-driver: Learning to drive with GPT. https://arxiv.org/abs/2310.01415 (2023).
14. Mao, J., Ye, J., Qian, Y., Pavone, M., & Wang, Y. A language agent for autonomous driving. https://arxiv.org/abs/2311.10813 (2023).

15. Ouyang, L. et al. Training language models to follow instructions with human feedback. *Adv. neural Inf. Process. Syst.* **35 2022**, 27730–27744 (2023).
16. Sun, H. et al. Determlr: Augmenting LLM-based logical reasoning from indeterminacy to determinacy. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, 9828-9862 (2023).
17. Havrilla, A. et al. Glore: When, where, and how to improve LLM reasoning via global and local refinements. *Proc. 41st Int. Conf. Mach. Learn. (ICML)* **235**, 17719–17733 (2024).
18. Evans, J. S. B. In two minds: dual-process accounts of reasoning. *Trends Cogn. Sci.* **7**, 454–459 (2003).
19. Konda, V. R., & Tsitsiklis, J. N. Actor-citic agorithms. In Proceedings of the 13th International Conference on Neural Information Processing Systems. 1008–1014 (1999).
20. Ji, Z. et al. Towards mitigating LLM hallucination via self reflection. *Find. Assoc. Computational Linguist.: EMNLP* **2023**, 1827–1843 (2023).
21. Renze, M., & Guven, E. Self-Reflection in LLM Agents: Effects on Problem-Solving Performance. In Proceedings of the IEEE International Conference on Foundation and Large Language Models (FLLM), 516–525 (2024).
22. Ericsson, K. A., Krampe, R. T. & Tesch-Römer, C. The role of deliberate practice in the acquisition of expert performance. *Psychological Rev.* **1003**, 363 (1993).
23. Wörmann, J. et al. Knowledge augmented machine learning with applications in autonomous driving: A survey. https://arxiv.org/abs/2205.04712 (2022).
24. Onat, N. C. et al. Rebound effects undermine carbon footprint reduction potential of autonomous electric vehicles. *Nat. Commun.* **14**, 6258 (2023).
25. Wei, J. et al. Chain-of-thought prompting elicits reasoning in large language models. *Adv. neural Inf. Process. Syst.* **35**, 24824–24837 (2022).
26. Wang, X. et al. Self-consistency improves chain of thought reasoning in language models. In The Eleventh International Conference on Learning Representations. https://openreview.net/forum?id=1PL1NIMMrw (2023).
27. Yao, S. et al. Tree of thoughts: deliberate problem solving with large language models. *Adv. neural Inf. Process. Syst.* **36**, 11809–11822 (2023).
28. Besta, M. et al. Graph of thoughts: Solving elaborate problems with large language models. *Proc. AAAI Conf. Artif. Intell.* **38**, 17682–17690 (2024).
29. Torabi, F., Warnell, G., & Stone, P. Behavioral Cloning from Observation. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, 4950-4957 (2018).
30. Lewis, P. et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Adv. Neural Inf. Process. Syst.* **33**, 9459–9474 (2020).
31. Ozay, N. Using control synthesis for falsification and corner case generation. In Proceedings of the 1st International Workshop on Verification of Autonomous & Robotic Systems, 1–2 (2021).
32. Zhang, P. et al. Vinvl: Revisiting visual representations in vision-language models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 5579–5588 (2021).
33. Zhou, K., Yang, J., Loy, C. C., & Liu, Z. Conditional prompt learning for vision-language models. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 16816–16825 (2022).
34. Zhou, J. et al. Pre-trained multimodal large language model enhances dermatological diagnosis using SkinGPT-4. *Nat. Commun.* **15**, 5649 (2024).
35. Guo, J., Kurup, U. & Shah, M. Is it safe to drive? An overview of factors, metrics, and datasets for driveability assessment in autonomous driving. *IEEE Trans. Intell. Transportation Syst.* **21**, 3135–3151 (2019).
36. Cheng, C. H., Knoll, A., & Liao, H. C. Safety metrics for semantic segmentation in autonomous driving. In 2021 IEEE International Conference on Artificial Intelligence Testing, 57–64. IEEE (2021).
37. Soares, N. et al. A review on current advances in the energy and environmental performance of buildings towards a more sustainable built environment. *Renew. Sustain. Energy Rev.* **77**, 845–860 (2017).
38. Danca, P., Vartires, A. & Dogeanu, A. An overview of current methods for thermal comfort assessment in vehicle cabin. *Energy Procedia* **85**, 162–169 (2016).
39. Boehm, W. & Muller, A. On de casteljau's algorithm. *Computer Aided Geometric Des., 16* **7**, 587–605 (1999).
40. Caesar, H. et al. Nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 11621–11631. https://doi.org/10.1109/CVPR42600.2020.01164. (2020).
41. Sun, P. et al. Scalability in perception for autonomous driving: Waymo open dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2446–2454. https://doi.org/10.1109/CVPR42600.2020.00252. (2020).
42. Li, K. et al. Coda: A real-world road corner case dataset for object detection in autonomous driving. In European Conference on Computer Vision, 406–423 (2022).
43. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., & Koltun, V. CARLA: An open urban driving simulator. In Proceedings of the 1st Annual Conference on Robot Learning, 1–16 (2017).
44. Qwen Team. Qwen2.5: A party of foundation models. Qwen. https://qwenlm.github.io/blog/qwen2.5/ (2024).
45. Bogdoll, D. et al. Description of corner cases in automated driving: Goals and challenges. In Proceedings of the IEEE/CVF International Conference on Computer Vision. 1023–1028 (2021).
46. Bolte, J. A., Bar, A., Lipinski, D., & Fingscheidt, T. (2019, June). Towards corner case detection for autonomous driving. In 2019 IEEE Intelligent vehicles symposium. 438–445.
47. Sun, H., Feng, S., Yan, X. & Liu, H. X. Corner case generation and analysis for safety assessment of autonomous vehicles. *Transportation Res. Rec.* **2675**, 587–600 (2021).
48. Ngiam, J. et al. Scene transformer: A unified architecture for predicting future trajectories of multiple agents. In International Conference on Learning Representations. https://openreview.net/forum?id=Wm3EA5OlHsG (2022).
49. Hwang, J. J. et al. EMMA: End-to-End Multimodal Model for Autonomous Driving. In Transactions on Machine Learning Research. https://openreview.net/forum?id=kH3t5lmOU8 (2025).
50. Kaelbling, L. P., Littman, M. L. & Moore, A. W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **4**, 237–285 (1996).
51. Zhao X., Zheng S. Robot trajectory planning optimization algorithm based on improved td3 algorithm. In International Conference on Intelligent Communication and Networking (ICN), 281–285. IEEE (2023).
52. Ronecker, M. P., & Zhu, Y. Deep Q-network based decision making for autonomous driving. In 2019 3rd international conference on robotics and automation sciences, 154–160 (2019).
53. Basheer, I. A. & Hajmeer, M. Artificial neural networks: fundamentals, computing, design, and application. *J. microbiological methods* **43**, 3–31 (2000).
54. Devlin, J., Chang, M. W., Lee, K. & Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proc. 2019 Conf. North Am. chapter Assoc. computational Linguist.: Hum. Lang. Technol.* **1**, 4171–4186 (2019).
55. Yan, S. Q., Gu, J. C., Zhu, Y., & Ling, Z. H. Corrective retrieval augmented generation. In Proceedings of the International Conference on Learning Representations. https://openreview.net/forum?id=JnWJbrnaUE (2024).

