Distill to Think, Foresee to Act: Cognitive-Physical Reinforcement Learning for Autonomous Driving

Yang Wu[1] **Qiang Meng**[†] **Zhaojiang Liu**[3] **Youquan Liu**[4] **Jian Yang**[1,2] **Jin Xie**[2,*]

[1]PCA Lab, Nanjing University of Science and Technology
[2]Nanjing University [3]Shanghai Jiao Tong University [4]Fudan University
wuyang98@njust.edu.cn csjxie@nju.edu.cn

Abstract

Current end-to-end autonomous driving models are fundamentally constrained by the behavioral cloning ceiling of imitation learning. While reinforcement learning offers a path to smarter autonomy, it demands two missing pieces of infrastructure: (1) a cognitive foundation that understands traffic semantics and driving intent, and (2) a foresighted physical environment that can anticipate the consequences of candidate actions. To this end, we propose **CoPhy**, a **Co**gnitive-**Phy**sical reinforcement learning framework for autonomous driving. To *distill to think*, we distill VLM knowledge into the BEV encoder and then discard the VLM entirely, so that at inference time the VLM is not required while the plug-in interface is kept for language command inputs. To *foresee to act*, we build an auto-regressive BEV world model that explicitly predicts future semantic maps conditioned on candidate actions, serving as an interpretable physical sandbox from which safety metrics are directly derived. Building upon this dual infrastructure, we optimize the driving policy via GRPO with a novel dual-reward mechanism: a *physical* reward derived from BEV rollouts enforces hard safety constraints, while a *cognitive* reward from a language-aligned scorer ensures intent compliance. Extensive experiments demonstrate that CoPhy not only achieves state-of-the-art results on NAVSIM v1 and v2 benchmarks, but also enables safer driving via cognitively informed scene compliance and flexible intent control through user-defined language instructions. Project page: https://wuyang98.github.io/CoPhy/.

1 Introduction

End-to-end (E2E) autonomous driving (AD) has advanced rapidly in recent years [6, 12, 4], learning driving policies directly from raw sensor inputs. Yet the dominant training paradigm, behavior cloning on expert demonstrations, introduces a fundamental bottleneck: models are largely confined to fitting predefined rules and expert trajectories [4, 66], severely restricting their safety and adaptability in open-world scenarios. A key limitation is the lack of cognitive scene understanding. Existing E2E methods [6, 12, 16, 24, 29] rely on pure visual features, skewing optimization towards short-sighted trajectory fitting and leaving them unable to comprehend complex traffic scenarios or incorporate driver intentions. Vision-Language Models (VLMs) [38, 42, 15, 71, 34] offer a promising pathway for scene reasoning. However, their prohibitive computational cost hinders edge deployment. Moreover, current VLM-based approaches over-rely on textual capabilities, either reducing driving to question answering or naively attaching an action head to the language model. Consequently, they overlook safety issues at the execution level, leaving semantic intent and physical execution misaligned, a limitation further compounded by reasoning hallucinations. As highlighted in Fig. 1, isolating cognitive semantics from physical rules inevitably leads to critical driving failures, such as ignoring

[†]Independent researcher. * Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

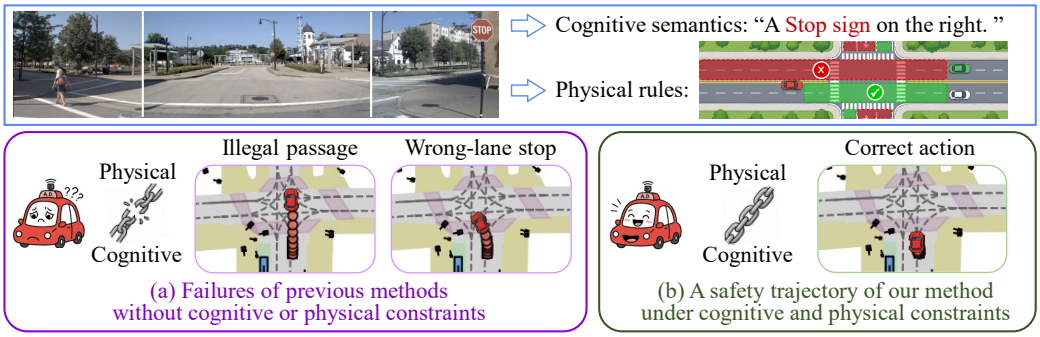


Figure 1: Safe driving requires jointly reasoning about *cognitive* cues, such as traffic rules and driver intent, and *physical* constraints, such as spatial feasibility and collision risk. Neglecting the former leads to semantic violations (e.g., ignoring a STOP sign), while neglecting the latter leads to spatial violations (e.g., stopping in the oncoming lane). **(a)** Previous methods isolate the two and therefore frequently produce such failures. **(b)** Our method enforces both constraints jointly, so that the output trajectory follows the cognitive semantics and obeys reasonable physical rules.

traffic signals or violating spatial constraints. This calls for a lightweight mechanism that equips the E2E model with rich cognitive priors without online VLM overhead.

Yet cognitive understanding alone is insufficient for safe driving. Regardless of how scene semantics are acquired, existing paradigms lack explicit physical-world reasoning to anticipate risks, such as collisions and lane violations. Similar to human drivers who mentally simulate outcomes before acting, an autonomous system should be able to imagine the consequences of candidate actions and proactively avoid dangerous trajectories. While some methods [24, 61, 62, 43] utilize implicit world models or simulators [8] to predict future latent features, they fundamentally detach from interpretable physical states, failing to provide actionable safety assessment. This motivates an explicit world model that rolls out future BEV states as a "what-if" physical sandbox for trajectory evaluation.

Furthermore, even with cognitive and physical foresight in place, models trained purely via imitation learning remain trapped within the behavioral cloning paradigm: the learned policy can only replicate the behaviors it has observed, without the ability to discover alternative strategies that may be equally or more effective [4]. This limitation manifests as distributional brittleness: when confronted with out-of-distribution scenarios, the policy lacks the exploratory experience needed to generalize. Reinforcement learning (RL) offers a principled mechanism to overcome this ceiling by allowing the agent to explore beyond expert data and optimize its policy through trial-and-error interaction. However, applying RL to E2E driving poses distinct challenges: it requires both a faithful environment for rollouts and meaningful reward signals. While recent approaches [17, 27, 71] have begun exploring RL for driving, a unified paradigm that integrates soft cognitive preferences (e.g., intent compliance) with hard physical constraints (e.g., collision avoidance) remains absent.

In this work, we propose **CoPhy**, a cognitive-physical learning framework that addresses all the above challenges in a unified architecture. To bridge the cognitive gap, we introduce a cognitive prior distillation that distills rich offline language priors into BEV representations, endowing the E2E model with scene understanding at zero additional inference cost. To enable physical foresight, we establish a physical world model rollout that explicitly simulates future spatiotemporal evolutions conditioned on candidate actions, serving as a physical sandbox for "what-if" trajectory rollouts and safety assessment. To overcome the limitations of imitation learning, we propose cognitive-physical policy optimization. Driven by Group Relative Policy Optimization (GRPO), it explores the policy space via a dual-reward mechanism that synergizes hard physical constraints derived from BEV rollouts with soft cognitive preferences from a distilled language-aligned scorer. By harmonizing these components, **CoPhy** achieves a driving policy that is cognitively aware, physically safe, and intention-aligned. Extensive experiments on NAVSIM v1 and v2 benchmarks demonstrate that **CoPhy** sets a new state-of-the-art, achieving 91.4 PDMS and 86.1 EPDMS respectively, with leading safety-critical metrics across all sub-scores. Moreover, the distilled cognitive channel naturally enables intent-controllable driving via user language commands. Our main contributions are summarized as:

- We propose **CoPhy**, a cognitive-physical framework for E2E AD. It grounds cognitive priors into spatial perception at zero inference cost via a cognitive prior distillation, while establishing an explicit "what-if" sandbox through a physical world model rollout.

- To fully exploit this predictive sandbox and break the traditional imitation ceiling, we introduce a cognitive-physical policy optimization that leverages a GRPO-driven dual-reward mechanism to balance hard physical constraints with soft cognitive preferences.
- Extensive experiments on NAVSIM v1 and v2 demonstrate that **CoPhy** achieves state-of-the-art performance with 91.4 PDMS and 86.1 EPDMS, alongside leading safety-critical sub-scores and an inherent intent-controllable driving capability.

2 Related Works

End-to-end Autonomous Driving. End-to-end autonomous driving advances rapidly by integrating multi-task perception and planning into unified architectures [39, 12, 16]. Recent innovations span diverse algorithmic paradigms. For instance, generative and diffusion-based models [67, 29, 25, 59] significantly enhance trajectory robustness, while world models [22, 24, 57, 65, 54] model future scene evolution to support driving prediction and planning. Complementary sensor-level generation methods synthesize diverse weather conditions in LiDAR point clouds [56]. Concurrently, efforts to optimize computational efficiency introduce state-space models like Mamba [64] and multi-teacher knowledge distillation [28]. Nevertheless, driving policies trained through behavioral cloning remain fundamentally bottlenecked by the inherent brittleness and poor interpretability characteristic of pure imitation learning.

VLMs for Autonomous Driving. Vision-Language Models significantly elevate autonomous driving via robust semantic understanding, predominantly following two technical routes. The first emphasizes semantic reasoning, leveraging visual question answering [18, 38, 42, 48, 15] and chain-of-thought processes [52, 17, 31] for high-level decision-making. The second paradigm targets direct trajectory generation by appending action heads to foundational models [13, 70, 45, 23], unifying diverse tasks within a single linguistic space. Beyond policy learning, language also enables controllable scene synthesis, such as text-guided LiDAR generation [55]. However, these language-centric driving architectures largely overlook the interplay between cognitive semantics and rigid physical constraints. While recent efforts [21, 26] infuse spatial awareness via positional encoding, they critically lack an explicit physical rollout mechanism to verify safety consequences.

Reinforcement Learning in Autonomous Driving. Following its success in foundational language models [10] and robotic systems [49], reinforcement learning aligns autonomous driving policies. DriveDPO [44] explicitly leverages it to unearth robust strategies, while broader vision-language-action frameworks [17, 43, 27] similarly bridge semantic reasoning with strategic planning. For closed-loop exploration, predictive approaches [47, 62] derive intrinsic rewards from scene imaginations, while language-guided frameworks [71, 14] refine low-level execution using semantic feedback. Crucially, these paradigms isolate physical constraints from cognitive preferences, necessitating a unified dual-reward formulation to jointly optimize deterministic safety and intent compliance.

3 Methodology

The core idea of CoPhy is to build a cognitive-physical learning framework, jointly ensuring physical safety and cognitive scene understanding for autonomous driving. As illustrated in Fig. 2, it comprises three modules: (1) cognitive prior distillation that injects language priors into the end-to-end model, (2) physical world model rollout that enables spatiotemporal reasoning of driving scenes, and (3) cognitive-physical policy optimization that breaks the imitation learning ceiling via dual rewards. We detail the training in Secs. 3.1 and 3.2 and the inference pipeline in Sec. 3.3.

