# EVOBELIEF: EVOLVING BEHAVIORAL BELIEFS VIA CONFORMAL MULTIMODAL DIFFUSION

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

In driving, the ego car continually negotiates with other road users under imperfect perception, and the same observations often admit more than one reasonable maneuver—at a merge, a yield, or an unprotected turn the scene does not yet reveal whether to proceed or hold. Such interactive, ambiguous moments call not for a single confident trajectory, but for an explicit account of the competing behaviors the scene still admits. Diffusion-based trajectory planners can represent multiple plausible future behaviors, yet existing methods largely exploit this multimodality for trajectory generation rather than behavioral uncertainty estimation. We present EVOBELIEF, which treats the multimodal distribution of generated trajectories as explicit evidence of alternative driving behaviors, such as waiting or committing, and converts this evidence into safety-aware decisions. Since sample frequencies are not calibrated probabilities, EVOBELIEF uses split conformal prediction to calibrate behavioral evidence into statistically controlled safety decisions, with an online adaptive mechanism for deployment shift. On Bench2Drive, EVOBELIEF improves success rate from $89.25\%$ to $91.36\%$ over state-of-the-art. More broadly, EVOBELIEF offers a new paradigm that reads generative multimodality as an evolving behavioral belief and turns it into safe interactive driving decisions, rather than treating diffusion merely as a trajectory generator.

## 1 INTRODUCTION

End-to-end driving policies plan by predicting the ego vehicle's future trajectory from raw perception, and are increasingly judged in *closed loop* (Dauner et al., 2023; Nguyen et al., 2026; Renz et al., 2025), where the ego car must repeatedly negotiate with other road users whose intentions it cannot directly observe (Zheng et al., 2024; Kedia et al., 2023; Lee et al., 2025). Perception is itself imperfect, so every estimate of the surrounding scene carries uncertainty that propagates into the ego's judgment about what will happen next (Lekeufack et al., 2024); and driving is inherently multi-choice—at a merge, a yield, or an unprotected turn, several maneuvers can be simultaneously reasonable (Xing et al., 2025; Jiang et al., 2023a;b). These factors leave the future genuinely open: the scene does not yet reveal whether the ego should proceed or hold, and the same observations admit more than one admissible continuation. Such moments therefore call not for a single confident trajectory, but for an explicit account of the competing futures the scene still admits.

Diffusion planners are attractive here because they can represent a multimodal conditional distribution over future ego trajectories (Liao et al., 2025; Zheng et al., 2025). Yet this distribution is not a decision oracle. The denoising objective fits the frequency and geometry of trajectories in the training data; it does not directly optimize closed-loop collision cost or the asymmetric consequences of an unsafe commit (Zheng et al., 2026; Xing et al., 2025). Thus, probability is not utility: selecting the highest-mass mode may favor a common committing behavior even when a less frequent wait mode is safety-critical. Finite data, model misspecification, and a limited sampling budget further distort mode frequencies (Yu et al., 2024; Lee et al., 2025), while accumulated prediction and control errors move the closed-loop policy away from the training distribution (Dauner et al., 2023). We therefore regard diffusion samples as expressive behavioral hypotheses, not calibrated probabilities.

Despite these limitations, the sampled distribution still reveals decision-relevant structure: it shows which behavioral alternatives the model considers plausible and how strongly the samples support each one. In particular, the sampled mass of wait trajectories measures the evidence for a safety-



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

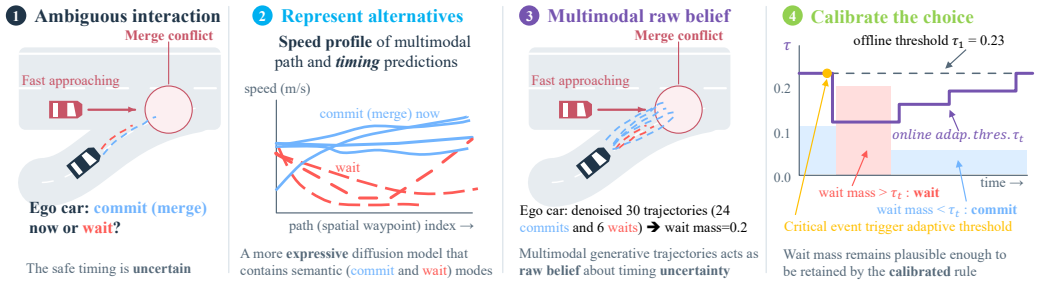



Figure 1: Commit and Wait alternatives are generated by a **more expressive multimodal** diffusion model and serve as **raw belief** about timing decision. This raw belief is **calibrated** by adaptive conformal inference.

critical alternative to commitment. Even when commit is the dominant mode, a non-negligible wait mass can signal unresolved ambiguity that a maximum-likelihood decision would discard. This signal is not sufficient by itself, however. Disagreement does not necessarily imply danger, while majority may reflect a confidently wrong model. The central challenge is therefore not only to represent alternative futures, but also to calibrate whether the evidence for the wait mode is large enough that it cannot yet be safely excluded.

We introduce EvoBelief, a new paradigm that **evolves** an untrusted multimodal **belief** into a certified wait-or-commit decision in three stages. It first **represents** behavior in a joint path-and-speed output expressive enough to hold both the wait and commit modes—a scalar speed head (Liu et al., 2026c) cannot. It then **approximates the posterior** with a multimodal diffusion planner whose samples populate the safety-critical wait mode, and reads their wait mass as an untrusted score rather than a calibrated probability. It finally **calibrates and decides**: split conformal prediction (Vovk et al.; Shafer & Vovk, 2008) and its online adaptive extension (Gibbs & Candes, 2021) turn this untrusted score into a wait-or-commit rule with a distribution-free bound on unsafe commits. On the Bench2Drive (Jia et al., 2024) closed-loop benchmark, EvoBelief raises the success rate to 91.36%, improving over the previous state of the art by 2.11%.

## 2 Method

### 2.1 Problem Formulation and Method Overview

Most driving policies are learned by imitation, which typically supervises each scene with a *single* demonstrated continuation as ground truth. In an ambiguous scene several behaviors are admissible, yet prevalent policies execute one decision and discard the others, so they are rarely encouraged to represent the **residual uncertainty** such scenes carry—slowing, probing, waiting—and tend to collapse onto one hypothesis where they should deliberate. We cast the resulting difficulty as an *evolving belief* the agent must resolve into a wait-or-commit action at each control cycle. At control cycle $k$ the agent holds an observation history $o_{1:k}$ and must choose to `wait` or `commit` (the continuous trajectory within either mode is produced by the planner). The safety-critical unknown is the ground-truth mode $Z_k \in \mathcal{Z} = \{\texttt{wait}, \texttt{commit}\}$—whether the scene genuinely requires waiting—which is resolved only later; write $p_k^* = P(Z_k = \texttt{wait} \mid o_{1:k})$ for its true posterior. The two errors are strongly asymmetric: committing when $Z_k = \texttt{wait}$ risks a collision (cost $J_{\text{col}}$), whereas an unnecessary wait costs only a per-cycle delay ($J_{\text{delay}}$), with $J_{\text{col}} \gg J_{\text{delay}}$. Minimizing the one-step expected cost gives the Bayes-optimal reference policy

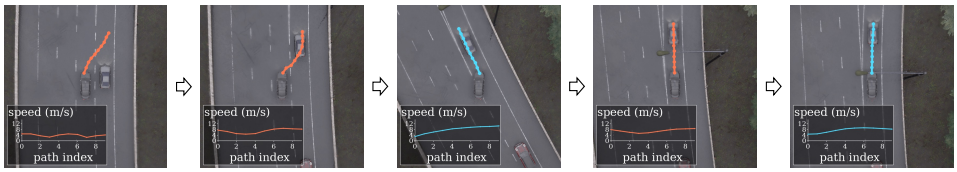
$$\pi_k^*(o_{1:k}) = \arg\min_{a \in \{\texttt{wait},\texttt{commit}\}} \mathbb{E}[J(a, Z_k) \mid o_{1:k}] = \begin{cases} \texttt{commit}, & J_{\text{col}}\, p_k^* \le J_{\text{delay}}, \\ \texttt{wait}, & J_{\text{col}}\, p_k^* > J_{\text{delay}}, \end{cases} \tag{1}$$

that is, `commit` iff $p_k^* \le \delta$ with the Bayes threshold $\delta = J_{\text{delay}}/J_{\text{col}}$. Since $p_k^*$ is latent, this reference policy is unattainable, and EvoBelief approaches it in three stages:

1. **Represent** (§2.2): predict behavior in an output space expressive enough to *contain* the competing wait/commit modes, so the safety-critical wait behavior is realizable.




(a) Scalar speed is less expressive and is only partially able to handle staged wait-then-commit: succeeded in merging, but crashed to the car ahead in the end.




(b) Speed profile is shown in bottom left diagram, indicating how planner decide to control the vehicle speed at each path point; therefore it is able to better express staged wait-then-commit behavior.

Figure 2: Merging process (evolved from left to right) of diffusion planners with scalar speed (a) and speed profile (b) representation, respectively. Blue: commit decision; Red: wait decision.

2. **Approximate the posterior** (§2.3): sample from a multimodal generative planner reliable enough that its samples cover the modes of $p_k^*$—so the wait mode is populated when it is genuinely plausible—and read the resulting wait mass $\hat{p}_k(\texttt{wait})$ as an *approximated* score. A diffusion planner can supply such a belief, whereas a weak source such as an MC-dropout head does not, and no downstream calibration can restore a mode the sampler never populates (Prop. 3).
3. **Calibrate and decide** (§2.4–§2.5): because above mentioned score only *approximates* the posterior, split-conformal calibration and its online Proxy-ACI extension turn this untrusted belief into a wait/commit rule with a distribution-free bound on the `wait` or `commit` error.

An expressive representation and a reliable multimodal sampler are thus the *enabling assumptions*; conformal calibration is what makes acting on their untrusted frequencies *safe*. We keep this exposition pipeline-ordered and defer the research questions to the experiments (§3), which interrogate the load-bearing stages: the representation, belief source, and online adaptation *together* (RQ1), the calibrated *object* (RQ2), and the offline certificate with its online preservation (RQ3). Concrete instantiation, hyperparameters, and computation cost are detailed in Appendix J.

### 2.2 Expressive Output Representation

A generative planner can only exhibit multimodality its output space can *express*: a scalar speed head aliases `wait` vs. slow-`commit` into one output, rendering the `wait`/`commit` event **not measurable**, so *no* threshold on such a head can effectively separate waiting from proceeding slowly (Prop. 4). We therefore predict the ego's future **path and speed profile** $y_0 = (\text{path}, v_{0:9})$ jointly over a 10 m horizon. Path is also known as spatial waypoints (i.e., predicting positions along the path at fixed spatial intervals). An illustration of speed profile is provided in Fig. 2b. Left unconstrained (not forced monotone), the model learns **staged** profiles—decelerate-then-accelerate (a profile's wait/commit label is set by the dynamics criterion $\mathcal{D}$, Appendix F)—the generative signature of deliberation that a single target speed cannot carry, so one joint representation spans both the geometric and interactive axes. A comparison of both representation is given in Fig. 2. These samples are raw material; §2.3 reduces them to the signal the gate consumes. We note that a concurrent work, SparseDriveV2 (Sun et al., 2026), adopts a similar path-and-velocity-profile factorization. Their goal, however, is orthogonal to ours: they compose paths and velocity profiles to *score and select a more accurate trajectory*, whereas we read the speed profile as evidence for *behavioral mode estimation* (wait vs. commit). A comparison with SparseDriveV2 is reported in Table 4.

### 2.3 Belief from Multimodal Diffusion Sampling

We use a multimodal diffusion planner as the sampler. Conditioned on the observation history $o_{1:k}$, the diffusion model induces a conditional model distribution $q_\theta(y \mid o_{1:k})$ over trajectories. We regard

this as an *uncalibrated predictive distribution*: it provides a principled distribution from which to sample behavioral hypotheses, but it need not match the true conditional distribution of trajectories. At decision cycle $k$ we draw $M$ conditionally independent trajectories $y_k^{(m)} \sim q_\theta(\cdot \mid o_{1:k})$. These samples define the following **finite-sample approximation of the model distribution**:

$$\widehat{q}_{\theta,M}(A \mid o_{1:k}) = \frac{1}{M}\sum_{m=1}^{M} \mathbf{1}\left[y_k^{(m)} \in A\right], \qquad y_k^{(m)} \overset{\text{i.i.d.}}{\sim} q_\theta(\cdot \mid o_{1:k}), \tag{2}$$

for any measurable set of trajectories $A$; that is, the empirical distribution placing mass $1/M$ on each sampled trajectory $y_k^{(m)}$. Applying the dynamics criterion $\mathcal{D}$ (Appendix F) to each sample gives the **sampled wait mass**

$$\hat{p}_k(\texttt{wait}) = \frac{1}{M}\sum_{m=1}^{M} \mathbf{1}\left[\mathcal{D}\left(y_k^{(m)}\right) = \texttt{wait}\right], \qquad \hat{p}_k(\texttt{commit}) = 1 - \hat{p}_k(\texttt{wait}), \tag{3}$$

which is a Monte Carlo estimate of the wait mass under $q_\theta$, not a calibrated estimate of the true posterior $p_k^*$. We therefore use $\hat{p}_k(\texttt{wait})$ only as a raw score and calibrate it in §2.4.

### 2.4 Calibration: Static Certificate + Online Extension

We now calibrate the wait mass of §2.3 into a wait/commit gate—first offline, then online. We write $\hat{p}_k(Z)$ for the sampled mass of outcome $Z \in \mathcal{Z}$, with $\hat{p}_k(\texttt{commit}) = 1 - \hat{p}_k(\texttt{wait})$.

