REARL: REFERENCE-GUIDED ONLINE CALIBRATION OF TRAFFIC SIMULATION FOR AUTONOMOUS DRIVING

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurate traffic simulation is important for developing and evaluating autonomous driving systems, but local driving rules and fixed initial conditions can drift from real traffic as a rollout evolves. Existing rule-based, data-driven, and initialization-focused methods either provide limited adaptation during a rollout or replace the native simulator with a learned policy. We propose REARL, a reference-guided online calibration framework that corrects this drift while retaining the simulator's native controller. REARL builds a library of representative scenes from the highD highway trajectory dataset, retrieves a reference from the observed simulation prefix, and measures speed-distribution and spacing discrepancies at scheduled checkpoints. When a discrepancy exceeds a calibrated threshold, a large language model (LLM) selects one simulator-native action from a time-aligned real-traffic snapshot; otherwise, the Intelligent Driver Model (IDM) with the Minimizing Overall Braking Induced by Lane changes (MOBIL) controller (IDM+MOBIL) continues unchanged. In a paired development study on highD scenes, REARL–Qwen reduces speed Hellinger distance from 0.3313 to 0.2945 and spacing error from 0.9190 to 0.8448 relative to Base. Component controls show that updating the reference improves speed alignment, deterministic correction achieves lower spacing error at a substantial jerk and headway cost, and a retrospective matched-budget periodic comparator has lower error than sparse discrepancy triggering. These results support online reference-guided calibration as a practical middle ground between fixed initialization and full policy replacement.

1 INTRODUCTION

Traffic simulation provides a controlled setting for developing and evaluating autonomous driving systems (Bolte et al., 2019; Zhou & Beyerer, 2023). A plausible initial state, however, does not ensure that simulated speed and spacing remain consistent with recorded traffic as vehicles interact. Local driving rules can produce coherent maneuvers while the aggregate rollout departs from the intended traffic pattern. We ask: *how can an evolving simulation use real traffic to correct this discrepancy without replacing its native motion model?*

Rule-based simulators encode car-following and lane-changing behavior through predefined dynamics (Yan et al., 2023; Jiang et al., 2024). Data-driven policies learn behavior from trajectories (Jaipuria et al., 2020; Wang et al., 2017), while initialization methods such as CRITICAL use real scenes and an LLM to configure the starting environment (Tian et al., 2024). These approaches provide different routes to plausible traffic, but an initial match or a learned local policy does not by itself specify how to compare the current rollout with a real reference and implement a later correction. Online calibration must resolve which recorded scene remains relevant, which discrepancy warrants intervention, and how to translate that evidence into an executable action.

Our central hypothesis is that a reference selected from the evolving simulation can support useful corrections beyond initialization. REARL couples a fixed library of representative highD scenes (Krajewski et al., 2018) with online reference selection and brief interventions. At scheduled checkpoints, it first matches the observed simulation prefix to the library and then measures speed and spacing discrepancies. The retrieved snapshot and current vehicle context condition an

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.

LLM that selects one simulator-native action. Applying that action for one policy step, followed by a return to IDM+MOBIL, preserves the underlying simulator while allowing aggregate traffic statistics to change through vehicle-level corrections.

We evaluate this hypothesis with paired rollouts and component substitutions. The main configuration improves both target metrics, and freezing the retrieved reference weakens speed alignment. Other comparisons delimit the result: direct rule-based correction achieves better spacing at a larger smoothness cost, and a retrospective matched-budget periodic comparator has lower error than sparse discrepancy triggering. Accordingly, the evaluation separates evidence for online reference-guided correction from claims about LLM necessity or adaptive timing. The five evaluated scenes also informed prompt development, so this is a development study rather than an independent generalization test.

Our contributions are:

- A formulation of traffic-rollout calibration that separates an evolving online reference from a fixed evaluation target.
- A correction mechanism that connects prefix-based retrieval and discrepancy measurement to brief, snapshot-guided actions within a native simulator.
- A paired empirical study of aggregate alignment, component roles, and temporal behavior, including controls that expose limits of discrepancy-triggered timing and the spacing–smoothness trade-off.

2 Related work

High-fidelity simulation supports the development and evaluation of autonomous driving. We review traditional, data-driven, and LLM-enhanced approaches, focusing on how they represent traffic interactions and maintain agreement with real traffic.

2.1 Traditional Simulation Methods

Traditional simulation methods rely on predefined rules and heuristic models to simulate traffic behavior. Representative tools include Simulation of Urban MObility (SUMO) (Lopez et al., 2018), VISSIM (PTV Planung Transport Verkehr AG, 2012), AIMSUN (AIMSUN, 2022), and highway-env (Leurent, 2018). Car-following models (Bando et al., 1995; Treiber et al., 2000) and lane-changing models (Kesting et al., 2007; Erdmann, 2015) provide interpretable descriptions of local interactions. Their fidelity depends on how well the selected rules and calibrated parameters represent the target conditions. Modeling time-dependent decisions and interactions remains an active topic, as illustrated by work on cognitive models of discretionary lane changing (Chen et al., 2024).

2.2 Data-driven Simulation Methods

Data-driven approaches use real-world traffic data to improve simulation realism. Techniques include extracting and clustering traffic primitives (Wang & Zhao, 2018), modeling scene distributions with factor graphs (Wheeler & Kochenderfer, 2016), and generating multi-vehicle trajectories (Ding et al., 2018). Neural networks learn car-following and lane-changing behavior (Wang et al., 2017; Zhu et al., 2018; Xie et al., 2019). For example, Mo et al. (2021) combine physics-based car-following models with neural networks to improve acceleration prediction. Liu et al. (2022) use long short-term memory (LSTM) networks and quantile regression to model stochastic, interactive driving behavior. The data-driven simulation system of Liu et al. (2023) uses a Sim-Hybrid Retraining Constrained LSTM model to jointly represent driving behaviors. D2Sim likewise addresses human driving behavior modeling and simulation (Zhang et al., 2022).

Behavior cloning learns driving policies from demonstrations, with dataset bias and generalization remaining important challenges (Codevilla et al., 2019). Gated recurrent units (GRUs) encode temporal histories (Cho et al., 2014); the behavior-cloning gated recurrent unit (BC-GRU) baseline combines this recurrent architecture with behavior cloning. Human-like and Diverse Driving Behavior Generation (HAD-Gen) (Wang et al., 2025) clusters driving styles, learns style-specific rewards through maximum-entropy inverse reinforcement learning, and combines offline pretraining with

multi-agent reinforcement learning for controllable scenario generation. The adapted HAD-Gen baseline draws on its style-conditioned behavior generation, rather than reproducing the complete training pipeline. REARL instead uses a retrieved real scene to guide brief corrections around an existing controller, allowing the reference to change during a rollout without training a replacement policy.

2.3 LLM-enhanced Autonomous Driving Simulation

LLMs offer reasoning and contextual processing capabilities (Zhao et al., 2023; Madaan et al., 2022) that have motivated applications in autonomous driving (Wu et al., 2025; Yang et al., 2023; Li et al., 2024). LinguaSim uses natural language instructions to generate interactive multi-vehicle test scenarios (Shi et al., 2025). Multi-agent LLM frameworks also support design space exploration for autonomous driving systems (Shih et al., 2025), while DiLu combines reasoning, reflection, and memory for driving decisions (Wen et al., 2023).

