PHYDRIVE: PLANNING-ORIENTED PHYSICAL EVIDENCE INTERACTION FOR END-TO-END AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Planning in end-to-end autonomous driving requires anticipating the future evolution of scenes, motivating the development of world models for informed decision-making. Existing methods primarily exploit latent representations from world models to improve planning, but often lack explicit physical evidence for reasoning about the physical consequences of candidate motions, limiting their ability to assess safety risks. To address this limitation, we propose **PhyDrive**, an end-to-end autonomous driving framework. First, we design a Physical-Evidence-Guided Scorer (PEGS) that extracts planning-oriented physical evidence and uses it to guide interactions between candidate-specific latent representations and the global scene context, enabling safety-aware trajectory scoring and selection. Second, we introduce a Planning-oriented Latent Representation Learning (PLRL) strategy with a training-only auxiliary network that encourages the world model to encode planning-relevant distinctions, thereby improving its sensitivity to motion-specific scene interactions. On the NAVSIM v1 and v2 benchmarks, **PhyDrive** achieves state-of-the-art performance, with PDMS and EPDMS scores of 95.3 and 93.0, respectively. The code will be publicly available.

1 INTRODUCTION

End-to-end autonomous driving has become an active research area, offering a compact and scalable alternative to conventional modular pipelines (Chitta et al., 2022; Hu et al., 2023). A key determinant of driving performance is the ability to model and understand how the driving scene will evolve. Recent advances in world models provide a natural basis for this capability by learning latent representations of future driving scenes (Zheng et al., 2024; Yang et al., 2025; Li et al., 2025a; Zheng et al., 2025; Zhang et al., 2026b; Liu et al., 2026a). Such future-aware representations are particularly relevant to autonomous driving, where decisions depend not only on the current observation but also on how the scene may evolve. However, how to effectively translate these future representations into information useful for downstream planners remains an open question.

Existing methods address this question by exposing future information to the planner in different ways. Some methods directly predict future images or occupancy representations (Zheng et al., 2024; Yang et al., 2025; Wang et al., 2024b;a; Hong et al., 2026). However, their dense reconstruction wastes computation on appearance details irrelevant to planning, while still *failing* to preserve sparse but critical structures such as road boundaries (Figure 1(a)). Other methods avoid pixel-level prediction and instead predict compact future latent states for downstream planning (Zhang et al., 2026b; Liu et al., 2026a; Hong et al., 2026). Although more compact, summarizing the full future scene into a global latent representation would obscure local distinctions among similar trajectory candidates that are crucial for planning (Figure 1 (b)). Recent trajectory-conditioned approaches alleviate this issue by associating latent scene features with individual trajectories for more motion-specific reasoning (Li et al., 2025b;a; Zhang et al., 2026a; Zheng et al., 2025). Nevertheless, they still model the concrete *physical consequences* of executing each candidate only *implicitly*.

Intuitively, trajectory safety is determined not only by where a reference path lies in the scene but also by how the full ego motion physically *interacts* with its surroundings. A candidate trajectory typically specifies only a sequence of reference poses for the vehicle, whereas its actual conse-

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

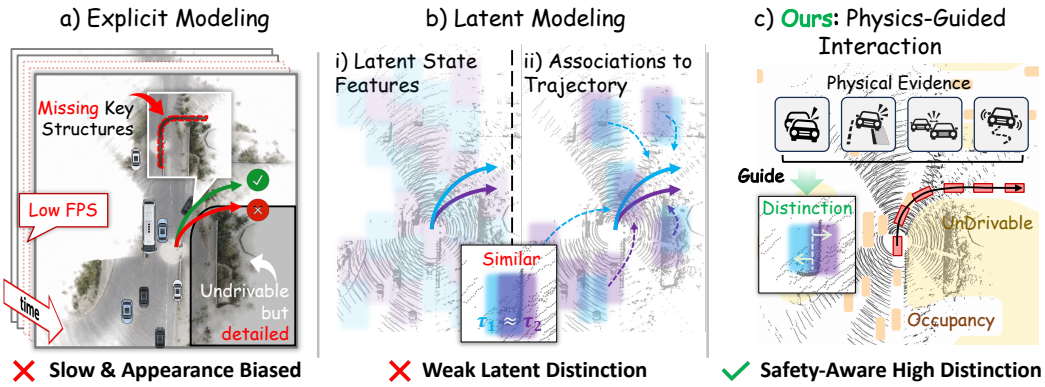

Figure 1: **Comparison of world models for planning.** (a) Explicit future-scene modeling is computationally costly and may allocate capacity to visually detailed but planning-irrelevant regions, and may miss structures critical to planning. (b) Latent modeling relies on latent scene features and their associations with candidate trajectories, but may not sufficiently distinguish similar candidates with different physical consequences. (c) PhyDrive uses planning-oriented physical evidence, such as collision risk and drivable-area constraints, to guide trajectory-scene interactions, enabling safety-aware distinctions among candidate trajectories.

quences additionally depend on the vehicle's dimensions, time-varying occupancy, velocity, and heading. As a result, geometrically similar trajectories can lead to substantially different collision risks when the full vehicle footprint is taken into account, while small changes in speed can induce different time-to-collision values. Therefore, *explicitly* encoding the physical consequences of trajectories in the scene representations is required for safe planning. This necessitates candidate-specific *physical evidence* that accounts for the vehicle's spatiotemporal occupancy and reflects collision risks, drivable-area constraints, and interactions with dynamic agents.

To this end, we introduce **PhyDrive**, a planning framework that places physical evidence at the interface between future-world modeling and trajectory evaluation (Figure 1 (c)). The key idea is to use planning-oriented physical evidence to guide interactions between trajectory-specific features and the scene representation, while encouraging the world model to preserve distinctions that are relevant to planning. Specifically, PhyDrive comprises two main components: (i) a *Physical-Evidence-Guided Scorer* (PEGS), and (ii) a *Planning-oriented Latent Representation Learning* (PLRL) strategy. PEGS extracts explicit physical evidence along each candidate trajectory and uses it to guide interactions between candidate-specific latent features and the global scene context, enabling safety-aware trajectory evaluation and selection. Complementarily, PLRL constructs auxiliary query trajectories covering plausible driving regions and supervises their safety outcomes, encouraging the world model to encode motion-specific distinctions that support downstream planning decisions.

Our contributions are summarized as follows:

- We propose **PhyDrive**, an end-to-end driving framework that introduces planning-oriented physical evidence into world-model-based trajectory evaluation, making the physical consequences of candidate motions explicit to the planner.
- We design two complementary components: PEGS, which uses candidate-specific physical evidence to guide trajectory evaluation, and PLRL, which encourages the world model to learn planning-oriented representations that distinguish motion-specific scene interactions.
- Extensive experiments on the NAVSIM v1 and v2 benchmarks demonstrate state-of-the-art performance, with PDMS and EPDMS scores of 95.3 and 93.0, respectively.

2 Related Work

Trajectory Generation and Scoring. Scoring-based planning generates multiple trajectory candidates using multi-head prediction, generative policies, or factorized vocabularies (Li et al., 2024; Liao et al., 2025; Xing et al., 2025; Sun et al., 2026; Song et al., 2026). Recent work improves

trajectory scoring through interpretable factor scores, scorer generalization, and coarse-to-fine evaluation (Kirby et al., 2026; Li et al., 2025c; Yao et al., 2026a). PhyDrive improves trajectory scoring with factor-specific physical evidence and cross-candidate interaction.

World Models for Trajectory Evaluation. Driving world models incorporate future scene evolution into planning by forecasting explicit visual, occupancy, or flow states (Wang et al., 2024b; Zheng et al., 2024; Yang et al., 2025), or by predicting BEV-space and latent future representations (Li et al., 2025a;b; Zheng et al., 2025; Zhang et al., 2026b; Hong et al., 2026). These future predictions support trajectory generation, refinement, and evaluation. In contrast, PhyDrive extracts candidate-specific physical evidence from future-world representations for trajectory evaluation.

Planning-Oriented Representation Learning. Planning-oriented methods learn scene representations through joint optimization, trajectory-conditioned prediction, or trajectory-guided perception (Hu et al., 2023; Li et al., 2025a; Zhang et al., 2026a). Recent work further aligns world representations with planning through intermediate supervision, cross-modal alignment, hard negatives, or hierarchical objectives (Xia et al., 2026; Yang et al., 2026). PhyDrive instead uses factor-level supervision on auxiliary query trajectories to learn planning-relevant representations.

3 PRELIMINARIES

Future World Modeling Trajectory evaluation requires modeling the interactions between candidate trajectories and their surrounding environments, which relies on reasoning about future scene evolution and motivates effective future-world modeling. To achieve this, the world model W_θ takes the multimodal observation history $\mathcal{H}_t$ as input. Following prior work on temporal aggregation (Zhang et al., 2026b), it aggregates temporally aligned features to preserve both persistent scene content and frame-to-frame variations, and subsequently produces the current BEV representation $\mathbf{F}_t \in \mathbb{R}^{C\times H_{\text{BEV}}\times W_{\text{BEV}}}$ together with a sequence of future BEV representations:

$$\mathcal{W}_t = W_\theta(\mathcal{H}_t) = \left(\mathbf{F}_t, \{\mathbf{F}_{t+h}\}_{h=1}^{H}\right), \quad H = 8, \tag{1}$$

where $\mathbf{F}_{t+h}$ denotes the predicted BEV feature at the h-th future step. Together, these representations capture the current scene and its evolution over a four-second prediction horizon, providing a shared world representation for all planning modules.

Trajectory Decoder We adopt the pretrained encoder $\mathcal{E}_{\phi_0}$ and trajectory decoder $\mathcal{G}_{\phi_0}$ from DrivoR (Kirby et al., 2026) to generate K candidate trajectories for subsequent evaluation:

$$\mathcal{T}_t = \mathcal{G}_{\phi_0}\left(\mathcal{E}_{\phi_0}(\mathbf{I}_t), \mathbf{e}_t\right) = \{\boldsymbol{\tau}_t^k\}_{k=1}^{K}, \qquad K = 64, \tag{2}$$

where $\mathbf{I}_t$ and $\mathbf{e}_t$ denote the multi-view images and the ego state at time t, respectively. ϕ_0 denotes the pretrained parameters, and $\boldsymbol{\tau}_t^k \in \mathbb{R}^{H\times 3}$ denotes the k-th candidate trajectory. Each trajectory consists of $H = 8$ future poses (x, y, θ) sampled at $\Delta t = 0.5$-second intervals. We keep the pretrained camera encoder and trajectory decoder frozen and focus on trajectory selection from $\mathcal{T}_t$.

