FLOWPLAN: FLOW-MATCHING TRAJECTORY PLANNING GUIDED BY COUNTERFACTUAL INTERACTION CONSEQUENCES

Anonymous authors
Paper under double-blind review

ABSTRACT

Modeling inter-agent interactions and their evolution in complex traffic scenarios remains a fundamental challenge in autonomous driving planning. Despite advances in ego-conditioned prediction and joint prediction–planning, it remains difficult to reliably characterize how changes in ego behavior affect critical conflicts and to select effective interaction responses based on these effects. To address this problem, we propose FlowPlan, a flow-matching trajectory planning framework guided by counterfactual interaction consequences. Its Counterfactual Interaction Consequence Model (CF-ICM) evaluates complete ego-trajectory queries under the same observed scene and predicts object- and time-resolved proximity and contact outcomes. Coupled lateral–longitudinal probes use these predictions to construct branch-specific consequence memories and fixed strategy conditions that guide learned corrections to the flow velocity field. By re-evaluating selected intermediate trajectory estimates, FlowPlan updates these memories as generation proceeds while retaining a fixed strategy condition within each branch. Completed candidates are re-evaluated and selected using predicted interaction consequences, feasibility, and driving quality. Experiments on the large-scale real-world nuPlan benchmark demonstrate strong closed-loop planning performance in both non-reactive and reactive settings. We further introduce nuPlan-CIP (nuPlan Counterfactual Interaction Perturbation Benchmark), which applies controlled perturbations of varying interaction intensity while preserving the underlying scene context. Results on nuPlan-CIP show consistent responses to controlled interaction perturbations, complementing conventional planning metrics with a diagnostic assessment of interaction-dependent behavior.

1 INTRODUCTION

Trajectory planning in complex traffic requires reasoning about how ego actions shape the future scene. A change in ego speed or path can alter surrounding agents' responses and the timing and severity of conflicts, an issue that motivates interactive prediction, joint prediction–planning, and learned traffic simulation (Tang et al., 2022; Huang et al., 2023; Sun et al., 2022; Salzmann et al., 2020; Suo et al., 2021; Zhang et al., 2023). For example, a detour around a static obstacle can create a near gap with an adjacent vehicle even when the nominal route is collision-free. The planner must therefore assess both the quality of a candidate trajectory and the interaction consequences of plausible alternatives.

Learning-based driving has progressed from conditional imitation and robust behavior cloning to object-centric, planning-oriented, and closed-loop planners (Codevilla et al., 2018; Bansal et al., 2019; Chen et al., 2019; Prakash et al., 2021; Chitta et al., 2021; Chen & Krähenbühl, 2022; Renz et al., 2022; Hu et al., 2022; 2023; Jiang et al., 2023; Jia et al., 2023; Dauner et al., 2023; Cheng et al., 2024a). In parallel, motion forecasting has moved from trajectory-set and vectorized representations toward multi-agent transformers and autoregressive interaction models (Chai et al., 2020; Phan-Minh et al., 2020; Gao et al., 2020; Liang et al., 2020; Zhao et al., 2021; Gu et al., 2021; Yuan et al., 2021; Zhou et al., 2022; Shi et al., 2022; Nayakanti et al., 2022; Seff et al., 2023; Shi et al., 2024). These developments provide strong scene representations and multimodal futures, but planning still requires deciding how a particular ego alternative changes the conflict structure of the scene.

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.

Joint prediction–planning and ego-conditioned prediction explicitly couple candidate ego trajectories with surrounding-agent futures (Huang et al., 2023; Liu et al., 2024; Huang et al., 2024; 2026). Yet conditioning a prediction on a proposed ego trajectory is not equivalent to estimating the outcome of executing it; conditional predictors may overstate the ego vehicle's influence (Tang et al., 2022). Driving logs add a supervision gap: they record the joint evolution after the action taken, but not what the same scene would have produced under another ego action. A model accurate on logged futures may therefore be unreliable for alternative maneuvers, particularly in tightly coupled interactions.

For conflict mitigation, the prediction should identify which surrounding objects are affected, when an interaction becomes critical, and whether an alternative changes proximity or contact outcomes. This calls for a task-specific consequence representation rather than only complete surrounding-agent trajectory forecasts. We propose FlowPlan, a flow-matching planner guided by these counterfactual interaction consequences. Its Counterfactual Interaction Consequence Model (CF-ICM) evaluates physically constrained alternative ego speed and path choices under the same observed scene and predicts object- and time-resolved outcomes. It is trained on logged executions and closed-loop rollouts with reactive surrounding agents; each simulated ego alternative is paired with the responses realized in that execution.

FlowPlan uses these predictions during generation as well as final selection. Coupled lateral–longitudinal trajectory probes first query CF-ICM. For each generation branch, their predictions construct an initial object–time consequence memory and a fixed strategy condition; the probe trajectory itself is not the starting state of generation. The flow decoder instead starts from Gaussian noise, uses the memory to guide learned corrections to the velocity field, and refreshes the active memory by re-evaluating selected intermediate trajectory estimates while the branch's strategy condition remains fixed. This design builds on recent progress in diffusion- and flow-based trajectory generation (Ho et al., 2020; Song et al., 2021b; Lipman et al., 2023; Liu et al., 2023; Tong et al., 2024; Zheng et al., 2025; Yang et al., 2024; Tan et al., 2025; Wang et al., 2025), but uses structured interaction consequences as an internal, dynamically updated guidance signal.

We evaluate FlowPlan on the large-scale real-world nuPlan benchmark (Karnchanachari et al., 2024), where it achieves strong closed-loop performance in non-reactive and reactive settings. We also introduce nuPlan-CIP (nuPlan Counterfactual Interaction Perturbation Benchmark). It varies interaction intensity through controlled perturbations while preserving scene context, complementing aggregate planning scores with a diagnostic test of interaction-dependent behavior. Our contributions are threefold:

- We introduce CF-ICM to predict object- and time-resolved consequences of alternative ego speed and path choices.
- We develop consequence-conditioned flow generation: coupled lateral–longitudinal probes initialize branch-specific consequence memories and fixed strategy conditions; the memories guide velocity-field corrections and are refreshed as candidates evolve.
- We introduce nuPlan-CIP to assess responses to controlled changes in interaction intensity alongside conventional nuPlan closed-loop evaluation.

2 Related Work

2.1 Interaction-Aware Trajectory Planning

Learning-based driving has progressed from direct imitation to structured trajectory planning with richer scene representations. Early end-to-end and imitation approaches include conditional imitation, ChauffeurNet, Deep Imitative Models, Learning by Cheating, TransFuser, NEAT, Learning from All Vehicles, and InterFuser (Codevilla et al., 2018; Bansal et al., 2019; Rhinehart et al., 2020; Chen et al., 2019; Prakash et al., 2021; Chitta et al., 2023; 2021; Chen & Krähenbühl, 2022; Shao et al., 2023). Later planners expose objects, occupancy, or vectorized structure explicitly, including PlanT, ST-P3, UniAD, VAD, and ThinkTwice (Renz et al., 2022; Hu et al., 2022; 2023; Jiang et al., 2023; Jia et al., 2023). PDM-Open and PLUTO further emphasize closed-loop evaluation and behavior-aware planning on nuPlan (Dauner et al., 2023; Cheng et al., 2024a; Karnchanachari et al., 2024).

Motion forecasting supplies many interaction representations used by planners. Multimodal trajectory-set methods (Chai et al., 2020; Phan-Minh et al., 2020), vectorized and graph encoders (Gao et al., 2020; Liang et al., 2020), and goal-conditioned predictors (Zhao et al., 2021; Gu et al., 2021) evolved into multi-agent interaction models such as Trajectron++, AgentFormer, HiVT, MTR, Wayformer, MotionLM, and MTR++ (Salzmann et al., 2020; Yuan et al., 2021; Zhou et al., 2022; Shi et al., 2022; Nayakanti et al., 2022; Seff et al., 2023; Shi et al., 2024). Interactive benchmarks and learned simulators likewise move evaluation toward joint, closed-loop behavior (Ettinger et al., 2021; Sun et al., 2022; Suo et al., 2021; Zhang et al., 2023; Wu et al., 2024).

Joint prediction–planning more directly exposes the planner to interaction-dependent futures. GameFormer, BeTopNet, DTPP, and OccDriver couple ego planning with game-theoretic reasoning, topology, ego-conditioned prediction, or occupancy forecasts (Huang et al., 2023; Liu et al., 2024; Huang et al., 2024; 2026). A complementary question is what conditioning on a planned ego trajectory means: Tang et al. (2022) show that conditional behavior prediction can overestimate ego influence and motivate an interventional interpretation. FlowPlan follows this distinction but predicts object- and time-resolved proximity/contact consequences rather than complete surrounding-agent trajectories or occupancy fields, and uses them during both generation and candidate selection.

