Beyond Open-loop Metrics: Closed-loop Testing for End-to-End Autonomous Driving

Yi Xu, Wentao Luo, Xun Zhao
Department of Computer Science and Engineering
Southern University of Science and Technology (SUSTech)
Shenzhen, China

***Abstract*—This paper explores closed-loop testing for end-to-end autonomous driving models, focusing on whether such models can be stably and reproducibly evaluated under interactive environments. As a proof-of-concept, we implement a real-time closed-loop testing prototype on a simulation platform, and analyze the behavior of end-to-end models under continuous control feedback. Our study highlights that models achieving strong open-loop performance may still exhibit instability, error accumulation, or failure modes when deployed in closed-loop conditions, motivating the need for closed-loop evaluation paradigms.**

***Index Terms*—End-to-End Autonomous Driving, Closed-loop Testing, Model Evaluation, Simulation-based Testing**

I. Introduction

End-to-end autonomous driving has recently gained significant attention as an alternative to traditional modular pipelines, due to its potential to jointly optimize perception, prediction, and planning within a unified learning framework. Moreover, end-to-end autonomous driving provides a lossless paradigm for information to go through the whole model.

By directly mapping sensor observations to driving actions, end-to-end models promise improved scalability and reduced system complexity, and have demonstrated strong performance on several large-scale autonomous driving benchmarks.

Despite this progress, the evaluation methodology for end-to-end autonomous driving models has not evolved at the same pace. Current research predominantly relies on **open-loop evaluation** protocols conducted on static datasets such as `nuScenes` [2], where model performance is measured using perception-oriented metrics including `AP` and `NDS`. While these metrics are effective for assessing detection accuracy under fixed distributions, they fail to capture how a model behaves when its outputs continuously influence future observations. In particular, open-loop evaluation does not verify whether the model's decisions **interact** with the dynamic environment in a coherent manner, as it never observes the consequences of the model's actions on future states or on other agents.

In real-world deployment, autonomous driving systems operate in a **closed-loop** manner, where control actions alter the environment state and errors may accumulate over time. Under such conditions, small prediction deviations can compound into significant behavioral instability or task failure. However, these temporal dependencies, state drifts, and feedback-induced errors are inherently invisible to open-loop evaluation, raising concerns about whether strong benchmark performance necessarily translates to reliable closed-loop behavior.

Motivated by this gap, this work focuses on closed-loop testing as an evaluation paradigm for end-to-end autonomous driving models. Rather than proposing new driving architectures or improving training objectives, we aim to investigate how end-to-end models behave under interactive closed-loop conditions, and what additional insights such evaluation can provide beyond conventional open-loop metrics.

To this end, we construct a closed-loop testing prototype based on the Carla simulation platform [3], enabling continuous perception–action–environment feedback for end-to-end driving models. Through a set of case studies involving both a parking-oriented end-to-end model and a state-of-the-art driving model evaluated without domain adaptation, we demonstrate that models with strong open-loop performance may still exhibit instability, error accumulation, or even complete failure when deployed in closed-loop settings. These observations highlight the importance of closed-loop testing for assessing model generalization and robustness.

In summary, this work makes the following contributions:

- We analyze the limitations of open-loop evaluation for end-to-end autonomous driving,
- We present a simulation-based closed-loop testing testbed that enables stable and reproducible evaluation, and
- We provide empirical case studies illustrating the behavioral discrepancies between open-loop and closed-loop performance.

II. Background & Motivation

A. End-to-End Autonomous Driving and Current Evaluation Practice

Traditional autonomous driving systems are typically designed as modular pipelines, where perception, prediction, planning, and control are developed and optimized separately. In contrast, end-to-end autonomous driving aims to learn a direct mapping from sensor observations to driving actions, enabling joint optimization across the entire decision-making process. Recent advances in large-scale datasets and deep learning architectures have significantly accelerated the development of end-to-end driving models, leading to strong performance on public benchmarks.

