Pseudo-Simulation for Autonomous Driving

Wei Cao[*3,5] **Marcel Hallgarten**[*1,3,6] **Tianyu Li**[*4]
Daniel Dauner[1] **Xunjiang Gu**[6] **Caojun Wang**[4] **Yakov Miron**[3]
Marco Aiello[5] **Hongyang Li**[4] **Igor Gilitschenski**[6,7] **Boris Ivanovic**[2]
Marco Pavone[2,8] **Andreas Geiger**[1] **Kashyap Chitta**[1,2]
[1]University of Tübingen, Tübingen AI Center [2]NVIDIA Research [3]Robert Bosch GmbH
[4]OpenDriveLab at Shanghai Innovation Institute [5]University of Stuttgart
[6]University of Toronto [7]Vector Institute [8]Stanford University

Abstract: Existing evaluation paradigms for Autonomous Vehicles (AVs) face critical limitations. Real-world evaluation is often challenging due to safety concerns and a lack of reproducibility, whereas closed-loop simulation can face insufficient realism or high computational costs. Open-loop evaluation, while being efficient and data-driven, relies on metrics that generally overlook compounding errors. In this paper, we propose *pseudo-simulation*, a novel paradigm that addresses these limitations. Pseudo-simulation operates on real datasets, similar to open-loop evaluation, but augments them with synthetic observations generated prior to evaluation using 3D Gaussian Splatting. Our key idea is to approximate potential future states the AV might encounter by generating a diverse set of observations that vary in position, heading, and speed. Our method then assigns a higher importance to synthetic observations that best match the AV's likely behavior using a novel proximity-based weighting scheme. This enables evaluating error recovery and the mitigation of causal confusion, as in closed-loop benchmarks, without requiring sequential interactive simulation. We show that pseudo-simulation is better correlated with closed-loop simulations ($R^2 = 0.8$) than the best existing open-loop approach ($R^2 = 0.7$). We also establish a public leaderboard for the community to benchmark new methodologies with pseudo-simulation. Our code is available at https://github.com/autonomousvision/navsim.

1 Introduction

Reliable evaluation is essential for developing decision-making systems. In the context of autonomous vehicles (AVs), this means assessing the system's ability to navigate complex traffic scenarios efficiently, comfortably, and safely. Existing evaluation strategies typically fall into two categories: closed-loop and open-loop evaluation [1].

Closed-loop evaluation assesses model performance by placing it in an interactive environment. The AV must safely navigate traffic while making progress toward a designated goal. Although real-world closed-loop deployment offers reliable feedback, it is costly, risky, and not reproducible, making it insufficient on its own for benchmarking at the scale needed to demonstrate robustness [2]. As a more reproducible alternative, closed-loop evaluation is often conducted in simulation [3, 4]. Simulators enable rapid iteration and controlled scenario generation, and provide structured metrics for downstream performance analysis, such as collision or route completion rates.

However, accurate simulation remains a significant challenge, particularly for vision-based end-to-end AV systems. Real-world driving is visually complex and behaviorally diverse, making it difficult to replicate in simulation. Most existing platforms are manually constructed by 3D artists and engineers [3, 5]. This limits their realism and the diversity across scenes. Moreover, simulation-based evaluation is a computationally intensive and inherently sequential process. It often relies on large amounts of correlated evaluation frame sequences due to the high frequency of simulation required to ensure fidelity (usually 10Hz or higher).

Primary contact: kchitta@nvidia.com. *Equal contribution.