CRITICAL is particularly relevant: it combines real traffic data with LLM-guided scenario configuration and uses training feedback to update configurations across episodes (Tian et al., 2024). This feedback operates at a different stage from correcting discrepancies during an individual rollout. Even when starting conditions resemble recorded traffic, subsequent vehicle interactions can change the speed and spacing distributions. REARL addresses this within-rollout calibration problem by monitoring those discrepancies, retrieving a relevant real snapshot, and applying discrete corrections while retaining the simulator's native motion model. We evaluate the roles of LLM actions and intervention timing separately in our component study.

3 Methodology

3.1 Calibration problem and testable mechanisms

Let x_t denote the simulated traffic state at sampled frame t, with $n = 10$ controlled background vehicles. A fixed library $\mathcal{M} = \{s_1, \ldots, s_{15}\}$ contains representative real scenes, each with aligned speed and position observations. The simulator runs at 15 Hz and records one frame per 1 Hz policy step. Over $T = 60$ frames, REARL may insert actions at checkpoints; IDM+MOBIL governs motion between these interventions.

We distinguish two references. Online retrieval selects $s_{u^\star(t)} \in \mathcal{M}$ from the observed prefix $x_{0:t}$. Evaluation uses a fixed real scene z assigned to the paired case, shared by all compared methods. Denote five-frame speed-distribution error by $h_t(x, z)$ and frame-wise spacing error by $s_t(x, z)$. The evaluation targets are

$$\bar{H}(x, z) = \frac{1}{T-4} \sum_{t=4}^{T-1} h_t(x, z), \qquad \bar{S}(x, z) = \frac{1}{T} \sum_{t=0}^{T-1} s_t(x, z). \tag{1}$$

The aim is to reduce these rollout errors relative to the uncorrected simulator, while inspecting the accompanying changes in motion smoothness and interactions. They are calibration targets, not a complete definition of realism.

Keeping these roles separate is important: if each method were scored against the scene it retrieved for itself, a change in reference could appear as an improvement even when the simulated rollout had not moved closer to the paired real case. The fixed evaluation target therefore makes the reported method differences comparable within each case, while the changing library reference remains part of the online control mechanism.

The component study tests three mechanisms: whether updating the real reference helps relative to freezing it; whether LLM-selected corrections offer a different fidelity–smoothness balance from a deterministic rule; and whether discrepancy-based timing improves on periodic intervention at a matched budget. Removing all online correction provides the native-controller baseline. These controls test the mechanism within the evaluated setting; they do not assume that every component must improve every metric. Figure 1 shows the resulting loop.

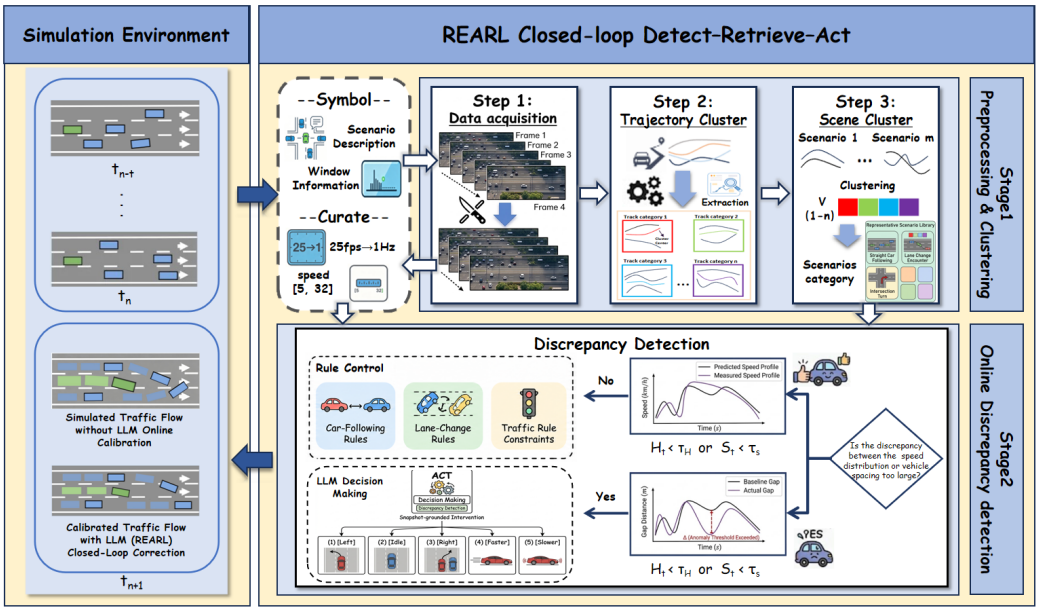


Figure 1: Overview of REARL. The simulator remains the execution environment. Offline, highD tracks are curated and clustered into a representative scene library. Online, discrepancy detection compares the rollout with the retrieved reference: rule-based control continues when speed and spacing stay within threshold, and a snapshot-guided discrete action is applied when they do not. Operational details are in Sections 3.3–3.4.

3.2 A FIXED LIBRARY OF REAL SCENES

We construct the library from a recording-disjoint highD reference split (Krajewski et al., 2018). Trajectories are sampled at 1 Hz, scaled to the simulator's speed range, and organized into 60-second scenes with ten retained vehicles per frame. Approximate Fréchet distance and klcluster assign three trajectory types (Buchin et al., 2019; Driemel et al., 2010); the proportions of these types represent each scene. Applying k-means yields 15 groups. For each group, we store the recorded scene nearest its centroid, not a synthetic centroid trajectory. The library is fixed throughout simulation. Online retrieval changes the selected scene without rerunning clustering. Filtering, split counts, and cluster diagnostics appear in Appendix A.

3.3 PREFIX-BASED RETRIEVAL AND DISCREPANCY DETECTION

For a simulated frame, let $d^x_{t,i}$ be vehicle i's mean distance to the other $n-1$ vehicles, and let $\bar{d}^x_t = n^{-1}\sum_i d^x_{t,i}$. For a real scene y, the implemented spacing error is

$$s_t(x,y) = \frac{n^{-1}\sum_{j=1}^{n} |d^y_{t,j} - \bar{d}^x_t|}{n^{-1}\sum_{j=1}^{n} d^y_{t,j}}. \tag{2}$$

This is a normalized absolute discrepancy against per-vehicle mean pairwise spacing. It is not nearest-leader headway, and its real-data denominator does not make it pointwise mean absolute percentage error (MAPE). We use "spacing error" throughout; the implementation stores this quantity under the key `spacing_mape`.

Let p_t and q_t be normalized eight-bin speed histograms over frames $t-4,\dots,t$ for simulated and real traffic. Their Hellinger distance is

$$h_t(x,y) = \frac{1}{\sqrt{2}}\left[\sum_b (\sqrt{p_{t,b}} - \sqrt{q_{t,b}})^2\right]^{1/2}. \tag{3}$$

At a checkpoint after at least five observations, the detector computes prefix averages for each candidate u:

$$\widehat{S}_t(u) = \frac{1}{t+1}\sum_{\tau=0}^{t} s_\tau(x, s_u), \quad \widehat{H}_t(u) = \frac{1}{t-3}\sum_{\tau=4}^{t} h_\tau(x, s_u), \quad u^\star(t) = \arg\min_u \widehat{S}_t(u). \tag{4}$$

Only the observed prefix is used. The selected reference may therefore change as the rollout evolves. Both detector statistics average the available history; $\widehat{H}_t$ is not merely the latest five-frame window.

Detection is performed after every five sampled frames, up to frame 55. If $\widehat{S}_t(u^\star) > \tau_S$, the next policy step receives a spacing-conditioned intervention; otherwise, $\widehat{H}_t(u^\star) > \tau_H$ triggers a speed-conditioned intervention. If neither test is positive, native control continues. Spacing has priority when both thresholds are exceeded. Thresholds are calibrated on a separate recording split; Appendix B gives their values and frame-index convention. Retrieval and detection use the library reference, whereas Eq. 1 is evaluated against the fixed case reference z.