3.1 Cognitive-Physical Imitation Learning

Our end-to-end model takes multi-modal sensor inputs $\mathcal{V} = \{\mathcal{V}_{\text{RGB}}, \mathcal{V}_{\text{LiDAR}}\}$ and projects them via a multi-modal encoder $\mathcal{E}_{\boldsymbol{\theta}}$ to obtain the BEV state $\mathbf{B}_0 = \mathcal{E}_{\boldsymbol{\theta}}(\mathcal{V})$. Separately, N kinematic trajectory prototypes $\mathcal{T}_p$ are pre-extracted by clustering expert demonstrations [5, 6, 24], constraining the action space within physically feasible bounds. A trajectory refiner π_θ then yields candidate trajectories $\hat{\mathcal{T}} = \mathcal{T}_p + \pi_\theta(\texttt{TE}(\mathcal{T}_p), \mathbf{B}_0)$, where TE is the trajectory encoder. We will later augment this formulation with cognitive conditioning in Eq. (4). The candidates are supervised by the expert trajectory $\mathcal{T}$ via

$$\mathcal{L}_{\text{Traj}} = \|\hat{\mathcal{T}} - \mathcal{T}\|_1. \tag{1}$$

Distill to Think: Cognitive Prior Distillation. Vision-Language Models (VLMs) excel at high-order reasoning, benefiting cognitive understanding in current VLA methods. However, their prohibitive

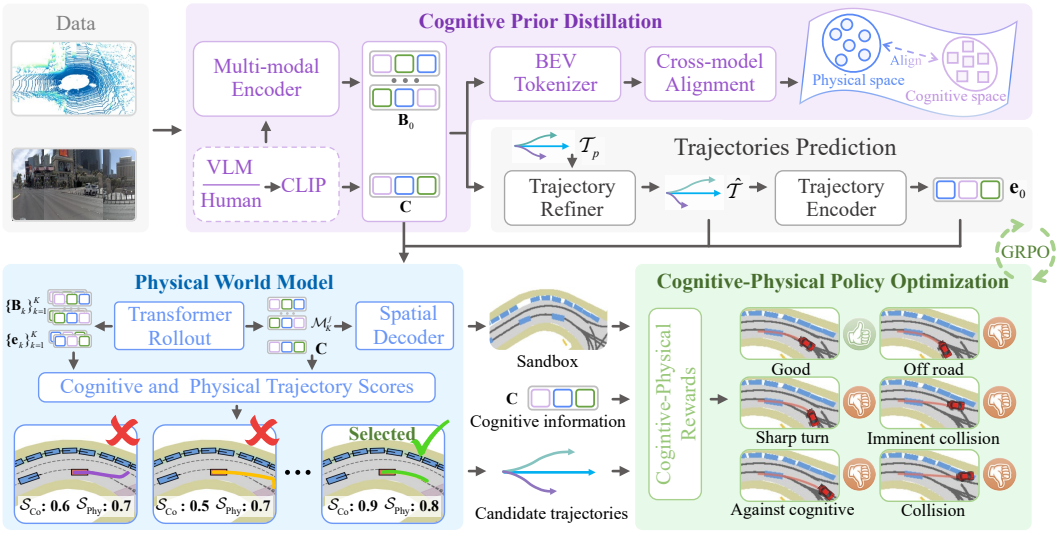


Figure 2: Overview of CoPhy. Multi-modal data are encoded into BEV state $\mathbf{B}_0$. **(a)** The cognitive prior distillation module aligns $\mathbf{B}_0$ with VLM-derived cognitive feature $\mathbf{C}$ via cross-modal distillation, and injects $\mathbf{C}$ into the trajectory refiner to produce cognitively-aware candidates. **(b)** The physical world model rollout module auto-regressively rolls out future BEV states over K steps as a physical sandbox; trajectories are then scored along cognitive and physical axes via hierarchical selection. **(c)** The cognitive-physical policy optimization optimizes the policy with dual cognitive-physical rewards.

computational cost hinders edge deployment in autonomous driving systems. Therefore, we distill the VLM's cognitive ability into the end-to-end model during training and discard the VLM entirely at inference. Specifically, we fine-tune InternVL3-8B [72] via LoRA [11] across diverse driving VQA datasets [35, 40, 60, 48, 36], making it a domain-expert reasoning engine optimized to output cognitive priors such as traffic signs, road topologies, and driver intentions. Conditioned on visual inputs $\mathcal{V}$ and textual prompt Q, this domain-specific cognition is supervised by

$$\mathcal{L}_{\texttt{VLM}} = -\sum_{l=1}^{L} \log P_{\boldsymbol{\xi}}(y_l \mid y_{<l}, \mathcal{V}, Q), \tag{2}$$

where y_l is the l-th token of the target sequence and $\boldsymbol{\xi}$ denotes the LoRA parameters.
Once trained, the VLM is frozen and used to generate textual descriptions during IL training, which are encoded by a CLIP text encoder [41] into the cognitive prior $\mathbf{C}$. To bridge the cognitive-physical gap, we force $\mathcal{E}_{\boldsymbol{\theta}}$ to internalize these language priors via a cross-modal cosine distance objective:

$$\mathcal{L}_{\texttt{Distill}} = 1 - \texttt{BE}(\mathbf{B}_0) \cdot \mathbf{C} / (\|\texttt{BE}(\mathbf{B}_0)\|_2 \|\mathbf{C}\|_2), \tag{3}$$

where BE is a BEV tokenizer to extract the cognitive feature from the BEV state $\mathbf{B}_0$.

In addition to the feature-level distillation, we further inject $\mathbf{C}$ into the trajectory refiner by

$$\hat{\mathcal{T}} = \mathcal{T}_p + \pi_\theta(\texttt{TE}(\mathcal{T}_p), \mathbf{B}_0, \mathbf{C}). \tag{4}$$

This not only enriches trajectories with cognitive awareness, but also opens an interface for intent-controllable driving via language commands, an underexplored capability of great practical value.

Foresee to Act: Physical World Model Rollout. While cognitive distillation enriches the model with semantic understanding, it alone cannot address the core safety challenges of autonomous driving. Equally critical is the ability to imagine future states and anticipate potential dangers, enabling the model to proactively avoid unsafe trajectories. Moreover, breaking the imitation learning ceiling requires closed-loop exploration, where the agent reasons about consequences of diverse actions, a core principle underlying reinforcement learning. Thus, we introduce an auto-regressive world model that serves two purposes: (1) providing a physical sandbox to assess candidate trajectories via future rollouts, and (2) acting as the environment for reinforcement learning-based policy optimization.

Unlike conventional methods constrained by external simulators or purely latent features [43, 24, 62], our world model operates directly within the BEV space, auto-regressively rolling out futures

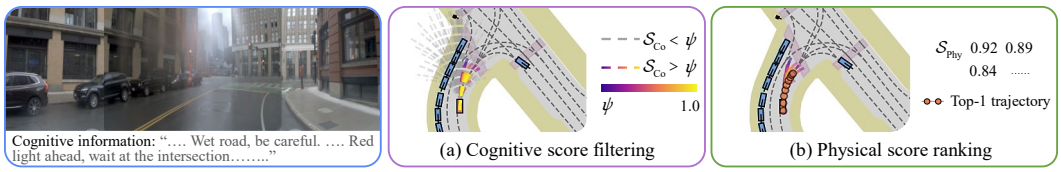


Figure 3: Hierarchical trajectory selection. **(a)** Candidates passing the cognitive threshold $\hat{s}_{\texttt{Co}} \geq \psi$ are retained and subsequently **(b)** ranked by the physical score $\hat{s}_{\texttt{Phy}}$ to determine the optimal trajectory.

conditioned on both spatial geometries and cognitive priors. Specifically, a Transformer-based [51] world model takes the current BEV state $\mathbf{B}_0$, the cognitive feature $\mathbf{C}$, and N initial ego features $\mathbf{e}_0 = \texttt{TE}(\hat{\mathcal{T}})$ as input. To predict future states and actions, the model recursively unrolls the transitions:

$$(\mathbf{B}_k, \mathbf{e}_k) = \texttt{WM}(\mathbf{B}_{k-1}, \mathbf{e}_{k-1}, \mathbf{C}), \tag{5}$$

where $k \in \{1, \cdots, K\}$ denotes the future time step. Since optimizing all future states is computationally expensive during training, we randomly sample an index $j \sim \mathcal{U}(1, N)$ in each iteration and use a lightweight spatial decoder to project the corresponding state $\mathbf{B}_K^j$ into dense BEV semantic maps $\hat{\mathcal{M}}_K^j \in \mathbb{R}^{H \times W \times C_c}$, where C_c denotes the category count. Using nuPlan [2] to generate ground-truth maps $\mathcal{M}_K^j$ [24], we train the physical world model by supervising the predictions via Focal Loss [30]:

$$\mathcal{L}_{\texttt{WM}} = \texttt{FocalLoss}(\hat{\mathcal{M}}_K^j, \mathcal{M}_K^j). \tag{6}$$

Cognitive and Physical Trajectory Scores. With trajectory generation and cognitive-physical representations in place, a key question remains: how to score the N predicted trajectories. We propose to score trajectories along two complementary axes, physical safety and cognitive compliance, and then select the optimal one.

The first criterion is the physical score [24], which relates to the distance d to the expert trajectory, yielding $s_{\texttt{Phy}} \propto \exp(-d)$. To predict this score, we concatenate the BEV states $\{\mathbf{B}_k\}_{k=1}^K$ along the temporal dimension and compact them into a BEV vector $\mathbf{b}$ via a few convolutional layers. The vector, along with the K-step ego features $\{\mathbf{e}_k\}_{k=1}^K$, is projected into the physical score $\hat{s}_{\texttt{Phy}} \in \mathbb{R}^{B \times N}$ via MLP and softmax. This explicitly favors expert-aligned behaviors and is optimized via cross-entropy:

$$\hat{s}_{\texttt{Phy}} = \texttt{Softmax}\big(\texttt{MLP}([\mathbf{b};\ \mathbf{e}])\big), \quad \mathcal{L}_{\texttt{Phy}} = -s_{\texttt{Phy}} \log(\hat{s}_{\texttt{Phy}}). \tag{7}$$

The second criterion is the cognitive alignment between $\mathbf{C}$ and the candidate trajectory $\hat{\mathcal{T}}$, defined as the cosine similarity after projecting $\mathbf{C}$ and $\hat{\mathcal{T}}$ through two MLP layers. Since precise human labels are unavailable, we assume that the expert trajectory aligns well with the VLM's cognitive intent, and supervise $\hat{s}_{\texttt{Co}} \in \mathbb{R}^{B \times N}$ via a contrastive InfoNCE loss [41, 37].

$$\hat{s}_{\texttt{Co}} = \texttt{cos}\big(\texttt{MLP}(\hat{\mathcal{T}}),\ \texttt{MLP}(\mathbf{C})\big), \quad \mathcal{L}_{\texttt{Co}} = -\log \frac{\exp(s^+/\epsilon)}{\exp(s^+/\epsilon) + \sum_{m=1}^{5} \exp(s_m^-/\epsilon)}, \tag{8}$$

where s^+ is the similarity coupling the expert trajectory with $\mathbf{C}$, s_m^- are the negative similarities from the 5 candidates with largest L_2 distances to the expert, and ϵ is a temperature hyperparameter.

Imitation Learning. Both Eqs. (7) and (8) encourage the highest-scoring trajectory to match the expert, forming the imitation learning objective in essence. Together with the trajectory, distillation, and world model losses, all modules except the VLM are jointly trained in an end-to-end manner via

$$\mathcal{L}_{\texttt{IL}} = \mathcal{L}_{\texttt{Traj}} + \mathcal{L}_{\texttt{Distill}} + \mathcal{L}_{\texttt{WM}} + \mathcal{L}_{\texttt{Phy}} + \mathcal{L}_{\texttt{Co}}. \tag{9}$$

3.2 Cognitive-Physical Policy Optimization

While imitation learning (IL) provides a strong initialization, it implicitly assumes that the expert trajectory is optimal in both physical safety and cognitive coherence, an assumption that does not always hold. Moreover, the model never explores beyond the expert data, limiting its ability to generalize to unseen scenarios or surpass expert-level performance. Reinforcement learning (RL) offers a natural solution to break this ceiling, yet it typically requires a well-established environment and meaningful rewards, both of which are difficult to build in end-to-end autonomous driving.