4 METHODOLOGY

Overall Pipeline Figure 2 illustrates our framework, which consists of two core components: (i) PEGS extracts physical evidence along candidate trajectories and uses it to guide interactions between candidate-specific features and global scene context, enabling safety-aware trajectory selection; (ii) PLRL generates auxiliary query trajectories across plausible driving regions and supervises their safety outcomes, steering the world model toward motion-relevant distinctions.

4.1 PHYSICAL EVIDENCE EXTRACTION

Trajectory evaluation requires reasoning about how candidate motions interact with the scene across multiple physical dimensions. To make these interactions explicit, we extract *physical evidence* along each candidate trajectory $\boldsymbol{\tau}_t^k = \{(x_{k,h}, y_{k,h}, \theta_{k,h})\}_{h=1}^{H}$. As illustrated in Figure 3, this evidence covers four complementary perspectives.

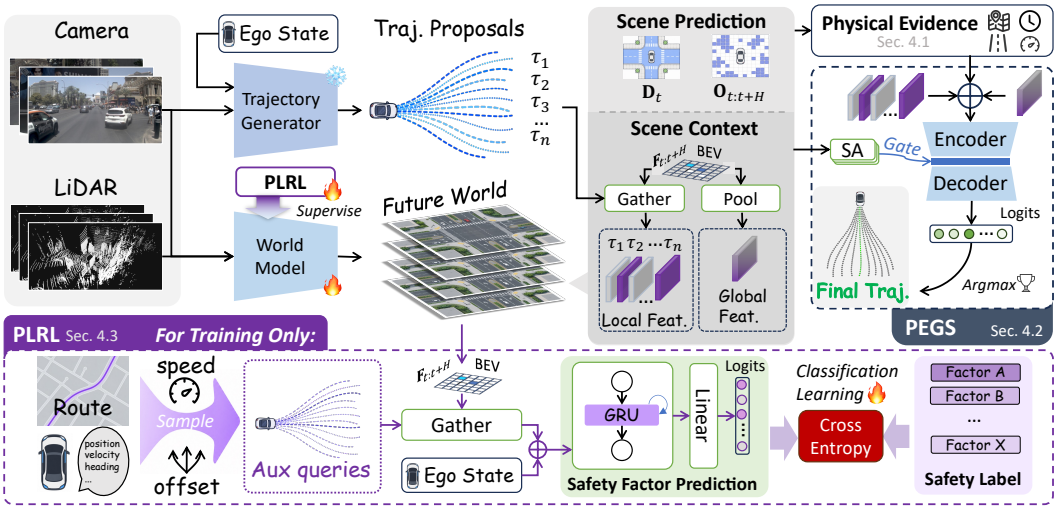

Figure 2: **Overview of PhyDrive.** A frozen trajectory generator produces candidate trajectories, while the world model uses historical camera and LiDAR observations to predict the future world. PEGS combines candidate-specific local and global scene features with physical evidence and uses cross-candidate interactions to predict factor scores and select the final trajectory. The *training-only* PLRL constructs route-conditioned auxiliary query trajectories spanning different speed modes and lateral offsets, and supervises the world representation with safety-factor labels.

Spatial Evidence. To characterize a candidate's interaction with obstacles and navigable space, we decode future occupancy $\mathbf{O}_{t:t+H} = \mathcal{F}_{\text{occ}}(\mathcal{W}_t)$ and a drivable-area map $\mathbf{D}_t = \mathcal{F}_{\text{dri}}(\mathcal{W}_t)$ from the future world representation $\mathcal{W}_t$. We query these maps at the vehicle center and corners along each candidate, using a denser interpolation of the poses to resolve contact over time. Additionally, we repeat this process for multiple safety margins, yielding a progressive spatial contact profile. During training, we additionally supervise $\mathcal{F}_{\text{occ}}$ and $\mathcal{F}_{\text{dri}}$ with ground-truth occupancy and drivable-area labels, respectively.

Temporal Evidence. To anticipate potential conflicts beyond each candidate pose, we apply constant-velocity extrapolation at multiple time horizons to each pose and query the temporally aligned future occupancy at the projected positions. This evidence captures the temporal safety margin between the planned trajectory and future environmental evolution, indicating the imminence of potential conflicts and the time available for safe reaction or re-planning.

Progress Evidence. To characterize each candidate's advancement relative to the scene, we infer a reference direction from the global scene representation, ego state (e_t), and driving command (c_t). The progress evidence is then obtained by projecting each candidate trajectory onto this direction:

$$\mathbf{d}_t = \mathcal{F}_{\text{ref}}(\text{Pool}(\mathcal{W}_t), e_t, c_t)\,, \quad \pi^{\parallel}_{k,h} = (\mathbf{r}_{k,h} - \mathbf{r}^{\text{ego}}_t)^\top \bar{\mathbf{d}}_t, \quad \pi^{\perp}_{k,h} = (\mathbf{r}_{k,h} - \mathbf{r}^{\text{ego}}_t)^\top \bar{\mathbf{d}}^{\perp}_t. \tag{3}$$

Here, $\mathbf{r}_{k,h} = [x_{k,h}, y_{k,h}]^\top$ is the candidate waypoint, and $\bar{\mathbf{d}}_t$ is the normalized reference direction. $\pi^{\parallel}_{k,h}$ and $\pi^{\perp}_{k,h}$ are the longitudinal and lateral projections, respectively. Aggregating these projections along the trajectory characterizes its longitudinal displacement and lateral deviation relative to the learned reference, providing complementary cues about cumulative progress and path adherence that are crucial for balancing safety and task completion.

Kinematic Evidence. To assess whether a candidate trajectory can be executed smoothly, we derive its speed, longitudinal and lateral acceleration, jerk, yaw dynamics, and curvature from the trajectory poses and current ego state. We summarize these measurements across the trajectory and record how much they exceed comfort limits, thereby capturing trajectory smoothness while revealing abrupt braking, sharp turns, and rapid changes in motion.

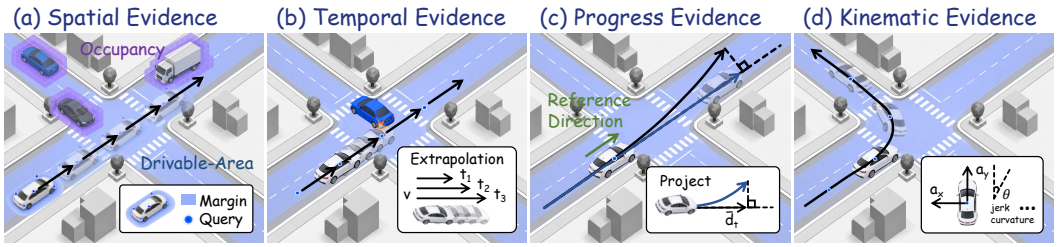

Figure 3: **Physical evidence extraction along candidate trajectories.** (a) Spatial evidence: Queries occupancy and drivable-area predictions around the vehicle footprint under varying safety margins. (b) Temporal evidence: Extrapolates candidate poses across short horizons to capture future occupancy risks. (c) Progress evidence: Projects trajectories onto a learned reference direction to measure longitudinal progress and lateral deviation. (d) Kinematic evidence: Extracts motion characteristics including speed, acceleration, jerk, yaw dynamics, curvature, and comfort violations.

4.2 Physical-Evidence-Guided Scorer

With the extracted physical evidence, PEGS guides interactions between candidate-specific features and the global scene context. Given trajectory proposals $\mathcal{T}_t = \{\boldsymbol{\tau}_t^k\}_{k=1}^K$, PEGS scores each candidate by combining its trajectory representations with planning-oriented physical evidence.

Evidence-driven Encoding. Physical evidence guides the encoding of trajectory representations through factor-specific encoders. For each scoring factor f, we concatenate the factor-specific physical evidence $\mathbf{z}_{k,f}^{\text{phy}}$ (from Sec. 4.1) and scene context $\mathbf{z}_{k,f}^{\text{ctx}}$ with the candidate trajectory embedding $\mathbf{z}_{k,f}^{\text{traj}}$, and then feed them into the corresponding encoder, thereby introducing physical guidance:

$$\mathbf{z}_{k,f}^{\text{enc}} = \text{Encoder}_f([\mathbf{z}_{k,f}^{\text{phy}}, \mathbf{z}_{k,f}^{\text{ctx}}, \mathbf{z}_{k,f}^{\text{traj}}]), \tag{4}$$

where $\mathbf{z}_{k,f}^{\text{enc}}$ is the evidence-aware encoding of candidate k for factor f. $\mathbf{z}_{k,f}^{\text{ctx}}$ is obtained by concatenating scene information such as global context and candidate-specific local features, while $\mathbf{z}_{k,f}^{\text{traj}}$ is obtained by linear encoding of the trajectory. $[\cdot]$ denotes concatenation.

Cross-candidate Interaction. Trajectory selection is inherently comparative, requiring modeling of interactions among candidate trajectories. Using self-attention across candidates, we aggregate the candidate trajectory embedding together with local and global scene features into a unified relational representation. The resulting representation $\mathbf{z}_k^{\text{rel}}$ thus encodes the shared context, integrating both candidate-specific features and comparative information. This representation is shared across all scoring factors, allowing each candidate to retain its own information while incorporating relationships with other candidates.

Factor-specific Decoding. Each factor-specific decoder integrates the encoded evidence-aware feature $\mathbf{z}_{k,f}^{\text{enc}}$ with the shared relational feature $\mathbf{z}_k^{\text{rel}}$ to produce a scoring logit. Firstly, we adaptively balance factor-specific representations and cross-candidate interactions through gating:

$$\boldsymbol{\gamma}_{k,f} = \sigma\big(\text{Linear}([\mathbf{z}_{k,f}^{\text{enc}}; \mathbf{z}_k^{\text{rel}}])\big), \quad \mathbf{z}_{k,f}^{\text{fuse}} = \text{LN}\big(\mathbf{z}_{k,f}^{\text{enc}} + \boldsymbol{\gamma}_{k,f} \odot \mathbf{z}_k^{\text{rel}}\big). \tag{5}$$

where $\boldsymbol{\gamma}_{k,f} \in [0,1]^D$ is the learned gate that modulates the contribution of the shared relational context, and $\mathbf{z}_{k,f}^{\text{fuse}}$ is the fused representation combining factor-specific evidence and gated shared context via addition. The decoder maps each fused representation to a factor-specific logit. A sigmoid then converts the logit into a normalized predicted factor score:

$$\ell_{k,f} = \text{Decoder}_f(\mathbf{z}_{k,f}^{\text{fuse}}), \qquad p_{k,f} = \sigma(\ell_{k,f}). \tag{6}$$

Here, $\ell_{k,f}$ is the raw logit for factor f and candidate trajectory k. The sigmoid function $\sigma(x) = 1/(1+e^{-x})$ maps it to the predicted factor score $p_{k,f} \in (0,1)$. During training, the factor logits are supervised with equally-weighted binary cross-entropy against their corresponding official factor score labels (Kirby et al., 2026). At inference, the predicted factor scores are combined to rank the candidates, as detailed in Appendix Sec. D.