3.4 Snapshot-guided discrete correction

A triggered query provides a time-aligned snapshot from the selected real scene together with the current vehicle state, road structure, and neighboring vehicles. The interface follows the structured action-selection approach of DiLu (Wen et al., 2023). It returns one identifier from

$$\mathcal{A} = \{0 : \textit{lane left},\ 1 : \textit{idle},\ 2 : \textit{lane right},\ 3 : \textit{faster},\ 4 : \textit{slower}\}.$$

The module is called separately for each of the ten controlled vehicles; the real scene is shared, while local observations are vehicle-specific. The simulator implements each accepted action for one policy step and then resumes IDM+MOBIL until the next intervention. Thus, the LLM selects corrections rather than continuous low-level controls.

Model interfaces differ in snapshot presentation and output parsing. Qwen and DeepSeek receive explicit simulated-versus-real comparisons; Qwen's selected prompt emphasizes conservative corrections and allows idle. Unparseable or unavailable actions can be re-asked or mapped to an available action, with idle preferred. This is an action-validity check, not a formal safety guarantee. The audited Qwen component required no fallback or remapping. Exact application programming interface (API) identifiers, prompt variants, and parser behavior are documented in Appendix B.1; comparisons across interfaces are not interpreted as a controlled model ranking.

4 Experiments

4.1 Protocol, metrics, and comparisons

We organize the evaluation around three questions: does online calibration improve aggregate alignment (research question 1, RQ1), which components account for the result (research question 2, RQ2), and does the difference persist later in a rollout (research question 3, RQ3)? highD recordings 01–45 supply 417 reference scenes, recordings 46–53 supply 110 calibration scenes, and recordings 54–60 supply an 85-scene evaluation pool. No evaluation recording enters clustering or threshold calibration. The paired study uses five scenes, one from each of recordings 54–58, crossed with five simulation seeds, giving 25 cases. These scenes subsequently informed prompt selection, so the reported results are development results despite the recording-disjoint preprocessing.

Each case has four lanes, ten controlled background vehicles, and 60 policy steps. Primary metrics $\bar{H}$ and $\bar{S}$ use the same fixed real case reference across methods. We additionally examine same-lane-gap Hellinger distance, jerk Wasserstein-1 distance (W_1), time headway (THW), lane-change rate (LCR), and collision vehicle rate. Lower distributional distances indicate closer agreement with that reference; THW and LCR have no universal best direction. Case metrics are averaged equally. For scene-level summaries, simulation seeds are averaged first, leaving five scene units. Metric definitions are in Appendix G.

Paired controls. Base is the unmodified IDM+MOBIL controller. BC-GRU is a recurrent behavior-cloning policy (Codevilla et al., 2019; Cho et al., 2014), trained on the reference split.

Table 1: Main comparison. Columns report speed Hellinger distance (H), spacing error (S), time headway (THW), lane-change rate (LCR), same-lane gap Hellinger distance (Gap), jerk Wasserstein distance (Jerk W_1), and collision vehicle rate (Coll.). Lower is better for H, S, Gap, Jerk, and Coll.; best and second-best values in those columns are bold and underlined. THW and LCR are descriptive and are not ranked.

Method	Speed Hellinger H	Spacing error S	Time headway (THW, s)	Lane-change rate (LCR)	Same-lane gap Hellinger distance	Jerk Wasserstein distance W_1	Collision vehicle rate
Base	0.3313	0.9190	2.6217	0.0081	0.495	**0.138**	0.220
BC-GRU	0.3332	1.6040	14.1134	0.0277	0.588	0.263	0.540
HAD-Gen	0.3445	1.4829	18.8562	0.0316	0.627	0.272	0.580
PPO	0.3557	1.1858	8.1690	0.0133	0.610	0.609	0.291
CRITICAL	0.3366	1.9132	9.7423	0.0209	0.612	0.522	—
REARL–GLM (Ours)	<u>0.2967</u>	<u>0.9040</u>	5.1242	0.0206	**0.376**	0.347	**0.196**
REARL–DeepSeek (Ours)	0.3199	0.9047	3.7660	0.0125	0.440	<u>0.160</u>	0.240
REARL–Qwen (Ours)	**0.2945**	**0.8448**	3.5742	0.0127	<u>0.423</u>	0.189	<u>0.212</u>

HAD-Gen denotes a style-conditioned adapter inspired by Wang et al. (2025). The latter two average three model seeds within the same scene–simulation-seed cases. REARL uses the fixed model identifiers `qwen3.8-flash`, `glm-5.3-flash`, and `deepseek-v4-pro`; Qwen supplies the component study.

Method-level references. Proximal policy optimization (PPO) (Schulman et al., 2017) and initialization-only CRITICAL (Tian et al., 2024) use their respective evaluation environments, including double simulation advance and different vehicle counts. They contextualize the results but are not identical-case causal controls (Appendix B).

4.2 RQ1: Does online calibration improve alignment?

All three REARL interfaces lower the pooled primary errors relative to Base (Table 1). Qwen reduces $\bar{H}$ from 0.3313 to 0.2945 and $\bar{S}$ from 0.9190 to 0.8448, reductions of 11.1% and 8.1%. GLM has a similar speed reduction and a smaller spacing change; DeepSeek's reductions are smaller. BC-GRU and HAD-Gen do not improve these calibration targets in the paired setting.

The gains have behavioral costs. Qwen increases THW from 2.62 to 3.57 s and jerk W_1 from 0.138 to 0.189, while improving same-lane-gap Hellinger from 0.495 to 0.423. Its collision vehicle rate is 0.212, compared with 0.220 for Base; this small descriptive difference is not evidence of a safety guarantee. GLM has the closest gap distribution, whereas DeepSeek has the lowest jerk error among the REARL interfaces. There is no single configuration that dominates every diagnostic.

The speed reduction is not confined to one scene: all three interfaces reduce scene-mean $\bar{H}$ in all five scenes. Qwen reduces spacing in four scenes. Its paired differences from Base are -0.0368 for speed and -0.0742 for spacing. The GLM and DeepSeek spacing intervals include zero (Table 5), so a lower pooled spacing error is not the same as a scene-stable spacing gain. With only five scene units and development-driven prompt choices, these intervals describe the observed suite rather than independent generalization.

4.3 RQ2: Which components account for the result?

Table 2 holds the Qwen configuration and paired cases fixed while removing one component at a time. w/o Online Loop returns exactly to Base. This identifies the effect of the complete correction layer but does not attribute it to any single component.

Updating the reference helps speed alignment. Freezing the first selected scene raises $\bar{H}$ from 0.2945 to 0.3147 and $\bar{S}$ from 0.8448 to 0.8753. Dynamic selection improves speed in all five scenes; the mean scene-level difference is -0.02021 (95% scene-bootstrap interval $[-0.02465, -0.01577]$). Its spacing difference is -0.03056 (interval $[-0.07801, 0.02643]$), which includes zero. This supports an evolving reference for speed alignment but gives less decisive evidence for spacing. The control tests online selection from the existing library, not the necessity of clustering or the optimality of 15 representatives.

Table 2: Component ablations using the same seven metrics as Table 1: speed Hellinger distance (H), spacing error (S), time headway (THW), lane-change rate (LCR), same-lane gap Hellinger distance (Gap), jerk Wasserstein distance (Jerk W_1), and collision vehicle rate (Coll.). Lower is better for H, S, Gap, Jerk, and Coll.; best and second-best values in those columns are bold and underlined. THW and LCR are descriptive and are not ranked.

