Prioritizing Perception-Guided Self-Supervision: A New Paradigm for Causal Modeling in End-to-End Autonomous Driving

Yi Huang[1,2]*, **Zhan Qu**[2]*, **Lihui Jiang**[2]†, **Bingbing Liu**[2], **Hongbo Zhang**[2]
[1]The Chinese University of Hong Kong, Shenzhen
[2]Huawei Noah's Ark Lab
yihuang11@link.cuhk.edu.cn
{quzhan, jianglihui1, liu.bingbing, zhanghongbo888}@huawei.com

Abstract

End-to-end autonomous driving systems, predominantly trained through imitation learning, have demonstrated considerable effectiveness in leveraging large-scale expert driving data. Despite their success in open-loop evaluations, these systems often exhibit significant performance degradation in closed-loop scenarios due to causal confusion. This confusion is fundamentally exacerbated by the overreliance of the imitation learning paradigm on expert trajectories, which often contain unattributable noise and interfere with the modeling of causal relationships between environmental contexts and appropriate driving actions. To address this fundamental limitation, we propose Perception-Guided Self-Supervision (PGS)—a simple yet effective training paradigm that leverages perception outputs as the primary supervisory signals, explicitly modeling causal relationships in decision-making. The proposed framework aligns both the inputs and outputs of the decision-making module with perception results—such as lane centerlines and the predicted motions of surrounding agents—by introducing positive and negative self-supervision for the ego trajectory. This alignment is specifically designed to mitigate causal confusion arising from the inherent noise in expert trajectories. Equipped with perception-driven supervision, our method—built on a standard end-to-end architecture—achieves a Driving Score of 78.08 and a mean success rate of 48.64% on the challenging closed-loop Bench2Drive benchmark, significantly outperforming existing state-of-the-art methods, including those employing more complex network architectures and inference pipelines. These results underscore the effectiveness and robustness of the proposed PGS framework, and point to a promising direction for addressing causal confusion and enhancing real-world generalization in autonomous driving.

1 Introduction

Autonomous driving, as a significant application of AI, has made impressive advancements in recent years. End-to-end neural networks, which allow vehicles to make decisions directly from raw sensor signals, are considered capable of overcoming the cumulative error issues inherent in traditional modular approaches and offer the potential to scale with vast amounts of data. In mainstream systems, perception, prediction, and planning tasks are integrated into a single network. The planning module uses explicit or implicit representations of the environment provided by perception to plan the future

*Equal contribution.
†Corresponding author.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

behavior, with human or expert trajectories being used as the target of training. Researchers have made significant efforts to leverage large-scale human driving data to enable models to learn the relationship between environmental context and vehicle behavior.

As a classic paradigm for end-to-end systems, imitation learning became prominent alongside early benchmarks such as nuPlan [3], Argoverse [4], Oxford RobotCar [2]. These benchmarks typically provide open-loop metrics, with L2 error between predicted and ground-truth trajectories as the key indicator. Consequently, researchers focus on designing complex network architectures, incorporating multi-modal sensor information, and using imitation learning objective functions aligned with these metrics, to enhance the model's ability to fit expert trajectories. However, recent studies have shown that trajectory fitting in open-loop evaluation cannot accurately reflect system performance in real-world scenarios [15, 19, 28]. In closed-loop simulation tests, pure imitation learning models often show significant degradation in safety, comfort, and feasibility in complex scenarios. This inability to generalize in real-world environments has become a major challenge for end-to-end systems.

Among the factors affecting the closed-loop performance, the most significant is causal confusion. Causal confusion refers to the model's inability to associate driving behavior with the primary causal factors in the environment, instead linking it to other noise factors. Although recent end-to-end approaches have reduced input noise by using sparse instance-level representation [14] of the environment, these methods still fail to fully address this problem. In this paper, We identify causal confusion as an unavoidable byproduct of the imitation learning framework, stemming from its reliance on suboptimal expert data. Expert or human trajectories often contain noise from factors like driving style, time of day, or control errors, making them suboptimal supervision targets. Learning from such noisy signals weakens the model's ability to capture true causal relationships. We argue that causal confusion stems not just from imperfect inputs, but more critically from noise in the supervision itself.

Unlike prior approaches that treat perception and prediction modules merely as feature extractors, we propose a framework leverages their outputs as primary supervision signals for decision-making. By aligning both the inputs and outputs of the decision-making module with perception outputs, our perception-guided self-supervision paradigm exhibits stronger causal modeling capabilities in closed-loop evaluations than pure imitation learning. Specifically, we introduce three novel self-supervision mechanisms: Multi-Modal Trajectory Planning Self-Supervision (MTPS), Spatial Trajectory Planning Self-Supervision (STPS), and Negative Trajectory Planning Self-Supervision (NTPS). MTPS and STPS utilize lane centerlines to enforce topological constraints and support multimodal decision-making across available lanes. NTPS incorporates the predicted future trajectories of dynamic agents as negative supervision to guide the ego vehicle away from potential collisions. In this framework, human expert trajectories are used to filter or regularize self-supervision targets when perception-based guidance is unavailable.

In summary, we propose an innovative training paradigm for end-to-end autonomous driving systems, which does not rely on specialized network designs but emphasizes the use of perception-guided self-supervision as the main learning objective. Our contributions include the following:

1. **Multi-Modal Trajectory Planning Self-Supervision as Target Lane Selection:** We reformulate multi-modal ego decision-making as a target lane selection problem based on lane perception. This approach enhances the system's ability to associate surrounding obstacles and available lanes with appropriate driving decisions, thereby improving the performance of lane-change planning.

2. **Spatial Trajecotry Planning Self-Supervision based on lane centerline:** We take the lane centerline outputted from perception module as a spatial trajectory without temporal dependency, and use them as the primary learning target for planning ego trajectory. This design effectively reduces lane departures and mitigating causal confusion induced by inconsistent and noisy expert demonstrations.

3. **Negative Trajectory Planning Self-Supervision from Dynamic Objects' Future bounding box:** Our framework selects and utilizes the predicted future trajectories of surrounding agents as negative supervision signals for ego trajectory learning, enforcing non-overlapping constraints between future bounding boxes. This facilitates the learning of interactions with dynamic agents and reduces collision risk.

4. We made minimal modifications to a simple end-to-end network architecture to adapt and validate our proposed self-supervision training paradigm. In experiments on the challenging closed-loop benchmark, Bench2Drive, the self-supervised model outperformed the pure imitation learning version of the same architecture and recent works using more complex network structures and pipelines by a large margin.

2 Related Work

End-to-end autonomous driving aims to generate planning trajectories directly from raw sensors. In the field, advancements have been categorized based on their evaluation methods: open-loop and closed-loop systems. We reviewed representative works based on these two evaluation schemes in the first and second subsections, and summarized existing techniques and improvements addressing the causal confusion in the third subsection.

2.1 Open-Loop End-to-End Driving Methods

In open-loop systems, UniAD [8] proposes a unified framework that integrates full-stack driving tasks with query-unified interfaces, enhancing task interaction. VAD [14] improves planning safety and efficiency, as demonstrated by its performance on the nuScenes dataset. SparseDrive [23] uses sparse representations to mitigate information loss and error propagation in modular systems. ParaDrive [26] organizes perception, motion prediction, and planning tasks in a parallelized architecture during training, retaining only the planning module in inference. This approach improves planning performance and significantly reduces runtime latency

2.2 Closed-Loop End-to-End Driving Methods

Existing works (e.g., BEVPlanner [19]) have found that metrics like L2 error and collision rate used in open-loop evaluations do not comprehensively reflect model performance in real-world scenarios. As a result, more approaches are being proposed for closed-loop evaluation. VADv2 [5] advances vectorized autonomous driving by generating action distributions for vehicle control, achieving outstanding performance on the CARLA Town05 benchmark. Transfuser [6] uses transformer modules at multiple resolutions to fuse perspective and bird's-eye view feature maps, outperforming prior work on the CARLA leaderboard. Hydra-MDP [17] employs knowledge distillation from both human and rule-based teachers to train the student model, enabling the selection of the trajectory with optimal overall performance and securing first place in the Navsim challenge. DriveTransformer [13] delves into task parallelism and sparse representation in architecture design, significantly improving driving scores and success rates on the Bench2Drive benchmark.