To evaluate such models, current research predominantly adopts open-loop evaluation protocols based on static datasets, most notably nuScenes [2]. Under this setting, model outputs are compared against pre-recorded ground truth trajectories or annotations, and performance is summarized using metrics such as detection mAP, planning L2 error, and the nuScenes Detection Score (NDS). These metrics have become the de facto standard for comparing end-to-end models and driving-related components, providing a convenient and reproducible evaluation framework.

B. *Limitations of Open-loop Metrics*

Despite their widespread adoption, open-loop evaluation metrics suffer from inherent limitations when applied to end-to-end autonomous driving models.

First, open-loop evaluation assumes a fixed data distribution, where model predictions do not influence future observations. This assumption fundamentally breaks down in real-world driving, where actions taken by the vehicle directly alter the environment state and subsequent sensory inputs. Moreover, open-loop metrics fail to capture the interactive nature of driving. As a result, open-loop evaluation cannot assess whether a model's decisions lead to coherent and safe interactions with dynamic agents.

Second, open-loop metrics are insensitive to temporal dependencies and error accumulation. Small prediction deviations that appear negligible under per-frame or short-horizon evaluation may compound over time in a closed-loop setting, eventually leading to unsafe or unstable behavior. As a result, models that achieve strong open-loop scores may still fail to maintain consistent and reliable driving performance when deployed in interactive environments.

Finally, open-loop evaluation provides limited insight into a model's generalization behavior. High performance on static datasets may reflect overfitting to specific data distributions, sensor configurations, or annotation biases, creating an illusion of robustness that does not necessarily transfer to unseen scenarios or dynamic interactions.

C. *Motivation for Closed-loop Testing*

In contrast to open-loop evaluation, closed-loop testing evaluates autonomous driving models under continuous interaction between perception, decision-making, and environment dynamics. By allowing model outputs to affect future states, closed-loop testing naturally captures temporal consistency, cumulative errors, and behavioral stability—properties that are essential for real-world deployment but largely invisible to static evaluation protocols.

Motivated by these observations, this work argues that closed-loop testing should be considered a primary evaluation paradigm for end-to-end autonomous driving models, rather than a secondary validation step. Our goal is not to replace existing open-loop benchmarks, but to complement them with closed-loop evaluation that provides deeper insights into model behavior, generalization, and failure modes under realistic operating conditions.

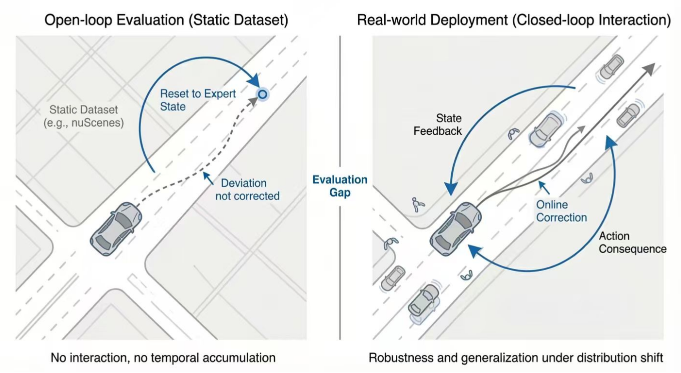

Open-loop Evaluation (Static Dataset)
Real-world Deployment (Closed-loop Interaction)
Static Dataset
(e.g., nuScenes)
Reset to Expert
State
Deviation
not corrected
Evaluation
Gap
State
Feedback
Online
Correction
Action
Consequence
No interaction, no temporal accumulation
Robustness and generalization under distribution shift

Fig. 1: Comparison between open-loop log replay and closed-loop interactive evaluation pipelines

III. Closed-loop Evaluation Paradigm

A. *Capability Coverage in Evaluation Paradigms*

To formally compare open-loop and closed-loop evaluation paradigms for end-to-end autonomous driving models, we abstract a set of capability dimensions that reflect fundamentally different aspects of model behavior that may be relevant for real-world deployment. Open-loop evaluation, which is typically conducted on static datasets such as nuScenes and measured using metrics like per-frame L2 error, detection mAP, or the nuScenes Detection Score (NDS), does not account for the effects of model decisions on subsequent observations [7] [8]. This limitation arises because open-loop metrics treat each sample independently and ignore temporal dependencies and feedback effects inherent in closed-loop control.