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

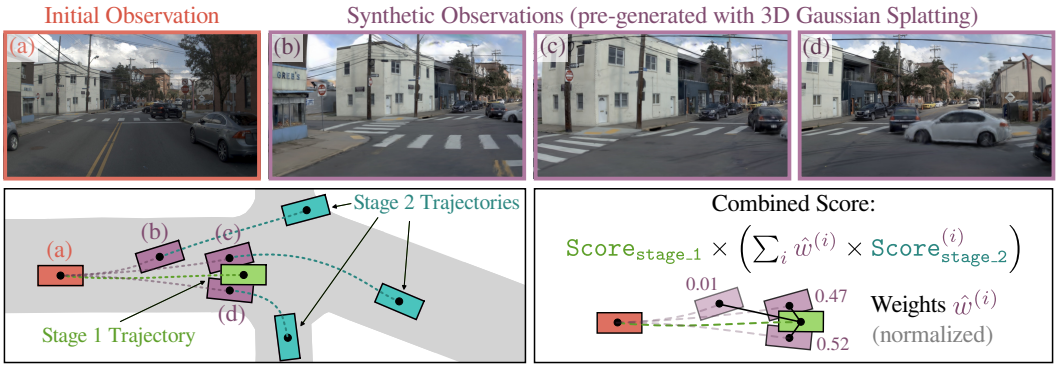

Figure 1: **Pseudo-simulation. (Top)** From an initial real-world observation (a), we generate synthetic observations (b, c, d) via a variant of 3D Gaussian Splatting specialized for driving scenes [9]. Crucially, these synthetic observations are **pre-generated prior to evaluation**, unlike traditional interactive simulation where observations are generated online. **(Bottom)** Pseudo-simulation involves two stages. In Stage 1, we evaluate the AV's trajectory output for (a). Stage 2 involves evaluation on trajectories output for (b, c, d). Stage 2 scores are weighted ($\hat{w}^{(i)}$) based on the proximity of the Stage 2 synthetic observation start point to the Stage 1 planned endpoint. The aggregated score assesses robustness to small variations near the intended path, prioritizing the most likely futures.

Open-loop evaluation, on the other hand, measures planning performance by comparing predicted trajectories to expert demonstrations in pre-recorded datasets. Each observation for evaluation includes sensor inputs, a goal location, and the future trajectory executed by a human expert driver. The AV predicts a fixed-horizon trajectory conditioned on the inputs, which is then scored against the expert using either displacement errors or metrics derived from ground-truth (GT) environment annotations, such as lane compliance or estimated collisions [6, 7, 8]. This approach operates entirely on real sensor data and avoids the complexities of interactive simulation, making it scalable and straightforward to apply over large datasets. However, it evaluates behavior only under expert-aligned conditions and does not account for distribution shifts. In deployment, the AV deviates from the demonstrated path, and open-loop protocols do not test its ability to recover from such drift.

To address the limitations of existing evaluation protocols, we introduce **pseudo-simulation**. This new paradigm aims to combine the scalability of open-loop evaluation with a comprehensive assessment traditionally restricted to interactive closed-loop testing. As shown in Fig. 1, our approach evaluates the AV's performance in two stages. Stage 1 uses the originally recorded real-world observations. Stage 2 uses synthetic observations generated based on these original frames. Crucially, these synthetic observations are generated before the evaluation process begins, enabling evaluation in a non-interactive manner. To generate Stage 2 observations, we adapt (to our data) a state-of-the-art driving scene reconstruction and rendering algorithm [9] based on 3D Gaussian Splatting [10].

We evaluate the output trajectories predicted by the AV, considering its performance on both the initial real-world observations (from Stage 1) and the generated synthetic observations (used in Stage 2). Our key idea lies in how we assess performance in Stage 2: we weight the importance of each synthetic observation based on its proximity to the endpoint of the trajectory that the AV initially predicted in Stage 1 (Fig. 1 bottom-right). This weighting strategy allows the evaluation to better reflect the AV's robustness and ability to recover from potential errors when facing conditions similar to what it may encounter in a closed-loop simulation. Our approach also inherently assigns lower weights to synthetic observations that significantly deviate from the initially predicted endpoint, thereby preventing undue penalties for failures in improbable or irrelevant future states.

In summary, pseudo-simulation combines **real** and **pre-rendered synthetic** data, enabling **scalable, parallel** evaluation.