4.3 Planning-oriented Latent Representation Learning

PEGS relies on the world model's latent representations to extract physical evidence and score trajectories. However, the world model is trained on scene reconstruction. Although it excels at scene modeling, it lacks the ability to capture motion-specific distinctions relevant to trajectory evaluation. To enhance the planning relevance of its representations, we introduce planning-oriented latent representation learning (PLRL) strategy that supervises auxiliary query trajectories during training.

Auxiliary Query Trajectories. During training, we generate a set of auxiliary query trajectories $\mathcal{Q}_t^{\text{aux}} = \{\tilde{\boldsymbol{\tau}}_t^j\}_{j=1}^J$ that span plausible driving regions. Unlike the candidate set $\mathcal{T}_t$ used by PEGS, these query trajectories are constructed directly from the route geometry and the ego state. For each speed mode m, we compute future travel distances from the ego state and sample the route centerline to obtain base waypoints $\{(x_h^{\text{route}}, y_h^{\text{route}})^{(m)}\}_{h=1}^H$. For each lateral offset δ and speed mode m, we construct a query trajectory:

$$\tilde{\boldsymbol{\tau}}_t^{(\delta,m)} = \left\{ \left((x_h^{\text{route}}, y_h^{\text{route}})^{(m)} + \delta_h \boldsymbol{n}_h^{(m)},\ \tilde{\theta}_h^{(\delta,m)} \right) \right\}_{h=1}^H . \tag{7}$$

Here, $\boldsymbol{n}_h^{(m)}$ is the lateral unit normal at the sampled route waypoint, δ_h is the lateral displacement at step h, which gradually approaches the target offset δ, and $\tilde{\theta}_h^{(\delta,m)}$ is the heading of the resulting query trajectory, determined by its consecutive waypoints. Combining multiple lateral offsets and speed modes produces a query set that systematically covers the local planning space.

Safety Factor Decoding. For each trajectory $\tilde{\boldsymbol{\tau}}_t^j$, an auxiliary decoder $\mathcal{P}_\eta$ predicts safety factors from the world model's representations. Specifically, the decoder samples local features along the trajectory footprint from $\mathcal{W}_t$, combines them with global scene features and trajectory-specific kinematic features $\mathbf{k}_t^j$, and aggregates them temporally via a GRU. A final linear layer then maps the aggregated features to safety-factor logits ℓ_j^{aux}:

$$\ell_j^{\text{aux}} = \mathcal{P}_\eta\big(\mathcal{W}_t, \tilde{\boldsymbol{\tau}}_t^j, \mathbf{k}_t^j\big), \qquad \ell_j^{\text{aux}} \in \mathbb{R}^{|F|}. \tag{8}$$

We supervise ℓ^{aux} using the binary cross-entropy loss against official safety-factor scores F. This auxiliary task encourages the world model W_θ to encode features that distinguish trajectories by their safety outcomes, thereby enhancing its learned representations. Crucially, the $\mathcal{P}_\eta$ is removed at inference, incurring no additional overhead.

5 Experiments

5.1 Benchmarks

NAVSIM (Dauner et al., 2024). We evaluate PhyDrive on NAVSIM, built on OpenScene (Contributors, 2023), a 2 Hz redistribution of nuPlan (Caesar et al., 2021). NAVSIM provides multi-view camera observations, aggregated LiDAR point clouds, ego states, and route goals, with approximately 103K *navtrain* scenes and 12,146 *navtest* scenes. On *navtest*, an 8-waypoint trajectory over four seconds is evaluated under non-reactive log replay. We report the PDM Score (PDMS, v1) (Dauner et al., 2024) and Extended PDM Score (EPDMS, v2) (Cao et al., 2025). PDMS evaluates collision avoidance (NC), drivable-area compliance (DAC), time-to-collision (TTC), comfort (C), and ego progress (EP), while EPDMS further considers history-aware comfort (HC), driving-direction compliance (DDC), traffic-light compliance (TLC), lane keeping (LK), and extended comfort (EC).

5.2 Implementation Details

We initialize and freeze the DrivoR camera encoder and $K = 64$ trajectory generator from its public checkpoint (Kirby et al., 2026). The future-world model, PEGS, and the PLRL auxiliary decoder are trained on 85,109 *navtrain* scenes. For LiDAR input, three temporal sweeps are aligned to the current ego frame and rasterized into 200×200 pillar-based BEV features at $0.5\,\text{m}$ resolution (Lang et al., 2019). Training takes 10 epochs with a global batch size of 64. AdamW is used with learning

Table 1: **Comparison with state-of-the-art methods on NAVSIM NavTest v1.** Methods are grouped by input modality. † follows the PhyDrive setup, using frozen "Trainval 103K + SimScale 134K" checkpoint for trajectory generation and "Train 85K" for training.

Method	Venue	NC	DAC	TTC	Comfort	EP	PDMS
Camera-only							
UniAD (Hu et al., 2023)	CVPR'23	97.8	91.9	92.9	100.0	78.8	83.4
PWM (Zhao et al., 2026)	NeurIPS'25	98.6	95.9	95.4	100.0	81.8	88.1
AutoVLA (Zhou et al., 2026)	NeurIPS'25	98.4	95.6	98.0	99.9	81.9	89.1
DIVER (Song et al., 2026)	TPAMI'26	98.5	96.5	94.9	100.0	82.6	88.3
DriveVLA-W0 (Li et al., 2026b)	ICLR'26	98.7	99.1	95.3	99.3	83.3	90.2
AutoDrive-P3 (Ye et al., 2026)	ICLR'26	99.1	97.4	96.5	100.0	84.8	90.6
ReCogDrive (Xiong et al., 2026)	ICLR'26	97.9	97.3	94.9	100.0	87.3	90.8
DriveVA (Liu et al., 2026b)	ECCV'26	99.2	97.5	98.7	100.0	83.5	90.9
SGDrive (Li et al., 2026a)	CVPR'26	98.6	97.8	96.2	100.0	85.8	91.1
DriveWorld-VLA (Liu et al., 2026a)	ICML'26	99.1	98.2	96.1	100.0	85.9	91.3
DynVLA (Shang et al., 2026)	ICML'26	98.6	98.7	95.5	100.0	86.8	91.7
DriveFine (Dang et al., 2026)	ECCV'26	98.8	98.6	96.2	100.0	86.9	91.8
SparseDriveV2 (Sun et al., 2026)	ECCV'26	98.5	98.4	95.0	99.9	88.6	92.0
DrivoR† (Kirby et al., 2026)	CVPR'26	98.4	98.5	94.8	100.0	91.5	93.2
DriveSuprim (Yao et al., 2026a)	AAAI'26	98.6	98.6	95.5	100.0	91.3	93.5
RAP-DINO (Feng et al., 2026)	ICLR'26	99.1	98.9	96.7	100.0	90.3	93.8
Camera + LiDAR							
TransFuser (Chitta et al., 2022)	TPAMI'23	97.7	92.8	92.8	100.0	79.2	84.0
DiffusionDrive (Liao et al., 2025)	CVPR'25	98.2	96.2	94.7	100.0	82.2	88.1
WoTE (Li et al., 2025b)	ICCV'25	98.5	96.8	94.9	99.9	81.9	88.3
ResWorld (Zhang et al., 2026b)	ICLR'26	98.9	96.5	95.6	100.0	83.1	89.0
HAD (Yao et al., 2026b)	ECCV'26	98.2	97.3	97.5	100.0	87.4	90.2
GoalFlow (Xing et al., 2025)	CVPR'25	98.4	98.3	94.6	100.0	85.0	90.3
SafeDrive (Kim et al., 2026)	CVPR'26	99.5	99.0	97.2	100.0	84.3	91.6
DiffE2E (Zhao et al., 2025)	NeurIPS'25	99.9	98.6	99.3	99.9	85.3	92.7
PhyDrive	–	99.3	98.8	96.9	100.0	93.7	95.3

rates of 5×10^{-4} and 2×10^{-4} for the future-world model and trajectory scorer, respectively, together with cosine learning-rate decay. Additional architecture, loss, and optimization details are provided in the Appendix Sec. D.

5.3 Main Results

We evaluate PhyDrive on the official NAVSIM NavTest benchmark (Dauner et al., 2024), which evaluates planning performance through non-reactive simulation. Tables 1 and 2 present comparisons with state-of-the-art methods on NavTest v1 and v2, respectively.

NavTest v1. As shown in Table 1, PhyDrive achieves a PDMS of 95.3, establishing a new state of the art on the NavTest v1 benchmark. PhyDrive surpasses the strongest competing method, RAP-DINO (93.8), by 1.5 points and DrivoR† (93.2) by 2.1 points. PhyDrive and DrivoR† share the same frozen pretrained trajectory generator and train their scoring components on the same 85K scenes. PhyDrive improves four of the five individual evaluation metrics over DrivoR† while matching its Comfort score. With the candidate set fixed, the simultaneous gains in safety and progress demonstrate the effectiveness of our method.

NavTest v2. As shown in Table 2, PhyDrive achieves an EPDMS of 93.0, establishing a new state of the art on the NavTest v2 benchmark. PhyDrive surpasses the strongest competing method, DrivoR‡ (90.7), by 2.3 points. The two methods again share the same frozen pretrained trajectory generator and train their scoring components on the same 85K scenes. PhyDrive improves seven of the nine individual evaluation metrics over DrivoR‡ and matches the remaining two. The improvements under these broader evaluation criteria show that PhyDrive can better satisfy safety-related

Table 2: **Comparison with state-of-the-art methods on NAVSIM NavTest v2.** Methods are grouped by input modality. ‡ follows the PhyDrive setup, using frozen "Train 85K + SimScale 65K" checkpoint for trajectory generation and "Train 85K" for training.

