DRIVE WHAT MATTERS: LEARNING AND FORECASTING DECISION RELEVANCE FOR END-TO-END AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent end-to-end driving methods combine current scene understanding with future scene prediction to support planning. However, predicting how the scene evolves does not explicitly reveal which elements are decision-critical. An actor that is irrelevant to the current decision may become critical under a particular future maneuver. It is therefore essential to model both an actor's current decision relevance and its future evolution. To this end, we introduce DriveMatter, a framework that learns current decision relevance and forecasts future evolution for two-stage trajectory refinement. To supervise this learning, we introduce complementary geometric and semantic supervision that captures route-conditioned spatial relevance and interaction-driven importance for planning. The learned current relevance first refines a base route and target speed into a candidate plan. Conditioned on this candidate and observation history, a Plan-Conditioned Relevance World Model (PCR-WM) forecasts future relevance to guide a second refinement of route and speed. Together, these stages allow the planner to respond to what matters now and anticipate what will matter as the candidate maneuver unfolds. On the Bench2Drive closed-loop benchmark, DriveMatter consistently improves success rate across distinct expert-data settings. Benefiting from joint modeling of current decision relevance and its future evolution, it outperforms the corresponding baselines by 10.91 and 8.1 percentage points with PDM-Lite demonstrations and LEAD expert data, respectively.

1 INTRODUCTION

End-to-end autonomous driving learns a policy that maps current and historical sensor observations to future ego motion. Recent systems jointly optimize perception, prediction, and planning, reducing reliance on hand-designed interfaces and allowing scene representations to adapt to the driving objective (Hu et al., 2023; 2022; Jiang et al., 2023; Ye et al., 2026; Li et al., 2025c). Nevertheless, effective planning requires sufficient contextual information to support appropriate ego decisions over time (Lin et al., 2025).

Modern planners can capture such rich scene context through latent representations and feature interaction mechanisms (Doll et al., 2024; Weng et al., 2024; Sun et al., 2025; Liao et al., 2025). However, the influence of individual scene elements on the ego maneuver is typically embedded implicitly within these representations. Internal responses may indicate which scene information is being utilized, but they do not directly specify whether or how a particular participant should constrain the ego motion, trigger deceleration, or alter the planned path (Ishihara et al., 2021). As illustrated in Figure 1(a), rich scene representations alone do not directly reveal which factors actually drive a change in the intended maneuver. Without an explicit scene-to-decision relationship, such influence remains difficult to directly inspect, supervise, and further refine during training.

More importantly, decision relevance is maneuver-dependent and evolves over time. The same neighboring vehicle may be irrelevant when the ego keeps its lane but become critical when the ego intends to merge, while an obstacle may strongly affect the current plan and quickly become irrelevant once passed. Existing world models predict *what will happen* by forecasting future latent, BEV, or visual states (Li et al., 2025a; Zheng et al., 2025; Li et al., 2025b; Xiong et al., 2026; Jia et al.,

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.

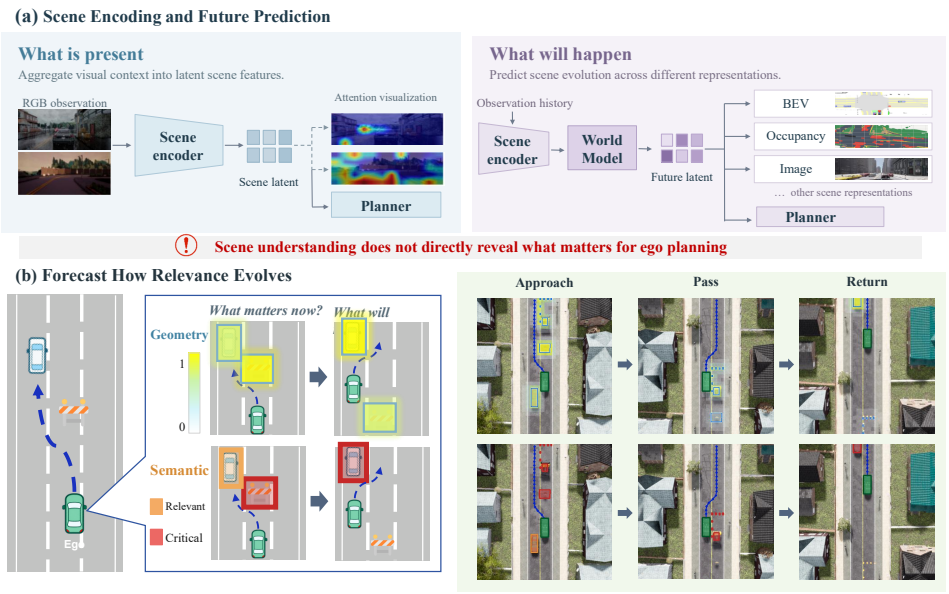


Figure 1: **Learning what matters now and forecasting what will matter.** (a) Scene encoding and future prediction capture scene content without explicitly identifying its influence on ego planning. (b) DriveMatter uses current and predicted future decision relevance to guide route and speed refinement, enabling the ego vehicle to pass the obstruction and return to its lane.

2026), but they do not explicitly identify *which future changes should alter the ego decision*. As illustrated in Figure 1(a), future scene prediction alone leaves the planner to infer decision relevance from a high-dimensional forecast. As a result, scene-centric world modeling may devote substantial representational and predictive capacity to factors that have little influence on the current maneuver, while decision-critical interactions remain under-emphasized. This can delay necessary braking or trajectory adjustment, or conversely induce unnecessary conservative responses. Planning therefore requires not only predicting future scene evolution, but also modeling how decision relevance changes as the candidate maneuver unfolds.

Motivated by this gap, we introduce **DriveMatter**, a framework that learns an explicit, planning-aligned representation of decision relevance from complementary geometric and semantic supervision. Rather than leaving scene-to-decision influence entangled within latent features, DriveMatter explicitly models *what matters now* and how it evolves into *what will matter* as the candidate maneuver unfolds (Figure 1(b)). Geometric cues provide route- and ego-conditioned spatial evidence of which participants may affect the planned motion, while semantic supervision captures higher-level interaction relevance and criticality. A navigation-conditioned Current Relevance Encoder reconciles these signals into a decision-relevance representation and uses it to refine the base route and target speed into an initial candidate plan. Building on this plan, our **Plan-Conditioned Relevance World Model (PCR-WM)** forecasts how relevance evolves under the intended maneuver from the observation history and candidate motion. The predicted future relevance is then queried along the candidate plan to guide a second refinement of speed and trajectory, allowing planning to adapt to interactions that become more or less decision-critical over time.

On the Bench2Drive closed-loop benchmark, DriveMatter consistently improves performance across two distinct base-model and expert-data settings. With PDM-Lite demonstrations, DriveMatter achieves 89.24 DS and 78.18% SR, while training on LEAD expert data yields 96.9 DS and 90.2% SR, demonstrating its effectiveness and transferability across different model and data configurations. Our contributions are threefold: (1) We propose **DriveMatter**, a decision-relevance-aware end-to-end driving framework that jointly models current relevance and its future evolution through two-stage refinement, yielding consistent closed-loop gains across distinct expert-data settings. (2) Complementary geometric and semantic supervision is designed to characterize current relevance, while a Current Relevance Encoder learns a planning-aligned representation for first-stage trajectory refinement. (3) A Plan-Conditioned Relevance World Model (PCR-WM) is introduced to forecast plan-dependent relevance evolution and guide future-aware second-stage trajectory refinement.

2 PRELIMINARIES

2.1 PROBLEM FORMULATION

The autonomous driving task aims to generate safe and feasible ego motion from onboard observations while following a navigation command. At time t, an end-to-end driving policy receives camera images $\mathbf{I}_t$, a LiDAR observation $\mathbf{L}_t$, ego state $\mathbf{e}_t$, and navigation input $\mathbf{n}_t$. Here $\mathbf{e}_t$ describes ego motion, including measured speed, and $\mathbf{n}_t$ specifies an ego-frame target point and a discrete driving command. We group these inputs as the sensor observation $\mathbf{o}_t$ and driving condition $\mathbf{c}_t$:

$$\mathbf{o}_t = (\mathbf{I}_t, \mathbf{L}_t), \qquad \mathbf{c}_t = (\mathbf{e}_t, \mathbf{n}_t). \tag{1}$$

The policy uses observations and driving conditions available up to the current time to predict a spatial route and target-speed logits:

$$(\hat{\boldsymbol{\xi}}_t, \hat{\mathbf{s}}_t) = \mathcal{P}(\mathbf{o}_{\le t}, \mathbf{c}_{\le t}), \qquad \hat{\boldsymbol{\xi}}_t = \{\hat{\mathbf{p}}_{t,i}\}_{i=1}^{N}. \tag{2}$$

Here $\mathcal{P}$ denotes the driving policy, and $\le t$ includes the current and available past inputs. Hats denote predictions. The route $\hat{\boldsymbol{\xi}}_t$ contains N points, where $\hat{\mathbf{p}}_{t,i} \in \mathbb{R}^2$ is the ith point in the current ego frame. These points specify path geometry, not fixed arrival times. The vector $\hat{\mathbf{s}}_t$ contains logits over ordered target-speed classes; its softmax gives their probabilities.

2.2 DECISION-RELEVANCE STATE

Decision relevance describes how a scene element affects ego planning, given the ego state and driving intent. We represent it as a three-channel field over an ego-centric bird's-eye-view (BEV) domain Ω_{bev}:

$$R_t(\mathbf{p}) = [G_t(\mathbf{p}), M_t^{\mathrm{rel}}(\mathbf{p}), M_t^{\mathrm{crit}}(\mathbf{p})] \in [0,1]^3, \qquad \mathbf{p} \in \Omega_{\mathrm{bev}}. \tag{3}$$

Here $\mathbf{p}$ is a 2D location in the BEV domain Ω_{bev}, and R_t is the relevance field at time t. $G_t(\mathbf{p}) \in [0,1]$ measures geometric relevance, while $M_t^{\mathrm{rel}}(\mathbf{p}), M_t^{\mathrm{crit}}(\mathbf{p}) \in \{0,1\}$ are semantic masks for relevance and criticality, respectively. Geometry measures an actor's spatial coupling to the navigation route and ego vehicle. The Relevant class denotes actors to monitor while the current plan remains executable; Critical denotes actors judged to require a planning response, such as braking, yielding, or avoidance. Irrelevant denotes actors not expected to affect the current maneuver. Geometry thus describes spatial relationships, while semantic labels describe their implications for driving decisions.

The Relevant channel covers both relevant and critical actors, and the Critical channel identifies the response-requiring subset. The inclusion holds pointwise:

$$M_t^{\mathrm{crit}}(\mathbf{p}) \le M_t^{\mathrm{rel}}(\mathbf{p}), \qquad \forall\, \mathbf{p} \in \Omega_{\mathrm{bev}}. \tag{4}$$

3 METHOD

DriveMatter augments end-to-end planning by modeling *what matters now* and predicting *what will matter* as the ego maneuver unfolds. We first construct geometric and semantic supervision (Section 3.1) to train a navigation-conditioned Current Relevance Encoder (Section 3.2). PCR-WM then forecasts future relevance conditioned on a candidate ego plan and observation history (Section 3.3). Section 3.4 connects these representations through two-stage refinement: current relevance refines the base route and target speed into the candidate plan, while predicted future relevance guides a second correction to produce the final plan. Figure 2 summarizes the pipeline.

3.1 STRUCTURED DECISION-RELEVANCE SUPERVISION

To supervise relevance-guided planning refinement, we construct three-channel targets from geometric and semantic cues, since standard driving demonstrations lack decision-relevance annotations.

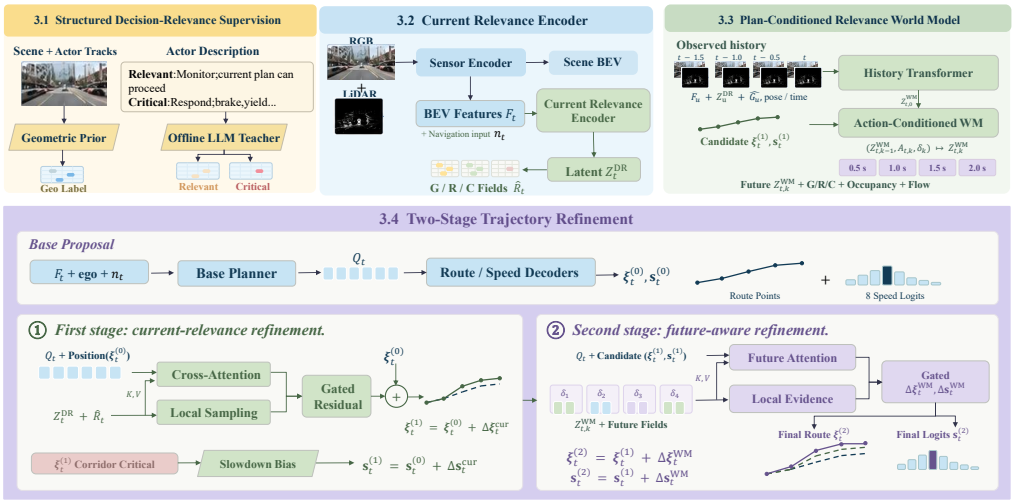


Figure 2: **Overview of DriveMatter.** Structured geometric and semantic supervision trains the Current Relevance Encoder. The Plan-Conditioned Relevance World Model forecasts relevance. Current and forecast relevance jointly guide two-stage trajectory refinement. Superscripts $j = 0, 1, 2$ denote base, candidate, and final predictions; dashed and solid routes indicate predictions before and after refinement.

Geometric prior. Actors near the intended route may constrain passage, while those close to the ego may affect local maneuvers. We capture both through route-coupling and ego-proximity scores:

$$g_{i,u}^{r} = \exp\left(-\frac{(d_{i,u}^{r})^2}{2\sigma_r^2}\right) \cdot \max\left[\rho, \mathrm{sigmoid}\left(\frac{s_{i,u}}{\tau_s}\right)\right], \tag{5}$$

$$g_{i,u}^{e} = \alpha_e \exp\left(-\frac{(d_{i,u}^{e})^2}{2\sigma_e^2}\right), \qquad g_{i,u} = \max(g_{i,u}^{r}, g_{i,u}^{e}). \tag{6}$$

For actor $i \in \mathcal{A}_t$ at time u, $d_{i,u}^{r}$ and $d_{i,u}^{e}$ are distances to the route and ego; $s_{i,u}$ is the relative longitudinal position along the route, positive ahead. The bandwidths σ_r, σ_e control spatial decay, τ_s controls the longitudinal transition, ρ sets its weight floor, and α_e scales ego proximity. Taking the maximum preserves either source of spatial relevance. To also capture actors that will soon approach the route or ego, we aggregate over a short future horizon:

$$\bar{g}_{i,t} = \max_{h \in \mathcal{H}_{i,t}} \exp(-\lambda_g h) \cdot g_{i,t+h}, \qquad \mathcal{H}_{i,t} \subseteq \{0, 0.5, \ldots, 2.0\}\,\mathrm{s}. \tag{7}$$

Here, $\mathcal{H}_{i,t}$ contains valid observation offsets h, and λ_g discounts more distant future scores. The resulting $\bar{g}_{i,t}$ labels the current actor using both present proximity and anticipated spatial coupling.

