PHLEX: PHYSICALLY LAYERED EXPLANATION FOR COUNTERFACTUAL MOTION PLANNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Predicted trajectories describe where an agent may move, but provide limited insight into the physical mechanisms behind that motion and are difficult to systematically vary for counterfactual reasoning. We introduce PHLEX (PHysically Layered EXplanation), a predictor-agnostic framework that maps predicted motion into a structured physical mechanism representation. PHLEX progressively explains each trajectory through kinematic models, longitudinal and lateral control laws, and scene references. Together, these layers form a structured physical explanation of how the predicted motion is generated and grounded in the scene. These explanations can be executed and varied to generate physically grounded counterfactual futures, which are further used to guide ego planning. This connects the flexibility of learned trajectory prediction with explicit physical interpretation and controllable what-if reasoning. Experiments across multiple predictors and datasets show that PHLEX improves physical feasibility and planning safety while preserving trajectory accuracy.

1 INTRODUCTION

Motion forecasting supports planning in autonomous driving by predicting how surrounding agents may move. To capture uncertainty in future behavior, multimodal predictors use motion histories and road context to generate multiple possible trajectories and their associated probabilities (Chai et al., 2020; Shi et al., 2022; Zhou et al., 2023; Nayakanti et al., 2023). To support reliable trajectory planning in the real world, predicted trajectories must be both accurate and physically feasible. However, traditional deep-learning-based predictors learn future positions from data without explicitly enforcing kinematic constraints. As a result, predicted trajectories may require acceleration or turning rates beyond the agent's physical limits (Cui et al., 2020). This motivates combining learned forecasts with explicit physical models, both to enforce physical constraints and to explain the predicted motion.

In transportation, kinematic models explicitly describe how control inputs produce physically plausible motion, but they do not determine how those controls should evolve from complex scene context. They therefore rely on predefined control or behavioral assumptions, which limits their ability to represent the diverse, context-dependent future behaviors captured by data-driven predictors (Cui et al., 2020). Hybrid methods combine learned predictions with explicit physical models. Deep Kinematic Models and Trajectron++ predict control inputs and integrate them through kinematic or agent-specific dynamics models (Cui et al., 2020; Salzmann et al., 2020). CoverNet instead learns to select from trajectory sets that can be constructed using vehicle dynamics (Phan-Minh et al., 2020). These approaches introduce physical structure into trajectory generation, but the resulting forecasts are not explicitly decomposed into interpretable physical mechanisms. This makes it difficult to identify which mechanisms explain a prediction or to systematically vary them to generate alternative futures for counterfactual reasoning.

Therefore, we introduce PHLEX, which explains predicted trajectories through *progressive physical disambiguation*. For each short trajectory segment, PHLEX progressively evaluates three mechanism layers: kinematic models, longitudinal and lateral control laws, and scene references. Each layer checks how well its candidate mechanisms reproduce the original prediction. These explanations can then be executed and varied to generate different counterfactual futures. The resulting mechanism evidence, counterfactual futures, and interaction risks are integrated over time and pro-

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.

cessed by a lightweight neural network, which learns how these physically grounded futures could refine the host ego trajectory. PHLEX therefore adds explicit physical reasoning and counterfactual analysis to an existing predictor without changing its architecture. Its contributions are:

- We introduce a physically layered explanation framework that decomposes predicted trajectories into kinematic models, control laws, and scene-grounded references while preserving multiple compatible explanations over time.
- PHLEX makes predicted motion physically interpretable and allows compatible explanations to be modified and re-executed, generating counterfactual ego and surrounding-agent futures for explicit counterfactual analysis.
- PHLEX is a predictor-agnostic framework that can be integrated with different trajectory prediction and planning models without modifying their internal architectures, with cross-host experiments demonstrating its broad applicability.

2 Related Work

Multimodal motion forecasting. Motion forecasters represent uncertain behavior through multiple future trajectories and associated probabilities. MultiPath uses trajectory anchors, while MultiPath++ improves context encoding and trajectory aggregation (Chai et al., 2020; Varadarajan et al., 2022). Trajectron++ and CDSTraj model multimodal futures through latent variables and diffusion, respectively (Salzmann et al., 2020; Liao et al., 2024a). Transformer-based predictors encode agent interactions and road context (Shi et al., 2022; Zhou et al., 2023; Nayakanti et al., 2023). MotionLM formulates joint prediction as autoregressive modeling over discrete motion tokens (Seff et al., 2023), and UniAD connects forecasting with driving decisions (Hu et al., 2023b). These methods model possible futures, but their output trajectories do not by themselves specify the physical control laws that explain the motion. PHLEX uses these forecasts as inputs to a mechanism representation that guides ego planning.

Physically grounded motion representations. Physical knowledge has been incorporated into motion forecasting through dynamics-based decoding, constrained trajectory generation, and structured motion models. Deep Kinematic Models embed vehicle kinematics in decoding, while Trajectron++ integrates agent-specific dynamics (Cui et al., 2020; Salzmann et al., 2020). CoverNet selects from dynamically feasible trajectory sets, PRIME combines constrained generation with learned evaluation, and PCMP uses surrogate dynamics for constrained forecasting (Phan-Minh et al., 2020; Song et al., 2022; Tumu et al., 2023). Physics-informed reconstruction and rule-based safety checks further improve physical feasibility (Liao et al., 2024b; Vitelli et al., 2022). Beyond feasibility, dynamical and probabilistic movement primitives provide structured representations of motion, while traffic models such as IDM describe behavior through explicit control laws (Ijspeert et al., 2013; Treiber et al., 2000). PHLEX builds on this direction by organizing predicted motion through body kinematics, longitudinal and lateral control laws, and scene references, while retaining multiple compatible explanations and supplying them to a neural planner.

