

Article

Data-Centric Autonomous Driving: A Data-Utilization Framework for Learning from Large-Scale Driving Logs

Yu Li ¹ and Wenhao Kang ^{2,*}

¹ Faculty of Mechanical Engineering and Transport Systems, Technical University of Berlin, 10623 Berlin, Germany; liyubertlin@163.com

² Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong 999077, China

* Correspondence: wenhao.kang@connect.polyu.hk

Abstract

Closed-loop autonomous driving increasingly combines model-centric architectural development with data-centric learning. These two directions are complementary: model-centric methods improve representation, prediction, planning, and control capabilities, whereas data-centric methods improve how training samples, supervision signals, and closed-loop failures are selected and reused. A key remaining challenge is therefore how to use heterogeneous driving logs more effectively to improve closed-loop policy learning. Existing methods have made certain progress in aspects such as representation learning, generative planning, and closed-loop evaluation. This paper proposes a data-centric closed-loop autonomous driving learning framework. Combining risk-aware data valuation, supervision-calibrated distillation, and counterfactual attribution-guided corrective replay, the framework adjusts the training distribution, calibrates sample-level supervision, and converts training-side failure trajectories into intervention-validated corrective signals. Closed-loop experiments on Bench2Drive demonstrate improved driving performance, while NAVSIM provides supplementary non-reactive cross-platform evaluation under the tested benchmark conditions. The experimental results show that data-centric optimization strategies can help reduce performance degradation under the evaluated perturbations and provide a feasible approach to high-risk sample management, supervision-quality calibration, and failure-informed corrective replay.

Keywords: closed-loop autonomous driving; data-centric learning; sample value estimation; supervision-calibrated distillation; failure attribution replay

Academic Editor: Felipe Jiménez

Received: 13 July 2026

Revised: 4 August 2026

Accepted: 6 August 2026

Published: 7 August 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](https://creativecommons.org/licenses/by/4.0/).

1. Introduction

How to extract truly valuable information from a large amount of driving records has become an important issue for improving the performance of autonomous driving [1]. The real road environment includes complicated vehicle movements, constantly changing traffic conditions, and dangerous emergencies, which help create an important foundation for the model of learning driving behaviors [2]. However, not all driving data has the same training value. How to screen important data, reasonably utilize different types of driving experiences, and continuously optimize the training process based on actual operation results have become the problems that data-centered autonomous driving research needs to solve [3].

The current research mainly focuses on the following directions. The first type of work focuses on the construction of training and evaluation infrastructure, providing support for the training of large-scale autonomous driving models, but it rarely discusses which data can lead to more effective learning outcomes [4]. There are also methods that emphasize future-state prediction, environment representation learning, and generative modeling to enhance the model's ability to understand complex scenarios. However, these methods lack further analysis of high-risk samples, long-tail scenarios, and the issue of insufficient supervision information [5,6]. Another type of research enhances decision-making capabilities by introducing language models and multimodal information, but pays insufficient attention to the differences between expert demonstrations and the model's actual execution in closed-loop learning, whether supervision signals are suitable for learning, and how failed experiences can be effectively utilized [7,8]. Overall, the problems that the existing methods have not fully addressed are: which data is worthy of focused learning, which demonstrations can truly help the model improve, and which failed cases need to be reused.

In response to the aforementioned issues, this paper proposes a closed-loop learning framework focused on data utilization. This method assesses the importance of different samples based on the scene risks and data characteristics, enabling the model to learn more valuable driving experiences. Subsequently, by comparing the differences between expert driving behaviors and model decisions, the influence of different samples in the training process is adjusted, making the supervisory information more suitable for model learning. Moreover, this paper further designs a failure experience replay mechanism to analyze the problems that occur during autonomous driving, identify the key scenarios that led to failure, and reuse these experiences for model training, thereby continuously improving the system performance.

Unlike methods that further expand the scale of perception or planning models, this paper focuses on how to establish more effective data utilization mechanisms, enabling the training process to fully exploit the information value in existing driving data. The research is centered around closed-loop decision learning and uses representative imitation learning and trajectory planning models as the basic framework. Although the proposed method has the potential to be extended to other models, this paper mainly evaluates its effects on closed-loop performance, safety, and data utilization under fixed benchmark configurations. The main contributions are as follows: (1) A risk-aware valuation mechanism combines safety risk, uncertainty, novelty, and distribution correction to prioritize informative samples while controlling sampling concentration. Unlike single-signal uncertainty, loss-based, or risk-only sampling, it integrates four independently calibrated factors with safeguards. (2) An asymmetry-calibrated distillation mechanism adjusts supervision according to demonstration learnability under expert–learner information differences. Unlike fixed or discrepancy-based distillation, its weights are independently supervised by learning progress and detached from policy optimization. (3) A failure attribution replay mechanism extracts intervention-validated corrective pairs from training-side failures. Unlike temporal, priority, or DAgger-style replay, it retains only interventions that reduce subsequent failure cost and excludes failed actions from supervision. The framework is evaluated through closed-loop, cross-platform, data-scale, scenario-specific, robustness, efficiency, and ablation analyses.

2. Related Work

2.1. Application Scenarios and Challenges

Data-centric research on autonomous driving focuses on how to utilize large-scale driving data to enhance closed-loop decision-making capabilities, covering directions such as decision learning, generative planning, multimodal interaction understanding,

and data-based model evaluation [9]. Related research is usually oriented towards complex driving tasks in real road environments, such as urban road driving, intersection interactions, and handling low-frequency high-risk scenarios, and uses real driving logs to support model training and continuous optimization [10]. The difficulties of these tasks lie in identifying the road environment and understanding the behaviors of other traffic participants, as well as in learning stable, safe, and commonly used driving strategies from continuously changing spatiotemporal information [11].

Currently, the data and evaluation platforms used in autonomous driving research mainly fall into two categories. One type of platform focuses on collecting large-scale real driving data to provide a foundation for tasks such as representation learning, future-state prediction, and supervised distillation [12]; the other type of platform uses a unified closed-loop test environment and evaluation standards to compare the performance of different methods in actual driving tasks.

The existing evaluation metrics typically cover task completion, safety, and the rationality of driving behavior, such as Success Rate, Driving Score, Collision Rate, route deviation, and trajectory error [13]. However, these metrics still struggle to fully reflect the performance of the model in real driving. For instance, open-loop prediction errors cannot capture the cumulative impact of continuous control processes, and a single Success Rate alone cannot explain the specific reasons for system failures. Therefore, further establishing a data mechanism that can assess sample value, adjust the supervision method, and utilize failure experience to optimize the training process remains a problem that needs to be addressed in current closed-loop autonomous driving research [14].

2.2. Overview of Mainstream Methods

The current studies mainly promote the development of closed-loop autonomous driving from three aspects. The first type of research focuses on how to establish a more comprehensive testing and evaluation system. By constructing a unified testing environment and evaluation standards, it helps researchers compare the performance of different methods [15]. This kind of work solves the problem of “how to evaluate autonomous driving systems”, but usually directly uses existing driving data for training, and rarely further considers the impact of different data on the learning effect of the model.

The second type of research focuses on how to enable the model to learn more from a large amount of driving records, including understanding environmental changes, predicting future situations, and constructing simulation environments that are closer to real roads [16]. These methods enhance the model’s ability to utilize data, but often assume that all training data have similar value, and pay insufficient attention to rare but important dangerous scenarios and the impact of different data qualities [17].

The third type attempts to utilize language information and various types of data to enhance the understanding and decision-making capabilities of autonomous driving systems [18]. These methods have improved the model’s ability to handle complex scenarios, but they mainly rely on expanding the model’s size to achieve performance improvements. When the training data is insufficient or there are differences between expert demonstrations and the actual model execution, merely enhancing the model’s capabilities is still difficult to solve the problem of limited learning effectiveness [19].

Overall, the existing studies have improved the evaluation methods, data utilization capabilities, and model decision-making abilities of the autonomous driving system. However, there is still a lack of sufficient research on a more fundamental issue during the training process: which data is truly worthy for the model to learn, how to adjust the training methods based on different data, and how to further improve the model by leveraging the failure experiences during the system operation.

2.3. Closest Related Work

The most relevant studies address three issues. First, closed-loop imitation learning has examined the difference between expert demonstrations and model execution [20]. Experts may use information unavailable to the deployed model, creating supervision asymmetry. The present work therefore adjusts supervision according to sample-level learning progress rather than treating all expert labels as equally learnable.

Second, driving-log research uses future-state prediction, representation learning, and large-scale pre-training to extract patterns from unlabeled data [21]. However, these methods mainly improve representation capability and seldom distinguish the training value of individual scenarios. The proposed method instead combines risk, uncertainty, novelty, and distribution correction to construct a safeguarded sampling distribution.

Third, multimodal and generative methods integrate language, context, and generative planning to improve decision-making [22]. However, they provide limited mechanisms for determining which closed-loop failures should be reused. The proposed replay module retains only corrective interventions that reduce subsequent failure cost, rather than relying on temporal proximity or loss priority.

Model-centric methods, including model-based control, remain important because they strengthen representation, prediction, planning, and control capabilities. Recent studies have developed static output-feedback LQR control for energy-efficient predecessor-leader platooning [23], fault-tolerant output-feedback control under measurement noise and road disturbances [24], and robust adaptive platoon control with disturbance compensation [25]. These methods use explicit dynamics to provide stability and robustness guarantees. By contrast, data-centric methods improve how samples, supervision signals, and failure experiences are used, but their effectiveness still depends on the underlying model and data coverage. The present work therefore does not replace low-level model-based control; it improves data selection, supervision calibration, and failure replay at the policy-learning level. The two directions are complementary and can be integrated within a hierarchical driving system.

In summary, existing studies have advanced supervision, driving-log utilization, multimodal reasoning, and model-based control. Model-centric development strengthens system capability, whereas data-centric learning improves the use of available data and supervision. Data-centric closed-loop learning still requires mechanisms to identify valuable samples, learnable demonstrations, and reusable failures.

2.4. Summary and Research Gaps

Previous studies have made significant progress in aspects such as closed-loop evaluation, future modeling, world models, and semantic-enhanced decision-making. However, there is still a lack of sufficient exploration regarding the value differences in training data itself, whether the supervisory information is suitable for model learning, and how to utilize failed experiences to improve the training process [26]. Existing research indicates that large-scale driving logs can effectively support model pre-training, generative planning, and complex scene reasoning. It also reveals that the inconsistency between expert demonstrations and model actual observations will affect closed-loop performance. Nevertheless, these issues are usually discussed separately and have not formed a unified data-utilization framework. Most existing methods treat training data as fixed inputs and pay limited attention to actively evaluating and adjusting the data for model learning.

Based on the above analysis, this work focuses on three interrelated research gaps. First, existing methods lack an explicit mapping from sample value to the training distribution, making it difficult to determine which data should be prioritized. Second, they lack sample-level modeling of supervision learnability, making it unclear which expert demonstrations are learnable for the current learner. Third, they lack a stable closed-loop

mechanism connecting failure feedback to training-set reconstruction, making it difficult to determine which failure experiences should be reused.

Therefore, the goal of this work is neither to replace model-based low-level control nor to design more complex perception or planning architectures, but to establish a more complete data-centric closed loop: identifying high-value samples, determining learnable demonstrations, and refining the training distribution through closed-loop failure feedback.

3. Methodology

3.1. Problem Formulation

In data-centric autonomous driving, the task considered in this work is to learn a decision function from benchmark driving logs that can generate safe and effective behaviors under closed-loop evaluation. Specifically, we formulate this problem as imitation-based policy learning, where the model learns from demonstrated behaviors in historical data and is ultimately evaluated based on its performance in closed-loop interactions.

Let a single sample be denoted as $x \in \mathcal{X}$, which consists of multimodal historical observations and contextual information prior to time t , defined as: $x = \{o_{t-H:t}, m_t, n_t\}$. Here, $o_{t-H:t}$ represents the historical observation sequence, including multi-view images, ego-vehicle states, and the motion of surrounding traffic participants; m_t denotes environmental priors such as road topology; and n_t represents navigation goals or high-level constraints. The output is denoted as $y \in \mathcal{Y}$, representing the future decision response $y = \{\tau_{t+1:t+T}, a_t\}$, where $\tau_{t+1:t+T}$ is the predicted future trajectory, and a_t denotes the current control action or its equivalent representation.

Assume that all sample pairs (x, y) come from an unknown joint distribution $P(X, Y)$. The training set is represented as $D_{\text{train}} = \{(x_i, y_i)\}_{i=1}^N$, where each sample corresponds to a temporal and spatial segment in the driving log. The validation set and test set are used to evaluate the generalization ability of the model. Although the samples follow the benchmark data division, their contributions to closed-loop performance differ: some correspond to routine driving, whereas others contain high-risk interactions, low-frequency conditions, or distribution changes. Therefore, this task not only requires learning the input–output relationship $P(Y | X)$ in general cases, but also needs to ensure that the model still has stable decision-making capabilities in critical risk scenarios.