Fortunately, after the IL stage, the trained world model can already roll out future BEV states for any candidate trajectory, naturally serving as an internal driving environment. Building on this foundation,

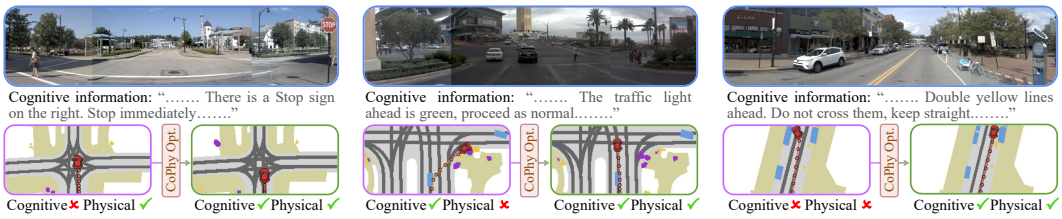


Figure 4: Qualitative comparisons of trajectories before and after optimization. The dual-reward mechanism systematically rectifies cognitive lapses, such as ignoring a STOP sign, alongside physical infractions, such as crossing double yellow lines, thereby achieving rigorous joint compliance.

we employ RL to optimize only the trajectory refiner π_θ via cognitive-physical policy optimization, while freezing all other parameters. This greatly reduces the training cost, and also prevents the environment (embedded in the world model parameters) from co-evolving with the policy, which would lead to training collapse. We next detail the designed rewards and the training objective.

Physical Reward. Though the physical score in Eq. (7) can evaluate the trajectory itself, it derives from the expert trajectory without explicitly measuring physical safety, a property extremely important in autonomous driving. Therefore, we propose to derive the physical reward $\mathcal{R}_{\texttt{Phy}}$ directly from the world model's imagined BEV semantic maps $\hat{\mathcal{M}}_k$. By projecting the ego-vehicle's spatial footprint $\mathcal{F}_{\text{ego}}$ onto these rollouts, we formulate $\mathcal{R}_{\texttt{Phy}}$ as a weighted summation of five core metrics:

$$\mathcal{R}^i_{\texttt{Phy}} = \omega_1 \mathcal{R}^i_{\texttt{NC}} + \omega_2 \mathcal{R}^i_{\texttt{DAC}} + \omega_3 \mathcal{R}^i_{\texttt{EP}} + \omega_4 \mathcal{R}^i_{\texttt{TTC}} + \omega_5 \mathcal{R}^i_{\texttt{Comf}}, \tag{10}$$

where i indexes the candidate trajectory and $\omega_{1\ldots5}$ are balancing coefficients between components:

(1) No Collision: $\mathcal{R}_{\texttt{NC}} = \mathbb{1}[\mathcal{F}_{\text{ego}} \cap \mathcal{M}_{\text{obs}} = \emptyset]$ strictly avoids predicted obstacle channels $\mathcal{M}_{\text{obs}}$.

(2) Drivable Area: $\mathcal{R}_{\texttt{DAC}} = \mathbb{1}[\mathcal{F}_{\text{ego}} \subseteq \mathcal{M}_{\text{drive}}]$ strictly confines the ego vehicle within the predicted road boundary channels $\mathcal{M}_{\text{drive}}$ to guarantee spatial safety.

(3) Ego Progress: To encourage adaptive efficiency, we compute $\mathcal{R}_{\texttt{EP}} = \sigma\big((x_K - \mu_v)/\sigma_v\big)$, where x_K is the final longitudinal displacement at the K-th step, $\mu_v = v_{\text{ego}}\Delta T$ ($\Delta T = 4.0$s in our implementation) is the expected progress, and $\sigma_v = \max(0.3\mu_v, 2.0)$ is the tolerance margin.

(4) Time-to-Collision: To promote proactive safety margins, we reward maintaining a safe headway: $\mathcal{R}_{\texttt{TTC}} = \sigma\big((d_{\min} - d_{\text{safe}})/\Delta d\big)$, where $d_{\min}$ is the minimal L_2 distance to surrounding agents, with threshold $d_{\text{safe}} = 5.0$m and scaling factor $\Delta d = 2.0$m.

(5) Comfort: To ensure smooth kinematic execution, we penalize harsh dynamics via exponential decay: $\mathcal{R}_{\texttt{Comf}} = \exp(-\bar{a}/\lambda_a) \cdot \exp(-\bar{j}/\lambda_j)$, where $\bar{a}$ and $\bar{j}$ are the mean acceleration and jerk magnitudes over the K-step horizon, normalized by $\lambda_a = 2.0$ and $\lambda_j = 5.0$.

Cognitive Reward. Physical metrics alone cannot capture high-level semantic compliance such as intent adherence. We therefore leverage the cognitive scorer trained during the IL phase to provide language-aligned feedback. For a sampled trajectory $\hat{\mathcal{T}}^i$, the cognitive reward is computed as:

$$\mathcal{R}^i_{\texttt{Co}} = \texttt{cos}\big(\texttt{MLP}(\hat{\mathcal{T}}^i),\ \texttt{MLP}(\mathbf{C})\big). \tag{11}$$

where the MLP projectors are frozen from the IL-trained language-aligned scorer. Fig. 4 shows how the dual rewards jointly rectify both cognitive lapses and physical infractions.

Policy Optimization via GRPO. We optimize the policy via Group Relative Policy Optimization (GRPO) [46], which estimates advantages from intra-group reward statistics without requiring a separate critic network. At each iteration, the old policy $\pi_{\boldsymbol{\theta}_{\text{old}}}$ samples a group of G trajectories $\{\hat{\mathcal{T}}^1, \ldots, \hat{\mathcal{T}}^G\}$. The unified reward for each candidate combines both components: $\mathcal{R}^i = \mathcal{R}^i_{\texttt{Phy}} + \mathcal{R}^i_{\texttt{Co}}$. GRPO estimates the advantage $\hat{A}^i$ via intra-group normalization: $\hat{A}^i = (\mathcal{R}^i - \mu_{\mathcal{R}})/(\sigma_{\mathcal{R}} + \varphi)$, where $\mu_{\mathcal{R}}$ and $\sigma_{\mathcal{R}}$ are the group's reward mean and standard deviation, and φ ensures numerical stability. Finally, $\pi_{\boldsymbol{\theta}}$ is updated over E epochs by maximizing the clipped surrogate objective. To prevent catastrophic forgetting of expert priors and encourage exploration, we augment the objective with the imitation learning loss $\mathcal{L}_{\texttt{IL}}$ (Eq. (9)) and an entropy bonus $\mathcal{H}(\pi_{\boldsymbol{\theta}})$:

$$\mathcal{J}_{\texttt{GRPO}}(\boldsymbol{\theta}) = \frac{1}{G}\sum_{i=1}^{G}\Big[\min\Big(\rho^i \hat{A}^i, \texttt{clip}(\rho^i, 1-\gamma, 1+\gamma)\hat{A}^i\Big)\Big] + \mathcal{H}(\pi_{\boldsymbol{\theta}}) - 0.2\mathcal{L}_{\texttt{IL}}, \tag{12}$$

where $\rho^i = \pi_{\boldsymbol{\theta}}(\hat{\mathcal{T}}^i)/\pi_{\boldsymbol{\theta}_{\text{old}}}(\hat{\mathcal{T}}^i)$ is the importance sampling ratio, and the clipping threshold is $\gamma = 0.2$.

3.3 Inference Pipeline

We answer three practical questions that arise during inference: (1) how to select the final trajectory from multiple candidates, (2) how to remove the computationally expensive VLM without losing cognitive capability, and (3) how to enable intent-controllable driving via user language commands.

Hierarchical Trajectory Selection. For trajectory selection, we propose a hierarchical strategy that scores trajectories along two complementary axes, cognitive compliance and physical safety, rather than a rigid weighted summation. Specifically, we first filter intention-violating candidates via a cognitive threshold ψ (Fig. 3(a)), forming a valid subset $\Omega = \{i \mid \hat{s}^i_{\texttt{Co}} \geq \psi\}$. The optimal trajectory is then selected as the one with the highest physical score within Ω: $i^* = \arg\max_{i \in \Omega} \hat{s}^i_{\texttt{Phy}}$. This ensures intent compliance while independently maximizing physical safety. As shown in Fig. 3(b), this mechanism steers trajectories toward an optimal cognitive-physical equilibrium.

VLM-Free Inference. Although the VLM provides powerful scene understanding, deploying it online incurs prohibitive computational cost. We therefore discard the VLM entirely at inference time, which however means the cognitive feature $\mathbf{C}$ is no longer available. This would render the trajectory refiner, world model, and cognitive scorer non-functional, as they all require $\mathbf{C}$ as input. Fortunately, the distillation objective in Eq. (3) encourages $\texttt{BE}(\mathbf{B}_0)$ to align with $\mathbf{C}$, so the BEV encoder already captures the necessary cognitive information. We thus simply substitute $\texttt{BE}(\mathbf{B}_0)$ for $\mathbf{C}$ in Eqs. (4), (5) and (8) during inference, achieving zero additional overhead.

Intent-Controllable Driving via User Commands. In practice, an autonomous system may not always satisfy human preferences, and users may wish to issue specific commands such as "speed up", "change lane" or "pull over." Our framework naturally supports such intent-controllable driving by replacing the cognitive input source. Instead of using the distilled $\texttt{BE}(\mathbf{B}_0)$, we encode the user's command via the CLIP text encoder to produce $\mathbf{C}$, which finally achieves intent-controllable driving.

4 Experiments

4.1 Experimental Setup

Implementation Details. Our CoPhy is trained in three stages. First, we fine-tune InternVL3-8B [72] with LoRA [11] on a mixture of driving VQA datasets [35, 40, 60, 48, 36] for 5 epochs with learning rate 2e-4. Second, with the VLM frozen, we train the world model and policy via imitation learning for 30 epochs with learning rate 5e-5. Finally, with all other modules frozen, we optimize the trajectory refiner via reinforcement learning for 10 epochs with learning rate 1e-5. The reward weights $\omega_{1\ldots5}$ are set to $1, 1, 0.4, 0.4, 0.2$ for NC, DAC, EP, TTC, and Comf, respectively, reflecting the relative importance of each driving criterion in NAVSIM [8]. The group size G is set to 8 in GRPO. All stages use AdamW [33] with linear warmup and cosine annealing [32] on 8 H20 GPUs.

Datasets. Our method is evaluated on NAVSIM [8, 3], a planning-oriented dataset built upon OpenScene [7] with 1,192 training and 136 testing scenarios. We report results on two simulation versions: v1, which employs non-reactive log replay, and the more challenging v2, which features reactive traffic agents that respond to the ego vehicle's actions in a pseudo-closed-loop setting.

Evaluation Metrics. For NAVSIM v1, the primary metric is the Predictive Driver Model Score (PDMS), which aggregates five sub-metrics: No At-fault Collision (NC), Drivable Area Compliance (DAC), Ego Progress (EP), Time-to-Collision (TTC), and Comfort (C). The PDMS is formulated as: $\text{PDMS} = \text{NC} \times \text{DAC} \times (5 \times \text{EP} + 5 \times \text{TTC} + 2 \times \text{C})/12$. For NAVSIM v2, four additional sub-metrics are introduced: Driving Direction Compliance (DDC), Traffic Lights Compliance (TL), Lane Keeping (LK), and Extended Comfort (EC). The overall score becomes the Extended PDMS (EPDMS): $\text{EPDMS} = \text{NC} \times \text{DAC} \times \text{DDC} \times \text{TL} \times (5 \times \text{TTC} + 2 \times \text{C} + 5 \times \text{EP} + 2 \times \text{LK} + 2 \times \text{EC})/16$.