Semantic relevance. Spatial proximity alone does not determine whether a participant requires the ego to brake, yield, or adjust its path. An offline large language model (LLM) pipeline $\mathcal{T}_{\mathrm{sem}}$ assigns a semantic label $\ell_{i,t}$ to participant i at time t:

$$\ell_{i,t} = \mathcal{T}_{\mathrm{sem}}\left(X_{i,t}^{\mathrm{cur}}, X_{i,t}^{\mathrm{priv}}\right) \in \{\mathrm{I}, \mathrm{R}, \mathrm{C}\}, \tag{8}$$

where $X_{i,t}^{\mathrm{cur}}$ and $X_{i,t}^{\mathrm{priv}}$ contain structured current states and privileged future interaction cues. Before rasterization, validators check the structured output and actor identifiers to ensure that annotations can be associated with the corresponding scene participants. Validated labels form M_t^{rel} and M_t^{crit}, complementing geometric proximity with planning-response requirements.

Dense target. We rasterize the future-aware scores $\bar{g}_{i,t}$ around actors' current footprints to obtain G_t. Together with the semantic relevance mask M_t^{rel} and criticality mask M_t^{crit}, this forms the supervision target R_t^* for the three-channel field defined in Section 2.2. These targets supervise the Current Relevance Encoder to identify what matters now for ego planning.

3.2 CURRENT RELEVANCE ENCODER

To identify *what matters now*, we condition scene understanding on the intended maneuver. The sensor encoder $\mathcal{E}_{\text{sens}}$ extracts BEV features F_t from camera observations $\mathbf{I}_t$ and LiDAR raster $\mathbf{L}_t$:

$$F_t = \mathcal{E}_{\text{sens}}(\mathbf{I}_t, \mathbf{L}_t). \tag{9}$$

Conditioned on navigation input $\mathbf{n}_t$, the Current Relevance Encoder $\mathcal{E}_{\text{rel}}$ maps F_t to a relevance latent Z_t^{DR} and an explicit relevance field $\hat{R}_t$:

$$(Z_t^{\text{DR}}, \hat{R}_t) = \mathcal{E}_{\text{rel}}(F_t, \mathbf{n}_t). \tag{10}$$

The field contains three predicted channels:

$$\hat{R}_t = [\hat{G}_t, \hat{M}_t^{\text{rel}}, \hat{M}_t^{\text{crit}}]. \tag{11}$$

We supervise them using the targets in Section 3.1:

$$\begin{aligned}\mathcal{L}_{\text{rel}} = \ell_{\text{SL1}}(\hat{G}_t, G_t) + \ell_{\text{BCE}}(\hat{G}_t, G_t) \\ + \frac{1}{2} \sum_{c \in \{\text{rel}, \text{crit}\}} \ell_{\text{BCE}}(\hat{M}_t^c, M_t^c).\end{aligned} \tag{12}$$

Here, $\hat{G}_t$, $\hat{M}_t^{\text{rel}}$, and $\hat{M}_t^{\text{crit}}$ predict geometric relevance, semantic relevance, and semantic criticality; their unhatted counterparts are supervision targets. ℓ_{SL1} and ℓ_{BCE} denote Smooth-L_1 and binary cross-entropy losses. Together, the latent and field guide the initial route and speed refinement, whose candidate plan conditions future prediction.

3.3 PLAN-CONDITIONED RELEVANCE WORLD MODEL

The Current Relevance Encoder identifies *what matters now*. PCR-WM anticipates *what will matter* under the resulting candidate plan. Patch Embedding and a History Transformer aggregate BEV features F_u, relevance latents Z_u^{DR}, and predicted geometry $\hat{G}_u$ into $Z_{t,0}^{\text{WM}}$, summarizing motion context from four observations u spanning the past 1.5 s. The candidate route, speed distribution, and measured ego speed define timed actions $A_{t,k}$. A shared two-layer transition Transformer f_{WM} advances the state using each action and horizon embedding E_A:

$$\begin{aligned}Z_{t,k}^{\text{WM}} = \text{Select}_{1:64}\big[f_{\text{WM}}\big([Z_{t,k-1}^{\text{WM}}; E_A(A_{t,k}, \delta_k)]\big)\big], \\ \delta_k = 0.5k\,\text{s}, \qquad k = 1, \ldots, 4.\end{aligned} \tag{13}$$

Each step predicts the latent state at horizon δ_k; $[\cdot;\cdot]$ concatenates tokens, and $\text{Select}_{1:64}$ retains 64 spatial tokens. A Field Decoder maps each state to relevance $\widehat{R}_{t,k}$ (Geometry, Relevant, and Critical), occupancy $\widehat{O}_{t,k}$, and backward flow $\widehat{\mathbf{U}}_{t,k}$ in the current ego frame. These fields and future latents jointly guide the second route and speed refinement (Section 3.4).

3.4 TWO-STAGE TRAJECTORY REFINEMENT

We connect current and predicted relevance through two successive route and speed corrections (Fig. 2). Superscripts (0), (1), and (2) identify the base proposal, current-corrected candidate, and final output; $\boldsymbol{\xi}_t^{(j)}$ and $\mathbf{s}_t^{(j)}$ denote route points and speed-class logits. These inference refinements are distinct from the two training stages.

First stage: current-relevance refinement. The base planner forms planning queries Q_t from scene features, ego state, and navigation input. Route and speed decoders produce the base proposal $(\boldsymbol{\xi}_t^{(0)}, \mathbf{s}_t^{(0)})$. Route queries augmented with base-point positions use *Cross-Attention* to read current context from Z_t^{DR} and $\hat{R}_t$. *Local Sampling* extracts features and relevance values at the proposed points. The *Gated Residual* head combines these features to predict route offsets $\Delta\boldsymbol{\xi}_t^{\text{cur}}$. Criticality within the corrected-route corridor controls a *Slowdown Bias* $\Delta\mathbf{s}_t^{\text{cur}}$ favoring lower speed classes:

$$\begin{aligned}\boldsymbol{\xi}_t^{(1)} = \boldsymbol{\xi}_t^{(0)} + \Delta\boldsymbol{\xi}_t^{\text{cur}}, \\ \mathbf{s}_t^{(1)} = \mathbf{s}_t^{(0)} + \Delta\mathbf{s}_t^{\text{cur}}.\end{aligned} \tag{14}$$

This candidate plan conditions PCR-WM.

Second stage: future-aware refinement. Planning queries Q_t are further conditioned on the candidate $(\boldsymbol{\xi}_t^{(1)}, \mathbf{s}_t^{(1)})$. *Future Attention* retrieves context from future latents $Z_{t,k}^{\mathrm{WM}}$, while *Local Evidence* samples predicted relevance, occupancy, and flow around the candidate route and timed ego poses, preserving horizon-specific information. Separate gated heads combine both branches with the query context to predict route offsets $\Delta\boldsymbol{\xi}_t^{\mathrm{WM}}$ and speed-logit adjustments $\Delta\mathbf{s}_t^{\mathrm{WM}}$:

$$\begin{aligned} \boldsymbol{\xi}_t^{(2)} &= \boldsymbol{\xi}_t^{(1)} + \Delta\boldsymbol{\xi}_t^{\mathrm{WM}}, \\ \mathbf{s}_t^{(2)} &= \mathbf{s}_t^{(1)} + \Delta\mathbf{s}_t^{\mathrm{WM}}. \end{aligned} \tag{15}$$

The final route and speed logits are $(\hat{\boldsymbol{\xi}}_t, \hat{\mathbf{s}}_t) = (\boldsymbol{\xi}_t^{(2)}, \mathbf{s}_t^{(2)})$. Unlike the first-stage slowdown bias, the future correction can increase or decrease individual speed logits. The final plan is not rolled out again.

After establishing the current-relevance policy, we jointly train future prediction and planning refinement with

$$\mathcal{L}_{\mathrm{joint}} = \lambda_{\mathrm{route}}\mathcal{L}_{\mathrm{route}} + \lambda_{\mathrm{speed}}\mathcal{L}_{\mathrm{speed}} + \lambda_{\mathrm{WM}}\mathcal{L}_{\mathrm{WM}} + \mathcal{L}_{\mathrm{reg}}. \tag{16}$$

Here, $\mathcal{L}_{\mathrm{route}}$ and $\mathcal{L}_{\mathrm{speed}}$ supervise the final route and target speed, while $\mathcal{L}_{\mathrm{WM}}$ supervises future relevance, occupancy, and flow. $\mathcal{L}_{\mathrm{reg}}$ regularizes the policy and refinement residuals, and the λ coefficients balance the objectives. Geometric and semantic labeling procedures are detailed in Appendices E and F, respectively. Network architecture and training details are provided in Appendix D. Together, the refinements connect *what matters now* with *what will matter* to adjust route and speed.

4 Experiments

We evaluate DriveMatter through three questions: (1) whether decision-relevance modeling improves closed-loop driving, (2) whether the same formulation remains effective when transferred to a different base planner and expert-data setting, and (3) how geometric and semantic supervision contribute to current-relevance planning, and what additional gains WM-guided refinement provides.

4.1 Experimental Setup

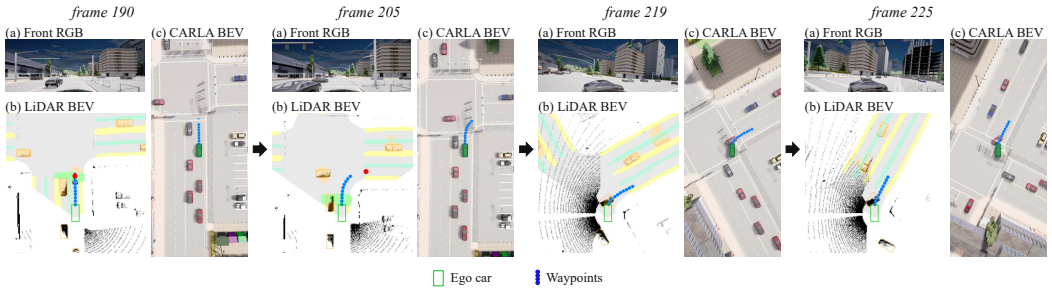
Our primary experiments build on TransFuser++ using PDM-Lite demonstrations (Beisswenger, 2024). We further transfer DriveMatter to TFv6 using LEAD demonstrations (Nguyen et al., 2026), preserving the three-channel relevance formulation and two-stage planning refinement while adapting to the host planner and sensor configuration. Both datasets are collected in CARLA (Dosovitskiy et al., 2017). Geometric and semantic labels are constructed offline and used only for training; inference uses predicted relevance. Training details and transfer configurations are provided in appendix. We evaluate on the 220 Bench2Drive routes (Jia et al., 2024), using Driving Score (DS) and Success Rate (SR) as the primary metrics. Efficiency, Comfort, and Multi-Ability scores further assess progress, motion smoothness, and driving abilities. We evaluate each of our model configurations on all 220 routes using three random seeds and report the mean across runs. Standard deviations are provided in the appendix. Results for prior methods are taken from their respective papers.

4.2 Comparison on Bench2Drive

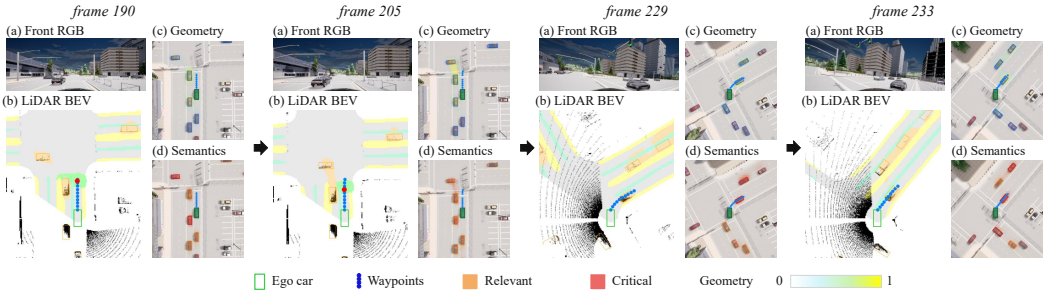
Main results. Table 1 reports the closed-loop results on Bench2Drive. DriveMatter achieves a Driving Score of 89.24 and a Success Rate of 78.18%, improving over the base planner by 5.03 and 10.91, respectively. It also exceeds the strongest listed results by 0.82 driving-score points and 3.19 success-rate percentage points. These gains demonstrate the effectiveness of relevance-guided trajectory refinement in improving overall closed-loop performance. KnowVal-SimLingo (Xia et al., 2026) also uses PDM-Lite demonstrations and incorporates a world model that predicts future ego, instance, and occupancy-grid features for candidate trajectory evaluation. DriveMatter exceeds its SR by 9.15%, providing a comparison with a world-model-based planning approach under the same expert-data source. The Multi-Ability results in Appendix B.1 further support this finding: DriveMatter improves all five abilities over the base planner and achieves the highest mean score among

Table 1: **Results on the Bench2Drive benchmark.** * denotes expert-feature distillation.