**Offline split-conformal.** We calibrate the `wait` detector *class-conditionally*, on the $n$ calibration frames whose expert outcome is $Z_i = \texttt{wait}$ (the safety-critical class, labelled by the dynamics criterion $\mathcal{D}$, Appendix F). For each such frame, define the nonconformity score

$$s_i = -\log \hat{p}_i(\texttt{wait}), \qquad i = 1, \dots, n, \tag{4}$$

so a large score means that the model gave the true `wait` mode little sampled mass. For a target miscoverage level $\alpha \in (0,1)$ fixed before calibration, let $\widehat{Q}_{1-\alpha}(S_{\text{cal}})$ be the $\lceil (n+1)(1-\alpha) \rceil$-th smallest value among $\{s_i\}_{i=1}^n$, with the usual $+\infty$ convention if the requested rank exceeds $n$. At cycle $k$, form the conformal prediction set

$$C_k = \{Z \in \mathcal{Z} : \hat{p}_k(Z) \geq \tau(\alpha)\}, \qquad \tau(\alpha) = \exp\left[-\widehat{Q}_{1-\alpha}(S_{\text{cal}})\right]. \tag{5}$$

If the `wait` calibration frames and a `wait` test frame are exchangeable, split conformal prediction gives the class-conditional guarantee $P(\texttt{wait} \notin C_k \mid Z_k = \texttt{wait}) \leq \alpha$. The policy commits only when $\texttt{wait} \notin C_k$; because $\mathcal{Z}$ is binary and only the wait outcome triggers caution, this gate is exactly the one-sided test $\texttt{wait} \in C_k \iff \hat{p}_k(\texttt{wait}) \geq \tau(\alpha)$. Writing the **unsafe-commit rate** $\text{USR} := P(\pi = \texttt{commit} \mid Z = \texttt{wait})$ for the frequency of committing on a genuine `wait` frame—and on such a frame committing is exactly the event $\texttt{wait} \notin C_k$—the guarantee reads

$$\boxed{\text{USR} \;=\; P(\pi = \texttt{commit} \mid Z = \texttt{wait}) \;\leq\; \alpha}. \tag{6}$$

**Online Monotonicity-preserving Proxy-ACI.** The offline certificate above holds only when the test frame is exchangeable with $S_{\text{cal}}$. At deployment this breaks down: closed-loop rollouts drift into scenes that are systematically harder than—and statistically dependent on—the calibration routes, so a frozen threshold $\tau(\alpha)$ can silently under-cover (§3.4). The natural remedy is Adaptive Conformal Inference (ACI), which restores coverage by nudging the level $\alpha_t$ from the realized error stream; but ACI needs a *dense* error signal, whereas true collisions are far too sparse (∼1–3/episode) to adapt on. We therefore introduce *Proxy-ACI*, which drives the ACI update with a dense risk proxy and caps it so the deployed gate can only tighten the offline certificate, never weaken it. We use the future risk map $R_k$ (a learning-free spatio-temporal occupancy estimate; detailed in Appendix G) as a **dense proxy** $\text{err}_t^{\text{proxy}}$ (∼300 signals/episode at 10 Hz) and update with a monotonicity constraint:

$$\alpha_{t+1} = \min\left(\alpha_t + \gamma(\alpha - \text{err}_t^{\text{proxy}}),\ \alpha_1\right), \qquad \alpha_1 = \alpha. \tag{7}$$

Because $\tau_t = \exp[-\widehat{Q}_{1-\alpha_t}(S_{\text{cal}})]$ is non-decreasing in $\alpha_t$, the cap $\alpha_t \leq \alpha_1$ makes the adaptive gate **pointwise at least as cautious** as the static gate certified at $\alpha$: on every input it forces a wait

whenever the static gate would (Lemma 2). This is a *sample-path dominance* statement and holds under *any* deployment distribution; it does **not** re-derive an absolute per-step bound $\mathrm{USR}_t \le \alpha_t$ on the deployed per-step unsafe-commit rate, which would require both exchangeability—broken at deployment, as §3.4(b) shows—and a level chosen independently of the rollout. What the cap buys is therefore: the offline certificate remains the *worst case* of the deployed rule, which can only tighten from it. The cap also converts unconstrained ACI's two-sided convergence into a **one-sided** bound on the proxy-trigger rate (Prop. 2 vs. Prop. 1): the deployed loop can be more conservative than $\alpha$ indefinitely, but not persistently less. Transfer to true collisions remains an **empirical** bridge via risk-map recall $\epsilon_{\text{deploy}}$, measured on the deployment distribution (Appendix I).

### 2.5 Decision Rule

The calibrated gate is exactly the one-sided threshold test established in §2.4: with the wait-mass threshold $\tau(\alpha) = \exp[-\widehat{Q}_{1-\alpha}(S_{\text{cal}})]$, the policy commits precisely when the sampled wait mass falls short of it,

$$\pi^*_\alpha(o_{1:k}) = \begin{cases} \texttt{wait}, & \hat{p}_k(\texttt{wait}) \ge \tau(\alpha), \\ \texttt{commit}, & \hat{p}_k(\texttt{wait}) < \tau(\alpha). \end{cases} \tag{8}$$

This gate is more than a heuristic threshold on the certificate: it is exactly the policy that minimizes a conformal upper bound on the per-cycle decision cost of §2 (App. E), so acting on the untrusted wait mass is *cost-optimal under the certified bound*—though not Bayes-optimal for the true cost, which, were $p^*_k$ known, would instead use the posterior threshold $\delta = J_{\text{delay}}/J_{\text{col}}$.

**Effect of $\alpha$.** Because $\widehat{Q}_{1-\alpha}$ is non-increasing in $\alpha$, $\tau(\alpha)$ is non-decreasing, so a smaller $\alpha$ only enlarges the wait region: for $\alpha_a < \alpha_b$, $\{o_{1:k} : \pi^*_{\alpha_a} = \texttt{wait}\} \supseteq \{o_{1:k} : \pi^*_{\alpha_b} = \texttt{wait}\}$. Online we initialize $\alpha_1 = \alpha$ and constrain $\alpha_t \le \alpha_1$, so $\tau_t \le \tau(\alpha)$ and the adaptive policy can only enlarge the static wait region in response to deployment risk.

Algorithm 1 implements (8) in two stages. First, the conformal gate selects the decision mode. Second, the planner selects a trajectory within that mode: for `wait`, the slowest sample satisfying $\rho(y) = 1 - \exp(-\int_s R_k(y(s))\,ds) \le \rho_{\max}$; for `commit`, the consensus representative when the sampled modes agree, or the feasible representative with minimum peak risk otherwise. This trajectory-selection stage is outside the conformal certificate. We cluster samples into modes $\{G_l\}$ to support this selection. Online adaptation modifies the gate through $\alpha_t \mapsto \tau_t$; the cap $\alpha_t \le \alpha$ ensures it can only become more cautious than the static policy.

## 3 Experiments

We organize the evaluation around three research questions that mirror the method's structure: **RQ1** establishes the headline closed-loop result and shows it is a genuine *synergy* between an expressive representation, belief source, and online calibration; **RQ2** isolates *what* is calibrated—holding the backbone of diffusion model and conformal pipeline fixed, it contrasts EvoBelief's behavior-level wait mass against PlanCP's (Sun et al., 2023) trajectory-level return band; **RQ3** validates the offline distribution-free certificate and shows that a *static* threshold—though sound under exchangeability—degrades under deployment distribution shift and must be repaired by the online monotonicity-preserving Proxy-ACI.

### 3.1 Setup

**Benchmark and metrics.** We evaluate on Bench2Drive (Jia et al., 2024), a closed-loop CARLA (Dosovitskiy et al., 2017) benchmark whose reactive agents let the ego vehicle actually negotiate merges, yields, and crossings—the interactive regime our decision layer targets. Closed-loop quality is reported as Success Rate (SR) and Driving Score (DS). To look inside the calibration layer we further report the empirical unsafe-commit rate (USR) against its target $\alpha$, the resulting over-conservatism, and the Proxy-ACI running coverage.

**Ablation factors.** Every variant shares the same anchor-based TransFuser-v6 perception backbone (Nguyen et al., 2026), anchor structure, and training; the experiments toggle only the planner head and the decision layer stacked on it. We vary four factors. **(1) Representation**: a scalar target

**Algorithm 1** EVOBELIEF decision rule (per control cycle $t$).

**Require:** observation history $o_{1:t}$; adaptive level $\alpha_t$ (persists across cycles); calibration score set $S_{\text{cal}}$; fixed safety level $\alpha$; feasibility limit $\rho_{\max}$; learn rate $\gamma$
**State:** $\alpha_1 = \alpha$ ▷ offline safety level and monotone cap
# —- decision evidence (Sec. 2.3) —-
$\{y_1..y_M\}, \{G_l\} \leftarrow \textsc{Sample}(o_{1:t}, M)$ ▷ $M$ trajectories + spatial modes
$R_k \leftarrow \textsc{RiskMap}(o_{1:t})$ ▷ dense collision-rate field
$\hat{p}_k(\texttt{wait}) \leftarrow (1/M)\sum_m \mathbf{1}[\,\mathcal{D}(y_m) == \texttt{wait}\,]$ ▷ sampled wait mass (Alg. 2)
# —- primary conformal gate (Sec. 2.4) —-
$\tau_t \leftarrow \exp\big(-\widehat{Q}_{1-\alpha_t}(S_{\text{cal}})\big)$
**if** $\hat{p}_k(\texttt{wait}) \geq \tau_t$ **then** ▷ `wait` is in the conformal set
  $F \leftarrow \{\, m : \rho(y_m) \leq \rho_{\max} \,\}$ ▷ $\rho(y) = 1 - \exp(-\int R_k(y(s))\, ds)$
  $\widetilde{F} \leftarrow F$ if $F \neq \varnothing$, else $\{1, \ldots, M\}$
  $y^* \leftarrow \arg\min_{m \in \widetilde{F}} \textsc{TargetSpeed}(y_m)$
  ▷ no $\alpha$ update on a wait (one-sided bandit)
**else** ▷ commit
  **if** modes $\{G_l\}$ agree **then**
    $y^* \leftarrow \textsc{ConsensusModeRepresentative}$
  **else**
    $y^* \leftarrow \arg\min_{\text{feasible modes}} \textsc{PeakRisk}(\cdot, R_k)$
  **end if**
  peak $\leftarrow \max_{s \in [t, t+T]} R_k(y^*(s))$
  $\text{err}_{\text{proxy}} \leftarrow \mathbf{1}[\,\text{peak} > \tau_{\text{risk}}\,]$
  $\alpha_{t+1} \leftarrow \text{clip}\big(\alpha_t + \gamma(\alpha - \text{err}_{\text{proxy}}),\ -\gamma,\ \alpha_1\big)$
  ▷ standard ACI is uncapped; the cap at $\alpha_1$ enforces no-weakening (Lemma 2)
**end if**
**return** executed trajectory $y^*$

speed vs. speed profile $v_{1:T}$. **(2) Belief source**: an MC-dropout head (Gal & Ghahramani, 2016) vs. a multimodal diffusion head. Crossing (1)×(2) yields the four planner heads of RQ1—**LEAD** (scalar, MC-dropout), **LEAD-variant** (profile, MC-dropout), **BridgeDrive** (scalar, diffusion), and **EvoBelief** (profile, diffusion; ours). **(3) Decision layer** (RQ3): **no-gate** (execute the planner directly), **risk-map** (execute the lowest-risk sample; App. H), **fixed-$\alpha$** (static split-conformal threshold, = Proxy-ACI with $\gamma$=0), and **Proxy-ACI** (our monotonicity-preserving online adaptation). **(4) Calibrated object** (RQ2): EvoBelief's behavior-level wait mass vs. PlanCP's trajectory-level return band, holding (1)–(3) fixed. Following common practice, the best configuration is run with three random seeds and the rest once, given the heavy cost of closed-loop evaluation (Liu et al., 2026c; Jia et al., 2025).

### 3.2 RQ1 — IS CALIBRATED ONLINE ADAPTATION ON AN EXPRESSIVE REPRESENTATION THE SOURCE OF THE GAIN, AND IS IT A GENUINE SYNERGY?

We cross factors (1) representation and (2) belief source into the $2 \times 2$ planner-head grid of Table 2, and read each head under factor (3), the decision layer (columns; the *risk-map* isolates the safety filter). The four heads share the *same* backbone and differ by exactly one factor between neighbors—scalar↔profile (representation) and dropout↔diffusion (belief source)—so the readout is not any single cell but the **synergy pattern**. We read the grid as a controlled ablation: its single conclusion is that the gain is a *synergy*, not a sum, which we then break down along the three axes it crosses.

**The gain is an synergy, not a sum.** The only configuration above the prior state-of-the-art (BridgeDrive in Table 2) is the single cell that combines all three ingredients—an expressive path-and-speed-profile representation, a diffusion belief that faithfully covers the wait/commit modes, and online risk-adaptive calibration—which reaches SR $\mathbf{91.36}$ by cutting *collisions* on the interaction-heavy subset. Remove any one ingredient and the score collapses back into the SR $\leq 89.25$ baseline band: no single factor, and no pair, reproduces it, and for the scalar arms this failure is *predicted* rather than merely observed (Prop. 4). We break the synergy down along its three axes below; the remaining arm—*calibration alone*, i.e. the static fixed-$\alpha$ column—is examined in RQ3, where a frozen threshold reduces to no-gate.

Table 1: EVOBELIEF outperforms the strongest prior planners. More comparison in Table 4

| Method | Calibration | SR(%) | DS |
|---|---|---|---|
| LEAD (Nguyen et al., 2026) | — | 86.8 ± 0.7 | 95.2 ± 0.3 |
| BridgeDrive (Liu et al., 2026c) | — | 89.25 ± 0.50 | 96.34 ± 0.55 |
| PlanCP† (Sun et al., 2023) | trajectory-level | 89.08 ±0.79 | 96.09 ±0.61 |
| EvoBelief (ours) | behavior-level | **91.36** ± 0.79 (+2.11) | **96.52** ± 0.62 (+0.18) |

†Our re-implementation on EvoBelief's backbone, differing only in the calibrated object; see §3.3 and App. C.

