applied sciences

MDPI

Article

FAR-MindDrive: Failure-Aware Risk Attribution for Efficient Post-Training of Autonomous Driving Vision–Language–Action Models

Zhuoyang Jia [1] and Xinyu Deng [2,*]

[1] School of International Education (European Institute), Yanshan University, Qinhuangdao 066004, China; 2325466320@stumail.ysu.edu.cn
[2] State Key Laboratory of Information Engineering in Surveying Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China
* Correspondence: mrdengxy@163.com

Abstract

Vision–language–action (VLA) models provide a promising interface between semantic reasoning and executable behavior for end-to-end autonomous driving, yet sparse closed-loop feedback leaves the decisive time step and responsible component of a structured driving decision unspecified. This ambiguity weakens credit assignment and can lead to broad policy updates that disturb previously reliable behaviors. We propose FAR-MindDrive, a failure-aware risk attribution framework that converts closed-loop failures into localized corrective supervision before policy repair. FAR-MindDrive first performs event-centered temporal backtracking to identify the short decision history most relevant to a violation, then applies factorized counterfactual interventions to longitudinal and lateral meta-actions and measures their event-specific geometric risk reduction. High-confidence corrections update the Decision Expert, while imitation replay and reference-policy regularization constrain unnecessary drift. On Bench2Drive, FAR-MindDrive achieves a five-run mean Driving Score of 79.12 and Success Rate of 57.73%, compared with 75.85/49.30% for MindDrive-IL and 78.04/55.09% for MindDrive; the mean collision rate is 5.91%. These results show that structured failure attribution enables targeted, interpretable, and resource-efficient VLA post-training while retaining the deployed action-generation architecture.

Keywords: autonomous driving; vision–language–action model; closed-loop driving; failure attribution; counterfactual risk; policy repair

Check for updates

Academic Editor: Arkadiusz Gola

Received: 9 October 2026
Revised: 27 October 2026
Accepted: 29 October 2026
Published: 5 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Autonomous driving has progressively shifted from modular stacks toward end-to-end (E2E) systems that optimize driving behavior directly from sensor observations. E2E learning coordinates perception, prediction, and planning under a shared driving objective, reducing the mismatch introduced by independently optimized modules. UniAD organizes multiple driving tasks around a shared planning objective, whereas VAD represents agents and map elements as explicit vectorized constraints for E2E planning [1,2]. ThinkTwice strengthens the planning decoder by explicitly reasoning about the consequences of coarse actions [3]. Generative formulations model driving through structured future generation or multi-modal action generation, as exemplified by GenAD and DiffusionDrive [4,5]. Open-loop trajectory accuracy and closed-loop reliability remain distinct because the deployed policy changes its subsequent observations. Bench2Drive exposes this gap through

interactive scenarios spanning diverse driving abilities and safety-critical events [6]. The resulting challenge is to learn decisions that remain reliable under closed-loop interaction, beyond predicting plausible trajectories on logged data.

Large vision–language models provide another path toward this goal. Their pretrained visual knowledge, semantic abstraction, and language reasoning capabilities offer a richer interface for understanding traffic rules, navigation intent, long-tail events, and interactions among road users. LMDrive demonstrated that language instructions can be incorporated into closed-loop E2E driving [7], while DriveVLM introduced hierarchical scene description, analysis, and planning for complex driving situations [8]. DriveLM further organized perception, prediction, and planning questions into graph-structured visual reasoning [9]. More recent work has begun to tighten the connection between language understanding and executable behavior. SimLingo explicitly studies language–action alignment in closed-loop driving [10], and ORION introduces a generative planner that bridges semantic reasoning with continuous trajectory generation [11]. These developments mark a transition from vision–language-assisted driving toward vision–language–action (VLA) driving, where the language model is no longer only an auxiliary explainer but participates directly in decision making and action generation.

The training paradigm is changing accordingly. Most E2E and VLA policies are initialized by imitation learning because demonstrations provide stable and scalable supervision. Deployment, however, visits states outside the expert distribution, where demonstration supervision becomes insufficient [12,13]. Recent work therefore introduces reinforcement learning, preference optimization, or closed-loop refinement into autonomous driving. AlphaDrive applies reinforcement learning to planning-oriented VLM reasoning [14]. DriveDPO converts imitation quality and rule-based safety signals into trajectory preferences for direct policy optimization [15]. Raw2Drive and RAD exploit world models or large-scale simulated interaction to improve closed-loop policies [16,17], while R2SE focuses on hard-case refinement and retention of the general driving knowledge acquired by an imitation learner [18]. MindDrive separates a Decision Expert that produces linguistic meta-actions from an Action Expert that converts those decisions into feasible trajectories, then optimizes the Decision Expert with online reinforcement learning [19]. This language-level exploration narrows the action search space and provides a direct route for trajectory feedback to influence high-level reasoning.

A remaining difficulty is **failure credit attribution**. A closed-loop failure provides a route-level outcome while leaving the responsible earlier decision unspecified. MindDrive propagates sparse success and penalty rewards through temporal-difference learning, generalized advantage estimation, and PPO [19–21]. This scalar feedback entangles two forms of responsibility that are structurally distinct in driving. First, responsibility is distributed over time: a collision observed at the current step may originate from an overly aggressive decision made several steps earlier. Second, responsibility is distributed over action factors: the same collision may be caused primarily by an inappropriate longitudinal decision, an incorrect lateral maneuver, or their interaction. A scalar return ranks actions in expectation; targeted repair additionally requires identifying **when the decisive error emerged** and **which component of the structured meta-action should be revised**. Broadly updating all preceding states increases sample demand and can disturb reliable driving skills, a failure mode also highlighted by hard-case reinforcement refinement [18].

The structure of VLA driving provides an opportunity to resolve this ambiguity. Safety-critical events define temporal anchors, and language-level meta-actions expose semantically meaningful intervention dimensions. We therefore place **failure attribution before policy repair**. A short temporal neighborhood around a closed-loop event localizes likely decision errors, after which factorized counterfactual probes measure how much

event-specific risk changes when one decision component is varied and the other remains fixed. The frozen trajectory generator and privileged simulator geometry provide a local short-horizon risk-sensitivity estimate. This training-time diagnostic is related to the short-horizon physical evaluation used by NAVSIM [22]; final performance is evaluated separately under interactive closed-loop execution.

Based on this insight, we propose **FAR-MindDrive**, a **F**ailure-**A**ware **R**isk attribution framework for efficient post-training of autonomous-driving VLA models. FAR-MindDrive preserves the pretrained perception and action-generation capability of MindDrive and modifies the Decision Expert through lightweight LoRA adaptation. First, **Event-Centered Temporal Credit Backtracking** converts each closed-loop violation into a short decision window and assigns temporally decaying responsibility to states preceding the event. Second, **Factorized Counterfactual Risk Attribution** intervenes on longitudinal and lateral meta-actions separately. With the other action factor fixed, the frozen Action Expert generates local candidate trajectories, and event-specific geometric costs measure how much each intervention reduces the observed risk. The relative risk reduction yields an interpretable attribution between speed control and path selection using simulator geometry. Third, **Confidence-Gated Conservative Policy Repair** retains corrections supported by both temporal relevance and sufficient risk reduction. These samples supervise the Decision Expert, while imitation replay and reference-policy regularization constrain policy drift. The resulting pipeline transforms a sparse route-level failure into a small set of localized, factorized, and confidence-weighted corrective examples.

FAR-MindDrive targets the conversion of closed-loop failures into VLA training supervision. The vision encoder, base language model, Action Expert, and trajectory decoder remain frozen during repair, so the deployed model retains the original inference structure. The evaluation protocol uses Bench2Drive [6] to assess Driving Score, Success Rate, safety-critical violations, and multi-ability performance, together with diagnostic analyses of attribution quality and policy retention.

Research hypotheses and validation. We test three linked hypotheses.

H1. *Weighting decisions by temporal proximity improves repair compared with uniform weighting over the same window (Methods (2)–(3), Section 4.3.1); the horizon study separately examines window length.*

H2. *Factorized counterfactual supervision improves the imitation-trained policy (Methods (1)–(2)), with the qualitative and attribution analyses in Sections 4.2 and 4.3.2 illustrating local risk changes.*

H3. *Confidence gating, imitation replay, and KL regularization improve the repair/retention trade-off (Methods (3)–(5) and Section 4.3.4).*

The main contributions are threefold:

1. We formulate closed-loop VLA post-training as a **structured failure-attribution problem** and introduce event-centered temporal backtracking to localize route-level failures to the short decision history most relevant to the observed event.
2. We propose **factorized counterfactual risk attribution**, which independently probes longitudinal and lateral language-level decisions and converts event-specific geometric risk reduction into interpretable corrective supervision.
3. We develop a **confidence-gated conservative repair strategy** that updates only the Decision Expert with high-confidence corrections, while replay and reference-policy regularization preserve previously acquired driving behavior and keep the inference architecture unchanged.

2. Related Work

2.1. End-to-End Autonomous Driving and Closed-Loop Planning

E2E autonomous driving has evolved from direct imitation of controls toward planning-centered models with increasingly explicit scene structure. Earlier closed-loop systems showed that compact intermediate representations can substantially improve policy learning. NEAT uses continuous attention fields to associate visual observations with BEV semantics and waypoints [23], while Learning to Drive from a World on Rails separates non-reactive environment evolution from ego dynamics to learn action values from logged driving data [24]. Roach takes a different route by training a privileged reinforcement-learning expert and distilling its supervision into a vision-based driving policy [25]. These approaches established two themes that remain important today: driving policies benefit from structured intermediate supervision, and privileged information can be useful during training even when it is unavailable at inference.

The next generation of E2E systems focused on stronger interaction modeling and more reliable closed-loop behavior. Learning From All Vehicles increases the diversity of training supervision by exploiting the observed behavior of surrounding agents [26]. TCP jointly predicts trajectories and low-level controls so that future motion can guide immediate action [27], while InterFuser fuses multi-modal sensor information into interpretable safety-related representations before control generation [28]. DriveAdapter further separates perception and planning knowledge through feature adapters and incorporates action-guided masking to handle imperfect teacher signals [29]. In parallel, planning-oriented unified frameworks demonstrated that task interactions themselves can be learned. UniAD organizes detection, tracking, mapping, motion prediction, occupancy, and planning around a common planning objective [1]. This line of work improves representation sharing and task coordination, but most models are still predominantly trained to imitate expert trajectories. Their training targets describe what the expert did more directly than why a particular decision failed after closed-loop interaction.

Recent work addresses this limitation from the planning representation and evaluation sides. DriveTransformer replaces serial perception–prediction–planning execution with parallel task interactions, sparse representations, and streaming temporal processing [30]. DiffusionDrive models multi-modal driving actions through a truncated diffusion policy that reduces the iterative cost of conventional diffusion planning [5], while MomAD injects trajectory and perception momentum to improve temporal consistency and reduce unstable control [31]. World-model-based approaches provide additional supervision about future consequences. LAW predicts future latent scene representations conditioned on ego trajectories and uses this self-supervised objective to strengthen E2E scene features [32]. WoTE explicitly predicts future BEV states to evaluate candidate trajectories online [33]. These methods show that trajectory quality can be improved by modeling future consequences rather than relying on pointwise regression alone.