Variant	Speed Hellinger H	Spacing error S	Time headway (THW, s)	Lane-change rate (LCR)	Same-lane gap Hellinger distance	Jerk Wasserstein distance W_1	Collision vehicle rate
w/o Online Loop	0.3313	0.9190	2.62	0.0081	0.495	0.138	0.220
w/o Low Threshold	0.3298	0.8445	2.96	0.0110	<u>0.454</u>	<u>0.134</u>	<u>0.212</u>
w/o Adaptive Timing	0.3240	<u>0.7795</u>	3.30	0.0105	<u>0.454</u>	**0.121**	**0.144**
w/o Retrieval	0.3147	0.8753	3.28	0.0125	0.439	0.175	0.248
w/o LLM	<u>0.3033</u>	**0.6486**	7.39	0.0098	0.472	0.577	0.220
Full REARL	**0.2945**	0.8448	3.57	0.0127	**0.423**	0.189	<u>0.212</u>

Action generation changes the fidelity–smoothness balance. Replacing the LLM with a deterministic real-data correction rule lowers spacing error from 0.8448 to 0.6486, but increases jerk W_1 from 0.189 to 0.577 and THW from 3.57 to 7.39 s. Full REARL has lower mean speed error (0.2945 versus 0.3033) and a smaller smoothness cost. The comparison supports a trade-off between corrective policies; it does not show that an LLM is necessary for improving speed and spacing, and it does not isolate language reasoning from prompt design.

Adaptive timing is not established. Raising the discrepancy thresholds reduces applied interventions from 275 to 132 and action calls from 2,750 to 1,320; the detector still checks all scheduled times. To separate timing from budget, we redistribute each sparse run's realized intervention count evenly over the checkpoints and reuse its ordered intervention types. This retrospective comparator reduces mean $\bar{H}$ from 0.3298 to 0.3240 and $\bar{S}$ from 0.8445 to 0.7795, with periodic scheduling better in all five scenes on both metrics. The paired mean differences are -0.00583 for speed and -0.06494 for spacing; their scene-bootstrap intervals are $[-0.00751, -0.00403]$ and $[-0.09125, -0.04456]$. Its budget is known from the completed sparse run, so it is not an independently deployable scheduling policy. Moreover, every full-setting checkpoint triggers through the spacing branch: its realized schedule is periodic. These results do not demonstrate an advantage for discrepancy-adaptive timing or independently validate the speed-trigger branch. Appendix F specifies the controls and audit.

4.4 RQ3: How does alignment change during the rollout?

Rollout averages can conceal when differences arise. Figure 2 therefore compares Qwen with Base at every recorded frame or speed window, pairing the same scene and simulation seed before aggregation. Thin curves retain the five scene-level trajectories; the thick curve is their mean. The zero line marks equal error against the fixed evaluation reference.

The speed advantage persists into the last 20 sampled frames: the mean paired Hellinger difference there is -0.0648, and all five scene means favor REARL. Spacing is more heterogeneous. Its last-20-frame mean difference is -0.1976, with four scenes improving and one worsening. Thus, the aggregate gains include later rollout behavior rather than only an initial transient, but spacing correction is not uniformly beneficial across scenes. These paired curves show relative alignment under the evaluated 60-step protocol; they do not establish that absolute drift decreases monotonically or that the same behavior persists over longer horizons.

4.5 Threshold selection and computational cost

Fixed operating point. The paired evaluation uses a fixed P30-speed/P5-spacing operating point, $(\tau_H, \tau_S) = (0.3356686506, 0.1921835063)$, selected from a separate calibration split before the five paired scenes were evaluated. A complete threshold-sensitivity experiment evaluates all 16 combinations of P5, P10, P20, and P30; its full results and CDF are reported in Appendix C.

Thresholds also control the number of online corrections. The sparse control uses the 60th-percentile pair and applies 132 interventions over the 25 paired cases, whereas the full setting applies 275 interventions. Both perform the same eleven detector evaluations per rollout; the difference is whether

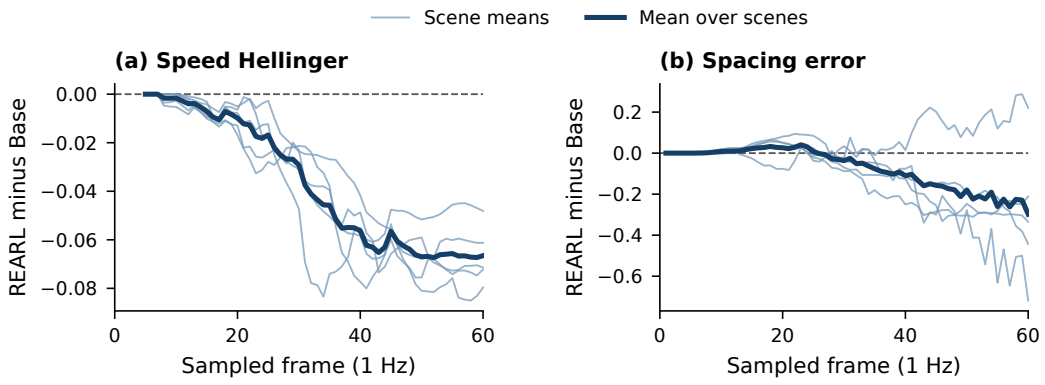


Figure 2: Temporal paired differences (REARL–Qwen minus Base). Negative values mean closer agreement with the fixed real reference. Each thin line averages the five paired seeds within one scene; the thick line averages the five scenes. All 25 cases are included, without smoothing. Speed points are placed at the last frame of each five-frame window; spacing is measured per frame. Scene curves show variation, not confidence intervals.

a checkpoint triggers an action. Thus, the threshold study is an intervention-budget analysis as well as a hyperparameter study, and it motivates the matched-budget periodic comparison in RQ2.

REARL is deliberately a simulation-calibration module rather than a low-latency vehicle controller. On the ten-step timing benchmark, IDM+MOBIL, PPO, and REARL require 0.461, 15.898, and 86,241.309 ms on average, respectively (Appendix E). The REARL measurement includes reference retrieval, discrepancy computation, and API reasoning, so it should not be compared with a single vehicle-level action latency. The cost is acceptable only in simulation-oriented uses where calibration fidelity is prioritized; reducing API calls and introducing a lightweight learned gate remain open engineering problems.

5 Discussion and Limitations

Three results delimit what the aggregate gains establish. First, updating the retrieved reference gives consistent speed improvements over freezing it, while the corresponding spacing interval includes zero; the evidence for this mechanism is therefore stronger for speed than for spacing. Second, a deterministic rule can reduce spacing error further, but its higher jerk and THW show that lower aggregate spacing error does not imply uniformly closer behavior. Third, the full configuration triggers at every scheduled checkpoint through spacing, and a retrospective periodic schedule outperforms the sparse trigger at matched budget. The experiments support reference-guided correction in this development suite, but they do not establish that LLM reasoning is necessary or that discrepancy-adaptive timing adds value.

The model comparison is likewise metric-specific. Qwen has the lowest primary speed and spacing errors among the tested interfaces, while GLM has the lowest gap-distribution error (0.376) and DeepSeek the lowest jerk among the REARL variants (0.160); Base remains smoother by jerk (0.138). A deterministic rule lowers spacing error further (0.6486) but raises jerk substantially (0.577). We therefore describe Qwen as the most balanced configuration for the two prespecified targets, not as the best on every diagnostic. Because prompts and interfaces differ, these results compare complete API configurations rather than isolating backbone effects.