2.3 Techniques Addressing Causal Confusion in Autonomous Driving

Causal confusion has been a persistent challenge in imitation learning. In end-to-end driving, ChauffeurNet [1] addresses this issue by using past ego-motion as intermediate BEV abstractions, and randomly dropping them during training. PrimeNet [25] improves performance by incorporating predictions from a single-frame model as additional input to a multi-frame model. DriveAdapter [11] mitigates the influence of noisy perception outputs by training a strong planner with privileged perception information, and aligning perception model output with the planner's input through an adapter. RAD [7] proposes a 3DGS-based closed-loop reinforcement learning framework, which uses specialized rewards to guide the policy in understanding real-world causal relationships more effectively. These approaches primarily aim to mitigate causal confusion by suppressing noise in the inputs to the planning module. In this paper, we propose a novel perspective and an innovative training paradigm, where perception-guided self-supervision plays a key role in addressing causal confusion by aligning the input and output of the planning module.

3 Method

In this section, we introduce a Prioritizing Perception-Guided Self-Supervision training paradigm built upon a typical end to end architecture. On one hand, sparse instance-level features extracted from perception are used as inputs to the unified decision and prediction module, helping minimize

input noise. On the other hand, the perception output is directly employed for self-supervision of the planning process. This alignment plays an important role in helping the planner learn causal relationships.

3.1 End-to-End Network Baseline

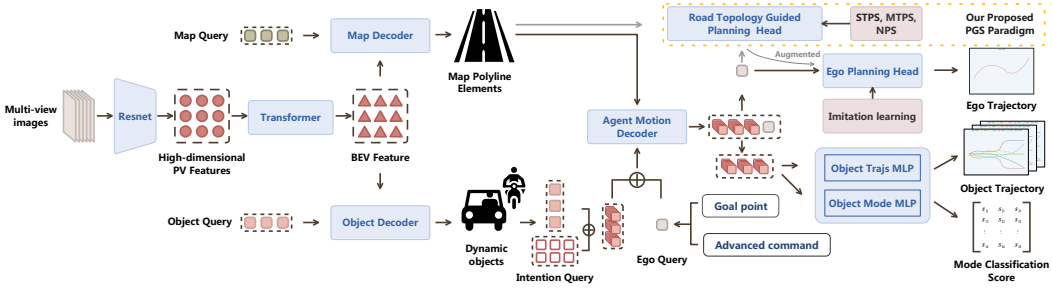


Figure 1: Overall Model Architecture. (1) Perception network provides map elements and future motions of dynamic agents. (2) The dashed box presents the proposed PGS, which generates three self-supervised signals from perception outputs to enhance causal reasoning in ego planning.

Our baseline architecture is simple and similar to VAD [14]. As illustrated in Figure 1, multi-view images are encoded into high-dimensional features in perspective view using a Resnet-based encoder. BEVFormer [18] then transform the features into BEV (Bird's Eye View) space as F_{BEV}.

For task-specific applications, learnable embeddings are used to query the BEV features, extracting sparse representations of the environment. The map query interacts with the BEV features to decode static road topology such as lane markings, lane centerlines, sidewalks, and other structures. The object query decodes information about dynamic objects (pedestrians, cyclists, and vehicles), including their positions, sizes, orientations, and velocities at the current timestep. The perception process can be formally expressed as follows:

$$q_i' = \text{PerceptionDecoder}\big(q_i, kv = F_{\text{bev}}\big), \quad q_i \in Q_{\text{map}} \cup Q_{\text{obj}}, \tag{1}$$

$$\hat{y}_i^{\text{map}} = \text{MLP}_{\text{map}}\big(q_i'^{\text{map}}\big) = \big(\{p_j^{(i)}\}_{j=1}^{N_{\text{points}}}, c^{(i)}\big), \quad p_j^{(i)} \in \mathbb{R}^2, \tag{2}$$

$$\hat{y}_i^{\text{obj}} = \text{MLP}_{\text{obj}}\big(q_i'^{\text{obj}}\big) = (x_t, y_t, w, l, \theta_t, v_t^x, v_t^y), \tag{3}$$

where q_i' is the enhanced instance-level query, $\hat{y}_i^{\text{map}}$ and $\hat{y}_i^{\text{obj}}$ are decoded information of map element and dynamic object. The former are represented as polyline, where $p_j^{(i)}$ is the 2D coordinates of the j-th point of map element, and $c^{(i)}$ denotes its class score. The latter includes the 2D coordinates of the object's center, its width and length, the heading angle, and the velocity components along the x and y axes at time t.

Next, instead of adopting the cascaded prediction–decision architecture, a unified decoder for both ego planning and object motion prediction is employed. Specifically, a learnable ego query q_{ego} is augmented with a goal point embedding and an intent embedding corresponding to the high-level command c. For other agents, each motion query is formed by combining the updated object query $q_i'^{\text{obj}}$ with one of six implicit intent embeddings e. The two sets of queries are concatenated to form the agent-level motion query for the current scenario:

$$Q_{\text{motion}} = \text{Concat}\left(\text{MLP}_{\text{ego}}(q_{\text{ego}} \oplus g \oplus c), \text{MLP}_{\text{obj}}\left(\{q_{\text{obj}}'^{(i)} \oplus e_k, k = 1 \ldots K, i = 1 \ldots N_{\text{obj}}\}\right)\right) \tag{4}$$

In the unified decoder, agent-level motion queries attend to each other via self-attention and to map queries via cross-attention, enabling the model to capture interactions between agents constrained by the current road topology. The refined motion queries are again split into those for the ego and dynamic objects, which passes through seperated MLP for predicting the future trajectories. The planning and motion prediction process can be formally expressed as follows:

$$Q'_{\text{motion}} = \text{MotionDecoder}(q = Q_{\text{motion}}, kv = Q'_{\text{map}}) \tag{5}$$

$$\hat{m}_j^k = \sigma\left(\text{MLP}_{\text{mod}}\left(q_{\text{m\_o}}^{j,k}\right)\right),\ \hat{\text{Traj}}_{\text{obj}}^{j,k} = \text{MLP}_{\text{traj\_o}}\left(q_{\text{m\_o}}^{j,k}\right),\ \text{where } k = 1,\ldots,K,\ j = 1,\ldots,N_{\text{obj}},\ q_{\text{m\_o}}^{j,k} \in Q'_{\text{motion\_obj}} \tag{6}$$

$$\hat{\text{Traj}}_{\text{ego}} = \text{MLP}_{\text{traj\_e}}(q_{\text{m\_e}}),\quad q_{\text{m\_e}} = Q'_{\text{motion\_ego}} \tag{7}$$

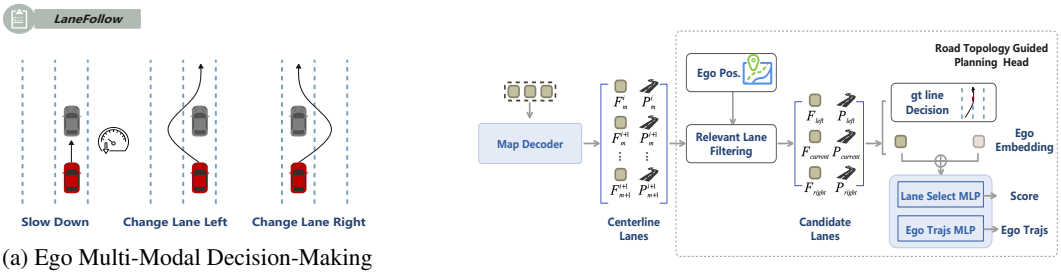
where σ denotes the sigmoid function; $\hat{m}_j^k$ represents the score of the k-th predicted modality for the j-th object, and $\hat{\text{Traj}}_{obj}^{j,k}$ denotes the corresponding trajectory over the prediction horizon T; $\hat{\text{Traj}}_{\text{ego}}$ is the planned trajectory of the ego vehicle.

In training phase, the loss terms of perception are same as in VAD [14], with imitation loss of L1 norm:

$$L_{\text{total}} = w_{\text{det\_map}}L_{\text{det\_map}} + w_{\text{det\_obj}}L_{\text{det\_obj}} + w_{\text{mod\_cls}}L_{\text{mod\_cls}} + w_{\text{motion\_obj}}L_{\text{motion\_obj}} + w_{\text{imi}}L_{\text{imi}} \tag{8}$$

The outputs of the perception and motion prediction tasks—namely, the implicit high-dimensional features and the structured trajectories and polylines—constitute the foundation of our self-supervised paradigm.