2.2 Generative Trajectory Planning

Diffusion and score-based models learn iterative transformations between data and noise (Ho et al., 2020; Song et al., 2021b;a); deterministic sampling, guidance, specialized solvers, and improved parameterizations make generation faster and controllable (Dhariwal & Nichol, 2021; Ho & Salimans, 2022; Lu et al., 2022; Karras et al., 2022; Song et al., 2023). These ideas extend to sequential decision making through Diffuser, Decision Diffuser, and Diffusion Policy (Janner et al., 2022; Ajay et al., 2023; Chi et al., 2023). In autonomous driving, Diffusion Planner combines trajectory generation with planning guidance, while Diffusion-ES couples diffusion with evolutionary search over external objectives (Zheng et al., 2025; Yang et al., 2024).

Continuous-time generative modeling builds on normalizing flows and neural differential equations (Rezende & Mohamed, 2015; Chen et al., 2018; Grathwohl et al., 2019). Flow matching regresses a target velocity field (Lipman et al., 2023), rectified flow encourages straighter transport paths (Liu et al., 2023), and conditional flow matching and stochastic interpolants generalize the probability path (Tong et al., 2024; Albergo et al., 2023). For driving, Flow Planner uses fine-grained trajectory tokenization and guidance (Tan et al., 2025), FlowDrive adds trajectory-pattern reweighting and in-the-loop guidance (Wang et al., 2025), and DFP-FM uses flow matching in a history-annealed planning framework (Zhang et al., 2026). These methods show that generative planning can be steered by learned or external objectives. FlowPlan instead uses structured object–time interaction consequences that are recomputed as the candidate trajectory evolves.

3 Method

3.1 Planning Task and Model Overview

At each planning instant, FlowPlan receives a vectorized scene $X = \{E, A, S, L, R_{\text{nav}}\}$: the current ego state, dynamic-agent histories, static objects, lane and signal information, and the navigation route. It returns ego poses $\tau = \{(p_{x,h}, p_{y,h}, \psi_h)\}_{h=1}^{H}$ at intervals Δt over horizon $T = H\Delta t$. All spatial inputs use the current ego frame.

FlowPlan has three steps. A scene encoder summarizes the observed traffic state. The Counterfactual Interaction Consequence Model (CF-ICM) evaluates complete ego-trajectory queries and predicts object–time interaction consequences. A consequence-conditioned flow generator uses coupled intention probes to construct a fixed strategy condition and an initial object–time consequence memory for each branch. Starting from Gaussian noise, it then generates candidate trajectories through flow decoding while periodically refreshing the active consequence memory, and finally selects a completed candidate. The distinction between a *complete trajectory query* to CF-ICM and a *local query* that reads its memory is central to the design. Figure 1 summarizes the complete architecture.

3.2 Predicting Interaction Consequences

The scene encoder applies sequence-and-channel mixing to dynamic-agent histories and lane polylines, followed by masked pooling, while a separate MLP embeds static objects. All tokens receive spatial and type embeddings; lane tokens additionally receive route-membership, speed-limit, and traffic-signal embeddings. Self-attention gives scene tokens $S_X = \mathrm{Enc}_\eta(X)$ and contextual object tokens z_i. The object index includes both dynamic and static objects.

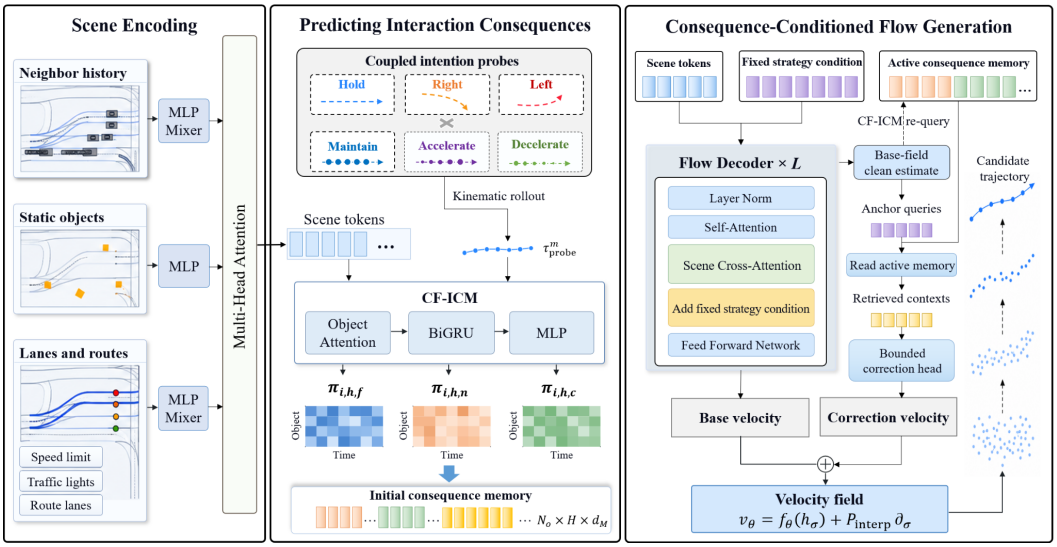

Figure 1: **FlowPlan architecture.** Coupled intention probes query CF-ICM to initialize branch-specific strategy conditions and object–time consequence memories. CF-Flow reads and refreshes the active memory while decoding candidate trajectories.

For a complete physical ego trajectory τ, CF-ICM first forms $Y = \mathrm{ToRep}(\tau) \in \mathbb{R}^{H\times 4}$, whose row at time h is $(p_{x,h}/L_p, p_{y,h}/L_p, \cos\psi_h, \sin\psi_h)$. It pairs each ego pose with each object token, fuses the pairs with scene context using attention, and models their temporal sequence with a bidirectional recurrent layer. The output is

$$(Z, \Pi) = F_\omega(S_X, \tau), \qquad \pi_{i,h} = (\pi_{i,h,f}, \pi_{i,h,n}, \pi_{i,h,c}), \tag{1}$$

where $Z_{i,h}$ is an object–time feature and $\pi_{i,h}$ gives probabilities of free clearance, non-contact near gap, and contact. This representation retains the object, time, and consequence class of each predicted interaction.

CF-ICM is trained on complete trajectory queries paired with interaction labels from the *same execution*. Logged trajectories provide factual pairs; reactive simulation rollouts provide alternative ego trajectories and their realized surrounding-agent responses. This pairing is necessary because logged scenes alone do not reveal the outcome of an unexecuted maneuver. The tokenization, paired-query, and classifier equations are provided in Appendix A.1; Eq. (1) gives the output used by the planner.

3.3 Consequence-Conditioned Generation

Coupled intention probes and branch memory. We construct counterfactual ego-motion probes by coupling lateral and longitudinal motion modes. The lateral mode is chosen from $\mathcal{A}_{\mathrm{lat}} = \{\mathrm{hold}, \mathrm{left}, \mathrm{right}\}$, and the longitudinal mode from $\mathcal{A}_{\mathrm{lon}} = \{\mathrm{decelerate}, \mathrm{maintain}, \mathrm{accelerate}\}$. Their Cartesian product $\mathcal{M}_{\mathrm{probe}} = \mathcal{A}_{\mathrm{lat}} \times \mathcal{A}_{\mathrm{lon}}$ yields nine coupled intentions. For each $m \in \mathcal{M}_{\mathrm{probe}}$, the corresponding lateral-offset and speed profiles are jointly rolled out through a kinematic model to obtain a complete, feasible probe τ^m_{probe} and a soft motion envelope. The probes are complete counterfactual trajectory queries used to construct branch conditioning; CF-Flow does not directly refine the probe trajectory itself. Each probe queries CF-ICM under the same S_X.

For any queried trajectory τ, we summarize the three consequence classes with a nested gap/contact exposure vector

$$r_{i,h}(\tau) = \big[\pi_{i,h,n} + \pi_{i,h,c},\ \pi_{i,h,c}\big] = \big[1 - \pi_{i,h,f},\ \pi_{i,h,c}\big].$$

The first component is the predicted exposure to a non-free-clearance event (near gap or contact), while the second isolates contact exposure. We then construct

$$\begin{aligned} U_{i,h}(X,\tau) &= \phi_M(\mathrm{sg}[Z_{i,h}; r_{i,h}(\tau)]) + \phi_O(O_i) + e_h^{\mathrm{mem}}, \\ M_0^m &= \{U_{i,h}(X, \tau^m_{\mathrm{probe}})\}_{i,h}. \end{aligned} \tag{2}$$

Here O_i holds the observed attributes of object i, e_h^{mem} encodes physical time, and sg stops gradients. The coupled-intention embedding q_m attends to M_0^m to form the fixed strategy condition $c_m = q_m + \phi_c(\mathrm{AttnPool}(q_m, M_0^m))$. Thus each probe initializes a fixed branch-level strategy condition and an object–time-resolved consequence memory. A free-motion branch reuses the consequence memory of the neutral probe $m_{\mathrm{ref}} = (\mathrm{hold}, \mathrm{maintain})$, uses a dedicated learned branch embedding, and has no motion envelope.