The experiments change background traffic only. They do not measure a downstream planner, perception stack, or ego-vehicle policy. The study is limited to four-lane, 60-step highway rollouts and five development scenes. Their recording-disjoint preprocessing prevents one form of leakage, but their use in prompt selection precludes claims of independent validation. Urban traffic, longer horizons, and frozen-configuration holdouts remain untested. Library coverage and the value of clustering require random or unclustered-reference controls. Likewise, stronger non-LLM policies

and shuffled-snapshot controls are needed to isolate the contribution of language-based action selection. API interfaces and prompts differ, so cross-interface results indicate workable configurations rather than model superiority. None of the reported metrics establishes general traffic realism or deployment safety.

6 CONCLUSION

REARL treats traffic simulation as an evolving calibration problem rather than a one-time initialization problem. It selects a real reference from the current rollout prefix and applies short, simulator-native corrections at scheduled checkpoints, while IDM+MOBIL remains the execution model. In the paired development setting, the Qwen configuration has the lowest primary speed and spacing errors among the tested interfaces.

Updating the retrieved reference improves speed alignment relative to freezing the first reference. The evaluation target stays fixed, so this gain is not produced by rescoring a rollout against a moving scene. A deterministic real-data rule can reduce spacing error further, at a much larger jerk and time-headway cost. At the reported thresholds the spacing branch triggers at every checkpoint, and a retrospective periodic schedule has lower error than sparse triggering at a matched budget. The study therefore supports reference-guided correction in this development suite, without showing that discrepancy-adaptive timing or an LLM is required. Independent scenes, longer horizons, and stronger non-LLM controls remain open.

The paired curves locate when the difference appears. Over the last 20 sampled frames, the mean Qwen-minus-Base speed difference is -0.0648, and every scene mean favors Qwen. The corresponding spacing difference is -0.1976, with four scenes lower and one higher. The aggregate improvement therefore includes later frames of this 60-step protocol. Those curves do not show that absolute error decreases at every step, and they do not carry the claim to a longer horizon.

The threshold grid separates the two targets. On the 15 selection scenes, the lowest mean speed error is 0.3448 at P20/P20, and the lowest mean spacing error is 0.8867 at P30/P30. P30/P5, the point used in the paired study, is one non-dominated grid point rather than a joint optimum; its selection-scene averages are not the scores in Table 1. At that point the full setting fires at all 275 scheduled checkpoints, while the sparse control fires at 132, and the matched-budget periodic comparator then has lower error. The reported gains are reference-guided correction under a dense schedule.

The same comparisons bound how the result should be read. Qwen is lowest among the tested interfaces on the two prespecified errors, GLM is closer on same-lane gap (0.376), and DeepSeek is closer on jerk (0.160); Base remains smoothest by jerk (0.138). Replacing the LLM with the deterministic rule lowers spacing error to 0.6486 and raises jerk to 0.577. These are different points on a fidelity–smoothness trade-off, not a ranking of language models, because the prompts and output constraints differ. The ten-step timing benchmark places the same distinction on cost: IDM+MOBIL, PPO, and REARL take 0.461, 15.898, and $86{,}241$ ms on average once retrieval, detection, and API calls are included. That cost belongs to offline simulation calibration.

AI USE STATEMENT

LLMs are used as action-selection components of the method, as described in Section 3.4. Generative artificial intelligence (AI) also assisted research planning, code and LaTeX editing, organization and interpretation of recorded results, manuscript revision, and draft figure design. Quantitative results and the temporal figure derive from saved experiment outputs; image-generation tools were used for conceptual design drafts, not experimental evidence. The authors are responsible for verifying the final manuscript, references, code, and figures.

ETHICS STATEMENT

The study uses recorded highD trajectories and a traffic simulator; it does not deploy a vehicle controller. Dataset use remains subject to highD's access and licensing terms. Limited distributional metrics and simulated collision statistics should not be interpreted as assurances about real-world driving safety.

REPRODUCIBILITY STATEMENT

Sections 3–4 specify the reference-selection rule, detector statistics, action interface, metrics, and paired evaluation. The appendices provide recording and seed splits, model identifiers and prompt variants, component-control construction, full diagnostic results, and temporal aggregation. An anonymized implementation and experiment records are intended for the supplementary material; hosted model identifiers do not guarantee that a provider will preserve identical model behavior over time.

REFERENCES

AIMSUN. Aimsun next, 2022. URL https://www.aimsun.com/aimsun-next. Accessed: 2025-09-04.

Masako Bando, Katsuya Hasebe, Akihiro Nakayama, Akihiro Shibata, and Yuki Sugiyama. Dynamical Model of Traffic Congestion and Numerical Simulation. *Physical Review E*, 51(2):1035, 1995.

Jan-Aike Bolte, Andreas Bar, Daniel Lipinski, and Tim Fingscheidt. Towards corner case detection for autonomous driving. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pp. 438–445. IEEE, 2019.

Kevin Buchin, Anne Driemel, Natasja van de L'Isle, and André Nusser. klcluster: Center-based clustering of trajectories. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp. 496–499, 2019.

Kehua Chen, Meixin Zhu, Lijun Sun, and Hai Yang. Combining Time Dependency and Behavioral Game: A Deep Markov Cognitive Hierarchy Model for Human-Like Discretionary Lane Changing Modeling. *Transportation Research Part B: Methodological*, 189:102980, 2024.

Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734. Association for Computational Linguistics, 2014. doi: 10.3115/v1/D14-1179. URL https://aclanthology.org/D14-1179/.

Felipe Codevilla, Eder Santana, Antonio M López, and Adrien Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9329–9338, 2019.

Wenhao Ding, Wenshuo Wang, and Ding Zhao. A new multi-vehicle trajectory generator to simulate vehicle-to-vehicle encounters. *arXiv preprint arXiv:1809.05680*, 2018.

Anne Driemel, Sariel Har-Peled, and Carola Wenk. Approximating the Fréchet Distance for Realistic Curves in Near Linear Time. In *Proceedings of the twenty-sixth annual symposium on Computational geometry*, pp. 365–374, 2010.

Jakob Erdmann. SUMO's Lane-Changing Model. In *Modeling Mobility with Open Data: 2nd SUMO Conference 2014 Berlin, Germany, May 15-16, 2014*, pp. 105–123. Springer, 2015.

Nikita Jaipuria, Xianling Zhang, Rohan Bhasin, Mayar Arafa, Punarjay Chakravarty, Shubham Shrivastava, Sagar Manglani, and Vidya N Murali. Deflating dataset bias using synthetic data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 772–773, 2020.

Kemou Jiang, Xuan Cai, Zhiyong Cui, Aoyong Li, Yilong Ren, Haiyang Yu, Hao Yang, Daocheng Fu, Licheng Wen, and Pinlong Cai. Koma: Knowledge-driven multi-agent framework for autonomous driving with large language models. *IEEE Transactions on Intelligent Vehicles*, 2024.

Arne Kesting, Martin Treiber, and Dirk Helbing. General Lane-Changing Model MOBIL for Car-Following Models. *Transportation Research Record*, 1999(1):86–94, 2007.

Robert Krajewski, Julian Bock, Laurent Kloeker, and Lutz Eckstein. The HighD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2118–2125. IEEE, 2018.

E. Leurent. An environment for autonomous driving decisionmaking, 2018. URL `https://github.com/eleurent/highway-env`. Accessed: 2025-09-08.

Yun Li, Kai Katsumata, Ehsan Javanmardi, and Manabu Tsukada. Large language models for human-like autonomous driving: A survey. In *2024 IEEE 27th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 439–446. IEEE, 2024.