Method	Venue	NC	DAC	DDC	TLC	EP	TTC	LK	HC	EC	EPDMS
Camera-only											
VADv2 (Jiang et al., 2026)	ICLR'26	98.0	98.3	99.4	99.8	87.1	96.8	95.2	98.3	88.1	85.8
DriveVLA-W0 (Li et al., 2026b)	ICLR'26	98.5	99.1	98.0	99.7	86.4	98.1	93.2	97.9	58.9	86.1
SGDrive (Li et al., 2026a)	CVPR'26	98.6	94.3	99.5	99.9	86.0	97.9	96.1	98.3	85.9	86.2
DiffRefiner (Yin et al., 2026)	AAAI'26	98.6	98.5	99.7	99.9	87.6	98.1	98.1	98.3	87.9	87.4
MeanFuser (Wang et al., 2026b)	CVPR'26	98.3	97.2	99.6	99.8	87.6	97.4	97.3	98.3	88.2	89.5
DriveFine (Dang et al., 2026)	ECCV'26	98.7	97.3	99.5	99.8	88.7	97.8	97.7	98.4	83.8	89.7
AutoDrive-P3 (Ye et al., 2026)	ICLR'26	99.1	97.4	99.2	99.8	88.0	98.7	96.3	98.3	85.5	89.9
SparseDriveV2 (Sun et al., 2026)	ECCV'26	98.1	98.1	99.6	99.8	91.1	97.3	96.9	98.2	78.4	90.1
DriveSuprim (Yao et al., 2026a)	AAAI'26	98.4	98.6	99.6	99.8	90.5	97.8	97.0	98.3	78.6	90.5
BeyondDrive (Wang et al., 2026a)	ECCV'26	98.5	98.4	99.5	99.8	87.4	97.8	97.3	98.3	87.7	90.5
DrivoR‡ (Kirby et al., 2026)	CVPR'26	98.0	98.3	99.6	99.4	93.0	97.6	95.9	98.3	82.6	90.7
Camera + LiDAR											
TransFuser (Chitta et al., 2022)	TPAMI'23	96.9	89.9	97.8	99.7	87.1	95.4	92.7	98.3	87.2	84.0
DiffusionDrive (Liao et al., 2025)	CVPR'25	98.2	95.9	99.4	99.8	87.5	97.3	96.8	98.3	87.7	84.5
ResAD (Zheng et al., 2026)	CVPR'26	97.8	97.2	99.5	99.8	88.2	96.9	97.0	98.4	88.2	85.5
SafeDrive (Kim et al., 2026)	CVPR'26	99.5	99.0	99.7	99.9	88.6	98.9	97.5	98.2	81.9	87.5
WoTE (Li et al., 2025b)	ICCV'25	98.5	96.8	98.8	99.8	86.1	97.9	95.5	98.3	82.9	87.7
HAD (Yao et al., 2026b)	ECCV'26	98.2	97.3	99.2	99.8	87.4	97.5	95.2	98.3	86.2	88.6
PhyDrive	–	99.2	98.5	99.6	99.6	93.1	98.7	97.4	98.3	89.6	93.0

Table 3: **Core-component ablation on NAVSIM NavTest v1.** *w/o PEGS* uses DrivoR's scorer adapted to our world-model features; *w/o PE* removes all physical evidences while retaining PEGS architecture.

Variant	NC	DAC	TTC	Comfort	EP	PDMS
PhyDrive	99.3	98.8	96.9	100.0	93.7	95.3
w/o PLRL	99.2	98.5	96.5	100.0	93.0	94.6
w/o PEGS	98.9	97.0	96.8	100.0	80.3	88.6
w/o PE	99.1	98.1	95.6	99.9	90.9	93.1
w/o LiDAR	98.6	97.9	95.4	100.0	89.9	92.5

Table 4: **Physical evidence ablation on NAVSIM NavTest v1.** ΔPDMS is relative to PhyDrive.

Configuration	PDMS	ΔPDMS
PhyDrive	95.28	–
NC w/o evidence	95.05	−0.23
DAC w/o evidence	94.89	−0.39
TTC w/o evidence	94.96	−0.32
Comfort w/o evidence	94.90	−0.38
EP w/o evidence	95.09	−0.19

requirements while sustaining driving progress, further supporting its ability to balance safety and efficiency in trajectory selection.

5.4 Ablation Studies

Effectiveness of Core Components. Table 3 evaluates the contribution of each core component, demonstrating complementary contributions from PEGS, PLRL, and LiDAR. Replacing PEGS with DrivoR's scorer adapted to our world-model features causes the largest decline, with PDMS dropping from 95.3 to 88.6 and EP from 93.7 to 80.3. This reflects the contribution of the complete PEGS design. Removing all physical-evidence inputs while retaining the PEGS architecture reduces PDMS from 95.3 to 93.1 and EP from 93.7 to 90.9, demonstrating the contribution of explicit physical evidence to trajectory selection. Without PLRL, PDMS falls to 94.6, suggesting that its auxiliary trajectory supervision helps the world model capture motion-specific scene interactions. Removing LiDAR lowers PDMS to 92.5, indicating that reliable physical evidence extraction relies on accurate geometric information.

Effectiveness of Physical Evidence Table 4 analyzes the role of different types of physical evidence by removing the evidence input associated with each scoring factor. Removing any individual

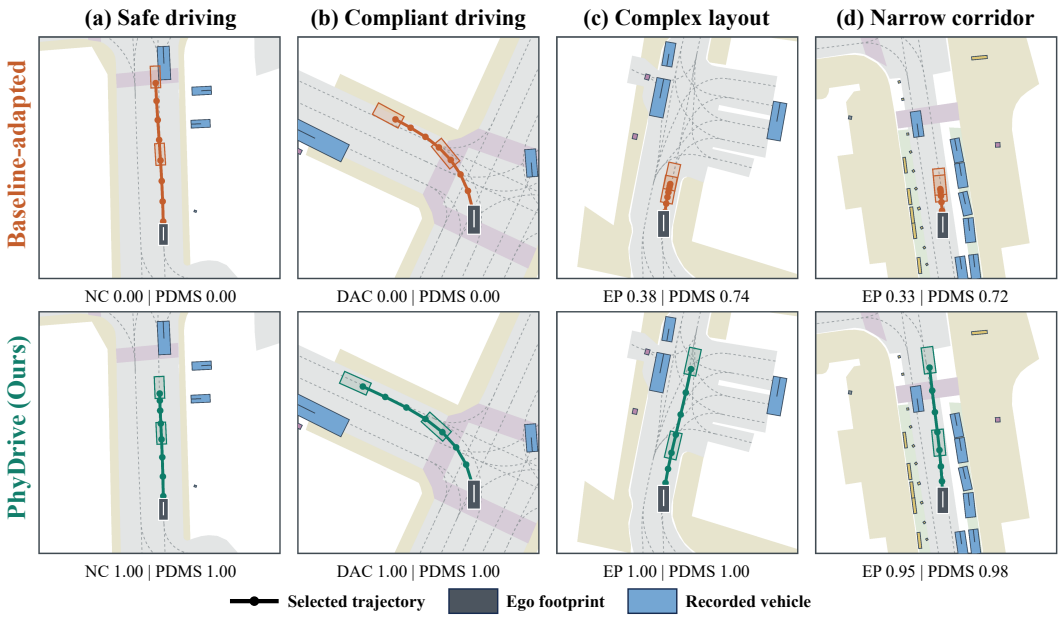

Figure 4: **Visualization of trajectory selection on NAVSIM.** Each column shows one scene evaluated by the Baseline-adapted method (top) and PhyDrive (bottom) using identical K=64 frozen candidates. Highlighted curves denote the selected trajectories, with markers every 0.5s. Colored outlines show ego vehicle footprints at t=2s and 4s. Blue boxes indicate recorded vehicles. Footers report official factor scores and PDMS for each selection.

evidence source decreases PDMS by 0.19–0.39 points relative to the full model, indicating that each evidence pathway contributes to trajectory selection. These evidence sources characterize the physical consequences of candidate motions from complementary perspectives, including collision risk, drivable-area constraints, temporal interactions, motion smoothness, and route consistency. Their combination enables PEGS to evaluate candidate trajectories across multiple physical dimensions, providing a comprehensive basis for trajectory ranking that balances safety and driving efficiency.

5.5 Visualization

We compare the qualitative trajectory-selection results of PhyDrive with the Baseline-adapted baseline (Kirby et al., 2026) in Figure 4. Both methods evaluate the same K=64 frozen trajectory candidates, and the baseline employs a standard trajectory scorer trained jointly with a world model. The trajectories selected by PhyDrive avoid collisions with static obstacles, maintain appropriate driving zones during turns, and exhibit stronger scene understanding in challenging scenarios, such as complex layouts and narrow passages.

6 Conclusion

We presented **PhyDrive**, an end-to-end driving framework that connects future-world modeling with trajectory selection through explicit physical evidence. PEGS uses this evidence to guide interactions between candidate-specific latent features and global scene context, while PLRL encourages the world model to preserve motion-specific distinctions relevant to planning. PhyDrive achieves state-of-the-art performance on NAVSIM v1 and v2. Our work highlights the value of explicit modeling the physical consequences of candidate motions when translating future-world representations into planning decisions that balance safety and driving efficiency.

DISCLOSURE OF LLM USE

We used large language models to assist with identifying and summarizing recent related work, as well as editing the manuscript to improve clarity, conciseness, and organization. The authors reviewed and revised the AI-assisted material to ensure that it accurately reflects the relevant literature, methodology, experimental evidence, and conclusions of this work. We take full responsibility for the final content of the paper, including all claims and AI-assisted material.

REFERENCES

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.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, et al. Pseudo-simulation for autonomous driving. *arXiv preprint arXiv:2506.04218*, 2025.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE transactions on pattern analysis and machine intelligence*, 45(11):12878–12895, 2022.

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

Chenxu Dang, Sining Ang, Yongkang Li, Haochen Tian, Jie Wang, Guang Li, Hangjun Ye, Jie Ma, Long Chen, and Yan Wang. Drivefine: Refining-augmented masked diffusion vla for efficient and robust driving. In *European Conference on Computer Vision*, pp. 412–429. Springer, 2026.

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. *Advances in Neural Information Processing Systems*, 37:28706–28719, 2024.

Lan Feng, Yang Gao, Eloi Zablocki, Quanyi Li, Wuyang Li, Sichao Liu, Matthieu Cord, and Alexandre Alahi. Rap: 3d rasterization augmented end-to-end planning. In *International Conference on Learning Representations*, volume 2026, pp. 37722–37737, 2026.

Yufeng Hong, Xiaotian Zhou, Yingyan Li, Xiangpo Zhou, Lin Liu, Yadan Luo, Shaoqing Xu, Lei Yang, and Ziying Song. Drivefuture: Future-aware latent world models for autonomous driving. *arXiv preprint arXiv:2605.09701*, 2026.

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 *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862. IEEE, 2023.

Bo Jiang, Shaoyu Chen, Hao Gao, Bencheng Liao, Qian Zhang, Wenyu Liu, and Xinggang Wang. Vadv2: End-to-end vectorized autonomous driving via probabilistic planning. In *International Conference on Learning Representations*, volume 2026, pp. 68886–68900, 2026.

Jungho Kim, Jiyong Oh, Seunghoon Yu, Hongjae Shin, Donghyuk Kwak, and Jun Won Choi. Safedrive: Fine-grained safety reasoning for end-to-end driving in a sparse world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24854–24864, June 2026.

Ellington Kirby, Alexandre Boulch, Yihong Xu, Yuan Yin, Gilles Puy, Éloi Zablocki, Andrei Bursuc, Spyros Gidaris, Renaud Marlet, Florent Bartoccioni, et al. Driving on registers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32058–32069, 2026.

Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 12689–12697. IEEE, 2019.

Jingyu Li, Junjie Wu, Dongnan Hu, Xiangkai Huang, Bin Sun, Zhihui Hao, Xianpeng Lang, Xiatian Zhu, and Li Zhang. Sgdrive: Scene-to-goal hierarchical world cognition for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4032–4042, June 2026a.

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. In *International Conference on Learning Representations*, volume 2025, pp. 42942–42959, 2025a.

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27137–27146. IEEE, 2025b.

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. In *International Conference on Learning Representations*, volume 2026, pp. 7890–7911, 2026b.

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.

Zhenxin Li, Wenhao Yao, Zi Wang, Xinglong Sun, Joshua Chen, Nadine Chang, Maying Shen, Zuxuan Wu, Shiyi Lan, and Jose M Alvarez. Generalized trajectory scoring for end-to-end multimodal planning. *arXiv preprint arXiv:2506.06664*, 2025c.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, and Xinggang Wang. Diffusiondrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12037–12047, June 2025.

Lin Liu, Ziying Song, Caiyan Jia, Hangjun Ye, Xiaoshuai Hao, Long Chen, et al. Driveworld-vla: Unified latent-space world modeling with vision-language-action for autonomous driving. *arXiv preprint arXiv:2602.06521*, 2026a.

Mengmeng Liu, Diankun Zhang, Jiuming Liu, Jianfeng Cui, Hongwei Xie, Guang Chen, Hangjun Ye, Michael Ying Yang, Francesco Nex, and Hao Cheng. Driveva: Video action models are zero-shot drivers. In *European Conference on Computer Vision*, pp. 315–335. Springer, 2026b.

Shuyao Shang, Bing Zhan, Yunfei Yan, Yuqi Wang, Yingyan Li, Yasong An, Xiaoman Wang, Jierui Liu, Lu Hou, Lue Fan, et al. Dynvla: Learning world dynamics for action reasoning in autonomous driving. *arXiv preprint arXiv:2603.11041*, 2026.

Ziying Song, Lin Liu, Hongyu Pan, Bencheng Liao, Mingzhe Guo, Lei Yang, Yongchang Zhang, Shaoqing Xu, Caiyan Jia, and Yadan Luo. Diver: Reinforced diffusion breaks imitation bottlenecks in end-to-end autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026.

Wenchao Sun, Xuewu Lin, Keyu Chen, Zixiang Pei, Xiang Li, Yining Shi, and Sifa Zheng. Sparsedrivev2: Scoring is all you need for end-to-end autonomous driving. *arXiv preprint arXiv:2603.29163*, 2026.

Junli Wang, Zhihua Hua, Xueyi Liu, Zebin Xing, Wei Zhang, Kun Ma, Guang Chen, Hangjun Ye, Long Chen, and Qichao Zhang. Beyond imitation: Learning safe end-to-end autonomous driving from hard negatives. In *European Conference on Computer Vision*, pp. 224–241. Springer, 2026a.

Junli Wang, Yinan Zheng, Xueyi Liu, Zebin Xing, Pengfei Li, Kun Ma, Hangjun Ye, Guang Chen, Guang Li, Long Chen, Zhongpu Xia, and Qichao Zhang. Meanfuser: Fast one-step multi-modal trajectory generation and adaptive reconstruction via meanflow for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17884–17893, June 2026b.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. Drivedreamer: Towards real-world-drive world models for autonomous driving. In *European conference on computer vision*, pp. 55–72. Springer, 2024a.

Yuqi Wang, Jiawei He, Lue Fan, Hongxin Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14749–14759. IEEE, 2024b.

Tianze Xia, Lijun Zhou, Kaixin Xiong, Jingfeng Yao, Yu Zhu, Zhenxin Zhu, Bing Wang, Guang Chen, Hangjun Ye, Wenyu Liu, et al. Reworld: Learning better representations for world action models. *arXiv preprint arXiv:2606.27504*, 2026.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1602–1611. IEEE, 2025.

Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, et al. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In *International Conference on Learning Representations*, volume 2026, pp. 157518–157556, 2026.

Pengxuan Yang, Ben Lu, Zhongpu Xia, Chao Han, Yinfeng Gao, Teng Zhang, Kun Zhan, XianPeng Lang, Yupeng Zheng, and Qichao Zhang. Worldrft: Latent world model planning with reinforcement fine-tuning for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11649–11657, 2026.

Yu Yang, Jianbiao Mei, Yukai Ma, Siliang Du, Wenqing Chen, Yijie Qian, Yuxiang Feng, and Yong Liu. Driving in the occupancy world: Vision-centric 4d occupancy forecasting and planning via world models for autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 9327–9335, 2025.

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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 11910–11918, 2026a.

Wenhao Yao, Xinglong Sun, Zhenxin Li, Shiyi Lan, Zi Wang, Jose M Alvarez, and Zuxuan Wu. Had: Combining hierarchical diffusion with metric-decoupled rl for end-to-end driving. *arXiv preprint arXiv:2604.03581*, 2026b.

Yuqi Ye, Zijian Zhang, Junhong Lin, Shangkun Sun, Changhao Peng, and Wei Gao. $autodrive\text{-}p^3$: Unified chain of perception-prediction-planning thought via reinforcement fine-tuning, 2026. URL https://arxiv.org/abs/2603.28116.

Liuhan Yin, Runkun Ju, Guodong Guo, and Erkang Cheng. Diffrefiner: Coarse to fine trajectory planning via diffusion refinement with semantic interaction for end to end autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 12009–12017, 2026.

Bozhou Zhang, Jingyu Li, Nan Song, and Li Zhang. Perception in plan: Coupled perception and planning for end-to-end autonomous driving. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 12376–12384, 2026a.

Jinqing Zhang, Zehua Fu, Zelin Xu, Wenying Dai, Qingjie Liu, and Yunhong Wang. Resworld: Temporal residual world model for end-to-end autonomous driving. *arXiv preprint arXiv:2602.10884*, 2026b.

Rui Zhao, Yuze Fan, Ziguo Chen, Fei Gao, and Zhenhai Gao. Diffe2e: Rethinking end-to-end driving with a hybrid action diffusion and supervised policy. *arXiv preprint arXiv:2505.19516*, 2025.

Zhida Zhao, Talas Fu, Yifan Wang, Lijun Wang, and Huchuan Lu. From forecasting to planning: Policy world model for collaborative state-action prediction. *Advances in Neural Information Processing Systems*, 38:134585–134611, 2026.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. Occworld: Learning a 3d occupancy world model for autonomous driving. In *European conference on computer vision*, pp. 55–72. Springer, 2024.

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 *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28632–28642. IEEE, 2025.

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. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3729–3739, June 2026.

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. *Advances in Neural Information Processing Systems*, 38: 27920–27956, 2026.

A Motivation for Freezing the DrivoR Proposal Generator

Recent scorer-based planners generate multiple candidate trajectories and select one for execution. Their performance depends on both candidate coverage and selection accuracy. To examine these factors, we evaluate the 64 trajectories generated by publicly available DrivoR (Kirby et al., 2026) checkpoints.

As shown in Table 5, the candidate sets achieve oracle scores of 99.0665–99.4096 PDMS on NAVSIM v1 and 97.8488–98.2068 EPDMS on NAVSIM v2. However, the scores of the trajectories selected by DrivoR remain substantially below these oracle values. For these high-coverage candidate sets, trajectory selection is therefore a key remaining bottleneck. We freeze the pretrained DrivoR weights and focus on selecting better trajectories from its unchanged candidate sets.

Table 5: Oracle and DrivoR selection scores for the same K64 proposals. The oracle selects candidates using evaluation labels.

DrivoR weights	**Oracle**	**Selected**
NAVSIM v1 NavTest (PDMS)		
Train 85K	99.1	93.1
Trainval 103K	99.3	93.7
103K + SimScale 65K	99.3	94.0
103K + SimScale 134K	99.4	94.6
NAVSIM v2 NavTest (EPDMS)		
Train 85K	97.8	88.8
85K + SimScale 65K	98.2	91.3
85K + SimScale 134K	98.1	91.0

Distinction from the matched baselines in the main text. The "Selected" scores in Table 5 are obtained directly from the officially released DrivoR checkpoints, including their original trajectory scorers. In contrast, the DrivoR baselines in Tables 1 and 2 follow the PhyDrive experimental setup: they use the same frozen pretrained trajectory generators as PhyDrive, while their scorers are retrained on the same 85,109 *navtrain* scenes. Thus, the scores reported here (94.6 PDMS and 91.3 EPDMS for the corresponding checkpoints) and those in the main text (93.2 PDMS and 90.7 EPDMS) reflect different scorer training settings, rather than different candidate trajectory sets.

B NavHard Evaluation Protocol and Sensor Availability

NavHard (Cao et al., 2025) evaluates planning in long-tail and safety-critical driving scenarios. Unlike the standard NavTest evaluation, it uses a two-stage protocol: the first stage is based on real observations, while the second uses pseudo-simulated observations. Performance is measured with EPDMS, which extends the original planning score with criteria such as driving-direction compliance, traffic-light compliance, lane keeping, and history-aware comfort. This protocol provides a more demanding assessment of planning behavior in challenging scenarios.

The pseudo-simulated second stage provides camera observations but no LiDAR point clouds. The PhyDrive configuration evaluated in this paper requires both modalities to construct its future scene representation. Removing LiDAR for the second stage would leave the physical evidence without proper geometric support, thereby degrading its accuracy. We therefore do not report the NavHard result for this configuration. Instead, we report PDMS on NavTest v1 and EPDMS on NavTest v2, where both camera and LiDAR inputs are available.

C Detailed Physical Evidence Extraction

We detail the construction of the physical evidence $\mathbf{z}_{k,f}^{\mathrm{phy}}$ used by PEGS, following the spatial, temporal, progress, and kinematic categories in the main text. We retain $\boldsymbol{\tau}_t^k$, $\mathbf{r}_{k,h}$, $\theta_{k,h}$, $H = 8$, and $\Delta t = 0.5\,\mathrm{s}$ for the candidate trajectories. All coordinates are expressed in the current ego frame, where $\mathbf{r}_t^{\mathrm{ego}} = \mathbf{0}$. The occupancy sequence $\mathbf{O}_{t+1:t+H}$ and drivable-area map $\mathbf{D}_t$ denote

the corresponding sigmoid probabilities. The implementation predicts H future occupancy slices $\{\mathbf{O}_{t+h}\}_{h=1}^{H}$ and one current drivable-area map. The extracted quantities are inputs to the learned factor encoders.