Method	Venue	Expert	Closed-loop Metrics			
			DS ↑	SR (%) ↑	Efficiency ↑	Comfort ↑
TCP-traj* Wu et al. (2022)	NeurIPS 2022	Think2Drive	59.90	30.00	76.54	18.08
UniAD-Base Hu et al. (2023)	CVPR 2023	Think2Drive	45.81	16.36	129.21	43.58
VAD Jiang et al. (2023)	ICCV 2023	Think2Drive	42.35	15.00	157.94	46.01
DriveTransformer Jia et al. (2025)	ICLR 2025	Think2Drive	63.46	35.01	100.64	20.78
ORION Fu et al. (2025)	ICCV 2025	Think2Drive	77.74	54.62	151.48	17.38
ORION diffusion Fu et al. (2025)	ICCV 2025	Think2Drive	71.97	46.54	-	-
DiffusionDrive$^{\text{temp}}$ Liao et al. (2025)	CVPR 2025	PDM-Lite	77.68	52.72	248.18	24.56
DiffusionDrive$^{\text{geo}}$ Liao et al. (2025)	CVPR 2025	PDM-Lite	80.79	58.18	245.34	15.49
SimLingo Renz et al. (2025)	CVPR 2025	PDM-Lite	85.07	67.27	259.23	33.67
BridgeDrive Liu et al. (2026)	ICLR 2026	PDM-Lite	87.99	74.99	236.49	20.98
KnowVal-SimLingo Xia et al. (2026)	CVPR 2026	PDM-Lite	88.42	69.03	–	–
BevAD-M Holtz et al. (2026)	CVPR 2026	PDM-Lite	88.11	72.73	–	–
TransFuser++ Zimmerlin et al. (2024)	arXiv 2024	PDM-Lite	84.21	67.27	-	-
DriveMatter	**Ours**	PDM-Lite	**89.24**	**78.18**	247.91	24.43



(a) TransFuser++ collides with crossing traffic



(b) DriveMatter completes the turn without collision

Figure 3: **Coordinating a right turn with crossing traffic.** (a) TransFuser++ enters the conflict region prematurely and collides at frame 219. (b) DriveMatter slows down and yields to the crossing vehicle, then completes the route without collision. Frame indices refer to the respective rollouts.

the listed methods, at 74.23%, an increase of 9.84 %. It also obtains the highest scores in merging and traffic-sign handling, highlighting its ability to coordinate maneuvers with surrounding traffic and road constraints.

Interaction behavior. Figure 3 compares a right turn that requires coordinating passage with crossing traffic. In Figure 3(a), the original TransFuser++ enters the conflict region before the crossing vehicle has cleared it, resulting in overlapping vehicle occupancy and a collision at frame 219. In Figure 3(b), DriveMatter reaches this region later in its rollout. At frame 229, the ego is approximately at the baseline collision position, but the crossing vehicle has already cleared the immediate conflict region. DriveMatter subsequently completes the turn and the entire route without collision.

This comparison illustrates the value of jointly modeling current and future decision relevance. Current relevance identifies the crossing vehicle as a constraint on the intended right turn, while

future prediction provides information about how this constraint evolves as the vehicle traverses the conflict region. Together, these signals support coordinating the ego's approach with the evolving interaction. DriveMatter's later arrival, after the crossing vehicle has passed, is consistent with the intended role of relevance prediction in guiding both where to drive and when to proceed.

Additional obstacle-passing and pedestrian-crossing cases are analyzed in Appendix C.2.

4.3 Generalization to LEAD Expert Data

Table 2: **Performance of DriveMatter adapted to LEAD.** * identifies the TFv6 configuration with a ResNet-34 image backbone, a 140° camera field of view, LiDAR, and radar, also used by DriveMatter.

Method	Expert	DS	SR (%)	Efficiency	Comfort
TFv6* Nguyen et al. (2026)	LEAD	94.7 ± 0.7	82.1 ± 3.6	-	-
DriveMatter (ours)	LEAD	96.9 ± 0.4	90.2 ± 1.7	206.5 ± 3.3	22.3 ± 0.8

Table 2 evaluates DriveMatter with TFv6 as the base planner and LEAD demonstrations as the expert data. DriveMatter improves DS from 94.7 ± 0.7 to 96.9 ± 0.4 and SR from $82.1\% \pm 3.6\%$ to $90.2\% \pm 1.7\%$. These results suggest that the benefits of decision-relevance modeling extend beyond the PDM-Lite/TransFuser++ setting to a different base planner and expert-data distribution.

The transferred model also achieves a mean Multi-Ability score of 84.65%. As detailed in Appendix B.2, it maintains strong performance across diverse interaction scenarios, providing further evidence for the generalizability of the proposed relevance formulation.

4.4 Ablation Studies

Our ablations examine how prediction targets, relevance supervision, and future-aware refinement affect closed-loop planning performance.

Prediction targets. Table 3 compares prediction targets under the same two-stage planning pipeline. Image-WM reconstructs the current RGB image and predicts future images over 2 s, while OCC-WM predicts current and future occupancy. Both guide route and speed refinement through the same planner interface. Image-WM achieves 86.69 DS and 70.45% SR, improving over the base planner. OCC-WM further increases these scores to 87.53 and 72.27%. DriveMatter performs best at 89.24 DS and 78.18% SR, exceeding OCC-WM by 1.71 DS points and 5.91% in absolute SR. These results suggest that appearance and occupancy prediction alone may not sufficiently emphasize interactions that matter for the intended maneuver. Relevance supervision directs learning toward these interactions, making current and future representations more useful for route and speed refinement.

Table 3: **Prediction-target comparison on Bench2Drive.**

Method	Current target	Future target	DS ↑	SR (%) ↑
TransFuser++ (baseline)	–	–	84.21	67.27
Image-WM	RGB	RGB	86.69	70.45
OCC-WM	OCC	OCC	87.53	72.27
DriveMatter	Relevance	Relevance	**89.24**	**78.18**

Relevance supervision. To assess the complementary roles of geometry and semantics, Table 4 compares three supervision variants for current-relevance planning without PCR-WM. Geometry-only achieves 85.88 DS and 70.45% SR, improving over the base planner by 1.67 DS points and 3.18% in absolute SR. Semantic-only reaches 87.05 DS and 73.64% SR, with corresponding gains of 2.84 points and 6.37%. Combining both as three-channel supervision yields the strongest results of 88.11 DS and 75.00% SR, exceeding the baseline by 3.90 points and 7.73%, respectively.

These results show that both supervision sources improve planning. The larger gains from semantic supervision highlight the value of distinguishing planning-response requirements, while the best combined performance supports the complementary role of geometric spatial cues.

Table 4: **Ablations on relevance supervision and future refinement.**

Method	Geometry	Relevant	Critical	PCR-WM	DS ↑	SR (%) ↑
TransFuser++ (baseline)					84.21	67.27
Geometry-only	✓				85.88	70.45
Semantic-only		✓	✓		87.05	73.64
Three-channel	✓	✓	✓		88.11	75.00
DriveMatter	✓	✓	✓	✓	**89.24**	**78.18**

Future refinement. To assess the additional benefit of future prediction and refinement, we compare the current-only Three-channel variant with the complete DriveMatter model under the same relevance supervision. Adding PCR-WM and its second route and speed correction increases DS from 88.11 to 89.24 and SR from 75.00% to 78.18%. These gains support using candidate-plan-conditioned future relevance to further refine current-relevance planning. Ability-specific ablations are reported in Appendix B.1.

5 Conclusion

We introduced DriveMatter, a decision-relevance world model for end-to-end autonomous driving that explicitly connects *what matters now* with *what will matter* as an ego maneuver unfolds. Structured geometric and semantic supervision trains a navigation-conditioned Current Relevance Encoder to guide the first refinement of a base route and target speed. Conditioned on the resulting candidate plan and observation history, PCR-WM forecasts future relevance and guides a second refinement. Privileged future cues are used only to construct training supervision. On Bench2Drive, DriveMatter achieves absolute Success Rate gains of 10.91% over the TransFuser++ baseline when trained with PDM-Lite demonstrations and 8.1% over the corresponding TFv6 baseline when trained with LEAD demonstrations.

Limitations and future work. (1) Although relevance is annotated for individual traffic participants, changes in the ego state and intended maneuver can rapidly shift a participant's label between *Relevant* and *Critical*. Learning these context-dependent transitions is challenging, particularly because Critical supervision is substantially sparser. In addition, despite automated annotation checks, manual review of 368,725 observations of individual traffic participants across 57,260 frames from 500 routes identified a semantic label error rate of 7.96%. The model occasionally predicts Critical for Relevant participants. Future work will explore temporal modeling of relevance transitions, class-balanced learning, and probabilistic relevance supervision. (2) The Critical channel identifies participants that require a planning response, but leaves the choice of braking, yielding, or avoidance to the planner. The planner must therefore learn the appropriate response implicitly and can struggle when a Critical participant changes its motion rapidly, as illustrated in Appendix C.3. Future work will investigate response-aware Critical representations and uncertainty-aware prediction of participant motion.

AI use statement

We use a language model to generate offline semantic labels for training (Section 3.1) and AI tools to polish the manuscript.

References

Jens Beisswenger. PDM-Lite: A rule-based planner for CARLA leaderboard 2.0. Technical report, University of Tübingen, 2024. URL https://github.com/OpenDriveLab/DriveLM/blob/DriveLM-CARLA/pdm_lite/docs/report.pdf.

Jiajun Cao, Qizhe Zhang, Peidong Jia, Xuhui Zhao, Bo Lan, Xiaoan Zhang, Xiaobao Wei, Sixiang Chen, Zhuo Li, Yang Wang, Liyun Li, Xianming Liu, Ming Lu, and Shanghang Zhang. Fastdrivevla: Efficient end-to-end driving via plug-and-play reconstruction-based token pruning. *arXiv preprint arXiv:2507.23318*, 2025.

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 66–75. PMLR, 2020. URL https://proceedings.mlr.press/v100/chen20a.html.

Simon Doll, Niklas Hanselmann, Lukas Schneider, Richard Schulz, Marius Cordts, Markus Enzweiler, and Hendrik P. A. Lensch. Dualad: Disentangling the dynamic and static world for end-to-end driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14728–14737, 2024.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 1–16. PMLR, 2017. URL https://proceedings.mlr.press/v78/dosovitskiy17a.html.

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*, 2026.

Haoyu Fu, Diankun Zhang, Zongchuang Zhao, Jianfeng Cui, Dingkang Liang, Chong Zhang, Dingyuan Zhang, Hongwei Xie, Bing Wang, and Xiang Bai. ORION: A holistic end-to-end autonomous driving framework by vision-language instructed action generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 24823–24834, 2025.

Mingzhe Guo, Yixiang Yang, Chuanrong Han, Rufeng Zhang, Shirui Li, Ji Wan, and Zhipeng Zhang. FlowAD: Ego-scene interactive modeling for autonomous driving. *arXiv preprint arXiv:2603.13399*, 2026.

David Holtz, Niklas Hanselmann, Simon Doll, Marius Cordts, and Bernt Schiele. What matters for scalable and robust learning in end-to-end driving planners? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 931–941, 2026.

Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. ST-P3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *European Conference on Computer Vision*, 2022.

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17853–17862, 2023. doi: 10.1109/CVPR52729.2023.01712. URL https://doi.org/10.1109/CVPR52729.2023.01712.

Keishi Ishihara, Anssi Kanervisto, Jun Miura, and Ville Hautamaki. Multi-task learning with attention for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 2902–2911, 2021.

Feiyang Jia, Lin Liu, Ziying Song, Caiyan Jia, Hangjun Ye, Xiaoshuai Hao, and Long Chen. Driveworld-vla: Unified latent-space world modeling with vision-language-action for autonomous driving, 2026. URL https://arxiv.org/abs/2602.06521.

Xiaosong Jia, Yulu Gao, Li Chen, Junchi Yan, Patrick Langechuan Liu, and Hongyang 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*, 2023a.

Xiaosong Jia, Penghao Wu, Li Chen, Jiangwei Xie, Conghui He, Junchi Yan, and Hongyang 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*, 2023b.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *Proceedings of the NeurIPS 2024 Datasets and Benchmarks Track*, 2024. URL https://arxiv.org/abs/2406.03877. Accepted by NeurIPS 2024 Datasets and Benchmarks Track.

Xiaosong Jia, Junqi You, Zhiyuan Zhang, and Junchi Yan. Drivetransformer: Unified transformer for scalable end-to-end autonomous driving. In *International Conference on Learning Representations*, 2025.

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Vad: Vectorized scene representation for efficient autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8340–8350, 2023. URL https://arxiv.org/abs/2303.12077. Accepted to ICCV 2023.

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 (ICLR)*, 2025a. URL https://arxiv.org/abs/2406.08481.

Yingyan Li, Shuyao Shang, Weisong Liu, Bing Zhan, Haochen Wang, Yuqi Wang, Yuntao Chen, Xiaoman Wang, Yasong An, Chufeng Tang, Lu Hou, Lue Fan, and Zhaoxiang Zhang. Drivevla-w0: World models amplify data scaling law in autonomous driving, 2025b. URL https://arxiv.org/abs/2510.12796. ArXiv preprint.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, Bing Wang, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving, 2025c. URL https://arxiv.org/abs/2506.08052. ArXiv preprint.

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*, 2025.

Hongbin Lin, Yiming Yang, Yifan Zhang, Chaoda Zheng, Jie Feng, Sheng Wang, Zhennan Wang, Shijia Chen, Boyang Wang, Yu Zhang, Xianming Liu, Shuguang Cui, and Zhen Li. Futurex: Enhance end-to-end autonomous driving via latent chain-of-thought world model, 2025. URL https://arxiv.org/abs/2512.11226. ArXiv preprint.

Shu Liu, Wenlin Chen, Weihao Li, Zheng Wang, Lijin Yang, Jianing Huang, Yipin Zhang, Zhongzhan Huang, Ze Cheng, and Hao Yang. BridgeDrive: Diffusion bridge policy for closed-loop trajectory planning in autonomous driving. In *International Conference on Learning Representations*, 2026.

Long Nguyen, Micha Fauth, Bernhard Jaeger, Daniel Dauner, Maximilian Igl, Andreas Geiger, and Kashyap Chitta. LEAD: Minimizing learner-expert asymmetry in end-to-end driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39775–39785, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Nguyen_LEAD_Minimizing_Learner-Expert_Asymmetry_in_End-to-End_Driving_CVPR_2026_paper.html.

Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7077–7087, 2021. URL https://openaccess.thecvf.com/content/CVPR2021/html/Prakash_Multi-Modal_Fusion_Transformer_for_End-to-End_Autonomous_Driving_CVPR_2021_paper.html.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. SimLingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11993–12003, 2025.

Chonghao Sima, Katrin Renz, Kashyap Chitta, Li Chen, Hanxue Zhang, Chengen Xie, Jens Beisswenger, Ping Luo, Andreas Geiger, and Hongyang Li. DriveLM: Driving with graph visual question answering. In *European Conference on Computer Vision*, pp. 256–274, 2024. URL https://www.ecva.net/papers/eccv_2024/papers_ECCV/html/6870_ECCV_2024_paper.php.

Wenchao Sun, Xuewu Lin, Yining Shi, Chuang Zhang, Haoran Wu, and Sifa Zheng. Sparsedrive: End-to-end autonomous driving via sparse scene representation. In *2025 IEEE International Conference on Robotics and Automation*, pp. 8795–8801. IEEE, 2025.

Linbo Wang, Yupeng Zheng, Qiang Chen, Shiwei Li, Yichen Zhang, Zebin Xing, Qichao Zhang, Xiang Li, Deheng Qian, Pengxuan Yang, Yihang Dong, Ce Hao, Xiaoqing Ye, Junyu Han, Yifeng Pan, and Dongbin Zhao. Latent-wam: Latent world action modeling for end-to-end autonomous driving. *arXiv preprint arXiv:2603.24581*, 2026.