Han Liu, Ye Tian, Jian Sun, and Di Wang. An exploration of data-driven microscopic simulation for traffic system and case study of freeway. *Transportmetrica B: Transport Dynamics*, 11(1): 301–324, 2023.

Lin Liu, Shuo Feng, Yiheng Feng, Xichan Zhu, and Henry X Liu. Learning-based stochastic driving model for autonomous vehicle testing. *Transportation Research Record*, 2676(1):54–64, 2022.

Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic Traffic Simulation Using SUMO. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2575–2582. IEEE, 2018.

Aman Madaan, Shuyan Zhou, Uri Alon, Yiming Yang, and Graham Neubig. Language models of code are few-shot commonsense learners. *arXiv preprint arXiv:2210.07128*, 2022.

Zhaobin Mo, Rongye Shi, and Xuan Di. A physics-informed deep learning paradigm for car-following models. *Transportation Research Part C: Emerging Technologies*, 130:103240, 2021.

PTV Planung Transport Verkehr AG. *VISSIM 5.40 User Manual*. epubli GmbH, 2012.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Qingyuan Shi, Qingwen Meng, Hao Cheng, Qing Xu, and Jianqiang Wang. Linguasim: Interactive multi-vehicle testing scenario generation via natural language instruction based on large language models. *arXiv preprint arXiv:2510.08046*, 2025.

Po-An Shih, Shao-Hua Wang, Yung-Che Li, Chia-Heng Tu, and Chih-Han Chang. A Multi-Agent LLM Framework for Design Space Exploration in Autonomous Driving Systems. *arXiv preprint arXiv:2512.08476*, 2025.

Hanlin Tian, Kethan Reddy, Yuxiang Feng, Mohammed Quddus, Yiannis Demiris, and Panagiotis Angeloudis. Enhancing autonomous vehicle training with language model integration and critical scenario generation. *arXiv preprint arXiv:2404.08570*, 2024.

Martin Treiber, Ansgar Hennecke, and Dirk Helbing. Congested Traffic States in Empirical Observations and Microscopic Simulations. *Physical Review E*, 62(2):1805, 2000.

Cheng Wang, Lingxin Kong, Massimiliano Tamborski, and Stefano V Albrecht. HAD-Gen: Human-like and diverse driving behavior modeling for controllable scenario generation. *Accident Analysis & Prevention*, 223:108270, 2025. doi: 10.1016/j.aap.2025.108270.

Wenshuo Wang and Ding Zhao. Extracting traffic primitives directly from naturalistically logged data for self-driving applications. *IEEE Robotics and Automation Letters*, 3(2):1223–1229, 2018.

Xiao Wang, Rui Jiang, Li Li, Yilun Lin, Xinhu Zheng, and Fei-Yue Wang. Capturing car-following behaviors by deep learning. *IEEE Transactions on Intelligent Transportation Systems*, 19(3): 910–920, 2017.

Licheng Wen, Daocheng Fu, Xin Li, Xinyu Cai, Tao Ma, Pinlong Cai, Min Dou, Botian Shi, Liang He, and Yu Qiao. DiLu: A knowledge-driven approach to autonomous driving with large language models. *arXiv preprint arXiv:2309.16292*, 2023.

Tim A Wheeler and Mykel J Kochenderfer. Factor graph scene distributions for automotive safety analysis. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 1035–1040. IEEE, 2016.

Yaozu Wu, Dongyuan Li, Yankai Chen, Renhe Jiang, Henry Peng Zou, Wei-Chieh Huang, Yangning Li, Liancheng Fang, Zhen Wang, and Philip S Yu. Multi-agent autonomous driving systems with large language models: A survey of recent advances. *arXiv preprint arXiv:2502.16804*, 2025.

Dong-Fan Xie, Zhe-Zhe Fang, Bin Jia, and Zhengbing He. A data-driven lane-changing model based on deep learning. *Transportation Research Part C: Emerging Technologies*, 106:41–60, 2019.

Xintao Yan, Zhengxia Zou, Shuo Feng, Haojie Zhu, Haowei Sun, and Henry X Liu. Learning naturalistic driving environment with statistical realism. *Nature Communications*, 14(1):2037, 2023.

Zhenjie Yang, Xiaosong Jia, Hongyang Li, and Junchi Yan. Llm4drive: A survey of large language models for autonomous driving. *arXiv preprint arXiv:2311.01043*, 2023.

Kunpeng Zhang, Cheng Chang, Wenqin Zhong, Shen Li, Zhiheng Li, and Li Li. A systematic solution of human driving behavior modeling and simulation for automated vehicle studies. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):21944–21958, 2022.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2), 2023.

Jingxing Zhou and Jürgen Beyerer. Corner cases in data-driven automated driving: Definitions, properties and solutions. In *2023 IEEE Intelligent Vehicles Symposium (IV)*, pp. 1–8. IEEE, 2023.

Meixin Zhu, Xuesong Wang, and Yinhai Wang. Human-like autonomous car-following model with deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 97:348–368, 2018.

A REFERENCE-LIBRARY CONSTRUCTION

The reference split consists of highD recordings 01–45 (417 scenes), the calibration split of recordings 46–53 (110 scenes), and the evaluation pool of recordings 54–60 (85 scenes). Reference normalization and clustering use only the reference recordings. Source trajectories are sampled from 25 fps to 1 Hz, with speeds mapped to $[5, 32]$ for the simulator. Each scene spans 60 sampled seconds and must retain at least ten visible vehicles at every frame. The ten longest-observed vehicles are retained per snapshot. Source trajectories shorter than 100 frames are removed.

A trajectory is represented by lateral position, longitudinal position, and absolute longitudinal speed. klcluster with approximate Fréchet distance assigns three trajectory groups (Buchin et al., 2019; Driemel et al., 2010). A scene's group proportions give $v_i \in \mathbb{R}^3$. The scene-level clustering minimizes

$$\sum_{j=1}^{15} \sum_{v_i \in C_j} \|v_i - \mu_j\|_2^2, \tag{5}$$

and the recorded scene nearest each μ_j is stored as the representative. No LLM call updates these clusters. The final trajectory-group counts are 59,953, 3,634, and 2,854; scene-group sizes are $125, 6, 65, 8, 5, 19, 3, 9, 53, 10, 15, 34, 55, 5, 5$. Silhouette, Davies–Bouldin, and Calinski–Harabasz scores are 0.511, 0.617, and 3515, respectively. These summarize the fitted clustering, not its downstream causal value.

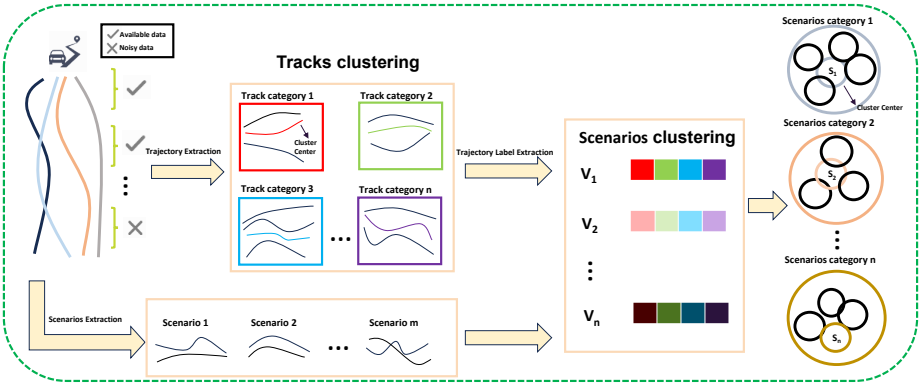


Figure 3: Offline reference construction: trajectory types yield scene-composition vectors, and each scene category contributes its nearest recorded representative.