Property	Open-Loop	Closed-Loop	Pseudo-Sim
Real data	Yes	No	Yes
Per-scene pre-processing	No	Yes	Yes
Synthetic data rendering	No	Online	Pre-rendered
Evaluation	Parallel	Sequential	Parallel
Compounding errors	No	Yes	Yes
Causal confusion	No	Yes	Yes

We show that pseudo-simulation achieves strong correlation with closed-loop results for a set of 83 diverse planners nu-

Plan [4], while being substantially more efficient (6× less environment interactions). To enable standardized benchmarking, we release NAVSIM v2, a framework for benchmarking autonomous driving built upon our proposed evaluation methodology. We find that it reveals previously unknown failure modes in popular AV algorithms [11, 12], thus establishing it as a challenging new testbed for future research. We hope that pseudo-simulation can accelerate AV development through more efficient experimentation cycles and ensuring that future models prioritize robustness.

2 Related Work

Counterfactual Data Augmentation. Counterfactual augmentation has been used to expose models to out-of-distribution data by generating structured perturbations [13, 14, 15]. Related work for AVs focuses on augmenting training data with viewpoint shifts [16, 17, 18, 19, 20, 21, 22, 23]. We make the first attempt to adopt such augmentations primarily for evaluation.

Closed-Loop Benchmarking. Graphics-based simulators support closed-loop evaluation, but are computationally expensive and introduce domain gaps in sensor fidelity [3, 5]. To improve scalability, data-driven planning simulators leverage recorded traffic data [4, 24, 25, 26]. However, these systems operate at the trajectory level and do not support sensor-based agents. Several works attempt to bridge this gap through data-driven sensor simulation, generating synthetic views from real-world logs. Early systems simulate ego-vehicle deviations via image-based rendering [27, 28, 29]. More recent methods explore neural rendering [30, 31, 32, 33, 34, 35]. However, they face challenges related to photorealism and runtime efficiency. As such, there is no widely established neural rendering based AV benchmark yet, which we aim to address in this work.

Open-Loop Benchmarking. Open-loop evaluation typically measures planning quality via displacement errors between predicted and expert trajectories [36]. These metrics are simple to compute, but often correlate poorly with real-world performance and tend to favor trivial or history-based baselines [7, 12, 37, 38]. Furthermore, benchmarks adopting the nuScenes dataset [39] exhibit inconsistencies in implementations for metrics such as ADE (Average Displacement Error) and collisions [6, 7], and overrepresent low-complexity scenes such as straight driving [7]. The closest work to ours, NAVSIM v1 [8], offers a more structured framework for open-loop benchmarking. The agent-under-test predicts a fixed-horizon trajectory from real sensor inputs, while other actors replay their recorded motion. This setup supports scalable evaluation and enables simulation-based metrics such as progress and collision rates. However, unlike pseudo-simulation, NAVSIM v1 remains limited to open-loop evaluation from expert-aligned initial observations and does not account for compounding errors or causal confusion [40].

3 Pseudo-Simulation

We consider a planning task where evaluation proceeds in two stages (Fig. 1). In both stages, an AV (also called planner/ego agent) generates a 4-second trajectory based on sensor inputs and a driving command [8]. The inputs include multi-view camera images and ego status features such as the velocity and motion history. The driving command specifies the intended maneuver in case of ambiguity, e.g., at intersections, and is provided as a discrete label: *left*, *straight*, or *right*. The ego agent outputs a trajectory (i.e., a sequence of desired future waypoints) in its local coordinate frame.

3.1 Stage 1: Initial Observations

In Stage 1, we infer the ego agent's motion based on an initial observation from the test dataset. We then simulate a simplified Bird's Eye View (BEV) representation of the scene forward for a fixed time horizon, obtaining a score as well as an endpoint to be used later in Stage 2.

BEV Simulation. The 4-second trajectory predicted by the agent is executed using a kinematic bicycle model [41] and an LQR controller [42] at 10Hz. The trajectory is committed for the entire simulation horizon, and no closed-loop feedback is provided to the agent during this time. Unlike

related prior work [8], which uses non-reactive traffic to simplify implementation (i.e., neighboring vehicles follow their recorded trajectories without reacting to the ego agent), we improve the simulation realism with reactive traffic. Background vehicles (represented as oriented bounding boxes) respond to the ego agent using a rule-based planner called the Intelligent Driver Model (IDM) [43].