World models and controlled rollouts. World models predict future states to support behavior learning and action evaluation. Dreamer learns behaviors through imagined latent trajectories (Hafner et al., 2020). Driving world models generate future videos (Hu et al., 2023a; Wang et al., 2024; Gao et al., 2024) or predict semantic occupancy and ego motion (Zheng et al., 2024). World4Drive and WoTE use predicted latent and bird's-eye-view states, respectively, to evaluate planning trajectories (Zheng et al., 2025; Li et al., 2025). PHLEX performs controlled rollouts in an explicit mechanism space. Changing a control law or scene reference produces an alternative future, and a learned head uses these futures to refine the ego plan. These interventions support what-if analysis without claiming to identify an agent's underlying causal decision process.

3 Problem Formulation

For each agent $i \in \{1, \ldots, N\}$, a host predictor provides K future trajectories and their probabilities:

$$\mathcal{Y}_i = \{(\boldsymbol{\tau}_{i,m}, \pi_{i,m})\}_{m=1}^{K}, \qquad \boldsymbol{\tau}_{i,m} = (\mathbf{p}_{i,m,1}, \ldots, \mathbf{p}_{i,m,T}), \tag{1}$$

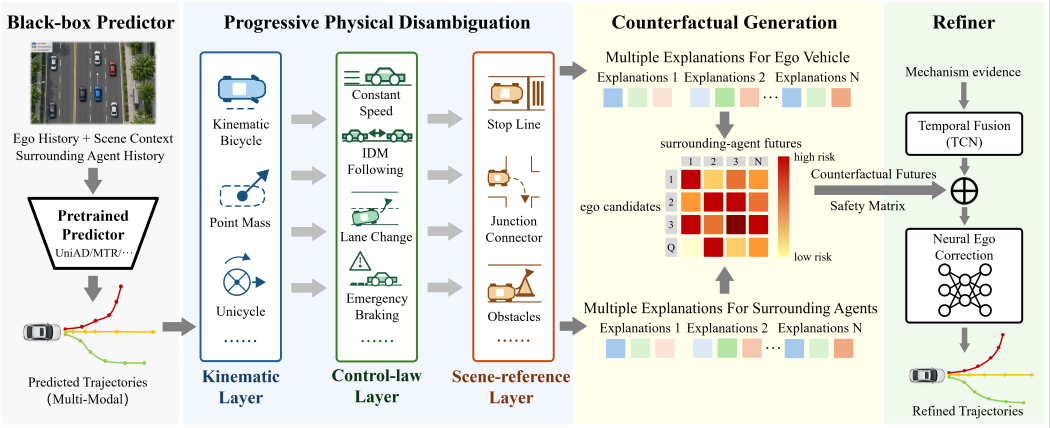

Figure 1: **Overview of PHLEX.** PHLEX decomposes predicted trajectories into layered physical explanations and executes them to generate counterfactual futures. Refiner uses the resulting mechanism and risk evidence to refine the ego trajectory.

where $\mathbf{p}_{i,m,t} \in \mathbb{R}^2$, $\pi_{i,m} \geq 0$, and $\sum_m \pi_{i,m} = 1$. We also use the host ego reference $\boldsymbol{\tau}_0^{\text{ego}}$, observed agent states and classes, and available scene context $\mathcal{C}$.

Given these inputs, PHLEX first maps each predicted trajectory, including the ego reference, to a set of compatible physical explanations and evaluates the evidence supporting them. These explanations are then executed and varied to construct alternative ego and surrounding-agent futures, from which their interactions and potential risks can be evaluated. The resulting mechanism and interaction evidence is integrated by Refiner to predict a residual correction $\delta\boldsymbol{\tau}_\theta$ to the host ego reference. The final output is

$$\hat{\boldsymbol{\tau}}^{\text{ego}} = \boldsymbol{\tau}_0^{\text{ego}} + \delta\boldsymbol{\tau}_\theta, \tag{2}$$

where θ denotes the learned parameters.

4 PHLEX: Physically Layered Explanation

PHLEX decomposes each predicted trajectory into a set of compatible mechanism explanations, executes these explanations, and uses their evidence to guide a learned ego correction. The physical library supplies alternatives and reconstruction errors. Refiner learns how those alternatives should influence the final trajectory.

4.1 Progressive Physical Disambiguation

We disambiguate each predicted trajectory into three mechanism layers: a kinematic layer, a control-law layer, and a scene-reference layer. The kinematic layer evaluates how well different kinematic models can reproduce the predicted motion under physical constraints. The control-law layer further explains the recovered controls using longitudinal and lateral control laws. The scene-reference layer connects these control laws to relevant agents and map elements in the scene. Together, the three layers progressively build a physical explanation.

We divide the prediction horizon into W consecutive short segments, with time indices $\mathcal{T}_w$ for segment w. For each segment, the three layers evaluate multiple compatible mechanisms rather than committing to a single explanation. The resulting physical explanation is represented by layer-wise mechanism evidence,

$$\mathcal{E}_w = \left\{\boldsymbol{\alpha}_w^{\text{kin}}, \boldsymbol{\alpha}_w^{\text{lon}}, \boldsymbol{\alpha}_w^{\text{lat}}, \boldsymbol{\alpha}_w^{\text{scene}}\right\}, \qquad \mathcal{E} = \{\mathcal{E}_w\}_{w=1}^{W}, \tag{3}$$

where $\boldsymbol{\alpha}_w^\ell$ contains the weights of the candidate mechanisms at layer ℓ. The weights are derived from how well complete mechanism combinations explain and reconstruct the predicted motion, as detailed in Sec. 4.2.

Table 1: Candidate mechanisms and scene references. Context-dependent laws are instantiated for each compatible reference. Appendix A gives a detailed explanation of every candidate and its bounds.