3.2 Multi-Modal Trajectory Planning Self-Supervision (MTPS) as Target Lane Selection



(a) Ego Multi-Modal Decision-Making in Cruising Scenarios (Lane Follow)

(b) Road Topology Guided Planning Head

Figure 2: Environment-Aware Lane Command and Road Topology Guided Planning.

Lane status in the surrounding environment plays a crucial role in guiding the ego's driving behavior. It defines the drivable area and constrains the range of feasible trajectories. As shown in Figure 2a, when the command is "LaneFollow" but an obstacle appears ahead, the ego can either decelerate in the current lane or change lanes to overtake. This turns lateral multi-modal planning into a lane selection problem shaped by the environment. This insight forms the basis of our Multi-Modal Trajectory Planning Self-Supervision(MTPS).

MTPS leverages the surrounding lane structure to guide the selection of the ego vehicle's planning modality. As illustrated in Figure 2b, this module includes a Road Topology Guided Planning Head, which generates both multi-modal decisions and corresponding ego trajectories, alongside a topology-aligned self-supervision mechanism. Firstly, a geometry-based lane filter is leveraged to select ego-relevant lane centerlines from the perception output. Given the set of lane centerline $P = \{P_j\}_{j=1}^{N_{map}}$, with each P_j represented as a sequence of centerline points, we compute the minimum Euclidean distance d_j and the relative angle φ_j between the ego vehicle's current position x_{ego}, heading θ_{ego} and each centerline. The relevant lane set $FP = \{(F_{\text{left}}, P_{\text{left}}), (F_{\text{current}}, P_{\text{current}}), (F_{\text{right}}, P_{\text{right}})\}$ is constructed according to the following criteria, where F denotes the implicit feature of each lane:

$$\begin{cases} d_j = \min\limits_{p \in P_j} \|x_{\text{ego}} - p\|_2 \\ \varphi_j = \left(p_j^* - x_{\text{ego}}\right) \times \begin{bmatrix} \cos\theta_{\text{ego}} \\ \sin\theta_{\text{ego}} \end{bmatrix} \end{cases},\quad \begin{cases} (F_{\text{c}}, P_{\text{c}}) = (F_j, P_j), & \text{if } \exists d_j \le 0.5W \\ (F_{\text{l}}, P_{\text{l}}) = (F_j, P_j), & \text{if } \exists d_j \in (0.5W, 1.5W] \text{ and } \varphi_j < 0 \\ (F_{\text{r}}, P_{\text{r}}) = (F_j, P_j), & \text{if } \exists d_j \in (0.5W, 1.5W] \text{ and } \varphi_j > 0 \\ (F, P) = (\mathbf{0}, \mathbf{0}), & \text{otherwise} \end{cases} \tag{9}$$

where p_j^* is the nearest point in P_j to the ego vehicle, W is the standard lane width, and the subscripts c, l, and r denote the current, left, and right. If no centerline satisfies the above criteria, the corresponding feature F and point set P are set to zero.

This geometry-based filter is simple yet effective, allowing robust and efficient identification of the ego vehicle's current lane and adjacent lanes, thereby capturing all feasible lateral motion options.

Next, the implicit features of relevant centerlines are fused with the ego motion query. Two additional MLPs are utilized to predict the lane selection score and the corresponding trajectory. These scores are normalized by softmax operator, transforming the ego's multi-modal trajectory planning into a lane-level classification task, as described below:

$$H_i = q \oplus F_i,\ \forall F_i \in FP, \quad \boldsymbol{S} = \text{softmax}(\text{MLP}_{\text{score}}(\boldsymbol{H}))\,, \quad \hat{\text{Traj}}_{\text{ego}} = \text{MLP}_{\text{traj}}\left(H_{\text{argmax}(\boldsymbol{S})}\right) \tag{10}$$

During training, the index of target centerline is provided by measuring the average spatial distance between the terminal portion of the ground-truth ego trajectory and each candidate lane centerline. The index of the centerline with the minimal distance is designated as the target lane index l^*, with the corresponding feature and polyline as F^* and P^*. The loss function of selecting target lane is defined as:

$$L_{\text{MTPS}} = L_{\text{CE}}(\boldsymbol{S}, l^*) \tag{11}$$

3.3 Spatial Trajectory Planning Self-Supervision (STPS) based on Lane Centerline

Lane centerlines, compared to other road topology cues like markings and boundaries, play a more critical role in learning robust driving behaviors. Human trajectories often deviate slightly from the centerline due to factors like driving style, weather, or control noise—difficult to attribute and thus regarded by the model as learning noise. This noisy supervision can negatively impact the model's causal understanding of driving behaviors, particularly in scenarios involving intersection turning. Lane centerlines naturally connect incoming and outgoing lanes, and training on trajectories that deviate from them increases the risk of drifting into the wrong lane due to cumulative errors.

Building on this insight, we propose a Spatial Trajectory Planning Self-Supervision (STPS) mechanism, in which the expert trajectory $Traj_{gt}$ is replaced by a centerline-aligned version as the primary supervision signal. Specifically, each expert trajectory point is checked against nearby target centerline points (from the previous stage); if a matched centerline point is found, it replaces the original point. Original expert point is retained only when no point is matched, which serves as a regularization term to preserve the smoothness of the target trajectory. The resulting trajectory $Traj'_{tgt}$ supervises the Road topology guided trajectory regression head, working as a spatial ground-truth path—free of temporal bias but more causally aligned. Formally, for expert trajectory point $Traj^t_{gt}$ at time step t, the resulting updated trajectory points $Traj'^t_{tgt}$ are given by:

$$p'_t = \arg\min_{p'_j \in P^*} \|Traj^t_{gt} - p'_j\|_2 \tag{12}$$

$$Traj'^t_{tgt} = \begin{cases} p'_t, & \text{if } \|Traj^t_{gt} - p'_t\|_2 \le w \\ Traj^t_{gt}, & \text{otherwise} \end{cases} \tag{13}$$

This new trajectory $Traj'_{tgt}$ is then used to supervise the trajectory regression head as:

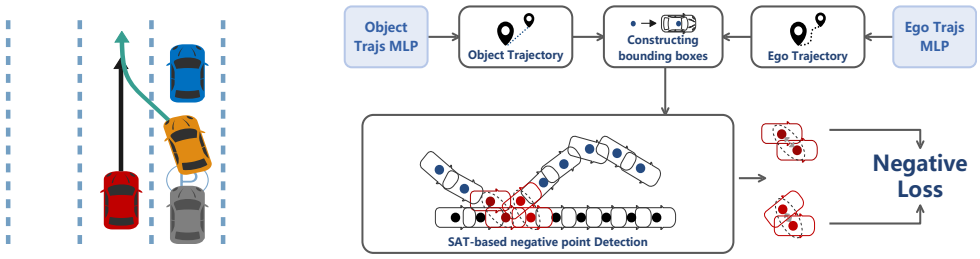
$$L_{\text{STPS}} = \frac{1}{N}\sum_{t=1}^{N} \|\hat{Traj}^t_{ego} - Traj'^t_{tgt}\|_1 \tag{14}$$

Last but not least, since the regression head also takes the target centerline features F^* as input, aligning the trajectory target with the centerline further strengthens causal reasoning in trajectory prediction by jointly leveraging topological cues and supervision consistency.

3.4 Negative Planning Self-supervision (NPS) from Dynamic Objects' Future Bounding Boxes

An autonomous systems must dynamically respond to surrounding agents. While MTPS and STPS support positive causal modeling for general planning, safe interaction requires negative causal modeling to proactively avoid risky outcomes—e.g., adjusting the ego trajectory to prevent overlap with the predicted motion of an encroaching parked vehicle as shown in the Figure 3a.