Let the decision function be $f_\theta: \mathcal{X} \rightarrow \mathcal{Y}$. The basic objective of model training is to minimize the prediction error under the true data distribution:

$$\theta^* = \arg \min_{\theta} \mathbb{E}_{(x,y) \sim P(X,Y)} [\mathcal{L}(f_\theta(x), y)], \quad (1)$$

where θ^* denotes the optimal parameter set, and $\mathbb{E}_{(x,y) \sim P(X,Y)} [\cdot]$ denotes the expectation over the joint data distribution. $\mathcal{L}(f_\theta(x), y)$ is the sample-wise imitation loss that measures the discrepancy between the decision response predicted by $f_\theta(x)$ and the reference demonstration y . In the implementation, this basic loss is instantiated by the trajectory and action losses defined in Equations (11) and (12).

Furthermore, since the autonomous driving system needs to operate in a continuous interactive environment, its ultimate goal is not only to improve the accuracy of single-step predictions, but also to achieve higher expected utility under the state distribution $P(\mathcal{S})$:

$$\theta^* = \max_{\theta} \mathbb{E}_{s \sim P(\mathcal{S})} [\mathcal{U}(s)], \quad (2)$$

where $s \in \mathcal{S}$ denotes a closed-loop driving state, d^{θ} is the state-visitation distribution induced by the learned policy f_θ , and $\mathcal{U}(s)$ denotes the benchmark utility reflecting task completion, safety, and driving-behavior quality. Accordingly, the problem is defined as

learning a decision function from the available training distribution that fits historical demonstrations and improves closed-loop utility under the evaluated benchmark conditions.

3.2. Overall Framework

The overall framework of this work, illustrated in Figure 1, is a data-centric learning system for closed-loop autonomous driving. Large-scale driving logs are first collected into an input data pool, and then sequentially processed through sample valuation, supervision calibration, policy learning, closed-loop evaluation, and failure feedback, forming an iterative loop of “data selection–policy training–closed-loop validation–data redistribution”. The framework consists of three core modules: the Risk-Aware Data Valuation Module, the Asymmetry-Calibrated Distillation Module, and the Failure Attribution Replay Module.

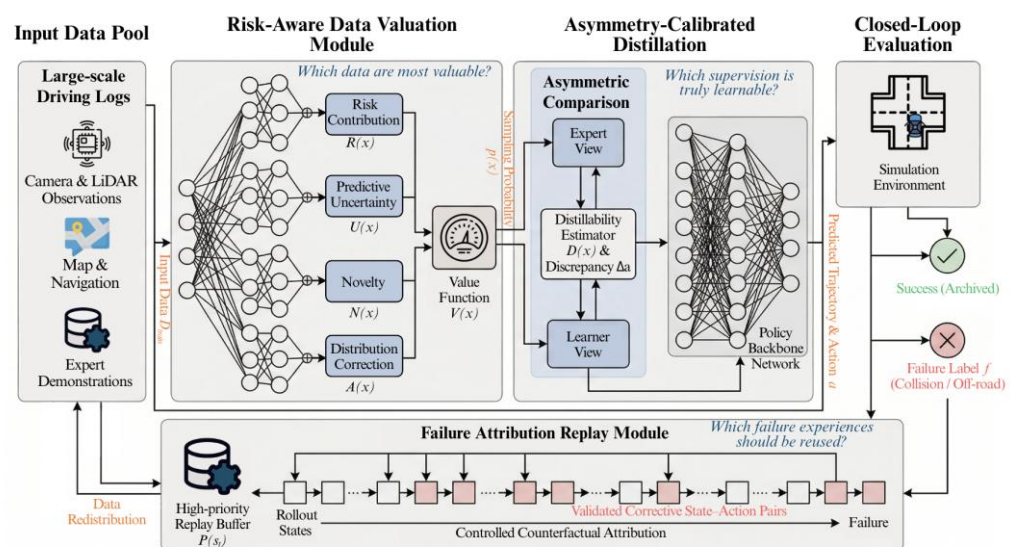


Figure 1. Overall framework of the proposed data-centric closed-loop learning system.

Specifically, the framework takes as input multimodal observations, map and navigation information, and expert demonstrations from benchmark driving logs, and outputs trajectory planning and control responses over future decision horizons. Firstly, the Risk-Aware Data Valuation Module is used to measure the importance of different samples during the training process. This module assigns greater sampling probability and training weight to samples associated with safety risk, low observed frequency, feature novelty, or distribution variation. Subsequently, the filtered samples are input into the Asymmetry-Calibrated Distillation Module. This module takes into account the differences between expert demonstrations and learners in terms of observed information and available information, and dynamically adjusts the supervision intensity according to the characteristics of the samples to alleviate the mismatch between the expert data and the model learning process.

Subsequently, the calibrated samples are fed into the policy backbone network to learn trajectories and actions, and are evaluated in a closed-loop environment. Failure trajectories generated only from training-side closed-loop rollouts are processed by the Failure Attribution Replay Module. Controlled counterfactual interventions identify historical states whose replacement actions reduce failure cost, and the validated corrective pairs are added to the replay buffer. Validation, Dev10, NAVSIM, and final evaluation failures are excluded from model updating. In this way, the three modules address which data

should be prioritized, which demonstrations are learnable, and which intervention-validated corrective experiences should be reused.

3.3. Module Details

3.3.1. Risk-Aware Data Valuation Module

Although all samples are drawn from benchmark driving logs and simulation records, their contributions to closed-loop performance differ. Uniform sampling can over-emphasize routine cases while underusing high-risk, long-tail, and distribution-shifted scenarios. This module therefore determines which samples should receive greater training priority.

Figure 2 illustrates how the Risk-Aware Data Valuation Module converts four independently calibrated branch scores into a safeguarded sampling distribution. Specifically, the four branches share the latent feature representation $\phi(x)$ and output the risk score $R(x)$, uncertainty score $U(x)$, novelty score $N(x)$, and distribution-correction score $A(x)$. Each branch is calibrated using an independent target normalized to $[0, 1]$. For sample x_i , the risk target is its proxy safety label y_i^{risk} . The uncertainty target is the variance over M stochastic forward passes, calculated from the deviations of the predictions y_i^m from their mean $\bar{y}_i$. The novelty target is the average Euclidean distance between $\phi(x_i)$ and its k nearest neighbors in the training-feature space, excluding x_i itself. The distribution-correction target is $(n_{c_i} + \varepsilon)^{-1}$, where n_{c_i} is the number of training samples belonging to scenario category c_i , and $\varepsilon > 0$ is a small numerical constant that prevents division by zero.

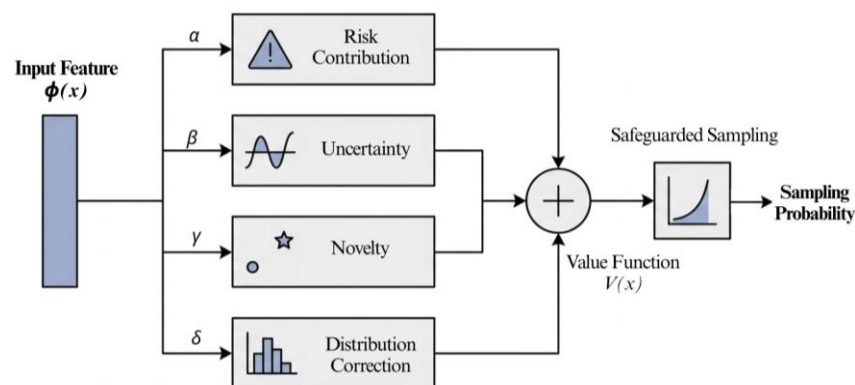


Figure 2. Architecture of the Risk-Aware Data Valuation Module. The independently calibrated branches are fused and mapped to a safeguarded sampling distribution.

Each raw target z_i is normalized as $\text{clip}[(z_i - z_{\min}) / (z_{\max} - z_{\min} + \varepsilon), 0, 1]$. The normalization statistics are computed only from the training split and fixed for validation and evaluation. All targets are detached from the computational graph, and the four branches are independently trained against their corresponding targets using the calibration loss in Equation (17). Thus, the scores retain distinct meanings rather than being jointly supervised by risk labels. Their fused value is

$$V(x) = \alpha R(x) + \beta U(x) + \gamma N(x) + \delta A(x), \quad (3)$$

where α, β, γ , and δ are weighting coefficients.

To prevent a few high-value samples from dominating training, the centered value score is converted into a clipped weight and mixed with uniform sampling:

$$w^v(x) = \text{clip}\left[\exp\left(\frac{V(x) - \bar{V}}{\tau}\right), w_{\min}, w_{\max}\right], \quad p(x) = (1 - \epsilon_u) \frac{w^v(x)}{\sum_{x' \in \mathcal{D}_{\text{train}}} w^v(x')} + \frac{\epsilon_u}{|\mathcal{D}_{\text{train}}|}, \quad (4)$$

where $w_v(x)$ denotes the clipped valuation weight assigned to sample x , $p(x)$ denotes its resulting sampling probability, and $\bar{V}$ is the mean value score over D_{train} . The parameter $\tau > 0$ is the sampling temperature, $w_{\min}$ and $w_{\max}$ are the lower and upper clipping bounds, $\epsilon_u \in [0, 1]$ controls the proportion of uniform mixing, and $|D_{\text{train}}|$ denotes the number of training samples. Clipping limits the maximum priority ratio, while uniform mixing preserves a nonzero sampling probability for every sample. Partial importance correction is applied according to Equation (19). The resulting scores adjust both sampling and loss weighting. This variance- and value-based prioritization is related to active sample emphasis, while the proposed formulation additionally integrates risk, novelty, distribution correction, priority clipping, and uniform mixing.

3.3.2. Asymmetry-Calibrated Distillation Module

Sample importance does not necessarily imply learnability. The expert may use privileged information available only during training, whereas the learner receives deployment-time observations. Specifically, x^{learner} contains only deployment-available observations, ego-vehicle states, map information, and navigation inputs, whereas x^{expert} additionally encodes the ground-truth future trajectory or control target available only during training. Applying equal supervision to all demonstrations may therefore introduce bias. This module estimates which expert demonstrations are learnable by the current policy.

Figure 3 illustrates the information flow of the Asymmetry-Calibrated Distillation Module, including the expert and learner encoders, action-discrepancy calculation, distillability estimation, and the generation of normalized supervision weights. The expert and learner representations are

$$z_e = E_e(x^{\text{expert}}), z_l = E_l(x^{\text{learner}}). \quad (5)$$

where E_e and E_l denote the expert-view and learner-view encoders, respectively, and z_e and z_l are their corresponding latent representations. The expert representation is used only to estimate the distillation weight and is neither provided to the policy input nor used during inference or evaluation, thereby preventing information leakage. The distillability estimator is defined as

$$D(x) = \sigma(W_d[z_e; z_l; \Delta a] + b_d), \quad (6)$$

where Δa is the expert–learner action discrepancy, $[\cdot; \cdot]$ denotes concatenation, and W_d, b_d are learnable parameters.

Rather than being optimized through the weighted policy loss, $D(x)$ is independently supervised using the learning-progress target in Equation (22). Its normalized and clipped form $\omega(x)$, defined in Equation (23), is detached before policy optimization. This prevents trivial near-zero scores for difficult samples.

The conceptual objective is

$$\mathcal{L}_{\text{distill}} = \sum_x \text{sg}[\omega(x)] l(\pi(x^{\text{learner}}), a^{\text{expert}}) + \lambda_D \mathcal{L}_D, \quad (7)$$

where π denotes the learner policy, a^{expert} is the expert action, and $l(\pi(x^{\text{learner}}), a^{\text{expert}})$ is the sample-wise imitation loss between the learner output and the expert action. The operator $\text{sg}[\cdot]$ denotes the stop-gradient, $\mathcal{L}_D$ is the independently optimized distillability-estimation loss, and λ_D controls its contribution to the conceptual objective. The complete importance-corrected formulation is given in Equations (24) and (25). Lower and upper clipping bounds retain difficult samples while preventing excessive supervision by a small subset. Demonstrations showing greater learning progress receive stronger supervision, whereas substantial expert–learner mismatches receive lower but nonzero weights.