Yuqi Wang, Jiawei He, Lue Fan, Hongcheng Li, Yuntao Chen, and Zhaoxiang Zhang. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Xinshuo Weng, Boris Ivanovic, Yan Wang, Yue Wang, and Marco Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15449–15458, 2024.

Penghao Wu, Xiaosong Jia, Li Chen, Junchi Yan, Hongyang Li, and Yu Qiao. Trajectory-guided control prediction for end-to-end autonomous driving. In *Advances in Neural Information Processing Systems*, 2022.

Zhongyu Xia, Wenhao Chen, Yongtao Wang, and Ming-Hsuan Yang. KnowVal: A knowledge-augmented and value-guided autonomous driving system. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026. URL https://arxiv.org/abs/2512.20299.

Zhexiao Xiong, Xin Ye, Burhan Yaman, Sheng Cheng, Yiren Lu, Jingru Luo, Nathan Jacobs, and Liu Ren. Unidrive-wm: Unified understanding, planning and generation world model for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2026. URL https://arxiv.org/abs/2601.04453.

Jiawei Xu, Zhizhou Zhong, Zhijian Shu, Mingkai Jia, Mingxiao Li, Jia-Wang Bian, Qian Zhang, Kaicheng Zhang, Jin Xie, Jian Yang, and Wei Yin. Eponav2: Driving world model with comprehensive future reasoning. *arXiv preprint arXiv:2605.14696*, 2026.

Zhenjie Yang, Yilin Chai, Xiaosong Jia, Qifeng Li, Yuqian Shao, Xuekai Zhu, Haisheng Su, and Junchi Yan. DriveMoE: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Yuqi Ye, Zijian Zhang, Junhong Lin, Shangkun Sun, Changhao Peng, and Wei Gao. Autodrive-p^3: Unified chain of perception–prediction–planning thought via reinforcement fine-tuning, 2026. URL https://arxiv.org/abs/2603.28116. ArXiv preprint.

Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, Xun Cao, and Wei Yin. Epona: Autoregressive diffusion world model for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2025.

Yumeng Zhang, Shi Gong, Kaixin Xiong, Xiaoqing Ye, Xiao Tan, Fan Wang, Jizhou Huang, Hua Wu, and Haifeng Wang. Bevworld: A multimodal world model for autonomous driving via unified bev latent space. *arXiv preprint arXiv:2407.05679*, 2024.

Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, and Dongbin Zhao. World4drive: End-to-end autonomous driving via intention-aware physical latent world model, 2025. URL https://arxiv.org/abs/2507.00603. First version accepted to ICCV 2025.

Julian Zimmerlin, Jens Beißwenger, Bernhard Jaeger, Andreas Geiger, and Kashyap Chitta. Hidden biases of end-to-end driving datasets, 2024. URL https://arxiv.org/abs/2412.09602.

A Related Work

End-to-end autonomous driving. End-to-end driving methods integrate perception, prediction, and planning in a learned policy. ST-P3, UniAD, and VAD couple scene understanding to planning through shared representations and task-specific objectives (Hu et al., 2022; 2023; Jiang et al., 2023). More recent methods study structured scene features, temporal interaction, and generative trajectory prediction (Jia et al., 2025; Guo et al., 2026; Feng et al., 2026; Liu et al., 2026). Vision-language-action models also introduce language reasoning or specialized experts into the policy (Yang et al., 2026; Ye et al., 2026; Li et al., 2025c;b). Across these designs, planning quality depends on which information is preserved by the learned scene representation. The supervision normally specifies perception targets, expert trajectories, or a closed-loop reward, but does not identify which actors change a particular decision. Closed-loop CARLA systems such as TransFuser (Prakash et al., 2021), PDM-Lite (Beisswenger, 2024), and LEAD (Nguyen et al., 2026) provide the training and evaluation setting used in this work. These approaches learn what the planner should retain implicitly. We instead supervise the relevance of individual scene elements.

World models for autonomous driving. Driving world models provide future context at several representation levels. LAW and Latent-WAM predict latent scene features or spatial tokens (Li et al., 2025a; Wang et al., 2026). BEVWorld and World4Drive forecast structured or latent bird's-eye-view states for planning (Zhang et al., 2024; Zheng et al., 2025). PCR-WM targets a single policy proposal, predicts relevance together with occupancy and flow, and exposes both native future tokens and local fields to the planner. Drive-WM and DriveVLA-W0 use future visual prediction as dense supervision (Wang et al., 2024; Li et al., 2025b), while Epona and EponaV2 extend visual forecasting to longer horizons and richer geometric or semantic content (Zhang et al., 2025; Xu et al., 2026). FastDriveVLA uses reconstruction quality to prune tokens while retaining information useful to the policy (Cao et al., 2025). These methods differ in whether the prediction target is an image, a structured map, or a latent feature. Their future target still describes scene content or its compressed representation. Our target instead measures an actor's relation to the upcoming ego decision, and the model forecasts a feature trained under that supervision.

Privileged and language-based supervision. Learning by Cheating (Chen et al., 2020) transfers knowledge from a privileged state-based teacher to a vision-based driving policy. TCP and ThinkTwice incorporate teacher-feature supervision into policy learning (Wu et al., 2022; Jia et al., 2023b), while DriveAdapter aligns perception features with a frozen privileged teacher through learnable adapters (Jia et al., 2023a). Complementing these approaches, DriveLM (Sima et al., 2024) represents driving reasoning through graph-structured question answering that connects perception, prediction, and planning. DriveMatter focuses on explicit supervision of how individual actors relate to ego planning. An offline language teacher uses current scene states and privileged future interaction cues to assign actor-level semantic relevance and criticality labels. Combined with geometric relevance targets, these annotations supervise a spatial decision-relevance representation that guides planning. Neither the language teacher nor privileged future information is required at inference.

B Additional Results

B.1 Multi-Ability Analysis on Bench2Drive

Comparison with existing methods. Table 5 reports performance across five driving abilities. DriveMatter improves all five over the TransFuser++ base planner, increasing the mean score from 64.39% to 74.23%, the highest among the listed methods. The largest improvement is in merging, increased from 58.75% to 76.25%. Compared with BridgeDrive, DriveMatter achieves higher merging and traffic-sign scores by 6.33% and 2.64%, respectively, and a higher mean by 1.08%. Overtaking and emergency braking remain lower by 2.23% and 1.67%, while giving way is tied at

50.00%. These gains span interactions with moving participants and responses to road constraints, suggesting that relevance-guided refinement benefits multiple aspects of closed-loop driving.

Table 5: **Performance on the Bench2Drive Multi-Ability benchmark.** * denotes expert-feature distillation. Methods with reported per-ability results are ordered as in Table 1.

Method	Ability (%) ↑					
	Merg.	Overtak.	Emer. Brake	Give Way	Traf. Sign	Mean
TCP-traj* Wu et al. (2022)	8.89	24.29	51.67	40.00	46.28	34.22
UniAD-Base Hu et al. (2023)	14.10	17.78	21.67	10.00	14.21	15.55
VAD Jiang et al. (2023)	8.11	24.44	18.64	20.00	19.15	18.07
DriveTransformer Jia et al. (2025)	17.57	35.00	48.36	40.00	52.10	38.60
ORION Fu et al. (2025)	25.00	**71.11**	78.33	30.00	69.15	54.72
DiffusionDrive$^{\text{temp}}$ Liao et al. (2025)	50.63	26.67	68.33	50.00	76.32	54.38
SimLingo Renz et al. (2025)	54.01	57.04	88.33	**53.33**	82.45	67.03
BridgeDrive Liu et al. (2026)	69.92	66.67	**90.00**	50.00	89.47	73.15
TransFuser++ Zimmerlin et al. (2024)	58.75	57.77	83.33	40.00	82.11	64.39
DriveMatter	**76.25**	64.44	88.33	50.00	**92.11**	**74.23**

Ablation analysis. Image-WM and OCC-WM follow the same training protocol as DriveMatter, with two stages of 30 epochs each, identical batch sizes, and the same total number of optimization steps. All three variants share the two-stage route and speed refinement interface. Image-WM uses current RGB reconstruction for the first refinement and predicts four future images at 0.5 s intervals over 2 s for the second. OCC-WM follows the same procedure using current occupancy prediction and four future occupancy predictions, respectively. Table 6 provides the ability breakdown for the ablations in Tables 3 and 4. Image-WM and OCC-WM achieve mean scores of 64.83% and 68.11%, respectively, compared with 74.23% for DriveMatter. DriveMatter improves over both variants across all five abilities. The largest gains over both variants occur in giving way, suggesting that relevance-oriented prediction is particularly useful when ego motion depends on other participants' evolving constraints.

Table 6: **Ablations on prediction targets and relevance supervision on the Bench2Drive Multi-Ability benchmark.** Image-WM and OCC-WM use RGB and occupancy targets, respectively. Geometry-only, Semantic-only, and Three-channel use current relevance without a world model; DriveMatter additionally incorporates predicted future relevance.

Variant	Ability (%) ↑					
	Merg.	Overtak.	Emer. Brake	Give Way	Traf. Sign	Mean
Image-WM	66.25	55.56	85.00	30.00	87.37	64.83
OCC-WM	68.75	62.22	81.67	40.00	87.89	68.11
Geometry-only	65.00	55.56	88.33	40.00	88.42	67.46
Semantic-only	73.75	60.00	83.33	40.00	90.53	69.52
Three-channel (w/o WM)	76.25	60.00	83.33	40.00	91.58	70.23
DriveMatter	**76.25**	**64.44**	**88.33**	**50.00**	**92.11**	**74.23**

For current-only planning, three-channel supervision achieves a mean score of 70.23%, compared with 67.46% for Geometry-only and 69.52% for Semantic-only. The larger gap without semantic supervision highlights the importance of planning-response requirements beyond spatial proximity, while the combined result supports their complementary roles.

Adding PCR-WM and the second planning correction increases the mean score from 70.23% to 74.23%. Giving way improves from 40.00% to 50.00%, emergency braking from 83.33% to 88.33%, and overtaking from 60.00% to 64.44%. Traffic-sign handling improves slightly, while

merging remains unchanged. These gains suggest that anticipating how interactions evolve helps the planner refine both its maneuver and passage timing beyond using current relevance alone.

B.2 Transfer to TFv6 with LEAD Demonstrations

We transfer DriveMatter from PDM-Lite/TransFuser++ to LEAD/TFv6 to evaluate its applicability across base planners and expert-data settings. Both configurations retain the same relevance definitions and two-stage refinement. Current relevance first refines the base route and target speed into a candidate plan. PCR-WM then forecasts future relevance conditioned on this candidate to guide the second refinement. The adaptation involves three components:

1. **Sensor encoding and BEV input.** PDM-Lite/TransFuser++ uses RegNetY-3.2GF with camera and LiDAR inputs. LEAD/TFv6 uses ResNet-34 with a 140° front camera, LiDAR, and radar. The relevance modules receive BEV features from the corresponding host encoder.
2. **Expert data and supervision.** We apply the same geometric and semantic labeling rules to both expert datasets, producing a consistent three-channel supervision target. Dataset-specific preprocessing accommodates differences in how vehicle states and future trajectories are stored.
3. **Planning and control.** Each configuration retains its host's base route and speed outputs and low-level controller. The current adapter and future bridge apply two successive corrections before control. Inference uses only onboard observations, ego state, and navigation, with privileged future information reserved for training supervision.

Closed-loop performance. Table 2 compares DriveMatter with the TFv6 reference under the listed LEAD sensor and backbone configuration. DriveMatter increases Driving Score from 94.7 to 96.9 and Success Rate from 82.1% to 90.2%. Together with the PDM-Lite/TransFuser++ results, these improvements show that relevance-guided refinement transfers across different planner and expert-data settings. The same relevance formulation remains effective despite changes in the underlying driving system.

Ability comparison across configurations. Table 7 compares DriveMatter under the two configurations. LEAD/TFv6 achieves a mean score of 84.65%, compared with 74.23% for PDM-Lite/TransFuser++. The largest differences occur in overtaking, which increases from 64.44% to 97.78%, and merging, from 76.25% to 86.25%. Emergency braking improves from 88.33% to 95.00%, and traffic-sign handling from 92.11% to 94.21%, while giving way remains at 50.00%. The largest differences therefore concern maneuvers that require coordinating lateral motion with surrounding traffic. The shared refinement structure remains applicable in the LEAD/TFv6 configuration, where these abilities reach substantially higher scores.

Table 7: **Multi-ability evaluation of DriveMatter across expert data sources.** All ability scores are reported in percentage (%), and higher values indicate better performance.

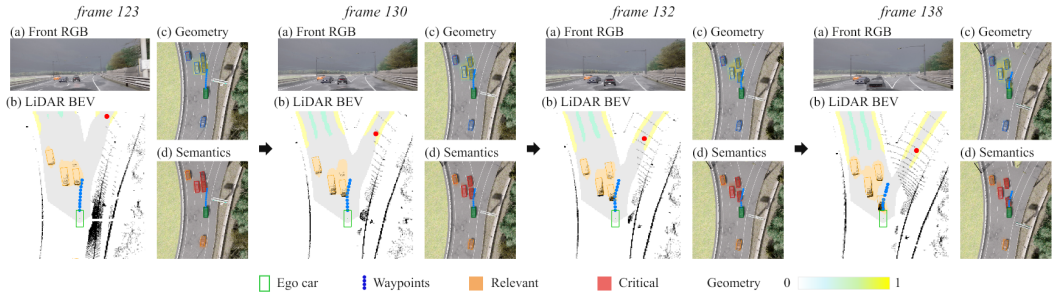
Method	Expert Data	Ability (%)					
		Merg.	Overtak.	Emer. Brake	Give Way	Traf. Sign	Mean
DriveMatter (ours)	PDM-Lite	76.25	64.44	88.33	50.00	92.11	74.23
DriveMatter (ours)	LEAD	86.25	97.78	95.00	50.00	94.21	84.65

C Qualitative Analysis and Failure Cases

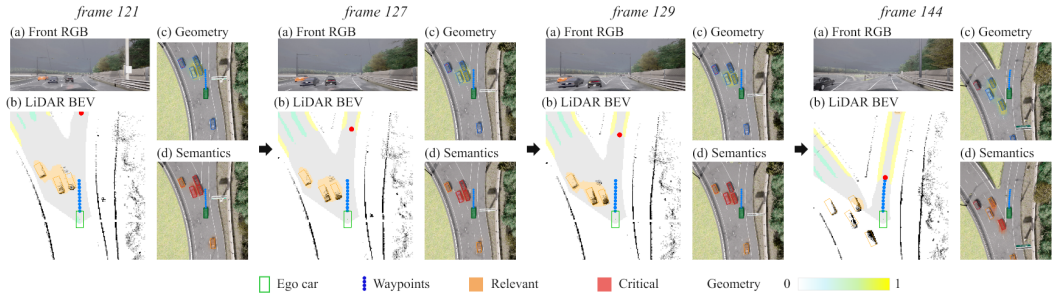
We compare the current-only three-channel policy with the full DriveMatter model to examine the contribution of candidate-conditioned future modeling. Both use Geometry, Relevant, and Critical; the full model additionally uses PCR-WM and its decision bridge. We also examine successful obstacle-passing and pedestrian-crossing interactions before analyzing failures and limitations. Qualitative cases illustrate individual behaviors, while Table 4 provides the aggregate comparison.