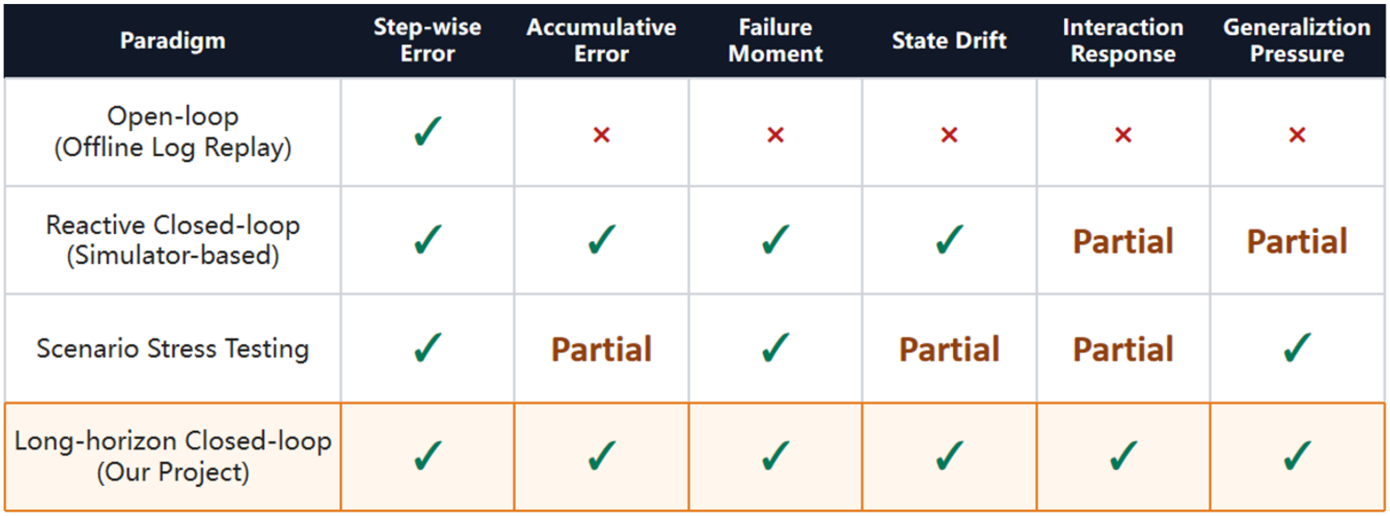
Closed-loop evaluation, by contrast, places the driving model in an interactive environment where predictions influence future sensory inputs and vehicle states. This interactive process enables the assessment of properties such as error accumulation, recovery behavior, and long-horizon performance, which are essential for evaluating real autonomous driving performance yet invisible to open-loop metrics [9] [10].

Table I summarizes the coverage of selected capability dimensions by the two evaluation paradigms. Each dimension characterizes a distinct aspect of model behavior or evaluation requirement, and closed-loop evaluation generally provides broader coverage across these aspects.