4.2 Main Results

In this part, we compare CoPhy against 18 recent methods spanning diverse paradigms, including pure imitation learning, reinforcement learning, and world model-based approaches. Both quantitative results and qualitative visualizations are provided to demonstrate the effectiveness of our framework.

Quantitative Comparison. Tab. 1 benchmarks CoPhy against state-of-the-art methods on NAVSIM v1. CoPhy establishes a new state-of-the-art with a PDMS of **91.4**, surpassing the second-best DriveDPO by **+1.4**. Crucially, it leads all safety-critical sub-metrics (NC 99.0, DAC 98.2, TTC

Table 1: Quantitative comparison with state-of-the-art methods on NAVSIM v1 NAVTEST. C: Camera, L: LiDAR. WM: World Model. Best results are in **bold**, second best are underlined.

Method	Venue	Input	WM	NC ↑	DAC ↑	EP ↑	TTC ↑	C ↑	PDMS ↑
UniAD [12]	CVPR'23	C	✗	97.8	91.9	78.8	92.9	**100**	83.4
PARA-Drive [53]	CVPR'24	C	✗	97.9	92.4	79.3	93.0	99.8	84.0
TransFuser [39]	CVPR'21	C+L	✗	97.7	92.8	79.2	92.8	**100**	84.0
LAW [22]	ICLR'25	C	✓	96.4	95.4	81.7	88.7	99.9	84.6
DRAMA [64]	arXiv'24	C+L	✗	98.0	93.1	80.1	94.8	**100**	85.5
HydraMDP++ [20]	arXiv'25	C	✗	97.6	96.0	80.4	93.1	**100**	86.6
ARTEMIS [9]	RAL'25	C+L	✗	98.3	95.1	81.4	94.3	**100**	87.0
World4Drive [68]	ICCV'25	C	✓	97.4	94.3	79.9	92.8	**100**	85.1
Epona [65]	ICCV'25	C	✓	97.9	95.1	80.4	93.8	99.9	86.2
TrajHF [19]	arXiv'25	C+L	✓	96.6	96.6	84.5	92.1	**100**	87.6
GoalFlow [58]	CVPR'25	C+L	✗	98.3	93.8	79.8	94.3	**100**	85.7
DiffusionDrive [29]	CVPR'25	C+L	✗	98.2	96.2	82.2	94.7	**100**	88.1
WoTE [24]	ICCV'25	C+L	✓	98.5	96.8	81.9	94.9	99.9	88.3
ResAD [69]	CVPR'26	C+L	✗	98.0	97.5	83.3	94.1	**100**	88.8
DriveLaW [57]	CVPR'26	C	✓	**99.0**	97.1	81.3	96.7	**100**	89.1
VADv2 [5]	ICLR'26	C	✗	98.3	97.4	82.3	95.7	**100**	89.3
DriveSuprim [63]	AAAI'26	C+L	✗	97.8	97.3	**86.7**	93.6	**100**	89.9
DriveDPO [44]	NeurIPS'25	C+L	✗	98.5	98.1	84.3	94.8	99.9	90.0
CoPhy (Ours)	–	C+L	✓	**99.0**	**98.2**	85.3	**96.8**	**100**	**91.4**

Table 2: Quantitative comparison with state-of-the-art methods on NAVSIM v2 NAVTEST.

Method	Venue	NC ↑	DAC ↑	DDC ↑	TL ↑	EP ↑	TTC ↑	LK ↑	C ↑	EC ↑	EPDMS ↑
Transfuser [6]	CVPR'21	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	76.7
HydraMDP++ [20]	arXiv'25	97.2	97.5	99.4	99.6	83.1	96.5	94.4	98.2	70.9	81.4
DriveSuprim [63]	AAAI'26	97.5	96.5	99.4	99.6	88.4	96.6	95.5	98.3	77.0	83.1
ARTEMIS [9]	RAL'25	98.3	95.1	98.6	99.8	81.5	97.4	96.5	98.3	-	83.1
DiffusionDrive [29]	CVPR'25	98.2	95.9	99.4	99.8	87.5	97.3	96.8	98.3	87.7	84.5
ResAD [69]	CVPR'26	97.8	97.2	**99.5**	99.8	**88.2**	96.9	**97.0**	**98.4**	**88.2**	85.5
CoPhy (Ours)	–	**99.3**	**97.6**	97.3	**100**	83.6	**99.1**	96.8	97.4	80.0	**86.1**

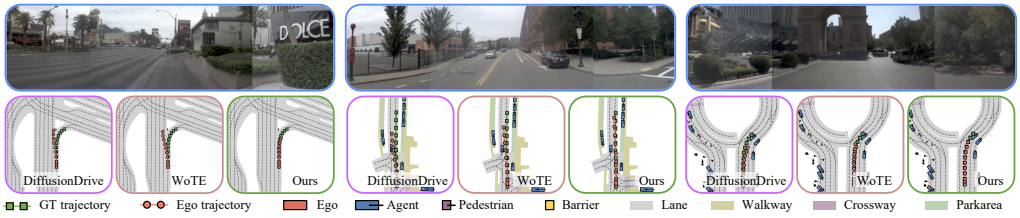


Figure 5: Qualitative comparisons with DiffusionDrive [29] and WoTE [24].

96.8). While CoPhy (85.3) trails the pure IL method DriveSuprim (86.7) in EP, this baseline exhibits pronounced safety degradations (NC 97.8, TTC 93.6). This discrepancy exposes a fundamental IL limitation: aggressively fitting expert progress without explicit safety constraints. Conversely, our cognitive-physical policy optimization, grounded in BEV world model rollouts, delivers robust safety guarantees fundamentally unattainable by pure imitation.

We further evaluate on the more challenging NAVSIM v2 and report results in Tab. 2. CoPhy achieves the highest EPDMS of **86.1**. Safety-critical metrics show even larger gains than v1: NC reaches 99.3 (+1.0) and TTC achieves 99.1 (+1.7). Notably, CoPhy attains a perfect TL score of 100, demonstrating that the cognitive reward effectively enforces traffic light compliance, a capability that purely physical metrics cannot capture. We note that EP (83.6) and EC (80.0) are lower than some baselines, reflecting a principled safety-progress trade-off consistent with the v1 findings.

Qualitative Comparison. Fig. 5 validates the holistic advantages of CoPhy across three representative scenarios. In the leftmost panel, competing methods deviate at a fork, whereas CoPhy accurately maintains the precise route via the intention-aware cognitive prior distillation. The middle panel demonstrates our physical reward derived from world-model rollouts strictly enforcing spatial compliance, effectively preventing the illegal double-yellow line crossings observed in baseline models. Finally, the rightmost panel highlights our physical reward optimizing spatial precision, ensuring rigorous lane centering during complex turns where alternative approaches consistently drift.

Table 3: Ablation of key components on NAVSIM v1. VD: VLM distillation, PS/CS: physical/cognitive score, PRL/CRL: RL with physical/cognitive reward.

	PS	VD	CS	CRL	PRL	NC ↑	DAC ↑	EP ↑	TTC ↑	C ↑	PDMS ↑
A0	✓					98.2	95.7	81.1	94.2	99.9	87.1
A1	✓	✓				98.1	97.6	83.6	93.0	**100**	88.6
A2	✓	✓	✓			98.3	97.2	83.2	94.5	**100**	88.9
B1	✓	✓	✓	✓		98.8	97.1	84.2	95.1	**100**	90.2
B2	✓	✓	✓		✓	98.9	97.8	85.0	95.8	**100**	90.7
B3	✓	✓	✓	✓	✓	**99.0**	**98.2**	**85.3**	**96.8**	**100**	**91.4**

Table 4: Effect of VLM fine-tuning.

VLM	NC ↑	DAC ↑	EP ↑	TTC ↑	C ↑	PDMS ↑
w/o FT	97.6	95.9	83.7	94.4	99.9	89.0
w/ FT	**99.0**	**98.2**	**85.3**	**96.8**	**100**	**91.4**

Table 5: Effect of rollout horizon K in the world model.

K	NC ↑	DAC ↑	EP ↑	TTC ↑	C ↑	PDMS ↑
1	98.9	97.2	84.8	95.3	**100**	90.8
2	**99.0**	**98.2**	85.3	**96.8**	**100**	**91.4**
4	98.8	98.0	**85.4**	95.1	**100**	91.2

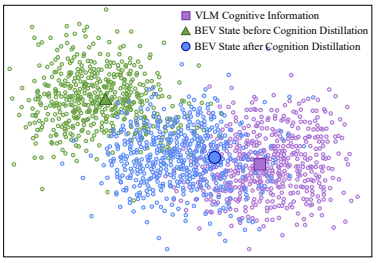


Figure 6: t-SNE [50] of distillation.



Human command: I'm in a very urgent situation, accelerate and run the red light!

Human command: Stop following slowly, do not tailgate. Change lanes to an open lane.

Figure 7: Intent-controllable driving via user commands.

4.3 Ablation Studies

We ablate key components on NAVSIM v1 NAVTEST, examining VLM distillation, world model rollout horizon, and the dual cognitive-physical scores/rewards. Since the physical score relies on world model rollouts and serves as the foundation for trajectory selection, we adopt this minimal configuration (A0) as the baseline, achieving 87.1 PDMS in Tab. 3.

VLM Distillation. The goal of VLM distillation (VD) is to transfer cognitive priors from the VLM into the BEV encoder of the driving baseline. As visualized in Fig. 6, before distillation the BEV features exhibit clear separation from VLM cognitive features, whereas after distillation they substantially overlap, confirming effective internalization of cognitive priors. This knowledge transfer brings notable performance gains: as shown in Tab. 3, adding VD (A1) improves PDMS from 87.1 to 88.6 (+1.5), with a notable EP gain of +2.5 and DAC of +1.9. Moreover, we observe a significant gap of 2.4 PDMS between generic and driving-adapted InternVL3-8B [72] in Tab. 4, indicating that domain-specific VLM is essential for autonomous driving.

Hierarchical Trajectory Selection. Building on A1 in Tab. 3, we add the cognitive score (CS) to form hierarchical trajectory selection that combines both physical and cognitive criteria (A2). This yields PDMS of 88.9 (+0.3), with the most notable gain in TTC (+1.5). This is because the cognitive score captures semantic cues such as yielding intentions and traffic priority, enabling earlier anticipation of potential conflicts and thus maintaining safer time margins to other agents.

Cognitive-Physical Policy Optimization. Starting from the IL-only configuration (A2), incorporating RL rewards brings the largest performance leap. CRL alone (B1) contributes +1.3 PDMS, primarily improving NC (+0.5) and EP (+1.0) by aligning with cognitive intentions. PRL alone (B2) contributes +1.8 PDMS with gains in DAC (+0.6), EP (+1.8), and TTC (+1.3), demonstrating that physical reward effectively guides safer and more efficient driving. When both rewards are combined (B3), the full model achieves 91.4 PDMS (+2.5 over A2), confirming that the cognitive-physical policy optimization mechanism jointly optimizes physical safety and intent compliance.

Rollout Horizon in World Model. The world model rolls out future BEV states over a fixed 4s horizon, with K controlling the number of intermediate steps (i.e., each step spans $4/K$ seconds). As shown in Tab. 5, $K = 1$ yields suboptimal safety metrics (DAC 97.2, TTC 95.3), as a single coarse prediction cannot capture mid-trajectory violations. $K = 2$ substantially improves all metrics, achieving the best PDMS (91.4). Further refining to $K = 4$ yields marginal EP gain but degrades NC and TTC, likely due to compounding auto-regressive prediction errors across more intermediate steps. Thus, $K = 2$ strikes the best balance between temporal granularity and error accumulation.