Table 2: A 2 × 2 ablation over *representation* × *belief source*, crossed with the *decision layer* (columns). Only the bottom-right cell—speed-profile × diffusion × Proxy-ACI—clears the prior SOTA SR 89.25; every marginal move alone stays in the baseline band. The best and second-best results are marked in **bold** and *underlined italics*, respectively. Results are presented as SR | DS.

| | | Decision layer | | | |
|---|---|---|---|---|---|
| Repr. | Belief source | no-gate | Risk-map | fixed-$\alpha$ ($\gamma$=0) | Proxy-ACI |
| scalar | MC-dropout (LEAD) | 86.8 \| 95.2 | 86.82 \| 95.05 | 85.91 \| 95.01 | 85.91 \| 93.77 |
| | diffusion (BridgeDrive) | 89.25 \| 96.34 | 89.09 \| 95.92 | 88.18 \| 95.19 | *89.54* \| *96.37* |
| profile | MC-dropout (LEAD-variant) | 87.27 \| 95.23 | 86.81 \| 95.42 | 86.36 \| 95.28 | 87.72 \| 95.71 |
| | diffusion (EvoBelief) | 88.67 \| 95.86 | 89.09 \| 95.53 | 89.54 \| 95.79 | **91.36 \| 96.52** |

**(a) Representation: expressive, yet inert on its own.** Swapping the scalar head for the speed profile *without* a calibrated consumer (EvoBelief no-gate, SR 88.67) does not even match the scalar-diffusion baseline (BridgeDrive no-gate, SR 89.25): a richer output space is dead weight until something *uses* its multimodality. The profile moreover pays off *only* under a faithful sampler—adding it to MC-dropout (LEAD → LEAD-variant, SR 86.8 → 87.72) slightly moves the needle, whereas adding it to diffusion delivers the gain—which is the representation×belief-source synergy of (b).

**(b) Belief source: diffusion covers the modes, MC-dropout does not.** Holding the representation fixed at the speed profile, the MC-dropout belief (LEAD-variant, SR ∼86.36–87.72) stays well below the diffusion belief (EvoBelief, SR **91.36**): dropout's perturbation-induced sample frequencies do not reliably populate the alternative wait/commit modes, so the gate has nothing faithful to threshold. A multimodal *distribution* estimator is therefore load-bearing—the expressive output helps only once the sampler actually places mass on the modes it can express.

**(c) Decision: the risk map alone is not the source.** The *Risk-map* column—an uncalibrated filter that executes the lowest-risk sample every frame, sharing the *same* risk map as Proxy-ACI—does not reproduce the SR 91.36 (Appendix H). The reason is temporal: early in an interaction the committing trajectory is not yet riskier than a waiting one, so a risk-greedy filter has no signal to slow the ego and reacts only once the conflict is imminent—too late to avoid the cumulative error. The calibrated belief gate instead forces the wait from the sampled multimodality *before* the risk materializes.

### 3.3 RQ2 — WITH EVERYTHING ELSE HELD FIXED, DOES *what* IS CALIBRATED MATTER?

RQ1 establishes *what* calibrated adaptation drives the gain; RQ2 asks *what* the gate should calibrate. We hold EvoBelief's backbone, joint path-and-speed representation, sample budget, and split-conformal pipeline fixed and change only the calibrated object, substituting PlanCP (Sun et al., 2023)—which calibrates a *trajectory-level* return band—for EvoBelief's *behavior-level* wait mass. Both methods reduce perception input to a safety measure, calibrate a threshold on it, and feed the *same* wait/commit interface, so the comparison is a clean test of granularity in which only the calibrated safety measure differs. Implementation details are provided in Appendix C.

**Result.** Table 1 reports the stronger of two PlanCP variants—the planner *retrained* with an auxiliary conformal loss (SR 89.08±0.79 / DS 96.09±0.61, mean over three seeds); applying PlanCP's calibrated gate *directly to EvoBelief's own trained model* (no retraining) attains only SR 81.81 /

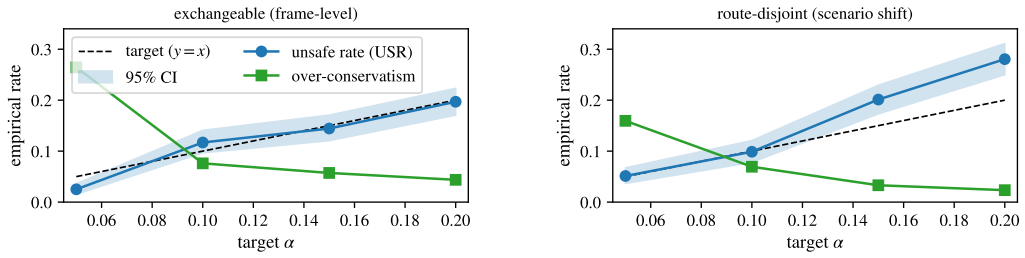



Figure 3: **Offline conformal coverage on a route-disjoint held-out cal/eval fold** (unsafe rate and over-conservatism vs. target $\alpha$; shaded = 95% bootstrap CI of the eval USR). *Left — exchangeable (frame-level) split:* empirical USR tracks $\alpha$ within CI, confirming the certificate (RQ3a). *Right — route-disjoint (scenario-shift) split:* the static threshold under-covers at large $\alpha$ (RQ3b), motivating the online adaptation of RQ3(c).

DS 91.23. Either way the success rate falls well below EvoBelief's SR 91.36 (more details in Appendix C). Because the backbone, representation, and conformal machinery are identical across all three settings, the gap is attributable *entirely* to the calibrated object.

**Analysis.** The gap traces to *what* each calibrated scalar measures. PlanCP calibrates the nominal plan's accumulated risk-map return; it does react to safety-critical events, but only once the intended plan itself accumulates risk. EvoBelief instead calibrates the sampled wait mass—the model's own behavioral evidence for the wait mode—which rises while the samples are still split between waiting and committing, before that risk materializes on the map. A marginal band over whole-trajectory risk therefore applies caution late and undifferentiated, whereas calibrating the behavior-level wait mode acts on the ambiguity before it becomes unavoidable.

### 3.4 RQ3 — Does the offline certificate hold, and is online adaptation needed to preserve it under deployment shift?

We first validate the static certificate (Theorem 1) offline on a route-disjoint held-out fold, then ask whether online adaptation is *needed* and *sufficient* to preserve it at deployment. At **offline** level, nonconformity scores $s = -\log \hat{p}_k(\texttt{wait})$ are computed on ground-truth wait frames; a calibration subfold sets $\tau(\alpha) = \exp(-Q_{1-\alpha})$ and a disjoint evaluation subfold measures the realized unsafe rate. At **deployment (online)** level, we instrument every control cycle of the running system—$\alpha_t$, $\tau_t$, the sampled wait mass, the risk-proxy trigger $\mathrm{err}_t^{\mathrm{proxy}}$, the wait/commit decision, and the applied controls (Fig. 4); the proxy-to-truth bridge is analyzed in Appendix I.

**(a) Coverage holds under exchangeability.** With an exchangeable (frame-level) cal/eval split, the empirical USR tracks the target $\alpha$ within its 95% CI across the whole range (Fig. 3 left), confirming the certificate machinery: $\mathrm{USR} = P(\pi = \texttt{commit} \mid Z = \texttt{wait}) \le \alpha$, distribution-free, with no assumption on the model's calibration.

**(b) A static threshold is not enough under deployment shift.** With a route-disjoint cal/eval split—a proxy for deployment-time scene shift that breaks exchangeability—coverage holds at small $\alpha$ but is increasingly violated as $\alpha$ grows (Fig. 3 right): the calibration routes are systematically easier than the evaluation routes, so the static $\tau$ under-covers. Nor can a single globally tight $\alpha$ fix this: shrinking $\alpha$ suppresses unsafe commits only by forcing waits on safe frames as well, so over-conservatism ($P(\pi = \texttt{wait} \mid Z = \texttt{commit})$) climbs uniformly (Fig. 3). A threshold frozen offline therefore cannot be trusted once the distribution moves—motivating the online adaptation tested next.

**(c) Online adaptation restores coverage and outperforms the static threshold.** The adaptive arm differs from the static one only in the learning rate $\gamma$: $\gamma$=0 freezes $\alpha_t$ at the certified level (fixed-$\alpha$), while $\gamma$>0 adapts it online from the dense risk proxy. The static arm (fixed $\alpha$) performs similar to no-gate, whereas Proxy-ACI reaches SR **91.36** (Table 2). A dense proxy is essential: true collisions are far too sparse to drive adaptation—over a full episode the risk proxy fires 10–20 times while *zero* real collisions occur (Fig. 4), so a collision-driven ACI would receive no signal, whereas the dense proxy (∼300 signals/episode at 10 Hz) lets the running coverage $\frac{1}{t}\sum \mathrm{err}^{\mathrm{proxy}}$ control the

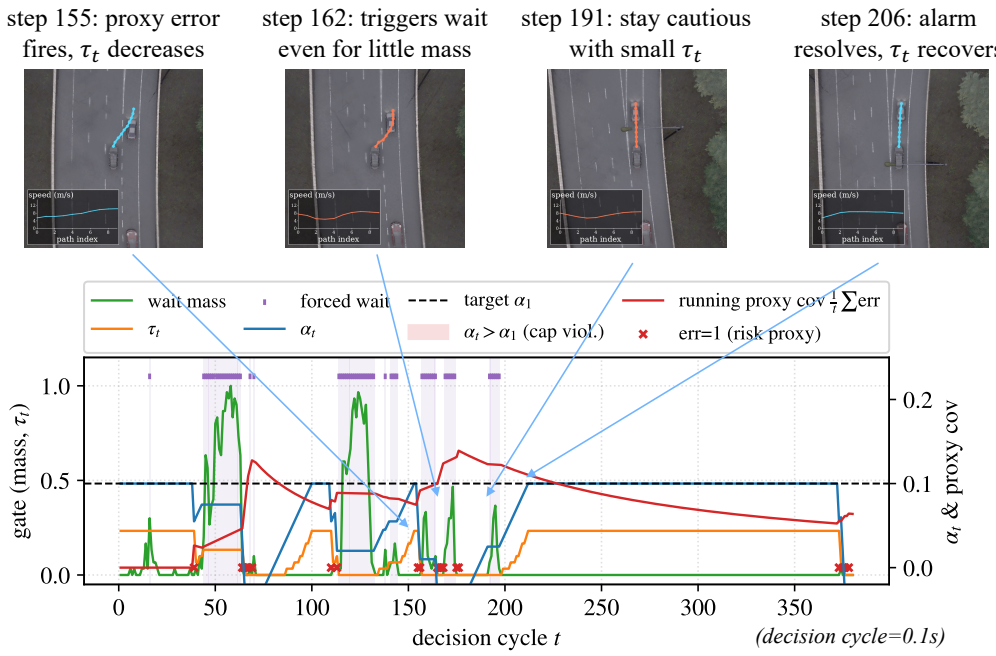



Figure 4: **Deployed Proxy-ACI, one interaction-heavy example.** The $\alpha_t$ rests at the certified $\alpha_1$ and only dips when the risk proxy fires — no cap violation. The gate ($\tau_t$ and the sampled wait mass) with forced-wait cycles marked. Running proxy coverage with $\mathrm{err}^{\mathrm{proxy}}$ events; note the frequent proxy triggers but **no** real collision, which is precisely why online adaptation needs a dense proxy.

proxy-trigger rate and react within seconds (Prop. 2). Crucially, the adaptation never weakens the offline guarantee: the monotone cap holds $\alpha_t \leq \alpha_1$ with **zero** cap violations (Lemma 2), so $\alpha_t$ sits at the cap $\alpha_1$ during safe cruising and dips below it only when the risk proxy fires, relaxing back to $\alpha_1$ once the interaction resolves. The mechanism is visible cycle-by-cycle: as a negotiation begins the sampled wait mass rises; on a risky commit $\mathrm{err}^{\mathrm{proxy}}$ fires, $\alpha_t$ tightens, and the ego decelerates until the conflict clears.

## 4 Conclusion

We presented EvoBelief, which reads a diffusion planner's multimodal samples as a *behavioral belief*—evidence for the safety-critical wait mode—rather than a set of candidate trajectories, and turns that untrusted belief into a conformally calibrated wait/commit decision. This reframes the safety-critical timing decision as an evidence-calibration problem: an expressive multimodal belief supplies *what* to act on, and distribution-free calibration—preserved online—decides *when*. On Bench2Drive this outperformed the prior state-of-the-art by $2.11\%$ in success rate, and our ablations show the gain is a genuine synergy among the representation, the belief source, and online adaptation rather than any one of them alone. We hope EvoBelief offers a new paradigm for turning generative multimodality into statistically grounded decisions in safety-critical autonomous systems. We discuss the limitations of our approach and directions for future work in Appendix K.

### AI USE STATEMENT

Generative AI tools were used only to assist with implementing portions of the code and polishing the manuscript's language. They were not used to develop the method, design experiments, analyze or interpret results; no other tasks require disclosure. We reviewed and tested all AI-assisted code and checked all AI-edited text for accuracy. We take full responsibility for the content of this work.

### REPRODUCIBILITY STATEMENT

To support reproducibility, Section 2 describes our method—the joint path-and-speed representation, the multimodal diffusion belief, the conformal calibration layer, and the decision rule—together with the full per-cycle decision procedure (Algorithm 1). Section 3 reports the Bench2Drive benchmark, evaluation metrics, ablation factors, and baselines. Concrete instantiation, all hyperparameters, and the computation-cost profiling are collected in Appendix J; the two hand-crafted components—the dynamics criterion and the risk map—are fully specified in Appendices F and G, and the PlanCP baseline in Appendix C. The assumptions, lemmas, theorem, and proofs underlying our theoretical analysis are provided in Appendix D. We plan to open-source our code and release our trained model to enable independent reproduction of the reported results and support further research.

## REFERENCES

Michael Samuel Albergo and Eric Vanden-Eijnden. Building normalizing flows with stochastic interpolants. In *The Eleventh International Conference on Learning Representations*, 2023.

Leo Andeol, Thomas Fel, Florence de Grancey, and Luca Mossina. Confident object detection via conformal prediction and conformal risk control: an application to railway signaling. In Harris Papadopoulos, Khuong An Nguyen, Henrik Boström, and Lars Carlsson (eds.), *Proceedings of the Twelfth Symposium on Conformal and Probabilistic Prediction with Applications*, volume 204 of *Proceedings of Machine Learning Research*, pp. 36–55. PMLR, 13–15 Sep 2023.

Anastasios N Angelopoulos and Stephen Bates. Conformal prediction: A gentle introduction. *Foundations and Trends in Machine Learning*, 16(4):494–591, 2023.