Flow correction and memory refresh. For each branch, draw $\epsilon^m \sim \mathcal{N}(0, I)$, set $x_1 = \epsilon^m$, and integrate flow time σ from 1 to 0. The decoder embeds each of the H four-coordinate state tokens, attends across trajectory time and to S_X, and adds c_m. Its base head f_θ predicts a velocity field. The intermediate state also yields a one-step clean estimate $\hat{Y}_\sigma$ under the linear flow-matching path. At K physical-time anchors, the local trajectory representation of $\hat{Y}_\sigma$ and its physical-time derivative form memory-reader queries $Q_{\sigma,j}$. Attention over the active object–time memory M_k^m returns contexts $R_{\sigma,j}$. These local queries only *read* memory; CF-ICM instead receives the complete physical trajectory $\mathrm{ToPose}(\hat{Y}_\sigma)$ when refreshing memory. The per-head attention, relative-time bias, and masking are specified in Appendix A.2.

The retrieved contexts produce a bounded correction to the base field:

$$\begin{aligned} &\delta_{\sigma,j} = d_{\max} \tanh\big(g_\theta([Q_{\sigma,j}; R_{\sigma,j}; c_m])\big), \\ &v_\theta(x_\sigma, \sigma \mid S_X, c_m, M_k^m) = f_\theta(h_\sigma) + P_{\mathrm{interp}}\delta_\sigma. \end{aligned} \tag{3}$$

The interpolation maps K anchor corrections onto the H trajectory steps. At selected solver steps, CF-ICM re-evaluates the complete clean estimate. The resulting object–time consequences are passed through the memory construction in Eq. (2) to form the refreshed active memory M_{k+1}^m, which is held fixed until the next refresh. The strategy condition c_m and scene tokens S_X remain fixed. Integration gives $\hat{Y}^m = x_0$ and the physical candidate $\tau^m = \mathrm{ToPose}(\hat{Y}^m)$.

3.4 Candidate Selection and Training

Each completed candidate is queried once more by CF-ICM. A temperature $T_{\mathrm{sel}} > 0$, fitted on held-out executions, calibrates its logits. Using the resulting probabilities $\bar{\pi}^m$, define

$$\begin{aligned} \hat{C}_m &= \max_{i,h} \bar{\pi}^m_{i,h,c}, \\ \hat{N}_m &= \sum_h w_h \max_i \bar{\pi}^m_{i,h,n}, \qquad \sum_h w_h = 1. \end{aligned} \tag{4}$$

For contact screening, $\hat{C}_m$ takes the maximum predicted contact probability over all object–time pairs. For near-gap exposure, only the most critical object at each future step contributes, and the resulting per-step scores are aggregated with normalized temporal weights. This design avoids artificial risk accumulation with the number of surrounding objects or with repeated high-risk frames. The contact score is a screening proxy, not the probability of at least one contact. Among finite, decodable, kinematically valid candidates with $\hat{C}_m \le \varepsilon_c$, FlowPlan selects

$$\begin{aligned} J_m^{\mathrm{sel}} &= \lambda_C \hat{C}_m + \lambda_N \hat{N}_m + J_{\mathrm{motion}}(X, \hat{Y}^m) \\ &\quad + \lambda_{\mathrm{feas}} J_{\mathrm{feas}}(X, \hat{Y}^m; m), \\ m^\dagger &= \arg\min_{m \in \mathcal{M}_{\mathrm{acc}}} J_m^{\mathrm{sel}}. \end{aligned} \tag{5}$$

The selected physical trajectory is $\tau^\star = \tau^{m^\dagger}$. The motion cost covers route progress and smoothness; soft feasibility penalties cover dynamics, vehicle bounds, road and signal compliance, and the branch envelope. A vehicle-bounded braking trajectory is returned if no candidate remains after validity checking and contact screening.

Training has two stages with disjoint updates. Stage I fits the scene encoder and CF-ICM by object–time cross-entropy. They are frozen in Stage II, which trains only the flow generator:

$$L_{\mathrm{ICM}} = \mathbb{E}\left[\frac{1}{N_o H}\sum_{i=1}^{N_o}\sum_{h=1}^{H}\mathrm{CE}(\pi_{i,h}, \ell_{i,h})\right], \qquad L_{\mathrm{Flow}} = L_{\mathrm{FM}} + \lambda_{\mathrm{corr}} L_{\mathrm{corr}} + \lambda_{\mathrm{plan}}\mathbb{E}[J_{\mathrm{plan}}]. \tag{6}$$

L_{FM} matches the conditional velocity along $x_\sigma = (1-\sigma)Y^\star + \sigma\epsilon$, whose target velocity is $\epsilon - Y^\star$; L_{corr} uses additional trajectory-level generator targets. J_{plan} regularizes progress, smoothness, dynamics, road and signal compliance, and motion envelopes. Additional descriptions of the training pairs and branch memory updates appear in Appendices A.4 and A.5.

4 Experiments

We evaluate FlowPlan in closed-loop simulation on nuPlan to assess overall planning performance and analyze how counterfactual interaction consequences inform the executed trajectory.

4.1 Experimental Setup

We use the nuPlan benchmark (Karnchanachari et al., 2024), including Val14 with 1,118 scenarios, Test14, and the challenging Test14-hard split with 272 scenarios. Each split is evaluated in non-reactive (NR) and reactive (R) traffic. Other agents replay logged trajectories in NR simulation and respond through an Intelligent Driver Model policy in R simulation. We report the official aggregate closed-loop score and, for Val14-NR, the available collision, time-to-collision (TTC), drivable-area compliance, comfort, and progress metrics.

We compare against eight published planners: PDM-Open (Dauner et al., 2023), GameFormer without refinement (Huang et al., 2023), PlanTF (Cheng et al., 2024b), PLUTO without refinement (Cheng et al., 2024a), Diffusion Planner (Zheng et al., 2025), Flow Planner (Tan et al., 2025), FlowDrive (Wang et al., 2025), and DFP-FM (Zhang et al., 2026). DFP-FM denotes the flow-matching variant of Diffusion Forcing Planner, and baseline values are taken from the cited evaluations. Table 2 uses the reproduced Diffusion Planner Val14-NR row reported by DFP (marked *), so its aggregate and component metrics come from the same run.

4.2 Main Results

Overall performance. Among the methods included in Table 1, FlowPlan obtains the highest score in four of the six settings, including the three reactive columns. On Val14-R, Test14-hard-R, and Test14-R, it scores 87.22, 74.67, and 88.28, exceeding the corresponding FlowDrive scores by 1.85, 1.58, and 1.00 points, respectively. Under non-reactive traffic, it ranks second, third, and first within this comparison on Val14, Test14-hard, and Test14, respectively. The consistent gains in reactive traffic within the reported comparison support the central motivation of FlowPlan: modeling how alternative ego motions change interaction outcomes when surrounding agents respond. Table 2 further shows the highest listed collision and drivable-area scores, 97.05 and 99.11, while progress remains 96.50. These results highlight collision avoidance and road compliance as primary strengths in this comparison.

Qualitative case analysis. We examine a challenging traffic-cone intrusion scene in which static obstacles narrow the drivable corridor. Figure 2 compares FlowPlan with the strong FlowDrive baseline in a matched closed-loop rollout. Figures 3 and 4 analyze the coupled counterfactual probes and the corresponding CF-ICM outputs at a representative planning cycle from the same FlowPlan rollout.

Table 1: Closed-loop scores (%) on nuPlan. NR and R denote non-reactive and reactive traffic. Best and second-best results among the methods listed in each column are bold and underlined, respectively.