C.1 Footprint Construction and Map Sampling

The candidate poses refer to the rear axle. Let $L_{\mathrm{veh}} = 5.18\,\mathrm{m}$, $W_{\mathrm{veh}} = 2.30\,\mathrm{m}$, and $b_{\mathrm{ctr}} = 1.46\,\mathrm{m}$ be the vehicle length, width, and rear-axle-to-center distance, respectively. For a footprint margin μ, the five local query offsets comprise the center and four corners:

$$\begin{aligned} \mathcal{B}^{(\mu)} &= \{(b_{\mathrm{ctr}}, 0)^\top\} \cup \left\{ \left(b_{\mathrm{ctr}} \pm \left(\frac{L_{\mathrm{veh}}}{2} + \mu \right),\ \pm \left(\frac{W_{\mathrm{veh}}}{2} + \mu \right) \right)^\top \right\}, \\ \mathbf{x}_{k,h,q}^{(\mu)} &= \mathbf{r}_{k,h} + \mathbf{R}(\theta_{k,h}) \mathbf{b}_q^{(\mu)}, \qquad \mathbf{b}_q^{(\mu)} \in \mathcal{B}^{(\mu)}, \quad q = 1, \ldots, 5, \end{aligned} \tag{9}$$

where the two corner signs vary independently and $\mathbf{R}(\theta)$ is the planar rotation matrix. We use $\mu \in \{0, 0.5, 1.0\}\,\mathrm{m}$ for the multi-margin occupancy queries. The center offset stays fixed as the rectangle expands.

For dense contact and temporal queries, we prepend $(\mathbf{r}_{k,0}, \theta_{k,0}) = (\mathbf{0}, 0)$ and interpolate the resulting $H+1$ poses to $N_{\mathrm{d}} = 33$ samples. We denote these by $(\widehat{\mathbf{r}}_{k,n}, \widehat{\theta}_{k,n})$, $n = 0, \ldots, N_{\mathrm{d}} - 1$, with $\Delta t_{\mathrm{d}} = H\Delta t/(N_{\mathrm{d}} - 1) = 0.125\,\mathrm{s}$. Positions are interpolated linearly; headings are recovered with $\mathrm{atan2}$ after interpolating their sine and cosine values. The same footprint transform gives $\widehat{\mathbf{x}}_{k,n,q}^{(\mu)}$ at these dense poses.

We denote bilinear map sampling by $\mathcal{I}(P, \mathbf{x})$. The longitudinal BEV range is $[-20, 80]\,\mathrm{m}$ and the lateral range is $[-50, 50]\,\mathrm{m}$. Samples outside the grid use zero padding. The occupancy slices correspond to physical times $h\Delta t$, $h = 1, \ldots, H$. For a query time s, we set $\bar{s} = \mathrm{clip}(s, \Delta t, H\Delta t)$ and define $a = \bar{s}/\Delta t - 1$, $j = \lfloor a \rfloor$, and $\lambda = a - j$. We interpolate occupancy probabilities as

$$\mathcal{O}_t(s) = (1 - \lambda)\mathbf{O}_{t+j+1} + \lambda \mathbf{O}_{t+\min(j+2, H)}. \tag{10}$$

At dense trajectory time $s_n = n\Delta t_{\mathrm{d}}$, we use $\widehat{\mathbf{O}}_n = \mathcal{O}_t(s_n)$. Queries before Δt use the first predicted slice, and queries after $H\Delta t$ use the last predicted slice. Within the predicted interval, occupancy and footprint queries therefore share the same physical timestamp.

C.2 Spatial Evidence

For NC, we sample occupancy at every dense footprint point and retain the drivable probabilities at the nominal footprint ($\mu = 0$):

$$\begin{aligned} o_{k,n,q}^{(\mu)} &= \mathcal{I}(\widehat{\mathbf{O}}_n, \widehat{\mathbf{x}}_{k,n,q}^{(\mu)}), \quad \rho_{k,n}^{(\mu)} = \max_q o_{k,n,q}^{(\mu)}, \\ \widehat{d}_{k,n,q} &= \mathcal{I}(\mathbf{D}_t, \widehat{\mathbf{x}}_{k,n,q}^{(0)}), \quad \chi_{k,n} = \max_q (1 - \widehat{d}_{k,n,q}). \end{aligned} \tag{11}$$

The curves $\rho_{k,n}^{(\mu)}$ characterize contact risk at different clearances, while $\chi_{k,n}$ describes the nominal footprint's off-road risk.

We summarize a risk curve $\mathbf{u} = (u_0, \ldots, u_{N_{\mathrm{d}}-1})$ with an operator $\mathcal{S}_{\mathrm{risk}}$ containing six statistics: its maximum, mean, final value, largest positive stepwise increase, risk-weighted normalized time, and maximum over the first quarter of samples. The time statistic is

$$\frac{\sum_n w_n\, n/(N_{\mathrm{d}} - 1)}{\max(\sum_n w_n, 10^{-5})}, \qquad w_n = \sigma(12(u_n - 0.5))\,. \tag{12}$$

It describes the temporal distribution of elevated risk, rather than the first collision time. Concatenating all occupancy samples, nominal drivable samples, and risk summaries gives

$$\begin{aligned} \mathbf{z}_{k,\mathrm{NC}}^{\mathrm{phy}} &= \left[\mathrm{vec}(o_k); \mathrm{vec}(\widehat{d}_k); \{\mathcal{S}_{\mathrm{risk}}(\rho_k^{(\mu)})\}_\mu; \mathcal{S}_{\mathrm{risk}}(\chi_k) \right] \in \mathbb{R}^{684}, \\ \mathbf{z}_{k,\mathrm{DAC}}^{\mathrm{phy}} &= \mathrm{vec}\Big(\{\mathcal{I}(\mathbf{D}_t, \mathbf{x}_{k,h,q}^{(0)})\}_{h,q} \Big) \in \mathbb{R}^{40}. \end{aligned} \tag{13}$$

Here, vec flattens the indicated samples and brackets denote concatenation. The DAC encoder uses the five nominal footprint points at the original $H = 8$ poses. These sampled probabilities provide geometric evidence for the learned NC and DAC predictions; they do not implement the official polygon-level collision and responsibility rules.

C.3 Temporal Evidence

At each dense pose, we estimate speed by position differences divided by Δt_{d}, using the measured ego speed at $n = 0$. For each lookahead duration $\Delta_{\mathrm{f}} \in \{0, 0.3, 0.6, 0.9\}$ s, we translate the nominal footprint along the current heading:

$$\widehat{\mathbf{x}}^{\mathrm{look}}_{k,n,q}(\Delta_{\mathrm{f}}) = \widehat{\mathbf{x}}^{(0)}_{k,n,q} + \widehat{v}_{k,n}\Delta_{\mathrm{f}} \begin{bmatrix} \cos\widehat{\theta}_{k,n} \\ \sin\widehat{\theta}_{k,n} \end{bmatrix}. \tag{14}$$

The occupancy query is evaluated at the dense pose time plus the lookahead:

$$o^{\mathrm{look}}_{k,n,q}(\Delta_{\mathrm{f}}) = \mathcal{I}\big(\mathcal{O}_t(s_n + \Delta_{\mathrm{f}}), \widehat{\mathbf{x}}^{\mathrm{look}}_{k,n,q}(\Delta_{\mathrm{f}})\big). \tag{15}$$

Temporal interpolation uses the continuous query time, including lookaheads that are not integer multiples of Δt_{d}, with the boundary handling defined above. For each lookahead, we take the maximum over footprint points to obtain a risk curve and retain its nonnegative increase relative to zero lookahead. The TTC evidence concatenates all lookahead occupancy samples, these risk increases, normalized speed $\widehat{v}_{k,n}/20$, the indicator $\mathbb{I}[\widehat{v}_{k,n} < 0.1\,\mathrm{m/s}]$, and $\mathcal{S}_{\mathrm{risk}}$ for each lookahead curve. This produces $\mathbf{z}^{\mathrm{phy}}_{k,\mathrm{TTC}} \in \mathbb{R}^{882}$, which describes prospective contact risk without explicitly estimating a collision time in seconds.

C.4 Progress Evidence

We use the reference direction $\mathbf{d}_t$, its normalized form $\bar{\mathbf{d}}_t$, and the projections $\pi^{\parallel}_{k,h}$ and $\pi^{\perp}_{k,h}$ defined in the main text. The reference predictor receives the pooled scene representation, the ego state $\mathbf{e}_t$, and the driving command c_t, which is included in the ego-state input. It does not receive an official route centerline. In addition to the two projections, we compute the stepwise travel distance and cumulative path length:

$$d^{\mathrm{step}}_{k,h} = \|\mathbf{r}_{k,h} - \mathbf{r}_{k,h-1}\|_2, \qquad d^{\mathrm{path}}_{k,h} = \sum_{a=1}^{h} d^{\mathrm{step}}_{k,a}. \tag{16}$$

The per-pose progress descriptor is

$$\mathbf{g}^{\mathrm{prog}}_{k,h} = \left[\frac{x_{k,h}}{50}, \frac{y_{k,h}}{50}, \sin\theta_{k,h}, \cos\theta_{k,h}, \frac{d^{\mathrm{path}}_{k,h}}{40}, \frac{d^{\mathrm{step}}_{k,h}}{10}, \frac{\pi^{\parallel}_{k,h}}{40}, \frac{\pi^{\perp}_{k,h}}{20} \right], \tag{17}$$

where distances are measured in meters. Concatenating the H descriptors gives 64 values, which a learned projection maps to the 256-dimensional $\mathbf{z}^{\mathrm{phy}}_{k,\mathrm{EP}}$. Path length and directional displacement provide distinct cues, and the lateral projection measures deviation from the learned reference axis.

C.5 Kinematic Evidence

Kinematic evidence uses the original H poses. We compute velocity, acceleration, and jerk by finite differences:

$$\mathbf{v}_{k,h} = \frac{\mathbf{r}_{k,h} - \mathbf{r}_{k,h-1}}{\Delta t}, \quad \mathbf{a}_{k,h} = \frac{\mathbf{v}_{k,h} - \mathbf{v}_{k,h-1}}{\Delta t}, \quad \mathbf{j}_{k,h} = \frac{\mathbf{a}_{k,h} - \mathbf{a}_{k,h-1}}{\Delta t}. \tag{18}$$

The initial velocity and acceleration come from $\mathbf{e}_t$. We rotate acceleration into the candidate's heading frame to obtain $a^{\parallel}_{k,h}$ and $a^{\perp}_{k,h}$, and similarly obtain longitudinal jerk $j^{\parallel}_{k,h}$. With the wrapped heading increment $\Delta\theta_{k,h} = \mathrm{atan2}(\sin(\theta_{k,h} - \theta_{k,h-1}), \cos(\theta_{k,h} - \theta_{k,h-1}))$, yaw rate and curvature are

$$\omega_{k,h} = \frac{\Delta\theta_{k,h}}{\Delta t}, \qquad \kappa_{k,h} = \frac{\Delta\theta_{k,h}}{\max(d^{\mathrm{step}}_{k,h}, 0.05\,\mathrm{m})}. \tag{19}$$

Their first differences divided by Δt give $\dot{\omega}_{k,h}$ and $\dot{\kappa}_{k,h}$, using $\omega_{k,0} = \kappa_{k,0} = 0$.