Andrew Bacha, Cheryl Bauman, Ruel Faruque, Michael Fleming, Chris Terwelp, Charles Reinholtz, Dennis Hong, Al Wicks, Thomas Alberi, David Anderson, et al. Odin: Team victortango's entry in the darpa urban challenge. *Journal of field Robotics*, 25(8):467–492, 2008.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Panpan Cai, Yuanfu Luo, Aseem Saxena, David Hsu, and Wee Sun Lee. Lets-drive: Driving in a crowd by learning from tree search. *arXiv preprint arXiv:1905.12197*, 2019.

Wei Cao, Marcel Hallgarten, Tianyu Li, Daniel Dauner, Xunjiang Gu, Caojun Wang, Yakov Miron, Marco Aiello, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. Pseudo-simulation for autonomous driving. In *Conference on Robot Learning (CoRL)*, 2025.

Rohan Chandra and Dinesh Manocha. Gameplan: Game-theoretic multi-agent planning with human drivers at intersections, roundabouts, and merging. *IEEE Robotics and Automation Letters*, 7(2): 2676–2683, 2022.

Rohan Chandra, Mingyu Wang, Mac Schwager, and Dinesh Manocha. Game-theoretic planning for autonomous driving among risk-aware human drivers. In *2022 International Conference on Robotics and Automation (ICRA)*, pp. 2876–2883. IEEE, 2022.

Yuxiao Chen, Peter Karkus, Boris Ivanovic, Xinshuo Weng, and Marco Pavone. Tree-structured policy planning with learned behavior models. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7902–7908. IEEE, 2023.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 24185–24198, 2024.

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

Florinel-Alin Croitoru, Vlad Hondru, Radu Tudor Ionescu, and Mubarak Shah. Diffusion models in vision: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

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

Daniel Dauner, Marcel Hallgarten, Tianyu Li, Xinshuo Weng, Zhiyu Huang, Zetong Yang, Hongyang Li, Igor Gilitschenski, Boris Ivanovic, Marco Pavone, Andreas Geiger, and Kashyap Chitta. NAVSIM: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In Sergey Levine, Vincent Vanhoucke, and Ken Goldberg (eds.), *Proceedings of the 1st Annual Conference on Robot Learning*, volume 78 of *Proceedings of Machine Learning Research*, pp. 1–16. PMLR, 2017.

Achref Doula, Tobias Güdelhöfer, Max Mühlhäuser, and Alejandro Sanchez Guinea. Conformal prediction for semantically-aware autonomous perception in urban environments. In *8th Annual Conference on Robot Learning*, 2024.

Di Feng, Christian Haase-Schütz, Lars Rosenbaum, Heinz Hertlein, Claudius Glaeser, Fabian Timm, Werner Wiesbeck, and Klaus Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3):1341–1360, 2020.

Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059. PMLR, 2016.

Isaac Gibbs and Emmanuel Candes. Adaptive conformal inference under distribution shift. *Advances in Neural Information Processing Systems*, 34:1660–1672, 2021.

Ke Guo, Haochen Liu, Xiaojun Wu, Jia Pan, and Chen Lv. ipad: Iterative proposal-centric end-to-end autonomous driving. *IEEE Robotics and Automation Letters*, 2026.

Guande He, Kaiwen Zheng, Jianfei Chen, Fan Bao, and Jun Zhu. Consistency diffusion bridge models. *Advances in Neural Information Processing Systems*, 37:23516–23548, 2024.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

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

Huiqun Huang, Sihong He, and Fei Miao. Cuqds: conformal uncertainty quantification under distribution shift for trajectory prediction. In *Proceedings of the AAAI conference on artificial intelligence*, volume 39, pp. 17422–17430, 2025.

Wenhui Huang, Songyan Zhang, Qihang Huang, Zhidong Wang, Zhiqi Mao, Collister Chua, Zhan Chen, Long Chen, and Chen Lv. Automot: A unified vision-language-action model with asynchronous mixture -of-transformers for end-to-end autonomous driving. In *Forty-third International Conference on Machine Learning*, 2026.

Xiaosong Jia, Zhenjie Yang, Qifeng Li, Zhiyuan Zhang, and Junchi Yan. Bench2Drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving. In *NeurIPS 2024 Datasets and Benchmarks Track*, 2024.

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

Bo Jiang, Shaoyu Chen, Qing Xu, Bencheng Liao, Jiajie Chen, Helong Zhou, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. VAD: Vectorized scene representation for efficient autonomous driving. *ICCV*, 2023a.

Bo Jiang, Shaoyu Chen, Hao Gao, Bencheng Liao, Qian Zhang, Wenyu Liu, and Xinggang Wang. VADv2: End-to-end vectorized autonomous driving via probabilistic planning. In *The Fourteenth International Conference on Learning Representations*, 2026a.

Chiyu Jiang, Andre Cornman, Cheolho Park, Benjamin Sapp, Yin Zhou, Dragomir Anguelov, et al. Motiondiffuser: Controllable multi-agent motion prediction using diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9644–9653, 2023b.

Guangfeng Jiang, Yueru Luo, Jun Liu, Yi Huang, Yiyao Zhu, Zhan Qu, Dave Zhenyu Chen, Bingbing Liu, and Xu Yan. Wpt: World-to-policy transfer via online world model distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 17842–17852, 2026b.

Kushal Kedia, Prithwish Dan, and Sanjiban Choudhury. A game-theoretic framework for joint forecasting and planning. In *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 6773–6778. IEEE, 2023.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Arne Kesting, Martin Treiber, and Dirk Helbing. General lane-changing model mobil for car-following models. *Transportation research record*, 1999(1):86–94, 2007.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

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

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and scalable predictive uncertainty estimation using deep ensembles. *Advances in neural information processing systems*, 30, 2017.

Giwon Lee, Wooseong Jeong, Daehee Park, Jaewoo Jeong, and Kuk-Jin Yoon. Interaction-merged motion planning: Effectively leveraging diverse motion datasets for robust planning. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1–12. IEEE, 2025.

Jordan Lekeufack, Anastasios N Angelopoulos, Andrea Bajcsy, Michael I Jordan, and Jitendra Malik. Conformal decision theory: Safe autonomous decisions from imperfect predictions. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11668–11675. IEEE, 2024.

Tong Li, Lu Zhang, Sikang Liu, and Shaojie Shen. Marc: Multipolicy and risk-aware contingency planning for autonomous driving. *IEEE Robotics and Automation Letters*, 8(10):6587–6594, 2023.

Yongkang Li, Kaixin Xiong, Xiangyu Guo, Fang Li, Sixu Yan, Gangwei Xu, Lijun Zhou, Long Chen, Haiyang Sun, BING WANG, Kun Ma, Guang Chen, Hangjun Ye, Wenyu Liu, and Xinggang Wang. Recogdrive: A reinforced cognitive framework for end-to-end autonomous driving. In *The Fourteenth International Conference on Learning Representations*, 2026.

Bencheng Liao, Shaoyu Chen, Haoran Yin, Bo Jiang, Cheng Wang, Sixu Yan, Xinbang Zhang, Xiangyu Li, Ying Zhang, Qian Zhang, et al. DiffusionDrive: Truncated diffusion model for end-to-end autonomous driving. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 12037–12047, 2025.

Lars Lindemann, Matthew Cleaveland, Gihyun Shim, and George J Pappas. Safe planning in dynamic environments using conformal prediction. *IEEE Robotics and Automation Letters*, 8(8):5116–5123, 2023.

George Ling, Lijin Yang, Hao Yang, and Zhongzhan Huang. Blue: Toward better language use in efficient vision-language-action models for autonomous driving. *arXiv preprint arXiv:2606.08684*, 2026.

Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=PqvMRDCJT9t.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023a.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. Llavanext: Improved reasoning, ocr, and world knowledge, 2024.

Lin Liu, Caiyan Jia, Guanyi Yu, Ziying Song, JunQiao Li, Feiyang Jia, Peiliang Wu, Xiaoshuai Hao, and Yadan Luo. Guideflow: Constraint-guided flow matching for planning in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3719–3728, 2026a.

Pei Liu, Qingtian Ning, Xinyan Lu, Haipeng Liu, Weiliang Ma, Dangen She, Xianpeng Lang, and Jun Ma. Cogdriver: Integrating cognitive inertia for temporally coherent planning in autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 18150–18160, 2026b.

Shu Liu, Wenlin Chen, Weihao Li, Zheng Wang, Lijin Yang, Jianing Huang, Yipin Zhang, Zhongzhan Huang, Ze Cheng, and Hao Yang. Bridgedrive: Diffusion bridge policy for closed-loop trajectory planning in autonomous driving. In *The Fourteenth International Conference on Learning Representations*, 2026c. URL https://openreview.net/forum?id=dJKhjK4zpp.

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

Yuanfu Luo, Panpan Cai, Aniket Bera, David Hsu, Wee Sun Lee, and Dinesh Manocha. Porca: Modeling and planning for autonomous driving among many pedestrians. *IEEE Robotics and Automation Letters*, 3(4):3418–3425, 2018.

Rowan Thomas McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Weller. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. International Joint Conferences on Artificial Intelligence, Inc., 2017.

Khaled A Mustafa, Daniel Jarne Ornia, Jens Kober, and Javier Alonso-Mora. Racp: Risk-aware contingency planning with multi-modal predictions. *IEEE Transactions on Intelligent Vehicles*, 10 (1):228–243, 2024.

Long Nguyen, Micha Fauth, Bernhard Jaeger, Daniel Dauner, Maximilian Igl, Andreas Geiger, and Kashyap Chitta. Lead: Minimizing learner-expert asymmetry in end-to-end driving. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.

Rahul Rahaman et al. Uncertainty quantification and deep ensembles. *Advances in neural information processing systems*, 34:20063–20075, 2021.

Katrin Renz, Long Chen, Elahe Arani, and Oleg Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2025.

Murat Sensoy, Lance Kaplan, and Melih Kandemir. Evidential deep learning to quantify classification uncertainty. *Advances in neural information processing systems*, 31, 2018.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of machine learning research*, 9(3), 2008.

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*, 2021.

Jiankai Sun, Yiqi Jiang, Jianing Qiu, Parth Nobel, Mykel J Kochenderfer, and Mac Schwager. Conformal prediction for uncertainty-aware planning with diffusion dynamics model. *Advances in Neural Information Processing Systems*, 36:80324–80337, 2023.

Max Muchen Sun, Francesca Baldini, Katie Hughes, Peter Trautman, and Todd Murphey. Mixed strategy nash equilibrium for crowd navigation. *The International Journal of Robotics Research*, 2024. doi: 10.1177/02783649241302342.

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

Wenchao Sun, Xuewu Lin, Keyu Chen, Zixiang Pei, Xiang Li, Yining Shi, and Sifa Zheng. Sparsedrivev2: Scoring is all you need for end-to-end autonomous driving. In *European Conference on Computer Vision*, pp. 446–463. Springer, 2026.

Jiacheng Tang, Zhiyuan Zhou, Zhuolin He, Jia Zhang, Kai Zhang, and Jian Pu. Causalvad: De-confounding end-to-end autonomous driving via causal intervention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 32124–32133, June 2026.

Sebastian Thrun, Mike Montemerlo, Hendrik Dahlkamp, David Stavens, Andrei Aron, James Diebel, Philip Fong, John Gale, Morgan Halpenny, Gabriel Hoffmann, et al. Stanley: The robot that won the darpa grand challenge. *Journal of field Robotics*, 23(9):661–692, 2006.

Keyu Tian, Yi Jiang, Zehuan Yuan, Bingyue Peng, and Liwei Wang. Visual autoregressive modeling: Scalable image generation via next-scale prediction. *Advances in neural information processing systems*, 37:84839–84865, 2024.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2):1805, 2000.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer.

Han Wang and Chao Ning. Conformal prediction in the loop: A feedback-based uncertainty model for trajectory optimization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jun Wang, Guocheng He, and Yiannis Kantaros. Probabilistically correct language-based multi-robot planning using conformal prediction. *IEEE Robotics and Automation Letters*, 10(1):160–167, 2024.

Mingyu Wang, Zijian Wang, John Talbot, J Christian Gerdes, and Mac Schwager. Game-theoretic planning for self-driving cars in multivehicle competitive scenarios. *IEEE Transactions on Robotics*, 37(4):1313–1325, 2021.

Zehao Wang, Huaide Jiang, Shuaiwu Dong, Yuping Wang, Hang Qiu, and Jiachen Li. Drive my way: Preference alignment of vision-language-action model for personalized driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 25204–25214, June 2026.

Zhongyu Xia, Wenhao Chen, Yongtao Wang, and Ming-Hsuan Yang. Knowval: A knowledge-augmented and value-guided autonomous driving system. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3740–3749, 2026.

Zebin Xing, Xingyu Zhang, Yang Hu, Bo Jiang, Tong He, Qian Zhang, Xiaoxiao Long, and Wei Yin. GoalFlow: Goal-driven flow matching for multimodal trajectories generation in end-to-end autonomous driving. *arXiv preprint arXiv:2503.05689*, 2025.

Zhenhua Xu, Yan Bai, Yujia Zhang, Zhuoling Li, Fei Xia, Kwan-Yee K Wong, Jianqiang Wang, and Hengshuang Zhao. Drivegpt4-v2: Harnessing large language model capabilities for enhanced closed-loop autonomous driving. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17261–17270. IEEE, 2025.

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 *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15342–15353. IEEE, 2024.

Zhenjie Yang, Xiaosong Jia, Qifeng Li, Xue Yang, Maoqing Yao, and Junchi Yan. Raw2drive: Reinforcement learning with aligned world models for end-to-end autonomous driving (in CARLA v2). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Zhenjie Yang, Yilin Chai, Xiaosong Jia, Qifeng Li, Yuqian Shao, Xuekai Zhu, Haisheng Su, and Junchi Yan. Drivemoe: Mixture-of-experts for vision-language-action model in end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10678–10688, 2026.

Taeseong Yoon and Heeyoung Kim. Uncertainty estimation by flexible evidential deep learning. *Advances in Neural Information Processing Systems*, 38:118601–118641, 2026.

Wenhao Yu, Jie Peng, Huanyu Yang, Junrui Zhang, Yifan Duan, Jianmin Ji, and Yanyong Zhang. Ldp: A local diffusion planner for efficient robot navigation and collision avoidance. In *2024 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pp. 5466–5472. IEEE, 2024.