Planner	Val14		Test14-hard		Test14	
	NR	R	NR	R	NR	R
PDM-Open	53.53	54.24	33.51	35.83	52.81	57.23
GameFormer w/o refine.	13.32	8.69	7.08	6.69	11.36	9.31
PlanTF	84.27	76.95	69.70	61.61	85.62	79.58
PLUTO w/o refine.	89.04	80.01	70.03	59.74	89.90	78.62
Diffusion Planner	89.87	82.80	75.99	69.22	89.19	82.93
Flow Planner	90.43	83.31	76.47	70.42	89.88	82.93
FlowDrive	91.21	<u>85.37</u>	<u>77.86</u>	<u>73.09</u>	<u>91.02</u>	<u>87.28</u>
DFP-FM	**92.68**	81.30	**79.43**	67.94	90.62	83.59
FlowPlan (ours)	<u>91.96</u>	**87.22**	77.18	**74.67**	**92.87**	**88.28**

Table 2: Val14 non-reactive aggregate and component scores (%). The best value among the methods listed in each column is bold. *Reproduced Diffusion Planner run from DFP (Zhang et al., 2026).

Planner	Score	Collision	TTC	Drivable	Comfort	Progress
PLUTO w/o refine.	89.04	96.18	**93.28**	98.53	**96.41**	89.56
Diffusion Planner*	87.80	95.53	92.13	97.32	91.86	**100.00**
FlowDrive	91.21	96.02	90.79	98.39	95.62	95.48
FlowPlan (ours)	**91.96**	**97.05**	91.06	**99.11**	92.31	96.50

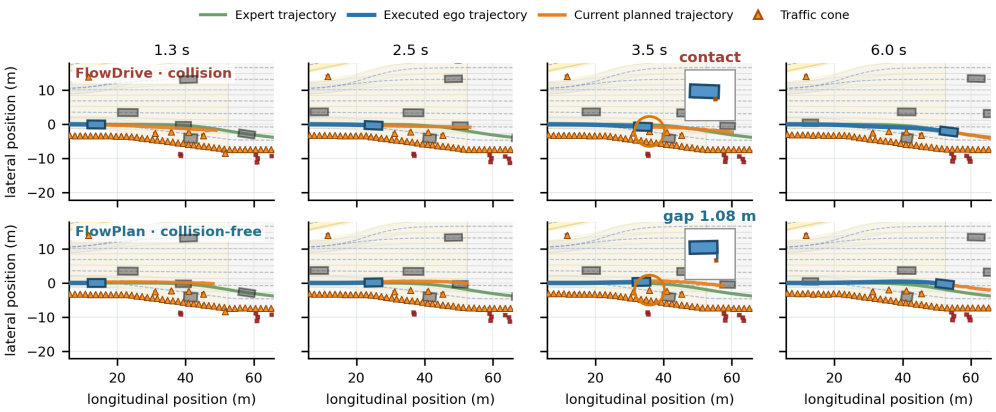

Figure 2: Matched closed-loop rollouts in a traffic-cone intrusion scene at 1.3, 2.5, 3.5, and 6.0 s. FlowDrive contacts a cone at 3.5 s, whereas FlowPlan maintains 1.08 m polygon clearance at the same instant and returns toward the route after passing the obstacle row. Green, blue, and orange curves denote the expert trajectory, executed ego trajectory, and current planned trajectory, respectively; triangles denote traffic cones.

In the matched rollout, as the corridor narrows, FlowDrive follows the route-directed path into the cone row and reaches zero clearance. FlowPlan initiates a localized lateral deviation before the conflict, clears the obstacle row, and then converges back toward the route.

Across the full rollout, FlowDrive incurs two at-fault collisions with static objects, whereas FlowPlan remains collision-free. The maneuver preserves the route objective while adapting the trajectory over the conflict interval.

All nine probes are evaluated from the same observed state. The left family moves away from the cone row, whereas the hold and right families remain closer to, or move toward, the narrowing corridor. These complete probe trajectories provide the alternative ego executions queried by CF-ICM.

Figure 4(a) compares how the predicted interaction exposure evolves over future time for the nine coupled behaviors. Within the hold family, deceleration shifts the rise in exposure later and makes the buildup more gradual than maintain or accelerate, although the underlying conflict remains. In contrast, the three leftward probes stay near zero across the horizon, suggesting that speed reduction mainly delays and softens the interaction, whereas lateral clearance removes it more directly in this scene.

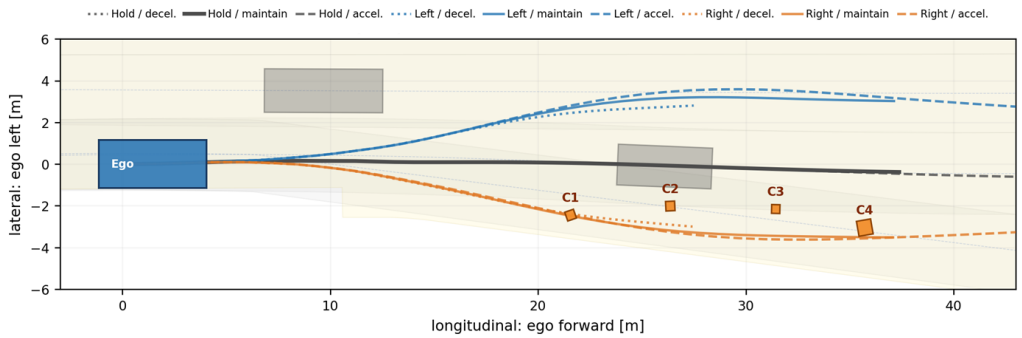

Figure 3: Coupled counterfactual probes at the analyzed planning cycle in the traffic-cone intrusion scene. The nine trajectories combine lateral hold, left, and right modes with longitudinal decelerate, maintain, and accelerate modes. C1–C4 identify the four cone objects used in the CF-ICM diagnosis.

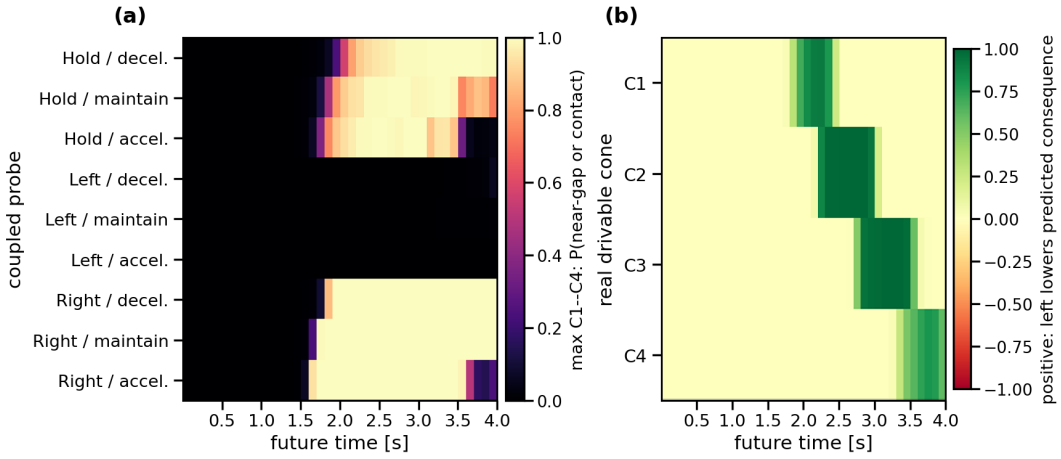

Figure 4: CF-ICM diagnosis for the same traffic-cone case. (a) Behavior-conditioned interaction exposure over future time, computed as the maximum predicted $P(\text{near-gap or contact})$ over C1–C4. (b) Object–time reduction in predicted consequence for left/maintain relative to hold/maintain. C1–C4 are the four labeled interaction objects in Figure 3; positive values favor left/maintain.

Figure 4(b) resolves the difference between left/maintain and hold/maintain by object and future time. C1–C4 are the four interaction objects marked in Figure 3; positive regions indicate lower predicted consequence under left/maintain. The improvement bands emerge from C1 through C4 in encounter order, showing that CF-ICM localizes where and when the leftward maneuver reduces the prospective conflicts.

4.3 Ablation Studies

We first isolate the effect of generation-time consequence conditioning on the complete Val14-R split. We compare three variants. *Scene-only* removes the coupled ego-motion probes, the branch-specific strategy condition and object–time consequence memory, and all intermediate memory refreshes. *Physics extrapolation* retains the probe set, branch structure, consequence interface, and refresh schedule, but replaces learned CF-ICM predictions with constant-velocity, constant-heading

(CV-CH) extrapolations mapped to the same free-clearance, near-gap, and contact representation. *FlowPlan* uses the learned CF-ICM. The variants otherwise share the scene encoder, CF-Flow architecture, training protocol, candidate budget, integration schedule, completed-candidate evaluation and selection, feasibility checks, and no-post-processing setting.

Table 3: Generation-time consequence-conditioning ablation on the complete Val14-R split. All planning metrics are percentages.