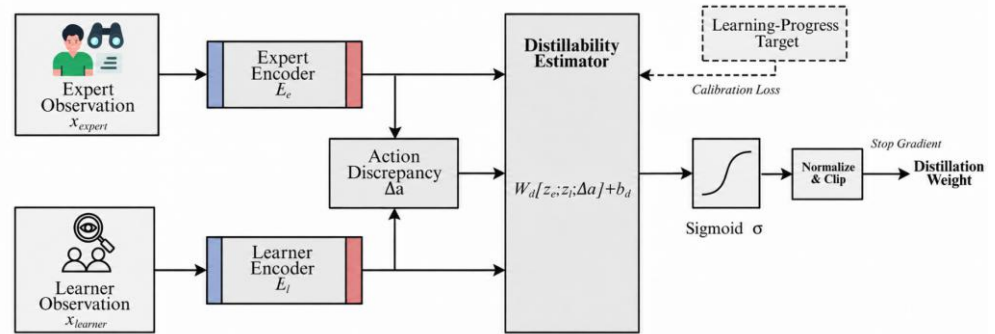


Figure 3. Architecture of the Asymmetry-Calibrated Distillation Module. Learning-progress supervision trains the estimator, while normalized detached weights control policy learning.

3.3.3. Failure Attribution Replay Module

This module converts training-side failures into corrective supervision. Collisions, off-road deviations, and deadlocks indicate poor performance but do not identify which earlier decisions should be corrected. Because failures may accumulate across several states, historical state–action pairs are evaluated by their intervention effects rather than temporal proximity. Figure 4 illustrates how training-side failure trajectories are evaluated through controlled counterfactual interventions and converted into validated corrective samples for prioritized replay.

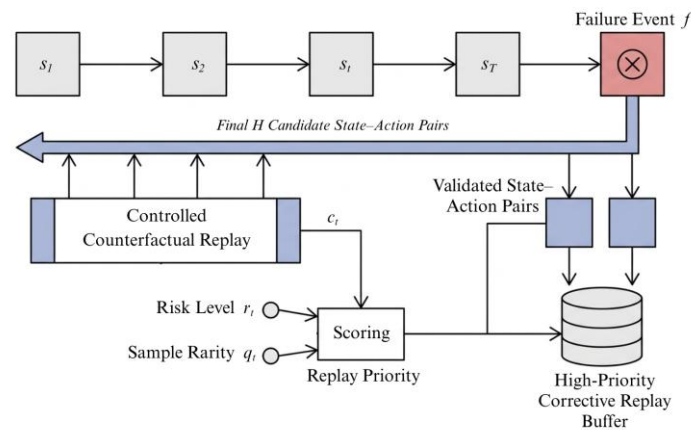


Figure 4. Controlled counterfactual attribution and corrective failure replay. Only interventions that reduce failure cost are stored as corrective targets.

Given a failed trajectory $\mathcal{T} = \{(s_t, a_t)\}_{t=1}^T$, the final H pairs are treated as candidates. The simulator is restored to s_t , and the recorded action is replaced for one step by the expert or reference-controller action a_t^e . The original policy then continues under the same route, traffic configuration, and random seed. The counterfactual contribution is

$$c_t = \text{clip} \left(\frac{F(\mathcal{T}) - F(\text{Rollout}(s_t, \text{do}(a_t = a_t^e), \pi, \xi))}{F(\mathcal{T}) + \varepsilon}, 0, 1 \right), \quad (8)$$

where $F(\cdot)$ is the normalized failure cost, π is the original policy, ξ denotes fixed simulator conditions, and ε prevents numerical instability. A high c_t indicates that replacing the decision reduces subsequent failure cost. Hence, c_t represents a local intervention effect rather than temporal correlation and is computed offline.

The intervention action a_t^e is stored as the corrective target only when it reduces failure cost. Failed policy actions are therefore excluded from supervision. The unnormalized replay priority is

$$\bar{P}(s_t) = \eta_1 \text{sg}[c_t] + \eta_2 r_t + \eta_3 q_t, \quad (9)$$

where $r_t, q_t \in [0, 1]$ are normalized risk and rarity scores, and η_1, η_2 , and η_3 are weighting coefficients. The priority is subsequently normalized and clipped to $[0.5, 2.0]$ according to Equation (26). Replay occupies at most 25% of each minibatch, and each scenario category is limited to 10% of the buffer, preventing repeated failures from dominating training.

The buffer is refreshed only using training-side rollouts under the schedule in Section 4.1. Validation, Dev10, NAVSIM, and final evaluation failures are excluded. This mechanism reuses intervention-validated corrective actions without assuming that states closest to the final failure are the most influential. Its priority normalization and importance-correction strategy follows the general rationale of prioritized replay [30], but the proposed buffer retains only corrective actions validated through controlled interventions.

3.4. Objective Function and Optimization

The framework jointly optimizes data valuation, supervision-calibrated distillation, and failure attribution replay. Let Θ denote the learnable parameters of the policy backbone, data valuation network, and distillability estimator. The counterfactual contribution used for replay is computed offline and excluded from Θ . The overall objective is

$$\mathcal{J} = \mathcal{L}_{\text{traj}} + \mu \mathcal{L}_{\text{act}} + \lambda_1 \mathcal{L}_{\text{value}} + \lambda_2 \mathcal{L}_{\text{distill}} + \lambda_3 \mathcal{L}_{\text{replay}}, \quad (10)$$

where $\mathcal{L}_{\text{traj}}$, $\mathcal{L}_{\text{act}}$, $\mathcal{L}_{\text{value}}$, $\mathcal{L}_{\text{distill}}$, and $\mathcal{L}_{\text{replay}}$ denote trajectory, action, valuation, distillation, and replay losses, respectively. The coefficients $\mu, \lambda_1, \lambda_2$, and λ_3 are selected from the validation set.

Let the predicted and reference trajectories of the sample i be $\hat{\tau}_i, \tau_i \in \mathbb{R}^{T \times 2}$, where T is the prediction horizon and the second dimension represents the two-dimensional waypoint coordinates. The sample-level trajectory loss is given in Equation (11), where $\|\cdot\|_1$ denotes the L_1 norm. Let $\hat{\mathbf{a}}_i, \mathbf{a}_i \in \mathbb{R}^{d_a}$ denote the predicted and reference action vectors, respectively, where d_a is the action-space dimension. Equation (12) uses $\|\cdot\|_2^2$, the squared L_2 norm, to measure their discrepancy. The sample-level trajectory loss is

$$\ell_{\text{traj},i} = \|\hat{\tau}_i - \tau_i\|_1. \quad (11)$$

Let $\hat{\mathbf{a}}_i, \mathbf{a}_i \in \mathbb{R}^{d_a}$ denote the predicted and reference actions. The sample-level action loss is

$$\ell_{\text{act},i} = \|\hat{\mathbf{a}}_i - \mathbf{a}_i\|_2^2. \quad (12)$$

For data valuation, let $R_i, U_i, N_i, A_i \in \mathbb{R}$ denote the learned branch outputs for risk, uncertainty, novelty, and distribution correction, respectively. Their fused learned value is denoted by V_i in Equation (13). Let $\hat{R}_i, \hat{U}_i, \hat{N}_i, \hat{A}_i \in [0, 1]$ denote the corresponding detached supervision targets constructed from the proxy safety label, stochastic-prediction variance, k -nearest-neighbor feature distance, and inverse scenario frequency. Their fused target value is denoted by $\hat{V}_i$ in Equation (14). Thus, variables without hats represent learned predictions, whereas variables with hats represent fixed supervision targets. Their combined value is

$$V_i = \alpha R_i + \beta U_i + \gamma N_i + \delta A_i. \quad (13)$$

Let $\tilde{R}_i, \tilde{U}_i, \tilde{N}_i, \tilde{A}_i \in [0, 1]$ denote the independent targets defined in Section 3.3.1 from proxy safety labels, stochastic prediction variance, k -nearest-neighbor feature distance, and inverse scenario frequency. Their fused target is

$$\tilde{V}_i = \alpha \tilde{R}_i + \beta \tilde{U}_i + \gamma \tilde{N}_i + \delta \tilde{A}_i. \quad (14)$$

To avoid excessive concentration on a few samples, the scores are centered, clipped, and mixed with uniform sampling:

$$w_i^v = \text{clip} \left[\exp \left(\frac{V_i - \bar{V}}{\tau} \right), w_{\min}, w_{\max} \right], \bar{V} = \frac{1}{N} \sum_{j=1}^N V_j, \pi_i^v = (1 - \epsilon_u) \frac{w_i^v}{\sum_{j=1}^N w_j^v} + \frac{\epsilon_u}{N}, \quad (15)$$

where τ is the sampling temperature, $w_{\min} = 0.5$, $w_{\max} = 2.0$, and $\epsilon_u = 0.10$. The uniform term preserves a nonzero probability for every sample, while clipping limits the priority ratio to 4:1.

The same mapping is applied to the target values:

$$\tilde{w}_i^v = \text{clip} \left[\exp \left(\frac{\tilde{V}_i - \bar{\tilde{V}}}{\tau} \right), w_{\min}, w_{\max} \right], \bar{\tilde{V}} = \frac{1}{N} \sum_{j=1}^N \tilde{V}_j, \tilde{\pi}_i^v = (1 - \epsilon_u) \frac{\tilde{w}_i^v}{\sum_{j=1}^N \tilde{w}_j^v} + \frac{\epsilon_u}{N}. \quad (16)$$

The branch calibration loss is

$$\mathcal{L}_{\text{branch}} = \frac{1}{N} \sum_{i=1}^N [(R_i - \tilde{R}_i)^2 + (U_i - \tilde{U}_i)^2 + (N_i - \tilde{N}_i)^2 + (A_i - \tilde{A}_i)^2]. \quad (17)$$

The valuation loss is

$$\mathcal{L}_{\text{value}} = - \sum_{i=1}^N \tilde{\pi}_i^v \log \pi_i^v + \lambda_b \mathcal{L}_{\text{branch}}, \quad (18)$$

where λ_b controls branch calibration. The distribution term aligns the learned and target sampling distributions, while the calibration term preserves the meaning of the four branches. Thus, uncertainty, novelty, and distribution correction are not supervised by risk labels.

To reduce non-uniform sampling bias without excessive gradient variance, the partial importance-sampling coefficient is

$$\kappa_i = \text{sg} \left[\text{clip}_{[0.5, 2.0]} \left(\frac{(N\pi_i^v)^{-\beta_{\text{IS}}}}{\frac{1}{N} \sum_{j=1}^N (N\pi_j^v)^{-\beta_{\text{IS}}}}, 0.5, 2.0 \right) \right], \quad (19)$$

where κ_i is the partial importance-sampling correction coefficient applied to the sample i , $\text{sg}[\cdot]$ denotes the stop-gradient, and $\beta_{\text{IS}} = 0.5$ is the importance-correction exponent. The normalization in Equation (19) keeps the mean correction coefficient near one, while clipping it to $[0.5, 2.0]$ limits gradient variance. The aggregate policy losses are defined in Equation (20). Sampling concentration is monitored using the normalized effective sample size N_{eff}/N , where $N_{\text{eff}} = (\sum_i (\pi_i^v)^2)^{-1}$ and N is the number of training samples. This quantity is used only as a diagnostic and does not enter the optimization objective. The aggregate policy losses are

$$\mathcal{L}_{\text{traj}} = \frac{1}{N} \sum_{i=1}^N \kappa_i \ell_{\text{traj},i}, \mathcal{L}_{\text{act}} = \frac{1}{N} \sum_{i=1}^N \kappa_i \ell_{\text{act},i}. \quad (20)$$

Sampling concentration is monitored using $\frac{N_{\text{eff}}}{N} = \frac{1}{N \sum_{i=1}^N (\pi_i^v)^2}$, which is used only as a diagnostic.

For supervision-calibrated distillation, let $z_i^e, z_i^l \in \mathbb{R}^{d_z}$ denote the expert-view and learner-view representations, where d_z is the latent-representation dimension, and let $\Delta a_i \in \mathbb{R}^{d_a}$ denote their action discrepancy. Distillability is defined in Equation (21), where $\sigma(\cdot)$ is the sigmoid function and W_d and b_d are learnable parameters. Distillability is defined as

$$D_i^{(t)} = \sigma(W_d[z_i^e; z_i^l; \Delta a_i] + b_d), \quad (21)$$

where W_d and b_d are learnable parameters.