Kaiwen Zhang, Zhenyu Tang, Xiaotao Hu, Xingang Pan, Xiaoyang Guo, Yuan Liu, Jingwei Huang, Li Yuan, Qian Zhang, Xiao-Xiao Long, et al. Epona: Autoregressive diffusion world model for autonomous driving. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 27220–27230. IEEE, 2025.

Wenzhao Zheng, Ruiqi Song, Xianda Guo, Chenming Zhang, and Long Chen. Genad: Generative end-to-end autonomous driving. In *European Conference on Computer Vision*, pp. 87–104. Springer, 2024.

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 *The Thirteenth International Conference on Learning Representations*, 2025.

Yinan Zheng, Tianyi Tan, Bin Huang, Enguang Liu, Ruiming Liang, Jianlin Zhang, Jianwei Cui, Guang Chen, Kun Ma, Hangjun Ye, Long Chen, Ya-Qin Zhang, Xianyuan Zhan, and Jingjing Liu. Unleashing the potential of diffusion models for end-to-end autonomous driving. *arXiv preprint arXiv:2602.22801*, 2026.

Guangyao Zhou, Sivaramakrishnan Swaminathan, Rajkumar Vasudeva Raju, J Swaroop Guntupalli, Wolfgang Lehrach, Joseph Ortiz, Antoine Dedieu, Miguel Lazaro-Gredilla, and Kevin Patrick Murphy. Diffusion model predictive control. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=pvtgffHtJm.

Zewei Zhou, Tianhui Cai, Seth Zhao, Yun Zhang, Zhiyu Huang, Bolei Zhou, and Jiaqi Ma. Autovla: A vision-language-action model for end-to-end autonomous driving with adaptive reasoning and reinforcement fine-tuning. *Advances in Neural Information Processing Systems*, 38:27920–27956, 2026.

# A NOTATION

Table 3 collects the symbols used in the Method (§2), grouped by the pipeline stage that introduces them. Hyperparameter *values* are listed separately in Table 5.

Table 3: Notation used in §2.

| Symbol | Meaning |
|---|---|
| *Problem formulation* | |
| $k$ | control (decision) cycle index |
| $o_{1:k}$ | observation history up to cycle $k$ |
| $a \in \{\texttt{wait}, \texttt{commit}\}$ | action (wait or commit) |
| $\mathcal{Z} = \{\texttt{wait}, \texttt{commit}\}$ | decision-mode space |
| $Z_k \in \mathcal{Z}$ | ground-truth decision mode at cycle $k$ |
| $p_k^* = P(Z_k{=}\texttt{wait} \mid o_{1:k})$ | true posterior `wait` probability |
| $J_{\text{col}},\ J_{\text{delay}}$ | collision cost / per-cycle delay cost ($J_{\text{col}} \gg J_{\text{delay}}$) |
| $J(a, Z_k)$ | cost of taking action $a$ under mode $Z_k$ |
| $\pi_k^*$ | Bayes-optimal reference policy |
| *Belief from diffusion sampling (§2.3)* | |
| $q_\theta(y \mid o_{1:k})$ | diffusion planner's conditional (uncalibrated) distribution |
| $y,\ y_k^{(m)}$ | trajectory / the $m$-th sampled trajectory at cycle $k$ |
| $M$ | number of samples drawn per cycle |
| $\widehat{q}_{\theta,M}$ | finite-sample (empirical) approximation of $q_\theta$ |
| $\mathcal{D}$ | dynamics criterion mapping a trajectory to $\{\texttt{wait}, \texttt{commit}\}$ (App. F) |
| $\hat{p}_k(\texttt{wait})$ | sampled wait mass (fraction of samples labeled wait); $\hat{p}_k(\texttt{commit}){=}1{-}\hat{p}_k(\texttt{wait})$ |
| *Calibration: static certificate + online extension (§2.4)* | |
| $S_{\text{cal}},\ n$ | calibration set (`wait` frames) and its size |
| $Z_i = \texttt{wait}$ | expert outcome of calibration frame $i$ (`wait` class, from $\mathcal{D}$) |
| $s_i = -\log \hat{p}_i(\texttt{wait})$ | nonconformity score (on `wait` frames) |
| $\alpha$ | target unsafe-commit (miscoverage) level, fixed before calibration |
| $\widehat{Q}_{1-\alpha}(S_{\text{cal}})$ | $\lceil (n{+}1)(1{-}\alpha) \rceil$-th smallest calibration score |
| $C_k,\ C_k^\alpha$ | conformal prediction set at cycle $k$ (at level $\alpha$) |
| $\text{USR},\ \text{USR}_t$ | unsafe-commit rate $P(\pi = \texttt{commit} \mid Z = \texttt{wait})$ (static / per-step) |
| $p_0$ | blind-spot floor (`wait` frames with zero wait mass; Prop. 3) |
| $R_k$ | hand-crafted spatio-temporal risk map (App. G) |
| $\text{err}_t^{\text{proxy}}$ | dense risk-proxy error signal at cycle $t$ |
| $\alpha_t,\ \alpha_1$ | online adaptive level / its cap ($\alpha_1{=}\alpha$) |
| $\gamma$ | Proxy-ACI online learning rate |
| $\tau(\alpha),\ \tau_t$ | wait-mass threshold $\exp[-\widehat{Q}_{1-\alpha}(S_{\text{cal}})]$ (static / online) |
| $\epsilon_{\text{deploy}}$ | risk-map deployment recall (proxy-to-truth bridge; App. I) |
| *Decision rule (§2.5)* | |
| $\pi$ | measurable policy $o_{1:k} \mapsto \{\texttt{wait}, \texttt{commit}\}$ |
| $\mathcal{J}(\pi),\ \overline{\mathcal{J}}_\alpha(\pi)$ | one-step expected cost / its conformal upper bound |
| $\pi_\alpha^*$ | policy minimizing the certified cost bound |
| $\delta = J_{\text{delay}}/J_{\text{col}}$ | Bayes decision threshold on $p_k^*$ |
| $\rho(y),\ \rho_{\max}$ | trajectory collision probability / feasibility limit |
| $\{G_l\},\ \pi_l = \lvert G_l \rvert / M$ | sample clusters (modes) / their empirical weights |
| $D_k = -\sum_l \pi_l \log \pi_l$ | mode-partition entropy (diagnostic only) |
| $\tau_{\text{risk}}$ | online-proxy risk threshold |

## B Related Work

**End-to-end autonomous driving.** Planning-centric end-to-end driving frameworks have gained prominence following pioneering proof-of-concept studies (Hu et al., 2023; Jiang et al., 2023a; Sun et al., 2025). These frameworks integrate task-specific modules, such as object detection, tracking, and online mapping, into a unified architecture optimized for planning. More recently, efficient (pseudo-)closed-loop evaluation metrics (Cao et al., 2025; Dauner et al., 2024) and simulation platforms (Jia et al., 2024; Dosovitskiy et al., 2017) have accelerated the development of more tightly integrated approaches. Such methods predict ego trajectories or control actions directly from sensor observations (Jiang et al., 2026a; Chitta et al., 2022; Guo et al., 2026; Li et al., 2026; Kirby et al., 2026; Sun et al., 2026), moving closer to a fully end-to-end paradigm by reducing reliance on explicit intermediate representations. In parallel, another line of research incorporates vision-language models (VLMs) (Chen et al., 2024; Liu et al., 2023a; 2024; Bai et al., 2025) into end-to-end driving systems (Renz et al., 2025; Xu et al., 2025; Zhou et al., 2026; Yang et al., 2026), leveraging their broad semantic knowledge and reasoning capabilities to improve scene understanding and driving decisions.

**Multi-modal planning in driving.** Driving scenes are inherently multimodal: multiple ego trajectories may be plausible, and maintaining such alternatives is crucial for safely responding to uncertain behaviors of other road users. Early approaches generate structured sets of trajectory hypotheses or contingency plans to cover these possibilities (Chen et al., 2023; Mustafa et al., 2024; Li et al., 2023). More recently, advances in generative modeling (Kingma & Welling, 2013; Tian et al., 2024), particularly diffusion models (Croitoru et al., 2023; Ho et al., 2020; Song et al., 2021; He et al., 2024) and flow matching (Lipman et al., 2023; Liu et al., 2023b; Albergo & Vanden-Eijnden, 2023), have enabled planners to learn complex distributions over driving trajectories (Zheng et al., 2024; Xing et al., 2025; Zhang et al., 2025; Yang et al., 2024). DiffusionDrive (Liao et al., 2025) efficiently generates diverse and plausible plans through a truncated diffusion process initialized from multimodal anchors, whereas BridgeDrive (Liu et al., 2026c) formulates planning as an anchor-guided diffusion bridge that refines coarse proposals into context-aware trajectories for closed-loop driving. Extending this generative perspective beyond diffusion, GuideFlow (Liu et al., 2026a) introduces constrained flow matching, jointly trained with an energy-based model, to enforce physical constraints during generation while allowing driving aggressiveness to be controlled explicitly. Collectively, these methods use generative models primarily as trajectory generators, with samples serving either as candidates for downstream best-of-$N$ selection or as plans refined according to an external cost, as in diffusion MPC (Zhou et al., 2025). Our approach is complementary but conceptually distinct: rather than selecting or optimizing individual samples, we treat their empirical distribution as the object of interest and calibrate decisions over behavioral modes.

**Uncertainty estimation in driving.** Monte Carlo dropout (Gal & Ghahramani, 2016; Kendall & Gal, 2017; McAllister et al., 2017), deep ensembles (Lakshminarayanan et al., 2017; Rahaman et al., 2021), and evidential regression (Sensoy et al., 2018; Yoon & Kim, 2026) are widely used to quantify uncertainty in driving perception and trajectory prediction (Feng et al., 2020; Nguyen et al., 2026). Complementing these model-based approaches, conformal prediction (Vovk et al.; Shafer & Vovk, 2008; Angelopoulos & Bates, 2023) provides distribution-free calibration and has been applied to trajectory forecasting (Lindemann et al., 2023; Huang et al., 2025), perception (Doula et al., 2024; Andeol et al., 2023), and, most relevantly, safety-aware planning (Sun et al., 2023; Wang & Ning, 2025; Wang et al., 2024). Most existing methods use conformal prediction as a static, offline calibration layer that provides marginal coverage under exchangeability. Adaptive Conformal Inference (ACI) (Gibbs & Candes, 2021; Lekeufack et al., 2024) extends this framework to non-stationary environments by adjusting the miscoverage level online. Its adaptation guarantee, however, applies to the error signal driving the update; in short driving episodes, actual collisions are too sparse to provide timely feedback. To address this limitation, our monotonicity-preserving Proxy-ACI uses dense risk-map feedback as a proxy signal for rapid online adaptation while constraining the adaptive gate to remain at least as cautious as its offline-calibrated counterpart. Thus, although deployment shift precludes a renewed conformal coverage guarantee, the adapted policy can tighten, but never weaken, the static gate. The closest work, PlanCP (Sun et al., 2023), calibrates a *trajectory-level* return band for a diffusion dynamics model. In contrast, EvoBelief calibrates prediction sets over *behavioral modes* using the sampled wait mass, directly determining whether the safety-critical wait mode can be excluded. Experiments using an identical backbone show that this distinction is decisive (§3.3; Appendix C).

Table 4: **Headline closed-loop result on Bench2Drive**. EVOBELIEF improves both Driving Score (DS) and Success Rate (SR) over the strongest prior planner.

| Method | SR(%) | DS | Comfort. | Efficiency |
|---|---|---|---|---|
| Raw2Drive (Yang et al., 2025) | 50.24 | 71.36 | 22.42 | 214.17 |
| BLUE (Ling et al., 2026) | 76.18 ± 0.64 | 90.58 ± 0.12 | N/A | N/A |
| AutoMoT (Huang et al., 2026) | 74.09 | 89.42 | N/A | N/A |
| SparseDrivev2 (Sun et al., 2026) | 70.00 | 89.15 | 18.32 | 199.84 |
| DMW (Wang et al., 2026) | 71.56 | 82.72 | 34.62 | 237.06 |
| WPT (Jiang et al., 2026b) | 54.54 | 79.23 | 16.39 | 188.63 |
| KnowVal (Xia et al., 2026) | 69.03 | 88.42 | N/A | N/A |
| DriveMoE (Yang et al., 2026) | 48.64 | 74.22 | 15.31 | 175.96 |
| GuideFlow (Liu et al., 2026a) | 75.21 | 51.36 | N/A | N/A |
| CogDriver (Liu et al., 2026b) | 56.93 | 78.21 | 20.50 | 169.52 |
| CausalVAD (Tang et al., 2026) | 19.42 | 49.83 | N/A | N/A |
| LEAD (Nguyen et al., 2026) | 86.8 ± 0.7 | 95.2 ± 0.3 | N/A | N/A |
| BridgeDrive (Liu et al., 2026c) | 89.25 ± 0.50 | 96.34 ± 0.55 | 23.24 | 202.92 |
| EvoBelief (ours) | **91.36** ± 0.79 (+2.11) | **96.52** ± 0.62 (+0.18) | 12.68 ± 0.62 | 205.82 ± 0.89 |

**Deliberation and interaction in driving.** The importance of hesitation, yielding, and wait-then-commit behavior has long been recognized in rule-based (Treiber et al., 2000; Thrun et al., 2006; Bacha et al., 2008; Kesting et al., 2007; Luo et al., 2018; Cai et al., 2019) and game-theoretic planners (Chandra et al., 2022; Wang et al., 2021; Kedia et al., 2023; Chandra & Manocha, 2022; Sun et al., 2024). Our contribution is to recast deliberation as a *generative* phenomenon—the presence of multiple modes in a sampled trajectory distribution—and to connect this representation to a statistically calibrated decision rule. This provides a bridge between the classical notion of deliberation and the modern toolkit of generative modeling and distribution-free inference.

## C PLANCP BASELINE

To isolate the effect of *what* is calibrated from *how* it is calibrated, we adapt PlanCP (Sun et al., 2023) onto the same backbone, output representation (a joint path and speed profile), and conformal-calibration pipeline as EvoBelief. The single point of difference is the calibrated object: PlanCP calibrates a *trajectory-level* return band that provides marginal coverage, whereas EvoBelief calibrates the *behavior-level* wait probability with coverage conditioned on the safety-critical `wait` mode.