B. *Discussion of Capability Dimensions*

a) Environmental interaction feedback.: Open-loop evaluation inherently assumes a static dataset and thus does not consider how a model's output would affect the future state of the vehicle or the environment. In contrast, closed-loop evaluation explicitly models this interaction, enabling the assessment of how the system behaves under influence from its own actions [8].

b) Temporal continuity and dependency.: Many autonomous driving tasks involve sequences of decisions over time. Because open-loop metrics compute errors on independent samples, they fail to capture temporal consistency and the



Paradigm	Step-wise Error	Accumulative Error	Failure Moment	State Drift	Interaction Response	Generaliztion Pressure
Open-loop (Offline Log Replay)	✓	×	×	×	×	×
Reactive Closed-loop (Simulator-based)	✓	✓	✓	✓	Partial	Partial
Scenario Stress Testing	✓	Partial	✓	Partial	Partial	✓
Long-horizon Closed-loop (Our Project)	✓	✓	✓	✓	✓	✓

TABLE I: **Comparison of Evaluation Signals Across Autonomous Driving Testing Paradigms**

impact of past decisions on future performance. Closed-loop evaluation, by maintaining continuity across time steps, can reveal instabilities that accumulate over long horizons [7].

c) Error accumulation and recovery behavior.: Small deviations that appear acceptable in isolated frames can compound into significant divergence when the model is placed in a dynamic feedback loop. Closed-loop paradigms are designed to reveal such compounding effects, as well as the model's ability to recover from deviations, which open-loop evaluation cannot measure [9].

d) Long-horizon task performance.: Open-loop metrics often center on short-horizon predictions or per-frame metrics. They do not evaluate how the model performs over an extended sequence of decisions. Closed-loop testing allows for long-horizon performance measurement, which is critical to tasks such as complete route following or complex maneuvers [10].

e) Robustness and perturbation sensitivity.: Closed-loop evaluation enables controlled perturbations and scenario variations, making it possible to assess how robust a model is under different environmental conditions, while open-loop testing is often limited to fixed dataset distributions.

f) Causal chain interpretability and model improvement guidance.: By observing the evolution of a model's behavior in response to its own actions, closed-loop evaluation offers richer information for diagnosing failures and guiding model or architecture improvements. This aspect is largely absent in open-loop settings focused solely on static similarity metrics.

IV. Related Work

A. End-to-End Autonomous Driving and Open-Loop Evaluation

The paradigm of autonomous driving has witnessed a significant shift from modular pipelines to end-to-end (E2E) learning-based approaches. Unlike traditional systems that isolate perception, prediction, and planning, E2E models aim to learn a direct mapping from raw sensor inputs to control outputs. Historically, the evaluation of these models has been dominated by open-loop testing on large-scale datasets such as nuScenes [2].

In this standard open-loop setting, the autonomous system is assessed against pre-recorded, deterministic driving logs. The system's generated trajectories are compared to ground truth expert data using geometric metrics, primarily *L2 Displacement Error* and *Collision Rate* (calculated against logged future actor positions). For perception-centric subtasks, metrics like *mean Average Precision (mAP)* and the *nuScenes Detection Score (NDS)* are utilized. While these metrics have driven progress by providing a standardized and reproducible comparison protocol, they fundamentally treat driving as a series of independent regression problems rather than a sequential decision-making process.

B. The Pitfalls of Open-Loop Assessment

The research community increasingly acknowledges that open-loop metrics are insufficient proxies for real-world driving proficiency [7], [11], [13]. This insufficiency stems from several critical limitations inherent to offline evaluation.

1) Causal Confusion: A significant issue in imitation learning-based E2E models is "causal confusion," where models learn spurious correlations rather than true causal factors. For example, a model may learn to associate the illumination of the ego-vehicle's brake light with the action of braking (the "Brake Light Illusion"), or learn to remain stationary solely because zero acceleration correlates with the presence of other stationary vehicles (the "Inertia Problem"). Open-loop metrics often fail to penalize these shortcuts, as the model's predictions align with the log data despite flawed reasoning.

2) Covariate Shift and Cumulative Error: Open-loop training and evaluation suffer from distribution shift, also known as covariate shift. During offline evaluation, if a model makes a minor prediction error, the simulation resets the ego-state to the ground truth in the subsequent frame. This "teacher forcing" effectively masks the error. However, in a real-world closed-loop setting, minor errors compound, pushing the vehicle into unfamiliar states (off-distribution) that were not covered in the expert training data. This cascade often leads to catastrophic failure, a phenomenon that static L2 metrics cannot capture [5], [6].