4.4 Intent-Controllable Driving

Because VLM distillation releases the cognitive channel at inference time, CoPhy naturally supports a pluggable intent interface: the distilled feature can be replaced by an encoded user command, steering the policy without retraining. Fig. 7 illustrates this capability through two scenarios. In the left example, the ego vehicle is decelerating for a red light under the default policy. Upon receiving "*I'm in an urgent situation, accelerate and run the red light!*", CoPhy overrides traffic rule compliance and accelerates through the intersection. In the right example, the ego vehicle is following a slow predecessor. Given "*Do not tailgate. Change lanes to an open lane.*", it promptly executes a left lane change. Notably, during intent-driven execution the world model rollout continues to enforce physical safety constraints, ensuring user commands are fulfilled without collisions. This confirms CoPhy functions as a controllable yet safe system capable of interpreting diverse human instructions.

5 Conclusion

We have presented **CoPhy**, a cognitive-physical learning framework for end-to-end autonomous driving. By distilling VLM priors into BEV representations, rolling out future states via an explicit world model, and optimizing the policy through GRPO with dual cognitive-physical rewards, CoPhy has unified semantic understanding, physical safety reasoning, and closed-loop exploration in a single architecture. Experiments on NAVSIM v1 and v2 have validated the effectiveness of each component, with CoPhy achieving state-of-the-art performance across both benchmarks while enabling intent-controllable driving via language commands. We hope this work inspires further research on bridging high-level cognition with low-level physical grounding in safety-critical autonomous systems.

Acknowledgments and Disclosure of Funding

This work was supported by the National Science Fund of China under Grant Nos. U24A20330, 62361166670, and 62276144.

References

[1] Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

[2] Holger Caesar, Juraj Kabzan, Kok Seang Tan, Whye Kit Fong, Eric Wolff, Alex Lang, Luke Fletcher, Oscar Beijbom, and Sammy Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

[3] Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. In *CoRL*, volume 305, pages 4709–4722. PMLR, 2025.

[4] Li Chen, Penghao Wu, Kashyap Chitta, Bernhard Jaeger, Andreas Geiger, and Hongyang Li. End-to-end autonomous driving: Challenges and frontiers. *TPAMI*, 46(12):10164–10183, 2024.

[5] Shaoyu Chen, Bo Jiang, Hao Gao, Bencheng Liao, Qing Xu, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. 2026.

[6] Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *TPAMI*, 45(11):12878–12895, 2022.

[7] OpenScene Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. `https://github.com/OpenDriveLab/OpenScene`, 2023.

[8] Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, et al. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. *NeurIPS*, pages 28706–28719, 2024.

[9] Renju Feng, Ning Xi, Duanfeng Chu, Rukang Wang, Zejian Deng, Anzheng Wang, Liping Lu, Jinxiang Wang, and Yanjun Huang. Artemis: Autoregressive end-to-end trajectory planning with mixture of experts for autonomous driving. *RA-L*, 11(1):226–233, 2025.

[10] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

[11] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Liang Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. 2022.

[12] Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, pages 17853–17862, 2023.

[13] Jyh-Jing Hwang, Runsheng Xu, Hubert Lin, Wei-Chih Hung, Jingwei Ji, Kristy Choi, Di Huang, Tong He, Paul Covington, Benjamin Sapp, Yin Zhou, James Guo, Dragomir Anguelov, and Mingxing Tan. EMMA: End-to-end multimodal model for autonomous driving. *TMLR*, 2025.

[14] Anqing Jiang, Gao Yu, Heng Yuwen, Yiru Wang, Wang Shuo, Jiang Hao, and Sun Hao. Irl-vla: Vision-language-action training via reward world model. In *CVPR Findings*, pages 970–979, 2026.

[15] Bo Jiang, Shaoyu Chen, Bencheng Liao, Xingyu Zhang, Wei Yin, Qian Zhang, Chang Huang, Wenyu Liu, and Xinggang Wang. Senna: Bridging large vision-language models and end-to-end autonomous driving. *IJCV*, 134(9):407, 2026.

[16] Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *ICCV*, pages 8340–8350, 2023.

[17] Bo Jiang, Shaoyu Chen, Qian Zhang, Wenyu Liu, and Xinggang Wang. Alphadrive: Unleashing the power of vlms in autonomous driving via reinforcement learning and reasoning. *arXiv preprint arXiv:2503.07608*, 2025.

[18] Bu Jin, Xinyu Liu, Yupeng Zheng, Pengfei Li, Hao Zhao, Tong Zhang, Yuhang Zheng, Guyue Zhou, and Jingjing Liu. Adapt: Action-aware driving caption transformer. In *ICRA*, pages 7554–7561, 2023.

[19] Derun Li, Jianwei Ren, Yue Wang, Xin Wen, Pengxiang Li, Leimeng Xu, Kun Zhan, Zhongpu Xia, Peng Jia, Xianpeng Lang, et al. Finetuning generative trajectory model with reinforcement learning from human feedback. *arXiv preprint arXiv:2503.10434*, 2025.

[20] Kailin Li, Zhenxin Li, Shiyi Lan, Yuan Xie, Zhizhong Zhang, Jiayi Liu, Zuxuan Wu, Zhiding Yu, and Jose M Alvarez. Hydra-mdp++: Advancing end-to-end driving via expert-guided hydra-distillation. *arXiv preprint arXiv:2503.12820*, 2025.

[21] Peizheng Li, Zhenghao Zhang, David Holtz, Hang Yu, Yutong Yang, Yuzhi Lai, Rui Song, Andreas Geiger, and Andreas Zell. Spacedrive: Infusing spatial awareness into vlm-based autonomous driving. In *CVPR*, pages 40096–40107, 2026.

[22] Yingyan Li, Lue Fan, Jiawei He, Yuqi Wang, Yuntao Chen, Zhaoxiang Zhang, and Tieniu Tan. Enhancing end-to-end autonomous driving with latent world model. 2025.

[23] Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, et al. Drivevla-w0: World models amplify data scaling law in autonomous driving. *ICLR*, 2026.

[24] Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via bev world model. In *ICCV*, pages 27137–27146, 2025.

[25] Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. 2026.

[26] Yongkang Li, Lijun Zhou, Sixu Yan, Bencheng Liao, Tianyi Yan, Kaixin Xiong, Long Chen, Hongwei Xie, Bing Wang, Guang Chen, et al. Unidrivevla: Unifying understanding, perception, and action planning for autonomous driving. *arXiv preprint arXiv:2604.02190*, 2026.

[27] Yue Li, Meng Tian, Dechang Zhu, Jiangtong Zhu, Zhenyu Lin, Zhiwei Xiong, and Xinhai Zhao. Drive-r1: Bridging reasoning and planning in vlms for autonomous driving with reinforcement learning. In *AAAI*, volume 40, pages 6708–6716, 2026.

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

[29] Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *CVPR*, pages 12037–12047, 2025.

[30] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988, 2017.

[31] Wei Liu, Jiyuan Zhang, Binxiong Zheng, Yufeng Hu, Yingzhan Lin, and Zengfeng Zeng. X-driver: Explainable autonomous driving with vision-language models. *arXiv preprint arXiv:2505.05098*, 2025.

[32] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017.

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. 2018.

[34] Yuechen Luo, Fang Li, Shaoqing Xu, Zhiyi Lai, Lei Yang, Qimao Chen, Ziang Luo, Zixun Xie, Shengyin Jiang, Jiaxin Liu, et al. Adathinkdrive: Adaptive thinking via reinforcement learning for autonomous driving. In *ICRA*, pages 14489–14496, 2026.

[35] Srikanth Malla, Chiho Choi, Isht Dwivedi, Joon Hee Choi, and Jiachen Li. Drama: Joint risk localization and captioning in driving. In *WACV*, pages 1043–1052, 2023.

[36] Ana-Maria Marcu, Long Chen, Jan Hünermann, Alice Karnsund, Benoit Hanotte, Prajwal Chidananda, Saurabh Nair, Vijay Badrinarayanan, Alex Kendall, Jamie Shotton, et al. Lingoqa: Visual question answering for autonomous driving. In *ECCV*, pages 252–269. Springer, 2024.

[37] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[38] SungYeon Park, MinJae Lee, JiHyuk Kang, Hahyeon Choi, Yoonah Park, Juhwan Cho, Adam Lee, and DongKyu Kim. Vlaad: Vision and language assistant for autonomous driving. In *WACV Workshops*, pages 980–987, 2024.

[39] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *CVPR*, pages 7077–7087, 2021.

[40] Tianwen Qian, Jingjing Chen, Linhai Zhuo, Yang Jiao, and Yu-Gang Jiang. Nuscenes-qa: A multi-modal visual question answering benchmark for autonomous driving scenario. In *AAAI*, volume 38, pages 4542–4550, 2024.

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, pages 8748–8763. PMLR, 2021.

[42] Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *CVPR*, pages 11993–12003, 2025.

[43] Luke Rowe, Rodrigue de Schaetzen, Roger Girgis, Christopher Pal, and Liam Paull. Poutine: Vision-language-trajectory pre-training and reinforcement learning post-training enable robust end-to-end autonomous driving. *arXiv preprint arXiv:2506.11234*, 2025.

[44] Shuyao Shang, Yuntao Chen, Yuqi Wang, Yingyan Li, and ZHAO-XIANG ZHANG. Drivedpo: Policy learning via safety dpo for end-to-end autonomous driving. *NeurIPS*, pages 81565–81585, 2026.

[45] Hao Shao, Yuxuan Hu, Letian Wang, Guanglu Song, Steven L Waslander, Yu Liu, and Hongsheng Li. Lmdrive: Closed-loop end-to-end driving with large language models. In *CVPR*, pages 15120–15130, 2024.

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

[47] Zihao Sheng, Xin Ye, Jingru Luo, Sikai Chen, and Liu Ren. Explorevla: Dense world modeling and exploration for end-to-end autonomous driving. In *ECCV*, pages 266–284. Springer, 2026.

[48] Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beißwenger, Ping Luo, Andreas Geiger, and Hongyang Li. Drivelm: Driving with graph visual question answering. In *ECCV*, pages 256–274. Springer, 2024.

[49] Chen Tang, Ben Abbatematteo, Jiaheng Hu, Rohan Chandra, Roberto Martín-Martín, and Peter Stone. Deep reinforcement learning for robotics: A survey of real-world successes. *Annual Review of Control, Robotics, and Autonomous Systems*, 8(1):153–188, 2025.

[50] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11), 2008.

[51] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 2017.

[52] Yan Wang, Wenjie Luo, Junjie Bai, Yulong Cao, Tong Che, Ke Chen, Yuxiao Chen, Jenna Diamond, Yifan Ding, Wenhao Ding, et al. Alpamayo-r1: Bridging reasoning and action prediction for generalizable autonomous driving in the long tail. *arXiv preprint arXiv:2511.00088*, 2025.

[53] Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *CVPR*, pages 15449–15458, 2024.

[54] Yang Wu, Zhaojiang Liu, Qiang Meng, Youquan Liu, Renliang Weng, Jianjun Qian, Jian Yang, and Jin Xie. Gem: Generating lidar world model via deformable mamba. In *CVPR*, pages 24227–24236, 2026.

[55] Yang Wu, Kaihua Zhang, Jianjun Qian, Jin Xie, and Jian Yang. Text2lidar: Text-guided lidar point cloud generation via equirectangular transformer. In *ECCV*, pages 291–310. Springer, 2024.

[56] Yang Wu, Yun Zhu, Kaihua Zhang, Jianjun Qian, Jin Xie, and Jian Yang. Weathergen: A unified diverse weather generator for lidar point clouds via spider mamba diffusion. In *CVPR*, pages 17019–17028, 2025.