B PROTOCOL AND MODEL INTERFACES

The paired scenes come from recordings 54–58 and were selected with seed 20260905. Each uses simulation seeds $5838, 2421, 7294, 9650, 4176$. Subsequent prompt development used these scenes, which is why we describe them as a development set. Each rollout has four lanes, ten controlled vehicles, density 2.0, 60 policy steps, 15 Hz simulation, and 1 Hz sampling and policy updates. BC-GRU and HAD-Gen average three model seeds for each paired case.

The detector is evaluated after five, ten, . . . , 55 observed frames. In the implementation's zero-based snapshot indexing, these are indices $4, 9, \ldots, 54$; a triggered action is applied on the next iteration, at indices $5, 10, \ldots, 55$. A complete rollout therefore has eleven detector evaluations. Full thresholds are $\tau_H = 0.3356686506$ and $\tau_S = 0.1921835063$ (calibration P30 and P5). Sparse thresholds are 0.3538320728 and 0.6981910910 (P60). Both schedules use the same five-frame checkpoint grid. Changing thresholds changes the number of interventions, not the number of detector evaluations.

PPO uses an 11-vehicle environment and double simulation advance. CRITICAL also uses double simulation advance, variable vehicle counts, and 38 reference-based rollouts per case. These 38 references are drawn from reference data. Merging their records reuses vehicle IDs, so CRITICAL's raw and unique-vehicle collision counts do not share the denominator of the ten-vehicle controls. We omit that collision rate and report PPO and CRITICAL separately from paired comparisons.

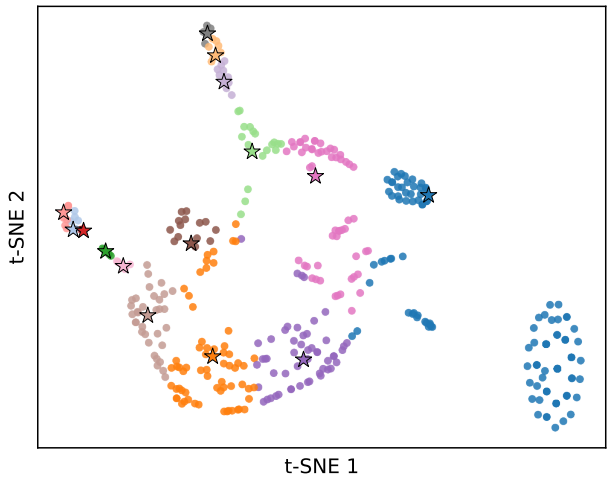


Figure 4: t-distributed stochastic neighbor embedding (t-SNE) display of the 15 scene categories. Stars mark recorded representatives. The coordinates are used only for visualization, not clustering or online retrieval.

B.1 Snapshot presentation and action parsing

The API request model IDs in the reported results are `qwen3.8-flash`, `glm-5.3-flash`, and `deepseek-v4-pro`. These identifiers are not parameter-count claims. All reported configurations use temperature zero.

Qwen uses the `dual_coached_stable` prompt configuration: both speed and average-distance snapshot fields, an explicit simulated-versus-real comparison, and a conservative action instruction. It prefers idle when the mismatch is small or uncertain, recommends modest speed changes when warranted, and retains lane changes in the available action set. GLM uses the baseline snapshot interface, whose presented fields depend on the active discrepancy branch. DeepSeek uses `pro_nothink_coached_dual`, with both snapshot fields, comparison guidance, disabled extended thinking, a 64-token output budget, strict terminal parsing, and at most two re-asks for invalid actions. No longitudinal lane-change mask is imposed by these reported configurations.

The general execution path checks parsing and action availability. An invalid action can be re-asked; unresolved outputs are mapped to an available action, with idle preferred. This differs from claiming that every rejected output immediately invokes native IDM+MOBIL for the same step. Native control resumes after the action step. The formal Qwen component audit records directly parsed, available outputs with no fallback or rule remapping. Prompt wording and parser differences prevent interpreting cross-interface comparisons as an isolated effect of the underlying language model.

C Threshold estimation and combination study

The threshold experiment measures sensitivity to the two discrepancy targets used by the detector: speed-distribution Hellinger distance and normalized mean-spacing error. Candidate speed and spacing thresholds are the P5, P10, P20, and P30 empirical percentiles, crossed into a 4×4 grid of 16 combinations. Calibration uses 95 scenes and provides 26,600 speed differences and 28,500 spacing differences; the 16 combinations are evaluated on 15 non-overlapping selection scenes, for 240 complete rollouts. The paired test scenes are separate from both calibration and threshold-selection scenes.

The lowest mean speed Hellinger distance is 0.3448 at P20/P20, whereas the lowest mean spacing error is 0.8867 at P30/P30. The two minima therefore come from different threshold combinations, and no single grid point dominates both objectives. On these 15 selection scenes, the same P30/P5

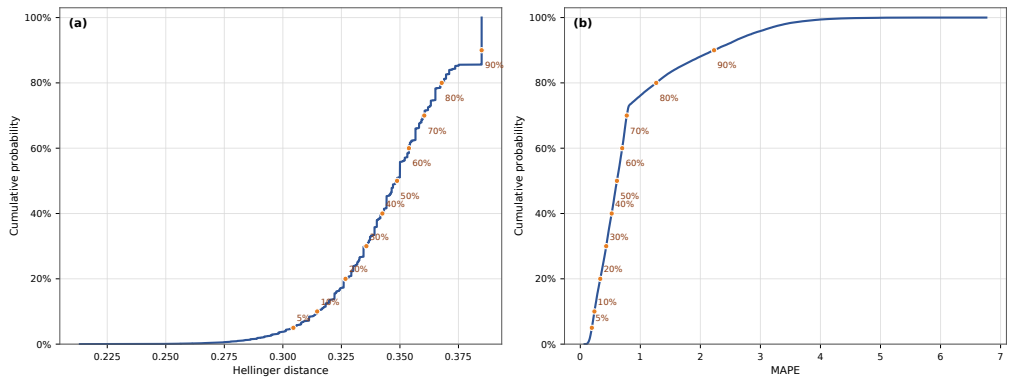


Figure 5: Empirical cumulative distribution functions (CDFs) of the calibration discrepancies. Panel (a) shows speed-distribution Hellinger distance and panel (b) shows normalized mean-spacing error. Markers denote calibration percentiles P5–P90; only P5, P10, P20, and P30 enter the 16-combination comparison.

Table 3: Threshold-combination results on 15 selection scenes per combination. Columns report the speed Hellinger threshold, spacing threshold, mean speed-distribution Hellinger distance, and mean spacing error. Rows are ordered by spacing percentile and then speed percentile.