Closed-loop benchmarks have reinforced the same conclusion. Bench2Drive evaluates policies across 44 interactive scenario types and exposes capability differences that are difficult to observe from open-loop displacement errors [6]. Non-reactive short-horizon evaluation provides a scalable approximation to physical consequences [22], whereas interactive evaluation also captures how other agents respond to a changed ego policy. FAR-MindDrive uses geometric rollout as a **training-time diagnostic signal** for local failure attribution and evaluates the repaired policy under closed-loop interaction. The remaining gap is an explicit mechanism that converts an observed failure into temporally and semantically localized correction targets for a structured VLA decision policy.

2.2. Vision–Language and Vision–Language–Action Driving

The introduction of language models changes the representation of driving decisions. Early VLM-oriented approaches mainly used language to enrich scene understanding or expose intermediate reasoning. VLP couples visual perception with language-based planning to improve contextual reasoning and long-tail generalization [34]. OmniDrive constructs counterfactual vision–language supervision for 3D perception, reasoning, decision making, and planning and evaluates complementary agent designs that emphasize vision–language alignment and 3D-aware representation learning [35]. Senna explicitly decouples high-level language planning from low-level E2E trajectory prediction, allowing Senna-VLM to produce semantic decisions while Senna-E2E preserves numerical trajectory precision [36]. EMMA moves further toward a unified multi-modal formulation by representing navigation, trajectories, object locations, and road structures in a common textual interface [37]. These methods demonstrate the value of world knowledge and semantic abstraction while exposing the persistent interface between language reasoning and continuous vehicle motion.

A second technical route therefore focuses on coupling language and action more tightly. LMDrive introduces language-guided closed-loop driving and shows that natural-language instructions can be integrated into an E2E policy [7]. SimLingo jointly learns driving, visual–language understanding, and language–action alignment, explicitly recognizing that correct verbal answers are insufficient if they are inconsistent with the executed behavior [10]. ORION bridges the VLM reasoning space and continuous trajectory space with a generative planner, so the semantic planning token directly conditions multi-modal trajectory generation [11]. SOLVE adopts a collaborative VLM-E2E design, including trajectory chain-of-thought reasoning and asynchronous cooperation between high-level language reasoning and real-time control [38]. DiMA distills multi-modal language-model knowledge into a deployed vision-based planner [39]. Together, these methods progress from language as explanation, to language as planning context, and finally to language-conditioned action generation.

The scope of driving-oriented multi-modal models is also expanding. RoboTron-Drive trains an all-in-one multi-modal model across perception, prediction, and planning tasks and emphasizes cross-dataset generalization [40]. VLR-Driver introduces spatiotemporal reasoning to make decision formation more transparent in embodied closed-loop driving [41]. HERMES combines 3D scene understanding and future scene generation within a unified driving world model, linking semantic knowledge with predicted environment evolution [42]. More recently, DriveVLA-W0 treats future image prediction as dense self-supervision for driving VLA models and uses a lightweight action expert for real-time action generation [43]. Outside autonomous driving, OpenVLA provides an influential open-source example of how pretrained vision–language representations can be adapted to embodied action policies through parameter-efficient fine-tuning [44]. DriveGPT4 is an earlier example of an interpretable multi-modal model that predicts vehicle controls from video and textual instructions [45].

Most VLM/VLA driving systems focus on three problems: improving scene reasoning, connecting semantics to numerical action, and scaling training supervision. FAR-MindDrive addresses the subsequent stage in which a language-level decision has been mapped to a valid trajectory and fails during closed-loop execution. The operative question is **which language-level decision should be changed for this specific failure**. Existing language–action alignment objectives leave responsibility across time and factorized longitudinal/lateral decisions implicit. FAR-MindDrive uses the structured meta-action interface as an intervention space, turning interpretable language actions into actionable post-training supervision.

2.3. Reinforcement Learning, Preference Optimization, and Failure-Driven Policy Refinement

Imitation learning is efficient when expert demonstrations cover the states encountered by the deployed policy, but sequential decision making violates this assumption as prediction errors change future observations. Classical analyses of imitation learning identify both distribution shift and causal misidentification as fundamental sources of error [12,13]. Reinforcement learning provides a natural corrective mechanism because feedback can be collected under the learner's own state distribution. PPO remains a widely used policy-gradient method because clipped policy updates provide a practical compromise between optimization stability and implementation simplicity [20], while generalized advantage estimation reduces policy-gradient variance through temporally smoothed value estimates [21]. These methods are effective for broad policy optimization, but their advantages remain scalar summaries of future return. In safety-critical driving, a scalar advantage does not necessarily expose which structured decision factor caused a particular violation.

Recent autonomous-driving research adapts reinforcement learning and preference optimization to planning and VLA models. ReCogDrive combines a cognitive VLM with a diffusion planner and applies group-relative reinforcement optimization to improve physical planning [46]. VLM-RL uses pretrained vision–language models to construct semantic reward signals for safe driving and combines them with vehicle-state rewards [47]. DriveDPO transfers the preference-optimization perspective to E2E planning by deriving trajectory preferences from imitation and rule-based safety scores [15]. This direction is connected to DPO, which learns preferences directly against a reference policy [48]. These methods optimize entire trajectories, responses, or candidate actions. FAR-MindDrive adds an explicit decomposition of each realized failure into temporal and action-factor responsibility.

Online VLA reinforcement learning makes this credit problem more visible. MindDrive samples discrete language meta-actions, maps them to continuous trajectories through an Action Expert, and feeds simulator rewards back to the Decision Expert [19]. This design is substantially more efficient than exploring arbitrary continuous trajectories and provides a clean semantic action space. Its value function and advantage estimator propagate sparse terminal and infraction feedback, leaving responsibility implicit. For example, the same collision penalty may correspond to an excessively aggressive speed choice, an unsafe lane change, or a decision several steps before the event. Targeted repair therefore requires a structured label beyond the scalar return.

Failure-driven refinement has begun to address policy retention and hard-case specialization. R2SE, for instance, isolates failure-prone cases, performs residual reinforcement fine-tuning, and expands specialist behavior back into a general policy [18]. Earlier privileged-expert and dataset-aggregation ideas likewise show that learner-induced states and targeted corrective supervision are valuable [13,25]. FAR-MindDrive derives its correction signal from the failure event as a temporal anchor, separate probes of longitudinal and lateral decisions, and a minimum event-specific geometric-risk reduction. This formulation narrows reinforcement-style post-training into **attribution-guided policy repair**.

Technical distinction from MindDrive. MindDrive combines a language-level Decision Expert with a continuous Action Expert and trains the Decision Expert through online PPO with scalar simulator rewards [19]. FAR-MindDrive builds on its two-expert architecture, meta-action vocabulary, and pretrained weights. Its contribution is the training-time transformation of sparse failures into factor-specific, confidence-weighted corrective targets through event-centered sampling, factorized risk probing with a frozen Action Expert, and selective offline LoRA repair. Cached counterfactual evaluations assign the feedback to individual decision factors, while imitation replay and reference anchoring stabilize its reuse during optimization.

2.4. Research Gap: From Failure Feedback to Structured Corrective Supervision

The three research lines above have solved complementary parts of the E2E driving problem. Planning-centered models produce increasingly expressive and temporally consistent trajectories; VLM/VLA models add semantic reasoning and interpretable decision interfaces; reinforcement and preference methods allow a policy to improve beyond static demonstrations. What remains underexplored is the conversion of a **realized closed-loop failure** into a **minimal corrective update** when the action is already structured at the language level.

Existing future-modeling and trajectory-scoring methods estimate trajectory risk while leaving the responsible high-level VLA factor implicit. Language-action alignment methods connect semantic decisions to executable motion, typically through dataset-level consistency objectives. RL propagates delayed reward through value estimation, producing updates entangled across time and action dimensions. Preference methods compare complete candidates, while factor-specific responsibility and state selection require an additional attribution mechanism.

FAR-MindDrive targets this gap through a deliberately local formulation. An event first localizes the relevant temporal region. Counterfactual interventions then isolate the sensitivity of the failure risk to longitudinal and lateral decisions. Finally, confidence gating determines whether the evidence is strong enough to justify modifying the Decision Expert. This sequence establishes the research chain developed across the first two sections of this paper:

closed-loop failure feedback $\rightarrow$ **temporal localization** $\rightarrow$ **factorized risk attribution** $\rightarrow$ **selective policy repair**.

The method supplies recent VLA reinforcement-learning pipelines with a structured stage before optimization. It converts closed-loop feedback into a smaller set of identified decisions while preserving capabilities learned from imitation.

3. Method

3.1. Overview and Problem Formulation

FAR-MindDrive is built on the language–action decomposition of MindDrive [19]. Given the current visual observation V_t, navigation instruction L_t, and the compact scene representation produced by the frozen visual encoder, the Decision Expert predicts a structured linguistic meta-action

$$m_t = \left(m_t^{\text{lon}}, m_t^{\text{lat}}\right), \tag{1}$$

where $m_t^{\text{lon}} \in \mathcal{M}_{\text{lon}}$ denotes a longitudinal decision such as slowing down, maintaining speed, or stopping, and $m_t^{\text{lat}} \in \mathcal{M}_{\text{lat}}$ denotes a lateral decision such as lane following or a directional maneuver. MindDrive uses separate LoRA adapters to specialize the shared language model into a Decision Expert and an Action Expert. The Action Expert maps the language-level decision to a temporal speed profile and a geometric path, which are decoded into an executable trajectory [19]. FAR-MindDrive retains this pretrained language–action interface and its inference architecture.

Let the Decision Expert be parameterized by π_θ and the frozen Action Expert by g_ϕ. For the state representation s_t, the executed short-horizon motion is written as

$$\xi_t^0 = g_\phi\left(s_t, m_t^{\text{lon}}, m_t^{\text{lat}}\right), \qquad m_t \sim \pi_\theta(\cdot \mid s_t). \tag{2}$$

In the original online reinforcement-learning formulation, a closed-loop route provides scalar rewards that are propagated to preceding actions through temporal-difference learn-

ing and generalized advantage estimation [19]. FAR-MindDrive instead focuses on routes that terminate or incur a safety-related event e, such as a collision, red-light violation, route deviation, or stop-sign violation. Suppose an event occurs at time T_e. The corresponding failure trace is

$$\mathcal{F}_e = \{(s_t, m_t, \zeta_t)\}_{t \le T_e}, \tag{3}$$

where ζ_t contains privileged post-training simulator states, including ego pose and velocity, nearby-agent geometry, route information, and traffic-control states. These states construct corrective labels and are excluded from deployed-policy observations.

FAR-MindDrive replaces a uniform failure return over $\mathcal{F}_e$ with a compact repair set

$$\mathcal{D}_{\text{rep}} = \left\{ \left(s_t, \hat{m}_t^{\text{lon}}, \hat{m}_t^{\text{lat}}, \alpha_t^{\text{lon}}, \alpha_t^{\text{lat}}, q_t \right) \right\}, \tag{4}$$

where $\hat{m}_t^{\text{lon}}$ and $\hat{m}_t^{\text{lat}}$ are locally preferred corrective decisions, α_t^{lon} and α_t^{lat} describe how strongly the observed failure is associated with the two decision factors, and q_t measures confidence in applying the correction.

The complete pipeline contains three stages. First, **Event-Centered Temporal Credit Backtracking** extracts a short decision history before each failure and assigns temporal responsibility. Second, **Factorized Counterfactual Risk Attribution** changes longitudinal and lateral meta-actions separately and measures the reduction in event-specific geometric risk under the frozen Action Expert. Third, **Confidence-Gated Conservative Policy Repair** uses sufficiently reliable corrections to update the Decision Expert LoRA, while the visual encoder, base language model, Action Expert, and trajectory decoder remain frozen. Closed-loop interaction discovers failures; offline attribution and lightweight policy repair then update the Decision Expert. Figure 1 summarizes this pipeline.