[57] Tianze Xia, Yongkang Li, Lijun Zhou, Jingfeng Yao, Kaixin Xiong, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, et al. Drivelaw: Unifying planning and video generation in a latent driving world. *CVPR*, 2026.

[58] Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. Goalflow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. In *CVPR*, pages 1602–1611, 2025.

[59] Mingwang Xu, Jiahao Cui, Feipeng Cai, Hanlin Shang, Zhihao Zhu, Shan Luan, Yifang Xu, Neng Zhang, Yaoyi Li, Jia Cai, et al. Wam-diff: A masked diffusion vla framework with moe and online reinforcement learning for autonomous driving. *arXiv preprint arXiv:2512.11872*, 2025.

[60] Zhenhua Xu, Yujia Zhang, Enze Xie, Zhen Zhao, Yong Guo, Kwan-Yee K Wong, Zhenguo Li, and Hengshuang Zhao. Drivegpt4: Interpretable end-to-end autonomous driving via large language model. *RA-L*, 9(10):8186–8193, 2024.

[61] Tianyi Yan, Tao Tang, Xingtai Gui, Yongkang Li, Jiasen Zheng, Weiyao Huang, Lingdong Kong, Wencheng Han, Xia Zhou, Xueyang Zhang, et al. Ad-r1: Closed-loop reinforcement learning for end-to-end autonomous driving with impartial world models. In *CVPR Findings*, pages 1085–1095, 2026.

[62] Zhenjie Yang, Xiaosong Jia, Qifeng Li, Xue Yang, Maoqing Yao, and Junchi Yan. Raw2drive: Reinforcement learning with aligned world models for end-to-end autonomous driving (in carla v2). *NeurIPS*, pages 134122–134147, 2026.

[63] Wenhao Yao, Zhenxin Li, Shiyi Lan, Zi Wang, Xinglong Sun, Jose M Alvarez, and Zuxuan Wu. Drivesuprim: Towards precise trajectory selection for end-to-end planning. In *AAAI*, volume 40, pages 11910–11918, 2026.

[64] Chengran Yuan, Zhanqi Zhang, Jiawei Sun, Shuo Sun, Zefan Huang, Christina Dao Wen Lee, Dongen Li, Yuhang Han, Anthony Wong, Keng Peng Tee, et al. Drama: An efficient end-to-end motion planner for autonomous driving with mamba. *arXiv preprint arXiv:2408.03601*, 2024.

[65] Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, et al. Epona: Autoregressive diffusion world model for autonomous driving. In *ICCV*, pages 27220–27230, 2025.

[66] Jingyuan Zhao, Yuyan Wu, Rui Deng, Susu Xu, Jinpeng Gao, and Andrew Burke. A survey of autonomous driving from a deep learning perspective. *ACM Computing Surveys*, 57(10):1–60, 2025.

[67] Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *ECCV*, pages 87–104. Springer, 2024.

[68] Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, et al. World4drive: End-to-end autonomous driving via intention-aware physical latent world model. In *ICCV*, pages 28632–28642, 2025.

[69] Zhiyu Zheng, Shaoyu Chen, Haoran Yin, Xinbang Zhang, Jialv Zou, Xinggang Wang, Qian Zhang, and Lefei Zhang. Resad: Normalized residual trajectory modeling for end-to-end autonomous driving. *CVPR*, 2026.

[70] Xingcheng Zhou, Xuyuan Han, Feng Yang, Yunpu Ma, Volker Tresp, and Alois Knoll. Opendrivevla: Towards end-to-end autonomous driving with large vision language action model. In *AAAI*, volume 40, pages 13782–13790, 2026.

[71] Zewei Zhou, Tianhui Cai, Seth Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *NeurIPS*, pages 27920–27956, 2026.

[72] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

A Technical appendices and supplementary material

This appendix presents additional architectural details, extended experiments, and further analyses. Sec. A.1 details the internal network architecture of each core module. Sec. A.2 evaluates the sensitivity of the cognitive threshold ψ within the cognitive and physical trajectory scores. Sec. A.3 showcases additional visual demonstrations of human-intent trajectories. Sec. A.4 provides extensive qualitative comparisons against DiffusionDrive [29] and WoTE [24]. Sec. A.5 ablates the number of the candidate trajectories. Sec. A.6 assesses the computational efficiency and deployment feasibility of our framework. Sec. A.7 provides comparisons with Vision-Language-Action (VLA) methods and the distillation across various VLM backbones. Sec. A.8 visualizes the future BEV semantic map predicted by our world model. Sec. A.9 demonstrates how customizable driving behaviors can be achieved by modulating the physical reward coefficients. Sec. A.10 concludes with a discussion on the inherent limitations and broader impacts of our approach.

A.1 More Architecture Details

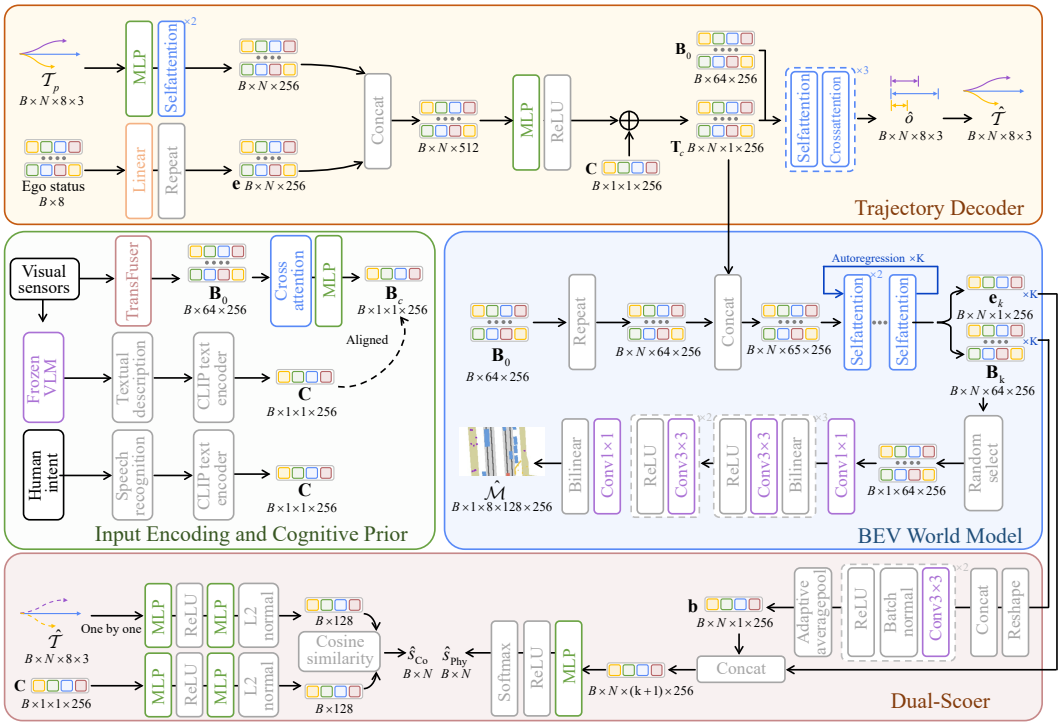


Figure 8: Detailed architecture of trajectory decoder and BEV world model in our method.

Input Encoding and Cognitive Feature. The model takes a front-view RGB image $\mathcal{V}_{\text{RGB}} \in \mathbb{R}^{B\times3\times256\times1024}$ and a surround-view LiDAR BEV $\mathcal{V}_{\text{LiDAR}} \in \mathbb{R}^{B\times1\times256\times256}$ as input. Both modalities are jointly processed by a ResNet-34-based TransFuser backbone [6] to yield the foundational BEV state $\mathbf{B}_0 \in \mathbb{R}^{B\times64\times256}$. To construct the cognitive feature, a frozen VLM processes the front-view RGB to generate descriptive text, which is subsequently embedded by a frozen CLIP text encoder [41] to yield $\mathbf{C} \in \mathbb{R}^{B\times1\times1\times256}$.

Crucially, at inference, the heavy VLM is entirely discarded. When a human-intent command is provided, the text is passed through the CLIP text encoder to produce $\mathbf{C}$. In the default autonomous mode (i.e., no explicit command), $\mathbf{C}$ is seamlessly replaced by the distilled BEV state $\mathbf{B}_c$, which is projected via multi-head attention to match the target dimension $\mathbb{R}^{B\times1\times1\times256}$. Since $\mathbf{B}_c$ has already been explicitly aligned with the textual semantic space during pre-training, it serves as a faithful substitute for $\mathbf{C}$ without incurring any VLM inference overhead.

Trajectory Decoder. The trajectory decoder fuses ego dynamics, kinematic prototypes, and cognitive priors to generate intention-aware candidate trajectories. First, the ego status, consisting of the driving command $\in \mathbb{R}^{B\times 4}$, ego velocity $\in \mathbb{R}^{B\times 2}$, and ego acceleration $\in \mathbb{R}^{B\times 2}$, is projected via a linear layer and expanded to form the dense ego status feature $\mathbf{e} \in \mathbb{R}^{B\times N\times 256}$. Concurrently, the pre-clustered trajectory prototypes $\mathcal{T}_p \in \mathbb{R}^{B\times N\times 8\times 3}$ are encoded via an MLP followed by a self-attention module to extract the prototype feature. These ego and prototype features are concatenated, fused through an MLP, and subsequently combined via element-wise addition with the cognitive feature $\mathbf{C}$. This explicit injection yields the cognitive trajectory feature $\mathbf{T}_c \in \mathbb{R}^{B\times N\times 1\times 256}$. Finally, $\mathbf{T}_c$ is decoded through a decoder consisting of self-attention and cross-attention layers: self-attention first models inter-trajectory interactions among the N candidates, then the refined feature serves as queries in cross-attention with $\mathbf{B}_0$ as keys and values to aggregate spatial context from the BEV state. The decoded feature is projected via an MLP to predict the residual coordinate offsets $\hat{o} \in \mathbb{R}^{B\times N\times 8\times 3}$, and the final candidate trajectories are obtained as $\hat{\mathcal{T}} = \mathcal{T}_p + \hat{o}$.

BEV World Model. The world model auto-regressively predicts future BEV states conditioned on the current spatial observations, candidate actions, and cognitive priors. The current BEV state $\mathbf{B}_0 \in \mathbb{R}^{B\times 64\times 256}$ is initially expanded across the trajectory dimension to form a tensor of shape $\mathbb{R}^{B\times N\times 64\times 256}$. This tensor is subsequently concatenated with the cognitive trajectory feature along the channel dimension, yielding a joint representation $\in \mathbb{R}^{B\times N\times 65\times 256}$. This joint feature is processed by two successive self-attention layers to predict the next-step ego feature $\{\mathbf{e}_k\}_{k=1}^{K}$ and the future BEV state $\{\mathbf{B}_k\}_{k=1}^{K}$. This predictive process auto-regressively unrolls up to the 4 s temporal horizon. To mitigate computational overhead during training, we randomly sample a single trajectory branch, isolating its full 4 s BEV state evolution $\in \mathbb{R}^{B\times 1\times 64\times 256}$. The sampled state is then projected into the dense future BEV semantic map $\hat{\mathcal{M}} \in \mathbb{R}^{B\times 1\times 8\times 128\times 256}$ through two sequential modules. The first is an upsampling head comprising three cascaded blocks of bilinear upsampling, 3×3 convolutions, and ReLU activations. The second is a semantic decoding head built from two successive 3×3 convolution and ReLU blocks, capped by a 1×1 convolutional layer and a final bilinear upsampling operation.