Motivated by this insight, we propose the Negative Trajectory Planning Self-Supervision (NTPS) mechanism, which imposes safety constraints on ego planning using the predicted future bounding boxes of surrounding agents. As illustrated in Figure 3b, we construct future bounding box sequences for both ego and dynamic objects using predicted trajectories and perceived object dimensions. The



(a) Example of Risky Interaction Scenario (b) Safety Constraint Mechanism of NTPS

Figure 3: Negative Trajectory Planning Self-Supervision (NTPS) for Safety-Constrained Ego.

orientation at each timestep is estimated via trajectory offsets, and overlap detection is performed using the Separating Axis Theorem (SAT) [9]. Upon detecting overlaps, we introduce a negative supervision loss that penalizes ego trajectories encroaching into occupied space by encouraging divergence from the overlapping region. as follows:

$$L_{\text{NTPS}} = \sum_{t \in T_{\text{coll}}} \max(0, \beta - \|\hat{Traj}^{t}_{\text{ego}} - \hat{Traj}^{t}_{\text{obj\_col}}\|_2) \tag{15}$$

where $t \in T_{\text{coll}}$ denotes each timestep in the set of detected collision timesteps T_{coll}, and $\hat{\text{Traj}}^{t}_{\text{obj\_col}}$ represents the trajectory point of the surrounding object that is predicted to collide with the ego vehicle at timestep t.

In this process, SAT-based overlap detection identifies risk-inducing points along the trajectories of surrounding agents. These are treated as negative supervision signals, guiding the ego trajectory to diverge from potential collision zones and thereby enhancing safety in dynamic interactions.

3.5 Perception-Guided Self-supervision in Optimization

During training, the proposed PGS paradigm introduces guidance from perception by integrating the three distinct self-supervision losses described above to total loss introduced in Section 3.1 as:

$$L'_{\text{total}} = L_{\text{total}} + w_{\text{MTPS}} L_{\text{MTPS}} + w_{\text{STPS}} L_{\text{STPS}} + w_{\text{NTPS}} L_{\text{NPS}} \tag{16}$$

where w represent the relative importance of each loss component.

4 Experiment

4.1 Dataset & Metrics

Dataset: To evaluate the real-world effectiveness of our self-training paradigm, we evaluate on the challenging closed-loop benchmark Bench2Drive [10], built on CARLA v2. The dataset includes 1,000 short clips across 44 complex scenarios (950 for training, 50 for open-loop validation) and 220 predefined routes for standardized closed-loop evaluation nd fair performance comparison.

Metrics: We adopt Bench2Drive's official metrics: Driving Score, Success Rate, Efficiency, and Comfortness for closed-loop evaluation, and L2 Displacement Error (L2) for open-loop evaluation.

4.2 Implementation Details

The training process of PGS is divided into two stages, each with distinct learning objectives.

Stage 1 focuses on perception learning. We enhance the online map detection task by introducing lane centerlines as a new class of map elements. While the task of motion prediction of dynamic objects is trained in this phase as well. In addition, traffic light detection from front-view images is incorporated to capture critical causal dependencies for safely navigating signalized intersections. Training in this stage lasts for 6 epochs.

Stage 2 builds upon the perception capabilities from Stage 1 and introduces joint optimization of the perception module and self-supervised objectives. Perception losses are retained to maintain accurate environmental understanding, while three self-supervised losses—MTPS, STPS, and NTPS—are introduced to supervise ego planning tasks. Stage 2 is also trained for 6 epochs.

Training is conducted on 16 NVIDIA RTX V100 GPUs using the AdamW [21] optimizer, with a weight decay of 0.01 and an initial learning rate of 4e-4. The loss weights are set to $w_{\text{MTPS}} = 1.0$, $w_{\text{STPS}} = 0.3$, and $w_{\text{NPS}} = 1.0$.

4.3 Comparison with State-of-the-Art Methods

Table 1: Open-loop and Closed-loop results of planning in Bench2Drive. Avg. L2 is averaged over the predictions in 2 seconds under 2Hz. * denotes expert feature distillation.

Method	Avg. L2 ↓	Driving Score ↑	Success Rate (%) ↑	Efficiency ↑	Comfortness ↑
AD-MLP [28]	3.64	18.05	0.00	48.45	22.63
UniAD-Base [8]	0.73	45.81	16.36	129.21	43.58
UniAD-Tiny [8]	0.80	40.73	13.18	123.92	47.04
VAD-Base [14]	0.91	42.35	15.00	157.94	46.01
VAD-Tiny [14]	1.15	34.28	10.45	70.04	**66.86**
SparseDrive [23]	0.87	44.54	16.71	170.21	48.63
GenAD [29]	-	44.81	15.90	-	-
DiFSD [22]	0.70	52.02	21.00	178.30	-
DriveTransformer [13]	**0.62**	63.46	35.01	100.64	20.78
DiffAD [24]	-	67.92	38.64	-	-
WoTE [16]	-	61.71	31.36	-	-
BridgeAD	0.71	50.06	22.73	-	-
PGS(Ours)	0.77	**78.08**	**48.64**	**181.31**	12.37
TCP-traj* [27]	1.70	59.90	30.00	76.54	18.08
ThinkTwice* [12]	0.95	62.44	31.23	69.33	16.22
DriveAdapter* [11]	1.01	64.22	33.08	70.22	16.01

Table 1 summarizes the comparative open-loop and closed-loop planning performance on Bench2Drive. Compared to VAD-Base—the baseline model for our approach—PGS reduces the open-loop L2 error from 0.91 to 0.77. more importantly, PGS achieves a remarkable Driving Score of 78.08, outperforming VAD-Base (42.35) by 35.73 points in closed-loop evaluation. The Success Rate improves significantly from 15.00% to 48.64%. These improvements are primarily attributed to the enhanced causal reasoning capabilities introduced by our self-supervised planning framework.

The comparison with contemporaneous methods [13, 24, 16] further validates the effectiveness of our proposed paradigm. These methods improve closed-loop performance by adopting more complex architectures, leveraging multi-modal sensor inputs, employing diffusion models for multi-modal decision, or combining trajectory generation with online ranking strategies, but still primarily rely on imitation learning and lack explicit consideration of causal reasoning. In contrast, the self-supervised, causality-driven PGS framework consistently outperforms these methods, highlighting the effectiveness of perception-guided self-supervision in capturing causal relationships.

Furthermore, PGS surpasses methods based on knowledge distillation (e.g., [27, 12, 11]) as well. Although distillation enhances planner robustness by transferring knowledge from expert models trained with noise-free privileged information, it fails to model the causal dependencies between redundant perception outputs and ego planning, and results in causal confusion and suboptimal decision-making policies. These results further validate the superiority of PGS in achieving causally grounded and robust driving performance.

Table 2 further compares the success rates of different approaches across specific driving scenarios. A scenario is considered successful only if the ego vehicle reaches the designated destination without any collisions or infractions. Our model consistently outperforms competitors across several critical driving skills, achieving notably high success rates in Merging (35.00%), Overtaking (73.33%), Emergency Braking (55.00%), and Give Way (60.00%). It also obtains the highest overall ability score of 53.40%. These results highlight the strong generalization capability of our approach in handling complex and highly interactive scenarios.

Table 2: **Multi-Ability Results of E2E-AD Methods.** * denotes expert feature distillation.

Method	Ability (%) ↑					
	Merging	Overtaking	Emergency Brake	Give Way	Traffic Sign	**Mean**
AD-MLP [28]	0.00	0.00	0.00	0.00	4.35	0.87
UniAD-Tiny [8]	8.89	9.33	20.00	20.00	15.43	14.73
UniAD-Base [8]	14.10	17.78	21.67	10.00	14.21	15.55
VAD [14]	8.11	24.44	18.64	20.00	19.15	18.07
DriveTransformer [13]	17.57	35.00	48.36	40.00	52.10	38.60
DiffAD [24]	30.00	35.55	46.66	40.00	46.32	38.79
PGS (Ours)	**35.00**	**73.33**	**55.00**	**60.00**	43.68	**53.40**
TCP* [27]	16.18	20.00	20.00	10.00	6.99	14.63
TCP-ctrl*	10.29	4.44	10.00	10.00	6.45	8.23
TCP-traj*	8.89	24.29	51.67	40.00	46.28	34.22
TCP-traj w/o distillation	17.14	6.67	40.00	50.00	28.72	28.51
ThinkTwice* [12]	27.38	18.42	35.82	50.00	54.23	37.17
DriveAdapter* [11]	28.82	26.38	48.76	50.00	**56.43**	42.08

4.4 Ablation Study on Bench2Drive

We conduct extensive ablation experiments to assess the contribution of each component in our self-supervised paradigm. For efficient closed-loop evaluation, we select the **Merging** and **Overtaking** scenarios, which are both complex and highly interactive. Together, they account for more than half of the total scenarios, making the evaluation metrics on them sufficiently representative.