C.1 Comparison with and without the PCR-WM

Both variants in Figure 4 use Geometry, Relevant, and Critical supervision. The current-only variant produces a candidate plan using the current relevance representation and first planning correction. DriveMatter additionally conditions PCR-WM on this candidate and uses the predicted future interaction to apply a second correction to the route and target speed, as described in Sections 3.3 and 3.4. The two rollouts use Route 23659 and are aligned by interaction stage because their closed-loop trajectories evolve at different rates.



(a) Without PCR-WM



(b) With PCR-WM

Figure 4: **Effect of PCR-WM on route and target-speed refinement.** Without PCR-WM, the predicted route turns toward the occupied traffic stream before sufficient separation is established, and the rollout ultimately results in a collision. With PCR-WM, DriveMatter preserves a feasible passing corridor, coordinates the route and target speed, and completes the interaction without collision. Frame indices refer to the respective rollouts.

Without PCR-WM, Figure 4(a), the planner initially approaches the interaction from the right lane. The dominant target-speed class changes from 64 km/h at frame 123 to 14–29 km/h at frame 130, returns to 50 km/h at frame 132, and falls to 0 km/h at frame 138. Meanwhile, the predicted waypoints progressively bend toward the occupied traffic stream, and the ego heading begins to turn left before sufficient longitudinal separation has been established. These repeated route and speed changes bring the ego into the conflict region alongside the neighboring vehicles, and the rollout ultimately records a vehicle collision.

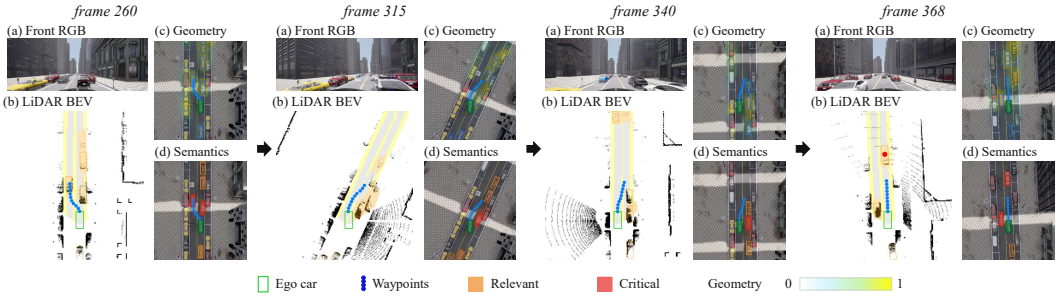
With PCR-WM (Figure 4(b)), the most likely target-speed class remains within 50–64 km/h across the selected frames. The planned waypoints maintain a clearer path through the right-side corridor during frames 121–129, together with a more stable ego heading. This coordinated route and speed refinement allows the ego to establish longitudinal separation before passing the neighboring vehicle cluster. By frame 144, these vehicles are behind and to the left of the ego, which continues along the intended route without collision or route-lane departure.

Current relevance identifies the participants that constrain the maneuver at the present time, but it does not directly describe how their constraints will evolve under the candidate plan. PCR-WM predicts future interactions to guide a second refinement of the route and target speed. In this example, the correction preserves a feasible spatial corridor while coordinating when the ego passes the

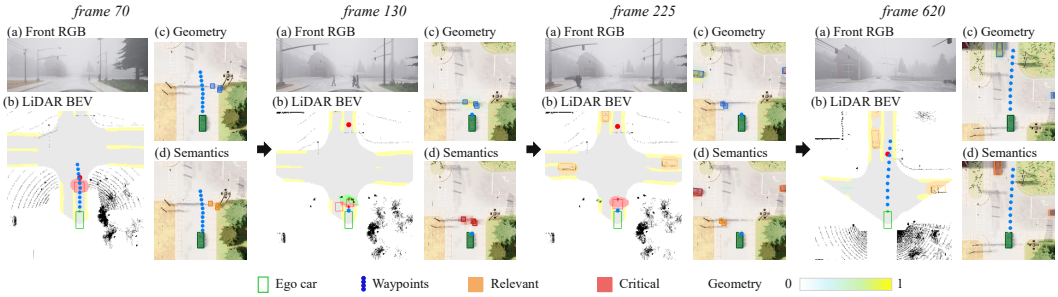
neighboring vehicles. The comparison therefore illustrates how future relevance supports the joint refinement of the candidate route and target speed.

C.2 Relevance-Guided Interaction Behavior

Beyond the comparison with the current-only policy, we examine how DriveMatter coordinates route and speed in two further interactions. Figure 5 shows obstacle passing on Route 1852 and pedestrian crossing on Route 25863.



(a) Passing obstructing vehicles



(b) Yielding to crossing pedestrians

Figure 5: **Successful interactions with vehicles and pedestrians.** (a) DriveMatter waits for oncoming traffic, passes the obstruction, and returns to its lane on Route 1852. (b) It stops for crossing pedestrians on Route 25863, whose semantic relevance changes as they enter and clear the ego path.

Passing obstructing vehicles. In Figure 5(a), obstructing vehicles occupy the ego lane while oncoming traffic limits access to the adjacent lane. DriveMatter waits behind the obstruction at frame 260, initiates a leftward maneuver after the oncoming traffic passes (315), proceeds alongside the obstruction (340), and returns to its original lane (368). The vehicles highlighted as Critical change as the maneuver unfolds. This sequence illustrates the need to coordinate the passing route with the timing of oncoming traffic, rather than treating obstacle avoidance as a purely spatial decision.

Yielding to crossing pedestrians. In Figure 5(b), pedestrians approaching from the right are marked Relevant at frame 70 and Critical as they cross the ego path at frame 130. DriveMatter stops to yield and continues waiting while the crossing progresses (225). By frame 620, the pedestrians have cleared its path. Their changing relevance reflects how they constrain the planned motion, rather than their visibility alone. Here, the appropriate response is to adjust speed and wait, whereas the obstacle-passing case also requires a lateral route change. Both routes achieve 100% route completion and a Driving Score of 100 without collision.

C.3 Failure Cases of the Full Model

However, abrupt changes in surrounding vehicles' speed remain challenging for closed-loop planning. As shown in Figure 6, the neighboring vehicle is ahead and to the right of the ego at frame 55, traveling at approximately 12.65 m/s. It then decelerates sharply to 2.58 m/s at frame 60 and approaches a standstill at frames 65–70. After a brief pause, the vehicle resumes motion, while the

ego begins accelerating to pass it. By frame 95, the neighboring vehicle has accelerated to 4.60 m/s, while the ego reaches 8.31 m/s and plans a rightward maneuver. Although the ego is already ahead, longitudinal separation remains insufficient. As the neighboring vehicle continues accelerating and the ego moves rightward, lateral clearance progressively decreases, resulting in a collision at frame 110. At this point, the neighboring vehicle is behind and to the right of the ego, traveling at approximately 6.46 m/s.

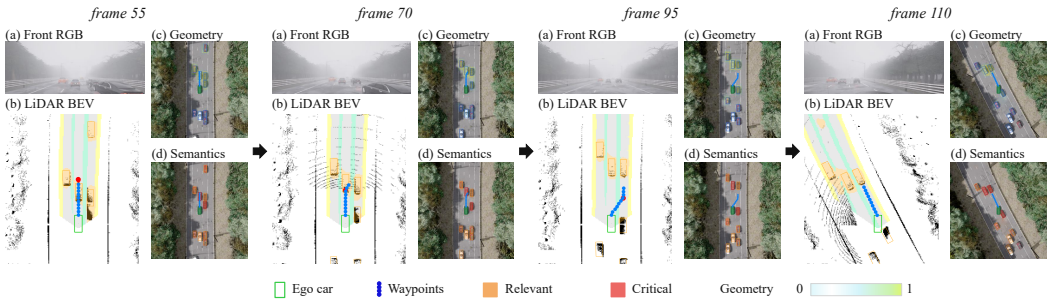


Figure 6: **Failure under rapidly changing neighboring-vehicle motion.** Frames 55, 70, 95, and 110 show front RGB, LiDAR BEV, geometry, and semantic relevance. The neighboring vehicle decelerates and subsequently accelerates again. A collision occurs at frame 110 as the ego moves rightward.

This case shows that recognizing an actor as Critical is necessary but not sufficient for safe interaction. The neighboring vehicle is highlighted as Critical, yet its stop–restart motion changes the appropriate response within a short interval: an apparent opportunity to pass becomes unsafe when the vehicle accelerates again before sufficient separation has formed. The Critical signal identifies the constraint but does not determine whether the ego should continue to wait, preserve a larger gap, or complete the pass. Improving coverage of stop–restart interactions and modeling response type and alternative motion outcomes may improve robustness in such cases, as discussed in Appendix C.4.

C.4 Limitations and Future Directions

Annotation quality, temporal consistency, and class imbalance. The offline annotation protocol promotes temporal consistency through ordered incremental queries, stable participant identities, event-triggered keyframes, and forward propagation between keyframes. Automated validators check output structure, participant identifiers, and label completeness. However, these checks cannot fully eliminate semantic judgment errors in complex interactions. We manually audit 500 routes comprising 57,260 frames and 368,725 participant observations within the training BEV. Among these frames, 4,558 frames contain incorrect semantic labels, corresponding to a frame-level error rate of 7.96%. Most errors assign *Critical* to participants judged *Relevant* by human reviewers, while genuinely *Critical* participants are rarely assigned a lower relevance category or left unlabeled. This bias may overstate response requirements and encourage unnecessary deceleration.

Critical supervision is also substantially sparser than *Relevant* supervision, both in frame coverage and occupied BEV area. The model therefore receives fewer effective examples of *Critical* interactions, especially when changes in ego state or intended maneuver shift a participant between *Relevant* and *Critical*. At inference, we occasionally observe *Critical* predictions for participants labeled *Relevant*. Such false activations can produce overly conservative speed corrections, since current *Critical* predictions directly contribute to the slowdown bias. Despite these limitations, the qualitative cases demonstrate successful coordination in challenging traffic interactions.

Future work will strengthen supervision for sparse *Critical* cases and transitions between *Relevant* and *Critical* through targeted sampling, manual verification, and temporal consistency checks along participant tracks. Class-balanced learning could improve the effective support for *Critical* interactions, while calibrated annotation confidence could provide soft targets for ambiguous transitions.

Criticality does not specify the required response. The Critical channel answers whether an actor requires a planning response, but it does not specify which response is appropriate. Braking, yielding, and avoidance are collapsed into the same spatial mask, although the annotation records these distinctions in `critical_actions`. The planner must therefore infer the response type and timing implicitly from trajectory supervision.

Figure 6 illustrates this limitation. The neighboring vehicle is identified as Critical in the semantic relevance output, so the failure is not simply caused by the actor being ignored. However, the vehicle first decelerates from 12.65 m/s to near standstill and then accelerates again. The ego begins to pass and later moves rightward before sufficient longitudinal separation has formed. A binary Critical signal does not indicate whether the ego should continue to wait, preserve a larger gap, or complete the pass before the neighboring vehicle accelerates. The rapidly changing relative motion consequently remains difficult even when the interaction is detected as planning-relevant.

A response-aware representation could preserve the annotated brake, yield, and avoid actions as separate channels or tokens and additionally predict response urgency. PCR-WM could then forecast multiple plausible motion and response outcomes rather than one evolution of a single Critical mask. Increasing training coverage of stop–restart interactions and coupling these predictions with uncertainty-dependent clearance margins are further directions for improving robustness.

D Implementation Details

D.1 Current Representation and Planner Interface

Sensor inputs and BEV representation. TransFuser++ uses RegNetY-3.2GF image/LiDAR encoders with one observation per forward pass; PCR-WM uses four separately observed frames. RGB is cropped from 1024×512 to 1024×384 pixels. The 256×256 LiDAR raster covers 64×64 m at 0.25 m per cell. Relevance receives 64-channel, 64×64 BEV features over $[-32, 32]$ m per axis, matching targets without resizing. Two normalized coordinate channels are appended. Navigation uses a BEV-normalized target point and six-way command.

Current Relevance Encoder architecture. Let C_{xy} contain normalized BEV cell coordinates. The navigation input $\mathbf{n}_t$ concatenates a 2D target point $\mathbf{p}_t^{\mathrm{nav}}$ and a 6D driving command. An MLP maps it to a 4D embedding, which is broadcast across the BEV grid and concatenated with F_t and C_{xy} before the convolutional trunk:

$$Z_t^{\mathrm{DR}} = f_{\mathrm{trunk}}([F_t, C_{\mathrm{xy}}, \mathrm{Broadcast}(\mathrm{MLP}(\mathbf{n}_t))]) \,. \tag{17}$$

The trunk comprises two 3×3 convolutional layers with $70 \to 128 \to 128$ channels, each followed by GroupNorm (32 groups) and SiLU, using stride one and padding one. It produces the 128-channel latent Z_t^{DR}. Two 1×1 prediction heads map this latent to one geometric logit and two semantic logits, which are decoded into the three-channel relevance field. The planner additionally receives ego state and navigation input, while the route decoder is conditioned on the navigation target point. The candidate plan obtained after the first refinement then conditions PCR-WM.

Current-relevance bridge. The feature bridge $\mathcal{B}$ uses adaptive average pooling, a 1×1 projection, group normalization, and SiLU. The residual scale is $\eta = 0.03$. The planner join $\mathcal{J}$ combines the resulting BEV features with ego and navigation input to produce ten route queries and one speed query of width 256:

$$Q_t = \mathcal{J}\big(F_t + \eta \mathcal{B}(Z_t^{\mathrm{DR}}), \mathbf{c}_t\big) \,. \tag{18}$$

$$\begin{aligned} \boldsymbol{\xi}_t^{(0)} &= D_{\mathrm{route}}(Q_{t,1:N}; \mathbf{p}_t^{\mathrm{nav}}), \\ \mathbf{s}_t^{(0)} &= D_{\mathrm{speed}}(Q_{t,N+1}), \end{aligned} \tag{19}$$

Here $N = 10$, and $Q_{t,1:N}$ and $Q_{t,N+1}$ denote the route and speed queries, respectively. D_{route} and D_{speed} are their decoders. The input $\mathbf{c}_t = (\mathbf{e}_t, \mathbf{n}_t)$ contains the ego state and navigation input, following the main-text notation. The ego-frame target point $\mathbf{p}_t^{\mathrm{nav}}$ is contained in $\mathbf{n}_t$ and conditions the route decoder. We write $\boldsymbol{\xi}_t^{(j)} = \{\mathbf{p}_{t,i}^{(j)}\}_{i=1}^{N}$ for the route at stage $j \in \{0, 1, 2\}$, with speed-class probabilities $\boldsymbol{\pi}_t^{(j)} = \mathrm{softmax}(\mathbf{s}_t^{(j)})$. In our configuration, the speed decoder receives only the speed query $Q_{t,N+1}$.