Variant	Consequence source	Score	Collision	TTC	Comfort	Progress
A: Scene-only	None	85.16	91.32	85.36	89.01	92.75
B: Physics extrapolation	CV-CH	86.78	**95.75**	88.65	88.79	87.68
C: FlowPlan	CF-ICM	**87.22**	94.44	**89.79**	**88.91**	**91.69**

Removing consequence conditioning reduces the aggregate score from 87.22 to 85.16 and produces the largest losses in the collision and TTC components. Replacing CF-ICM with CV-CH extrapolation recovers much of the safety score and attains the highest collision component, but it lowers progress to 87.68. In contrast, learned consequences give the best aggregate score and TTC while preserving substantially more progress. These results indicate that the gain is not explained by the probe/branch structure alone: the learned consequence model provides a more useful safety–progress trade-off than explicit kinematic extrapolation at generation time.

Aggregate closed-loop scores do not directly reveal whether a planner changes its behavior for the *right reason* when one interaction is altered. We therefore evaluate the same three variants on nuPlan-CIP, a controlled diagnostic cohort derived from Val14-R. The cohort contains 140 frozen base scenes, with 10 scenes from each of the 14 official scenario types. For each base scene, we construct a native world ($z = 0$) and attempt a conflict-increasing arm ($z = +$), a conflict-removing arm ($z = -$), and an irrelevant-actor control ($z = N$). The target actor and intervention are fixed before planner evaluation, and only interventions that pass a fixed-ego reactive-IDM audit are retained. The deterministic scene sampling, intervention dynamics, geometric pre-screening, and validity audit are detailed in Appendix C.

Table 4: nuPlan-CIP closed-loop scores and gap-response metric in reactive traffic. Parentheses give the number of retained scenes for each arm. CIR-Gap measures the fraction of intervention-induced minimum-gap degradation recovered by replanning; higher is better.

Variant	$z = 0$ (140)	$z = +$ (46)	$z = -$ (21)	$z = N$ (114)	CIR-Gap ↑ (46)
A: Scene-only	87.56	81.05	87.59	87.52	0.34
B: Physics extrapolation	89.02	87.51	88.89	89.01	0.75
C: FlowPlan	**89.23**	**88.76**	**89.22**	**89.17**	**0.83**

The conflict-removing and irrelevant-actor arms remain close to each variant's native-world score, while the conflict-increasing arm provides the clearest separation. FlowPlan retains 88.76 points in $z = +$ versus 89.23 in the native world, a drop of only 0.47 points; the corresponding drops are 1.51 points for physics extrapolation and 6.51 points for scene-only. CIR-Gap shows the same ordering: FlowPlan recovers 83% of the intervention-induced minimum-gap degradation, compared with 75% for physics extrapolation and 34% for scene-only. The agreement between the aggregate score and the interaction-specific recovery metric indicates that learned CF-ICM consequences help the planner adapt to a targeted change in interaction geometry while remaining stable when the conflict is removed or an irrelevant actor is perturbed.

5 Conclusion

We presented FlowPlan, a consequence-conditioned flow-matching planner in which CF-ICM evaluates alternative ego trajectories and exposes object–time interaction consequences to generation. Coupled lateral–longitudinal probes initialize strategy conditions and consequence memories that guide the flow field and are refreshed as candidates evolve. On nuPlan, FlowPlan performs strongly across the reactive evaluations included in our comparison, while the ablations and nuPlan-CIP show improved safety–progress trade-offs and recovery from targeted interaction perturbations. These results support dynamically updated interaction consequences as a useful planning signal; future work will broaden intervention coverage, reactive-agent diversity, and validation beyond simulation.

AI USE STATEMENT

Generative AI tools were used only to assist with language editing and with searching for and organizing potentially relevant literature. The authors independently checked the cited literature against the original sources and reviewed all AI-assisted wording before inclusion in the manuscript. Generative AI was not used to design the method or experiments, implement the system, generate or modify experimental data, produce simulation outcomes or evaluation labels, analyze the reported results, or make scientific conclusions. The authors take full responsibility for the final content of this work.

ETHICS STATEMENT

This work studies autonomous-driving planning algorithms in simulation using the nuPlan benchmark and does not introduce new human-subject experiments, collect new personal data, or deploy a system in the real world. The experiments use existing benchmark data and simulation interfaces under their applicable terms. As with autonomous-driving research generally, insufficiently validated planning systems could create safety risks if deployed outside the intended research setting; the results reported here are simulation-based and should not be interpreted as evidence of deployment readiness. We are not aware of additional ethics concerns specific to this study.

REPRODUCIBILITY STATEMENT

The main paper describes the FlowPlan method and experimental protocol, and the appendix provides additional implementation details for the consequence model, coupled probes, object–time memory, training-data construction, calibration, candidate budget, and memory refresh. The anonymous code release contains the core inference modules used by FlowPlan, including the scene encoder, the coupled-probe generator and kinematic rollout, CF-ICM inference, consequence-memory construction, the consequence-conditioned flow adapter with relative-time memory reading and memory refresh, and the top-level inference wrapper for producing candidate trajectories together with their consequence predictions. The release also includes the dependency specification needed to set up these modules. The code is available at `https://anonymous.4open.science/r/Flowplan-7B00/`.

REFERENCES

Anurag Ajay, Yilun Du, Abhi Gupta, Joshua B. Tenenbaum, Tommi S. Jaakkola, and Pulkit Agrawal. Is conditional generative modeling all you need for decision-making? In *International Conference on Learning Representations*, 2023.

Michael S. Albergo, Nicholas M. Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions, 2023.

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In *Proceedings of Robotics: Science and Systems*, 2019. doi: 10.15607/RSS.2019.XV.031.

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, 2020.

Dian Chen and Philipp Krähenbühl. Learning from all vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17222–17231, 2022.

Dian Chen, Brady Zhou, Vladlen Koltun, and Philipp Krähenbühl. Learning by cheating, 2019.

Ricky T. Q. Chen, Yulia Rubanova, Jesse Bettencourt, and David Duvenaud. Neural ordinary differential equations. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Jie Cheng, Yingbing Chen, and Qifeng Chen. Pluto: Pushing the limit of imitation learning-based planning for autonomous driving, 2024a.

Jie Cheng, Yingbing Chen, Xiaodong Mei, Bowen Yang, Bo Li, and Ming Liu. Rethinking imitation-based planners for autonomous driving. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2024b. doi: 10.1109/ICRA57147.2024.10611364.

Cheng Chi, Siyuan Feng, Yilun Du, Zhenjia Xu, Eric Cousineau, Benjamin Burchfiel, and Shuran Song. Diffusion policy: Visuomotor policy learning via action diffusion. In *Proceedings of Robotics: Science and Systems*, 2023. doi: 10.15607/RSS.2023.XIX.026.

Kashyap Chitta, Aditya Prakash, and Andreas Geiger. Neat: Neural attention fields for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15793–15803, 2021.

Kashyap Chitta, Aditya Prakash, Bernhard Jaeger, Zehao Yu, Katrin Renz, and Andreas Geiger. Transfuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):12878–12895, 2023. doi: 10.1109/TPAMI.2022.3200245.

Felipe Codevilla, Matthias Müller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2018.

Daniel Dauner, Marcel Hallgarten, Andreas Geiger, and Kashyap Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Proceedings of the Conference on Robot Learning*, volume 229 of *Proceedings of Machine Learning Research*, pp. 1268–1281, 2023.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, 2021.

Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R. Qi, Yin Zhou, Zoey Yang, Aurélien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9710–9719, 2021.

Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 11525–11533, 2020.

Will Grathwohl, Ricky T. Q. Chen, Jesse Bettencourt, Ilya Sutskever, and David Duvenaud. Ffjord: Free-form continuous dynamics for scalable reversible generative models. In *International Conference on Learning Representations*, 2019.

Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 15303–15312, 2021.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, volume 33, pp. 6840–6851, 2020.

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*, 2023.

Zhao Huang, Bowen Zhang, Zhongzhu Li, and Di Lin. Occdriver: Future occupancy guided dual-branch trajectory planner in autonomous driving. In *International Conference on Learning Representations*, 2026.

Zhiyu Huang, Haochen Liu, and Chen Lv. Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023.

Zhiyu Huang, Peter Karkus, Boris Ivanovic, Yuxiao Chen, Marco Pavone, and Chen Lv. DTPP: Differentiable joint conditional prediction and cost evaluation for tree policy planning in autonomous driving. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 6806–6812, 2024.

Michael Janner, Yilun Du, Joshua B. Tenenbaum, and Sergey Levine. Planning with diffusion for flexible behavior synthesis. In *Proceedings of the 39th International Conference on Machine Learning*, pp. 9902–9915, 2022.