3) Lack of Interactivity: Open-loop evaluation ignores the interactive nature of driving. Since other road users in the log cannot react to the ego-vehicle's deviations, metrics cannot measure whether a planner's action would induce dangerous behavior in others or successfully negotiate a merge.

C. Emergence of Closed-Loop Benchmarks

To overcome these limitations, the field is transitioning toward closed-loop simulation, where the agent's actions directly influence future environmental states.

1) CARLA and Standardized Leaderboards: The CARLA simulator [3] has become a de-facto standard for vision-based closed-loop evaluation. The CARLA Leaderboard ranks agents based on a *Driving Score (DS)*, a composite metric of Route Completion and infraction penalties. Recent frameworks such as **Bench2Drive** [9] and **DriveE2E** [10] have extended this by incorporating diverse, short-horizon scenarios derived from real-world logs (e.g., cut-ins, overtaking) to rigorously test capability boundaries.

2) Data-Driven Simulation: The **nuPlan** benchmark [4] represents the first large-scale attempt to bridge the gap between dataset learning and closed-loop planning. Built on over 1200 hours of real-world data, nuPlan features a lightweight simulator with reactive agents (using IDM or learned behaviors). It moves beyond geometric accuracy to evaluate ”closed-loop testability,” utilizing metrics such as *Time-to-Collision (TTC)*, *Drivable Area Compliance*, and *Ride Comfort*.

Despite these advancements, a gap remains in the diagnostic accessibility of these tools. While closed-loop benchmarks indicate *when* a model fails, systematically diagnosing *why*—specifically regarding the discrepancy between open-loop validation scores and closed-loop deployment failures—remains an active area of research.

V. Closed-loop Testbed and Diagnostic Evaluation

In this section, we present the design of our closed-loop evaluation testbed together with empirical case studies. Unlike traditional system-oriented papers that decouple platform design from experimental results, our testbed is intrinsically diagnostic: its value is manifested through the observable behaviors and failure modes of evaluated models under closed-loop interaction. Accordingly, we organize this section by progressively coupling the evaluation setup with representative model behaviors, demonstrating how closed-loop execution enables signals that are inaccessible under open-loop metrics.

A. Closed-loop Evaluation Setup

Our closed-loop evaluation is conducted in the CARLA simulator, where the ego vehicle interacts continuously with the environment without state reset or ground-truth correction. At each timestep, the simulator provides the ego vehicle with sensor observations and ego-state information, which are forwarded to the evaluated end-to-end driving model. The model outputs low-level control commands (steering, throttle, brake), which are executed immediately to advance the simulator state.

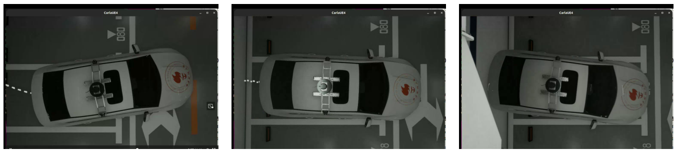
Crucially, unlike open-loop log replay, the ego state is not replaced by logged ground truth at subsequent timesteps. As a result, small control deviations are allowed to accumulate naturally over time, potentially leading to state drift, loss of controllability, or task failure. This design explicitly exposes temporal dependency and cumulative error, which are masked by teacher-forced evaluation.

To ensure reproducibility, all experiments are conducted with fixed random seeds and deterministic simulator configurations. During execution, we record ego trajectories, control commands, collision events, and termination conditions at each timestep. These logs form the basis for subsequent diagnostic analysis.

B. ParkingE2E: Long-horizon Closed-loop Behavior and Failure Modes

We first integrate ParkingE2E [14] into our testbed to examine its long-horizon closed-loop behavior under continuous perception–action feedback. In many initial conditions, ParkingE2E completes the parking maneuver and converges to the target slot, indicating that the testbed supports sustained execution over extended horizons rather than collapsing within only a few steps.