Table 8: **Implementation parameters.** Sensor and architecture settings follow TransFuser++; the reference experiment uses a shared 30+30-epoch protocol for current-policy learning and PCR-WM integration.

Parameter	Value
Sensor inputs and BEV	
Image input / LiDAR raster size	1024×384 / 256×256
Camera views / input timesteps	1 front view / 1
Relevance-grid range	$[-32, 32]$ m on each axis
Relevance-grid size / cell size	64×64 / 1 m
Dense BEV / coordinate channels	64 / 2
Current Relevance Encoder and planner	
Relevance trunk	Two 3×3 layers, 128 channels
Normalization / activation	GroupNorm (32 groups) / SiLU
Navigation conditioning	Planner join and candidate plan
Geometry / semantic output heads	1×1, 1 / 2 output channels
Planner feature fusion / scale	Additive bridge / 0.03 (fixed)
Refinement token grid	16×16
Attention dimension / heads	256 / 8
Route points / coordinate residual bound	10 / 1 m
PCR-WM	
History / transition layers; hidden width	1 / 2; 256
Temporal inputs / future offsets	4 / 0.5, 1.0, 1.5, 2.0 s
Conditioning / planner interface	Single timed plan / direct residuals
Tokens per horizon / attention heads	64 / 8
Future route / speed residual bounds	0.3 m / 1.5 logit units
WM GRC / occupancy / flow weights	1 / 1 / 0.1
Losses and optimization	
WM / anchor / stop weights	0.2 / 0.5 / 0.2
Current-policy auxiliary perception weights	0
Geometry Smooth-L_1 / BCE weights	1 / 1
Smooth-L_1 transition parameter	0.05
Mean semantic BCE weight	1
Route / speed / relevance weights	1 / 1 / 0.2
Residual magnitude / smoothness weights	0.02 / 0.05
Optimizer / schedule	AdamW (AMSGrad) / cosine
Stage I / Stage II epochs	30 / 30
Per-GPU batch size	4
Training time (Stage I / II)	25.9 / 17.1 h
Weight decay	0.01
H20 GPUs (Stage I / II)	16 / 24
Inference efficiency and memory	
Total / additional parameters	129.50 / 8.84 M
Inference latency	45.09 ± 1.06 ms
Throughput	22.18 frames/s
Peak inference GPU memory	1.12 GB
Peak training GPU memory	12.92 GB

Route refinement. For refinement, a convolutional projection maps $V_t Z_t^{\mathrm{DR}}$ to F_t^{cur}. The current latent and field are masked by the BEV visibility mask V_t before spatial encoding. Pooling onto a 16×16 grid gives 256-dimensional tokens. Route-conditioned queries and global attention are computed as

$$\begin{aligned} T_{t,m} &= \mathrm{LN}\Big(\mathrm{Pool}_m(F_t^{\mathrm{cur}}) + E_R(\mathrm{Pool}_m(V_t \hat{R}_t)) + E_{\mathrm{pos}}(\mathbf{q}_m^{\mathrm{grid}})\Big), \\ \widetilde{Q}_{t,i} &= \mathrm{LN}\Big(Q_{t,i} + E_{\mathrm{route}}(\mathbf{p}_{t,i}^{(0)})\Big), \\ H_t^{\mathrm{attn}} &= \mathrm{MHA}(\widetilde{Q}_t, T_t, T_t). \end{aligned} \tag{20}$$

Here i indexes route points and m indexes the 256 pooled cells. LN denotes layer normalization, Pool_m extracts the pooled feature at cell m, and $\mathbf{q}_m^{\mathrm{grid}}$ is its normalized coordinate. E_R, E_{pos}, and

E_{route} embed relevance values, grid coordinates, and route-point positions, respectively. H_t^{attn} is the eight-head attention output. Local bilinear sampling $\mathcal{S}$ complements attention; brackets concatenate features:

$$h_{t,i}^{\text{cur}} = [\widetilde{Q}_{t,i}, H_{t,i}^{\text{attn}}, \mathcal{S}(F_t^{\text{cur}}, \mathbf{p}_{t,i}^{(0)}), \mathcal{S}(V_t \hat{R}_t, \mathbf{p}_{t,i}^{(0)})]. \tag{21}$$

Point-wise gates and $\delta_{\text{cur}} = 1\,\text{m}$ bound the correction:

$$\begin{aligned} \Delta\mathbf{p}_{t,i}^{\text{cur}} &= \sigma(u_i)\delta_{\text{cur}} \tanh\big(f_{\text{cur}}(h_{t,i}^{\text{cur}})\big)\,, \\ \mathbf{p}_{t,i}^{(1)} &= \mathbf{p}_{t,i}^{(0)} + \Delta\mathbf{p}_{t,i}^{\text{cur}}. \end{aligned} \tag{22}$$

Here σ is the sigmoid function, u_i are learned gate logits, and f_{cur} is the route-offset head. The point-wise offsets form $\Delta\boldsymbol{\xi}_t^{\text{cur}} = \{\Delta\mathbf{p}_{t,i}^{\text{cur}}\}_{i=1}^{N}$ in Eq. 14.

Criticality-conditioned speed. The candidate route and two lateral offsets define a vehicle-width corridor $\mathcal{C}$. Its maximum Critical value gates a correction whose context h_t^{speed} combines the speed query, relevance-weighted pooled features, and field statistics:

$$\begin{aligned} r_t &= \max_{\mathbf{p}\in\mathcal{C}(\boldsymbol{\xi}_t^{(1)})} \mathcal{S}(V_t \hat{M}_t^{\text{crit}}, \mathbf{p}), \\ \mathbf{s}_t^{(1)} &= \mathbf{s}_t^{(0)} + \beta\sigma(u_{\text{speed}})\sigma(f_{\text{speed}}(h_t^{\text{speed}}))r_t\mathbf{d}. \end{aligned} \tag{23}$$

Here r_t is the maximum predicted Critical value in the corridor. The learned gate logit u_{speed} and scalar-output head f_{speed} scale the correction. The hyperparameter $\beta = 2.0$ bounds its magnitude in each speed logit, controlling how strongly this branch favors lower-speed classes. Vector $\mathbf{d}$ decreases linearly from 1 to -1 over low-to-high speed classes, favoring lower speeds. The additive term is $\Delta\mathbf{s}_t^{\text{cur}}$ in Eq. 14. PCR-WM then refines the candidate $(\boldsymbol{\xi}_t^{(1)}, \mathbf{s}_t^{(1)})$.

D.2 PCR-WM: Architecture and Forecast Evaluation

Figure 7 summarizes PCR-WM and its interface with the planner.

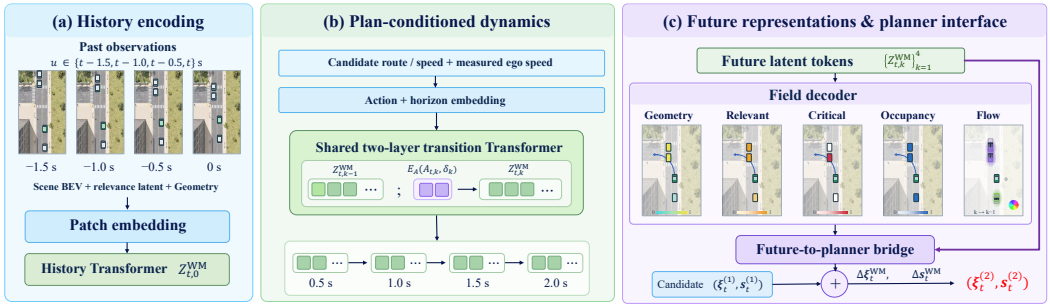


Figure 7: **PCR-WM architecture and planner integration.** Four observed BEV representations initialize the history encoder. The candidate route and target speed condition a shared two-layer Transformer rollout, whose horizon-specific latent tokens and structured fields are read by the future-to-planner bridge to correct route and speed.

Figure 8 compares predicted geometric and semantic relevance with their targets across four future horizons.

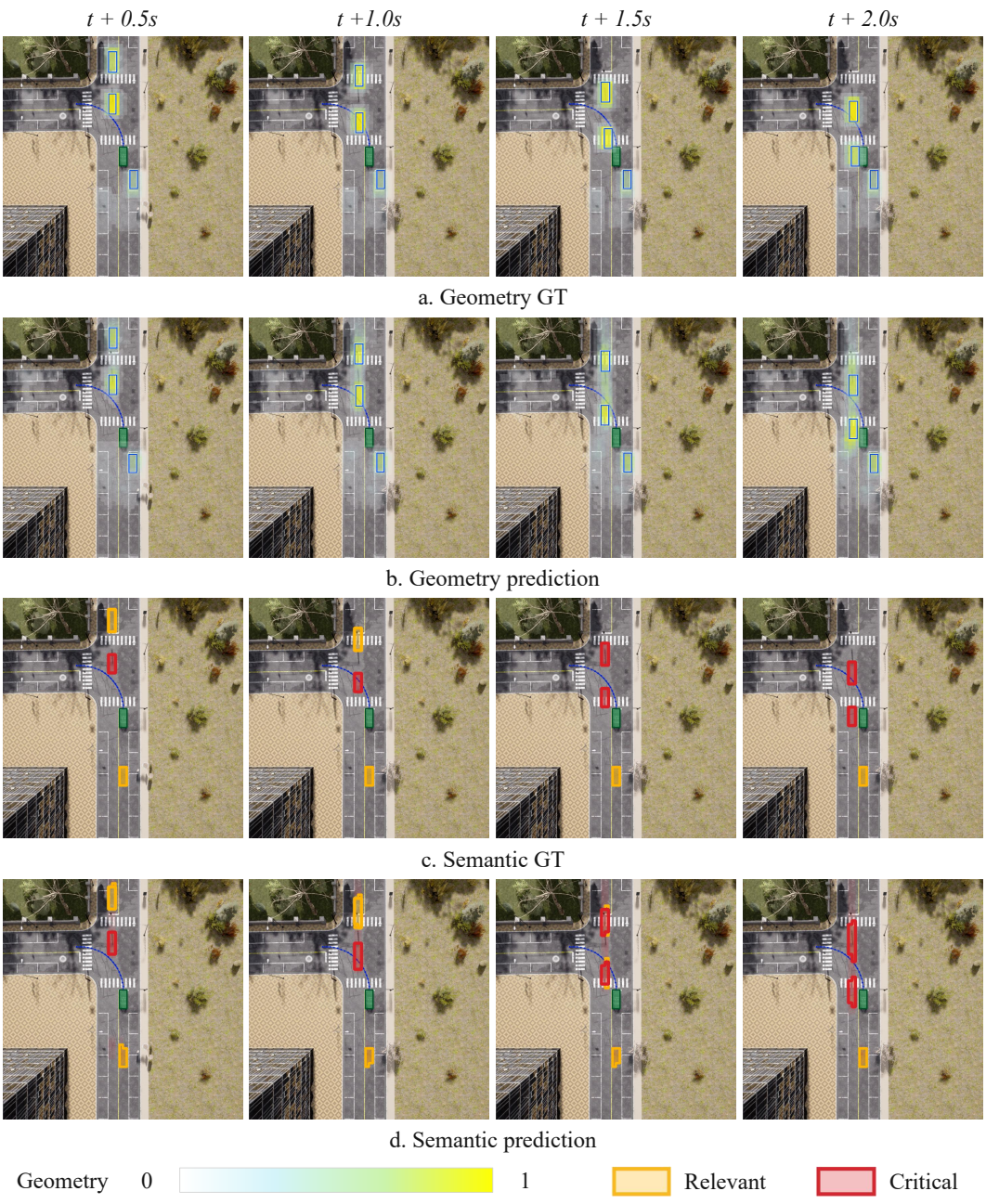


Figure 8: **Geometric and semantic relevance over a highway cut-in sequence.** Columns show 0.5, 1.0, 1.5, and 2.0 s; rows show Geometry targets/predictions and semantic targets/predictions. Geometry is continuous; orange/red denote Relevant/Critical, green denotes ego, and the blue dotted line is the route. "GT" denotes constructed labels, including teacher judgments. Critical is also Relevant but is displayed in red.

Forecast evaluation. Table 9 evaluates factual rollouts conditioned on logged ego motion on a route-disjoint held-out split. Predictions and targets share the current ego frame and channel-specific validity masks. G-MAE measures continuous Geometry error; R-AP/C-AP measure average precision for Relevant/Critical with aligned targets thresholded at 0.5. Occ-IoU thresholds predictions and targets at 0.5; Flow-EPE measures backward-flow Euclidean error in metres. Metrics pool valid cells separately at each horizon; unavailable support is not scored zero.

Table 9: **PCR-WM forecast quality.** PCR-WM is evaluated at four future horizons. Relevance and physical-state metrics are reported separately.

Horizon	G-MAE ↓	R-AP ↑	C-AP ↑	Occ-IoU ↑	Flow-EPE ↓
0.5 s	0.006799	0.8085	0.7008	0.6138	0.2637
1.0 s	0.007885	0.8102	0.7065	0.6168	0.2303
1.5 s	0.009332	0.7785	0.6868	0.6171	0.2307
2.0 s	0.010399	0.7335	0.6467	0.6044	0.2406

G-MAE increases with forecast horizon. R-AP and C-AP peak at 1.0 s (0.8102 and 0.7065), then decline to 0.7335 and 0.6467 at 2.0 s, indicating increasing forecast difficulty. Occ-IoU remains within 0.6044–0.6171 and Flow-EPE within 0.2303–0.2637. These results measure prediction quality.

History and plan-conditioned transition. Four observations at offsets $\{-1.5, -1.0, -0.5, 0\}$ s each concatenate 64 BEV, 128 current relevance, and one predicted Geometry channel on a 64×64 grid. At each observed time u, these inputs are F_u, Z_u^{DR}, and $\hat{G}_u$, respectively. Current relevance and Geometry retain the current adapter's visibility mask. An 8×8 patch projection produces 64 tokens per frame at width 256. Relative-pose and time embeddings locate historical tokens in the current ego frame; one history Transformer layer aggregates the sequence, and its final-frame tokens initialize $Z_{t,0}^{\mathrm{WM}}$. This positional encoding does not warp the feature grids.