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*, pp. 21983–21994, 2023.

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*, pp. 8340–8350, 2023.

Napat Karnchanachari, Dimitris Geromichalos, Kok Seang Tan, Nanxiang Li, Christopher Eriksen, Shakiba Yaghoubi, Noushin Mehdipour, Gianmarco Bernasconi, Whye Kit Fong, Yiluan Guo, and Holger Caesar. Towards learning-based planning: The nuplan benchmark for real-world autonomous driving, 2024.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *European Conference on Computer Vision*, 2020.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *International Conference on Learning Representations*, 2023.

Haochen Liu, Li Chen, Yu Qiao, Chen Lv, and Hongyang Li. Reasoning multi-agent behavioral topology for interactive autonomous driving. In *Advances in Neural Information Processing Systems*, 2024.

Xingchao Liu, Chengyue Gong, and Qiang Liu. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *International Conference on Learning Representations*, 2023.

Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

Nigamaa Nayakanti, Rami Al-Rfou, Aurick Zhou, Kratarth Goel, Khaled S. Refaat, and Benjamin Sapp. Wayformer: Motion forecasting via simple and efficient attention networks, 2022.

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.

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*, 2021.

Katrin Renz, Kashyap Chitta, Otniel-Bogdan Mercea, A. Sophia Koepke, Zeynep Akata, and Andreas Geiger. Plant: Explainable planning transformers via object-level representations, 2022.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 1530–1538, 2015.

Nicholas Rhinehart, Rowan McAllister, and Sergey Levine. Deep imitative models for flexible inference, planning, and control. In *International Conference on Learning Representations*, 2020.

Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*, 2020.

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.

Hao Shao, Letian Wang, Ruobing Chen, Hongsheng Li, and Yu Liu. Safety-enhanced autonomous driving using interpretable sensor fusion transformer. In *Proceedings of the Conference on Robot Learning*, volume 205 of *Proceedings of Machine Learning Research*, pp. 726–737, 2023.

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.

Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Mtr++: Multi-agent motion prediction with symmetric scene modeling and guided intention querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(5):3955–3971, 2024. doi: 10.1109/TPAMI.2024.3352811.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021a.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2021b.

Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32211–32252, 2023.

Qiao Sun, Xin Huang, Junru Gu, Brian C. Williams, and Hang Zhao. M2i: From factored marginal trajectory prediction to interactive prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6543–6552, 2022.

Simon Suo, Sebastian Regalado, Sergio Casas, and Raquel Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10400–10409, 2021.

Tianyi Tan, Yinan Zheng, Ruiming Liang, Zexu Wang, Kexin Zheng, Jinliang Zheng, Jianxiong Li, Xianyuan Zhan, and Jingjing Liu. Flow matching-based autonomous driving planning with advanced interactive behavior modeling. In *Advances in Neural Information Processing Systems*, 2025.

Chen Tang, Wei Zhan, and Masayoshi Tomizuka. Interventional behavior prediction: Avoiding overly confident anticipation in interactive prediction, 2022.

Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*, 2024.

Lingguang Wang, Ömer Şahin Taş, Marlon Steiner, and Christoph Stiller. Flowdrive: Moderated flow matching with data balancing for trajectory planning, 2025.

Wei Wu, Xiaoxin Feng, Ziyan Gao, and Yuheng Kan. Smart: Scalable multi-agent real-time motion generation via next-token prediction. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

Brian Yang, Huangyuan Su, Nikolaos Gkanatsios, Tsung-Wei Ke, Ayush Jain, Jeff Schneider, and Katerina Fragkiadaki. Diffusion-es: Gradient-free planning with diffusion for autonomous and instruction-guided driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M. Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9813–9823, 2021.

Zehan Zhang, Yaoyi Li, Neng Zhang, and Jia Cai. Diffusion forcing planner: History-annealed planning with time-dependent guidance for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 39796–39805, 2026.

Zhejun Zhang, Alexander Liniger, Dengxin Dai, Fisher Yu, and Luc Van Gool. Trafficbots: Towards world models for autonomous driving simulation and motion prediction. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2023.

Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, Congcong Li, and Dragomir Anguelov. Tnt: Target-driven trajectory prediction. In *Proceedings of the Conference on Robot Learning*, volume 155 of *Proceedings of Machine Learning Research*, pp. 895–904, 2021.

Yinan Zheng, Ruiming Liang, Kexin Zheng, Jinliang Zheng, Liyuan Mao, Jianxiong Li, Weihao Gu, Rui Ai, Shengbo Eben Li, Xianyuan Zhan, and Jingjing Liu. Diffusion-based planning for autonomous driving with flexible guidance. In *International Conference on Learning Representations*, 2025.

Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8823–8833, 2022.

A Additional Method Details

A.1 Scene and consequence-network equations

The main text defines the complete physical trajectory query $\tau = \{(p_{x,h}, p_{y,h}, \psi_h)\}_{h=1}^{H}$ and its four-coordinate representation Y. The reference implementation starts from the same physical poses and augments the per-step query features with the displacement from the previous step. Let $\Delta p_1 = (0, 0)$ and $\Delta p_h = (p_{x,h} - p_{x,h-1}, p_{y,h} - p_{y,h-1})$ for $h > 1$. After fixed feature-wise scaling, the trajectory token is

$$q_h = \phi_q([p_{x,h}, p_{y,h}, \sin\psi_h, \cos\psi_h, \Delta p_{x,h}, \Delta p_{y,h}]) + \phi_t(\gamma_h), \tag{7}$$

where γ_h is a deterministic physical-time encoding containing normalized time and first- and second-frequency sinusoidal terms. This is an internal parameterization of the complete trajectory query in Eq. (1); the CF-ICM output remains an object–time token and a three-way consequence distribution for each queried trajectory.

For each valid surrounding object, the current object state is normalized and fused with a pooled scene summary. The resulting object token then attends to the full scene-token set. CF-ICM additionally constructs ego-relative geometry for every object–time pair. Writing the object position and heading as $(p^o_{x,i}, p^o_{y,i}, \psi^o_i)$, define

$$\begin{aligned} \Delta x_{i,h} &= p^o_{x,i} - p_{x,h}, \qquad \Delta y_{i,h} = p^o_{y,i} - p_{y,h}, \\ d^{\parallel}_{i,h} &= \cos\psi_h\, \Delta x_{i,h} + \sin\psi_h\, \Delta y_{i,h}, \\ d^{\perp}_{i,h} &= -\sin\psi_h\, \Delta x_{i,h} + \cos\psi_h\, \Delta y_{i,h}, \\ g_{i,h} &= [\Delta x_{i,h}, \Delta y_{i,h}, d^{\parallel}_{i,h}, d^{\perp}_{i,h}, \|\Delta p_{i,h}\|_2, \sin(\psi^o_i - \psi_h), \cos(\psi^o_i - \psi_h)]. \end{aligned} \tag{8}$$

The pair representation is formed additively from the contextual object token, the trajectory token q_h, and an MLP embedding of $g_{i,h}$, followed by layer normalization. At each future time, multi-head attention is applied over valid surrounding objects. The resulting per-object sequence is then processed along physical time by a bidirectional GRU and a categorical head:

$$\begin{aligned} P_{i,h} &= \mathrm{LN}(o_i + q_h + \phi_{\mathrm{rel}}(g_{i,h}))\,, \\ \widetilde{P}_{:,h} &= P_{:,h} + \mathrm{MHA}_{\mathrm{obj}}(P_{:,h}, P_{:,h}, P_{:,h}), \\ Z_{i,1:H} &= \mathrm{BiGRU}(\widetilde{P}_{i,1:H}), \qquad \pi_{i,h} = \mathrm{softmax}(W_r Z_{i,h} + b_r). \end{aligned} \tag{9}$$

Masks remove padded objects from attention and set their output tokens and logits to zero.

Coupled-probe realization and kinematic projection. The nine semantic probes in the main text are implemented as smooth desired control profiles that operationalize the lateral and longitudinal intentions. The implementation uses acceleration and curvature as control coordinates. A route-following curvature prior supplies the nominal path shape. The longitudinal modes add smooth acceleration priors with nominal levels -1.8, 0, and $1.2\,\mathrm{m/s^2}$ for decelerate, maintain, and accelerate, respectively. The lateral modes add a smooth curvature wave with nominal amplitudes 0, $+0.024$, and $-0.024\,\mathrm{m^{-1}}$ for hold, left, and right. Thus the nine probes form the full 3×3 Cartesian product used in the main text; the reported implementation uses these semantic control priors for the probe queries rather than learning a separate executable probe policy.