Importantly, closed-loop execution also reveals failure modes that are difficult to diagnose from open-loop metrics. In the SUSTech CoE parking lot, we observe that under certain initial poses, small control biases (e.g., slight steering offsets) can accumulate over time, gradually inducing state drift such as yaw misalignment and lateral deviation. Although these deviations may appear minor at early timesteps, the closed-loop feedback loop amplifies them, eventually resulting in lane violation or collision. Fig. 2 illustrates a representative failure case from the initial state to the collision moment.



(a) Start (b) Drift (c) Collision

Fig. 2: A failure case of ParkingE2E in the SUSTech CoE parking lot under closed-loop execution.

This case study highlights the diagnostic value of closed-loop testing: it enables us to analyze *where* failures occur (sensitivity to initial conditions), *when* failures become inevitable (failure horizon), and *how* they form (drift and error accumulation through feedback), which are largely invisible to per-step open-loop displacement errors.

C. SparseDrive: Building the Integration Pipeline and Closed-loop Findings

This case study serves a dual purpose. **(Engineering)** We build a practical dataflow pipeline that enables running an

TABLE II: Closed-loop Parking Task Outcomes for ParkingE2E tested in SUSTech CoE ParkingLot

Outcome Type	Percentage
Successful Parking	≈78%
Lane Violation	≈17%
Collision	≈5%

open-loop nuScenes model (SparseDrive) inside an interactive CARLA closed-loop loop. **(Evaluation)** Using this executable pipeline, we observe that a strong open-loop SOTA model may still be unstable or even *not testable* in closed-loop without adaptation.

1) Our Engineering Deliverable: A CARLA→nuScenes Dataflow Pipeline: To make SparseDrive [12] runnable in CARLA, we implement an end-to-end integration pipeline that bridges simulator signals to the model's expected nuScenes-style inputs. Concretely, our deliverable is a reusable workflow that converts CARLA runtime data into a nuScenes-like schema *at each timestep*, enabling online inference and closed-loop execution. 3.

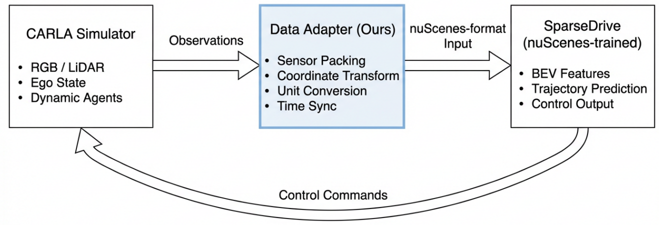


Fig. 3: Overview of the closed-loop evaluation pipeline bridging CARLA and nuScenes-based end-to-end driving models.

a) Sensor packaging and schema conversion: We package multi-view camera images and ego states from CARLA into a nuScenes-style input format, including synchronized frames, calibrated camera intrinsics/extrinsics, and timestamped ego status. This step ensures SparseDrive receives inputs consistent with its training/evaluation interface.

b) Coordinate-frame and unit alignment: We implement explicit coordinate and convention alignment (ego/world frames, heading/yaw definition, units, and sign conventions) to avoid silent mismatches that would otherwise dominate closed-loop behavior. This alignment is essential to ensure that failures observed in closed-loop reflect model limitations rather than interface bugs.

c) Online execution loop and logging for diagnosis: We build a real-time inference loop that (i) feeds converted observations into SparseDrive, (ii) applies the predicted controls to CARLA, and (iii) logs trajectories, control commands, and termination events. These logs make the closed-loop behavior reproducible and analyzable (e.g., drift growth and failure horizon).

TABLE III: Illustrative Closed-loop Stability Statistics for Driving Models in SUSTech CoE ParkingLot, relative stability trends.