The candidate route $\boldsymbol{\xi}_t^{(1)}$, speed probabilities $\boldsymbol{\pi}_t^{(1)}$, and measured ego speed define timed actions $A_{t,k}$. These actions encode ego pose, interval velocity, acceleration, and validity. An MLP produces the action embedding $E_A(A_{t,k}, \delta_k)$, which is concatenated with the spatial tokens. Here $\delta_k = 0.5k$ s for $k = 1, \ldots, 4$. A shared **two-layer transition Transformer** advances them at 0.5 s intervals to 2.0 s, as in Eq. 13. All Transformer layers use width 256, eight attention heads, and feed-forward width 512; transition parameters are shared across horizons.

Structured decoding. At horizon k, the semantic logits $a_{t,k}$ and $b_{t,k}$ parameterize relevance and criticality conditional on relevance:

$$\widehat{M}_{t,k}^{\mathrm{rel}} = \sigma(a_{t,k}), \qquad \widehat{M}_{t,k}^{\mathrm{crit}} = \sigma(a_{t,k})\sigma(b_{t,k}). \tag{24}$$

This parameterization ensures that predicted Critical does not exceed predicted Relevant at any grid cell.

Future decision bridge. Route/speed queries attend to 4×64 future tokens $\{Z_{t,k}^{\mathrm{WM}}\}_{k=1}^{4}$ with spatial/horizon embeddings. Local field samples use a three-offset route corridor and a nine-point footprint at each timed ego pose. Separate gated heads combine candidate context, attention, and samples; corrections are bounded by 0.3 m per route point and 1.5 per speed logit. The outputs are $\Delta\boldsymbol{\xi}_t^{\mathrm{WM}}$ and $\Delta\mathrm{s}_t^{\mathrm{WM}}$ in Eq. 15. Zero initialization preserves the candidate before future-aware optimization.

Online execution. Each control tick encodes a fresh four-observation window and rolls out the current candidate in the current ego frame. Incomplete history or fully invalid future evidence leaves the candidate unchanged. The bridge performs one correction without rerolling the final plan; forecasts therefore describe the pre-correction candidate. Factual supervision does not identify arbitrary counterfactual actor responses, and accurate Critical fields alone do not establish collision-probability calibration.

D.3 Training Details

As described in Section 3.4, current relevance first refines the base route and target speed into a candidate plan. PCR-WM then forecasts future interactions conditioned on this candidate, and the future bridge refines both outputs to obtain the final plan. Training establishes the current-relevance policy before learning the future-aware refinement. Both expert-data settings use 30 epochs for each training stage.

First stage: current-relevance refinement. Perception, current relevance, and planning are learned with relevance and expert supervision. This policy initializes the second training stage and supplies a frozen reference for decision regularization.

Second stage: future-aware refinement. The second training stage optimizes PCR-WM, the future bridge, and route/speed output heads, freezing perception, relevance, planner join, and the current adapter. Planning gradients pass through the additive candidate, native tokens, and sampled fields; candidate coordinates, interval actions, and conditioning speed probabilities are detached. The frozen reference copy provides route and speed predictions for training regularization and is not used at inference.

We use AdamW with AMSGrad, weight decay 0.01, cosine scheduling, and four samples/GPU. Task-adaptation initialization transfers only WM parameters.

Current-policy losses and weights. Geometry combines Smooth-L_1 (transition 0.05) and BCE with unit internal weights; semantic BCE averages two channels. Route loss is MAE; speed uses the base classification loss. Residual magnitude and differences between adjacent route-point residuals are penalized using MAE. The weights for the route, speed, relevance, residual-magnitude, and smoothness losses are 1, 1, 0.2, 0.02, and 0.05, respectively, and are normalized by the sum of the active loss weights. Auxiliary perception, map-alignment, and feature-scale losses have zero weight.

PCR-WM and joint losses. During training, PCR-WM predicts future states under two action sequences from the same observation history. Predictions conditioned on the candidate ego plan guide planning refinement, while those conditioned on recorded ego motion are supervised by the corresponding future labels. Recorded future motion and labels are used only during training.

Future G/R/C and occupancy targets are transformed from their future ego frames to the current ego frame using recorded poses. For backward flow, the source and destination endpoints are first transformed from their respective adjacent ego frames into this common frame; their difference is then sampled on the current grid. Spatial support, plan validity, and channel-specific label validity determine the loss masks. Invalid plan steps invalidate subsequent steps. Losses use validity masks and include the appropriate channel dimension:

$$\mathcal{L}_{\mathrm{WM}} = \mathcal{L}_{\mathrm{GRC}} + \mathcal{L}_{\mathrm{occ}} + 0.1\mathcal{L}_{\mathrm{flow}}. \tag{25}$$

Here $\mathcal{L}_{\mathrm{GRC}}$, $\mathcal{L}_{\mathrm{occ}}$, and $\mathcal{L}_{\mathrm{flow}}$ supervise the future relevance fields, occupancy, and backward flow, respectively; 0.1 is the flow-loss weight. G/R/C uses binary cross-entropy on the decoded probabilities and occupancy uses binary cross-entropy with logits, both with target-dependent weight $w_j = 1 + 4y_j^{\mathrm{field}}$. For either weighted BCE term, we use

$$\mathcal{L}_{\mathrm{wBCE}} = \frac{\sum_j m_j w_j \, \ell_{\mathrm{BCE}}(\hat{y}_j, y_j^{\mathrm{field}})}{\max(\sum_j m_j w_j, \epsilon)}. \tag{26}$$

where j indexes batch, horizon, channel, and grid cell; m_j is the validity mask. y_j^{field} and $\hat{y}_j$ are the target value and predicted probability for the corresponding Geometry, Relevant, Critical, or occupancy channel. ℓ_{BCE} is the per-element binary cross-entropy, and $\epsilon > 0$ prevents division by zero. The weight w_j increases the contribution of cells with larger target values. Occupancy uses the equivalent BCE-with-logits implementation for numerical stability. An empty valid set contributes zero. Flow uses Smooth-L_1 averaged over valid backward-flow components. Native latent tokens have no explicit future-latent matching target; they receive gradients from factual decoding and, during joint optimization, planning.

Joint route and speed losses supervise the final second-stage outputs. A frozen reference copy of the trained current policy supplies a soft anchor. Let $(\boldsymbol{\xi}_t^{\mathrm{ref}}, \boldsymbol{\pi}_t^{\mathrm{ref}})$ denote its route and speed probabilities, and $(\boldsymbol{\xi}_t^*, \mathbf{y}_t)$ the expert targets. The speed target $\mathbf{y}_t$ may be a soft distribution. Reference weights are

$$w_{r,t} = \exp\left(-\frac{N^{-1}\sum_i \|\mathbf{p}_{t,i}^{\mathrm{ref}} - \mathbf{p}_{t,i}^*\|_2}{\tau_{\mathrm{ref}}}\right), \qquad w_{s,t} = \sum_c \pi_{t,c}^{\mathrm{ref}} y_{t,c}. \tag{27}$$

Here $w_{r,t}$ and $w_{s,t}$ weight the route and speed reference constraints. i indexes the $N = 10$ route points, and c indexes speed classes. $\mathbf{p}_{t,i}^{\mathrm{ref}}$ and $\mathbf{p}_{t,i}^*$ are reference and expert route points; $\pi_{t,c}^{\mathrm{ref}}$ and

$y_{t,c}$ are the corresponding speed-class probabilities. The scale $\tau_{\mathrm{ref}} = 0.25\,\mathrm{m}$ controls how quickly the reference weight decreases as its route deviates from the expert trajectory. The final speed probabilities are $\boldsymbol{\pi}_t^{(2)} = \mathrm{softmax}(\mathbf{s}_t^{(2)})$. The anchor combines $w_{r,t}$-weighted mean absolute route deviation from the reference with $w_{s,t}\,\mathrm{KL}(\boldsymbol{\pi}_t^{\mathrm{ref}} \| \boldsymbol{\pi}_t^{(2)})$, averaged over the batch, where KL denotes Kullback–Leibler divergence. Reference predictions and weights are detached, so a reference that disagrees with expert supervision exerts a weaker constraint. Stop calibration applies binary cross-entropy to the final stop logit $s_{t,0}^{(2)} - \mathrm{logsumexp}_{c>0}\, s_{t,c}^{(2)}$, with target $y_{t,0}$, where class $c = 0$ denotes stopping and $c > 0$ denotes moving speed classes. Together with the retained current-policy terms $\mathcal{L}_{\mathrm{cur}}$, these form the regularization in Eq. 16:

$$\mathcal{L}_{\mathrm{reg}} = \lambda_{\mathrm{anchor}}\mathcal{L}_{\mathrm{anchor}} + \lambda_{\mathrm{stop}}\mathcal{L}_{\mathrm{stop}} + \mathcal{L}_{\mathrm{cur}}. \tag{28}$$

Here $\mathcal{L}_{\mathrm{anchor}}$ is the reference constraint described above, and $\mathcal{L}_{\mathrm{stop}}$ is the stop-calibration loss. $\mathcal{L}_{\mathrm{cur}}$ collects the weighted current-relevance, residual-magnitude, and smoothness losses. The coefficients $\lambda_{\mathrm{anchor}} = 0.5$ and $\lambda_{\mathrm{stop}} = 0.2$ set the relative strengths of the reference and stop constraints. The weights for the route, speed, WM, anchor, and stop losses are 1, 1, 0.2, 0.5, and 0.2, respectively, before active-loss normalization. The retained current-relevance, residual-magnitude, and smoothness losses have weights of 0.2, 0.02, and 0.05, respectively. Frozen relevance contributes no parameter gradient, but route-dependent regularizers may constrain trainable output heads. WM-only adaptation optimizes $\mathcal{L}_{\mathrm{WM}}$ alone.

D.4 Computational Cost

Inference efficiency and resource usage. DriveMatter has 129.50 M parameters, 8.84 M more than TransFuser++ without the relevance modules. The current-relevance path accounts for 2.11 M of these additional parameters, while PCR-WM and its future bridge add 6.74 M over the current-only model. The two 30-epoch stages take 25.9 hours on 16 NVIDIA H20 GPUs and 17.1 hours on 24 H20 GPUs, respectively, for 43.0 hours in total.

We measure neural-network inference with batch size one on a single H20 in FP32 with TF32 enabled. A real Bench2Drive sample and the complete online history are used. After 110 warm-up iterations, five runs of 200 forward passes give 45.09 ± 1.06 ms per frame, or 22.18 frames per second. The timing includes current relevance, both planning corrections, and PCR-WM, but excludes data loading, simulation, and the CPU controller. Peak allocated inference memory is 1.12 GB. The maximum device memory recorded during training is 12.92 GB per GPU with four samples per GPU.

E Relevance label construction

The geometric channel in Section 3.1 is constructed offline from PDM-Lite actor annotations and the navigation route stored at each frame. It provides a continuous measure of route-conditioned spatial coupling. Future annotations are used only to construct the training target; the resulting score is an observational pseudo-label, rather than a causal effect or calibrated collision probability. Figure 9 illustrates the resulting labels in three interactions. Figure 8 compares geometric and semantic targets with future predictions, showing the distinct information carried by the channels.

E.1 Geometric score and actor selection

Geometry and actor set. At frame u, let $B_{i,u}$ denote the oriented two-dimensional footprint of actor i in the ego-centered BEV coordinate system (x forward and y right), and let Γ_u denote the native route polyline stored in the PDM-Lite measurement. We retain physical actors of classes `car`, `walker`, `static`, and `static_prop_car`; the ego vehicle, traffic lights, traffic signs, and other non-physical infrastructure are excluded. Actor class is used only for filtering and temporal identity consistency and does not otherwise change the geometric score.

For Eq. 5, $d_{i,u}^{r}$ is the minimum Euclidean distance from the oriented footprint $B_{i,u}$ to the finite route polyline Γ_u, with intersecting geometries having zero distance. The signed longitudinal coordinate $s_{i,u}$ is the actor-center arc-length position relative to the projection of the ego origin onto the route. The recorded route is used at its native extent: consecutive duplicate points are removed, but the



(a) Unprotected left turn (b) Parking exit / cut-in (c) Pedestrian crossing

Figure 9: **Future-aware geometric labels in three interaction scenarios.** The three groups show (a) an unprotected left turn, (b) a parking-exit/cut-in interaction, and (c) a pedestrian crossing. Each group contains four frames arranged chronologically from left to right and top to bottom. Green marks the ego vehicle, blue dots indicate the navigation route, and colors represent geometric relevance on a shared $[0, 1]$ scale. Scores incorporate current and short-term future spatial relations and are rasterized around each participant's current footprint.

route geometry is neither resampled nor extended. Endpoint tangents are used only when evaluating the signed arc-length coordinate beyond the ends of the finite polyline. For Eq. 6, $d^e_{i,u}$ is the minimum distance from the actor footprint to the ego origin $(0, 0)$, rather than the clearance between the actor and the ego-vehicle body.

Parameters. We use

$$\sigma_r = 4\,\mathrm{m}, \qquad \tau_s = 2\,\mathrm{m}, \qquad \rho = 0.08, \qquad \alpha_e = 0.85, \qquad \sigma_e = 6\,\mathrm{m}.$$

The route component therefore favors actors close to and ahead along the intended route, while the ego component retains nearby actors that may be weakly aligned with that route. The two components are combined by the maximum in Eq. 6. We do not introduce class-dependent weights, actor speed, time-to-collision, or an additional score threshold.

E.2 Future-aware temporal aggregation

PDM-Lite annotations are saved at $4\,\mathrm{Hz}$. For an anchor frame t, we evaluate the geometric score over

$$\mathcal{H}_g = \{0, 0.5, 1.0, 1.5, 2.0\}\,\mathrm{s},$$

corresponding to saved-frame offsets $\{0, 2, 4, 6, 8\}$. For each actor present at the anchor frame, we apply Eq. 7 with $\lambda_g = 0.15\,\mathrm{s}^{-1}$, where $\mathcal{H}_{i,t} \subseteq \mathcal{H}_g$ is the valid temporal prefix of actor i at anchor time t. Each $g_{i,t+h}$ is recomputed using the actor footprint and native route in the ego coordinate system of frame $t + h$. Consequently, temporal aggregation is applied only to actor-level scalars; BEV fields from different future ego frames are never combined directly. The exponential discount is applied before the maximum, and ties are assigned to the earlier horizon. Since $h = 0$ is included, $\bar{g}_{i,t} \geq g_{i,t}$.