Table 3: Ablation Study of the Proposed PGS Framework.

Method	Avg. L2	Ability (%) ↑		
		Merging	Overtaking	**Mean**
VAD-Base	0.91	8.11	24.44	16.28
VAD-Tiny	1.15	9.33	11.11	10.22
PGS_{Base}	0.87	16.46	13.33	14.89
$PGS_{Base+STPS}$	0.78	24.44	26.25	25.35
$PGS_{Base+STPS+MTPS}$	0.75	23.75	44.44	34.10
PGS_{All}	0.77	35.00	73.33	54.17
PGS_{NTPS}	0.90	25.00	6.67	15.84
PGS_{self}	2.89	31.25	35.56	33.40

As shown in Table 3, PGS_{Base} denotes our baseline model, where the perception network is trained with the perception loss used in VAD, and the planning head is trained with the imitation loss. Compared to VAD, it achieves a slightly lower L2 error and a comparable success rate. PGS_{STPS} introduces the centerline-aligned Spatial Trajectory Planning Self-Supervision, which strengthens the alignment between road topology cues and the ego's planned trajectory, leading to significant improvements in both L2 error and success rate. Building upon this, $PGS_{STPS+MTPS}$ incorporates a relevant lane filter and reformulates the multi-modal ego decision as a lane selection task within this filtered set. This design yields a substantial performance boost in the **Overtaking** scenarios, where frequent lane changes occur. Finally, PGS_{All} further adds Negative Trajectory Planning Self-Supervision by identifying risky future positions of surrounding dynamic objects and penalizing ego trajectories that overlap with them. This additional constraint reduces collision risk in both selected scenarios. Overall, PGS_{All} achieves a mean success rate of 54.17%, with an improvement of over 39% compared to PGS_{Base}, which is trained purely via imitation learning. We further isolate the effect of Negative Trajectory Planning Self-Supervision through a dedicated variant, PGS_{NTPS}, to assess model behavior in unstructured environments. Despite the absence of structured road priors, this variant exhibits strong performance in Merging scenarios, demonstrating the capability of NTPS to mitigate collision risks in complex, geometry-agnostic contexts. However, its performance degrades in Overtaking scenarios, likely due to overly conservative behavior in the presence of static obstacles and the lack of contextual lane information. These limitations are effectively addressed

when NTPS is integrated with STPS and MTPS, as structured priors offer richer spatial cues and broaden the maneuver space, enabling more balanced and flexible decision-making.

Besides, we retrained a model, PGS_{self}, using only PGS self-supervision, without any imitation loss. As expected, the L2 error increases significantly due to the absence of expert trajectory knowledge. However, it still achieves a respectable success rate of 33.40%, outperforming both PGS_{Base} and VAD by a large margin. This underscores the importance of perception-consistent ego planning in causal modeling.

5 Conclusion & Limitation

Conclusion: We introduce a perception-guided self-supervision paradigm for end-to-end autonomous driving. By leveraging road topology and dynamic agent motion as both inputs and supervisory signals, our approach aligns ego trajectory prediction with structured, causally relevant cues, enabling stronger causal reasoning and state-of-the-art closed-loop performance. Extensive ablation studies further substantiate the efficacy of our self-supervision mechanisms, highlighting a promising direction for enhancing the real-world robustness of end-to-end autonomous driving.

Limitation: The framework's success relies on accurate and robust perception of dynamic/static agents and road structures. Limited perception accuracy or generalization may impair planning performance, making perception robustness a key challenge for future work.

References

[1] M. Bansal, A. Krizhevsky, and A. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst in robotics: Science and systems xv. *University of Freiburg, Freiburg im Breisgau, Germany*, 2019.

[2] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 6433–6438. IEEE, 2020.

[3] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. *arXiv preprint arXiv:2106.11810*, 2021.

[4] M.-F. Chang, J. W. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays. Argoverse: 3d tracking and forecasting with rich maps. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[5] S. Chen, B. Jiang, H. Gao, B. Liao, Q. Xu, Q. Zhang, C. Huang, W. Liu, and X. Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. *arXiv preprint arXiv:2402.13243*, 2024.

[6] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2022.

[7] H. Gao, S. Chen, B. Jiang, B. Liao, Y. Shi, X. Guo, Y. Pu, H. Yin, X. Li, X. Zhang, et al. Rad: Training an end-to-end driving policy via large-scale 3dgs-based reinforcement learning. *arXiv preprint arXiv:2502.13144*, 2025.

[8] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, et al. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17853–17862, 2023.

[9] J. Huynh. Separating axis theorem for oriented bounding boxes. *URL: jkh. me/files/tutorials/Separating% 20Axis% 20Theorem% 20for% 20Oriented% 20Bounding% 20Boxes. pdf*, 2009.

[10] X. Jia, Z. Yang, Q. Li, Z. Zhang, and J. Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[11] X. Jia, Y. Gao, L. Chen, J. Yan, P. L. Liu, and H. Li. Driveadapter: Breaking the coupling barrier of perception and planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7953–7963, 2023.

[12] X. Jia, P. Wu, L. Chen, J. Xie, C. He, J. Yan, and H. Li. Think twice before driving: Towards scalable decoders for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21983–21994, 2023.

[13] X. Jia, J. You, Z. Zhang, and J. Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. *arXiv preprint arXiv:2503.07656*, 2025.

[14] B. Jiang, S. Chen, Q. Xu, B. Liao, J. Chen, H. Zhou, Q. Zhang, W. Liu, C. Huang, and X. Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8350, 2023.

[15] H. Li, M. Yuan, Y. Zhang, C. Wu, C. Zhao, C. Song, H. Feng, E. Ding, D. Zhang, and J. Wang. Xld: A cross-lane dataset for benchmarking novel driving view synthesis. *CoRR*, 2024.

[16] Y. Li, Y. Wang, Y. Liu, J. He, L. Fan, and Z. Zhang. End-to-end driving with online trajectory evaluation via bev world model. *arXiv preprint arXiv:2504.01941*, 2025.

[17] Z. Li, K. Li, S. Wang, S. Lan, Z. Yu, Y. Ji, Z. Li, Z. Zhu, J. Kautz, Z. Wu, et al. Hydra-mdp: End-to-end multimodal planning with multi-target hydra-distillation. *arXiv preprint arXiv:2406.06978*, 2024.

[18] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[19] Z. Li, Z. Yu, S. Lan, J. Li, J. Kautz, T. Lu, and J. M. Alvarez. Is ego status all you need for open-loop end-to-end autonomous driving? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14864–14873, 2024.

[20] C. Liang and X. Liu. The research of collision detection algorithm based on separating axis theorem. *Int. J. Sci*, 2(10):110–114, 2015.

[21] I. Loshchilov and F. Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

[22] H. Su, W. Wu, and J. Yan. Difsd: Ego-centric fully sparse paradigm with uncertainty denoising and iterative refinement for efficient end-to-end autonomous driving. *arXiv preprint arXiv:2409.09777*, 2024.

[23] W. Sun, X. Lin, Y. Shi, C. Zhang, H. Wu, and S. Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. *arXiv preprint arXiv:2405.19620*, 2024.

[24] T. Wang, C. Zhang, X. Qu, K. Li, W. Liu, and C. Huang. Diffad: A unified diffusion modeling approach for autonomous driving. *arXiv preprint arXiv:2503.12170*, 2025.

[25] C. Wen, J. Qian, J. Lin, J. Teng, D. Jayaraman, and Y. Gao. Fighting fire with fire: Avoiding dnn shortcuts through priming. In *International Conference on Machine Learning*, pages 23723–23750. PMLR, 2022.

[26] X. Weng, B. Ivanovic, Y. Wang, Y. Wang, and M. Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15449–15458, 2024.

[27] P. Wu, X. Jia, L. Chen, J. Yan, H. Li, and Y. Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. *Advances in Neural Information Processing Systems*, 35:6119–6132, 2022.