Layer	Candidates
Kinematic	Point-mass, unicycle, and kinematic bicycle models, with reverse motion for vehicles
Control-law	*Longitudinal:* constant speed (including standstill), constant acceleration, target-speed tracking, IDM following, target-position stopping, emergency braking *Lateral:* zero curvature, constant curvature, clothoid curvature profile, lane-change profile, lane-keeping
Scene-reference	Leading agent, conflicting agent or conflict point, stop line, current/target lane, junction connector, obstacle

Kinematic layer. The kinematic layer evaluates how well each candidate kinematic model can physically reproduce the host-predicted trajectory. For each model b applicable to the agent type, we first recover the control sequence required to follow the original prediction:

$$u^b = g_b(\boldsymbol{\tau}, s_0; \theta_b), \tag{4}$$

where b indexes the candidate kinematic model, $\boldsymbol{\tau} = (\mathbf{p}_1, \ldots, \mathbf{p}_T)$ is the predicted trajectory being explained, and s_0 is the observed initial state. $u^b = \{u_t^b\}_t$ denotes the recovered control sequence, g_b is the model-specific inverse mapping from trajectory to controls, and θ_b contains the fixed parameters of model b. The recovered controls may exceed the physical limits of the model. We therefore constrain them and roll the model forward from the observed initial state:

$$\tilde{u}_t^b = \mathcal{P}_b(u_t^b, \hat{s}_t^b, \tilde{u}_{t-1}^b), \qquad \hat{s}_{t+1}^b = f_b(\hat{s}_t^b, \tilde{u}_t^b; \theta_b), \qquad \hat{s}_0^b = s_0. \tag{5}$$

Here, t denotes the time step, u_t^b is the recovered control at step t, and $\tilde{u}_t^b$ is its physically constrained version. $\hat{s}_t^b$ is the state obtained by rolling out model b, $\mathcal{P}_b$ enforces its control and rate limits, and f_b is the corresponding forward kinematic model. We then measure how closely the resulting rollout matches the original prediction in each segment:

$$e_{w,b}^{\text{kin}} = \frac{1}{|\mathcal{T}_w|} \sum_{t \in \mathcal{T}_w} \|\hat{\mathbf{p}}_t^b - \mathbf{p}_t\|_2^2. \tag{6}$$

Here, $\hat{\mathbf{p}}_t^b$ is the position component of the rolled-out state $\hat{s}_t^b$, $\mathbf{p}_t$ is the original predicted position, and $|\mathcal{T}_w|$ is the number of time steps in segment w. Thus, $e_{w,b}^{\text{kin}}$ is the segment-wise reconstruction error for model b. A smaller value indicates that the model can reproduce the predicted motion more closely under its physical constraints.

Control-law layer. The kinematic layer determines the controls needed to reproduce the predicted trajectory under each candidate model. The control-law layer further explains these controls using interpretable longitudinal and lateral control laws. For each kinematic model b, we fit candidate longitudinal laws to its acceleration a_t^b and candidate lateral laws to its curvature κ_t^b. Their fitting error in segment w is

$$e_{w,c}^{\ell} = \frac{1}{|\Omega_w^{\ell}|} \sum_{t \in \Omega_w^{\ell}} \left| y_t^{b(c),\ell} - \hat{y}_t^{c,\ell} \right|^2, \quad \ell \in \{\text{lon}, \text{lat}\}, \quad y^{b,\text{lon}} = a^b,\ y^{b,\text{lat}} = \kappa^b, \tag{7}$$

where c is a complete explanation, $b(c)$ is its kinematic model, ϕ_c is the fitted parameters of its control laws, and $\hat{y}_t^{c,\ell}$ is the control produced by its layer-ℓ law under ϕ_c. Each candidate control law is fitted to the recovered controls within predefined parameter ranges. The fitted longitudinal and lateral laws are then applied together to the corresponding kinematic model to reconstruct the trajectory. As in the kinematic layer, we compare this reconstruction with the original prediction. This verifies that a candidate law not only matches the recovered controls, but also reproduces the predicted motion when executed.

Scene-reference layer. Some control laws depend on the surrounding scene and therefore require a specific scene reference to determine their controls. For example, IDM following requires a leading

agent, lane-keeping requires a lane reference, and stopping behavior may depend on a stop line. For a candidate c with scene relation g and reference r, we measure how well the reference agrees with the predicted motion:

$$e^{\text{scene}}_{w,c} = \frac{1}{|\Omega^{\text{scene}}_{w,c}|} \sum_{t \in \Omega^{\text{scene}}_{w,c}} \|d_g(\boldsymbol{\tau}, \phi_c, r_t)\|^2_{\Sigma_g^{-1}}, \tag{8}$$

where d_g measures how well candidate reference r_t matches the predicted motion under relation g, and $\Omega^{\text{scene}}_{w,c}$ contains the valid time steps. For example, d_g can compare a predicted stopping target with a stop line or check whether a candidate leader has the required relative position and path alignment for following. Σ_g provides a fixed scale estimated from the training data. Detailed binding rules and residual definitions are given in Appendix A.4. Each grounded candidate is then executed using the controls computed from its scene reference and compared with the original prediction. This verifies that the selected reference can actually reproduce the predicted motion, rather than only providing a semantic label.

Reconstruction check. For each compatible explanation c, we execute its combined mechanisms to obtain $\hat{\boldsymbol{\tau}}^c$ and compare it with the original prediction:

$$e^{\text{pos}}_{w,c} = \frac{1}{|\mathcal{T}_w|} \sum_{t \in \mathcal{T}_w} \|\hat{\mathbf{p}}^c_t - \mathbf{p}_t\|_2^2. \tag{9}$$

Unlike the layer-specific residuals, $e^{\text{pos}}_{w,c}$ checks whether the complete mechanism combination can reproduce the predicted motion. This check is applied after control fitting and scene grounding.