56. Fang, X. et al. A method for multiple-sequence-alignment-free protein structure prediction using a protein language model. *Nat. Mach. Intell.* **5**, 1087–1096 (2023).
57. Yang, J. et al. A survey of few-shot learning in smart agriculture: developments, applications, and challenges. *Plant Methods* **18**, 28 (2022).
58. Meta. Introducing Meta Llama 3: The Most Capable Openly Available LLM to Date. https://ai.meta.com/blog/meta-llama-3/ (2024).
59. Enis, M., & Hopkins, M. From LLM to NMT: Advancing Low-Resource Machine Translation with Claude. https://arxiv.org/abs/2404.13813 (2024).
60. Wu, Y., Hu, X., Fu, Z., Zhou, S., & Li, J. GPT-4o: Visual perception performance of multimodal large language models in piglet activity understanding. https://arxiv.org/abs/2406.09781 (2024).
61. Openai. Introducing OpenAI O1-Preview. https://openai.com/index/introducing-openai-o1-preview/ (2024).
62. Brown, T. B. et al. Language models are few-shot learners. *InAdvances Neural Inf. Process. Syst.* **33**, 1877–1901 (2020).
63. Hu, E. J. et al. LoRA: Low-Rank Adaptation of Large Language Models. In International Conference on Learning Representations. https://openreview.net/forum?id=nZeVKeeFYf9 (2022).
64. Wei, J. et al. Finetuned language models are zero-shot learners. In International Conference on Learning Representations. https://openreview.net/forum?id=gEZrGCozdqR (2022).
65. Schulman, L. S. Techniques and applications of path integration. Courier Corporation (2012).
66. Shao, Z. et al. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. https://arxiv.org/abs/2402.03300 (2024).
67. Minderhoud, M. M. & Bovy, P. H. Extended time-to-collision measures for road traffic safety assessment. *Accid. Anal. Prev.* **33**, 89–97 (2001).
68. Johnson, J., Douze, M. & Jégou, H. Billion-scale similarity search with GPUs. *IEEE Trans. big data* **7**, 535–547 (2019).
69. Petrides, M. On the evolution of polysensory superior temporal sulcus and middle temporal gyrus: A key component of the semantic system in the human brain. *J. Comp. Neurol.* **531**, 1987–1995 (2023).
70. Wu, T. et al. Semantic alignment for multimodal large language models. In Proceedings of the 32nd ACM International Conference on Multimedia, 3489–3498 (2024).
71. Philion, J., & Fidler, S. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In European conference on computer vision, 194–210 (2020).
72. Shazeer, N. et al. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In International Conference on Learning Representations. https://openreview.net/forum?id=B1ckMDqlg (2017).
73. Vaswani, A. et al. Attention is all you need. In Advances in Neural Information Processing Systems, 30 (2017).
74. Xu, Z. et al. Drivegpt4-v2: Harnessing large language model capabilities for enhanced closed-loop autonomous driving. In Proceedings of the Computer Vision and Pattern Recognition Conference. 17261–17270 (2025).
75. Tian, X. et al. Drivevlm: The convergence of autonomous driving and large vision-language models. https://arxiv.org/abs/2402.12289 (2024).
76. Hegde, D. et al. Distilling multi-modal large language models for autonomous driving. In Proceedings of the Computer Vision and Pattern Recognition Conference, 27575–27585. (2025).
77. Zhang X. et al. Autonomous Driving System based on Dual Process Theory and Deliberate Practice Theory. CogniDrive. https://doi.org/10.5281/zenodo.19179549 (2026).
78. Shazeer, N. et al. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer. In International Conference on Learning Representations. https://openreview.net/forum?id=B1ckMDqlg&noteId=B1ckMDqlg (2017).