Track association and valid anchors. Actors are associated by stable identifier within each route. Temporal aggregation is restricted to actors already present at t; actors that first appear in the future are not projected back into the current target. The first missing sampled observation or a `class`/`type_id` change terminates the actor's temporal prefix, and a later reappearance is not reconnected. We do not use nearest-neighbor association or trajectory interpolation. This gap check is performed at the sampled $0.5\,\mathrm{s}$ horizons rather than at every intervening $0.25\,\mathrm{s}$ saved frame.

An actor-level track termination only shortens $\mathcal{H}_{i,t}$ and does not invalidate the complete anchor. At the frame level, however, we require measurement and box annotations at all five source offsets $\{0, 2, 4, 6, 8\}$, and all corresponding snapshots must be successfully constructed. Frames missing measurements or actor annotations at any required timestamp within the $2\,\mathrm{s}$ horizon are excluded from geometric supervision, rather than assigned zero-valued targets.

E.3 Dense rasterization and label validity

After temporal aggregation, $\bar{g}_{i,t}$ is rasterized around the actor's *current* footprint $B_{i,t}$. Let $d_{\text{out}}(\mathbf{p}, B)$ denote the Euclidean distance from the BEV cell center $\mathbf{p}$ to the actor footprint B, with zero distance inside the footprint. We use the compact soft kernel

$$K(B, \mathbf{p}) = \begin{cases} 1, & \mathbf{p} \in B, \\ \exp\left(-\dfrac{d_{\text{out}}(\mathbf{p}, B)^2}{2\sigma_b^2}\right), & 0 < d_{\text{out}}(\mathbf{p}, B) \le b, \\ 0, & d_{\text{out}}(\mathbf{p}, B) > b, \end{cases} \tag{29}$$

with $\sigma_b = 0.75\,\text{m}$ and $b = 2.25\,\text{m}$. The geometric field is then

$$G_t(\mathbf{p}) = \max_{i \in \mathcal{A}_t} \bar{g}_{i,t}\, K(B_{i,t}, \mathbf{p}), \tag{30}$$

where $\mathcal{A}_t$ is the set of eligible physical actors at the anchor frame. Actor overlaps are resolved by a pixelwise maximum. Only current footprints are rasterized: future actor boxes, trajectories, and future BEV fields do not appear directly in G_t. The navigation route conditions the actor-level score but is not itself rasterized into the geometric target. For a small actor whose footprint contains no cell center, the cell containing its center is used as a fallback when that center lies inside the grid.

The PDM-Lite target covers $x, y \in [-32, 32)\,\text{m}$ at $1\,\text{m}$ resolution, producing a 64×64 field. Cell centers are sampled on this metric grid. The cached array is stored in (X, Y) order and is transposed once by the data loader to the model layout $(C, Y, X) = (1, 64, 64)$. No additional distance threshold is applied before scoring. Actors whose centers lie outside the BEV grid can still contribute where their footprints or surrounding soft-label regions overlap the grid.

Geometric labels are stored in the original ego frame. Spatial augmentation applied to sensor inputs must also be applied to the labels to preserve alignment. Target validity is represented explicitly: a valid frame may contain no eligible actor and hence have a genuine all-zero field, whereas a frame without the required temporal context is invalid and is excluded from the geometry loss. An all-zero field must therefore not be used to infer label validity.

E.4 Interpreting the qualitative examples

Across the three scenarios in Figure 9, geometric relevance reflects each participant's spatial relation to the ego vehicle and its intended route, incorporating both current and short-term future positions. The target at each displayed frame summarizes future actor-level coupling but remains anchored to current footprints. Thus, a high geometric score indicates spatial coupling within the labeling horizon; it does not by itself specify whether the ego should brake, yield, or steer.

The semantic channels supply this complementary distinction in Fig. 8. Relevant identifies actors to monitor, whereas Critical identifies those judged to require a planning response. The prediction rows retain prominent concentrations of relevance along the route but also show differences in spatial extent and semantic assignment relative to the targets. This example illustrates the forecast representation and its errors; it does not establish a quantitative accuracy gain or the benefit of the second planning correction. Those questions require forecast metrics and closed-loop ablations, respectively.

F Semantic teacher

The semantic teacher in Section 3.1 assigns each physical actor a three-class interaction label. It receives structured text only: no images, no LiDAR, and no geometric score.

F.1 Construction pipeline and keyframes

Pipeline. For each route we (i) collect physically volumetric actors and exclude traffic lights as well as oversized unnamed static objects; (ii) stitch future trajectories in the current ego frame using stable actor identities; (iii) select keyframes; (iv) query the teacher in temporal order, with a full-frame annotation on the first keyframe and incremental annotations thereafter; (v) parse the output,

retry and fill missing labels if needed, and write the result into a running state; (vi) forward-fill class labels to every saved frame u and hard-rasterize the current oriented boxes into M_u^{rel} and M_u^{crit}; (vii) write the route pack.

Track cuts and forward fill. A future track is cut at the first missing frame or when `class`/`type` changes. If the same actor appears again, it is treated as a new track and is not joined to the track before the break. On each saved frame, the class is taken from the most recent keyframe teacher output, while the raster is filled from that frame's current boxes rather than by reusing the previous mask. Actors listed as `forgotten` in an incremental response are removed from the state and receive no labels on later frames.

Keyframe selection. The teacher is queried only at keyframes. A saved frame is selected if any of the following conditions holds: it is the first frame of the route; the discrete navigation command differs from that at the previous saved frame; ego speed increases by more than 2 m/s or decreases by at least 1.5 m/s relative to the previous saved frame; a construction or accident hazard is first observed within 40 m or 20 m, or subsequently crosses either threshold; ego decelerates while a hazard is within 40 m; the ego–actor center distance computed from logged future trajectories falls below 3 m along the stitched future track; the Jaccard distance between the current actor-ID set and that of the last keyframe exceeds 0.30; or ten saved frames have elapsed since the last keyframe. The hazard-related deceleration and future-distance conditions require at least three saved frames since the last keyframe. The 3 m threshold matches `POSSIBLE_COLLISION`. The `close_crossing` tag uses a 6 m threshold and provides additional evidence to the teacher, but does not independently trigger a keyframe.

F.2 Privileged inputs and three-class labels

Actor set and serialized evidence. Each frame provides the ego state, an ego-frame route polyline, a discrete navigation command, and oriented 3D bounding boxes in the current ego frame (x forward, y right). Table 10 summarizes the input-field mappings for each dataset. Roadside parked vehicles are retained as `car` instances rather than excluded by the static-object filter. Boxes with non-finite positions, orientations, or dimensions, or with non-positive dimensions, are discarded; the remaining frame data are still processed.

Table 10: **Teacher-input adapters for the two expert logs.** The irrelevant / relevant / critical contract is shared. Boxes in both logs are already in the current ego frame. Semantic BEV resolution is chosen by the downstream model (LEAD planning 80×96, or TransFuser-v5 64×64), not by the expert.

Field	PDM-Lite	LEAD
Lane identifiers	No `road_id` or `lane_id`; lane relation uses geometric proxies	`road_id` and `lane_id` are stored on ego and actor boxes
Future tracks	No stored futures; tracks are stitched with `ego_matrix` on the 0.25 s dump	`future_positions` are stored in the current ego frame
Untyped statics	Nameless `class=static` boxes are kept only if the long side is at most 3.5 m	Assets carry a type; the unnamed-static size gate is not applied
Construction cues	Hazards use on-corridor geometry	Privileged construction / accident distances are available in the meta when the scenario provides them

Each keyframe is serialized as structured text: ego motion and navigation commands, a privileged ground-truth ego-motion summary, and, for each actor, identifier, category, pose, size, speed, lane and route relations, a trajectory-conflict tag (future center distance < 3 m: `POSSIBLE_COLLISION`; < 6 m: `close_crossing`), a cut-in indicator, and time-aligned samples of logged future geometry in the current ego frame. The future is available only to the teacher.

Decision rule and interpretation. We guide the teacher with the counterfactual question of whether the ego could maintain its current lane and speed if a given actor disappeared. The resulting labels are teacher judgments rather than measured causal effects, because the logged expert trajectory may already reflect a response to that actor.

Label definitions. Table 11 summarizes the three classes. Labels are obtained directly from the parsed model responses, without additional rule-based overrides. The following guidelines are included in the prompt rather than enforced through post-processing.

Table 11: **Semantic teacher classes.** Proximity alone does not promote an actor to critical.

Class	Definition	Critical actions
irrelevant	No remaining effect on the ego plan. Being outside the logged expert corridor alone is insufficient.	empty
relevant	The actor is still watched, but the current plan remains executable without an immediate response.	empty
critical	The actor already forces a response, occupies or will occupy the original lane, a conflict zone, the target-lane gap, or a borrowed oncoming lane.	at least one

Cones, wrecks, and stopped vehicles obstructing the lane the ego would otherwise maintain are labeled Critical upon first observation. A same-lane lead vehicle that is decelerating, moving slowly, or approaching a stop is also Critical, including at distances of 20–40 m. At intersections, the nearest unoccluded vehicle on each approach is at least Relevant and becomes Critical when it occupies the conflict region or accelerates into the ego vehicle's turning path.

Queued vehicles occluded by a nearer vehicle along the same line of interaction are treated as Irrelevant. An adjacent parked vehicle remains Relevant while within its parking space and becomes Critical when it enters the ego lane. A vehicle already occupying the intended gap during a lane change, ramp entry, or merge is Critical from the start of the maneuver. After an interaction ends, the participant remains Relevant briefly before being reclassified as Irrelevant.

These judgments account for participants' physical footprints, including the ego vehicle's approximate dimensions of 4.8×1.9 m, rather than relying on center distance alone.

F.3 Teacher interface and rasterization

Output schema. The first keyframe returns one record per current actor, with `semantic_label`, `critical_actions`, and `reason`; `critical_actions` must be empty for irrelevant and relevant, and every current identifier must appear exactly once. Later keyframes use four top-level keys:

```
{ "unchanged": ["3696"],
  "updated": [ {"id": "3716", "semantic_label": "relevant",
      "critical_actions": [], "reason": "..."} ],
  "new_actors": [ {"id": "3801", "semantic_label": "relevant",
      "critical_actions": [], "reason": "..."} ],
  "forgotten": ["3702"] }
```

`confidence` is stored and unused at rasterization.

Rasterization. For a BEV cell $\mathbf{p}$, the two semantic masks are

$$M_t^{\mathrm{rel}}(\mathbf{p}) = \max_{i:\, \ell_{i,t} \in \{\mathrm{R},\mathrm{C}\}} \mathbf{1}_{\mathbf{p} \in B_{i,t}}, \qquad M_t^{\mathrm{crit}}(\mathbf{p}) = \max_{i:\, \ell_{i,t} = \mathrm{C}} \mathbf{1}_{\mathbf{p} \in B_{i,t}}. \tag{31}$$

Empty maxima are zero, and a Critical actor is therefore written to both masks. A box that covers no cell center, including a sub-cell cone, lights the cell that contains the box center. Actors with `training_eligible=false` are omitted from the raster; production incremental labels set this flag to true. Training reads the sidecar by route key and frame index and does not reassemble labels online by actor id. The raster grid follows the downstream model (Table 10).

F.4 How the labels are assigned

Figure 10 shows the offline teacher output after the protocol in Section F.1, not a network prediction.

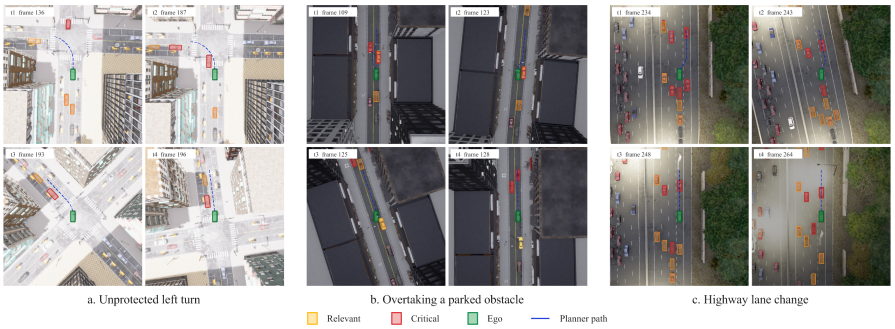


Figure 10: Teacher-generated semantic labels in three driving interactions: (a) an unprotected left turn, (b) overtaking a parked obstacle, and (c) a highway lane change. Each group is four saved frames in row-major temporal order (indices $136, 187, 193, 196$ in (a); $109, 123, 125, 128$ in (b); $234, 243, 248, 264$ in (c)). Orange and red mark relevant and critical actors; green is the ego. A critical actor is written on both semantic masks, but shown only in red. The blue path is the logged expert route. An unboxed vehicle is not a verified irrelevant label. The frames are qualitative snapshots, not four teacher queries; the boxes are actor footprints, not mask contours.

Interaction-dependent labels. In Figure 10(a), the critical set highlights traffic encountered during the left turn and along the outgoing lane, while other nearby actors remain relevant. The changing ego heading also changes the view of the intersection across the four frames. In Figure 10(b), the expert route bends around the parked obstruction; the red and orange boxes distinguish the currently critical interactions from actors that remain relevant as the ego passes the obstruction. A path that already avoids an obstacle is not evidence that the obstacle was irrelevant to planning. In Figure 10(c), critical vehicles occur both ahead of and behind the ego during the lane-change interaction, whereas other traffic is marked relevant. This illustrates why semantic importance depends on the maneuver and the surrounding traffic configuration rather than on forward distance alone.

F.5 Teacher model and labeling cost

Model and API. Labels are generated using Qwen3.8-Flash (`qwen3.8-flash`) through the DashScope API. We use temperature 0, disable thinking, and limit each response to 4096 tokens. Requests have a 120 s timeout and allow up to six transport retries. If a response omits no more than twelve participant identifiers, we issue one follow-up query for the missing labels. Any labels still missing after this query are assigned Irrelevant, and the route is retained.

Annotation of the 5368-route PDM-Lite corpus required one query per keyframe and approximately 4.3×10^3 additional queries for missing labels, consuming approximately 1.9×10^9 input tokens and 2.0×10^7 output tokens. The process completed in 5.4 hours using 100 parallel workers. Table 12 compares the query counts.

Table 12: **LLM annotation query counts on PDM-Lite.** We compare labeling every saved frame with our keyframe-based incremental annotation protocol. Counts exclude retries and missing-label follow-up queries. The query ratio is normalized by the per-frame baseline.

Protocol	Queries	Output	Query ratio
Every saved frame, full JSON	6.96×10^5	full `actors` list	1.00
Keyframes + incremental (ours)	1.53×10^5	full list, then incremental updates	0.22