To prevent $D_i^{(t)}$ from collapsing towards zero, it is supervised by sample-level learning progress. Let $\bar{\ell}_i^{(t)}$ denote the exponential moving average of the unweighted imitation loss at stage t . The target is

$$\hat{D}_i^{(t)} = \sigma \left(\frac{\bar{\ell}_i^{(t-1)} - \bar{\ell}_i^{(t)}}{T_D} \right), \quad (22)$$

where T_D is a temperature parameter, and the target is detached. The predicted score is normalized and clipped as

$$\omega_i^{(t)} = \text{sg} \left[\text{clip} \left(\frac{D_i^{(t)}}{\frac{1}{N} \sum_{j=1}^N D_j^{(t)} + \varepsilon}, \omega_{\min}, \omega_{\max} \right) \right]. \quad (23)$$

The lower bound retains difficult samples, while normalization prevents global collapse. The distillation loss is

$$\mathcal{L}_{\text{distill}} = \frac{1}{N} \sum_{i=1}^N \kappa_i \omega_i^{(t)} \ell(\pi(\mathbf{x}_i^l), \mathbf{a}_i^e) + \lambda_D \mathcal{L}_D, \quad (24)$$

where

$$\mathcal{L}_D = -\frac{1}{N} \sum_{i=1}^N [\widehat{D}_i^{(t)} \log D_i^{(t)} + (1 - \widehat{D}_i^{(t)}) \log(1 - D_i^{(t)})]. \quad (25)$$

Here, x_i^l is the learner input, a_i^e is the expert action, and $l(\cdot)$ is the L_1 imitation loss. In Equation (25), $\widehat{D}_i^t$ is the detached learning-progress target, and D_i^t is the predicted distillability score; therefore, $\mathcal{L}_D$ is their binary cross-entropy calibration loss. The estimator is optimized only through $\mathcal{L}_D$, whereas the policy uses detached κ_i and $\omega_i^{(t)}$. It therefore cannot reduce the policy loss by shrinking all distillability scores.

For failure replay, only failures generated from training-side closed-loop rollouts are reused. Let $\mathcal{B}_f$ be the replay buffer, $c_t \in [0, 1]$ the offline counterfactual contribution from Equation (8), and $r_t, q_t \in \mathbb{R}$ the risk and rarity scores. Each retained entry is $(\mathbf{x}_t^l, \mathbf{y}_t^{\text{cf}}, c_t, r_t, q_t)$, where $\mathbf{y}_t^{\text{cf}} = \mathbf{a}_t^e$ is the expert or reference-controller action validated by the intervention in Equation (8).

To prevent a few failures from dominating replay, the priority is normalized and clipped:

$$\bar{P}_t = \eta_1 \text{sg}[c_t] + \eta_2 r_t + \eta_3 q_t, \quad P_t = \text{clip} \left(\frac{\bar{P}_t}{\frac{1}{|\mathcal{B}_f|} \sum_{k \in \mathcal{B}_f} \bar{P}_k + \varepsilon}, 0.5, 2.0 \right), \quad (26)$$

where η_1, η_2 , and η_3 are weighting coefficients. The replay loss is

$$\mathcal{L}_{\text{replay}} = \frac{1}{|\mathcal{B}_f|} \sum_{t \in \mathcal{B}_f} P_t \ell(\hat{\mathbf{y}}_t, \mathbf{y}_t^{\text{cf}}). \quad (27)$$

Only interventions with $c_t > 0$ are retained. The original failed action a_t is excluded from supervision, and the replay loss updates the policy without modifying c_t .

Replay samples constitute at most 25% of each minibatch, and each scenario category is limited to 10% of the 20,000-entry buffer. When full, the lowest-priority samples are removed. These constraints preserve ordinary samples, maintain scenario diversity, and reduce overfitting to repeated failures.

AdamW is used for optimization. As detailed in Section 4.1, training begins with uniform policy learning, followed by alternating updates of the policy, valuation network, and distillability estimator. During final fine-tuning, the auxiliary modules and replay buffer are frozen. Sample values and distillability weights are refreshed once per epoch, whereas the replay buffer is refreshed every five epochs. All hyperparameters are selected on the validation set and fixed before final evaluation.

Accordingly, Equations (13)–(20) define safeguarded data valuation and importance-corrected policy learning, Equations (21)–(25) define supervision-calibrated distillation, and Equations (26) and (27) define counterfactual attribution-guided failure replay.

4. Dataset-Based Simulation Results

4.1. Dataset and Simulation Setup

To evaluate the proposed method under standardized driving scenarios, dataset-based experiments and simulation evaluations were conducted using Bench2Drive and NAVSIM (Table 1). Bench2Drive experiments used the official CARLA-based benchmark code, whereas NAVSIM evaluation used the official NAVSIM development toolkit and its

non-reactive data-driven simulation protocol. Bench2Drive supports the main closed-loop simulation comparisons and method analyses, whereas NAVSIM provides supplementary data-driven non-reactive cross-platform evaluation and is not interpreted as reactive closed-loop evidence. The two benchmarks were selected because they provide complementary evaluation settings: Bench2Drive supports standardized reactive closed-loop testing across diverse routes and conditions, while NAVSIM enables independent cross-platform evaluation on recorded driving scenes. NAVSIM was used only for evaluation and did not participate in training, hyperparameter selection, or replay-buffer construction.

Bench2Drive provides Mini, Base, and Full training configurations, with the Full set containing tens of thousands of driving videos. It also includes 220 closed-loop routes and the Dev10 environment for rapid evaluation. NAVSIM uses the Extended PDM Score as its primary metric. Success Rate measures completed routes, while Driving Score combines route progress with penalties for unsafe or noncompliant behavior. Collision Rate is the proportion of evaluated routes containing collisions. EPDMS jointly reflects driving progress, time-to-collision, lane keeping, trajectory comfort, and traffic-rule compliance, providing physical meanings related to longitudinal progress, lateral control, safety margin, acceleration, and jerk. Raw velocity- and lateral-tracking errors were not reported separately because the evaluated backbones use different control and trajectory representations. All dataset-based experiments and simulation evaluations followed the official data divisions and processing protocols. Bench2Drive training data were used for policy learning, data valuation, and distillation, while replay samples were generated only from closed-loop rollouts on training-side scenarios. The validation split was used for hyperparameter selection and early stopping. Dev10 was used only for diagnostic and ablation evaluation, and its failures were never used for replay or parameter updates. After the configuration was fixed, the model was frozen and evaluated on the reported Bench2Drive routes and NAVSIM. No failure from Dev10, NAVSIM, or final evaluation routes was fed back into training.

Table 1. Dataset overview.

Dataset	Primary Use	Task Type	Data Modalities	Data Scale (Train/Val/Test)	Key Characteristics/Target Variables	Source
Bench2Drive	Training and replay generation from training-side scenarios; Dev10 for diagnostic ablations; reported routes for frozen-model evaluation	Reactive closed-loop simulation in CARLA	Multimodal driving logs, maps, and navigation information	Official Mini/Base/Full splits; 220 closed-loop routes; Dev10 for fast ablation	Emphasizes standardized closed-loop benchmarks for evaluating overall driving performance and safety	https://github.com/Thinklab-SJTU/Bench2Drive
NAVSIM	Evaluation-only cross-platform validation	Non-reactive data-driven simulation	Scenario-driven logs and evaluation states	Defined by official protocol	Emphasizes a balance between open-loop efficiency and closed-loop robustness; core metric is Extended PDM Score	https://github.com/autonomousvision/navsim

Note: Failure replay samples were generated exclusively from Bench2Drive training-side scenarios. Dev10, NAVSIM, and reported final Bench2Drive routes were excluded from replay-buffer construction and parameter updates.

Experiments were conducted on two workstations equipped with NVIDIA RTX 4090 GPUs with 24 GB of memory each. One workstation used an AMD Ryzen 9 7950X processor, and the other used an Intel Core i9-14900KF processor. Both were equipped with 128 GB RAM and a 2 TB SSD. The software environment included Ubuntu 22.04, Python 3.10, PyTorch 2.1, and CUDA 11.8. Under the matched TCP configuration, one complete 80-epoch training run of the proposed method required 43.2 h, compared with 38.6 h for TCP. The reported time includes auxiliary-network updates and replay-buffer refreshes but excludes final benchmark evaluation. All methods used identical data splits, evaluation routes, and simulator seeds. TCP, UniAD, and GenAD retained their native architectures, task losses, optimizers, and training schedules; within each backbone pair, the baseline and “+Ours” variants used the same backbone-specific procedure. The following 80-epoch AdamW configuration was used for the matched TCP-based data-utilization comparisons. AdamW training lasted 80 epochs with a batch size of 32, a learning rate of 1×10^{-4} , and a weight decay of 1×10^{-4} . The objective was $L_{\text{total}} = L_{\text{traj}} + \mu L_{\text{act}} + \lambda_1 L_{\text{value}} + \lambda_2 L_{\text{distill}} + \lambda_3 L_{\text{replay}}$, where $\mu = 0.5$, $\lambda_1 = 0.2$, $\lambda_2 = 0.3$, and $\lambda_3 = 0.2$. The valuation targets for R_i , U_i , N_i , and A_i were derived from proxy safety labels, variance over $M = 5$ stochastic passes, distance to the $k = 10$ nearest training features, and inverse scenario frequency. Targets were normalized using training statistics and detached during optimization. The fusion coefficients were $\alpha = 0.35$, $\beta = 0.25$, $\gamma = 0.20$, and $\delta = 0.20$, with $\tau = 0.8$ and $\lambda_b = 0.1$. The safeguarded sampler used $w_{\min} = 0.5$, $w_{\max} = 2.0$, $\epsilon_u = 0.10$, and $\beta_{\text{IS}} = 0.5$, with correction coefficients clipped to $[0.5, 2.0]$; the resulting N_{eff}/N was 0.69. Distillation used $\omega_{\min} = 0.5$, $\omega_{\max} = 2.0$, $T_D = 0.5$, $\lambda_D = 0.1$, and $\varepsilon = 10^{-6}$. Replay priority used $r_t, q_t \in [0, 1]$, $\eta_1 = 0.5$, $\eta_2 = 0.3$, and $\eta_3 = 0.2$, with clipping to $[0.5, 2.0]$.

Main comparisons used five independent training seeds (13, 21, 42, 87, and 102). Bench2Drive metrics were averaged over three matched closed-loop repetitions per seed, using identical routes and simulator seeds across methods. NAVSIM was evaluated once per trained checkpoint under its fixed protocol. Results are reported as mean $\pm$ standard deviation across the five seed-level results, with two-sided 95% confidence intervals based on Student’s t -distribution. Two-sided paired t -tests paired methods by training seed and evaluation conditions. Holm correction was applied across Driving Score, Success Rate, Collision Rate, and EPDMS within each prespecified comparison at $\alpha = 0.05$.

Training followed three stages. During epochs 1–10, only the policy backbone was trained with uniform sampling; auxiliary networks were frozen, and replay was disabled. During epochs 11–70, the policy, valuation network, and distillability estimator were alternated at a 1:1:1 epoch-level ratio. Values and distillability weights were refreshed each epoch, while the replay buffer was updated every five epochs. Each minibatch contained 24 ordinary and 8 replay samples. During epochs 71–80, auxiliary networks and the buffer were frozen, and only the policy was fine-tuned.

After warm-up, unweighted imitation loss was tracked using an exponential moving average with decay 0.9. The Equation (22) target was updated from stage-wise loss reduction. The estimator was optimized only through L_D , whereas the policy used detached weights from Equation (23) and importance coefficients from Equation (19). This schedule avoided simultaneous changes in sampling, distillability weighting, and replay contents.

The replay buffer contained only training-side failures. For each failure, the final $H = 8$ state–action pairs were replayed under identical routes, traffic, and random seeds. The original action was replaced for one step by the expert or reference-controller action. The normalized failure-cost reduction determined c_t , and y_t^{cf} was retained only when the intervention reduced failure cost. The 20,000-entry buffer removed the lowest-priority samples when full and limited each scenario category to 10%. Failed policy actions were excluded from supervision. Hyperparameters were fixed before final evaluation, after

which no model or buffer updates occurred. Default robustness tests used Gaussian image noise of 0.01 and state perturbations of 1%, with extended results reported in Section 4.5.

4.2. Baselines

Two sets of comparisons were conducted. First, the framework was applied to different autonomous-driving backbones to examine compatibility. Second, it was compared with data-utilization methods to evaluate the effects of data valuation, supervision calibration, and corrective replay.

TCP [26], UniAD [27], and GenAD [28] were selected as representative backbones. TCP is an end-to-end planning and control model, UniAD integrates perception, prediction, and planning, and GenAD adopts generative planning. Their native topology, output heads, task losses, optimizers, and training schedules were retained. The proposed framework appended lightweight feature adapters and auxiliary valuation and distillability networks, accounting for the additional parameters and FLOPs.

A common interface preserved each backbone's native representation and target format. The learner view was taken from the final planning feature before the output head, while the expert view encoded the corresponding training-time trajectory or control target. Action discrepancy was calculated between the native prediction and reference target; valuation used the learner feature; and replay labels remained in the native output space as intervention-validated targets. TCP used waypoint and control representations, UniAD used its planning-query feature and ego-trajectory target, and GenAD used its latent planning feature and generated-trajectory target. Auxiliary losses were connected only to the planning path. All methods used identical data divisions and evaluation criteria.

Uncertainty sampling [29] and conventional prioritized replay [30] were retained as basic controls. Five stronger controls were further evaluated on TCP: hard-example mining, curriculum learning, risk-aware sampling, DAgger-style aggregation, and proportional prioritized replay. Hard-example mining selected samples with high recent imitation losses; curriculum learning progressively introduced higher-loss samples; risk-aware sampling used only the calibrated risk score; DAgger-style aggregation added corrective actions from training-side rollouts without counterfactual attribution; and proportional replay sampled failures according to normalized imitation loss with importance correction.