Trajectory Composition

MindDrive represents longitudinal and lateral motion using different waypoint structures [19]. To evaluate alternative meta-actions with a common geometric criterion, we convert the Action Expert outputs into a short-horizon spatiotemporal trajectory. Let

$$W_t^s = \{v_{t,k}\}_{k=1}^{K_s}, \qquad W_t^p = \{p_{t,j}\}_{j=1}^{K_p} \tag{5}$$

be the predicted speed waypoints and path waypoints, respectively. We first integrate the speed profile over interval Δt to obtain traveled arc length

$$\ell_{t,k} = \sum_{i=1}^{k} \frac{v_{t,i-1} + v_{t,i}}{2} \Delta t. \tag{6}$$

The geometric path is then interpolated at $\ell_{t,k}$,

$$\bar{p}_{t,k} = \text{Interp}\left(W_t^p, \ell_{t,k} \right), \tag{7}$$

yielding

$$\xi_t = \left\{ (\bar{p}_{t,k}, v_{t,k}) \right\}_{k=1}^{K_s}. \tag{8}$$

This deterministic composition allows longitudinal and lateral interventions to be evaluated in the same trajectory space without introducing an additional learned planner.

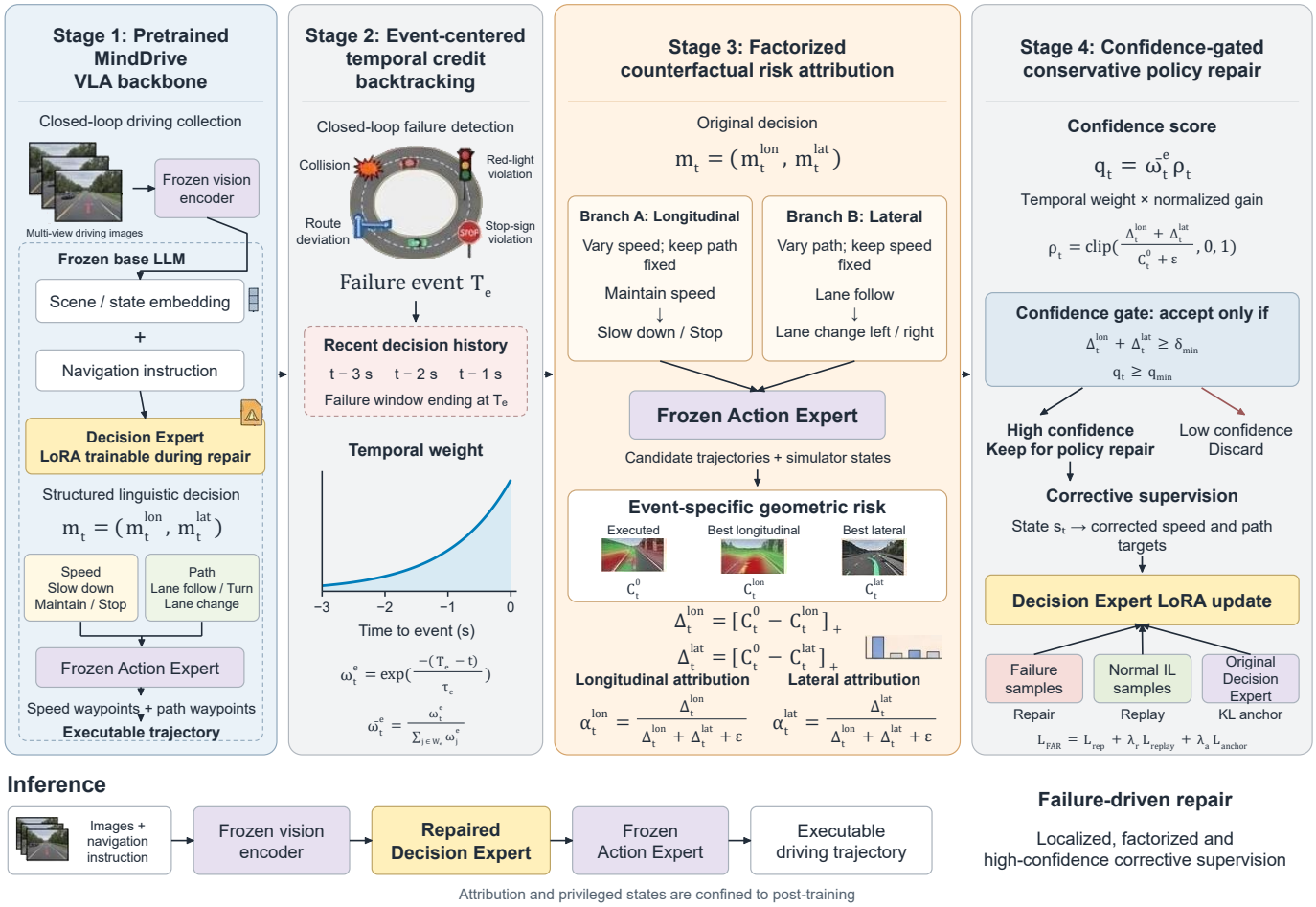


Figure 1. Overall architecture of FAR-MindDrive. The framework starts from the pretrained MindDrive language–action backbone, localizes a closed-loop failure with event-centered temporal credit backtracking, attributes the failure to longitudinal and lateral meta-actions through local counterfactual risk evaluation, and finally performs confidence-gated conservative repair of the Decision Expert. All attribution components are used only during post-training; inference retains the original Decision Expert-to-Action Expert pipeline. Pale blue, light gray, pale orange, and light gray backgrounds distinguish the pretrained backbone, temporal backtracking, factorized attribution, and conservative repair stages, respectively. Yellow highlights identify the trainable Decision Expert, purple blocks identify frozen experts, and arrows indicate information flow. Road-scene thumbnails and the event illustration are retained from the original diagram; the three risk-evaluation thumbnails illustrate the executed decision and the best longitudinal and lateral alternatives.

3.2. Event-Centered Temporal Credit Backtracking

A closed-loop violation provides a precise event time but only weak information about which preceding decision was responsible. Updating an entire route with the same negative label is unnecessarily broad because the relevance of an action generally decreases as its temporal distance from the event increases. We therefore localize each failure to a short history window.

For an event e occurring at T_e, we define the candidate backtracking set

$$\mathcal{W}_e = \{t \mid 0 \le T_e - t \le H_e\}, \tag{9}$$

where H_e is the maximum backtracking horizon. Each state receives an event-centered temporal weight

$$\omega_t^e = \exp\left(-\frac{T_e - t}{\tau_e}\right), \qquad t \in \mathcal{W}_e, \tag{10}$$

where $\tau_e > 0$ controls the decay rate. For numerical stability and comparability across events with different numbers of recorded decisions, we normalize the weights within the window:

$$\bar{\omega}_t^e = \frac{\omega_t^e}{\sum_{j \in \mathcal{W}_e} \omega_j^e}. \tag{11}$$

Equation (10) encodes a temporal prior: decisions immediately preceding an event receive greater scrutiny than temporally distant decisions. Temporal proximity supplies first-stage localization, and the subsequent counterfactual attribution tests action responsibility. A temporally close state enters the repair set when an alternative meta-action demonstrates a meaningful reduction in event-specific risk.

This two-step design separates **temporal relevance** from **action responsibility**. The temporal module identifies where to search, while the counterfactual module determines whether the decision at that state admits a safer local alternative. As a result, FAR-MindDrive avoids assigning high repair weights solely because a state happens to occur near a failure.

Logged decisions and candidate corrections. The executed decision at an event-window state, $m_t^0 = (m_t^{\text{lon}}, m_t^{\text{lat}})$, comes from the logged policy rollout. We construct alternative meta-actions at that recorded state and use the frozen Action Expert to generate their trajectories. The resulting retrospective candidates provide corrective targets when their event-specific proxy risk decreases and the thresholds in Equation (29) are satisfied. Action-generation failures and unsuccessful corrections are examined separately in Section 4.3.6.

3.3. Factorized Counterfactual Risk Attribution

MindDrive already decomposes high-level control into longitudinal and lateral meta-actions [19]. FAR-MindDrive exploits this structure as an intervention space. For each selected state s_t, we construct small local candidate sets

$$\mathcal{N}_{\text{lon}}\left(m_t^{\text{lon}}\right) \subseteq \mathcal{M}_{\text{lon}}, \qquad \mathcal{N}_{\text{lat}}\left(m_t^{\text{lat}}\right) \subseteq \mathcal{M}_{\text{lat}}, \tag{12}$$

containing the original decision and a few semantically adjacent alternatives. This avoids enumerating the full Cartesian product of the discrete action space.

3.3.1. Factorized Interventions

To test longitudinal responsibility, we vary only the longitudinal decision while fixing the executed lateral decision:

$$\zeta_t^{\text{lon}}(a) = g_\phi\left(s_t, a, m_t^{\text{lat}}\right), \qquad a \in \mathcal{N}_{\text{lon}}. \tag{13}$$

Similarly, lateral responsibility is probed by

$$\zeta_t^{\text{lat}}(b) = g_\phi\left(s_t, m_t^{\text{lon}}, b\right), \qquad b \in \mathcal{N}_{\text{lat}}. \tag{14}$$

The Action Expert remains frozen during this procedure, so differences among candidates are induced by the intervened linguistic decision under the same scene representation. The resulting trajectories are **local short-horizon counterfactual proposals** under a fixed scene state. They estimate local risk sensitivity; interactive closed-loop evaluation supplies the final performance measure.

3.3.2. Event-Specific Geometric Risk

A single manually weighted reward that mixes all driving objectives can obscure why a particular failure occurred. We instead associate each observed event with a dedicated geometric risk function

$$C_e(\xi; \zeta_t) \in [0, 1]. \tag{15}$$

Only the risk corresponding to the observed event is used to construct the local correction. This avoids forcing collision avoidance, traffic-rule compliance, and route following into a single fixed scalarization.

For a collision-related event, surrounding-agent boxes are extrapolated over the short horizon using their current simulator states. Let $B_k^{\text{ego}}(\xi)$ be the ego box induced by candidate trajectory ξ, and $\hat{B}_{j,k}$ the predicted box of nearby agent j. We define

$$d_{\min}(\xi) = \min_{k,j} d_{\text{box}}\left(B_k^{\text{ego}}(\xi), \hat{B}_{j,k}\right), \tag{16}$$

and use the normalized proximity cost

$$C_{\text{col}}(\xi) = \exp\left(-\frac{d_{\min}(\xi)}{\sigma_d}\right), \tag{17}$$

where overlapping boxes yield $d_{\min} = 0$ and therefore maximum risk.

For a red-light event, the risk checks whether the candidate trajectory crosses the active stop line while the signal is red:

$$C_{\text{red}}(\xi) = \max_k \mathbf{1}[\text{CrossStopLine}(\bar{p}_{t,k}) \wedge r_{t,k} = \text{red}]. \tag{18}$$

For route deviation, we measure the maximum normalized distance to the valid route corridor:

$$C_{\text{route}}(\xi) = \min\left(1, \frac{\max_k d_{\text{route}}(\bar{p}_{t,k})}{d_{\max}}\right). \tag{19}$$

For a stop-sign event, a candidate is penalized when it enters the stop region without reducing speed below a threshold:

$$C_{\text{stop}}(\xi) = \max_k \left[\mathbf{1}(\bar{p}_{t,k} \in \Omega_{\text{stop}}) \cdot \min\left(1, \frac{v_{t,k}}{v_{\text{stop}} + \epsilon}\right)\right]. \tag{20}$$

These definitions use privileged geometry during repair-label construction. The deployed driving policy retains its original observation space.