Acknowledgements

This work was supported in part by National Natural Science Foundation of China (No. U25B20273, H.M.), National Natural Science Foundation of China (No. 62172036, T.H.), Beijing Natural Science Foundation (No. L257003, H.M.), National Natural Science Foundation of China (No. 62227801, H.M.), and National Science and Technology Major Project (No. 2022ZD0116305, T.H.).

Author contributions

X.Z. and T.H. conceived the study and designed the framework. X.Z. and J.L. developed the CogniDrive framework and conducted the InstinctNav and ReflectPlan experiments. Q.L. and Q.Z. contributed to the theoretical grounding and formulation of the dual-process and deliberate practice components. M.L., K.L. and S.H. carried out the open-loop and closed-loop evaluations. J.C. and H.M. supervised the project and provided resources. X.Z. and T.H. wrote the manuscript with input from all authors. H.M. and T.H. served as the corresponding authors and were responsible for project administration.

Competing interests

The authors declare no competing interests.

Additional information

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s41467-026-72030-6.

Correspondence and requests for materials should be addressed to Tianyu Hu or Huimin Ma.

Peer review information *Nature Communications* thanks Dongsu Lee and the other, anonymous, reviewer(s) for their contribution to the peer review of this work. A peer review file is available.

Reprints and permissions information is available at http://www.nature.com/reprints

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2026