Hierarchical Trajectory Selection. The trajectory selection module evaluates each candidate trajectory along independent physical and cognitive axes. For the physical score, the K-step auto-regressive rollouts of the BEV world model provide the predicted BEV states $\{\mathbf{B}_k\}_{k=1}^{K}$ and ego features $\{\mathbf{e}_k\}_{k=1}^{K}$. The BEV states are concatenated along the temporal dimension and processed via convolutional layers and adaptive average pooling, yielding a compact BEV vector $\mathbf{b} \in \mathbb{R}^{B\times N\times 1\times 256}$. This spatial vector is subsequently concatenated with the K-step ego features $\{\mathbf{e}_k\}_{k=1}^{K}$ and passed through an MLP and a softmax activation, producing the final physical scores $\hat{s}_{\texttt{Phy}} \in \mathbb{R}^{B\times N}$ across all candidate trajectories. For the cognitive score, we independently compute the alignment metric $\hat{s}_{\texttt{Co}}$ for the predicted candidate trajectories $\hat{\mathcal{T}}$. Specifically, an individual candidate $\hat{\mathcal{T}}^i$ is projected through an MLP and L_2-normalized to extract a trajectory feature vector $\in \mathbb{R}^{B\times 128}$. The cognitive feature $\mathbf{C}$ undergoes the exact same MLP mapping and L_2 normalization to produce a cognitive vector of identical dimensions. Iterating this alignment process over every candidate yields the dense cognitive scores $\hat{s}_{\texttt{Co}} \in \mathbb{R}^{B\times N}$, which mathematically represent the cosine similarity between the two normalized representations.

A.2 Cognitive Threshold Sensitivity

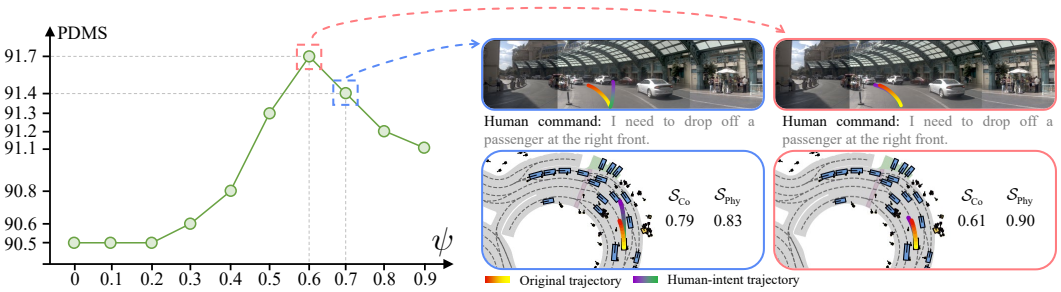


Figure 9: Sensitivity analysis of the cognitive threshold ψ.

As illustrated in the left panel of Fig. 9, increasing ψ progressively enhances the quality of the filtered candidates. A stricter threshold enforces stronger cognitive priors, equipping the model to better decode traffic signals and navigate complex environments. Although pure driving metrics peak at $\psi = 0.6$ before declining, we empirically adopt $\psi = 0.7$ to secure the optimal equilibrium between physical driving safety and strict compliance with human intents.

Complementing this, the right panel of Fig. 9 visualizes the thresholding dynamics. Under a lenient threshold of $\psi = 0.6$, a candidate with a marginal cognitive score of $\hat{s}_{\texttt{Co}} = 0.61$ might survive the filter and be selected solely due to its superior physical score $\hat{s}_{\texttt{Phy}}$, despite blatantly violating the human command. By elevating the threshold to $\psi = 0.7$, this non-compliant candidate is effectively pruned, strictly guaranteeing the selection of an intention-aligned trajectory.

A.3 More Human-Intent Driving Visualizations

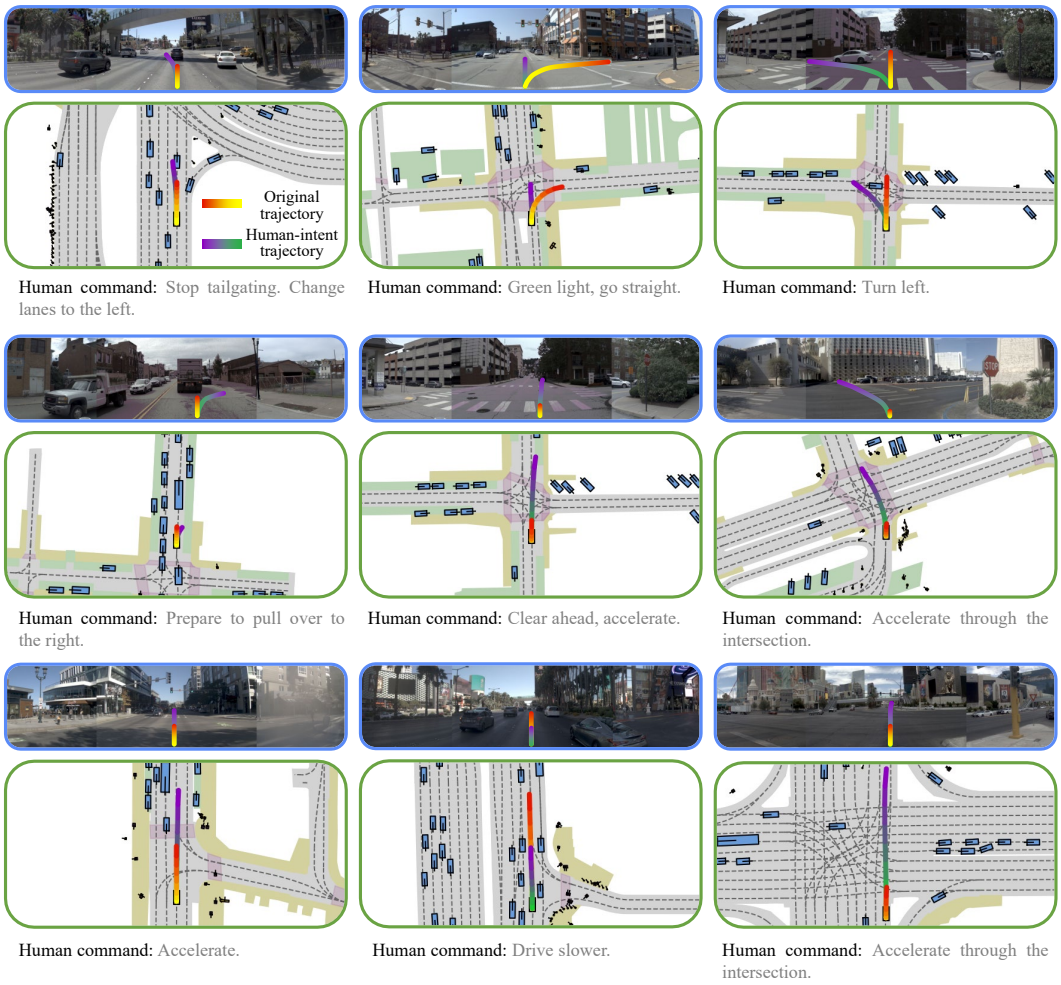


Figure 10: More human-intent driving visualizations.

To further validate our cognitive alignment, Fig. 10 presents diverse human-guided driving scenarios across three distinct categories. The first row showcases direction adjustments, encompassing lane changes and turning maneuvers. For instance, in the second example, the ego vehicle dynamically overrides its initial physical plan to proceed straight through a green light, seamlessly executing a right turn solely based on the cognitive command.

The second row highlights start-and-stop maneuvers, such as transitioning from car-following to pulling over. Notably, in the third example, despite the presence of a STOP sign dictating a halt, the vehicle strictly complies with a human prompt to accelerate through the intersection. This deliberate

rule-breaking underscores a crucial capability: our policy faithfully understands and prioritizes high-level cognitive intents over rigid physical rules when explicitly instructed.

Finally, the third row depicts speed modulations. In the second example, while the physical environment permits high-speed cruising, a human driver might feel unsafe amidst dense traffic. Upon receiving a deceleration command, the vehicle correctly adopts a cautious profile and slows down, perfectly mirroring human driving etiquette.

A.4 More Qualitative Comparisons

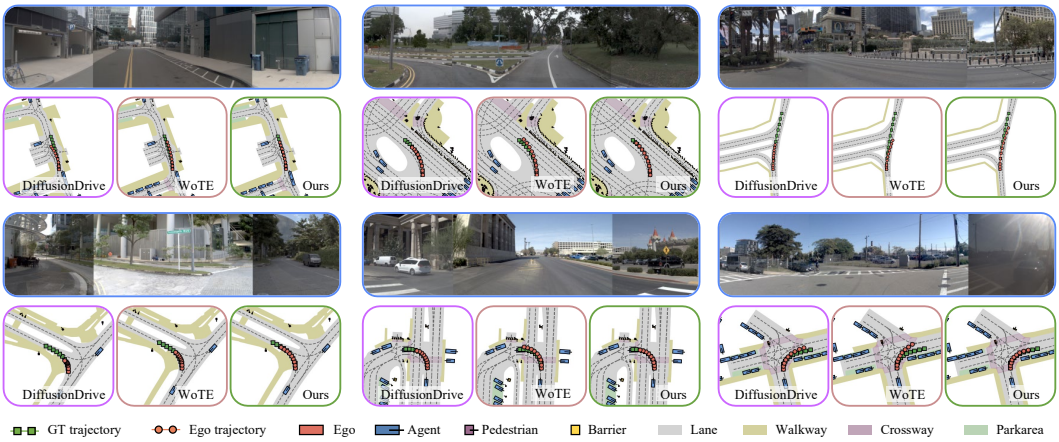


Figure 11: More qualitative comparison results.

To further demonstrate our framework's superiority, Fig. 11 presents qualitative comparisons against popular leading open-source methods [29, 24]. The first row illustrates straight-line and merging scenarios. In the first example, competing baselines illegally cross the double yellow line or encroach upon hatched markings. In contrast, our method maintains strict lane adherence, enforced by the physical constraints derived from world-model rollouts. The second narrow-road example reveals that baselines suffer from sluggish ego progress or trajectory deviations, whereas our policy executes efficient and stable lane-keeping. In the third merging scenario, while our method exhibits a minor trajectory adjustment, it safely remains within the drivable area and progresses smoothly. Conversely, the baselines either drift entirely off-road or erroneously halt at the subsequent fork.

The second row showcases complex left-turn and right-turn maneuvers. Although turning inherently limits the overall ego progress across all methods, ours sustains a noticeably higher speed profile. Benefiting from the seamless integration of high-level cognitive awareness and low-level physical constraints, our agent successfully navigates the intersections without any lane departure.

A.5 Impact of Trajectory Quantity

Table 6: Impact of the number of candidate trajectories N on NAVSIM NAVTEST v1.

N	NC ↑	DAC ↑	EP ↑	TTC ↑	C ↑	PDMS ↑	FPS ↑
64	98.3	98.1	<u>85.4</u>	95.5	**100**	90.6	**65.0**
128	98.7	<u>98.2</u>	85.1	96.2	**100**	91.2	<u>63.1</u>
256	**99.0**	<u>98.2</u>	85.3	**96.8**	**100**	<u>91.4</u>	62.8
512	<u>98.9</u>	**98.3**	**85.6**	<u>96.7</u>	<u>99.9</u>	**91.5**	59.5

We investigate the impact of the candidate trajectory count N on the overall driving performance. As reported in Tab. 6, we vary $N \in \{64, 128, 256, 512\}$ while maintaining all other hyperparameters constant.