A parameter-matched auxiliary control retained the same adapters and auxiliary-network capacity as TCP + Ours but used uniform sampling, standard imitation supervision, and no corrective replay. All extended controls used the same initialization, split, 80 epochs, batch size, optimizer steps, and total sample exposure. Sampling- and replay-based methods also used the same 25% selected-sample quota and 20,000-entry buffer. The proposed method differs by jointly using multi-branch valuation, learning-progress-based supervision weights, and counterfactual intervention-validated corrective targets.

4.3. Quantitative Results

The framework was evaluated mainly through closed-loop testing on Bench2Drive, with NAVSIM used for supplementary non-reactive cross-platform evaluation. Table 2 compares data scales, backbones, and data-utilization strategies. Mini, Base, and Full configurations were evaluated on TCP, while the remaining TCP controls used Full. All experiments used five training seeds. Bench2Drive metrics were averaged over three matched repetitions per seed, whereas NAVSIM was evaluated once per checkpoint.

For the scale analysis, only the training-data volume changed; the architecture, backbone-specific training procedure, routes, and simulator seeds were fixed. TCP + Ours improved Driving Score by 2.3, 2.4, and 2.9 points on Mini, Base, and Full, respectively. Success Rate increased by 2.1, 2.5, and 2.4 percentage points, Collision Rate decreased by 1.8,

2.2, and 2.0 percentage points, and EPDMS increased by 0.8, 1.0, and 1.2 points. Although the Mini Success Rate and EPDMS differences were not significant after Holm correction, all metrics improved consistently across scales.

Under Full, TCP + Ours increased Driving Score from 61.8 to 64.7 and Success Rate from 42.6% to 45.0%, reduced Collision Rate from 18.9% to 16.9%, and increased EPDMS from 74.6 to 75.8. UniAD gained 0.7 Driving Score points and reduced Collision Rate by 0.7 percentage points. GenAD gained 0.8 Driving Score points and reduced Collision Rate by 1.0 percentage point. Its EPDMS changed from 78.5 to 78.4, but the difference was not significant ($p = 0.672$; 95% CI: $[-0.31, 0.11]$).

Among matched TCP controls, hard-example mining and curriculum learning mainly improved driving quality or completion, whereas Dagger-style aggregation and proportional prioritized replay produced stronger safety gains. Proportional prioritized replay achieved the lowest Collision Rate, 16.5% versus 16.9%, but the difference was not significant. The full framework achieved significantly higher Driving Score, Success Rate, and EPDMS. The parameter-matched control showed no significant improvement, indicating that additional capacity alone does not explain the gains.

Table 2. Quantitative comparison of baselines and the proposed method.

Method	Driving Score	Success Rate	Collision Rate	EPDMS	Holm-Adjusted (p): DS/SR/CR/EPDMS
TCP (Mini)	55.7 ± 0.6 (55.0–56.4)	36.4 ± 0.7 (35.5–37.3)	24.3 ± 0.5 (23.7–24.9)	71.4 ± 0.3 (71.0–71.8)	—
TCP + Ours (Mini)	58.0 ± 0.5 (57.4–58.6)	38.5 ± 0.6 (37.8–39.2)	22.5 ± 0.5 (21.9–23.1)	72.2 ± 0.3 (71.8–72.6)	0.038/0.061/0.044/0.083
TCP (Base)	59.2 ± 0.5 (58.6–59.8)	40.0 ± 0.6 (39.3–40.7)	21.1 ± 0.4 (20.6–21.6)	73.3 ± 0.3 (72.9–73.7)	—
TCP + Ours (Base)	61.6 ± 0.4 (61.1–62.1)	42.5 ± 0.5 (41.9–43.1)	18.9 ± 0.4 (18.4–19.4)	74.3 ± 0.2 (74.1–74.5)	0.026/0.039/0.022/0.047
TCP (Full)	61.8 ± 0.4 (61.3–62.3)	42.6 ± 0.5 (42.0–43.2)	18.9 ± 0.4 (18.4–19.4)	74.6 ± 0.2 (74.4–74.8)	—
TCP + Uncertainty-Only Sampling	63.0 ± 0.4 (62.5–63.5)	43.4 ± 0.5 (42.8–44.0)	18.2 ± 0.4 (17.7–18.7)	75.0 ± 0.2 (74.8–75.2)	0.043/0.081/0.094/0.118
TCP + Conventional Priority Replay	63.6 ± 0.4 (63.1–64.1)	43.7 ± 0.5 (43.1–44.3)	16.7 ± 0.4 (16.2–17.2)	75.3 ± 0.2 (75.1–75.5)	0.036/0.074/0.021/0.058
TCP + Hard-Example Mining	63.7 ± 0.4 (63.2–64.2)	44.0 ± 0.5 (43.4–44.6)	17.9 ± 0.4 (17.4–18.4)	75.2 ± 0.2 (75.0–75.4)	0.031/0.052/0.112/0.071
TCP + Curriculum Learning	63.5 ± 0.4 (63.0–64.0)	44.2 ± 0.5 (43.6–44.8)	18.0 ± 0.4 (17.5–18.5)	75.2 ± 0.2 (75.0–75.4)	0.039/0.038/0.124/0.069
TCP + Risk-Aware Sampling	63.9 ± 0.4 (63.4–64.4)	44.1 ± 0.5 (43.5–44.7)	17.6 ± 0.4 (17.1–18.1)	75.4 ± 0.2 (75.2–75.6)	0.024/0.046/0.071/0.041
TCP + Dagger-Style Aggregation	64.1 ± 0.4 (63.6–64.6)	44.4 ± 0.5 (43.8–45.0)	16.6 ± 0.4 (16.1–17.1)	75.5 ± 0.2 (75.3–75.7)	0.019/0.033/0.017/0.028
TCP + Proportional Prioritized Replay	64.2 ± 0.4 (63.7–64.7)	44.3 ± 0.4 (43.8–44.8)	16.5 ± 0.4 (16.0–17.0)	75.5 ± 0.2 (75.3–75.7)	0.022/0.029/0.014/0.025
TCP + Parameter-Matched Control	62.5 ± 0.4 (62.0–63.0)	43.0 ± 0.5 (42.4–43.6)	18.5 ± 0.4 (18.0–19.0)	74.9 ± 0.2 (74.7–75.1)	0.116/0.184/0.213/0.241
TCP + Ours (Full)	64.7 ± 0.3 (64.3–65.1)	45.0 ± 0.4 (44.5–45.5)	16.9 ± 0.3 (16.5–17.3)	75.8 ± 0.2 (75.6–76.0)	0.014/0.028/0.019/0.031
UniAD	66.9 ± 0.4 (66.4–67.4)	48.4 ± 0.5 (47.8–49.0)	15.8 ± 0.3 (15.4–16.2)	77.8 ± 0.2 (77.6–78.0)	—
UniAD + Ours	67.6 ± 0.4 (67.1–68.1)	48.5 ± 0.5 (47.9–49.1)	15.1 ± 0.3 (14.7–15.5)	78.1 ± 0.2 (77.9–78.3)	0.049/0.624/0.037/0.128
GenAD	67.5 ± 0.4 (67.0–68.0)	49.3 ± 0.4 (48.8–49.8)	16.4 ± 0.3 (16.0–16.8)	78.5 ± 0.2 (78.3–78.7)	—
GenAD + Ours	68.3 ± 0.5 (67.7–68.9)	49.6 ± 0.5 (49.0–50.2)	15.4 ± 0.3 (15.0–15.8)	78.4 ± 0.2 (78.2–78.6)	0.046/0.214/0.041/0.672

Note: Results are mean ± standard deviation over five seeds; parentheses give 95% confidence intervals. Bench2Drive metrics were averaged over three matched repetitions, whereas NAVSIM EPDMS was evaluated once per checkpoint. Paired t-tests used matched seeds and conditions, with Holm correction across DS, SR, CR, and EPDMS ($\alpha = 0.05$). Mini, Base, and Full denote official

Bench2Drive configurations. Scale comparisons used TCP and TCP + Ours; other TCP controls used Full. For TCP + Ours versus proportional prioritized replay, adjusted p -values were 0.041/0.047/0.218/0.044. Matched controls used equal training budgets and exposure; sampling and replay variants used a 25% quota and a 20,000-entry buffer.

Figure 5 presents the validation Driving Score curves on Bench2Drive-Dev10. TCP showed the clearest improvement, whereas the changes for UniAD and GenAD were less pronounced. Conventional prioritized replay improved rapidly during the early stages but exhibited greater subsequent fluctuation. Under the staged schedule described in Section 4.1, sample values and distillability weights were refreshed once per epoch, while the replay buffer was updated every five epochs and fixed between updates. The auxiliary modules and replay buffer were frozen during the final ten epochs. The validation curves stabilized after repeated updates without persistent late-stage oscillation, providing empirical evidence of optimization stability rather than a formal convergence guarantee.

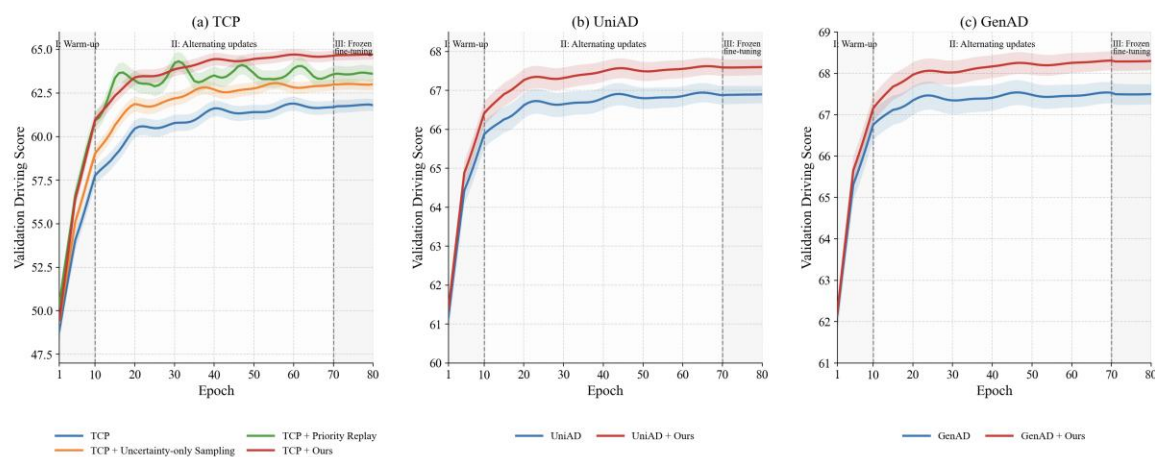


Figure 5. Validation Driving Score curves on Bench2Drive-Dev10 for (a) TCP, (b) UniAD, and (c) GenAD. The staged optimization schedule generally achieves earlier stabilization than the corresponding baselines, with the clearest improvement for TCP.

As shown in Table 3, the proposed method preserves the backbone topology while adding lightweight auxiliary components, resulting in moderate computational overhead. These costs arise mainly from data valuation, supervision calibration, and failure replay. TCP shows limited increases in parameters, training time, and inference latency, while UniAD and GenAD exhibit similar trends. Overall, the added costs remain limited relative to the improvements in closed-loop driving and cross-platform evaluation, indicating a practical trade-off between computational overhead and performance.

Table 3. Training-framework efficiency across representative backbones.

Method	Parameters (M)	FLOPs (G)	Relative Training Time per Epoch	Inference Time (ms)
TCP	31.2	86.4	1.00×	42
TCP + Ours	34.7	91.8	1.12×	45
UniAD	54.6	131.7	1.00×	57
UniAD + Ours	58.1	137.9	1.09×	59
GenAD	61.8	149.2	1.00×	63
GenAD + Ours	64.9	154.8	1.08×	65

Note: “+Ours” includes the backbone-specific feature adapters and the auxiliary valuation and distillability networks. The native backbone layers and prediction heads remain unchanged. Parameters and FLOPs refer to the complete training/evaluation graph, while counterfactual attribution is computed offline.

Table 3 quantifies computational overhead under the fixed experimental data configuration and should not be interpreted as a Mini/Base/Full scaling study. Let N denote the number of training samples, d the stored feature dimension, B the replay-buffer capacity, and d_r the replay-entry dimension. Auxiliary-score refresh requires $O(Nd)$ computation, while cached feature storage requires $O(Nd)$ memory. Exact k -nearest-neighbor novelty retrieval has a worst-case cost of $O(N^2d)$, although indexed retrieval can reduce its practical cost. Replay storage requires $O(Bd_r)$ memory. Because B is capped at 20,000, replay memory remains bounded as the log pool grows. Although performance was compared across the Mini, Base, and Full training configurations, training time, memory consumption, and auxiliary-processing cost were not systematically measured as the log-pool size increased. The results therefore support performance consistency across the evaluated data configurations but do not establish empirical computational scalability.