Model	TTF (s)	DTF (m)	Collision Rate
SparseDrive (w/o adaptation)	4.6 ± 1.9	3.2 ± 1.1	92%

Overall, this pipeline is a core outcome of our project: it makes it possible to *execute* and *evaluate* an open-loop nuScenes model under closed-loop interaction in CARLA.

2) Closed-loop Findings: Open-loop SOTA May Fail to be Testable: With the above pipeline, SparseDrive can be executed in CARLA, but we observe severe instability when no domain-specific adaptation is applied. In many runs, the model diverges within a short horizon, producing control outputs that rapidly drive the ego vehicle into unrecoverable states. Typical failure patterns include oscillatory steering, excessive acceleration, and quick loss of lane adherence. Fig. 4 shows a representative failure case.

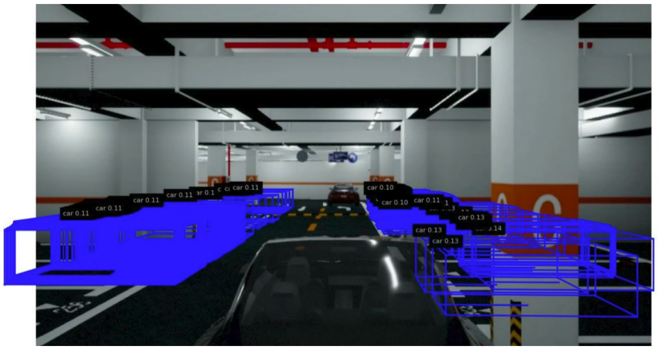


Fig. 4: Representative closed-loop failure: SparseDrive exhibits rapid divergence in CARLA without adaptation.

From the perspective of our evaluation paradigm, this result is not merely a negative outcome. **The inability to sustain closed-loop execution is itself a diagnostic signal:** high open-loop scores do not guarantee stable interactive behavior, and a model may be effectively *not testable* under closed-loop settings when distribution shift and feedback effects are present. This finding reinforces our central argument that closed-loop testing provides behavior-level signals (e.g., drift dynamics and failure horizon) that are fundamentally invisible to open-loop log replay.

D. Implications of Closed-loop Testability

Rather than treating this outcome as a defect of the testbed or an implementation issue, we interpret it as an informative evaluation result. The inability to maintain closed-loop execution itself constitutes a critical diagnostic signal, revealing a gap between open-loop validation and closed-loop deployability. This observation underscores that high open-loop scores do not guarantee that a model is even *testable* under realistic closed-loop conditions.

E. Summary of Observed Closed-loop Behaviors

Across the evaluated models, the closed-loop testbed exposes several behavior-level signals that are inaccessible under open-loop metrics. These include error accumulation over time, state drift induced by feedback, failure horizons, and qualitative differences in recoverability. While ParkingE2E demonstrates stable long-horizon interaction, SparseDrive highlights how models optimized for open-loop benchmarks may fail to sustain closed-loop execution altogether.

These findings reinforce the necessity of closed-loop testing as a complementary evaluation paradigm. Rather than replacing open-loop metrics, closed-loop evaluation provides a diagnostic lens for analyzing stability, generalization, and deployability—properties that are critical for real-world autonomous driving but remain obscured by static dataset scores.

VI. Conclusion & Future Work

In this work, we revisit the evaluation methodology of end-to-end autonomous driving from the perspective of closed-loop interaction. We argue that the prevailing reliance on open-loop metrics, such as displacement error and perception-centric scores on static datasets, provides an incomplete and potentially misleading assessment of real-world driving capability [7], [11]. By decoupling prediction accuracy from interactive execution, open-loop evaluation fundamentally masks temporal instability, error accumulation, and state drift.

To address this gap, we propose a closed-loop testing paradigm that emphasizes *testability* rather than leaderboard performance. We design a diagnostic closed-loop testbed and demonstrate, through empirical case studies, that closed-loop execution exposes behavior-level signals that are inaccessible under open-loop evaluation. Our experiments show that models capable of achieving strong open-loop scores may nevertheless fail to sustain stable closed-loop interaction, while models designed for long-horizon control exhibit fundamentally different stability characteristics that are invisible to static metrics.