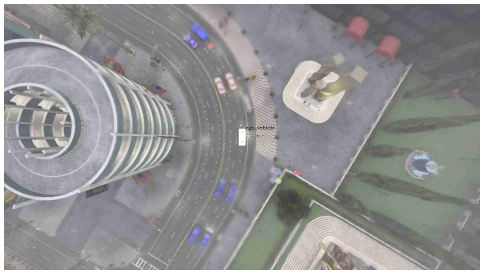
[28] J.-T. Zhai, Z. Feng, J. Du, Y. Mao, J.-J. Liu, Z. Tan, Y. Zhang, X. Ye, and J. Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv preprint arXiv:2305.10430*, 2023.

[29] W. Zheng, R. Song, X. Guo, C. Zhang, and L. Chen. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, pages 87–104. Springer, 2024.

A Qualitative Analysis of Lane Centerline Perception

In CARLA, road topology information is stored using a graph structure, where each node represents an individual lane. The attributes of each node include the global coordinates of the lane's centerline, lane width, and other geometric properties. For each frame of training data, the ground-truth centerlines near the ego vehicle are generated by transforming all lane centerline coordinates from the global map into the ego-centric local coordinate system based on the recorded ego pose.

(a) Perception results in straight road scenarios



(b) Perception results in curved road scenarios

(c) Perception results at intersections

(d) Perception results in obstacle-present scenarios

Figure 4: Visualization of lane centerline perception under diverse road conditions

The qualitative examples presented in Figure 4 demonstrate that, even in the absence of salient visual features for lane centerlines, the map perception module in PGS can still reliably detect them by leveraging visual context from the scene. This ability is particularly evident in complex areas such as intersections. Such robust perception of centerlines forms the foundation for the effectiveness of the self-supervised MTPS and STPS components in our framework.

B Relevant Lane Filtering and Target Lane Determination in MTPS

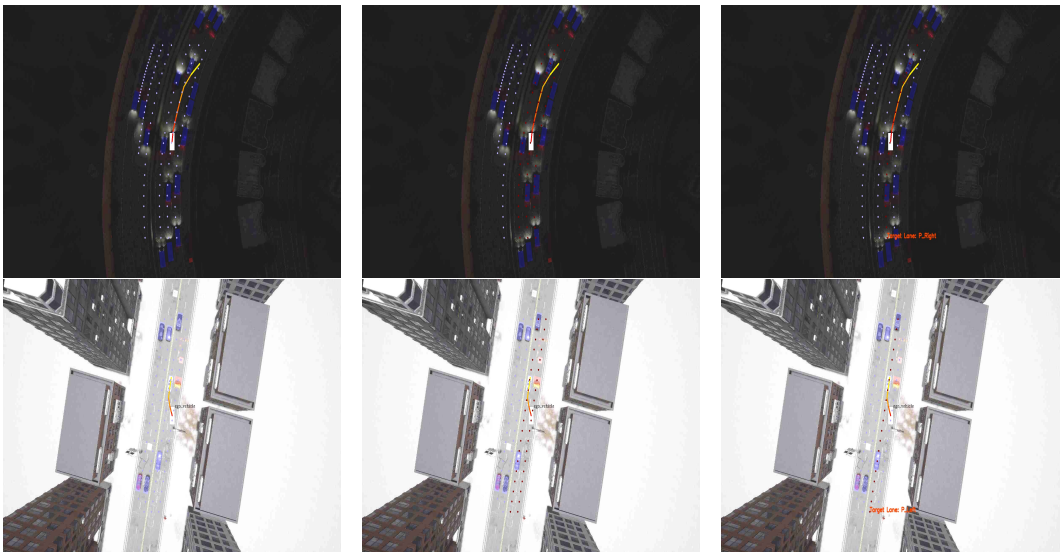
In the Multi-Modal Trajectory Planning Self-Supervision (MTPS), we frame the ego vehicle's multi-modal decision-making as a target lane selection task. The relevant lane filter extracts candidate lanes—namely, the left, current, and right lanes—based on the perceived road topology. Then, by referencing the ground-truth trajectory, the system identifies the target lane corresponding to the intended driving behavior.

Visualization of Supervision Signal Construction for Target Lane Selection

This subsection visually illustrates how the Multi-modal Trajectory Planning (MTP) module leverages the surrounding lane topology to inform the selection of the ego vehicle's planning modality, as described in Section 3.2. Figure 5 provides a step-by-step illustration of the pipeline for constructing the supervision signal for target lane selection, starting from the perception outputs, moving through the filtering of relevant candidate lanes, and culminating in the determination of the target lane by referencing the endpoint of the ground-truth trajectory.

Classification Accuracy of the Target Lane Selection Task

To assess the effectiveness of the topology-guided supervision in MTPS, we evaluate the classification accuracy and recall of target lane prediction on the B2D open-loop validation set. As shown in



(a) Perceived lane centerlines (b) Selected relevant lane set (c) Selected target lane

Figure 5: Visualization of Target Lane Selection. Light pink points indicate the perceived lane centerlines, while dark red points represent the selected relevant lane set. The final target lane is also highlighted in dark red, with its name labeled in orange.

Table 4, training with the full 3-second ground-truth trajectory and standard cross-entropy (CE) loss achieves high accuracy across all classes. However, due to severe class imbalance, the recall for rare categories such as "oriented to left lane" and "oriented to right lane" remains relatively low, at 0.63 and 0.80 respectively.

Table 4: Accuracy and Recall of Target Lane Classification under Different Training Strategies.

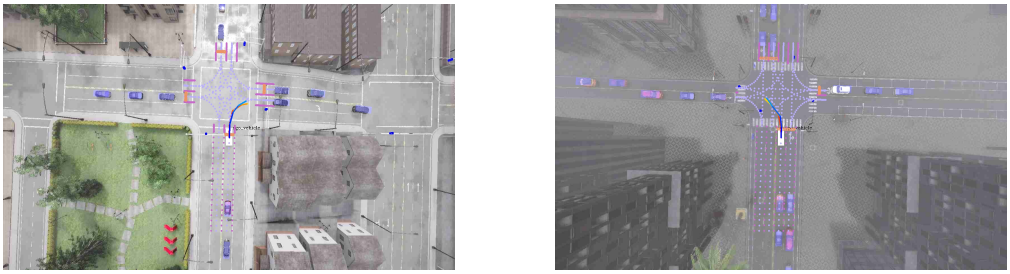
Metric	Label	CE	Weighted CE	Weighted CE + 2s Trajectory
Accuracy	Oriented to current lane	0.9685	0.9554	0.9690
	Oriented to left lane	0.9785	0.9669	0.9748
	Oriented to right lane	0.9887	0.9832	0.9909
Recall	Oriented to current lane	0.9855	0.9586	0.9676
	Oriented to left lane	0.6268	0.7344	0.8788
	Oriented to right lane	0.7978	0.9011	0.9233

To mitigate this imbalance, we adopt a class reweighting strategy based on inverse class frequencies. Specifically, class weights are computed from the training data distribution as $[1.074,\ 32.480,\ 26.505]$, corresponding to current, left, and right lane orientations, respectively. This adjustment significantly improves recall for the left and right lane categories by approximately 10 percentage points.

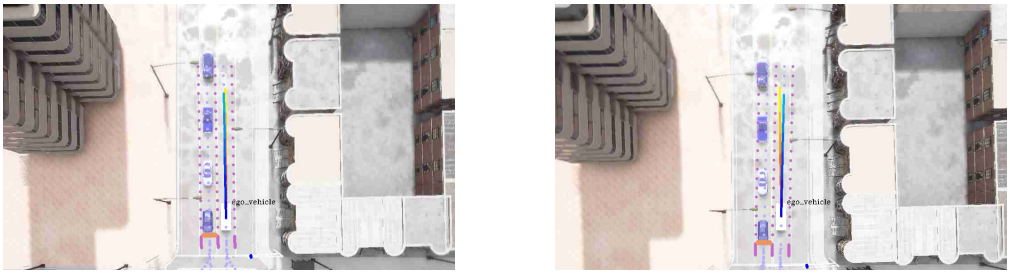
Furthermore, when shortening the trajectory horizon used for matching from 3 seconds to 2 seconds, both recall and precision improve. Shorter trajectories more accurately reflect immediate driving intentions, reducing ambiguity. Based on these results, we adopt inverse frequency weighting and 2-second trajectory matching as the default configuration for PGS training in MTPS.