4.2 Physical Evidence Across Layers

Each layer evaluates a different aspect of the predicted motion, so the residuals of different layers have different scales. We first normalize these residuals using fixed scales estimated from the training data. For each complete explanation c in segment w, we then combine the normalized residuals as

$$S_{w,c} = \bar{e}^{\text{kin}}_{w,b(c)} + \bar{e}^{\text{lon}}_{w,c} + \bar{e}^{\text{lat}}_{w,c} + \bar{e}^{\text{scene}}_{w,c} + \bar{e}^{\text{pos}}_{w,c}. \tag{10}$$

Here, $S_{w,c}$ summarizes how well the explanation—including its kinematic model, control laws, scene references, and final reconstruction—agrees with the predicted trajectory. A smaller score indicates a more compatible explanation. Rather than selecting a single explanation, we convert these scores into normalized support:

$$A_{w,c} = \frac{\exp(-S_{w,c}/\tau)}{\sum_{c' \in \mathcal{B}_w} \exp(-S_{w,c'}/\tau)}, \qquad \alpha^{(\ell)}_{w,h} = \sum_{c:\, c_\ell = h} A_{w,c}, \tag{11}$$

where $\mathcal{B}_w$ is the set of valid complete explanations. $A_{w,c}$ represents the support for a complete explanation c, based on its overall physical consistency with the predicted motion. To obtain the support for a mechanism h at layer ℓ, we sum the weights of all complete explanations that use h at that layer. Thus, $\alpha^{(\ell)}_{w,h}$ reflects not only how well h fits the motion, but also how well it works together with compatible mechanisms at the other layers. Figure 2 visualizes these weights.

4.3 Counterfactual Futures and Safety Matrix

We use the retained complete explanations to generate counterfactual futures. A sequence $j = (c_1, \ldots, c_W)$ selects one complete explanation $c_w \in \mathcal{B}_w$ for each segment, including its kinematic model, longitudinal and lateral control laws, and required scene references. During rollout, these mechanisms are executed jointly rather than independently selected from the layer-wise evidence. The scene references provide the inputs required by the control laws, the control laws produce the controls, and the kinematic model converts them into the next state under the constraints in Eq. (5). The final state of each segment initializes the next, while scene-dependent controls are recomputed from the simulated state. Different retained explanation sequences therefore produce different physically grounded futures.

For the ego vehicle, these sequences generate Q candidate futures $\{\boldsymbol{\tau}^{\text{cand}}_q\}_{q=1}^{Q}$. For surrounding agent i, host mode m, and retained sequence j, the corresponding future is denoted by $\hat{\boldsymbol{\tau}}_{i,m,j}$.

The candidate construction and budget are detailed in Appendix B.1. We then evaluate every ego candidate against every retained surrounding-agent future:

$$\sigma_{q,i,m,j} = \lambda_T \exp\left(-\frac{\mathrm{TTC}_{q,i,m,j}}{t_{\mathrm{safe}}}\right) + (1-\lambda_T)\left[1 - \tanh\left(\frac{d^{\min}_{q,i,m,j}}{d_{\mathrm{safe}}}\right)\right], \tag{12}$$

where $\mathrm{TTC}_{q,i,m,j}$ is the time to the first footprint overlap and $d^{\min}_{q,i,m,j}$ is the minimum footprint separation. If no overlap occurs within the horizon, TTC is set to infinity. A larger σ indicates a riskier interaction. These pairwise risks form the counterfactual safety matrix

$$\mathcal{R}_{q,(i,m,j)} = \sigma_{q,i,m,j}, \tag{13}$$

whose rows correspond to ego candidates and columns to surrounding-agent futures. The counterfactual trajectories, mechanism evidence, and safety matrix are then passed to Refiner.

4.4 Temporal Evidence and Neural Ego Correction

Mechanism evidence is computed over short segments, so it can fluctuate from one segment to the next. We therefore pass the segment-wise evidence through a lightweight temporal convolutional network (TCN) (Bai et al., 2018), which integrates it across neighboring segments and outputs temporally refined explanation weights $\widetilde{A}_{w,c}$. Refiner combines these weights with the counterfactual trajectories and the safety matrix $\mathcal{R}$ from Sec. 4.3. It aggregates the futures of each surrounding agent using the host-mode probabilities $\pi_{i,m}$ and the refined mechanism support, and it encodes each ego candidate with its mechanism sequence and risk features. From the resulting scene representation, Refiner predicts normalized weights β_q over the Q ego candidates. Their weighted average and its offset from the host ego trajectory are

$$\bar{\boldsymbol{\tau}}^{\mathrm{cand}} = \sum_{q=1}^{Q} \beta_q \boldsymbol{\tau}_q^{\mathrm{cand}}, \qquad \boldsymbol{\tau}^{\mathrm{hint}} = \bar{\boldsymbol{\tau}}^{\mathrm{cand}} - \boldsymbol{\tau}_0^{\mathrm{ego}}, \tag{14}$$

where $\boldsymbol{\tau}^{\mathrm{hint}}$ is the physics hint for the correction. A recurrent decoder takes the host trajectory, $\boldsymbol{\tau}^{\mathrm{hint}}$, and the encoded scene evidence as input and predicts the bounded residual $\delta\boldsymbol{\tau}_\theta$ in Eq. (2). The counterfactual candidates therefore guide the correction through the decoder inputs, and the final ego trajectory is the host trajectory plus this learned residual. Appendix C specifies the temporal module, the encoders, and the decoder.

5 Training

The initial physical definitions follow standard models from the literature (Rajamani, 2012; Treiber et al., 2000; Reeds & Shepp, 1990). We calibrate the mechanism library, physical bounds, residual scales, and evidence temperatures on the training data and keep them fixed afterwards. With the host predictor frozen, we jointly train the Refiner with

$$\mathcal{L}_{\mathrm{plan}} = \mathcal{L}_{\mathrm{traj}} + \lambda_{\mathrm{safe}}\mathcal{L}_{\mathrm{safe}} + \lambda_{\mathrm{anchor}}\mathcal{L}_{\mathrm{anchor}}, \tag{15}$$

where $\mathcal{L}_{\mathrm{traj}}$ is the smooth-L_1 error between the final ego positions and the ground truth, averaged over time steps with valid ground truth. The safety term penalizes worst-case risk in both the candidate weights and the final ego trajectory:

$$\mathcal{L}_{\mathrm{safe}} = \sum_q \beta_q \max_{i,m,j} \sigma_{\mathrm{train}}\big(\boldsymbol{\tau}_q^{\mathrm{cand}}, \hat{\boldsymbol{\tau}}_{i,m,j}\big) + \lambda \max_{i,m,j} \sigma_{\mathrm{train}}\big(\hat{\boldsymbol{\tau}}^{\mathrm{ego}}, \hat{\boldsymbol{\tau}}_{i,m,j}\big). \tag{16}$$