Scene-state approximation. Proxy risks use simulator-provided agent boxes, velocities, route geometry, and signal states during label construction. Nearby-agent motion is extrapolated over the short horizon with a non-reactive forecast. Acceleration, braking, or yielding by other agents can change the relative risk of candidate trajectories. Section 4.3.5 evaluates sensitivity to position and velocity perturbations, and Section 5 discusses transfer requirements.

3.3.3. Risk Reduction and Factor Attribution

Let

$$C_t^0 = C_e(\xi_t^0; \zeta_t) \tag{21}$$

be the risk of the executed decision. We choose the lowest-risk longitudinal and lateral alternatives independently:

$$\hat{m}_t^{\text{lon}} = \arg\min_{a \in \mathcal{N}_{\text{lon}}} C_e\left(\xi_t^{\text{lon}}(a); \zeta_t\right), \tag{22}$$

$$\hat{m}_t^{\text{lat}} = \arg\min_{b \in \mathcal{N}_{\text{lat}}} C_e\left(\xi_t^{\text{lat}}(b); \zeta_t\right). \tag{23}$$

The corresponding positive risk reductions are

$$\Delta_t^{\text{lon}} = \left[C_t^0 - C_e\left(\xi_t^{\text{lon}}(\hat{m}_t^{\text{lon}}); \zeta_t\right)\right]_+, \tag{24}$$

$$\Delta_t^{\text{lat}} = \left[C_t^0 - C_e\left(\xi_t^{\text{lat}}(\hat{m}_t^{\text{lat}}); \zeta_t\right)\right]_+. \tag{25}$$

We then normalize the two improvements into attribution weights

$$\alpha_t^{\text{lon}} = \frac{\Delta_t^{\text{lon}}}{\Delta_t^{\text{lon}} + \Delta_t^{\text{lat}} + \epsilon}, \qquad \alpha_t^{\text{lat}} = \frac{\Delta_t^{\text{lat}}}{\Delta_t^{\text{lon}} + \Delta_t^{\text{lat}} + \epsilon}. \tag{26}$$

If neither factor yields a meaningful improvement, both risk reductions are close to zero and the sample is rejected in the next stage. If changing speed sharply lowers risk while changing path has little effect, α_t^{lon} approaches one. The reverse holds for a lateral error. When both interventions are helpful, the supervision is shared between the two components.

Equation (26) quantifies local **risk sensitivity** to each meta-action factor under the fixed scene state and frozen action generator.

3.4. Confidence-Gated Conservative Policy Repair

Counterfactual candidates are useful only when they provide sufficiently strong evidence that the executed decision should be changed. FAR-MindDrive therefore converts temporal relevance and risk reduction into a confidence score before updating the policy.

For state t, the normalized corrective gain is

$$\rho_t = \text{clip}\left(\frac{\Delta_t^{\text{lon}} + \Delta_t^{\text{lat}}}{C_t^0 + \epsilon}, 0, 1\right). \tag{27}$$

We combine it with the normalized temporal weight from Equation (11):

$$q_t = \bar{\omega}_t^e \rho_t. \tag{28}$$

A sample is added to the repair set only when the local intervention produces a minimum absolute risk reduction and the resulting confidence exceeds a threshold:

$$\Delta_t^{\text{lon}} + \Delta_t^{\text{lat}} \geq \delta_{\min}, \qquad q_t \geq q_{\min}. \tag{29}$$

This rule admits states that are both temporally relevant and supported by a clearly safer local decision. The minimum-gain condition filters negligible changes in the geometric proxy, and the confidence condition combines that gain with temporal relevance.

3.4.1. Factor-Weighted Repair Objective

The Decision Expert represents longitudinal and lateral decisions as language-level outputs. We denote the corresponding conditional distributions by π_θ^{lon} and π_θ^{lat}. For the repair set $\mathcal{D}_{\text{rep}}$, the corrective objective is

$$\mathcal{L}_{\text{rep}} = -\frac{1}{|\mathcal{D}_{\text{rep}}|} \sum_{t \in \mathcal{D}_{\text{rep}}} q_t \left[\alpha_t^{\text{lon}} \log \pi_\theta^{\text{lon}} \left(\hat{m}_t^{\text{lon}} \mid s_t \right) + \alpha_t^{\text{lat}} \log \pi_\theta^{\text{lat}} \left(\hat{m}_t^{\text{lat}} \mid s_t \right) \right]. \tag{30}$$

The attribution weights ensure that the factor that more strongly reduces the event-specific risk receives a larger gradient. Unlike route-level reinforcement learning, the training target is therefore explicit at both the state level and the action-factor level.

3.4.2. Imitation Replay and Reference Anchoring

A repair set is concentrated on failures and underrepresents normal driving states. Optimizing Equation (30) alone can therefore overfit difficult events and weaken previously learned behavior. We retain a small replay buffer $\mathcal{D}_{\text{IL}}$ of normal or expert-aligned driving states and apply the original meta-action likelihood objective

$$\mathcal{L}_{\text{replay}} = -\mathbb{E}_{(s,m^*)\sim\mathcal{D}_{\text{IL}}} \log \pi_\theta(m^* \mid s). \tag{31}$$

In addition, the pre-repair Decision Expert is frozen as a reference policy π_{ref}. We constrain the repaired policy with

$$\mathcal{L}_{\text{anchor}} = \mathbb{E}_{s\sim\mathcal{D}_{\text{rep}}\cup\mathcal{D}_{\text{IL}}} D_{\text{KL}}(\pi_{\text{ref}}(\cdot \mid s) \| \pi_\theta(\cdot \mid s)). \tag{32}$$

The final optimization objective is

$$\mathcal{L}_{\text{FAR}} = \mathcal{L}_{\text{rep}} + \lambda_{\text{r}} \mathcal{L}_{\text{replay}} + \lambda_{\text{a}} \mathcal{L}_{\text{anchor}}, \tag{33}$$

where λ_{r} and λ_{a} control capability retention. Only the Decision Expert LoRA parameters are updated. The vision encoder, shared base LLM weights, Action Expert LoRA, and trajectory decoder remain frozen.

3.4.3. Training and Inference

The complete procedure consists of three operational phases:

1. **Failure collection.** The pretrained MindDrive policy is evaluated in the closed-loop simulator. Failure traces and simulator states are stored when a target event is triggered.
2. **Offline attribution.** Event-centered windows are extracted, local longitudinal and lateral interventions are generated by the frozen Action Expert, and Equations (15)–(29) produce high-confidence corrective labels.
3. **Policy repair.** The Decision Expert LoRA is optimized using Equation (33). No simulator interaction is required during this optimization phase.

At inference time, the repaired Decision Expert selects a meta-action through the original language–action pipeline, and the unchanged Action Expert maps it to executable motion. Attribution modules, privileged simulator states, counterfactual generation, and replay objectives are confined to training, so deployment retains the original decision path.

4. Experiments

4.1. Experimental Details

4.1.1. Dataset and Evaluation Protocol

We conduct experiments on Bench2Drive [6], a closed-loop benchmark built on CARLA that evaluates autonomous-driving policies under diverse interactive scenarios. Following the MindDrive protocol [19], the language–action policy operates with 7 longitudinal speed meta-actions and 6 lateral path meta-actions. The Action Expert predicts a 3 s temporal speed profile with 6 waypoints and a geometric path represented by 20 waypoints. Our local training data are drawn from an approximately 50 GB subset of the Bench2Drive Base training set. The subset is used to construct imitation replay samples and to support the failure-repair pipeline, while the final closed-loop evaluation follows the official Bench2Drive route definitions.

We use the public Qwen2-0.5B MindDrive model and its released weights as the starting point. The experiments initialize from the imitation-trained MindDrive policy and apply FAR-MindDrive in place of the original online PPO stage. The vision encoder, shared base language-model parameters, Action Expert LoRA, and trajectory decoder are frozen throughout repair. The Decision Expert LoRA is the trainable module.

Closed-loop performance is evaluated using **Driving Score (DS)** and **Success Rate (SR)** as the primary metrics. DS jointly reflects route completion and infraction penalties, while SR measures the percentage of routes completed within the specified time. We additionally report the official multi-ability score when available and record event-level violation rates for collision, traffic-light violation, route deviation, and stop-sign violation. Lower violation rates indicate safer behavior. For attribution diagnostics, we further record the number of retained repair samples and the distribution of longitudinal and lateral responsibility weights.

Independent-run protocol. Locally conducted quantitative comparisons are reported as arithmetic means of five independent runs with common route definitions and evaluation metrics. Evaluation counts in Table 1 are pooled across runs; repair-buffer and diagnostic counts refer to the separate cohorts identified below. Case-level figures report individual candidates. For measurements $x_1, \ldots, x_5$, the sample standard deviation is $s = \sqrt{\sum_{i=1}^{5}(x_i - \bar{x})^2/4}$ and the two-sided 95% Student-t confidence interval is $\bar{x} \pm 2.776 s/\sqrt{5}$. We use two-sided Welch tests to compare each locally repeated baseline with FAR-MindDrive. The five-run statistical analysis below reports the corresponding standard deviations, confidence intervals, and p values.

4.1.2. Failure Collection and Attribution Settings

FAR-MindDrive focuses on four failure types used by the original MindDrive online-RL protocol: **collision**, **traffic-light violation**, **route deviation**, and **stop-sign violation** [19]. Once one of these events is detected, the corresponding state history is stored and only a short interval before the event is considered for attribution. The maximum backtracking horizon is set to

$$H_e = 3.0 \text{ s}, \tag{34}$$

and the temporal decay coefficient in Equation (10) is

$$\tau_e = 1.0 \text{ s}. \tag{35}$$

For each selected state, at most three longitudinal and three lateral candidates are generated from the local neighborhood of the executed meta-action. Candidate trajectories are produced by the frozen Action Expert without gradient computation. We use the

event-specific risk definitions in Equations (17)–(20) to rank candidates. Unless otherwise stated, a state is retained only when the absolute risk reduction satisfies

$$\delta_{\min} = 0.15 \tag{36}$$

and the confidence score in Equation (28) satisfies

$$q_{\min} = 0.05. \tag{37}$$

The candidate generation and risk evaluation stages are cached so that the same repair set can be reused by different training ablations without rerunning the closed-loop simulator.

Failure cohort and candidate selection. Table 1 separates the final evaluation, repair-buffer selection, and diagnostic samples. The repair buffer is collected independently of the final evaluation routes. Its event-window states yield local candidates that are filtered by proxy-risk improvement, confidence, feasibility, and consistency, retaining 400 repairs. Evaluation event counts describe the four targeted infractions, with each event assigned to one primary Bench2Drive ability group; they are not a partition of incomplete routes. The 26 Pareto-improving diagnostic candidates and the 100 diagnostic trials discussed in Section 4.3.6 are drawn from Bench2Drive Base data and used separately from repair training. Four candidates illustrate the local attribution mechanism in the qualitative and attribution analyses in Sections 4.2 and 4.3.2.

Table 1. Failure cohort and candidate-selection statistics. Evaluation counts pool five independent runs; repair-buffer and diagnostic counts describe separate cohorts.