C Spatial Trajectory Generation in STPS

In STPS, we construct a spatial trajectory by combining the target lane centerline with the ground-truth trajectory, which is then used as supervision for the trajectory regression head to facilitate better causal modeling.



(a) Ground-truth and STP trajectories in turning scenarios at intersections



(b) Ground-truth and STP trajectories in straight-driving scenarios

Figure 6: Comparison of STP-generated and ground-truth trajectories. Perceived lane centerlines are rendered as light pink points, and other perceived lanes in orchid. Ground-truth trajectories are visualized using red-to-yellow gradients, while STP trajectories are represented by blue gradients.

As illustrated in Figure 6, we present two representative scenarios. In Figure 6a, when navigating intersections, ground-truth trajectories occasionally deviate from the centerline of the designated outbound lane—either drifting left or right. Such sporadic deviations, likely due to labeling noise or imperfect execution, introduce spurious signals that hinder the model's ability to capture causally valid motion patterns. In contrast, STP trajectories, aligned with the intended lane centerline, exhibit high directional consistency and semantic alignment, thereby facilitating more robust causal learning.

Similarly, in Figure 6b, ground-truth trajectories in straightforward driving scenarios exhibit oscillations around the lane centerline, potentially introducing ambiguity in lane-following behavior. The STP-generated trajectories, by contrast, maintain a stable, forward-directed course, offering clearer intent supervision and reducing trajectory-level noise.

D Separating Axis Theorem (SAT) Algorithm Description

In NTPS, the SAT[20] algorithm is employed to generate negative supervision signals for the ego vehicle trajectory, conditioned on the predicted trajectories of surrounding agents.

SAT is a classical method for collision detection between convex polygons in 2D space. It is based on the principle that two convex polygons do not intersect if and only if there exists a separating axis—an axis along which the projections of the two polygons do not overlap. In practice, the set of candidate axes is constructed by computing the outward normals of all edges from both polygons. If a separating axis is found, the polygons are guaranteed to be disjoint. Otherwise, the polygons must intersect.

As shown in Algorithm 1, the SAT algorithm iteratively tests all potential separating axes derived from the polygon edges.

Algorithm 1: Separating Axis Theorem (SAT)

Input: Vertex sets of polygons A and B
Output: Whether they intersect (true/false)
for *each polygon* $P \in \{A, B\}$ **do**
  for *each edge* e *of* P **do**
    Compute edge normal as projection axis **axis**;
    Project polygons A and B onto **axis**, obtaining intervals $proj_A$ and $proj_B$;
    if $proj_A$ *and* $proj_B$ *do not overlap* **then**
      return false ; `// Separating axis found - polygons don't intersect`
return true ; `// No separation found on any axis - polygons intersect`

E Visualization of Closed-loop Evaluation

Representative Cases in Diverse and Complex Scenarios

Figure 7 presents the closed-loop evaluation results of our model, PGS_{All}, in the CARLA simulation environment. These visualizations are drawn from the full set of 220 closed-loop test scenarios in the Bench2Drive benchmark, encompassing a diverse spectrum of challenging conditions, such as adverse weather (e.g., heavy rain, dense fog), varied lighting (e.g., daytime, nighttime), and complex traffic scenes (e.g., intersections, lane merging, overtaking, and traffic light negotiation). As illustrated, the model consistently produces smooth, goal-directed trajectories that respect the intended global route while dynamically responding to contextual hazards.

The results indicate that PGS_{All} is capable of dynamically adjusting its trajectory to avoid obstacles while maintaining route fidelity. The model exhibits a strong capacity to align its predictions with the underlying semantic road structure, reflecting a nuanced understanding of the causal relationships between environmental cues and appropriate driving behavior. This capability contributes to reliable and safe autonomous decision-making in closed-loop execution.

Limitations and Failure Case Analysis

While PGS_{All} demonstrates robust performance across a wide range of scenarios, we observe several failure cases that expose current limitations in perception and causal reasoning.

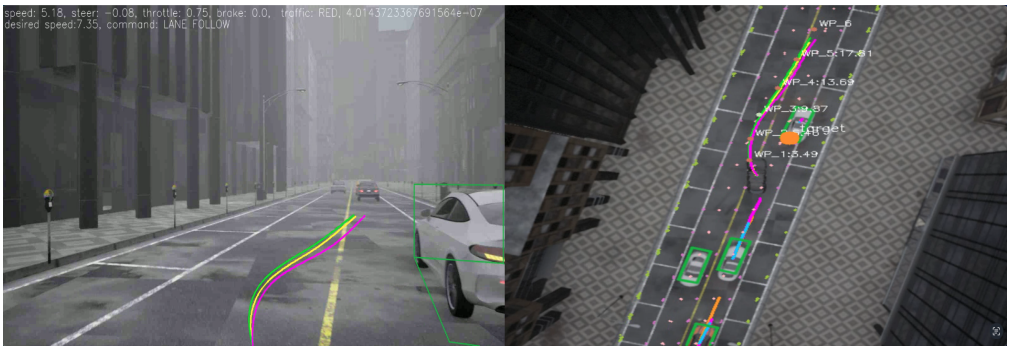
As shown in Figure 8a, the ego vehicle fails to avoid a parked car with an opened door. This failure is attributed to the perception module treating such vehicles the same as regular static obstacles, without distinguishing the opened door as a separate semantic element. Consequently, the model fails to learn the causal relationship between door-opening events and the necessity of avoidance. Similarly, in Figure 8b, the model does not yield to an approaching emergency vehicle from behind, likely due to the absence of semantic differentiation between emergency and regular vehicles in the perception process.

These cases suggest the need to enhance the perception module by introducing specialized object categories or refining bounding box representations (e.g., to cover opened doors or siren-bearing vehicles). Such improvements would allow the model to better capture the causal structures required for socially compliant and context-aware planning in these critical situations.

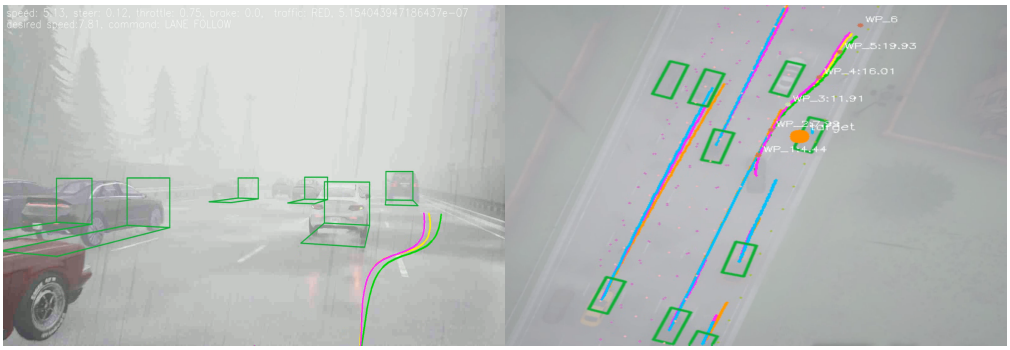
Supplementary Visualizations and Reproducibility

To further support our analysis, we provide an extended set of visualizations in video format, accessible via the following GitHub repository: Supplementary Materials. Within this repository, the folder `0_representative_cases` contains two subfolders, `DiverseChallengingScenarios` and `FailureScenarios`, which showcase representative cases that highlight both the strengths and limitations of the proposed model.