In this loss, σ_{train} is a smooth footprint-distance surrogate defined in Appendix E.1. The risk features that Refiner receives for each candidate are still computed with Eq. (12) and remain fixed. The evidence anchor, $\mathcal{L}_{\mathrm{anchor}} = \mathrm{mean}_w\, \mathrm{KL}(A_w \| \widetilde{A}_w)$, is averaged over all valid agent–mode pairs and the source trajectories of the ego candidates. It softly penalizes refined weights $\widetilde{A}_w$ that drift from the physical evidence A_w. Because each segment is compared only with its own evidence and the objective has no explicit smoothness term, the TCN learns temporal consistency from neighboring segments under the planning objective. Training requires no mechanism or intent labels, and candidate generation and discrete reference selection remain fixed and are not trained end to end. The calibration stages, the gradient path, and the optimization settings are listed in Appendix E.

Table 2: Ego planning results of three frozen hosts on nuScenes.

	L2 (m) ↓				Collision (%) ↓				Miss	Comfort	mPYR	Yaw viol.
Variant	1 s	2 s	3 s	Avg.	1 s	2 s	3 s	Avg.	(%) ↓	(%) ↑	(rad/s) ↓	(%) ↓
UniAD	0.48	0.96	1.65	1.03	0.05	0.17	0.71	0.31	31.96	69.84	0.143	20.38
+ Neural-only corrector	0.47	0.96	1.62	1.02	0.04	0.14	0.60	0.26	31.24	70.31	0.138	19.86
+ PHLEX	**0.45**	**0.93**	**1.56**	**0.98**	**0.02**	**0.08**	**0.38**	**0.16**	**29.71**	**76.25**	**0.111**	**13.68**
VAD-Base	0.54	1.15	1.98	1.22	0.04	0.39	1.17	0.53	40.99	75.20	0.131	16.08
+ Neural-only corrector	0.53	1.13	1.94	1.20	0.04	0.33	1.03	0.47	40.33	75.46	0.129	15.80
+ PHLEX	**0.51**	**1.12**	**1.90**	**1.18**	**0.03**	**0.24**	**0.79**	**0.35**	**38.99**	**81.89**	**0.104**	**12.38**
SparseDrive-S	0.44	0.92	1.69	1.02	0.07	0.19	0.71	0.32	33.10	80.47	0.107	12.60
+ Neural-only corrector	0.44	0.92	1.67	1.01	0.07	0.19	0.65	0.30	32.71	80.60	0.102	12.46
+ PHLEX	**0.43**	**0.90**	**1.65**	**0.99**	**0.06**	**0.15**	**0.56**	**0.26**	**31.92**	**84.33**	**0.086**	**10.74**

6 EXPERIMENTS

Datasets and Host Models. We evaluate PHLEX on nuScenes (Caesar et al., 2020) and Argoverse 2 (Wilson et al., 2021) in two settings. Ego planning evaluates the single ego plan of UniAD, VAD-Base, and SparseDrive-S on nuScenes with a 3 s horizon. Multimodal agent forecasting evaluates the K forecast modes of surrounding agents from UniAD on nuScenes (3 s), and from MTR and QCNet on both nuScenes and Argoverse 2 (6 s) (Hu et al., 2023b; Shi et al., 2022; Zhou et al., 2023). All host models remain frozen. Only the temporal module and Refiner are trained.

Metrics. We follow the UniAD protocol for ego planning and report the L2 error and collision rate at 1, 2, and 3 s, and the Miss Rate. For agent forecasts, minADE and minFDE measure prediction accuracy over the K modes, while KVR measures the fraction of trajectories that violate class-specific limits on peak acceleration, yaw rate, or jerk. Comfort is the fraction of trajectories whose longitudinal and lateral accelerations and jerks stay within fixed bounds. For ego plans, we also report the mean peak absolute yaw rate (mPYR) and the fraction of plans whose peak yaw rate exceeds a fixed threshold. Recon. is the mean squared position error between an executed explanation and the host forecast it explains. All thresholds are fixed before testing and shared by all compared methods (Appendix D).

6.1 EGO PLANNING PERFORMANCE

Table 2 evaluates the ego plans of three frozen hosts. The neural-only corrector provides modest improvements over the original host outputs, while PHLEX achieves larger and more consistent gains across planning accuracy, safety, and physical motion quality. For UniAD, VAD-Base, and SparseDrive-S, PHLEX reduces the average L2 error from 1.03, 1.22, and 1.02 m to 0.98, 1.18, and 0.99 m, respectively. The corresponding average collision rates decrease from 0.31%, 0.53%, and 0.32% to 0.16%, 0.35%, and 0.26%, while Miss Rates are also consistently reduced. Beyond trajectory accuracy, PHLEX produces noticeably more physically well-behaved ego plans. Comfort increases by 3.86–6.69 percentage points across the three hosts, while mPYR decreases by 0.021–0.032 rad/s and yaw-rate violations by 1.86–6.70 percentage points. These improvements are consistently larger than those obtained by the neural-only corrector, indicating that the physical mechanism representation contributes beyond generic learned trajectory correction.

6.2 MULTIMODAL AGENT FORECASTING

Table 3 evaluates the K-mode forecasts of surrounding agents from UniAD, MTR, and QCNet on nuScenes and Argoverse 2. In all five settings, PHLEX keeps forecast accuracy close to the host, with minADE and minFDE increasing by at most 0.02 and 0.05 m. KVR drops more than 80%, to 1.54–4.21%, and comfort rises by 10.61–15.84 percentage points. These effects hold for the planning-oriented UniAD and for the dedicated forecasters MTR and QCNet, and on both datasets.