Evaluation and Selection Unit	Count
Final closed-loop evaluation	
Unique evaluation routes/independent repeats	220/5
Executed trials/successful completions/incomplete routes	1100/635/465
Collision/traffic-light/route-deviation/stop-sign events	65/50/50/35
Primary ability: merging/overtaking/emergency braking	42/55/38
Primary ability: give way/traffic sign	34/31
Independent repair buffer	
Event-window states/non-original local candidate trajectories	1000/4000
Rejected: no proxy gain/low confidence/infeasible	2100/920/250
Rejected: inconsistent or out-of-scope proxy assessment	330
Accepted high-confidence repair candidates	400
Diagnostics from Bench2Drive Base	
Diagnostic Pareto-improving subset/displayed examples	26/4

4.1.3. Optimization and Implementation

All FAR-MindDrive experiments are trained on a **single NVIDIA RTX 4090 GPU with 24 GB memory**. We use LoRA rank $r = 8$ and scaling factor $\alpha = 16$ for the Decision Expert. The optimizer is AdamW with an initial learning rate of 1×10^{-4}, weight decay of 1×10^{-2}, and a cosine learning-rate schedule. The per-GPU batch size is 8 with gradient accumulation over 4 steps, giving an effective batch size of 32. We train for 3 epochs using mixed-precision arithmetic. The replay ratio is set to 0.15. The loss weights in Equation (33) are

$$\lambda_{\mathrm{r}} = 0.30, \qquad \lambda_{\mathrm{a}} = 0.05. \tag{38}$$

The repair stage uses supervised LoRA optimization with cached candidates. Cache construction takes 2.6 h and Decision Expert optimization takes 3.8 h, for a total first-run

time of 6.4 h on a single RTX 4090. Subsequent ablations reuse the cache and complete in 3.2–4.1 h.

Table 2 summarizes the default settings.

Table 2. Default experimental settings.

Setting	Value
Base model	MindDrive, Qwen2-0.5B
Training GPU	1 × RTX 4090 24 GB
Trainable module	Decision Expert LoRA
LoRA rank/alpha	8/16
Backtracking horizon H_e	3.0 s
Temporal decay τ_e	1.0 s
Longitudinal candidates	$\leq$3 per state
Lateral candidates	$\leq$3 per state
Minimum risk reduction $\delta_{\min}$	0.15
Minimum confidence $q_{\min}$	0.05
Batch size/accumulation	8/4
Effective batch size	32
Learning rate	1×10^{-4}
Epochs	3
Replay ratio	0.15
λ_r/λ_a	0.30/0.05

4.2. Comparison with State-of-the-Art Methods

Table 3 compares FAR-MindDrive with representative methods from three relevant paradigms: conventional E2E imitation learning, model-based or preference-based policy refinement, and VLA driving. The comparison includes ThinkTwice [3], DriveAdapter [19,29], DriveTransformer [30], DriveDPO [15], Raw2Drive [16], ORION [11], ReCogDrive [46], and the two MindDrive variants [19]. This selection covers planning-oriented E2E systems, preference-based optimization, reinforcement-learning-based refinement, and language–action driving.

The MindDrive-IL, MindDrive, and FAR-MindDrive rows report five-run means from the present evaluation. Other baseline values follow the published Bench2Drive Base comparison reported by MindDrive [19]. All rows use Bench2Drive metrics; the locally repeated methods share the Qwen2-0.5B backbone.

Table 3. Closed-loop comparison on Bench2Drive. Higher is better for DS, SR, and mean ability; bold values mark the best result within the listed comparison set. Mean ability is the average of the five Bench2Drive advanced-driving abilities when reported.

Method	Venue	Scheme	LLM	DS ↑	SR (%) ↑	Mean Ability (%) ↑
ThinkTwice [3]	CVPR 2023	IL	–	62.44	31.23	37.17
DriveAdapter [19,29]	ICCV 2023	IL	–	64.22	33.08	42.08
DriveTransformer-L [30]	ICLR 2025	IL	–	63.46	35.01	38.60
DriveDPO [15]	NeurIPS 2025	Offline preference	–	62.02	30.62	–
Raw2Drive [16]	NeurIPS 2025	Model-based RL	–	71.36	50.24	53.34
ORION [11]	ICCV 2025	IL	Qwen2-0.5B	72.89	45.83	51.37
ReCogDrive [46]	ICLR 2026	RL refinement	Qwen2.5	71.36	45.45	42.03
MindDrive-IL [19]	ECCV 2026	IL	Qwen2-0.5B	75.85	49.30	49.44
MindDrive [19]	ECCV 2026	Online RL	Qwen2-0.5B	78.04	55.09	56.94
FAR-MindDrive	Ours	Failure-aware repair	Qwen2-0.5B	**79.12**	**57.73**	**58.81**

Note: ↑ indicates that higher values are better. Bold numeric values mark the best result in each metric; the corresponding method name is also bold.

Five-run statistical analysis. Table 4 summarizes variation across the locally repeated MindDrive-IL, MindDrive, and FAR-MindDrive runs. Each baseline is compared with FAR-MindDrive using a two-sided Welch test for DS and SR separately.

Table 4. Five-run statistical summary. Mean, standard deviation, 95% confidence interval, and Welch-test results for locally repeated methods.

Method	Metric	Mean	SD	95% CI	p vs. FAR
MindDrive-IL	DS	75.85	0.51	[75.21, 76.49]	<0.001
	SR (%)	49.30	1.07	[47.98, 50.63]	<0.001
MindDrive (online RL)	DS	78.04	0.54	[77.37, 78.71]	0.0104
	SR (%)	55.09	0.82	[54.07, 56.11]	<0.001
FAR-MindDrive	DS	79.12	0.48	[78.53, 79.71]	–
	SR (%)	57.73	0.72	[56.83, 58.63]	–

FAR-MindDrive improves DS from 75.85 to 79.12 and SR from 49.30% to 57.73% over the matched MindDrive-IL initialization. Relative to the listed online-RL MindDrive model, the gains are 1.08 DS and 2.64 percentage points in SR. The result supports the central hypothesis of this work: localizing closed-loop feedback to relevant time steps and decision factors produces an effective repair signal for the Decision Expert.

For mechanism inspection, we form a diagnostic Pareto-improving subset from the evaluated candidate branches by requiring lower event-specific proxy risk and lower branch-matched Expert/GT average displacement error (ADE). This diagnostic selection is separate from the training gate in Equation (29). Figure 2 shows 4 of the 26 diagnostic candidates and compares the MindDrive baseline, the selected FAR-MindDrive candidate, and the corresponding Expert/GT trajectory in obstacle-avoidance, normal-following, stopping, and merge scenes.

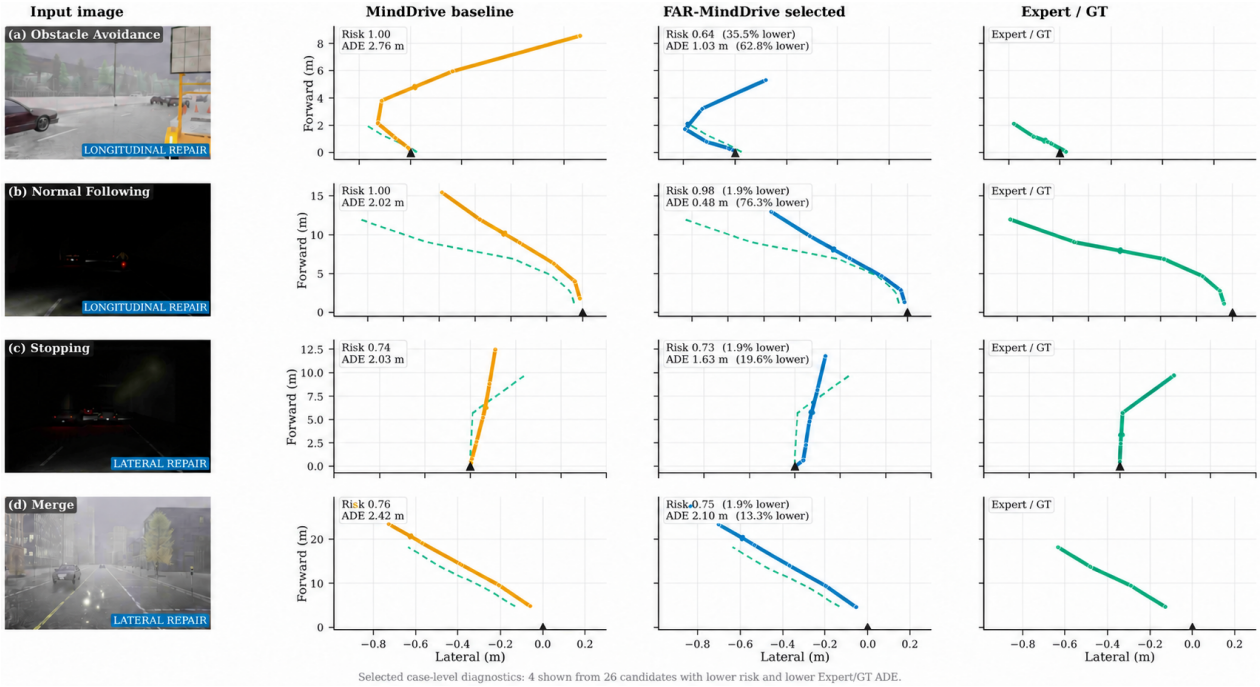


Figure 2. Case-level trajectory comparison for four diagnostic candidates. Each row shows the input scene, the MindDrive baseline, the selected FAR-MindDrive candidate, and the Expert/GT trajectory. The displayed cases are 4 of 26 evaluated candidates that reduce both event-specific proxy risk and branch-matched Expert/GT ADE. The training repair set is filtered separately by Equation (29). Orange, blue, and green curves denote the MindDrive baseline, selected FAR-MindDrive candidate, and Expert/GT trajectory, respectively; dashed green curves repeat the reference in the comparison panels. Black triangular markers indicate the ego vehicle at the trajectory origin.

Table 5 reports event-level violation rates measured in the common evaluation runs used in this study. The citations identify the evaluated model definitions, and all rates in the table come from the present evaluation records.

Table 5. Event-level closed-loop error analysis as means from five independent evaluation runs. Lower is better; FAR-MindDrive records the lowest rate for all four event types in this comparison.

Method	Collision (%) ↓	Traffic Light (%) ↓	Route Deviation (%) ↓	Stop Violation (%) ↓
MindDrive-IL [19]	10.91	7.27	7.73	5.45
MindDrive [19]	8.18	5.91	5.91	4.09
FAR-MindDrive	**5.91**	**4.55**	**4.55**	**3.18**

Note: ↓ indicates that lower values are better. Bold values mark the lowest violation rate in each column; the corresponding method name is also bold.

FAR-MindDrive reduces the collision rate from 10.91% for MindDrive-IL and 8.18% for MindDrive to 5.91%. Traffic-light, route-deviation, and stop-sign violation rates decrease to 4.55%, 4.55%, and 3.18%, respectively. DS and SR increase alongside these lower event rates, linking the aggregate driving improvement to fewer recorded violations in the common evaluation.

Figure 3 summarizes the same diagnostic Pareto-improving subset, independently of the training gate in Equation (29). Across 26 candidates, the mean route-deviation proxy risk decreases from 0.84 to 0.78 and the mean stop-violation proxy risk decreases from 0.62 to 0.58. The corresponding branch-matched ADE decreases from 2.44 m to 1.53 m and from 2.28 m to 1.56 m. These proxy statistics describe the selected candidate subset, while Table 5 reports the separate closed-loop evaluation.