4.4. Scenario-Specific and Qualitative Results

To complement the aggregate metrics, the frozen Bench2Drive episodes were grouped into unprotected turns, pedestrian and vulnerable road user (VRU) interactions, construction zones, adverse weather, and rare multi-agent interactions. Panel A of Table 4 reports scenario-specific Success Rate and Collision Rate, while Panel B summarizes four failure categories. TCP + Ours improved both metrics across all categories, with larger gains in unprotected turns and pedestrian/VRU interactions and a smaller gain under adverse weather.

Table 4. Scenario-specific performance, failure categories, and realistic robustness evaluation.

Panel A. Scenario-specific closed-loop results					
Scenario category	Episodes	TCP Success Rate (%)	TCP + Ours Success Rate (%)	TCP Collision Rate (%)	TCP + Ours Collision Rate (%)
Unprotected turns	34	37.8 ± 2.2	42.6 ± 2.0	22.1 ± 1.7	17.9 ± 1.5
Pedestrian/VRU interactions	28	35.2 ± 2.5	40.8 ± 2.1	24.5 ± 1.9	20.1 ± 1.6
Construction zones	25	39.1 ± 2.3	43.7 ± 2.0	22.9 ± 1.8	19.4 ± 1.7
Adverse weather	31	36.5 ± 2.4	38.8 ± 2.3	23.8 ± 2.0	21.9 ± 1.8
Rare multi-agent interactions	22	33.9 ± 2.7	38.6 ± 2.4	25.7 ± 2.1	22.5 ± 1.9
Panel B. Failure-category counts					
Failure category		TCP	TCP + Ours	Change (%)	
Vehicle collision		28	23	−17.9	
Pedestrian/VRU collision		10	8	−20.0	
Off-road deviation		19	16	−15.8	
Deadlock or route timeout		13	11	−15.4	
Panel C. Realistic robustness results					
Test condition		TCP Driving Score	TCP + Ours Driving Score	TCP Collision Rate (%)	TCP + Ours Collision Rate (%)
Motion blur		57.6 ± 0.7	60.1 ± 0.6	22.7 ± 0.6	20.0 ± 0.5
Partial camera occlusion		56.2 ± 0.8	58.5 ± 0.7	24.3 ± 0.7	22.1 ± 0.6
10% frame dropout		58.9 ± 0.6	61.0 ± 0.5	21.6 ± 0.5	19.3 ± 0.5
Adverse weather/low light		53.9 ± 0.9	55.6 ± 0.8	26.4 ± 0.8	24.6 ± 0.7

Note: Scenario categories were assigned using benchmark annotations and may overlap when several risk factors occur in one episode. Results are mean ± standard deviation across five training

seeds. Failure counts are reported for the same fixed set of matched evaluation episodes used for both TCP and TCP + Ours. Each episode was evaluated under identical routes, traffic configurations, and simulator seeds. These counts are descriptive and were not used for statistical significance testing. All tests used frozen models and identical routes, traffic settings, and simulator seeds; evaluated failures were not reused for training.

Two representative cases from the closed-loop evaluation are shown in Figures 6 and 7. Figure 6 presents a successful case, whereas Figure 7 presents a remaining failure. The results are consistent with the quantitative analysis: the proposed method better utilizes complex and failure-critical samples, but its gains vary across high-risk scenarios and do not eliminate errors under multiple simultaneous risks.

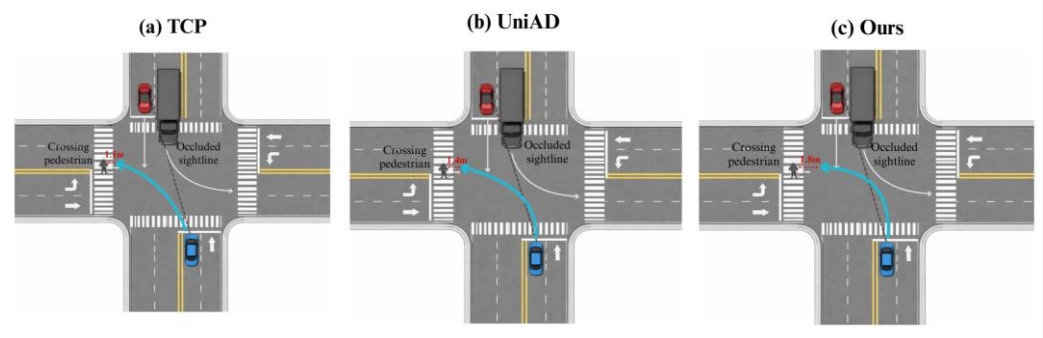


Figure 6. Visual results in the unprotected left-turn scenario. (a) TCP: The turn is initiated too early. (b) UniAD: Insufficient braking. (c) Our method: The trajectory is more stable and maintains a larger safety margin.

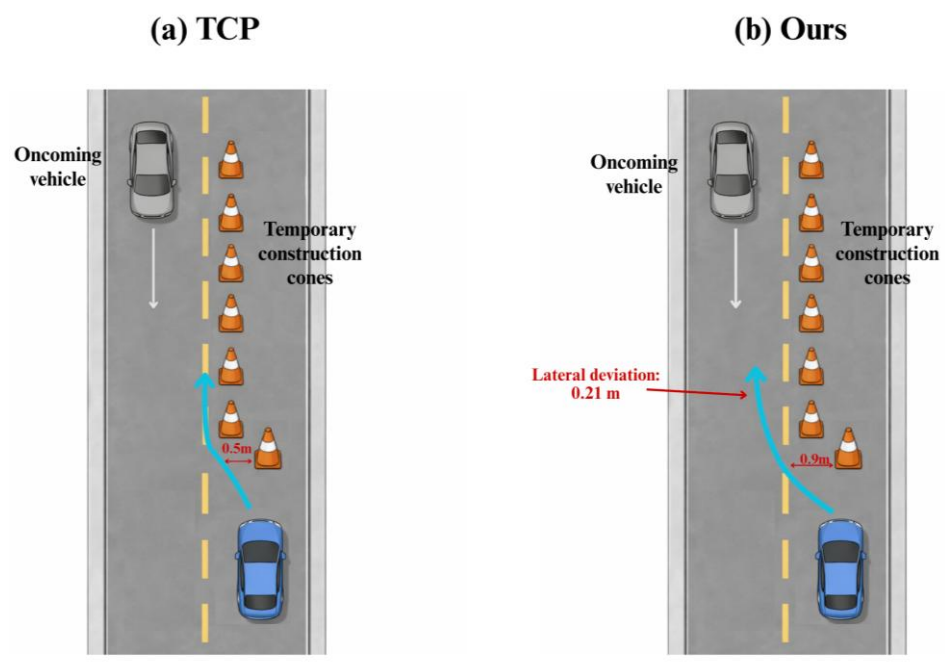


Figure 7. Failure cases in the scenario of temporary construction cones on narrow roads. (a) TCP: Delayed avoidance, with a relatively small distance from the obstacle. (b) Our method: Decelerated in advance, but still experienced a slight deviation from the road.

Figure 6 presents an unprotected left-turn scenario involving oncoming vehicles, crossing pedestrians, and partial visual obstruction. The baseline methods either initiate the turn too early or brake insufficiently, whereas the proposed method maintains a

smoother trajectory and larger safety margin. The vehicle completes the turn without collision or route deviation. This case is consistent with the unprotected-turn results in Panel A of Table 4, but does not imply uniform improvement across all critical situations.

Figure 7 shows a narrow-road scenario with temporary construction cones. The proposed method detects the obstacle and decelerates earlier than TCP, maintaining a larger safety distance. However, it underestimates the required lateral clearance and slightly deviates from the road. This result is consistent with the lower construction-zone Collision Rate in Table 4, while showing that rare or compound conditions can still cause local control errors.

4.5. Robustness

Robustness was evaluated empirically on Bench2Drive-Dev10, with supplementary validation on NAVSIM. Besides Gaussian image noise and ego-state perturbations, motion blur, partial camera occlusion, 10% frame dropout, and an adverse-weather/low-light shift were tested. Figures 8 and 9 report the original numerical perturbations, while Panel C of Table 4 reports the additional realistic conditions. Because the framework modifies data utilization and supervision rather than imposing vehicle-dynamics constraints, it provides no formal stability or collision-avoidance guarantee.

Figure 8 shows that Driving Score decreases as image noise increases. The proposed method differs little from TCP under weak noise, but declines more slowly under moderate interference, indicating smaller degradation rather than general noise invariance. Under strong noise, all methods deteriorate substantially, showing that data optimization cannot recover severely missing perception information.

Figure 9 reports the state-perturbation results. Collision Rate increases for all methods, but the proposed method remains lower across the evaluated amplitudes. The trend is not fully linear under medium perturbations, suggesting sensitivity to particular state errors. At stronger perturbation levels, its task-completion advantage narrows, confirming that the method reduces but does not remove sensitivity to extreme state deviations.

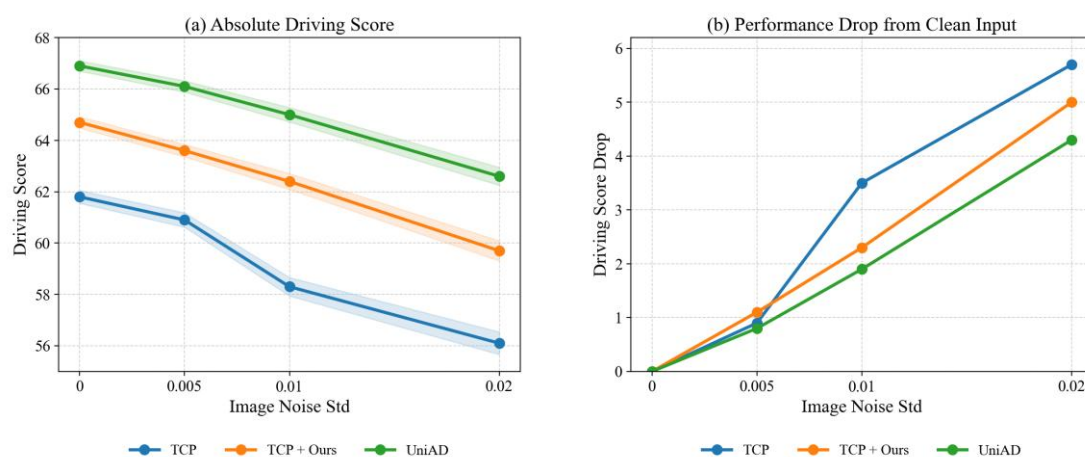


Figure 8. Image robustness test of Bench2Drive-Dev10 under Gaussian noise conditions. (a) Driving Scores at different noise levels. (b) Performance degradation compared to the case without noise input.

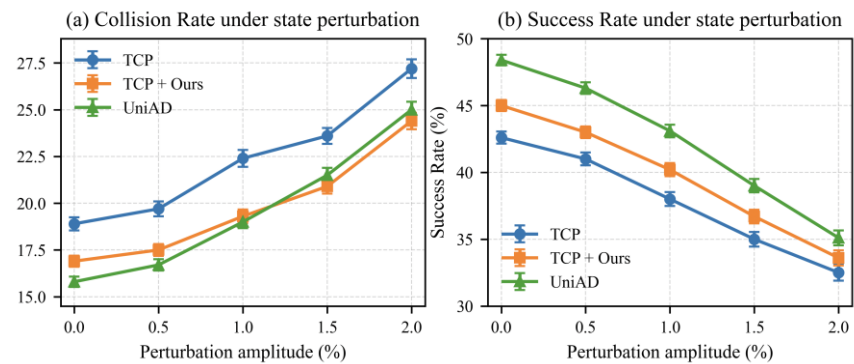


Figure 9. Robustness under ego-state perturbations on Bench2Drive-Dev10. (a) Collision Rate at different perturbation amplitudes. (b) Success Rate at different perturbation amplitudes.

Panel C shows that TCP + Ours retained higher Driving Scores and lower Collision Rates under all four realistic conditions. The Driving Score gains were 2.5, 2.3, 2.1, and 1.7 points for motion blur, camera occlusion, frame dropout, and adverse-weather/low-light shift, respectively. The narrowing gain under the combined domain shift indicates that data-level mechanisms remain limited when both visibility and illumination deteriorate.