Table 3: Multimodal agent forecasting results of three frozen hosts on nuScenes and Argoverse 2.

		minADE (m) ↓		minFDE (m) ↓		KVR (%) ↓		Comfort (%) ↑	
Host model	Dataset, horizon	Host	PHLEX	Host	PHLEX	Host	PHLEX	Host	PHLEX
UniAD	nuScenes, 3 s	**0.71**	0.72	**1.02**	1.04	18.37	**2.28**	71.62	**83.14**
MTR	nuScenes, 6 s	**1.06**	1.07	**2.33**	2.37	14.76	**1.96**	75.36	**86.72**
QCNet	nuScenes, 6 s	**0.94**	**0.94**	**1.84**	1.86	11.58	**1.54**	78.34	**88.95**
MTR	Argoverse 2, 6 s	**0.73**	0.75	**1.44**	1.49	23.69	**4.21**	65.74	**81.58**
QCNet	Argoverse 2, 6 s	**0.73**	0.74	**1.27**	1.30	19.18	**2.63**	70.83	**84.87**

Table 4: Mechanism-library ablation on UniAD/nuScenes. The Kin/Lon/Lat/Scene column gives the number of candidate mechanisms available at the kinematic, longitudinal-control, lateral-control, and scene-reference layers. Col. is the collision rate averaged over 1, 2, and 3 s.

Variant	Kin/Lon/Lat/Scene	Recon. (m^2) ↓	KVR (%) ↓	minADE (m) ↓	minFDE (m) ↓	Col. (%) ↓
Matched flat candidates	3 / 6 / 5 / 6	0.094	4.87	0.76	1.13	0.24
Early hard selection	3 / 6 / 5 / 6	0.071	3.28	0.74	1.09	0.21
Minimal library	1 / 2 / 2 / 0	0.131	6.14	0.79	1.19	0.26
Small library	2 / 4 / 3 / 3	0.089	4.02	0.76	1.12	0.22
Medium library	3 / 5 / 4 / 5	0.061	2.78	0.73	1.06	0.18
w/o scene layer	3 / 6 / 5 / 0	0.079	3.44	0.75	1.10	0.22
w/o longitudinal diversity	3 / 2 / 5 / 6	0.073	3.19	0.74	1.08	0.20
w/o lateral diversity	3 / 6 / 2 / 6	0.070	3.11	0.74	1.08	0.20
PHLEX (full library)	3 / 6 / 5 / 6	**0.052**	**2.28**	**0.72**	**1.04**	**0.16**

6.3 Ablation Studies

Mechanism representation. We ablate how the physical mechanism representation is constructed, keeping the downstream planner fixed. Table 4 varies the hierarchical design, the library size, and the diversity inside individual layers. *Matched flat candidates* keeps the full library and the same candidate budget, but treats complete mechanism combinations as a single flat set without layer representation. *Early hard selection* commits to one kinematic branch before the remaining layers are evaluated, which restricts the control-law and scene-reference search to that branch.

The full representation is best on every metric, with 0.052 m^2 reconstruction error and 2.28% KVR. Flattening the candidate set or committing early to one kinematic branch degrades all five metrics, raising the reconstruction error to 0.094 and 0.071 m^2. Performance also degrades monotonically as the library shrinks, to 0.131 m^2 and 6.14% KVR at the minimal setting. Within the full library, performance degrades when diversity is reduced in any layer, with the largest degradation observed for the scene layer at 0.079 m^2, versus 0.073 and 0.070 m^2 for the longitudinal and lateral layers.

Safety-related components. Table 5 evaluates the contribution of the rollout-risk features across three frozen ego-planning hosts. Removing them has little effect on trajectory accuracy, but consistently increases collision and yaw-rate violation rates while reducing comfort. In particular, collision rises from 0.16%, 0.35%, and 0.26% to 0.21%, 0.41%, and 0.28% for UniAD, VAD-Base, and SparseDrive-S, respectively. These results show that rollout-risk features mainly improve planning safety and physical motion quality.

6.4 Qualitative Analysis of Physical Explanations

Figure 2(a) shows that the dominant mechanisms change consistently with the predicted motion, from lane-following straight motion to turning and back to straight motion. After TCN integration, these motion phases are preserved while local fluctuations are reduced (Fig. 2(b)), indicating more temporally consistent physical explanations. Figure 2(c) shows that a single forecast can retain multiple compatible physical explanations rather than being assigned one hard mechanism. The connections across layers keep control laws consistent with their scene references, making the explanations directly executable and modifiable. For example, replacing IDM following with target-speed tracking removes the response to the leading agent, producing a corresponding counterfactual future.

Table 5: Safety-related component ablations across three frozen ego-planning hosts on nuScenes (3 s horizon). All metrics are computed on the ego plan as in Table 2. *Frozen host* reports the unmodified host output without any explanation or learned correction.

Host	Variant	L2 (m) ↓	Col. (%) ↓	Comfort (%) ↑	Yaw viol. (%) ↓
UniAD	Frozen host	1.03	0.31	69.84	20.37
	PHLEX w/o rollout risk	1.00	0.21	75.73	14.92
	Full PHLEX	**0.98**	**0.16**	**76.25**	**13.68**
VAD-Base	Frozen host	1.22	0.53	75.20	16.08
	PHLEX w/o rollout risk	1.19	0.41	80.64	13.16
	Full PHLEX	**1.18**	**0.35**	**81.89**	**12.38**
SparseDrive-S	Frozen host	1.02	0.32	80.47	12.60
	PHLEX w/o rollout risk	**0.99**	0.28	83.51	11.25
	Full PHLEX	**0.99**	**0.26**	**84.33**	**10.74**