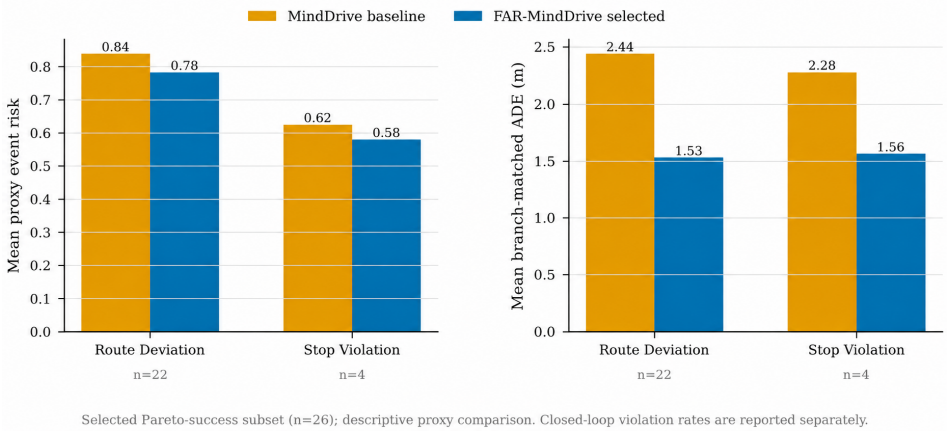


Figure 3. Descriptive comparison on the diagnostic Pareto-improving subset (26 candidates). (**Left**) mean event-specific proxy risk for route-deviation and stop-violation cases. (**Right**) mean branch-matched Expert/GT ADE. Orange bars show MindDrive baselines and blue bars show selected FAR-MindDrive candidates. Sample counts appear below each event group.

Efficiency

Table 6 reports the measured computational footprint of cache construction and offline repair on one RTX 4090. Initial simulator collection precedes these stages and is excluded from their reported times.

FAR-MindDrive updates 7.6 M parameters and reaches a peak memory footprint of 21.3 GB. Cache construction and the first optimization run take 6.4 h, while subsequent optimization runs reuse the cache and finish in 3.2–4.1 h.

Computation compared with online RL. MindDrive reports approximately 24 h of parallel experience collection with 24 CARLA instances and online training on 32 A800

GPUs [19]. FAR-MindDrive reuses a failure buffer and caches counterfactual evaluations for offline LoRA optimization. These resource profiles describe different collection and training configurations. A hardware-matched end-to-end comparison would include initial rollout time, simulator CPU-hours, GPU-hours, and cache reuse.

Table 6. Computational profile on a single RTX 4090. Cache construction and repair optimization take 6.4 h in total; initial simulator collection is excluded.

Item	Value
Trainable parameters	7.6 M
Peak GPU memory	21.3 GB
Counterfactual cache construction	2.6 h
Decision Expert repair training	3.8 h
Total first-run time	6.4 h
Subsequent ablation training with cache	3.2–4.1 h

4.3. Ablation Studies

We conduct ablations around the three questions raised by the method design: whether temporal localization is necessary, whether factorized counterfactual attribution provides useful corrective labels, and whether conservative repair is needed to retain previously learned driving ability.

4.3.1. Component Analysis

Table 7 denotes the component variants as Method (1) through Method (5).

- **Method (1)** is the original MindDrive-IL initialization without failure-driven repair.
- **Method (2)** adds factorized counterfactual risk attribution within a fixed 3 s window but assigns uniform temporal weights.
- **Method (3)** adds event-centered temporal credit backtracking to Method (2).
- **Method (4)** further applies confidence-gated sample filtering.
- **Method (5)** is the complete FAR-MindDrive model with imitation replay and reference-policy anchoring.

Table 7. Component ablation of FAR-MindDrive. Successive attribution, localization, filtering, replay, and anchoring components improve the reported closed-loop metrics.

Method	Factorized Attribution	Temporal Backtracking	Confidence Gating	Replay + Anchor	DS ↑	SR (%) ↑	Collision (%) ↓	Mean Ability (%) ↑
Method (1)					75.85	49.30	10.91	49.44
Method (2)	✓				77.05	52.27	8.64	52.73
Method (3)	✓	✓			77.74	53.64	7.73	54.62
Method (4)	✓	✓	✓		78.46	55.45	6.82	57.06
Method (5)	✓	✓	✓	✓	**79.12**	**57.73**	**5.91**	**58.81**

Note: ↑/↓ indicate that higher/lower values are better. A checkmark indicates that the component is enabled; a blank cell indicates that it is disabled. Bold numeric values mark the best result in each metric; the corresponding method name is also bold.

Method (2) establishes the contribution of factorized intervention: independently changing longitudinal and lateral decisions provides more specific supervision than assigning the same route-level signal to all preceding decisions. Method (3) further improves performance by concentrating the search around states temporally closer to the event. The gain of Method (4) supports confidence gating, which excludes counterfactual candidates with weak or ambiguous risk reduction. Method (5) achieves the best overall trade-off; replay and reference anchoring stabilize the update and preserve normal driving behavior while the policy learns from a failure-biased repair set.

4.3.2. Attribution Behavior

Beyond aggregate metrics, Figure 4 inspects the same four diagnostic candidates as Figure 2. For each case, the left panel shows temporal backtracking weights over the final 3 s, the center panel overlays the MindDrive baseline, selected FAR-MindDrive candidate, and Expert/GT trajectory, and the right panel reports longitudinal/lateral attribution weights with the measured proxy-risk and ADE changes. The first two route-deviation examples are longitudinally dominated, whereas the stopping and merge examples are laterally dominated. The displayed candidate branch and factor weights expose case-specific attribution structure.

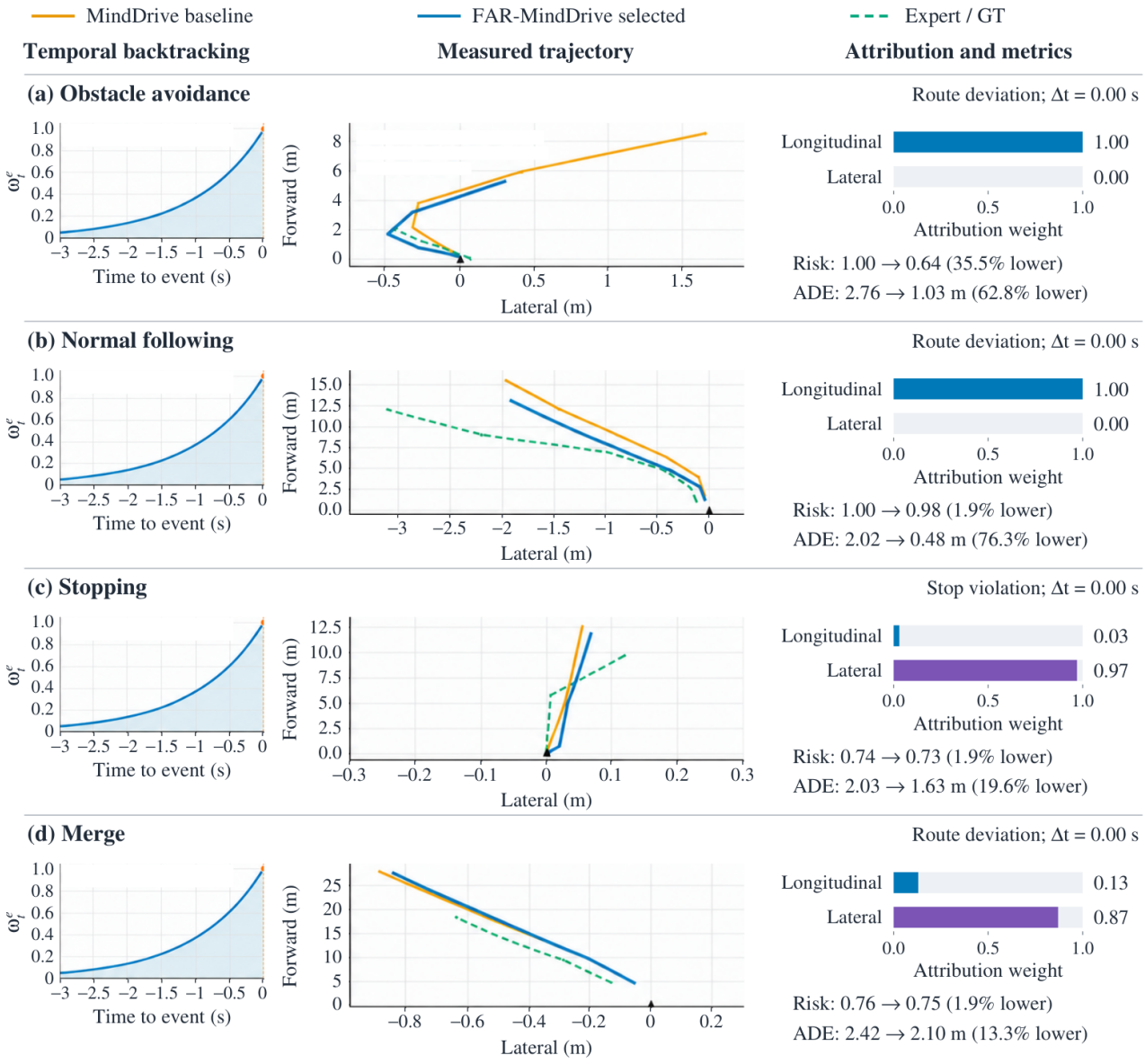


Figure 4. Failure-attribution diagnostics for four selected candidates from the diagnostic Pareto-improving subset. Each row shows the temporal backtracking profile, baseline/selected/reference trajectories, and longitudinal/lateral attribution weights with the associated proxy-risk and branch-matched ADE reductions. Training samples are admitted separately by Equation (29). In the trajectory panels, orange, blue, and dashed green denote the baseline, selected candidate, and Expert/GT reference. In the attribution panels, blue and purple bars denote longitudinal and lateral weights, respectively; light gray backgrounds show the full unit-weight scale. Black triangular markers indicate the ego vehicle at the trajectory origin.

Together, Figures 2 and 4 connect the method to observable repair behavior: temporal localization identifies the states under inspection, factorized risk reduction exposes the relative sensitivity of speed and path decisions, and each selected trajectory can be compared with the baseline and Expert/GT reference. Tables 3 and 5 provide the closed-loop evaluation, while the figures supply case-level and selected-subset evidence for the attribution mechanism.

4.3.3. Backtracking Horizon

The temporal horizon determines how far the method searches before a failure. A window that is too short may miss a decision whose consequence appears later, while an excessively long window introduces unrelated states. Table 8 evaluates this trade-off.

The results show that a 3 s window provides the best balance between delayed consequences and attribution noise. Increasing the window beyond this point adds candidate states and weakens the relation between the observed event and the repaired decision. The temporal module therefore concentrates the attribution search within a short event-centered interval.

Table 8. Effect of the backtracking horizon. A 3 s window gives the highest DS and SR and the lowest collision rate among the tested settings.

H_e	DS ↑	SR (%) ↑	Retained Repair Samples	Collision (%) ↓
1 s	78.31	55.91	1.00×	6.82
2 s	78.78	56.82	1.34×	6.36
3 s	**79.12**	**57.73**	1.58×	**5.91**
4 s	78.97	57.27	1.83×	6.14
5 s	78.55	56.36	2.07×	6.59

Note: ↑/↓ indicate that higher/lower values are better. Bold numeric values mark the best DS, SR, and collision results; the corresponding horizon is also bold.

4.3.4. Conservative Repair

We finally isolate the stabilization strategy used after the repair labels have been constructed. All variants in Table 9 use the same temporal backtracking, factorized attribution, and confidence filtering. They differ only in whether normal imitation replay and the reference-policy KL term are applied.