These results can be associated with three components: risk-aware valuation increases exposure to high-risk samples, supervision-calibrated distillation reduces expert-learner mismatch, and failure attribution replay reinforces failure-critical states. However, the analysis concerns low-frequency conditions represented in the available benchmark and does not establish generalization to entirely unseen tail categories. On NAVSIM, Gaussian noise with a standard deviation of 0.01 reduced EPDMS by 1.3 points for TCP, 0.8 for UniAD, and 0.7 for the proposed method. Overall, the method shows smaller degradation under the evaluated low-to-moderate failures, while its advantage narrows under severe perception loss and domain shift. Therefore, the lower Collision Rate represents benchmark-specific empirical evidence rather than guaranteed ride safety; deployment would require an independent safety monitor or fallback controller.

4.6. Ablation Study

To evaluate the contribution of each component, we conducted ablation experiments using the TCP backbone. Driving Score, Success Rate, and Collision Rate were evaluated on Bench2Drive, while EPDMS was reported using NAVSIM. Results are presented as mean $\pm$ standard deviation over three independent runs. As shown in Table 5, removing any module degraded at least one metric, indicating that the improvements were not attributable to a single mechanism.

Table 5. Ablation results of different components.

Method	Driving Score	Success Rate	Collision Rate	EPDMS
TCP baseline	61.8 $\pm$ 0.4	42.6 $\pm$ 0.5	18.9 $\pm$ 0.4	74.6 $\pm$ 0.2
+Ours (Full)	64.7 $\pm$ 0.3	45.0 $\pm$ 0.4	16.9 $\pm$ 0.3	75.8 $\pm$ 0.2
w/o Data Valuation	63.3 $\pm$ 0.4	43.8 $\pm$ 0.5	17.8 $\pm$ 0.4	75.0 $\pm$ 0.2
w/o Distillation	63.8 $\pm$ 0.3	43.9 $\pm$ 0.4	17.6 $\pm$ 0.3	74.8 $\pm$ 0.3
w/o Replay	63.6 $\pm$ 0.4	44.2 $\pm$ 0.4	17.3 $\pm$ 0.3	75.1 $\pm$ 0.2
Valuation only	63.1 $\pm$ 0.4	43.5 $\pm$ 0.5	18.1 $\pm$ 0.4	74.9 $\pm$ 0.2
Distillation only	62.9 $\pm$ 0.3	44.0 $\pm$ 0.4	18.2 $\pm$ 0.3	74.7 $\pm$ 0.2
Replay only	63.0 $\pm$ 0.4	43.0 $\pm$ 0.5	16.7 $\pm$ 0.4	74.8 $\pm$ 0.2
Valuation + Distillation	64.1 $\pm$ 0.4	45.3 $\pm$ 0.5	17.5 $\pm$ 0.3	75.3 $\pm$ 0.2
Valuation + Replay	63.9 $\pm$ 0.4	44.2 $\pm$ 0.4	16.6 $\pm$ 0.3	75.1 $\pm$ 0.2
Distillation + Replay	63.6 $\pm$ 0.3	44.0 $\pm$ 0.4	17.1 $\pm$ 0.3	75.0 $\pm$ 0.2

Removing data valuation reduced the Driving Score from 64.7 to 63.3 and increased the Collision Rate from 16.9 to 17.8. Removing distillation lowered the Success Rate and EPDMS to 43.9 and 74.8, while removing replay increased the Collision Rate to 17.3. Among the single-module variants, valuation alone produced the largest Driving Score gain, replay alone achieved the lowest Collision Rate, and distillation mainly improved task completion. Across all ablation variants, valuation plus distillation obtained the highest Success Rate, whereas valuation plus replay achieved the lowest Collision Rate. The full framework achieved the highest Driving Score and EPDMS and provided the most balanced performance across driving quality, task completion, safety, and cross-platform evaluation. Thus, the full configuration provides a broader balance among driving quality, completion, safety, and cross-platform performance.

Table 6 reports the branch-level ablation results obtained by removing each valuation branch while keeping the architecture, sampling budget, and optimization settings unchanged.

Table 6. Branch-level ablation results for the data valuation module.

Variant	Driving Score	Success Rate	Collision Rate	EPDMS
Full valuation module	64.7 ± 0.3	45.0 ± 0.4	16.9 ± 0.3	75.8 ± 0.2
w/o Risk branch	64.8 ± 0.4	44.4 ± 0.5	17.8 ± 0.4	75.3 ± 0.2
w/o Uncertainty branch	64.0 ± 0.3	44.3 ± 0.4	17.1 ± 0.3	75.1 ± 0.3
w/o Novelty branch	63.9 ± 0.5	44.6 ± 0.4	17.4 ± 0.4	75.0 ± 0.2
w/o Distribution Correction	64.2 ± 0.4	45.2 ± 0.5	17.2 ± 0.3	75.2 ± 0.2

Removing the risk branch slightly increased the Driving Score but raised the Collision Rate, indicating more aggressive behavior. Removing uncertainty reduced the Driving Score, Success Rate, and EPDMS. Novelty removal caused the largest decreases in Driving Score and EPDMS, supporting its role in scenario coverage. Removing distribution correction slightly improved Success Rate but reduced Driving Score and EPDMS. These distinct degradation patterns support the separate functions of the four branches.

We next examined whether V_i was associated with the benefit of adding a sample group to training. As shown in Table 7, utility generally increased with value, although the relationship was not strictly monotonic. Equal numbers from each quantile were added separately to the same base set, with the backbone, initialization, optimization steps, and total data exposure fixed. The approximate marginal utility was

$$\Delta DS_q = DS(\mathcal{D}_{\text{base}} \cup \mathcal{B}_q) - DS(\mathcal{D}_{\text{base}}), \quad (28)$$

where $\mathcal{B}_q$ denotes samples from quantile q . The association was measured by

$$\rho_V = \text{Spearman}(\bar{V}_q, \Delta DS_q). \quad (29)$$

Table 7. Relationship between sample value and approximate marginal closed-loop utility.

Value Quantile	Mean V_i	Added Samples	Δ Driving Score	Δ Success Rate	Δ Collision Rate
Q1	0.17	3000	0.20 ± 0.22	0.11 ± 0.18	−0.08 ± 0.12
Q2	0.33	3000	0.52 ± 0.24	0.39 ± 0.21	−0.17 ± 0.14
Q3	0.49	3000	0.44 ± 0.21	0.57 ± 0.23	−0.24 ± 0.15
Q4	0.66	3000	1.02 ± 0.28	1.08 ± 0.26	−0.46 ± 0.17
Q5	0.83	3000	1.18 ± 0.31	0.96 ± 0.29	−0.43 ± 0.18

Note: Results are mean ± standard deviation over three retraining runs. A negative Δ Collision Rate indicates improved safety.

Utility generally increased with value, although the relationship was not strictly monotonic. Q3 produced a smaller Driving Score gain than Q2, while Q4 exceeded Q5 in Success Rate and Collision Rate improvement. Nevertheless, Q4 and Q5 outperformed lower-value groups overall. The Spearman correlation was $\rho_V = 0.90$, indicating that V_i generally ranks samples by subsequent training benefit. This is a group-level association rather than an exact causal estimate for individual samples.

We further validated c_t through controlled counterfactual replay. Equal numbers of state–action pairs were selected by random sampling, temporal proximity, risk ranking, or c_t -based ranking. At each position, the recorded action was replaced for one step by the expert or reference-controller action under identical simulator conditions. Only interventions that reduced failure cost were retained, and the intervention action was stored as y_t^{cf} . Table 8 compares the failure-cost reduction and avoidance rate produced by the four state–action selection methods.

Table 8. Counterfactual validation of the failure contribution.

State–Action Selection Method	Failure Cost Reduction (%)	Failure Avoidance Rate (%)
Random historical states	8.9 ± 1.2	13.1 ± 1.7
Temporally nearest states	14.6 ± 1.4	21.8 ± 2.0
Risk-score ranking	17.3 ± 1.5	25.6 ± 2.2
Top-ranked by c_t	21.1 ± 1.6	29.4 ± 2.3

Note: Results are mean $\pm$ standard deviation over three replay runs. All methods use equal numbers of pairs and identical interventions.

The c_t -based method achieved the largest cost reduction and avoidance rate. Its advantage over temporal proximity indicates that c_t does not merely identify states close to the failure. The validated actions were used as corrective targets, while failed policy actions were excluded. However, c_t represents a local effect under fixed simulator conditions rather than unrestricted real-world causal responsibility.

Finally, we evaluated whether the sampling safeguards prevented excessive concentration and repeated-failure dominance. Temperature, uniform mixing, priority clipping, and importance correction were varied under the same TCP backbone and training budget. Sampling concentration was measured by $N_{\text{eff}}/N = 1/[N \sum_i (\pi_i^v)^2]$. Table 9 reports the sensitivity results for temperature, uniform mixing, priority clipping, and importance correction.

Table 9. Sensitivity of sampling safeguards on Bench2Drive-Dev10.

Setting	τ	Priority Clip	ϵ_u	β_{IS}	$\frac{N_{\text{eff}}}{N}$	Driving Score	Collision Rate
Low temperature	0.4	[0.5, 2.0]	0.10	0.5	0.46	63.9 ± 0.4	17.5 ± 0.4
Moderate temperature	0.6	[0.5, 2.0]	0.10	0.5	0.58	64.4 ± 0.3	17.1 ± 0.3
Selected setting	0.8	[0.5, 2.0]	0.10	0.5	0.69	64.7 ± 0.3	16.9 ± 0.3
High temperature	1.0	[0.5, 2.0]	0.10	0.5	0.77	64.5 ± 0.4	17.0 ± 0.3
Near-uniform sampling	1.2	[0.5, 2.0]	0.10	0.5	0.83	64.2 ± 0.4	17.2 ± 0.4
Without uniform mixture	0.8	[0.5, 2.0]	0	0.5	0.52	64.2 ± 0.4	17.3 ± 0.4
Without priority clipping	0.8	None	0.10	0.5	0.54	64.0 ± 0.5	17.5 ± 0.4
Without IS correction	0.8	[0.5, 2.0]	0.10	0	0.69	64.4 ± 0.4	17.1 ± 0.3
Full IS correction	0.8	[0.5, 2.0]	0.10	1.0	0.69	64.3 ± 0.5	17.2 ± 0.4

Note: Results are mean $\pm$ standard deviation over three runs. IS denotes importance sampling.

Low temperature concentrated sampling probability and reduced effective sample size, whereas high temperature weakened prioritization. Removing uniform mixing or

clipping also reduced N_{eff}/N and degraded performance. Partial importance correction performed better than no or full correction. Therefore, $\tau = 0.8$, $\epsilon_u = 0.10$, clipping to $[0.5 \cdot 2.0]$, and $\beta_{\text{IS}} = 0.5$ were adopted. Combined with the 25% replay ratio and 10% per-category buffer cap, these settings preserve diversity and prevent a small group of samples from dominating training.

5. Discussion

The experimental results support the complementary relationship between model-centric and data-centric development. Model-centric methods primarily improve the representational, predictive, planning, and control capacity of the backbone, whereas the proposed data-centric framework improves the allocation of training samples, the calibration of supervision, and the reuse of closed-loop failures. The latter cannot compensate for an inadequate backbone or severe information loss, but it can improve how an existing backbone uses the available data. Unlike PARA-Drive [15], which mainly improves performance through architectural and computational design, this paper preserves the evaluated backbone structures and modifies their data and supervision processes. The improvements observed for TCP, UniAD, and GenAD therefore indicate that data-centric optimization can complement different model architectures rather than replace them. The proposed method achieves a relatively balanced trade-off among driving performance, safety, and cross-platform evaluation.

Further analysis shows that the three modules affect different aspects of training. The data valuation module primarily improves overall driving performance, while branch-level ablations and the positive association between V_i and closed-loop gains support the distinct roles of risk, uncertainty, novelty, and distribution correction. The supervision-calibrated distillation module improves task completion and cross-platform performance by weighting demonstrations according to independently supervised learning progress. The Failure Attribution Replay Module mainly contributes to safety. Unlike feedback-guided behavior-cloning approaches [14], failed trajectories in the proposed framework undergo controlled counterfactual replay, and only interventions that reduce failure cost are retained. Their expert or reference-controller actions serve as corrective targets, preventing failed actions from being relearned. Together, the modules provide complementary mechanisms for sample prioritization, supervision calibration, and corrective replay.

Under the Full configuration, TCP + Ours increased Driving Score from 61.8 to 64.7 and Success Rate from 42.6% to 45.0%, reduced Collision Rate from 18.9% to 16.9%, and increased EPDMS from 74.6 to 75.8. These gains were accompanied by a 12% increase in training time per epoch and an inference-latency increase from 42 to 45 ms. Thus, the framework provides a practical data-level extension to existing backbones, although its benefits remain dependent on the evaluated datasets and simulation conditions. These results do not establish the superiority of data-centric learning over model-centric design; rather, they show that data-level optimization can provide additional gains after the backbone architecture and native task losses have been fixed.