We retain ten normalized channels:

$$\mathbf{u}^{\mathrm{kin}}_{k,h} = \Big[\frac{\|\mathbf{v}_{k,h}\|_2}{20}, \frac{a^{\parallel}_{k,h}}{4.05}, \frac{a^{\perp}_{k,h}}{4.89}, \frac{\|\mathbf{a}_{k,h}\|_2}{4.89}, \frac{\|\mathbf{j}_{k,h}\|_2}{8.37}, \frac{j^{\parallel}_{k,h}}{4.13}, \frac{\omega_{k,h}}{0.95}, \frac{\dot{\omega}_{k,h}}{1.93}, \frac{\kappa_{k,h}}{0.2}, \frac{\dot{\kappa}_{k,h}}{0.2}\Big], \tag{20}$$

with each normalization constant expressed in the corresponding SI units. For each channel, we retain its mean absolute value, maximum absolute value, and signed final value. We additionally summarize violations of the six comfort conditions: longitudinal acceleration, lateral acceleration, jerk magnitude, longitudinal jerk, yaw acceleration, and yaw rate. For each condition, we subtract one from the magnitude-to-limit ratio, clamp at zero, and retain its temporal mean and maximum. Longitudinal acceleration uses separate limits of $2.40\,\mathrm{m/s^2}$ and $4.05\,\mathrm{m/s^2}$ for acceleration and braking; the remaining limits match the corresponding normalizers above. The complete evidence contains 8×10 sequence values, 30 channel summaries, and 12 violation summaries, giving $\mathbf{z}^{\mathrm{phy}}_{k,\mathrm{C}} \in \mathbb{R}^{122}$. These features use comfort-related limits to expose motion characteristics to the learned encoder; the final comfort factor is predicted by PEGS.

D Implementation Details

Data and input representation. We train the future-world model, PEGS, and the PLRL auxiliary decoder on 85,109 navtrain scenes, while keeping the pretrained DrivoR camera encoder and trajectory generator frozen. For NavTest v1, the camera encoder and $K = 64$ trajectory generator are initialized from the pretrained Trainval 103k + SimScale 134k checkpoint, while for NavTest v2, they are initialized from the pretrained Train 85k + SimScale 65k checkpoint. In both settings, these modules are kept frozen during training (Kirby et al., 2026). Each sample uses three camera–LiDAR frames. LiDAR sweeps are aligned to the current ego frame before rasterization into ten-channel pillar features on a 200×200 BEV grid. The grid covers $[-20, 80]\,\mathrm{m}$ longitudinally and $[-50, 50]\,\mathrm{m}$ laterally at $0.5\,\mathrm{m}$ resolution. The future-world model predicts eight future states at $0.5\,\mathrm{s}$ intervals.

PEGS and PLRL settings. PEGS scores the frozen $K = 64$ trajectories with eight-head attention across candidates. For PLRL, the training-only query bank contains 28 trajectories, formed by seven lateral offsets $\{-3, -2, -1, 0, 1, 2, 3\}\,\mathrm{m}$ and four speed modes. The lateral displacement at step h is $\delta_h = \delta(3u_h^2 - 2u_h^3)$, where $u_h = h/H$ and $H = 8$. Given the current speed $v \in [0, 20]\,\mathrm{m/s}$, the target speeds for stop, yield, keep, and accelerate are 0, $0.5v$, v, and $\min(v{+}3, 18)\,\mathrm{m/s}$, respectively. Each mode uses constant acceleration $a = \mathrm{clip}((v_{\mathrm{target}} - v)/4, -4, 2)\,\mathrm{m/s^2}$ and route progress $s(t) = vt + \frac{1}{2}at^2$ at eight half-second steps. Because deceleration is capped, the stop mode need not reach zero speed within four seconds. The auxiliary decoder uses a 128-dimensional hidden state and a GRU over these eight poses. For each auxiliary query trajectory, the supervision vector F contains the NC, DAC, and TTC scores, and the decoder outputs three logits. TTC targets with the invalid sentinel value 2.0 are masked in the auxiliary binary cross-entropy loss.

PEGS supervision. Let $\ell_{b,k,f}$ denote the predicted logit for factor f of candidate k in scene b, and let $y_{b,k,f}$ be the corresponding factor target computed by the official NAVSIM evaluator. PEGS uses equally weighted binary cross-entropy over candidates and factors:

$$\mathcal{L}^{(b)}_{\mathrm{PEGS}} = \frac{1}{K|\mathcal{F}|} \sum_{k=1}^{K} \sum_{f \in \mathcal{F}} \mathrm{BCEWithLogits}\big(\ell_{b,k,f}, \tilde{y}_{b,k,f}\big). \tag{21}$$

For NavTest v1, $\mathcal{F}$ contains NC, DAC, EP, TTC, and Comfort. Before computing the loss, TTC targets with the sentinel value 2.0 are mapped to 1.0, and all targets are clipped to $[0, 1]$. For NavTest v2, $\mathcal{F}$ contains NC, DAC, DDC, TLC, EP, TTC, LK, and HC. Its targets are clipped to $[0, 1]$, retaining valid partial factor scores. The loss is averaged over factors and candidates within each scene. In contrast to the v1 PEGS loss, the PLRL auxiliary loss masks TTC targets with the sentinel value 2.0, as described above.

Candidate ranking. Let $p_{k,f} = \sigma(\ell_{k,f})$ be the predicted factor score, where $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function. For NavTest v1, we clip the scores to $\bar{p}_{k,f} = \mathrm{clip}(p_{k,f}, 10^{-6}, 1-10^{-6})$ for numerical stability. The ranking score is

$$R_k^{\mathrm{v1}} = \log \bar{p}_{k,\mathrm{NC}} + \log \bar{p}_{k,\mathrm{DAC}} + \log(5\bar{p}_{k,\mathrm{TTC}} + 5\bar{p}_{k,\mathrm{EP}} + 2\bar{p}_{k,\mathrm{Comfort}}) \,. \tag{22}$$

For NavTest v2, let $\mathcal{M} = \{\mathrm{NC}, \mathrm{DAC}, \mathrm{DDC}, \mathrm{TLC}\}$ be the multiplicative factors. The ranking score is

$$\begin{aligned} R_{t,k}^{\mathrm{v2}} = & \sum_{f \in \mathcal{M}} \log p_{t,k,f} \\ & + \log(5p_{t,k,\mathrm{EP}} + 5p_{t,k,\mathrm{TTC}} + 2p_{t,k,\mathrm{LK}} + 2p_{t,k,\mathrm{HC}}) \,. \end{aligned} \tag{23}$$

These logarithms preserve the ordering of the corresponding positive product scores. NavTest v1 selects the candidate with the highest R_k^{v1}; the NavTest v2 selection maximizes $R_{t,k}^{\mathrm{v2}}$.

Optimization. We train the future-world model, PEGS, and the PLRL decoder jointly for 10 epochs on four GPUs. The per-GPU batch size is 4, with four gradient-accumulation steps, giving an effective batch size of 64. We use AdamW with learning rates of 5×10^{-4} for the world model and PLRL decoder and 2×10^{-4} for PEGS, weight decay 10^{-4}, and gradient-norm clipping at 1.0. The learning rate has a 5% linear warm-up followed by cosine decay to 5% of its initial value. The coefficients of the world-reconstruction, PLRL, and PEGS losses are 0.25, 0.25, and 1.0, respectively. Forward computation uses BF16, while parameters and optimizer states remain in FP32. The random seed is 2.

E Ablation Studies

We evaluate four groups of ablations: PLRL query design, the PLRL auxiliary decoder, PEGS architecture, and physical-evidence sampling. All variants retain the frozen DrivoR generator and the same $K = 64$ execution candidates. Tables report NAVSIM NavTest v1 PDMS on a 0–100 scale. Differences relative to full PhyDrive are computed before rounding.

E.1 PLRL Query Design and Representation Learning

Query coverage. The full PLRL bank combines seven lateral offsets from -3 to $3\,\mathrm{m}$ at $1\,\mathrm{m}$ intervals with four speed modes (stop, yield, keep, and accelerate), yielding 28 training-only route-conditioned queries. To evaluate the query coverage used for PLRL supervision, Table 6 changes one query dimension at a time. For lateral coverage, we use five offsets over $\pm 2\,\mathrm{m}$, eleven over $\pm 5\,\mathrm{m}$, or the centerline alone, retaining all four modes. For speed-mode coverage, we retain the seven-offset bank and either remove the yield mode or use only the keep mode.

Gradient coupling. To examine how the auxiliary objective shapes the world representation, the gradient-flow variant retains the full 28-query bank and auxiliary decoder but detaches both the future-world features and their global pooled features at the auxiliary decoder input, preventing the PLRL loss from updating the world model.

Table 6: PLRL query-design ablations on NavTest v1. The full bank combines seven lateral offsets and four speed modes. J is the number of training-only queries; ΔPDMS is relative to PhyDrive.

Group	Query design	J	PDMS	ΔPDMS
Full	Seven offsets × four modes	28	**95.28**	–
Lateral coverage	$\pm 2\,\mathrm{m}$ (five offsets)	20	95.01	−0.27
	$\pm 5\,\mathrm{m}$ (eleven offsets)	44	95.14	−0.14
	Centerline only	4	95.07	−0.21
Speed modes	Without yield mode	21	95.19	−0.09
	Keep mode only	7	94.98	−0.30
Gradient flow	Stop gradient to world features	28	94.73	−0.55

E.2 PLRL Auxiliary Decoder

The training-only auxiliary decoder samples local world features along each query footprint, combines them with global scene and kinematic features, and aggregates the sequence with a GRU. To examine temporal aggregation and global scene context, Table 7 compares the full decoder with two variants: replacing the GRU by temporal mean pooling and removing the pooled global scene feature. The auxiliary factor supervision is kept the same.

Table 7: PLRL auxiliary-decoder ablations on NavTest v1. "No GRU" uses temporal mean pooling; the auxiliary factor supervision is unchanged.