Table 9. Effect of conservative repair. Combining imitation replay and KL anchoring gives the highest DS, SR, and mean ability.

Method	IL Replay	KL Anchor	DS ↑	SR (%) ↑	Mean Ability (%) ↑
Method (1)			78.46	55.45	57.06
Method (2)	✓		78.73	56.36	57.94
Method (3)		✓	78.84	56.82	58.13
Method (4)	✓	✓	**79.12**	**57.73**	**58.81**

Note: ↑ indicates that higher values are better. A checkmark indicates that the component is enabled; a blank cell indicates that it is disabled. Bold numeric values mark the best result in each metric; the corresponding method name is also bold.

Replay exposes the updated policy to representative non-failure states, while KL anchoring constrains drift from the pretrained action distribution. Their combination yields the highest DS, SR, and mean ability in Table 9, supporting their complementary use during repair.

4.3.5. Hyperparameter Sensitivity and Proxy Robustness

Hyperparameter sensitivity and risk-proxy uncertainty. We vary one parameter at a time while holding the candidate buffer, remaining coefficients, and evaluation routes fixed.

Table 10 reports sensitivity to candidate-set size, temporal decay, correction thresholds, replay weight, and reference-policy regularization. The default configuration is reproduced in each group. Short-horizon risk calculations additionally depend on agent-state measurements and a non-reactive prediction of other agents. Table 11 reports the effect of controlled position and velocity perturbations on candidate-rank agreement, proxy false-safe rate, DS, and SR.

Table 10. Parameter sensitivity analysis. The default setting in each group is shown in bold.

Varied Parameter	Setting	DS	SR (%)
Local candidates per factor	2	78.74	56.88
	3	**79.12**	**57.73**
	4	79.07	57.58
Temporal decay τ_e (s)	0.5	78.70	56.91
	1.0	**79.12**	**57.73**
	2.0	78.85	57.14
Risk threshold $\delta_{\min}$	0.10	78.83	57.05
	0.15	**79.12**	**57.73**
	0.20	78.92	57.28
Confidence threshold $q_{\min}$	0.03	78.86	57.22
	0.05	**79.12**	**57.73**
	0.07	78.81	57.02
Replay coefficient λ_{r}	0.15	78.82	57.04
	0.30	**79.12**	**57.73**
	0.45	78.91	57.19
Anchor coefficient λ_{a}	0.02	78.78	57.00
	0.05	**79.12**	**57.73**
	0.10	78.96	57.29

Table 11. Geometric perturbation analysis. Rank agreement is measured relative to the unperturbed inputs.

Position/Speed Perturbation	Rank Agreement (%)	Proxy False-Safe (%)	DS	SR (%)
0 m/0 m s^{-1}	100.0	2.3	79.12	57.73
0.5 m/0.5 m s^{-1}	93.4	4.1	78.93	57.31
1.0 m/1.0 m s^{-1}	86.7	7.8	78.58	56.64

4.3.6. Action-Generation Failures and Unsuccessful Corrections

Table 12 evaluates 100 diagnostic trials drawn from Bench2Drive Base data separately from repair training. It distinguishes successful closed-loop corrections, proxy-improving candidates that fail during closed-loop execution, and infeasible Action Expert outputs despite suitable meta-actions. The 76 successful cases, 15 unsuccessful corrections, and 9 infeasible outputs account for all 100 trials, identifying failures in both proxy ranking and trajectory generation.

Table 12. Diagnostic failure cases. The 100 diagnostic trials are evaluated separately from repair training.

Diagnostic Outcome	Count
Proxy improvement and successful closed-loop correction	76
Proxy improvement but unsuccessful in closed-loop execution	15
Action Expert generates infeasible motion despite suitable meta-action	9
Total diagnostic cases	100

4.3.7. Sequential Repair and Ability-Wise Retention

Repeated use of a failure-biased repair set can alter previously learned behaviors even when a single repair pass improves closed-loop performance. We therefore evaluate the same held-out route suite after consecutive repair cycles and record the five Bench2Drive abilities separately. Table 13 reports the resulting cycle-wise scores and shows that the main driving abilities remain stable after repeated repair.

Table 13. Sequential repair and ability-wise retention. Results are reported after each repair cycle on the same held-out route suite.

Policy Checkpoint	Merging	Overtaking	Emergency Brake	Give Way	Traffic Sign	DS/SR
MindDrive-IL (initial)	30.26	66.67	60.00	33.33	56.91	75.85/49.30
FAR repair cycle 1	35.10	77.20	71.05	52.80	57.90	79.12/57.73
FAR repair cycle 2	35.35	77.55	71.20	52.85	58.10	79.28/58.09
FAR repair cycle 3	35.10	77.30	71.00	52.50	57.90	79.15/57.82

5. Limitations and Future Research

FAR-MindDrive constructs training labels from privileged simulator geometry and short-horizon, non-reactive agent forecasts. State-estimation errors and interactive responses can change candidate rankings, as the perturbation analysis illustrates. Extending the method to other simulators or road data requires equivalent scene geometry, reliable state reconstruction, and evaluation of interactive forecasts under a shared transfer protocol.

A fixed Action Expert also constrains repair. In the 100-trial diagnostic evaluation, nine suitable meta-actions produced infeasible motion and fifteen proxy-improving candidates remained unsuccessful during closed-loop execution. These outcomes motivate action-model adaptation and improved risk prediction alongside Decision Expert repair. The selected trajectory figures illustrate individual mechanisms, while the complete evaluation and diagnostic cohorts quantify aggregate outcomes.

Further resource comparisons should measure simulator collection and GPU optimization together under matched hardware budgets. Such measurements would establish how failure-buffer collection and cache reuse affect the end-to-end cost of repeated repair.

6. Conclusions

This paper presented **FAR-MindDrive**, a failure-aware risk attribution framework for efficient post-training of autonomous-driving VLA models. Sparse route-level feedback leaves both the decisive time step and the dominant longitudinal or lateral factor unspecified. FAR-MindDrive resolves this ambiguity in three steps. Event-centered temporal backtracking first localizes the short decision history most relevant to a violation. Factorized counterfactual risk attribution then measures how local changes to speed and path meta-actions alter event-specific geometric risk. Finally, confidence-gated conservative repair applies well-supported corrections while imitation replay and reference-policy anchoring preserve pretrained driving behavior.

On Bench2Drive, FAR-MindDrive achieves five-run means of **79.12 DS**, **57.73% SR**, and **58.81% mean ability**, improving over MindDrive-IL by 3.27 DS and 8.43 percentage points in SR and over the listed online-RL MindDrive model by 1.08 DS and 2.64 percentage points in SR. In the common event-level evaluation, the collision, traffic-light, route-deviation, and stop-sign violation rates are 5.91%, 4.55%, 4.55%, and 3.18%, respectively. Cache construction and Decision Expert optimization take 6.4 h on a single RTX 4090, excluding initial simulator collection, and update 7.6 M parameters. By converting sparse failures into localized, factorized, and confidence-weighted supervision, FAR-MindDrive

improves the reported closed-loop driving metrics while retaining the deployed action-generation architecture.

The component and horizon ablations identify the contributions of factorized supervision and temporal weighting. Geometric perturbations and unsuccessful-correction diagnostics characterize the reliability of the local risk estimates, while sequential repair shows retention of the five evaluated driving abilities across three cycles.

Author Contributions: Conceptualization, Z.J.; methodology, Z.J.; software, Z.J.; validation, Z.J.; data curation, Z.J.; writing—original draft preparation, Z.J.; writing—review and editing, Z.J.; visualization, X.D.; supervision, X.D.; funding acquisition, X.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding. The APC was funded by Zhuoyang Jia.

Data Availability Statement: Bench2Drive is a publicly available benchmark. No new dataset is introduced in this work.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hu, Y.; Yang, J.; Chen, L.; Li, K.; Sima, C.; Zhu, X.; Chai, S.; Du, S.; Lin, T.; Wang, W.; et al. Planning-Oriented Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2023; pp. 17853–17862. https://doi.org/10.1109/CVPR52729.2023.01712.
2. Jiang, B.; Chen, S.; Xu, Q.; Liao, B.; Chen, J.; Zhou, H.; Zhang, Q.; Liu, W.; Huang, C.; Wang, X. VAD: Vectorized Scene Representation for Efficient Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2023; pp. 8340–8350. https://doi.org/10.1109/ICCV51070.2023.00766.
3. Jia, X.; Wu, P.; Chen, L.; Xie, J.; He, C.; Yan, J.; Li, H. Think Twice Before Driving: Towards Scalable Decoders for End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2023; pp. 21983–21994. https://doi.org/10.1109/CVPR52729.2023.02105.
4. Zheng, W.; Song, R.; Guo, X.; Zhang, C.; Chen, L. GenAD: Generative End-to-End Autonomous Driving. In *Computer Vision—ECCV 2024, Lecture Notes in Computer Science*; Springer Nature: Cham, Switzerland, 2025; Volume 15123, pp. 87–104. https://doi.org/10.1007/978-3-031-73650-6_6.
5. Liao, B.; Chen, S.; Yin, H.; Jiang, B.; Wang, C.; Yan, S.; Zhang, X.; Li, X.; Zhang, Y.; Zhang, Q.; et al. DiffusionDrive: Truncated Diffusion Model for End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 12037–12047. https://doi.org/10.1109/CVPR52734.2025.01124.
6. Jia, X.; Yang, Z.; Li, Q.; Zhang, Z.; Yan, J. Bench2Drive: Towards Multi-Ability Benchmarking of Closed-Loop End-to-End Autonomous Driving. *Adv. Neural Inf. Process. Syst. (NeurIPS) Datasets Benchmarks Track* **2024**, *37*, 819–844. https://doi.org/10.52202/079017-0025.
7. Shao, H.; Hu, Y.; Wang, L.; Song, G.; Waslander, S.L.; Liu, Y.; Li, H. LMDrive: Closed-Loop End-to-End Driving with Large Language Models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 15120–15130. https://doi.org/10.1109/CVPR52733.2024.01432.
8. Tian, X.; Gu, J.; Li, B.; Liu, Y.; Wang, Y.; Zhao, Z.; Zhan, K.; Jia, P.; Lang, X.; Zhao, H. DriveVLM: The Convergence of Autonomous Driving and Large Vision-Language Models. *Proc. Mach. Learn. Res.* **2025**, *270*, 4698–4726.
9. Sima, C.; Renz, K.; Chitta, K.; Chen, L.; Zhang, H.; Xie, C.; Beißwenger, J.; Luo, P.; Geiger, A.; Li, H. DriveLM: Driving with Graph Visual Question Answering. In *Computer Vision—ECCV 2024, Lecture Notes in Computer Science*; Springer Nature: Cham, Switzerland, 2025; Volume 15110, pp. 256–274. https://doi.org/10.1007/978-3-031-72943-0_15.
10. Renz, K.; Chen, L.; Arani, E.; Sinavski, O. SimLingo: Vision-Only Closed-Loop Autonomous Driving with Language-Action Alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 11993–12003. https://doi.org/10.1109/CVPR52734.2025.01120.
11. Fu, H.; Zhang, D.; Zhao, Z.; Cui, J.; Liang, D.; Zhang, C.; Zhang, D.; Xie, H.; Wang, B.; Bai, X. ORION: A Holistic End-to-End Autonomous Driving Framework by Vision-Language Instructed Action Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2025; pp. 24823–24834. https://doi.org/10.1109/ICCV51701.2025.02302.
12. de Haan, P.; Jayaraman, D.; Levine, S. Causal Confusion in Imitation Learning. *Adv. Neural Inf. Process. Syst. (NeurIPS)* **2019**, *32*, 11693–11704.