Importantly, our findings do not suggest replacing open-loop benchmarks. Instead, we advocate for closed-loop testing as a complementary evaluation lens that reveals temporal robustness, recoverability, and generalization under interaction. Such signals are essential for understanding whether an end-to-end driving model is suitable for deployment beyond offline validation.

Looking forward, this work opens several directions for future research. First, the closed-loop testbed can be extended with more diverse and systematically perturbed scenarios, enabling controlled stress testing across a broader range of environmental and behavioral conditions. Second, integrating domain-adaptive simulation and learned reactive agents may further narrow the gap between closed-loop evaluation and real-world deployment. Finally, future work may explore how insights from closed-loop diagnostic signals can be fed back into model design, training objectives, or curriculum learning, thereby closing the loop between evaluation and improvement.

We believe that advancing autonomous driving requires not only more powerful models, but also more faithful and diagnostic evaluation paradigms. Closed-loop testing provides a critical step toward this goal by transforming evaluation from static score comparison into dynamic behavior analysis.

References

[1] Y. Hu, J. Li, Z. Li, X. Jia, P. Liu, Y. Zhu, L. Yuan, L. Wang, X. Li, Y. Cao, et al., "Planning-oriented autonomous driving," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 17634–17644.

[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11621–11631.

[3] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. on Robot Learning (CoRL)*, 2017, pp. 1–16.

[4] H. Caesar, R. Choudhury, K. Shabalina, E. Tuts, T. Fervers, H. Müller, W. K. Punn, S. Suo, S. Al-Stouhi, S. Vora, et al., "nuPlan: A closed-loop ML-based planning benchmark for autonomous driving," in *Proc. Conf. on Robot Learning (CoRL)*, 2021, pp. 1660–1672.

[5] S. Ross, G. Gordon, and J. A. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proc. Int. Conf. on Artificial Intelligence and Statistics (AISTATS)*, 2011, pp. 627–635.

[6] L. Gauerhof, H. Müller, K. Stensbo-Smidt, T. B. Moeslund, and K. Shabalina, "EpileptiCar: A vicious cycle of perception and control in autonomous driving," in *Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024, pp. 14836–14845.

[7] J.-T. Zhai et al., "Rethinking the open-loop evaluation of end-to-end autonomous driving in nuScenes," arXiv preprint, 2023.

[8] NVIDIA Autonomous Driving Research, "Beyond behavior cloning in autonomous driving: A survey of closed-loop training techniques," arXiv preprint, 2025.

[9] X. Jia et al., "Bench2Drive: Towards multi-ability benchmarking of closed-loop end-to-end autonomous driving," arXiv preprint, 2024.

[10] H. Yu et al., "DriveE2E: Closed-loop benchmark for end-to-end autonomous driving through real-to-simulation," arXiv preprint, 2025.

[11] Z. Li, Z. Yu, S. Lan, J. Li, J. Kautz, T. Lu, and J. M. Alvarez, "Is Ego Status All You Need for Open-Loop End-to-End Autonomous Driving?" arXiv preprint arXiv:2312.03031v2, 2024.

[12] W. Sun, X. Lin, Y. Shi, C. Zhang, H. Wu, and S. Zheng, "SparseDrive: End-to-End Autonomous Driving via Sparse Scene Representation," arXiv preprint arXiv:2405.19620v2, 2024.

[13] L. Chen, H. Li, P. Wu, K. Chitta, B. Jaeger, and A. Geiger, "End-to-End Autonomous Driving: Challenges and Frontiers," arXiv preprint arXiv:2306.16927v3, 2024.

[14] C. Li, Z. Ji, Z. Chen, T. Qin, and M. Yang, "ParkingE2E: Camera-based End-to-end Parking Network, from Images to Planning," arXiv preprint arXiv:2408.02061v1, 2024.