Before integration, the desired controls are projected to an admissible set. The implementation uses acceleration limits $[-4, 2]\ \mathrm{m/s^2}$, curvature magnitude at most $0.2\ \mathrm{m^{-1}}$, jerk magnitude at most $4\ \mathrm{m/s^3}$, and lateral acceleration at most $5\ \mathrm{m/s^2}$. The projected controls are integrated from the current ego state with $\Delta t = 0.1$ s for 40 future steps (4 s). These probes are used as complete CF-ICM queries for consequence conditioning, consistent with the main text; they are not treated as final executable plans.

A.2 Object–time memory reader

For a queried trajectory, CF-ICM returns the object–time token $Z_{i,h}$ and probabilities $\pi_{i,h}$. We use the same nested exposure representation as in the main text,

$$r_{i,h} = [\pi_{i,h,n} + \pi_{i,h,c},\ \pi_{i,h,c}], \tag{10}$$

and embed the concatenation of the stopped-gradient token and exposure vector. A learned semantic/probe embedding and a physical-time embedding are added to the token; an object-attribute projection can be included in the same additive form as Eq. (2). Invalid probe/object/time entries are masked to zero before attention. Thus the implementation preserves the main paper's object–time memory semantics while retaining the semantic identity and physical time of each memory item.

For local memory reads, the implementation uses $K = 10$ physical-time anchors. At a flow step with state x_σ, the base field produces a one-step clean estimate used only to construct the local reader query. The future portion of this estimate and its physical-time derivative are linearly interpolated to the K anchors. Endpoint derivatives use one-sided finite differences, while interior derivatives use centered differences. Concatenating the interpolated state, derivative, and flow-time embedding gives the local query $Q_{\sigma,j}$.

For one attention head, with the head index omitted,

$$\begin{aligned} Q_{\sigma,j} &= \phi_{\mathrm{mq}}([\mathrm{Interp}(\hat{Y}_\sigma, \bar{t}_j); \partial_t \mathrm{Interp}(\hat{Y}_\sigma, \bar{t}_j); e^{\mathrm{flow}}(\sigma)]), \\ \alpha_{\sigma,j,i,h} &= \mathrm{softmax}_{i,h}\left[\frac{(W_Q Q_{\sigma,j})^\top W_K U_{i,h}}{\sqrt{d_{\mathrm{head}}}} + b_{\mathrm{rel}}\left(\frac{\bar{t}_j - t_h}{T}\right)\right], \\ r_{\sigma,j} &= \sum_{i,h} \alpha_{\sigma,j,i,h} W_V U_{i,h}. \end{aligned} \tag{11}$$

The relative-time bias is produced by a small MLP from the normalized time offset. Softmax excludes invalid entries; when no valid evidence is present, the retrieved context is set to zero. The multi-head contexts are concatenated and projected back to the decoder hidden width before producing the bounded correction in Eq. (3).

A.3 Ranking-cost components

The selection rule is given in Eq. (5) of the main text. Its motion term combines route-progress deficit with jerk and curvature-rate penalties. Its soft feasibility term covers kinematic consistency, vehicle bounds, road and signal constraints, and the branch motion envelope. Candidates that cannot be decoded or grossly violate kinematic limits are removed before ranking.

A.4 Training data and consequence labels

Stage-I CF-ICM data construction. We sample logged scenes from the nuPlan training split and retain the logged ego behavior as the factual query. To broaden the counterfactual coverage, we add reactive-simulation queries from two complementary sources. The first source uses structured lateral–longitudinal intention trajectories generated with the same kinematic construction as the coupled probes. The second source uses ego trajectories sampled from a flow-matching planner conditioned on the logged scene. Both types of alternative ego trajectories are executed in reactive simulation so that surrounding traffic can respond to the changed ego motion. Reactive rollouts that lead to contact are retained rather than filtered, because they provide direct supervision for adverse interaction outcomes.

For every tuple, the complete ego trajectory is paired with object–time consequence labels obtained from the ego and surrounding-object trajectories of the same execution. Contact is determined using the standard nuPlan contact logic. For object–time pairs without contact, we use the clearance returned by the corresponding nuPlan geometry routines: a pair is labeled *near* when $d < 1.0$ m and *free* otherwise. Thus the three labels are mutually exclusive and ordered as free, near gap, and contact.

Stage-II planning data. Stage II uses the original nuPlan training set without the reactive counterfactual expansion used for Stage I. The scene encoder and CF-ICM are kept fixed as described in the main text while the consequence-conditioned flow planner is trained on the planning data.

Implementation dimensions. The reported checkpoints use the default dimensions of the released implementation. The flow backbone follows the FlowDrive-style configuration (Wang et al., 2025); the consequence modules use the dimensions listed below.

Component	Setting
Planning horizon	40 future steps at 0.1 s (4 s)
Scene encoder	hidden width 192; depth 3; 6 attention heads
CF-ICM pair/object representation	model width 96; 4 attention heads; dropout 0
CF-ICM temporal encoder	1-layer bidirectional GRU; 32 hidden units per direction; 64-D output token
CF-Flow backbone	3 DiT blocks; hidden width 192; 6 heads; dropout 0.1
Consequence adapter / memory reader	hidden width 128; 4 heads; $K = 10$ physical-time anchors
Flow correction bound	$d_{\max} = 0.12$ in normalized trajectory-state coordinates
Flow schedule	2000 training timesteps; 8 inference steps; one refresh after step 5

A.5 Inference order and memory refresh

The released inference implementation uses an eight-step flow solve and one CF-ICM memory refresh after the fifth field evaluation (refresh fraction $5/8 = 0.625$). This is a concrete instance of the selected solver-step refresh rule described in the main text. At every solver step, the frozen base decoder first produces f_θ. The local memory reader constructs its query from the base-field clean estimate

$$\hat{x}_0^{\text{base}} = x_\sigma - \sigma f_\theta, \tag{12}$$

which avoids defining the local read recursively through its own correction. The memory context and fixed strategy condition then produce the bounded correction, giving the full velocity $v_\theta = f_\theta + \delta_\theta$.

At the refresh step, the complete trajectory sent to CF-ICM is reconstructed from this corrected velocity,

$$\hat{x}_0^{\text{refresh}} = x_\sigma - \sigma v_\theta. \tag{13}$$

The future state is converted to physical poses, CF-ICM is evaluated on the complete trajectory, and the resulting consequences replace the active memory for the remainder of the solve. The initial strategy condition is not recomputed after refresh. The numerical scheduler update is then applied using the same corrected velocity. In pseudocode, the relevant order is

1. evaluate the base field f_θ;
2. build local reader queries and retrieve active memory;
3. compute the bounded correction and v_θ;
4. at step 5, reconstruct the complete clean trajectory and query CF-ICM;
5. replace the active consequence memory;
6. advance the flow solver with v_θ.

After the final solver step, decoded physical trajectories can be queried by CF-ICM again for the completed-candidate consequence scores used by the selection procedure in the main text.

Candidate budget. The reported implementation decodes 15 stochastic trajectory candidates per planning cycle. Nine samples are associated with the nine coupled intentions, and six additional generic/free-motion samples provide neutral alternatives. We include this extra generic budget because, in simple scenes, even the nominal keep/maintain probe can introduce a small unnecessary deviation through its kinematic rollout. The additional samples reduce dependence on that probe shape while preserving the same consequence-conditioned decoding and final candidate evaluation. Each of the 15 candidates starts from an independently sampled Gaussian latent in the default inference path.

A.6 Model selection and calibration

The planner uses a calibrated CF-ICM output to screen completed candidates by a contact-exposure proxy before comparing feasible trajectories on near-gap exposure, motion quality, and feasibility penalties. The contact score is a screening proxy; its sum over object–time pairs need not equal the probability of at least one contact event.

We calibrate CF-ICM with standard scalar temperature scaling on held-out validation data. Let $z_{i,h}$ denote the three-class logits and $y_{i,h}$ the consequence label for a valid object–time pair. We fit one

positive scalar T by minimizing multiclass negative log-likelihood,

$$T^* = \arg\min_{T>0} - \sum_{(i,h)\in\mathcal{V}} \log \left[\mathrm{softmax}(z_{i,h}/T)\right]_{y_{i,h}}, \tag{14}$$

where $\mathcal{V}$ contains valid labeled object–time pairs from the held-out split. The fitted scalar is then fixed at inference time and is applied before constructing the near-gap/contact exposure used by memory and candidate selection.

B Evaluation Protocol and Ablation Design

B.1 nuPlan protocol

We use Val14 (1,118 scenarios), Test14, and Test14-hard (272 scenarios), each in non-reactive and reactive traffic. In non-reactive simulation, surrounding agents replay logged trajectories; in reactive simulation, they respond through an Intelligent Driver Model policy.