**Relation to the original PlanCP.** PlanCP's released implementation calibrates a quantile band over a *scalar trajectory-level return*—the task score of each sampled rollout—using the standard split-conformal quantile, giving marginal coverage over returns; it also augments the diffusion training loss with a quantile term (our Variant 2). We retain both ingredients but make two adaptations required by our setting. First, closed-loop urban driving exposes no per-frame task reward, so we replace PlanCP's rollout return with the risk-map return $s = -\sum_{t=1}^{T} R_k\big(\tau_{\mathrm{ref}}(t), t\big)$ computed from the same risk map $R_k$ EvoBelief consumes (App. G). Second, the original PlanCP draws a *single* trajectory per decision and scores complete rollouts, whereas we act at every frame on a batch of $M$ samples and reduce it to a geometric-medoid nominal to obtain one return per frame before calibrating. Taking the medoid of many samples rather than a single stochastic draw yields a more robust, lower-variance, on-manifold nominal; this adaptation therefore only *strengthens* the baseline relative to PlanCP's original single-sample form, making our comparison conservative—even this strengthened PlanCP trails EvoBelief. These substitutions leave the calibrated *object* unchanged—a marginal quantile band over a trajectory-level return—which is precisely the axis our study isolates against EvoBelief's behavior-level conditional certificate.

**Nonconformity score.** For each frame we draw a set of $M$ trajectory samples from the diffusion planner and reduce them to a single nominal plan $\tau_{\text{ref}}$, taken as their *geometric medoid*: the sample minimizing the summed $\ell_2$ distance—computed over the spatial $(x, y)$ channels only, so the speed columns do not distort the geometry—to all other samples. The medoid is an actual, on-manifold sample that sits at the densest mode of the set, and collapsing the multimodal samples to this single trajectory is itself the step that discards the alternative (e.g. waiting) modes. We score this nominal trajectory by its negative accumulated risk, $s = -\sum_{t=1}^{T} R_k\big(\tau_{\text{ref}}(t), t\big)$, where $R_k$ is the same time-sliced constant-velocity risk map used throughout (App. G), read along the medoid's forward-projected footprint; $s$ is thus a trajectory-level return, with higher values (less accumulated risk) indicating a safer plan. Calibrating a lower quantile of $s$ over held-out frames yields a marginal band over this trajectory-level return, in contrast to the mode-conditional certificate used by EvoBelief.

**From the calibrated band to a decision.** PlanCP calibrates a band over the trajectory-level return $s$; it does *not* calibrate the wait/commit decision. Waiting is instead the shared closed-loop interface that both methods must ultimately feed, and we derive it from PlanCP's band with a single threshold: at each frame we evaluate $s$ on the medoid and trigger a *wait*—a deterministic deceleration-to-stop that does not read the sampled distribution—whenever $s$ falls below $q_{\text{lo}}$, the calibrated lower quantile of the risk-map returns, i.e. whenever the nominal trajectory is deemed too risky; otherwise the ego simply executes the medoid unchanged. This "otherwise" branch is not a calibrated commit in EvoBelief's sense—no behavioral mode is being certified, the planner just runs its nominal plan. The wait behavior is therefore *induced* by thresholding one trajectory-level return, not a quantity that PlanCP calibrates directly—which, because the calibrated scalar carries no measurable `wait` event, is precisely why the resulting gate cannot be conditioned on the `wait` mode.

**Variant 1: PlanCP's calibrated gate on EvoBelief's own model.** We first apply the calibrated band *directly* to EvoBelief's trained model, without any retraining. Over $n$=38,826 calibration frames we collect the risk-map returns $\{s_i\}$ and take their $\alpha$-quantile at $\alpha$=0.10; this sets the deployed threshold $q_{\text{lo}} = -0.761$ (a value of $s$, hence on the risk-map return scale), and a frame waits when its medoid return satisfies $s < q_{\text{lo}}$. Because this single threshold on $s$ is marginal over whole trajectories, it cannot single out the `wait` mode and instead applies undifferentiated caution: it slows the ego whenever the nominal return is low, regardless of whether the low return reflects a genuine `wait` interaction. In closed-loop evaluation this variant attains a success rate of $81.81\%$ and a driving score of $91.23$.

**Variant 2: trained with an auxiliary conformal loss.** We next add a conformal training term to the planner's objective and retrain the model. The term penalizes the calibrated lower quantile of the return band and is weighted by $\lambda$=$10^{-3}$ at $\alpha$=0.1. Averaged over three random seeds, this variant reaches a success rate of $89.08 \pm 0.79$ and a driving score of $96.09 \pm 0.61$.

Both variants remain below EvoBelief ($91.36\%$ / $96.52$). Since the backbone, representation, and calibration pipeline are held fixed, the gap is attributable solely to the calibrated object: on this architecture, calibrating a trajectory-level return is less effective than calibrating the behavior-level wait probability, which is the central claim of §3.3.

# D PROOFS

## D.1 SPLIT-CONFORMAL MARGINAL COVERAGE (LEMMA A.1)

**Setup.** Let $(X_i, Y_i)_{i=1}^{n}$ be calibration points and $(X_{n+1}, Y_{n+1})$ the test point. Assume (A1) exchangeability, (A2) no ties, (A3) resolved labels, (A4) that the level $\alpha$ is fixed independently of the calibration scores, and (A5) that the sampler configuration $(M, \eta, \text{steps})$ used to compute the calibration scores is the one deployed. (A4) is implicit in the usual statement but is the assumption that fails for the *online* level $\alpha_t$, which is why the deployed system is covered by Lemma 2 rather than by a per-step application of Lemma 1. (A5) is required because the nonconformity score $s = -\log \hat{p}_k(Y)$ is computed from $M$ draws, so the sampler configuration is part of the score function and must be held fixed across calibration and deployment.

**Lemma 1** (Split-conformal marginal coverage)**.** *Let $s_i = s(X_i, Y_i)$, and $\hat{q}$ be the $k$-th smallest of $\{s_1, \ldots, s_n\}$ with $k = \lceil (n+1)(1-\alpha) \rceil$. Define $C(X_{n+1}) = \{y : s(X_{n+1}, y) \le \hat{q}\}$. Then $P(Y_{n+1} \in C(X_{n+1})) \ge 1 - \alpha$.*

*Proof.* By exchangeability, rank $R$ of $s_{n+1}$ is uniform on $\{1, \ldots, n+1\}$. If $R \le k$, then $s_{n+1} \le \hat{q}$. Hence $P(s_{n+1} \le \hat{q}) \ge k/(n+1) \ge 1 - \alpha$. ■ □

### D.2 Distribution-Free Safety Bound (Theorem A.2)

**Theorem 1** (Distribution-free safety bound)**.** *With decision rule* safe $\iff$ $\texttt{invalid} \notin C(X_{n+1})$*, $P($unsafe decision$) \le \alpha$, for any label set $\mathcal{Y}$ and any constraint family.*

*Proof.* Unsafe decision implies $Y_{n+1} \notin C(X_{n+1})$; apply Lemma 1. ■ □

### D.3 No-Weakening under Monotone Adaptation (Lemma A.3)

Lemma 1 requires a level chosen *before* the calibration scores are seen and a test point exchangeable with $S_{\text{cal}}$. Neither survives deployment: $\alpha_t$ is a function of the ongoing rollout, and §3.4(b) shows empirically that route-level scene shift breaks exchangeability. We therefore state what the monotone cap actually delivers—a *deterministic, distribution-free dominance* of the adaptive gate over the static one—and leave the absolute bound where it is earned, in Theorem 1.

**Lemma 2** (No-weakening / pointwise dominance)**.** *For a level $\alpha'$ let $\tau(\alpha') = \exp\big(-Q_{1-\alpha'}(S_{cal})\big)$ and let the wait gate be $G_{\alpha'}(x) = \mathbf{1}\big[\hat{p}_k(\texttt{wait} \mid x) \ge \tau(\alpha')$ and $M\hat{p}_k(\texttt{wait} \mid x) \ge 1\big]$. If $\alpha_t \le \alpha_1$ for all $t$, then $\tau(\alpha_t) \le \tau(\alpha_1)$ and*

$$G_{\alpha_t}(x) \;\ge\; G_{\alpha_1}(x) \qquad \textit{for every input } x \textit{ and every } t,$$

*i.e. the adaptive gate forces a wait whenever the static gate would. Consequently, for* any *input distribution—exchangeable or not, stationary or adversarial—and every $x$,*

$$P\big(\pi = \texttt{commit},\, Z = \texttt{wait} \mid X = x\big)\big|_{\textit{adaptive}} \;\le\; P\big(\pi = \texttt{commit},\, Z = \texttt{wait} \mid X = x\big)\big|_{\textit{static } \alpha_1},$$

*and whenever the hypotheses of Theorem 1 do hold, the right-hand side is at most $\alpha$, so the adaptive gate inherits the bound.*

*Proof.* $Q_{1-\alpha'}(S_{\text{cal}})$ is non-increasing in $\alpha'$, hence $\tau(\alpha') = \exp(-Q_{1-\alpha'})$ is non-decreasing in $\alpha'$, and $\alpha_t \le \alpha_1$ gives $\tau(\alpha_t) \le \tau(\alpha_1)$. A gate of the form $\{\hat{p}_k \ge \tau\}$ with the smaller $\tau$ fires on a superset of inputs, so $G_{\alpha_t} \ge G_{\alpha_1}$ pointwise; committing is the complement of firing, so the adaptive commit event is a subset of the static one. Intersecting with $\{Z = \texttt{wait}\}$ and applying monotonicity of probability gives the display. ■ □

**Remark 1** (Scope: a statement about the rule, not the rollout)**.** *The dominance is* pointwise in the input*. In closed loop, once the adaptive and static systems make different decisions their state trajectories diverge, so Lemma 2 does not assert that the deployed rollout visits a subset of the static rollout's unsafe states. It asserts that the deployed* decision rule *is never more permissive than the certified one—which is precisely what "the cap cannot weaken the offline guarantee" means, and all that we claim.*

### D.4 Long-Run Proxy Control: Unconstrained vs. Capped (Propositions A.4–A.5)

**Proposition 1** (Unconstrained ACI: two-sided long-run coverage (reference case))**.** *Under the* ***uncapped*** *update $\alpha_{t+1} = \alpha_t + \gamma(\alpha - \mathrm{err}_t)$ with $\mathrm{err}_t \in \{0, 1\}$ and $\alpha_t$ self-bounded in $[-\gamma, 1+\gamma]$, $\big|\frac{1}{T}\sum_{t \le T} \mathrm{err}_t - \alpha\big| \le \frac{1+2\gamma}{\gamma T} \to 0$.*

*Proof.* Telescoping gives $\alpha_{T+1} - \alpha_1 = \gamma \sum_{t \le T}(\alpha - \mathrm{err}_t)$; divide by $\gamma T$ and bound the left-hand side by the range of $\alpha_t$. ■ □

**Remark 2** (Proposition 1 does not describe the deployed system)**.** *Our update is capped at $\alpha_1$ (Lemma 2) and is applied only on commit cycles. The cap discards exactly the increments that would raise $\alpha_t$ back toward the target—the increments the telescoping argument relies on—so the two-sided conclusion is unavailable. The correct deployed statement is one-sided.*

**Proposition 2** (Capped Proxy-ACI: one-sided long-run control)**.** *Let $\mathcal{C}$ be the set of commit cycles and $N = |\mathcal{C}|$. On $t \in \mathcal{C}$ let $\alpha_{t+1} = \mathrm{clip}\big(\alpha_t + \gamma(\alpha - \mathrm{err}_t^{\mathrm{proxy}}),\ \ell,\ \alpha_1\big)$ with $\ell = -\gamma$ and $\alpha_1 = \alpha$, and let $\alpha_{t+1} = \alpha_t$ on wait cycles. Let $L = \#\{t \in \mathcal{C} : \alpha_t + \gamma(\alpha - \mathrm{err}_t^{\mathrm{proxy}}) < \ell\}$ count lower-saturation events. Then*

$$\frac{1}{N}\sum_{t\in\mathcal{C}} \mathrm{err}_t^{\mathrm{proxy}} \ \le\ \alpha \ +\ \frac{\alpha_1 - \ell}{\gamma N} \ +\ (1-\alpha)\frac{L}{N}.$$

*No matching lower bound holds: the empirical proxy rate may remain strictly below $\alpha$ for arbitrarily long.*

*Proof.* For $t \in \mathcal{C}$ set $u_t = \alpha_t + \gamma(\alpha - \mathrm{err}_t^{\mathrm{proxy}})$, so $\alpha_{t+1} = \max\{\ell, \min(u_t, \alpha_1)\} \le \max\{\ell, u_t\}$. If $u_t \ge \ell$ this is at most $u_t$. If $u_t < \ell$, then since $\alpha_t \ge \ell$ and $\mathrm{err}_t^{\mathrm{proxy}} \le 1$ we have $u_t \ge \ell + \gamma(\alpha - 1)$, hence $\max\{\ell, u_t\} = \ell \le u_t + \gamma(1-\alpha)$. In both cases $\alpha_{t+1} \le \alpha_t + \gamma(\alpha - \mathrm{err}_t^{\mathrm{proxy}}) + \gamma(1-\alpha)\mathbf{1}[u_t < \ell]$. Summing over $t \in \mathcal{C}$ and telescoping, $\alpha_{\text{last}} - \alpha_1 \le \gamma\sum_{t\in\mathcal{C}}(\alpha - \mathrm{err}_t^{\mathrm{proxy}}) + \gamma(1-\alpha)L$. Using $\alpha_{\text{last}} \ge \ell$ and rearranging yields the display. For the second claim take $\mathrm{err}_t^{\mathrm{proxy}} \equiv 0$: every increment is positive and is removed by the cap, so $\alpha_t \equiv \alpha_1$ and the empirical rate is $0 < \alpha$ for every $N$. ■ □

**Remark 3** (The asymmetry is the design, not an oversight)**.** *Proposition 2 retains the safety-relevant direction—the proxy-trigger rate cannot sit persistently* above *$\alpha$—and gives up the efficiency-relevant direction: the loop may be over-conservative indefinitely. This is the price of Lemma 2, and it matches what we observe: $\alpha_t$ rests* at *the cap during safe cruising and only dips (§3.4). The term $L$ counts saturation at the conservative end ($\tau_t \to 0$), which empirically occurs on confidently-wrong interactions.*