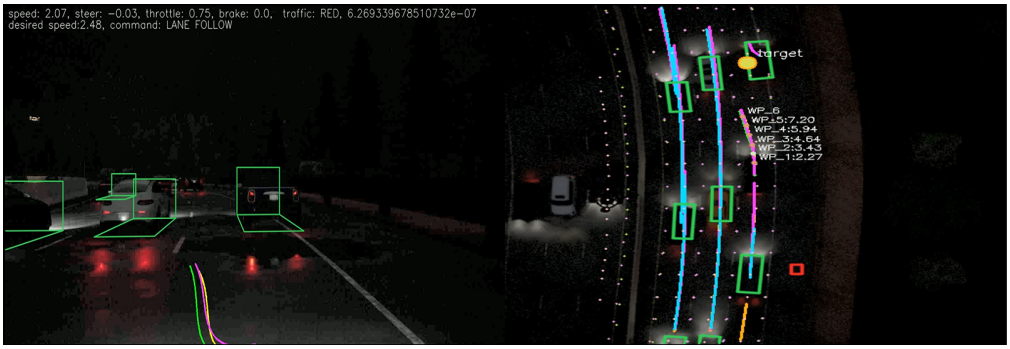
Full closed-loop evaluation results are available in the folder `1_PGS_all_metric_files`, which contains `merged.json` (overall results) and `merged_ability.json` (per-scenario metrics). Frame-level logs for all 220 scenarios, including ego states, control commands (steering, throttle, brake), predicted traffic light states, and high-level decisions, are stored in `eval_PGS_all`.The evaluation



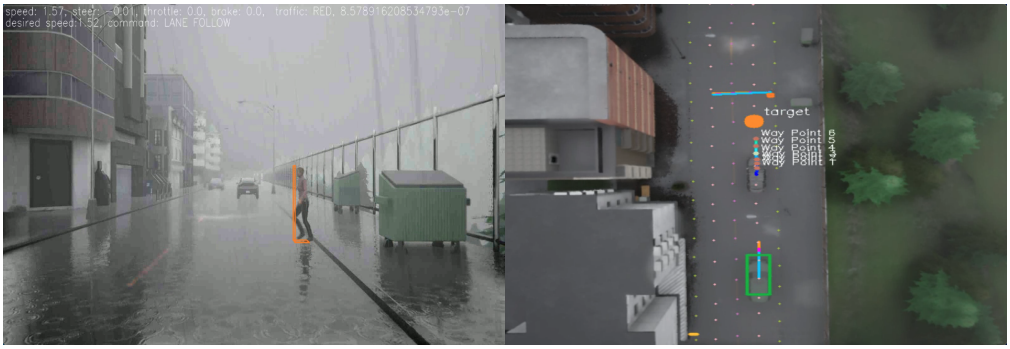
(a) Overtaking a stationary vehicle



(b) Lane change under rainy weather conditions



(c) Driving at night with low visibility



(d) Pedestrian avoidance in straight-road scenarios

(e) Traffic light recognition in the evening

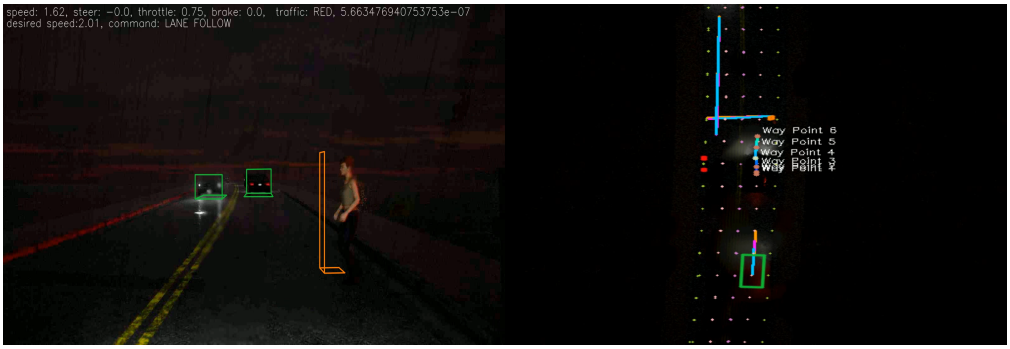
(f) Emergency stop due to malfunctioning vehicle ahead

(g) Traffic light recognition under rainy weather conditions

(h) Vehicle yielding right of way

(i) Merging into highway with yielding

(j) Merging into highway with a parked vehicle ahead

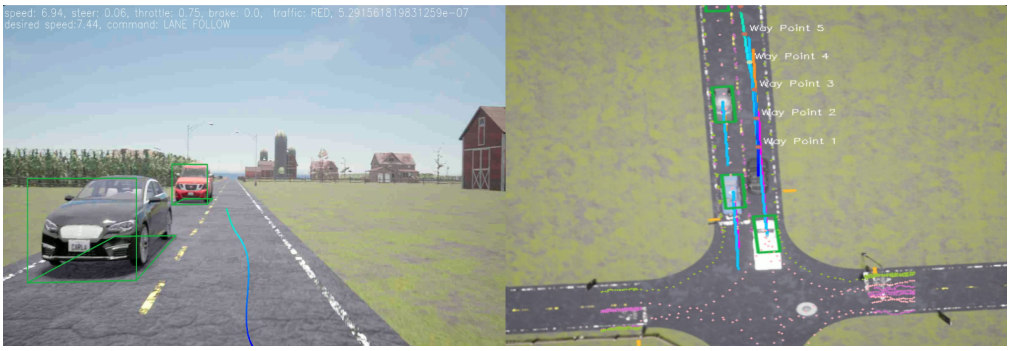


(k) Pedestrian violation crossing the road at night

(l) Obstacle avoidance when other vehicles merge into the lane

Figure 7: Visualization of representative closed-loop scenarios from the Bench2Drive benchmark. The wp or way point denotes predicted trajectory points. The target is the mid-range goal issued by the global route planner in simulation environment, encoded as input to the PGS planning network.

(a) Failure to avoid a vehicle with an opened door.



(b) Failure to yield to an approaching emergency vehicle.

Figure 8: Representative failure cases due to incomplete perception and missing causal cues.

process can be reproduced for each individual town by following the instructions in the Metric section of the Eval Tools.

F PID Controller Configuration

We build upon the original PID controller parameters from Bench2Drive[10] and modify the aim point selection strategy. Instead of using a fixed 4.0-meter target, we adopt a dynamic approach based on vehicle speed: selecting a near aim point of 4.0 meters for low-speed scenarios (below 6.5 m/s) and a far aim point of 10.0 meters for high-speed scenarios (above 6.5 m/s). This design stems from the principle that at higher speeds, the vehicle requires a longer look-ahead distance to follow the planned trajectory accurately. By providing sufficient foresight, the farther aim point facilitates smoother steering adjustments, thereby enhancing trajectory stability and reducing oscillations observed during high-speed maneuvers in our experiments.

G Experiments compute resources

All experiments were conducted with the following specifications:

Hardware Configuration

- **CPU:** Each node is equipped with dual *Intel(R) Xeon(R) Gold 6278C @ 2.60GHz* processors, providing 52 physical cores and 104 threads per node. With 2 nodes in total, the system utilizes 4 CPU sockets and 208 logical processors.
- **Memory:** 512 GB RAM per node
- **GPU:** 16 *NVIDIA V100* GPUs (32 GB each), with 8 GPUs per node across 2 nodes

Training Time

- **Stage 1 Training:** Completed in approximately 1 day using 16 GPUs, equivalent to around 384 GPU hours.
- **Stage 2 Training:** Completed in approximately 1 day using 16 GPUs, equivalent to around 384 GPU hours.

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: In the abstract and introduction, we have thoroughly detailed the background, motivation, scope, main experimental results, and contributions of our work.

   Guidelines:

   - The answer NA 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 NA 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: The limitations of the work are discussed in section 5.

   Guidelines:

   - The answer NA 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: The full set of assumptions and complete proofs are provided in section 3 and section 4 of the main paper.

Guidelines:

- The answer NA 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: The proposed method is implemented with a public closed-loop dataset benchmark. All the implementation details are reported in the main paper.

   Guidelines:

   - The answer NA 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: The datasets used in this paper are publicly available, and the code will be released upon acceptance of the paper.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.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://nips.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, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The experimental details are presented in section 4.

   Guidelines:

   - The answer NA 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: [Yes]

   Justification: We adopt standard evaluation datasets and metrics, which areaccompanied by statistical significance.

   Guidelines:

   - The answer NA 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: Computer resources are provided in both section 4 and the appendix.

   Guidelines:

   - The answer NA 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: The research conducted in the paper strictly conforms the code of ethics.

   Guidelines:

   - The answer NA 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: The broader impacts of this work are discussed in the main paper.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA 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., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA 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: The datasets and benchmarks used are all publicly available with licensees and are properly acknowledged in our paper.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The code will be made public upon acceptance.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA 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: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA 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 rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: No LLMs were involved in the development of the core methods presented in this paper.

Guidelines:

- The answer NA 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 (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.