B.2 Ablation controls

We consider (A) scene-only generation, (B) a physics-extrapolation consequence interface using constant velocity and heading, and (C) learned CF-ICM consequences. The intended comparison holds the scene encoder, generator architecture, training protocol, candidate budget, integration schedule, completed-candidate evaluation, feasibility checks, and post-processing setting fixed.

C nuPlan-CIP Construction and Audit

C.1 Diagnostic role and deterministic base-scene sampling

nuPlan-CIP is a diagnostic cohort derived from the 1,118 Val14 scenarios and is used only with reactive traffic. It is not an independent test split and does not replace the official Val/Test evaluations. Its purpose is to test whether a controlled change to one surrounding actor produces an ego response consistent with the resulting change in interaction geometry.

We freeze 10 scenes from each of the 14 official scenario types, giving 140 base scenes. Selection uses only the scenario type, scenario token, and a fixed seed; it does not depend on planner outputs, scores, or the ease of constructing an intervention. Within each type, scenarios are sorted by

$$h(s) = \mathrm{SHA256}(\mathrm{seed} \,\|\, \text{“|”} \,\|\, \mathrm{type}(s) \,\|\, \text{“|”} \,\|\, \mathrm{token}(s)), \tag{15}$$

and the first 10 are retained. The 14 categories are

behind_long_vehicle	starting_left_turn
changing_lane	starting_right_turn
following_lane_with_lead	starting_straight_traffic_light_intersection_traversal
high_lateral_acceleration	stationary_in_traffic
high_magnitude_speed	stopping_with_lead
low_magnitude_speed	traversing_pickup_dropoff
near_multiple_vehicles	waiting_for_pedestrian_to_cross

C.2 Controlled traffic worlds

For each base scene, one IDM-controlled target actor and its intervention are selected and frozen before planner evaluation. We consider four traffic worlds: $z = 0$ is the native IDM world; $z = +$ modifies the target so that the minimum ego–target clearance decreases relative to $z = 0$; $z = -$ modifies the same interaction directionally so that the minimum clearance increases; and $z = N$ perturbs an actor that remains well separated from the reference ego trajectory, serving as an irrelevant-actor control. Only the selected target's speed update is modified. The ego initial state, map, route, traffic signals, vehicle dimensions, target identity, and random seed remain fixed, while all other actors continue to follow the native IDM policy. The signs $+$ and $-$ denote the resulting interaction direction, not a fixed acceleration/deceleration label.

C.3 SPEED INTERVENTION IN REACTIVE SIMULATION

Let v_k be the target actor's realized speed at the previous simulator step, $\tilde{v}_{k+1}$ the native IDM speed update for the current step, v_0 the initial IDM speed, and s_k the progress along the actor's map path. We use horizon 4.0 s, time step $\Delta t = 0.1$ s, intervention onset $\tau = 0.5$ s, and acceleration magnitude $a = 2.0\,\mathrm{m/s^2}$. Candidate targets must have initial speed at least $1.5\,\mathrm{m/s}$ and a valid IDM path. The simulator first computes the native IDM update, after which the intervention modifies the end-of-step speed. Position is not rewritten within the current step; on a valid path, progress evolves as

$$s_{k+1} = s_k + v_k \Delta t. \tag{16}$$

Thus the modified speed affects position only from the following step, avoiding instantaneous actor displacement.

For a deceleration operation, define

$$L_{\mathrm{dec}} = \max(0.8, 0.55 v_0), \tag{17}$$

and, for $\eta = t_{k+1} - \tau \geq 0$,

$$v_{k+1}^{\mathrm{dec}} = \min[\tilde{v}_{k+1},\, \max\{L_{\mathrm{dec}},\, v_0 - a\eta,\, v_k - a\Delta t\}]\,. \tag{18}$$

This intervention only adds braking; if the native IDM already brakes more strongly, the lower native speed is preserved. The quantity L_{dec} is the target of the intervention envelope rather than a hard lower bound on the realized speed.

For an acceleration operation, let

$$U_{\mathrm{acc}} = \max(v_0, 1.25 v_0). \tag{19}$$

If $\tilde{v}_{k+1} < v_k - 0.25 a \Delta t$, the native IDM is already braking materially and its safety response is preserved, i.e., $v_{k+1} = \tilde{v}_{k+1}$. Otherwise,

$$v_{k+1}^{\mathrm{acc}} = \max[\tilde{v}_{k+1},\, \min\{U_{\mathrm{acc}},\, v_0 + a\eta,\, v_k + a\Delta t\}]\,. \tag{20}$$

The operation is applied at every simulator step after onset rather than by changing only the initial observation or replacing the target with a constant-velocity model.

C.4 GEOMETRIC PRE-SCREENING AND FIXED-EGO AUDIT

Candidate interventions are first searched using the recorded ego poses over 4 s and the target's IDM map path. This stage uses approximate control/acceleration/deceleration speed envelopes only to reduce the search space; final validity is always determined by a reactive-IDM rollout with the intervention above.

Let $B_e(k)$ and $B_j(k)$ be the oriented ego and target vehicle polygons at step k. For non-overlapping polygons, d_k is their Euclidean distance; contact is assigned the sentinel value $d_k = -0.25$ m. Define the minimum-clearance proxy

$$D^z = \min_{k=1,\ldots,40} d_k^z, \qquad \Delta D^z = D^z - D^0. \tag{21}$$

During geometric pre-screening, a $z = +$ candidate must satisfy $\hat{D}_0 - \hat{D}_z \geq 0.5$ m and $\hat{D}_z \leq 10$ m. Candidates are ordered by larger clearance reduction, smaller post-intervention clearance, target token, and operation, with at most eight retained. The $z = -$ candidate pool is generated by applying the opposite speed operation to the same target actors identified by the $z = +$ search. The $z = N$ search requires $\hat{D}_0, \hat{D}_N \geq 15$ m, $|\Delta \hat{D}_N| \leq 2$ m, and at least $0.5\,\mathrm{m/s}$ terminal speed change, with at most 12 candidates retained. The current negative-control construction is geometrically constrained but is not strictly matched in actor class, distance, and speed.

Every candidate then undergoes an audit in which the native and intervened traffic worlds are run with the *same recorded ego trajectory*; the planner under evaluation is not called. The target must actually be modified and satisfy

$$E_v = \max_k |v_{j,k}^z - v_{j,k}^0| \geq 0.3\,\mathrm{m/s}, \qquad E_p = \max_k \|p_{j,k}^z - p_{j,k}^0\|_2 \geq 0.3\,\mathrm{m}. \tag{22}$$

The arm-specific audit additionally requires

$$\begin{aligned} z = +: &\quad \Delta D^{+} \le -0.5\,\mathrm{m}, \qquad D^{+} \le 10\,\mathrm{m}, \\ z = -: &\quad \Delta D^{-} \ge 0.5\,\mathrm{m}, \\ z = N: &\quad D^{0}, D^{N} \ge 15\,\mathrm{m}, \qquad |\Delta D^{N}| \le 2\,\mathrm{m}. \end{aligned} \tag{23}$$

After this audit, the current cohort contains 46 valid $z = +$ arms, 21 valid $z = -$ arms, and 114 valid $z = N$ arms in addition to the 140 native worlds.

C.5 Planner evaluation and CIR-Gap

For each valid $z = +$ arm and planner m, we first obtain the factual ego trajectory e_0^m in the native $z = 0$ world. Let D_0^m denote its minimum ego–object gap in that native world. We then replay the same fixed trajectory e_0^m in the $z = +$ world without replanning, giving minimum gap $D^m_{\mathrm{replay},+}$, and separately run the planner with free replanning in the same perturbed world, giving $D^m_{\mathrm{replan},+}$. CIR-Gap measures the fraction of the intervention-induced gap loss recovered by replanning:

$$\text{CIR-Gap}^m = \min\left(1,\ \frac{D^m_{\mathrm{replan},+} - D^m_{\mathrm{replay},+}}{D_0^m - D^m_{\mathrm{replay},+}}\right). \tag{24}$$

Thus, a value of 0 means that replanning recovers none of the lost clearance, while a value of 1 means that it restores the native minimum gap or exceeds it; values above one are clipped to one. For example, if the native minimum gap is 1.0 m, replay in $z = +$ reduces it to 0.3 m, and replanning raises it to 0.9 m, then CIR-Gap is $(0.9 - 0.3)/(1.0 - 0.3) = 0.857$. The main text reports CIR-Gap together with the aggregate closed-loop score in all four worlds. The $z = -$ arm tests behavior after conflict removal, whereas $z = N$ tests stability to an irrelevant-actor perturbation.