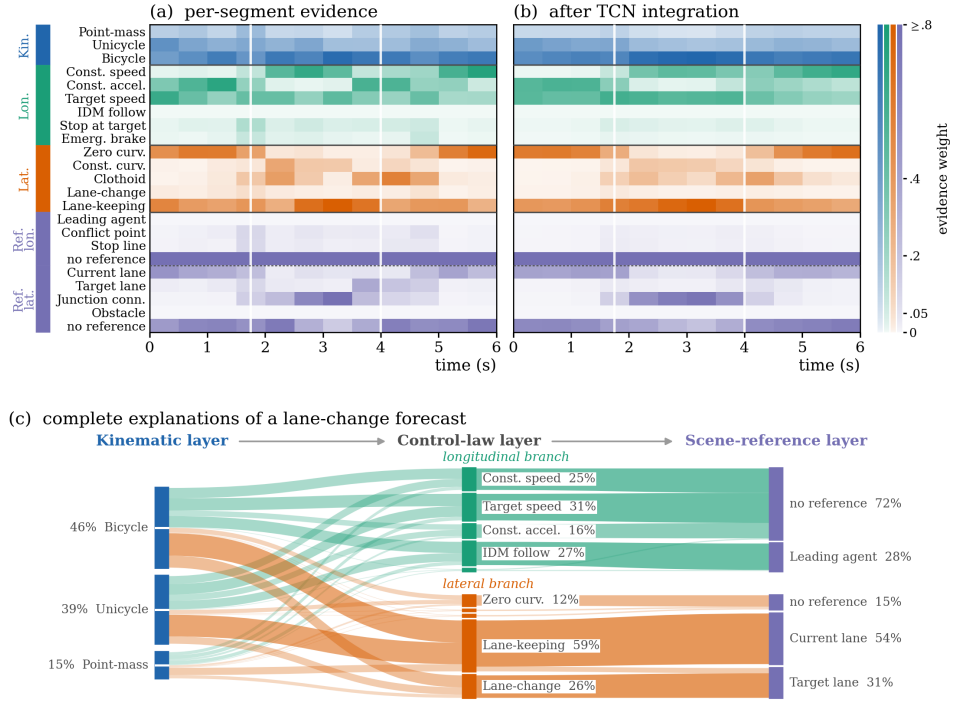

Figure 2: **Qualitative visualization of physical explanations.** (a) Per-segment mechanism evidence over a 6 s horizon, where darker cells indicate higher support. (b) Mechanism evidence after TCN integration. (c) Complete explanations of a lane-change forecast. Block height denotes mechanism support, while ribbon width denotes the support shared by connected mechanisms.

7 Conclusion

We presented PHLEX, a predictor-agnostic framework that maps predicted trajectories into layered physical explanations through kinematic models, control laws, and scene references. By retaining multiple compatible explanations, PHLEX can execute and vary them to generate physically grounded counterfactual futures for refining ego planning without modifying the host predictor. Experiments across multiple host models and datasets show that PHLEX improves physical feasibility, planning safety, and motion quality while largely preserving trajectory accuracy, and that the same mechanism representation transfers across different predictors and datasets. These results show that explicit physical mechanisms can complement learned trajectory prediction with interpretable and controllable counterfactual reasoning for motion planning. However, PHLEX is limited by the coverage of its mechanism library. Appendix F discusses the limitations in more detail.

AI USE STATEMENT

Generative AI tools assisted with organizing the method, identifying inconsistencies in the method description, searching for related work, drafting and editing text, and checking references. We also used AI tools to assist in implementing parts of the experimental and evaluation code. The authors reviewed all AI-assisted content, verified every cited reference against its original source, and take full responsibility for the content of this paper.

ETHICS STATEMENT

We have read and follow the ICLR Code of Ethics. All experiments use public driving datasets under their terms of use, and no new data or human subjects are involved. Our evaluation is open loop, so the results do not establish deployment safety. Mechanism support measures physical compatibility with a forecast, it does not estimate a driver's intent.

REPRODUCIBILITY STATEMENT

Sections 4 and 5 define the method and training objective. Appendices A, B, and C describe the mechanism library and its bounds, candidate construction, and the Refiner architecture. Appendices D and E give the metrics and training settings. All experiments use public datasets with frozen host models.

REFERENCES

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling, 2018. URL https://arxiv.org/abs/1803.01271.

Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11621–11631, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Caesar_nuScenes_A_Multimodal_Dataset_for_Autonomous_Driving_CVPR_2020_paper.html.

Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *Proceedings of the Conference on Robot Learning*, volume 100 of *Proceedings of Machine Learning Research*, pp. 86–99. PMLR, 2020. URL https://proceedings.mlr.press/v100/chai20a.html.

Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, Jeff Schneider, David Bradley, and Nemanja Djuric. Deep kinematic models for kinematically feasible vehicle trajectory predictions. In *IEEE International Conference on Robotics and Automation*, pp. 10563–10569, 2020. doi: 10.1109/ICRA40945.2020.9197560. URL https://arxiv.org/abs/1908.00219.

Shenyuan Gao, Jiazhi Yang, Li Chen, Kashyap Chitta, Yihang Qiu, Andreas Geiger, Jun Zhang, and Hongyang Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 37, 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/a6a066fb44f2fe0d36cf740c873b8890-Abstract-Conference.html.

Danijar Hafner, Timothy Lillicrap, Jimmy Ba, and Mohammad Norouzi. Dream to control: Learning behaviors by latent imagination. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. URL https://openreview.net/forum?id=S1lOTC4tDS.

Anthony Hu, Lloyd Russell, Hudson Yeo, Zak Murez, George Fedoseev, Alex Kendall, Jamie Shotton, and Gianluca Corrado. GAIA-1: A generative world model for autonomous driving, 2023a. URL https://arxiv.org/abs/2309.17080.

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, pp. 17853–17862, 2023b. URL https://openaccess.thecvf.com/content/CVPR2023/html/Hu_Planning-Oriented_Autonomous_Driving_CVPR_2023_paper.html.

Auke Jan Ijspeert, Jun Nakanishi, Heiko Hoffmann, Peter Pastor, and Stefan Schaal. Dynamical movement primitives: Learning attractor models for motor behaviors. *Neural Computation*, 25 (2):328–373, 2013. doi: 10.1162/NECO_a_00393.