Speed Hellinger threshold	Spacing threshold	Mean speed Hellinger	Mean spacing error
0.3044933281 (P5)	0.1921835063 (P5)	0.3507	0.9346
0.3146632582 (P10)	0.1921835063 (P5)	0.3511	0.8905
0.3267608026 (P20)	0.1921835063 (P5)	0.3505	0.9039
0.3356686506 (P30)	0.1921835063 (P5)	0.3470	0.9332
0.3044933281 (P5)	0.2352122969 (P10)	0.3500	0.9373
0.3146632582 (P10)	0.2352122969 (P10)	0.3500	0.9024
0.3267608026 (P20)	0.2352122969 (P10)	0.3476	0.9334
0.3356686506 (P30)	0.2352122969 (P10)	0.3510	0.9043
0.3044933281 (P5)	0.3327673037 (P20)	0.3493	0.9545
0.3146632582 (P10)	0.3327673037 (P20)	0.3491	0.9065
0.3267608026 (P20)	0.3327673037 (P20)	0.3448	0.9628
0.3356686506 (P30)	0.3327673037 (P20)	0.3502	0.9507
0.3044933281 (P5)	0.4316209103 (P30)	0.3503	0.9185
0.3146632582 (P10)	0.4316209103 (P30)	0.3493	0.9398
0.3267608026 (P20)	0.4316209103 (P30)	0.3482	0.9307
0.3356686506 (P30)	0.4316209103 (P30)	0.3516	0.8867

thresholds obtain 0.3470 speed Hellinger distance and 0.9332 spacing error and lie on the observed Pareto frontier. These selection-scene averages are not the paired-study scores in Table 1. We use this operating point for the paired study as a reproducible choice on that frontier, rather than as the unique optimum of the grid. Overall, the sweep shows a trade-off among speed alignment, spacing alignment, and intervention frequency.

D CLUSTERING PARAMETER STUDY

The trajectory-clustering configuration uses three klcluster trajectory groups and three iterations, selected by a 500-iteration Bayesian search on a 5% trajectory sample. The evaluation uses these settings without rerunning that search. We then fit the 15 scene-level groups on the current reference split, selecting the nearest recorded scene to each fitted centroid as a representative. Appendix A reports this current clustering's group sizes and validation scores and shows its t-SNE view. The embedding is descriptive only; neither its coordinates nor visual separation determine the online retrieval rule.

E RUNTIME COMPARISON

Table 4: Average inference time over ten decision steps in the timing benchmark.

Method	Average time (ms)
IDM+MOBIL	0.461
PPO	15.898
REARL	86,241.309

The REARL timing includes discrepancy detection, reference processing, and LLM API reasoning. It is therefore a system-level simulation-calibration measurement rather than a per-vehicle policy inference latency. The benchmark documents the computational trade-off of the evaluated system.

F COMPONENT-CONTROL CONSTRUCTION AND AUDIT

w/o Online Loop removes all online interventions and reproduces Base. w/o Retrieval retains the first retrieved representative for the rest of the rollout. To keep the full-setting intervention schedule and evidence branch fixed, this control intervenes at every checkpoint using the spacing branch. w/o LLM uses the same schedule and reference-selection mechanism but replaces LLM generation with a deterministic policy. For a spacing intervention, it compares simulated and reference mean spacing and prioritizes slower when the simulated spacing is smaller, and faster when it is larger, subject to available actions. For a speed intervention, the direction is chosen to reduce the mean speed discrepancy. The full setting's 275 applied checkpoints use the spacing branch.

w/o Low Threshold raises both thresholds. For each completed w/o Low Threshold case, w/o Adaptive Timing reads its applied-intervention count and ordered intervention-type sequence. If there are $k > 1$ interventions, it chooses checkpoint indices by rounding $i(11-1)/(k-1)$ for $i = 0, \ldots, k-1$; $k = 1$ uses the middle checkpoint and $k = 0$ uses none. It forces intervention at those points and reuses the sparse run's type sequence, while recomputing the selected reference and actions on its own evolving trajectory. Count and type are matched per case; timing and resulting traffic differ. Because the construction uses the completed sparse run, it is a retrospective timing control, not a policy with a prospectively known budget.

The formal audit verifies 25 cases per variant, locked protocol settings, complete case keys, recomputed aggregate metrics, identical w/o Low Threshold and w/o Adaptive Timing counts and type sequences, evenly distributed periodic checkpoints, and a constant reference in w/o Retrieval. Across these three variants, all $1{,}320$, $1{,}320$, and $2{,}750$ action outputs, respectively, are parsed directly with no fallback, remapping, or unavailable actions.

Dynamic retrieval improves scene-mean speed error over fixed retrieval in all five scenes: mean difference -0.02021, scene-bootstrap 95% interval $[-0.02465, -0.01577]$. Its spacing difference is -0.03056, with interval $[-0.07801, 0.02643]$. Periodic minus sparse gives speed difference -0.00583 and spacing difference -0.06494, with intervals $[-0.00751, -0.00403]$ and $[-0.09125, -0.04456]$; both favor the periodic comparator in all five scenes. These intervals condition on the development suite and the retrospective control construction.

G METRIC DEFINITIONS AND PAIRED COMPARISONS

Speed histograms use edges $[5, 8.375, 11.75, 15.125, 18.5, 21.875, 25.25, 28.625, 32]$. A 60-frame rollout yields 56 overlapping five-frame windows and 60 spacing observations. Eq. 2 defines the quantity stored as `spacing_mape`; reported values are fractions, not percentages. All primary evaluation metrics use each case's fixed real scene rather than the dynamically retrieved library scene.

THW is the net gap to the nearest same-lane leader divided by follower speed, averaged over valid observations. LCR counts lane-identifier (ID) changes between adjacent policy steps, excludes changes coincident with a collision, and normalizes by frames and vehicles. Gap Hellinger compares the pooled nearest-leader net-gap distributions. Jerk W_1 compares distributions of changes in

acceleration across one-second samples. Collision vehicle rate is the fraction of unique vehicles that collide at least once, averaged over rollouts. THW, LCR, and collision rates describe behavior and are not ranked as a single realism score.

G.1 Paired scene-level differences

Table 5 restricts scene-level paired comparisons to methods with shared simulation semantics. Each scene contributes one mean difference after averaging simulation seeds (and model seeds where applicable). Bootstrap intervals resample the five scene differences, not the 25 runs. They quantify variation within this development suite, not uncertainty over all highD recordings. PPO and CRITICAL are excluded from this paired table.

Table 5: Paired differences from Base. Brackets give scene-bootstrap 95% intervals. Wins count scenes with lower error; the resampling unit is a scene ($n = 5$).

Metric	Method	Difference [95% interval]	Wins
Speed $\bar{H}$	BC-GRU	$+0.0018$ $[-0.0032, 0.0060]$	1/5
	HAD-Gen	$+0.0132$ $[0.0076, 0.0175]$	0/5
	REARL–DeepSeek	-0.0114 $[-0.0139, -0.0089]$	5/5
	REARL–GLM	-0.0346 $[-0.0399, -0.0278]$	5/5
	REARL–Qwen	-0.0368 $[-0.0400, -0.0335]$	5/5
Spacing $\bar{S}$	BC-GRU	$+0.6850$ $[0.6663, 0.7036]$	0/5
	HAD-Gen	$+0.5639$ $[0.5476, 0.5802]$	0/5
	REARL–DeepSeek	-0.0143 $[-0.0364, 0.0215]$	4/5
	REARL–GLM	-0.0150 $[-0.1064, 0.0778]$	3/5
	REARL–Qwen	-0.0742 $[-0.1173, -0.0108]$	4/5

H Temporal aggregation

Figure 2 uses the saved per-time metrics for all 25 paired Base and full-Qwen cases. At each observation, we subtract the Base value from the Qwen value for the same scene, seed, and time index. We average five seed differences within each scene and then average the five scene means. The figure displays all scene means directly rather than an uncertainty band. No observations are removed, interpolated, or smoothed. There are 25×56 paired speed-window values and 25×60 paired spacing values. Zero-based speed-window indices are plotted at their one-based final frame $(5, \ldots, 60)$; spacing uses frames $1, \ldots, 60$. Late-rollout summaries average points ending in frames 41–60.