Decoder	PDMS	ΔPDMS
Full PhyDrive	**95.28**	–
No temporal GRU	95.19	−0.09
No global scene context	95.07	−0.21

E.3 PEGS Architecture and Context

DrivoR scorer adaptation for ablation. For the *w/o PEGS* in Table 3, we use a trajectory scorer with the same architectural template as the DrivoR baseline (Kirby et al., 2026): trajectory embeddings are processed by a Transformer decoder with self-attention across candidates and cross-attention to scene tokens, followed by factor-specific MLP heads. The adaptation maintains three key constraints to isolate the scoring mechanism: (i) candidate trajectories are held fixed, both configurations consume identical frozen K64 proposals from the DrivoR generator; (ii) both configurations use the same world-model architecture and BEV feature interface; (iii) training data and supervision are matched: both scorers are trained on the same 85,109 navtrain scenes with the same factor-level supervision. The 103K Trainval + 134K SimScale data were used to pretrain the frozen DrivoR trajectory generator for the NavTest v1 experiments; they are not additional training data for either scorer. The baseline scorer uses BEV-derived scene tokens through the shared feature interface, but does not use PEGS's metric-aligned physical evidence extraction or evidence-guided encoding. With identical trajectory proposals and matched world-model architecture, training data, and supervision, the 88.6 versus 95.3 PDMS comparison demonstrates the benefit of the complete PEGS scoring design over the DrivoR-style scorer.

PEGS component ablations. To evaluate evidence-driven encoding, cross-candidate interaction, and factor-specific decoding in PEGS, we vary one component at a time (Table 8). The decoding-input variants use only the factor-specific evidence-aware encoding or only the shared relational feature. The interaction variant bypasses cross-candidate self-attention. For the gated decoders, we replace adaptive fusion with a fixed gate of $\gamma = 0.5$. The encoding variants share one encoder across factors or omit the candidate trajectory embedding from the factor encoder inputs while retaining trajectory geometry. The context variant excludes local scene features.

Table 8: PEGS architecture ablations on NavTest v1. "No candidate attention" skips cross-candidate self-attention.

Group	Variant	PDMS	ΔPDMS
Reference	Full PhyDrive	**95.28**	–
Decoding input	Evidence-aware encoding only	94.86	−0.42
	Shared relational feature only	93.04	−2.24
Interaction	No candidate attention	94.85	−0.43
Fusion	Fixed gate $\gamma = 0.5$	95.15	−0.13
Encoding	Shared factor encoder	94.89	−0.39
	No candidate embedding in encoders	95.01	−0.27
Context	No local scene context	94.10	−1.18

E.4 PHYSICAL-EVIDENCE SAMPLING

The full physical-evidence extractor samples the vehicle center and four corners, footprint margins $\{0, 0.5, 1.0\}$ m, and TTC lookaheads $\{0, 0.3, 0.6, 0.9\}$ s. Table 9 changes these choices separately: removing the outer 1.0 m margin, omitting the 0.6 s TTC lookahead, or sampling only the center or only the four corners.

Table 9: Physical-evidence sampling ablations on NavTest v1. Each row changes only the indicated spatial or temporal sampling choice.

Group	Sampling choice	PDMS	ΔPDMS
Reference	Five points; three margins; four lookaheads	**95.28**	–
Margins	$\{0, 0.5\}$ m	94.94	-0.34
TTC lookahead	$\{0, 0.3, 0.9\}$ s	95.04	-0.24
Footprint points	Center only	94.85	-0.43
	Four corners only	95.07	-0.21

E.5 SAFETY-FACTOR RANKING ANALYSIS

Within each scene, we consider unordered pairs of the 64 candidate trajectories. We measure trajectory distance by the root mean square (RMS) of their planar position differences over eight future steps. A pair is retained if its RMS distance is greater than 10^{-4} m and at most 1 m, and its endpoint distance is at most 2 m. For factor f, a retained pair is valid when its two official factor scores differ by more than 10^{-9}; official-score ties are excluded. TTC comparisons are also excluded if either official TTC score equals the sentinel value 2.0. Across all evaluated scenes, this yields 168,376 NC, 316,870 DAC, and 378,982 TTC valid pair-factor comparisons. Factor-wise accuracy is the proportion of valid comparisons in which the predicted factor logits rank the candidate with the higher official factor score first; an exact predicted-logit tie receives half credit. Combined accuracy pools all 864,228 valid pair-factor comparisons across the three factors. All variants use the same frozen $K = 64$ trajectory proposals.

Table 10: **Safety-factor ranking on similar trajectory pairs.** Pairs have similar trajectory geometry but different oracle outcomes for the evaluated factor. Higher is better.

Model	PDMS	NC	DAC	TTC	Combined
PhyDrive	**95.28**	**94.43**	**95.17**	**93.14**	**94.13**
w/o PE	93.13	93.13	90.75	90.69	91.19
w/o PLRL	94.62	94.42	93.37	92.78	93.31

PhyDrive improves combined ranking accuracy by 2.94 percentage points over w/o PE and 0.82 percentage points over w/o PLRL. This result supports the role of physical evidence and PLRL in distinguishing candidates with different safety outcomes.

E.6 MULTI-SEED ROBUSTNESS OF CORE ABLATIONS

To evaluate the robustness of the core-component ablations, we repeat all experiments using three random seeds. All configurations use the same frozen trajectory generator, training data, and optimization settings. As shown in Table 11, the relative ordering of the variants remains consistent across all seeds. PhyDrive achieves an average PDMS of 95.22, outperforming the variants without PLRL, PEGS, and physical evidence by 0.50, 6.24, and 1.94 points, respectively. These results demonstrate that the contributions of the proposed components are consistent across different random seeds.

Table 11: Multi-seed core-component ablation on NAVSIM NavTest v1. We report PDMS for each random seed and the mean with sample standard deviation over three runs. All variants use the same frozen trajectory proposals.

Configuration	Seed 2	Seed 3	Seed 5	Mean $\pm$ Std.
PhyDrive	**95.28**	**95.18**	**95.21**	$\mathbf{95.22 \pm 0.05}$
w/o PLRL	94.62	94.70	94.84	94.72 ± 0.11
w/o PEGS	88.61	88.98	89.35	88.98 ± 0.37
w/o PE	93.13	93.18	93.54	93.28 ± 0.22

F Inference Efficiency

We profile inference on a single NVIDIA RTX 3090 with batch size 1 and FP32 precision. Measurements use eight NavTest v1 scenes, eight warm-up iterations, and 32 timed iterations with GPU synchronization. Timing includes camera encoding, LiDAR processing, trajectory generation, future-world modeling, scoring, and trajectory selection, but excludes disk I/O. Peak memory is the allocated GPU memory reported by PyTorch. The w/o PE variant skips unused physical-evidence computation, and the training-only PLRL decoder is excluded from inference.

Table 12: **Inference efficiency on NAVSIM NavTest v1.** Latency is reported as median / 95th percentile.

Model	PDMS	Latency (ms)	Peak memory (GB)
PhyDrive	**95.28**	1301.49 / 1318.67	1.171
w/o PE	93.13	1282.56 / 1293.30	1.167
w/o PEGS (Transformer)	88.61	1311.65 / 1334.04	1.167

Compared with w/o PE, PhyDrive gains 2.15 PDMS points with an additional 18.93 ms of median inference latency in this setup.

G Visualization

Figures 5 and 6 provide additional qualitative examples of trajectory selection in challenging driving scenes.

Figure 5: **Additional visualization of trajectory selection on NAVSIM.**

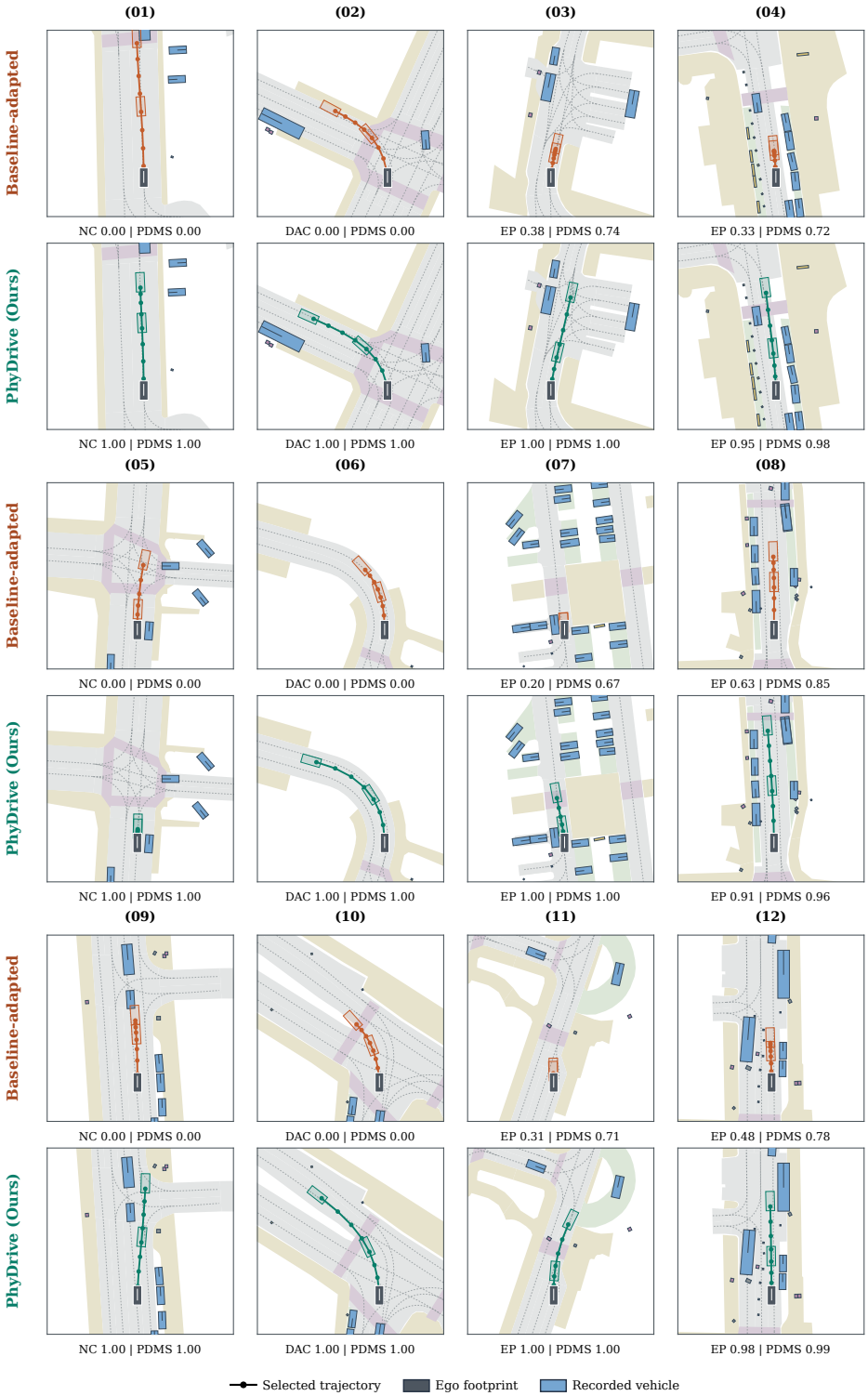

Figure 6: **Additional visualization of trajectory selection on NAVSIM.**