Yingyan Li, Yuqi Wang, Yang Liu, Jiawei He, Lue Fan, and Zhaoxiang Zhang. End-to-end driving with online trajectory evaluation via BEV world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27137–27146, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Li_End-to-End_Driving_with_Online_Trajectory_Evaluation_via_BEV_World_Model_ICCV_2025_paper.html.

Haicheng Liao, Xuelin Li, Yongkang Li, Hanlin Kong, Chengyue Wang, Bonan Wang, Yanchen Guan, KaHou Tam, and Zhenning Li. CDSTraj: Characterized diffusion and spatial-temporal interaction network for trajectory prediction in autonomous driving. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 7331–7339, 2024a. doi: 10.24963/ijcai.2024/811. URL https://www.ijcai.org/proceedings/2024/811.

Haicheng Liao, Chengyue Wang, Zhenning Li, Yongkang Li, Bonan Wang, Guofa Li, and Chengzhong Xu. Physics-informed trajectory prediction for autonomous driving under missing observation. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pp. 6841–6849, 2024b. doi: 10.24963/ijcai.2024/756. URL https://www.ijcai.org/proceedings/2024/756.

Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S. Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple & efficient attention networks. In *IEEE International Conference on Robotics and Automation*, pp. 2980–2987, 2023. doi: 10.1109/ICRA48891.2023.10160609. URL https://arxiv.org/abs/2207.05844.

Tung Phan-Minh, Elena Corina Grigore, Freddy A. Boulton, Oscar Beijbom, and Eric M. Wolff. CoverNet: Multimodal behavior prediction using trajectory sets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14074–14083, 2020. URL https://openaccess.thecvf.com/content_CVPR_2020/html/Phan-Minh_CoverNet_Multimodal_Behavior_Prediction_Using_Trajectory_Sets_CVPR_2020_paper.html.

Rajesh Rajamani. *Vehicle Dynamics and Control*. Springer, New York, NY, 2 edition, 2012. doi: 10.1007/978-1-4614-1433-9.

James A. Reeds and Lawrence A. Shepp. Optimal paths for a car that goes both forwards and backwards. *Pacific Journal of Mathematics*, 145(2):367–393, 1990. doi: 10.2140/pjm.1990.145.367.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*, pp. 683–700. Springer, 2020. doi: 10.1007/978-3-030-58523-5_40.

Ari Seff, Brian Cera, Dian Chen, Mason Ng, Aurick Zhou, Nigamaa Nayakanti, Khaled S. Refaat, Rami Al-Rfou, and Benjamin Sapp. MotionLM: Multi-agent motion forecasting as language modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8579–8590, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Seff_MotionLM_Multi-Agent_Motion_Forecasting_as_Language_Modeling_ICCV_2023_paper.html.

Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. In *Advances in Neural Information Processing Systems*, volume 35, pp. 6531–6543, 2022. doi: 10.52202/068431-0473. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/2ab47c960bfee4f86dfc362f26ad066a-Abstract-Conference.html.

Haoran Song, Di Luan, Wenchao Ding, Michael Y. Wang, and Qifeng Chen. Learning to predict vehicle trajectories with model-based planning. In *Proceedings of the 5th Conference on Robot Learning*, volume 164, pp. 1035–1045, 2022. URL https://proceedings.mlr.press/v164/song22a.html.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical Review E*, 62(2):1805–1824, 2000. doi: 10.1103/PhysRevE.62.1805.

Renukanandan Tumu, Lars Lindemann, Truong Nghiem, and Rahul Mangharam. Physics constrained motion prediction with uncertainty quantification. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–8, 2023. doi: 10.1109/IV55152.2023.10186812. URL https://arxiv.org/abs/2302.01060.

Balakrishnan Varadarajan, Ahmed Hefny, Avikalp Srivastava, Khaled S. Refaat, Nigamaa Nayakanti, Andre Cornman, Kan Chen, Bertrand Douillard, Chi Pang Lam, Dragomir Anguelov, and Benjamin Sapp. MultiPath++: Efficient information fusion and trajectory aggregation for behavior prediction. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7814–7821, 2022. URL https://arxiv.org/abs/2111.14973.

Matt Vitelli, Yan Chang, Yawei Ye, Ana Ferreira, Maciej Wołczyk, Błażej Osiński, Moritz Niendorf, Hugo Grimmett, Qiangui Huang, Ashesh Jain, and Peter Ondruska. SafetyNet: Safe planning for real-world self-driving vehicles using machine-learned policies. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pp. 897–904, 2022. URL https://ieeexplore.ieee.org/document/9811576.

Xiaofeng Wang, Zheng Zhu, Guan Huang, Xinze Chen, Jiagang Zhu, and Jiwen Lu. DriveDreamer: Towards real-world-driven world models for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024. URL https://arxiv.org/abs/2309.09777.

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/4734ba6f3de83d861c3176a6273cac6d-Abstract-round2.html.

Wenzhao Zheng, Weiliang Chen, Yuanhui Huang, Borui Zhang, Yueqi Duan, and Jiwen Lu. OccWorld: Learning a 3D occupancy world model for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 55–72, 2024. URL https://arxiv.org/abs/2311.16038.

Yupeng Zheng, Pengxuan Yang, Zebin Xing, Qichao Zhang, Yuhang Zheng, Yinfeng Gao, Pengfei Li, Teng Zhang, Zhongpu Xia, Peng Jia, XianPeng Lang, and Dongbin Zhao. World4Drive: End-to-end autonomous driving via intention-aware physical latent world model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 28632–28642, 2025. URL https://openaccess.thecvf.com/content/ICCV2025/html/Zheng_World4Drive_End-to-End_Autonomous_Driving_via_Intention-aware_Physical_Latent_World_Model_ICCV_2025_paper.html.

Zikang Zhou, Jianping Wang, Yung-Hui Li, and Yu-Kai Huang. Query-centric trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17863–17873, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Zhou_Query-Centric_Trajectory_Prediction_CVPR_2023_paper.html.