13. Ross, S.; Gordon, G.; Bagnell, J.A. A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics (AISTATS)*; PMLR: Fort Lauderdale, FL, USA, 2011; Volume 15, pp. 627–635.
14. Jiang, B.; Chen, S.; Zhang, Q.; Liu, W.; Wang, X. AlphaDrive: Unleashing the Power of VLMs in Autonomous Driving via Reinforcement Learning and Reasoning. *arXiv* **2025**, arXiv:2503.07608.
15. Shang, S.; Chen, Y.; Wang, Y.; Li, Y.; Zhang, Z. DriveDPO: Policy Learning via Safety DPO for End-to-End Autonomous Driving. *Adv. Neural Inf. Process. Syst. (NeurIPS)* **2025**, *38*, 90460–90480. https://doi.org/10.52202/085713-2724.
16. Yang, Z.; Jia, X.; Li, Q.; Yang, X.; Yao, M.; Yan, J. Raw2Drive: Reinforcement Learning with Aligned World Models for End-to-End Autonomous Driving (in CARLA v2). *Adv. Neural Inf. Process. Syst. (NeurIPS)* **2025**, *38*, 148637–148662. https://doi.org/10.52202/085713-4471.
17. Gao, H.; Chen, S.; Jiang, B.; Liao, B.; Shi, Y.; Guo, X.; Pu, Y.; Yin, H.; Li, X.; Zhang, X.; et al. RAD: Training an End-to-End Driving Policy via Large-Scale 3DGS-based Reinforcement Learning. *Adv. Neural Inf. Process. Syst. (NeurIPS)* **2025**, *38*, 36642–36667. https://doi.org/10.52202/085713-1092.
18. Liu, H.; Li, T.; Yang, H.; Chen, L.; Wang, C.; Guo, K.; Tian, H.; Li, H.; Li, H.; Lv, C. Reinforced Refinement With Self-Aware Expansion for End-to-End Autonomous Driving. *IEEE Trans. Pattern Anal. Mach. Intell.* **2026**, *48*, 5774–5792. https://doi.org/10.1109/TPAMI.2026.3653866.
19. Fu, H.; Zhang, D.; Zhao, Z.; Cui, J.; Xie, H.; Wang, B.; Chen, G.; Ye, H.; Liang, D.; Bai, X. MindDrive: A Vision-Language-Action Model for Autonomous Driving via Online Reinforcement Learning. In *Computer Vision—ECCV 2026, Lecture Notes in Computer Science*; Springer Nature: Cham, Switzerland, 2026; Volume 17008, pp. 503–522. https://doi.org/10.1007/978-3-032-37026-6_28.
20. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
21. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.I.; Abbeel, P. High-Dimensional Continuous Control Using Generalized Advantage Estimation. In *Proceedings of the International Conference on Learning Representations (ICLR)*; ICLR: Appleton, WI, USA, 2016.
22. Dauner, D.; Hallgarten, M.; Li, T.; Weng, X.; Huang, Z.; Yang, Z.; Li, H.; Gilitschenski, I.; Ivanovic, B.; Pavone, M.; et al. NAVSIM: Data-Driven Non-Reactive Autonomous Vehicle Simulation and Benchmarking. *Adv. Neural Inf. Process. Syst. (NeurIPS) Datasets Benchmarks Track* **2024**, *37*, 28706–28719. https://doi.org/10.52202/079017-0902.
23. Chitta, K.; Prakash, A.; Geiger, A. NEAT: Neural Attention Fields for End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2021; pp. 15793–15803.
24. Chen, D.; Koltun, V.; Krähenbühl, P. Learning To Drive From a World on Rails. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2021; pp. 15590–15599.
25. Zhang, Z.; Liniger, A.; Dai, D.; Yu, F.; Van Gool, L. End-to-End Urban Driving by Imitating a Reinforcement Learning Coach. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2021; pp. 15222–15232.
26. Chen, D.; Krähenbühl, P. Learning From All Vehicles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2022; pp. 17222–17231.
27. Wu, P.; Jia, X.; Chen, L.; Yan, J.; Li, H.; Qiao, Y. Trajectory-Guided Control Prediction for End-to-End Autonomous Driving: A Simple Yet Strong Baseline. *Adv. Neural Inf. Process. Syst. (NeurIPS)* **2022**, *35*, 6119–6132. https://doi.org/10.52202/068431-0443.
28. Shao, H.; Wang, L.; Chen, R.; Li, H.; Liu, Y. Safety-Enhanced Autonomous Driving Using Interpretable Sensor Fusion Transformer. *Proc. Mach. Learn. Res.* **2023**, *205*, 726–737.
29. Jia, X.; Gao, Y.; Chen, L.; Yan, J.; Liu, P.L.; Li, H. DriveAdapter: Breaking the Coupling Barrier of Perception and Planning in End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2023; pp. 7953–7963.
30. Jia, X.; You, J.; Zhang, Z.; Yan, J. DriveTransformer: Unified Transformer for Scalable End-to-End Autonomous Driving. In *International Conference on Learning Representations (ICLR)*; ICLR: Appleton, WI, USA, 2025.
31. Song, Z.; Jia, C.; Liu, L.; Pan, H.; Zhang, Y.; Wang, J.; Zhang, X.; Xu, S.; Yang, L.; Luo, Y. Don't Shake the Wheel: Momentum-Aware Planning in End-to-End Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 22432–22441.
32. Li, Y.; Fan, L.; He, J.; Wang, Y.; Chen, Y.; Zhang, Z.; Tan, T. Enhancing End-to-End Autonomous Driving with Latent World Model. In *Proceedings of the International Conference on Learning Representations (ICLR)*; ICLR: Appleton, WI, USA, 2025.
33. Li, Y.; Wang, Y.; Liu, Y.; He, J.; Fan, L.; Zhang, Z. End-to-End Driving with Online Trajectory Evaluation via BEV World Model. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2025; pp. 27137–27146. https://doi.org/10.1109/ICCV51701.2025.02519.
34. Pan, C.; Yaman, B.; Nesti, T.; Mallik, A.; Allievi, A.G.; Velipasalar, S.; Ren, L. VLP: Vision Language Planning for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2024; pp. 14760–14769. https://doi.org/10.1109/CVPR52733.2024.01398.

35. Wang, S.; Yu, Z.; Jiang, X.; Lan, S.; Shi, M.; Chang, N.; Kautz, J.; Li, Y.; Alvarez, J.M. OmniDrive: A Holistic Vision-Language Dataset for Autonomous Driving with Counterfactual Reasoning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 22442–22452. https://doi.org/10.1109/CVPR52734.2025.02090.
36. Jiang, B.; Chen, S.; Liao, B.; Zhang, X.; Yin, W.; Zhang, Q.; Huang, C.; Liu, W.; Wang, X. Senna: Bridging Large Vision-Language Models and End-to-End Autonomous Driving. *Int. J. Comput. Vis.* **2026**, *134*, 407. https://doi.org/10.1007/s11263-026-03002-y.
37. Hwang, J.-J.; Xu, R.; Lin, H.; Hung, W.-C.; Ji, J.; Choi, K.; Huang, D.; He, T.; Covington, P.; Sapp, B.; et al. EMMA: End-to-End Multimodal Model for Autonomous Driving. Transactions on Machine Learning Research. 2025. Available online: https://openreview.net/forum?id=kH3t5lmOU8 (accessed on 28 October 2026).
38. Chen, X.; Huang, L.; Ma, T.; Fang, R.; Shi, S.; Li, H. SOLVE: Synergy of Language-Vision and End-to-End Networks for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 12068–12077. https://doi.org/10.1109/CVPR52734.2025.01127.
39. Hegde, D.; Yasarla, R.; Cai, H.; Han, S.; Bhattacharyya, A.; Mahajan, S.; Liu, L.; Garrepalli, R.; Patel, V.M.; Porikli, F. Distilling Multi-modal Large Language Models for Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*; IEEE: New York, NY, USA, 2025; pp. 27575–27585. https://doi.org/10.1109/CVPR52734.2025.02568.
40. Huang, Z.; Feng, C.; Yan, F.; Xiao, B.; Jie, Z.; Zhong, Y.; Liang, X.; Ma, L. RoboTron-Drive: All-in-One Large Multimodal Model for Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2025; pp. 8011–8021. https://doi.org/10.1109/ICCV51701.2025.00751.
41. Kong, F.; Li, Y.; Chen, W.; Min, C.; Li, Y.; Gao, Z.; Li, H.; Guo, Z.; Sun, H. VLR-Driver: Large Vision-Language-Reasoning Models for Embodied Autonomous Driving. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2025; pp. 26966–26976.
42. Zhou, X.; Liang, D.; Tu, S.; Chen, X.; Ding, Y.; Zhang, D.; Tan, F.; Zhao, H.; Bai, X. HERMES: A Unified Self-Driving World Model for Simultaneous 3D Scene Understanding and Generation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*; IEEE: New York, NY, USA, 2025; pp. 27817–27827. https://doi.org/10.1109/ICCV51701.2025.02582.
43. Li, Y.; Shang, S.; Liu, W.; Zhan, B.; Wang, H.; Wang, Y.; Chen, Y.; Wang, X.; An, Y.; Tang, C.; et al. DriveVLA-W0: World Models Amplify Data Scaling Law in Autonomous Driving. In *Proceedings of the International Conference on Learning Representations (ICLR)*; ICLR: Appleton, WI, USA, 2026.
44. Kim, M.J.; Pertsch, K.; Karamcheti, S.; Xiao, T.; Balakrishna, A.; Nair, S.; Rafailov, R.; Foster, E.P.; Sanketi, P.R.; Vuong, Q.; et al. OpenVLA: An Open-Source Vision-Language-Action Model. *Proc. Mach. Learn. Res.* **2025**, *270*, 2679–2713.
45. Xu, Z.; Zhang, Y.; Xie, E.; Zhao, Z.; Guo, Y.; Wong, K.-Y.K.; Li, Z.; Zhao, H. DriveGPT4: Interpretable End-to-End Autonomous Driving via Large Language Model. *IEEE Robot. Autom. Lett.* **2024**, *9*, 8186–8193. https://doi.org/10.1109/LRA.2024.3440097.
46. Li, Y.; Xiong, K.; Guo, X.; Li, F.; Yan, S.; Xu, G.; Zhou, L.; Chen, L.; Sun, H.; Wang, B.; et al. ReCogDrive: A Reinforced Cognitive Framework for End-to-End Autonomous Driving. In *Proceedings of the International Conference on Learning Representations (ICLR)*; ICLR: Appleton, WI, USA, 2026.
47. Huang, Z.; Sheng, Z.; Qu, Y.; You, J.; Chen, S. VLM-RL: A Unified Vision Language Models and Reinforcement Learning Framework for Safe Autonomous Driving. *Transp. Res. Part C Emerg. Technol.* **2025**, *180*, 105321. https://doi.org/10.1016/j.trc.2025.105321.
48. Rafailov, R.; Sharma, A.; Mitchell, E.; Manning, C.D.; Ermon, S.; Finn, C. Direct Preference Optimization: Your Language Model Is Secretly a Reward Model. *Adv. Neural Inf. Process. Syst. (NeurIPS)* **2023**, *36*, 53728–53741. https://doi.org/10.52202/075280-2338.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.