Constraining N to a minimal value of 64 yields a notable performance drop, since the limited prototype pool fails to encapsulate diverse multimodal maneuvers, thereby severely restricting the expressiveness of the action space. As N increases, the driving metrics steadily improve but exhibit clear diminishing returns alongside a noticeable degradation in inference speed. Specifically, scaling

N from 256 to 512 provides a marginal 0.1% performance gain at the expense of a 5.3% speed reduction. Consequently, we adopt $N = 256$ as our default configuration to secure the optimal balance between trajectory diversity and computational efficiency.

A.6 Computational Efficiency Analysis

In this section, we evaluate the computational efficiency of our framework against open-source methods and those reporting relevant metrics [39, 20, 63, 69, 29, 24]. As illustrated in Fig. 12, driven by lightweight cognitive distillation and an efficient world model design, our approach achieves the highest performance while maintaining a compact footprint and rapid inference, fully satisfying the rigorous constraints of real-time on-vehicle deployment. Specifically, compared to the fastest and most parameter-efficient baseline, WoTE [24], our method delivers a substantial 3.5% performance improvement at a negligible cost, adding only 1.3% in parameters and slowing down by a mere 1.4%. Furthermore, against the second-best performing method, DriveSuprim [63], our framework not only secures a 1.7% performance margin but also achieves a staggering 130.9% increase in inference speed.

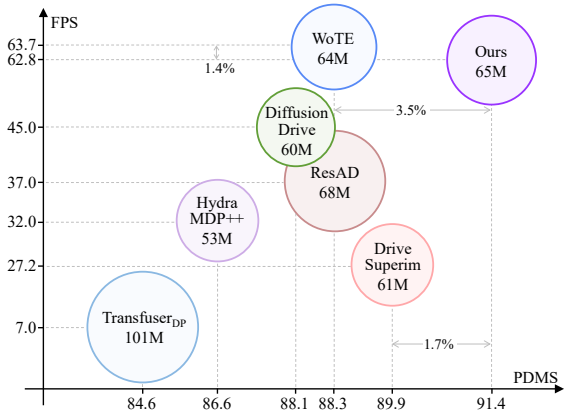


Figure 12: Computational efficiency and performance comparisons.

A.7 VLA Comparison and VLM Distillation Analysis

Table 7: Comparison with VLA baselines and VLM distillation ablation on NAVSIM NAVTEST v1.

Method / VLM Source	NC ↑	DAC ↑	EP ↑	TTC ↑	C ↑	PDMS ↑
Comparison with Vision-Language-Action (VLA) Baselines						
AutoVLA [71]	98.4	97.1	81.9	**97.1**	99.9	89.1
ReCogDrive [25]	98.2	97.8	83.5	95.2	99.8	89.6
DriveVLA-W0 [23]	98.7	**99.1**	83.3	95.3	99.3	90.2
ExplorVLA [47]	98.8	98.4	83.5	96.5	99.9	90.4
Ablation on Different VLM Distillation Sources						
Qwen2.5-7B [1]	97.6	95.4	82.2	93.7	**100**	88.5
Qwen3-8B [1]	97.8	95.7	82.0	93.9	**100**	88.3
InternVL3-8B [72]	97.6	95.9	83.7	94.4	99.9	89.0
Ours (Task-Aligned)	**99.0**	98.2	**85.3**	96.8	**100**	**91.4**

To comprehensively evaluate our approach, Tab. 7 extends the benchmarking to recent Vision-Language-Action (VLA) architectures [71, 23, 25, 47] alongside different VLM distillation variants. To ensure equitable comparisons, all reported VLA baselines are strictly evaluated under a best-of-1 setting [71]. While end-to-end VLAs exhibit strong semantic reasoning, our framework consistently surpasses them by grounding cognitive priors within explicit physical simulations. Furthermore, regarding the distillation source, the results reveal that generic, non-adapted VLMs yield only marginal performance variations. We therefore adopt InternVL3-8B [72] for its robust adaptability, explicitly fine-tuning it to inject domain-specific cognition into our policy. For rigorous benchmarking, all evaluated models undergo uniform LoRA fine-tuning (rank $r = 16$, scaling factor $\alpha = 32$, learning rate 2e-4). Crucially, this task-aligned adaptation unlocks substantial improvements, proving that domain-expert supervision fundamentally dictates the efficacy of cognitive distillation.

Beyond performance metrics, end-to-end VLAs are intrinsically bottlenecked by billion-scale parameter counts and seconds-level inference latency [71, 23], precluding realistic edge deployment. Conversely, by completely decoupling the heavy cognitive engine during inference, our deployed policy operates on a compact million-scale parameter count. This structural elegance unlocks millisecond-level inference, bridging the critical gap between high-level cognitive reasoning and the stringent real-time execution demands of autonomous driving.

A.8 Future BEV Map Prediction

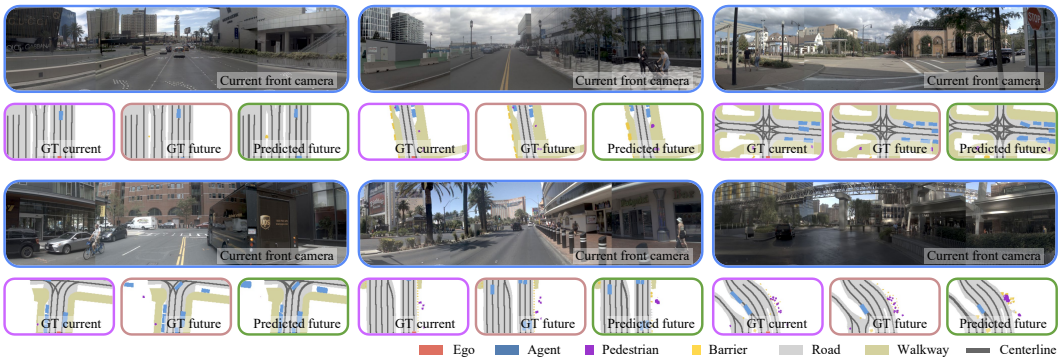


Figure 13: Qualitative results of future BEV semantic map prediction.

To visually validate our predictive capabilities, Fig. 13 showcases the current front-view camera observation, the corresponding current BEV semantic map, and the predicted future BEV semantic map at the 4-second horizon. These visualizations corroborate the world model's proficiency in accurately forecasting the evolving spatial configurations of road boundaries, lane markings, and dynamic agents. While exceptionally small objects might present marginal boundary imprecision, their core semantic occupancies remain strictly preserved, ensuring that these localized artifacts never compromise the overarching structural integrity of the generated environments. Crucially, this predictive capability establishes a high-fidelity sandbox for the reinforcement learning phase. Grounded in this reliable spatial context, the dual-score trajectory evaluation ultimately empowers the policy to execute exceptionally safe and intention-aware planning decisions.

A.9 Customizable Driving Behavior via Physical Reward Modulation

Table 8: Effect of different weights of physical rewards on NAVSIM NAVTEST v1.

Policy priorities	NC ↑	DAC ↑	EP ↑	TTC ↑	C ↑	PDMS ↑
Speed and efficiency priority	98.1	96.2	86.6	92.5	99.9	90.4
Comfort and safety priority	99.0	98.5	81.1	97.0	100	90.9
Balanced priorities	99.0	98.2	85.3	96.8	100	91.4

In addition to high-level intent control via human language commands, our framework also supports microscopic preference customization directly through its physical reward mechanism. In real-world autonomous driving deployments, humans exhibit highly diverse preferences. While some passengers prefer efficient and fast commutes, others prioritize smooth kinematics and conservative safety margins. Because our CoPhy framework explicitly models these factors through the physical reward formulation $\mathcal{R}_{Phy}$ during the reinforcement learning phase, it offers an elegant mechanism to accommodate these distinct preferences. By dynamically adjusting the weighting coefficients $\omega_{1\ldots5}$, we can tailor the driving policy to execute specific driving priorities without requiring any structural modifications to the underlying model architecture.

Recall the physical reward formulation for the i-th candidate trajectory:

$$\mathcal{R}^i_{\texttt{Phy}} = \omega_1 \mathcal{R}^i_{\texttt{NC}} + \omega_2 \mathcal{R}^i_{\texttt{DAC}} + \omega_3 \mathcal{R}^i_{\texttt{EP}} + \omega_4 \mathcal{R}^i_{\texttt{TTC}} + \omega_5 \mathcal{R}^i_{\texttt{Comf}}$$

To ensure foundational driving safety, the weights for the hard boolean constraints, no collision (ω_1) and drivable area compliance (ω_2), remain strictly fixed across all variants. We adjust the remaining coefficients to define three distinct policy priorities.

(1) Balanced priority: This represents our default configuration designed to strike an optimal balance between safety and efficiency. Following our primary experimental setup, the weights are set to $\omega_3 = 0.4$, $\omega_4 = 0.4$, and $\omega_5 = 0.2$.

(2) Comfort and safety priority: This configuration prioritizes passenger comfort and extensive safety margins over task efficiency, making it highly suitable for elderly individuals or families with infants.

By increasing the weight for Comfort, and decreasing the weight for Ego Progress (e.g., $\omega_3 = 0.1$, $\omega_4 = 0.4$, $\omega_5 = 1.0$), the policy is heavily incentivized to maintain larger follow distances and execute softer kinematic maneuvers.

(3) Speed and efficiency priority: This configuration maximizes travel efficiency while maintaining baseline safety. By up-weighting Ego Progress and marginally relaxing the optimization for Time-to-Collision and Comfort (e.g., $\omega_3 = 1.0$, $\omega_4 = 0.2$, $\omega_5 = 0.1$), the policy actively seeks safe opportunities to overtake or maintain higher cruising speeds. This effectively addresses the inherent trade-off between safety and progress characteristic of standard imitation learning bounds.

As shown in Tab. 8, by simply modulating the reward coefficients, our framework produces distinctly different driving behaviors while consistently maintaining high overall performance, demonstrating the practical flexibility of CoPhy for diverse real-world deployment preferences.

A.10 Limitations and Broader Impacts

While our framework demonstrates robust performance, its real-world deployment must be rigorously contextualized within the systemic challenges inherent to the broader autonomous driving domain.

First, a persistent limitation across current methodologies, including ours, is the inability to autonomously adapt to the highly divergent traffic regulations across global jurisdictions. This geographic generalization bottleneck inevitably necessitates redundant, region-specific policy fine-tuning.

Furthermore, the rapid advancement of autonomous driving technologies significantly outpaces the evolution of corresponding legal and regulatory frameworks. This discrepancy creates critical ambiguities regarding accident liability and accountability in complex scenarios.

Finally, the aggressive commercialization of these intelligent systems poses profound socioeconomic risks. The potential widespread displacement of professional human drivers could precipitate substantial job losses, inadvertently catalyzing wealth centralization and exacerbating societal inequity. Addressing these multidimensional challenges requires unprecedented collaborative efforts among technologists, policymakers, and ethicists to ensure the responsible integration of autonomous systems into global infrastructure.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Please refer to Secs. 3, 4, and App. A.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Please refer to App. A.10.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: Please refer to Secs. 3, 4, and App. A.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Please refer to Secs. 3, 4, and App. A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Please refer to Secs. 3, 4, and App. A. The code will be made publicly available immediately after the paper is accepted and passes the commercial confidentiality review.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Please refer to Sec. 4 and App. A.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

   Justification: Due to high computational costs and domain-specific paradigms, we have not provided error bars.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please refer to Sec. 4.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Our research conforms to the standards in every respect.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Please refer to App. A.10.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our model is a domain-specific driving policy and does not pose high misuse risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We respect all assets, and have added citations wherever they are involved.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Please refer to Sec. 4 and App. A. The code will be made publicly available immediately after the paper is accepted and passes the commercial confidentiality review.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: Our paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Please refer to Secs. 3, 4, and App. A.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.