The robustness experiments further support this interpretation. The method remained relatively stable under mild and moderate perturbations but degraded under strong interference. This pattern is consistent with improved data utilization but does not imply that data-level optimization can compensate for severe information loss. Compared with VisionPAD [21], which relies on large-scale pre-training and stronger representation learning, the proposed method adjusts sample importance, calibrates demonstration supervision, and reuses corrected failure-critical samples. Compared with DriveGPT4-V2 [20], which introduces stronger reasoning and language-enhancement modules, the proposed method operates mainly at the data and supervision levels without substantially changing the policy backbone.

This method still has several limitations. Performance remains affected under strong interference, and scenarios involving multiple uncertain factors may still produce local decision errors. Data optimization and corrective replay cannot cover all situations. Moreover, branch-target construction, uncertainty estimation, nearest-neighbor retrieval, and counterfactual replay introduce additional training costs. The attribution score c_t represents a local intervention effect under restored simulator states, fixed traffic configurations, and controlled random seeds; it does not establish unrestricted causal responsibility in real-world traffic. Its reliability also depends on the expert or reference controller used to generate y_t^{cf} . Balancing performance gains, replay costs, and deployment efficiency remains necessary. Moreover, the current evaluation does not hold out complete low-frequency scenario categories. Performance improvements were observed across the Mini, Base, and Full training configurations, showing that the gains were not limited to a single evaluated data configuration. However, training time, memory consumption, and auxiliary-processing cost were not systematically measured as the log-pool size increased. In addition, complete low-frequency scenario categories were not held out during training. The current evidence therefore supports performance consistency across the evaluated data configurations and improved utilization of low-frequency instances within the observed distribution, but it does not establish computational scalability or generalization to entirely unseen tail categories.

Overall, closed-loop autonomous driving performance depends not only on model capability but also on how training data are selected, supervised, and fed back. Future work should examine whether the method remains effective with limited demonstrations. Further studies should evaluate attribution under broader traffic configurations and simulator seeds, improve corrective-target generation, and assess whether intervention-validated replay transfers to real-world driving logs.

6. Conclusions

This study developed a data-centric closed-loop learning framework integrating risk-aware data valuation, supervision-calibrated distillation, and intervention-validated failure replay. Its main contribution is not the isolated use of sampling, distillation, or replay, but their integration into a unified data-utilization loop that determines which samples should be prioritized, which demonstrations are learnable, and which closed-loop failures should be reused. Unlike single-signal sampling, fixed or discrepancy-based distillation, and temporal or loss-prioritized replay, the framework uses independently calibrated valuation factors, learning-progress-based supervision, and controlled interventions that retain only failure-reducing corrective actions.

For the Bench2Drive Full configuration, TCP + Ours increased Driving Score from 61.8 to 64.7 and Success Rate from 42.6% to 45.0%, reduced Collision Rate from 18.9% to 16.9%, and improved NAVSIM EPDMS from 74.6 to 75.8. Improvements were also observed with UniAD and GenAD, supporting compatibility across the evaluated backbones. Ablation results showed that data valuation mainly improved overall driving performance, distillation supported task completion and cross-platform evaluation, and failure replay contributed most directly to collision reduction. However, the evidence is limited to dataset-based and simulation evaluations, provides no formal safety guarantee, and shows narrower gains under severe perturbations. Future work should examine unseen scenarios, real-world driving logs, and integration with independent safety monitors or model-based fallback controllers.

Author Contributions: Conceptualization, Y.L.; Methodology, Y.L.; Software, Y.L.; Validation, Y.L.; Formal analysis, Y.L.; Investigation, W.K.; Resources, W.K.; Data curation, W.K.; Writing—original

draft, Y.L.; Writing—review & editing, W.K.; Visualization, W.K.; Supervision, W.K.; Project administration, W.K.; Funding acquisition, W.K. All authors have read and agreed to the published version of the manuscript.

Funding: Not applicable.

Data Availability Statement: The data can be requested from the corresponding author.

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

Notations and Abbreviations

Notation	Definition
x, y	Driving-log input and reference decision output
$f_\theta, \theta^*, \mathcal{L}$	Decision function, optimal parameters, and imitation loss
$s, d^{f_\theta}, \mathcal{U}(s)$	Closed-loop state, policy-induced state distribution, and benchmark utility
$\phi(x)$	Latent feature representation
R, U, N, A	Risk, uncertainty, novelty, and distribution-correction scores
V, w^v, p	Fused sample value, valuation weight, and sampling probability
E_e, E_l, z_e, z_l	Expert/learner encoders and latent representations
$\Delta a, D, \omega$	Action discrepancy, distillability score, and distillation weight
$sg[\cdot], \sigma(\cdot)$	Stop-gradient and sigmoid operators
c_t, r_t, q_t	Counterfactual contribution, risk, and rarity scores
$L_{\text{traj}}, L_{\text{act}}, L_{\text{value}}, L_{\text{distill}}, L_{\text{replay}}$	Trajectory, action, valuation, distillation, and replay losses
$\pi_i^v, \hat{\pi}_i^v, \kappa_i$	Learned sampling probability, target probability, and importance coefficient
$D_i^t, \widehat{D}_i^t, \omega_i^t$	Predicted distillability, learning-progress target, and normalized weight
B_f, P_t, y_t^{cf}	Replay buffer, replay priority, and validated corrective target
$\Delta DS_q, \bar{V}_q, \rho_V$	Marginal Driving Score gain, mean quantile value, and Spearman correlation
Abbreviation	Definition
TCP	Trajectory-guided Control Prediction
UniAD	Unified Autonomous Driving
GenAD	Generative End-to-End Autonomous Driving
EPDMS	Extended PDM Score
DS/SR/CR	Driving Score/Success Rate/Collision Rate
IS	Importance sampling
DAGger	Dataset Aggregation
FLOPs	Floating-point operations

References

- Sanato, B.; Wijayanti, R. Distributed Computing Paradigms for Scalable Big Data Architectures in Autonomous Driving Applications. *Trans. Artif. Intell. Mach. Learn. Cogn. Syst.* **2024**, *9*, 11–22.
- Peltonen, E.; Basnayake, V.; Elgendy, N.; Kämä, B.; Seppänen, P.; Päiväranta, T. Towards Data-Centric and Context-Aware Decision Making in Software-Defined Vehicles. In *2025 IEEE 22nd International Conference on Software Architecture Companion (ICSAC)*; IEEE: New York, NY, USA, 2025; pp. 574–577.
- Zha, D.; Bhat, Z.P.; Lai, K.H.; Yang, F.; Jiang, Z.; Zhong, S.; Hu, X. Data-centric artificial intelligence: A survey. *ACM Comput. Surv.* **2025**, *57*, 1–42.
- Liu, X.; Fang, Q. Mathematical framework for robust autonomous driving via stochastic graph optimization and CVaR policies. *Alex. Eng. J.* **2026**, *138*, 128–140.

5. Hu, X.; Cui, J.; Zhang, R.; Fang, Q. Reconfigurable Wireless Channel Optimization and Low-Complexity Control Methods Driven by Intelligent Metasurfaces 2.0. *Telecom* **2026**, *7*, 15.
6. Yan, T.; Wu, D.; Han, W.; Jiang, J.; Zhou, X.; Zhan, K.; Xu, C.-Z.; Shen, J. Drivingsphere: Building a high-fidelity 4d world for closed-loop simulation. In Proceedings of the Computer Vision and Pattern Recognition Conference, Nashville, TN, USA, 10–17 June 2025; pp. 27531–27541.
7. Conradi Hoffmann, J.L.; Passig Horstmann, L.; Fröhlich, A.A. Transparent integration of autonomous vehicles simulation tools with a data-centric middleware. *Des. Autom. Embed. Syst.* **2024**, *28*, 45–66.
8. Escorihuela, E. Data-Centric Operational Design Domain. In *Computer Safety, Reliability, and Security: 42nd International Conference, SAFECOMP 2023, Toulouse, France, 20–22 September 2023, Proceedings*; Springer Nature: Cham, Switzerland, 2023; p. 227.
9. 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, Seattle, WA, USA, 16–22 June 2024; pp. 15120–15130.
10. Yang, B.; Su, H.; Gkanatsios, N.; Ke, T.W.; Jain, A.; Schneider, J.; Fragkiadaki, K. Diffusion-es: Gradient-free planning with diffusion for autonomous and instruction-guided driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 15342–15353.
11. 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.* **2024**, *37*, 819–844.
12. Kim, H.J.; Yin, Z.; Lai, L.; Lee, J.; Ohn-Bar, E. BranchOut: Capturing Realistic Multimodality in Autonomous Driving Decisions. In *Proceedings of the 9th Annual Conference on Robot Learning*; PMLR: Cambridge, MA, USA, 2025; Volume 305, pp. 1940–1952.
13. Paniego, S.; Calvo-Palomino, R.; Cañas, J. Behavior metrics: An open-source assessment tool for autonomous driving tasks. *SoftwareX* **2024**, *26*, 101702.
14. Zhang, J.; Huang, Z.; Ray, A.; Ohn-Bar, E. Feedback-guided autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 15000–15011.
15. Weng, X.; Ivanovic, B.; Wang, Y.; Wang, Y.; Pavone, M. Para-drive: Parallelized architecture for real-time autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 15449–15458.
16. Wang, Y.; He, J.; Fan, L.; Li, H.; Chen, Y.; Zhang, Z. Driving into the future: Multiview visual forecasting and planning with world model for autonomous driving. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 14749–14759.
17. Rao, B.; Wang, C.; Liao, H.; Wang, Q.; Guan, Y.; Zhang, J.; Liu, X.; Zhu, M.; Wang, K.Y.; Li, Z. Differentiable Semantic Meta-Learning Framework for Long-Tail Motion Forecasting in Autonomous Driving. *Proc. AAAI Conf. Artif. Intell.* **2026**, *40*, 962–970.
18. 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, Honolulu, HI, USA, 19–23 October 2025; pp. 24823–24834.
19. Fu, C.; Fang, Q. Curriculum-aware cognitive diagnosis via graph neural networks. *Information* **2025**, *16*, 996.
20. Xu, Z.; Bai, Y.; Zhang, Y.; Li, Z.; Xia, F.; Wong, K.Y.K.; Wang, J.; Zhao, H. Drivegpt4-v2: Harnessing large language model capabilities for enhanced closed-loop autonomous driving. In Proceedings of the Computer Vision and Pattern Recognition Conference, Nashville, TN, USA, 10–17 June 2025; pp. 17261–17270.
21. Zhang, H.; Zhou, W.; Zhu, Y.; Yan, X.; Gao, J.; Bai, D.; Cai, Y.; Liu, B.; Cui, S.; Li, Z. Visionpad: A vision-centric pre-training paradigm for autonomous driving. In Proceedings of the Computer Vision and Pattern Recognition Conference, Nashville, TN, USA, 10–17 June 2025; pp. 17165–17175.
22. Zhou, X.; Liu, M.; Yurtsever, E.; Zagar, B.L.; Zimmer, W.; Cao, H.; Knoll, A.C. Vision language models in autonomous driving: A survey and outlook. *IEEE Trans. Intell. Veh.* **2024**. <https://doi.org/10.1109/TIV.2024.3402136>
23. Viadero-Monasterio, F.; Meléndez-Useros, M.; Boada, B.L.; Boada, M.J.L. Static Output-Feedback LQR Control for Predecessor-Leader Following Vehicle Platooning with Energy Efficiency Analysis via Aerodynamic Drag Reduction. *IEEE Open J. Intell. Transp. Syst.* **2026**, *7*, 1329–1340.
24. Viadero-Monasterio, F.; Meléndez-Useros, M.; Jiménez-Salas, M.; Boada, M.J.L. Fault-Tolerant Robust Output-Feedback Control of a Vehicle Platoon Considering Measurement Noise and Road Disturbances. *IET Intell. Transp. Syst.* **2025**, *19*, e70007.
25. Viadero-Monasterio, F.; Meléndez-Useros, M.; Jiménez-Salas, M.; Boada, B.L. Robust adaptive heterogeneous vehicle platoon control based on disturbances estimation and compensation. *IEEE Access* **2024**, *12*, 96924–96935.

26. 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.* **2022**, *35*, 6119–6132.
27. 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, Vancouver, BC, Canada, 18–22 June 2023; pp. 17853–17862.
28. Zheng, W.; Song, R.; Guo, X.; Zhang, C.; Chen, L. Genad: Generative end-to-end autonomous driving. In *Proceedings of the European Conference on Computer Vision, Milan, Italy, 29 September–4 October 2024*; Springer Nature: Cham, Switzerland, 2024; pp. 87–104.
29. Chang, H.S.; Learned-Miller, E.; McCallum, A. Active bias: Training more accurate neural networks by emphasizing high variance samples. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1003–1013.
30. Neves, D.E.; Ishitani, L.; do Patrocínio Júnior, Z.K.G. Advances and challenges in learning from experience replay. *Artif. Intell. Rev.* **2024**, *58*, 54.

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.