**Remark 4** (Proxy, not truth)**.** *Both propositions govern the proxy event $\mathrm{err}^{\mathrm{proxy}}$. Transfer to true collisions requires the empirical recall bridge $\epsilon_{deploy}$, measured in Appendix I; it is not a theorem. An analysis of the bright from proxy to truth is provided in Appendix I.*

### D.5 The Blind-Spot Floor (Proposition A.6)

**Proposition 3** (Impossibility floor of any wait-mass gate)**.** *Let $\hat{p}_k(\texttt{wait} \mid X) \in \{0, \frac{1}{M}, \ldots, 1\}$ be the sampled wait mass from $M$ draws, and let the gate be $G_\tau(X) = \mathbf{1}\big[\hat{p}_k(\texttt{wait} \mid X) \ge \tau$ and $M\hat{p}_k(\texttt{wait} \mid X) \ge 1\big]$—a forced wait must be supported by at least one sampled wait trajectory. Define*

$$p_0(\theta, M) \ =\ P\big(\hat{p}_k(\texttt{wait} \mid X) = 0 \ \big|\ Z = \texttt{wait}\big).$$

*Then for every $\tau \in [0,1]$,*

$$\mathrm{USR}(\tau) \ =\ P\big(\pi = \texttt{commit} \mid Z = \texttt{wait}\big) \ \ge\ p_0(\theta, M).$$

*Hence no threshold—and no calibration procedure that produces one—attains $\mathrm{USR} < p_0$; a target $\alpha < p_0$ is reachable only by a rule that ignores the belief and always waits.*

*Proof.* On the event $\{\hat{p}_k(\texttt{wait} \mid X) = 0\}$ the support condition $M\hat{p}_k \ge 1$ fails, so $G_\tau = 0$ for every $\tau$ and the system commits. Therefore $\{\hat{p}_k = 0\} \cap \{Z = \texttt{wait}\} \subseteq \{\texttt{commit}\} \cap \{Z = \texttt{wait}\}$, and monotonicity of probability gives the claim. ■ □

**Remark 5** (What $p_0$ does and does not measure)**.** *$p_0$ is a property of the generative model $\theta$ and the sample budget $M$, never of the calibration set. If $q(X)$ denotes the model's true conditional wait probability, then $p_0(\theta, M) = \mathbb{E}\big[(1 - q(X))^M \mid \texttt{wait}\big]$, which is non-increasing in $M$ and decreases to $P\big(q(X) = 0 \mid \texttt{wait}\big)$—the probability that the model places* no *mass on waiting at all. A measured $p_0$ at finite $M$ therefore mixes a sampling-budget effect with genuine support absence; we report $p_0$ at the deployed $M$ and attribute only the limit to $\theta$ alone. Either way $p_0$ is reducible by training or by a larger $M$, and never by the threshold—which is the sense in which it lower-bounds the achievable safety of the entire calibration layer.*

### D.6 Representation Separation: Why a Scalar Speed Head Cannot Be Calibrated (Proposition A.7)

The conformal machinery of §2.4 thresholds the sampled wait mass $\hat{p}_k(\texttt{wait})$. That quantity is meaningful only if the wait/commit event can be *read off* an individual sample. We show it cannot be when the planner's dynamic output is a single scalar.

**Setting.** Let a speed profile be $v = (v_1, \ldots, v_T) \in \mathbb{R}^T_{\geq 0}$ and let $\mathcal{D}$ be the dynamics criterion (Appendix F). Call a planner head *scalar-speed* if its dynamic output is a single target speed, i.e. each sample $y_m$ exposes only a summary $\sigma(v^{(m)}) \in \mathbb{R}$ of the profile it realizes, so the decision layer is restricted to functions measurable with respect to the $\sigma$-algebra generated by $\sigma$. The *profile* head exposes $v^{(m)}$ itself.

**Proposition 4** (Representation separation)**.** *There exist scenes $c_1, c_2$ with expert profiles $v^{(1)}, v^{(2)}$ such that*

$$\mathcal{D}(v^{(1)}) = \texttt{wait}, \qquad \mathcal{D}(v^{(2)}) = \texttt{commit}, \qquad \sigma(v^{(1)}) = \sigma(v^{(2)}),$$

*where $\sigma$ is the time-averaged speed. Consequently, for a scalar-speed head every labelling $\ell : \mathbb{R} \to \{\texttt{wait}, \texttt{commit}\}$ assigns the same label in both scenes, the induced wait mass satisfies $\hat{p}_k(\texttt{wait} \mid c_1) = \hat{p}_k(\texttt{wait} \mid c_2)$, and therefore* every *gate $\mathbf{1}[\hat{p}_k(\texttt{wait}) \geq \tau]$—for every $\tau$ and every calibration set—takes the same action in both. Its error on the pair is at least $1/2$ under any prior placing mass on both scenes. The profile head separates them exactly, since $\mathcal{D}$ is a function of $v$ that is not a function of $\sigma(v)$.*

*Proof.* Take $v^{(1)} = (8, 4, 0, 0, 4, 8)$ (stop, then go) and $v^{(2)} = (4, 4, 4, 4, 4, 4)$ (proceed slowly throughout), in m/s, with the deployed constants of Appendix J ($\theta = \theta' = 1.0$, $v_{\text{low}} = 2.0$, $\kappa = 0.5$). Their time averages coincide at $4$. For $v^{(1)}$ the near-term change is $\Delta v = 4 - 8 = -4 < -\theta$, so the first branch of Alg. 2 fires and $\mathcal{D}(v^{(1)}) = \texttt{wait}$. For $v^{(2)}$ we have $\Delta v = 0$, $\Delta v_{\text{net}} = 0$ and $\max_i v_i^{(2)} = 4 > v_{\text{low}}$, so no wait branch fires and $\mathcal{D}(v^{(2)}) = \texttt{commit}$. Since $\sigma(v^{(1)}) = \sigma(v^{(2)})$, any composition $\ell \circ \sigma$ is constant on $\{v^{(1)}, v^{(2)}\}$; hence so is the wait mass, and hence so is the gate's action. One of the two actions is wrong, giving error $\geq 1/2$. ■ □

**Remark 6** (What this does and does not explain)**.** *The proposition is a worst-case, two-point argument. It establishes that the representation gap is* sufficient *to render the conformal gate uninformative on a scalar-speed head, which is why we read the scalar rows of Table 2 as a representational limitation rather than a tuning failure. It does* not *claim this is the only mechanism behind those cells, nor that scalar-speed heads are inferior planners in general—only that the wait/commit event lies outside their output $\sigma$-algebra, leaving the calibration layer nothing to threshold. It also delimits our own method: the profile head makes the event* expressible*, which is necessary but not sufficient; whether the model places sampling mass on it is the separate question measured by $p_0$ (Prop. 3).*

## E Cost-Optimality of the Conformal Gate

We show the one-sided gate of §2.5 is exactly the policy that minimizes a conformal upper bound on the per-cycle decision cost, without access to the true posterior $p_k^*$. Recall the wait-mass threshold and conformal set

$$\tau(\alpha) = \exp\left[-\widehat{Q}_{1-\alpha}(S_{\text{cal}})\right], \qquad C_k^\alpha = \{Z \in \mathcal{Z} : \hat{p}_k(Z) \geq \tau(\alpha)\}. \tag{9}$$

Optimizing over all measurable policies $\pi : o_{1:k} \mapsto \{\texttt{wait}, \texttt{commit}\}$, the one-step expected cost is

$$\mathcal{J}(\pi) = J_{\text{col}} P(Z_k = \texttt{wait}, \pi = \texttt{commit}) + J_{\text{delay}} P(\pi = \texttt{wait}). \tag{10}$$

Although the first probability depends on the unknown true posterior, its event satisfies

$$\{Z_k = \texttt{wait}, \pi = \texttt{commit}\} \subseteq \{Z_k = \texttt{wait},\ \texttt{wait} \notin C_k^\alpha\} \cup \{\texttt{wait} \in C_k^\alpha, \pi = \texttt{commit}\}. \tag{11}$$

The class-conditional certificate $\text{USR} = P(\texttt{wait} \notin C_k^\alpha \mid Z_k = \texttt{wait}) \leq \alpha$ (§2.4) bounds the first term, $J_{\text{col}} P(Z_k = \texttt{wait}, \texttt{wait} \notin C_k^\alpha) = J_{\text{col}}\, P(\texttt{wait} \notin C_k^\alpha \mid Z_k = \texttt{wait})\, P(Z_k = \texttt{wait}) \leq J_{\text{col}} \alpha$, so for every policy

$$\mathcal{J}(\pi) \leq \overline{\mathcal{J}}_\alpha(\pi) := J_{\text{col}}\alpha + J_{\text{col}} P(\texttt{wait} \in C_k^\alpha, \pi = \texttt{commit}) + J_{\text{delay}} P(\pi = \texttt{wait}). \tag{12}$$

Minimizing this bound needs no knowledge of $p_k^*$. The term $J_{\text{col}}\alpha$ is constant in $\pi$; if $\texttt{wait} \in C_k^\alpha$, committing contributes $J_{\text{col}}$ whereas waiting contributes $J_{\text{delay}}$, and if $\texttt{wait} \notin C_k^\alpha$, committing contributes zero whereas waiting still contributes $J_{\text{delay}}$. Since $J_{\text{col}} > J_{\text{delay}} > 0$ and $\mathcal{Z}$ is binary so that $\texttt{wait} \in C_k^\alpha \iff \hat{p}_k(\texttt{wait}) \geq \tau(\alpha)$, the pointwise minimizer $\pi_\alpha^* \in \arg\min_\pi \overline{\mathcal{J}}_\alpha(\pi)$ is exactly the one-sided threshold gate (8) of §2.5. Because the objective is a conformal upper bound rather than the inaccessible true cost, the rule is not Bayes-optimal for (10); had $p_k^*$ been known, minimizing the true cost would recover the posterior threshold $\delta = J_{\text{delay}}/J_{\text{col}}$.

## F The Dynamics Criterion

The **dynamics criterion** $\mathcal{D}$ is a single fixed rule that maps a frame to $\{\texttt{wait}, \texttt{commit}, \texttt{excluded}\}$. It serves two purposes with one definition: it labels calibration frames with the expert's realized outcome (§2.4), and it classifies each sampled trajectory as wait or commit at deployment (§2.5). It has two stages.

**Stage 1 — decision-point gate (offline only).** A frame is a wait/commit *decision point* iff it involves a genuine maneuver **and** a yield object; otherwise it is *excluded*. A maneuver is one of: a junction turn (navigation command LEFT/RIGHT); a lane change (near the lane-change point *and* with the ego actively steering — the future route's heading deviates beyond a threshold or |steer| exceeds a threshold); a go-around (an obstacle on the ego's future path, or a stopped vehicle on the path while lane-changing); or avoiding a crossing agent (a pedestrian or laterally-moving vehicle on the path, a walker hazard, or a cut-in). A yield object is any agent lying on the ego's future path — within a distance $d_{\text{conf}}$ of a future route point, excluding parked cars, traffic lights and stop signs — or a vehicle/walker hazard or a cut-in. Frames under a red light or stop sign are excluded as passive stops. Because this stage reads privileged log signals (navigation command, hazard flags, ground-truth boxes), it is used **only** to select which frames enter the calibration set; it is not evaluated online.

**Stage 2 — wait/commit from the speed profile.** From a future speed profile with start speed $v_0$, near-term change $\Delta v = v_k - v_0$, net change $\Delta v_{\text{net}} = \overline{v}_{\text{late}} - \overline{v}_{\text{early}}$, peak $v_{\max}$ and end $v_{\text{end}}$, the profile is a **wait** if it decelerates near-term ($\Delta v < -\theta$), or brakes over the horizon and ends slow ($\Delta v_{\text{net}} < -\theta'$ and $v_{\text{end}} < \max(v_{\text{low}}, \kappa v_0)$; this captures hard or delayed braking that does not fully stop), or stays near-stationary ($v_{\max} < v_{\text{low}}$); otherwise it is a **commit** (it accelerates, $\Delta v > +\theta$, or proceeds). Stage 2 depends *only* on the speed profile, so it applies unchanged to the expert's realized future — giving the hindsight label $Z$ for a calibration frame — and to each sampled trajectory at deployment — giving the sampled wait mass $\hat{p}_k(\texttt{wait})$, the fraction labeled wait. Threshold values ($\theta, \theta', v_{\text{low}}, \kappa, d_{\text{conf}}, \dots$) are listed with the other hyperparameters (Table 5).

## G Risk-Map Construction

The risk map $R_k$ is a hand-crafted, spatio-temporal occupancy estimate with **no** learned parameters and no ground-truth supervision — an interpretable safety summary that serves three roles: a conditioning input to the diffusion planner, the feasibility filter of §2.5, and the dense online-ACI proxy. We describe the deployed constant-velocity variant; the only train/inference difference is the source of the $t{=}0$ agent state (ground-truth boxes in training, predicted boxes at inference).

**Grid and inputs.** $R_k$ is an ego-centric BEV stack of $T$ time slices, $R_k \in [0,1]^{T\times H\times W}$, aligned one-to-one with the BEV semantic map (row = lateral $y$, column = longitudinal $x$); slice $t$ encodes where the scene is undrivable at future time $t\,\Delta t$ (deployed horizon 1.5 s). All inputs come from the perception heads: the BEV semantic segmentation and the detected agent boxes (position, heading, extent, and a dense predicted speed field).

**Static layer.** Cells that are off-road or a static obstacle are undrivable ($= 1$) and time-invariant; road surface and lane markings are drivable (the ego may cross them).

**Dynamic layer.** Each detected agent is propagated with a constant-velocity model using its *predicted* speed $v$ (never ground truth): its center at time $t$ is $(x_0, y_0) + v\,t\,(\cos\psi_0, \sin\psi_0)$. Around this pose we rasterize a soft, heading-aligned footprint

$$f(x) = \exp\Big(-\tfrac{1}{2}\big[(d_\parallel/\sigma_\parallel)^2 + (d_\perp/\sigma_\perp)^2\big]\Big), \tag{13}$$

**Algorithm 2** The dynamics criterion $\mathcal{D}$.

```
    # —- Stage 1: decision-point gate (offline only; uses privileged log signals) —-
 1: function IsDecisionPoint(frame)
 2:     if light_hazard or stop_sign_hazard then
 3:         return False                                                  ▷ passive stop
 4:     end if
 5:     is_turn ← command ∈ {LEFT, RIGHT}
 6:     is_lane_change ← |signed_dist_to_lane_change| < d_lc and (route_turn_deg > θ_turn or
        |steer| > θ_steer)
 7:     is_go_around ← obstacle_on_path or (stopped_vehicle_on_path and is_lane_change)
 8:     is_avoid_cross ← walker_on_path or crossing_vehicle_on_path or walker_hazard or cut_in
 9:     maneuver ← is_turn or is_lane_change or is_go_around or is_avoid_cross
10:     if not maneuver then
11:         return False                                                  ▷ no maneuver
12:     end if
13:     yield_object ← (n_conflict > 0) or vehicle_hazard or walker_hazard or cut_in
14:     return yield_object                                               ▷ else: excluded
      ▷ n_conflict = # GT agents (excl. parked cars / lights / signs) within d_conf of a future route point
    (an agent on the ego path)
15: end function
    # —- Stage 2: wait/commit from the future speed profile (offline AND online) —-
16: function Classify(v[1..T])
17:     v_0 ← v[1];    Δv ← v[k] − v_0
18:     Δv_net ← mean(v[last third]) − mean(v[first third])
19:     if Δv < −θ then
20:         return Wait                                                   ▷ decelerate near-term
21:     end if
22:     if Δv_net < −θ′ and v[T] < max(v_low, κ · v_0) then
23:         return Wait                                                   ▷ delayed / hard braking
24:     end if
25:     if Δv > +θ then
26:         return Commit                                                 ▷ accelerate
27:     end if
28:     if max(v) < v_low then
29:         return Wait                                                   ▷ near-stationary
30:     end if
31:     return Commit                                                     ▷ proceed
32: end function
    # —- two uses of the one rule —-
    # calibration label (offline): for frames with IsDecisionPoint(frame),
    #     Z = Classify(expert_realized_speed_profile)
    # deployment wait mass (online): p̂_k(wait) = (1/M) Σ_m 1[Classify(sample_m.v) ==
    Wait]
```

where $d_{\parallel}, d_{\perp}$ are the along-/cross-track distances to the agent's box (half-length, half-width), so the footprint is flat inside the box and decays outside. The standard deviations grow with the horizon to model prediction uncertainty, $\sigma_{\parallel} = \sigma_0 + c_{\parallel}\, v\, t$ and $\sigma_{\perp} = \sigma_0 + c_{\perp}\, v\, t$, with $\sigma_{\perp}$ capped at one lane half-width $w_{\text{lane}}$ so a slightly-angled vehicle keeps its risk within the lane. Agents combine by a pixel-wise maximum; the dynamic risk is clipped to the drivable surface and the static undrivable layer is always blocked, so cells an agent vacates become drivable again in later slices.

**Reading a trajectory.** $R_k(x, t)$ is the probability that cell $x$ is undrivable at future time $t$. A candidate trajectory $y$ is scored by rolling the ego forward with a kinematic (bicycle) model and reading the map along its time-sliced footprint: its collision probability is $\rho(y) = 1 - \prod_s \big(1 - R_k(y(s), s)\big)$ (§2.5), and the online proxy uses the peak $\max_{s \in [t, t+T]} R_k(y(s), s)$. The construction constants $(\Delta t, \sigma_0, c_{\parallel}, c_{\perp}, w_{\text{lane}})$ are listed in Appendix J.

## H RISK-MAP-ALONE BASELINE

The *risk-map-alone* baseline (arm (v) in §3.1) isolates the marginal value of the calibrated belief layer beyond the hand-crafted safety filter: it keeps the identical backbone, sample budget, and risk map, but removes the belief structure (clustering into modes), the primary conformal wait/commit gate, and the online adaptation. It reduces the decision rule (Algorithm 1) to its risk-map selection primitive alone.

At each control cycle it draws the *same* $M$ trajectories, builds the *same* constant-velocity risk map $R_k$, and simply executes the single sampled trajectory whose peak look-ahead risk is lowest (Algorithm 3). There is no wait mass, no conformal threshold $\tau_t$, no adaptive level $\alpha_t$, and no speed grafting: the executed plan is always a real, coherent sample.

**Algorithm 3** Risk-map-alone baseline (per control cycle $t$).

**Require:** observation $o_t$; sample budget $M$
1: $\{y_1..y_M\} \leftarrow \text{SAMPLE}(o_t, M)$ ▷ same belief sampler, no clustering
2: $R_k \leftarrow \text{RISKMAP}(o_t)$ ▷ same dense collision-rate field
3: **for** $m = 1..M$ **do**
4: $\quad s_m \leftarrow \max_{s\in[t,t+T]} R_k(y_m(s), s)$ ▷ peak look-ahead risk (§2.5)
5: **end for**
6: $m^* \leftarrow \arg\min_m s_m$ ▷ lowest-risk sample; no gate, no conformal set, no $\alpha$ update
7: **return** executed trajectory $y_{m^*}$

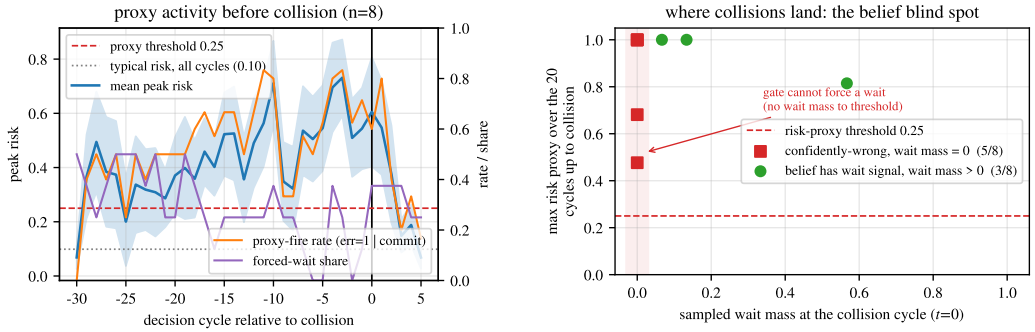
**Comparability to Algorithm 1.** Both rules share an identical action space—execute one of the $M$ sampled trajectories, never a synthesized full stop (a forced wait in Algorithm 1 likewise executes the lowest-risk *sampled* candidate, not a grafted brake). The only ablated element is the conformal gate's calibrated switch between the wait selection rule and the commit selection rule (consensus / $\arg\min$ peak risk), together with the online $\alpha_t$ update. Because peak look-ahead risk scales with the ego forward projection, and slower samples project shorter footprints, $\arg\min$ peak risk is biased toward slower candidates: the baseline therefore realizes an *uncalibrated conservatism-favor* planner that approximates the wait behavior without the certificate.

## I PROXY-TO-TRUTH BRIDGE

Proxy-to-truth bridge: the risk proxy covers every collision, yet recall $\neq$ prevention. To connect proxy coverage to true safety we measure the risk map's deployment recall $\epsilon_{\text{deploy}}$, the empirical step behind $P(\text{collision}) \leq \alpha + \epsilon_{\text{deploy}}$. Over the full 220-route deployment there are only $8$ real collisions; for each we ask whether the risk proxy fired ($\text{err}^{\text{proxy}}{=}1$) within $W$ decision cycles before impact. Recall rises from $0.23$ at $W{=}0$ to $\mathbf{1.00}$ at $W{=}20$ (the interaction-resolution window), i.e. $\epsilon_{\text{deploy}} \to 0$: at the deployment window the dense proxy *never misses* a real collision, so bounding the proxy-trigger rate (Prop. 2) upper-bounds the true collision rate with a near-zero empirical gap. This is not a base-rate artifact—the proxy fires $2.8\times$ more densely in the pre-collision window than its suite-wide rate ($6.85$ vs. $2.47$ expected fires), and the mean peak risk ramps from the background level ($0.10$) to $\sim 0.6$ in the cycles before impact (Fig. 5a). *Yet recall is not prevention*: $5/8$ ($62.5\%$) collisions occur with *zero* sampled wait mass—the confidently-wrong blind spot—where the belief offers nothing to threshold and the gate cannot force a wait even at $\tau \to 0$ (Fig. 5b). The residual collisions are therefore dominated by the belief blind spot $p_0$ (reducible only by training, Prop. 3), *not* by proxy misses: the calibration/adaptation layer is sound, and the remaining floor is exactly the $p_0$ we characterized offline.

## J CONCRETE INSTANTIATION, HYPERPARAMETERS, AND COMPUTATION COST

The planner is a TransFuser-style perception–planning network with an anchor-conditioned diffusion head emitting a joint path-and-speed-profile trajectory; per cycle we draw $M{=}30$ samples and cluster them into spatial modes. We prescribe the safety level $\alpha{=}0.10$ before calibration; the route-disjoint




(a) risk proxy before the collision

(b) the belief blind spot

Figure 5: **Proxy-to-truth bridge over the full** 220**-route deployment (**8 **real collisions).** *(a)* Profile aligned at the collision ($t$=0): the mean risk proxy ramps from the background level (0.10, over all cycles) to ∼0.6 just before the collision, well above the trigger threshold (0.25). The proxy-fire rate (orange) and forced-wait share (purple) show that as the collision nears the system increasingly *forces a wait*, so $\mathrm{err}^{\mathrm{proxy}}$—scored only on commits—dips at the collision, not because risk vanishes. *(b)* Every collision is preceded by a proxy warning within 20 cycles ($\epsilon_{\mathrm{deploy}} \to 0$), yet $5/8$ land in the confidently-wrong blind spot (wait mass $= 0$), where the gate cannot force a wait even at $\tau \to 0$: recall $\neq$ prevention, and the residual is the training-reducible floor $p_0$, not a proxy miss.

set $S_{\mathrm{cal}}$ ($n$=1,398) then gives $\tau \approx 0.23$ and a blind-spot floor $p_0 \approx 0.024$. Online adaptation uses learning rate $\gamma$=0.05 and is capped at $\alpha_1$=$\alpha$ (Lemma 2); a forced wait executes the slowest feasible sample. The two hand-crafted components—the risk map (§2.5) and the dynamics criterion $\mathcal{D}$ (§2.4). Table 5 provides detailed information about the specific hyperparameters.

Table 5: Hyperparameters.

| Symbol | Meaning | Value | Ref |
|---|---|---|---|
| $M$ | samples per cycle | 30 | §2.3 |
| $\eta$ | sampler stochasticity | 0.5 | §2.3 |
| — | denoising steps | 20 | §2.3 |
| — | cluster merge dist / speed weight / min count | 0.3 m / 0.2 / 2 | §2.3 |
| $\alpha$ | target unsafe-commit level | 0.10 | §2.4 |
| $\tau$ | wait-mass threshold at $\alpha$ | 0.23 | §2.4 |
| $\gamma$ | online (ACI) learning rate | 0.05 | §2.4 |
| $n$ | $S_{\mathrm{cal}}$ size | 1,398 | §2.4 |
| $p_0$ | intrinsic blind-spot floor | 0.024 | §2.4 |
| $\rho_{\max}$ | feasibility tolerance | 0.30 | §2.5 |
| $\tau_{\mathrm{risk}}$ | online-proxy risk threshold | 0.25 | §2.5 |
| $T, \Delta t$ | risk horizon / slice | 1.5 s / 0.5 s | §2.5, G |
| $\sigma_0, c_\parallel, c_\perp$ | footprint base / long / lat growth | 0.5 m / 0.1 / 0.03 | G |
| $w_{\mathrm{lane}}$ | lateral-spread cap | 1.75 m | G |
| $\theta, \theta'$ | wait/commit speed margins | 1.0 m/s | F |
| $v_{\mathrm{low}}, \kappa$ | near-stationary / brake-end fraction | 2.0 m/s / 0.5 | F |
| $d_{\mathrm{conf}}$ | on-path conflict distance | 3.5 m | F |

**Computation cost.** We profile the full per-cycle pipeline (diffusion sampling with $\eta$=0.5 and 20 denoising steps, conformal gate, and risk-map construction) on a single NVIDIA H20 with CUDA synchronization, over 100 frames after dropping 10 warm-up frames (Table 6). Latency is dominated by the shared perception backbone, so per-cycle cost does not grow linearly with the sample budget $M$: a single trajectory already costs a median 211.9 ms (the fixed backbone and denoising overhead paid once per cycle), and up to $M$≈10 the extra samples are batched with little marginal cost (253.2 ms median, 4.0 FPS). Only for large budgets does the per-sample sampling cost

become visible—$M{=}30$ runs at $447.4$ ms median (2.2 FPS) and $M{=}100$ at $1034.9$ ms (1.0 FPS). The deployed configuration $M{=}30$ therefore trades a $\sim 2\times$ slowdown over the single-sample planner for the multimodal belief the decision layer needs. The conformal gate and the Proxy-ACI update themselves are negligible (sub-millisecond), so the belief–safety layer adds essentially no cost beyond the sampling it consumes.

Table 6: Per-cycle wall-clock latency vs. sample budget $M$ (NVIDIA H20, $\eta{=}0.5$, 20 denoising steps, 100 frames averaged).

| $M$ (samples/cycle) | median (ms) | mean (ms) | p95 (ms) | min (ms) | FPS (median) |
|---|---|---|---|---|---|
| 1 | 211.9 | 240.3 | 375.3 | 141.9 | 4.7 |
| 5 | 205.3 | 244.9 | 428.1 | 166.1 | 4.9 |
| 10 | 253.2 | 286.0 | 447.3 | 196.1 | 4.0 |
| 30 | 447.4 | 461.1 | 602.6 | 357.9 | 2.2 |
| 100 | 1034.9 | 1047.6 | 1144.3 | 977.2 | 1.0 |

## K Limitations and Outlook

Several limitations remain and motivate future work, three of which are particularly relevant. First, behavioral modes are defined by a hand-crafted dynamics criterion rather than discovered by the model itself. Learning the mode partition end-to-end is a natural next step. Second, conformal calibration is applied *post hoc* and does not feed back into training, so the model is never optimized for calibration. Folding the certificate into the learning objective could instead train the model to produce a belief that directly supports the wait/commit decision, rather than one corrected only after the fact. Third, we validate the paradigm only in driving. Extending it to richer interactive settings—such as embodied AI agents—would test whether interpreting generative multimodality as a calibrated behavioral belief generalizes beyond autonomous driving.