Extended PDM Score. Our metric, the Extended Predictive Driver Model Score (EPDMS) [44], builds on the PDMS introduced in prior work [8]. Besides minor modifications (detailed in the supplementary material), the design of the metric is largely consistent with [44]. It combines multiplicative penalties for rule violations with a weighted average of several subscores:

$$\text{EPDMS} = \underbrace{\prod_{m \in \mathcal{M}_{\text{pen}}} \text{filter}_m(\text{agent}, \text{human})}_{\text{penalty terms}} \cdot \underbrace{\frac{\sum_{m \in \mathcal{M}_{\text{avg}}} w_m \cdot \text{filter}_m(\text{agent}, \text{human})}{\sum_{m \in \mathcal{M}_{\text{avg}}} w_m}}_{\text{weighted average terms}} \tag{1}$$

Here, $\mathcal{M}_{\text{pen}} = \{\text{NC}, \text{DAC}, \text{DDC}, \text{TLC}\}$ and $\mathcal{M}_{\text{avg}} = \{\text{TTC}, \text{EP}, \text{HC}, \text{LK}, \text{EC}\}$ (Table 1). Unlike prior work [44], to prevent penalizing contextually justified maneuvers, we introduce a novel filtering mechanism (filter_m) for the EPDMS. If a rule violation is also committed by the human expert driver in the same scene, the penalty is ignored. This avoids penalizing infractions due to label noise or valid behaviors, such as briefly entering the opposite lane to bypass a static obstacle.

Subscore	w_m	**Range**
No at-fault Coll. (NC)	-	$\{0, \frac{1}{2}, 1\}$
Drivable Area Compl. (DAC)	-	$\{0, 1\}$
Driving Direction Compl. (DDC)	-	$\{0, \frac{1}{2}, 1\}$
Traffic Light Compl. (TLC)	-	$\{0, 1\}$
Ego Progress (EP)	5	$[0, 1]$
Time to Collision (TTC)	5	$\{0, 1\}$
Lane Keeping (LK)	2	$\{0, 1\}$
History Comfort (HC)	2	$\{0, 1\}$
Extended Comfort (EC)	2	$\{0, 1\}$

Table 1: **EPDMS.** Subscores, weights, and ranges.

3.2 Stage 2: Synthetic Observations

In Stage 2, the agent's behavior is inferred on pre-generated synthetic observations. The scoring pipeline from Stage 1 is repeated for each of these synthetic observations. Stage 2 scores correspond to a range of plausible futures. We propose to weight their contributions towards a final combined score based on the proximity of Stage 2 start points to the Stage 1 endpoint. This prioritizes futures that are more likely. We show some examples of such generated scenes in Fig. 2. In the following, we provide details regarding the scenario pre-generation, scoring, and score aggregation processes. Note that we choose to create synthetic observations after unrolling for 4 seconds, instead of directly at the Stage 1 start point, since (1) this allows background traffic to react to the updated ego state, and (2) it provides a physically plausible history trajectory, which is a required planner input.

Start Point Sampling. As a data pre-processing step prior to the evaluation of any specific planner, we generate Stage 2 synthetic observations that approximate the range of possible rollout endpoints for Stage 1 observations in the dataset. Each Stage 2 observation must have a valid start point and heading, with an associated motion history, and multi-view camera image inputs for a planner.

We sample start points around the expert driver's observed endpoint after 4 seconds in the scene. Importantly, this sampling does not depend on the Stage 1 endpoint produced by a planner, but only the expert driving trajectories from the original dataset, available prior to evaluation. We define a sampling region around this expert endpoint: laterally, viewpoints are sampled every 0.5 meters up to 2.0 meters on each side; longitudinally, viewpoints are sampled every 5.0 meters. The longitudinal sampling spans the physically plausible range from the minimum stopping distance to the maximum reachable distance (assuming accelerations of ± 4.0 m/s^2 for 4 seconds). This naturally produces more potential states for high-speed scenarios (up to 20 in practice) compared to low-speed ones.

Heading and History Generation. For each sampled start point, we generate a plausible heading and motion history by matching it to the nearest trajectory in a human driving dataset. This matching process includes filtering: we discard candidate trajectories if they differ in velocity by more than 1.0 m/s, acceleration by more than 1.0 m/s^2, or heading by more than 20 degrees relative to the

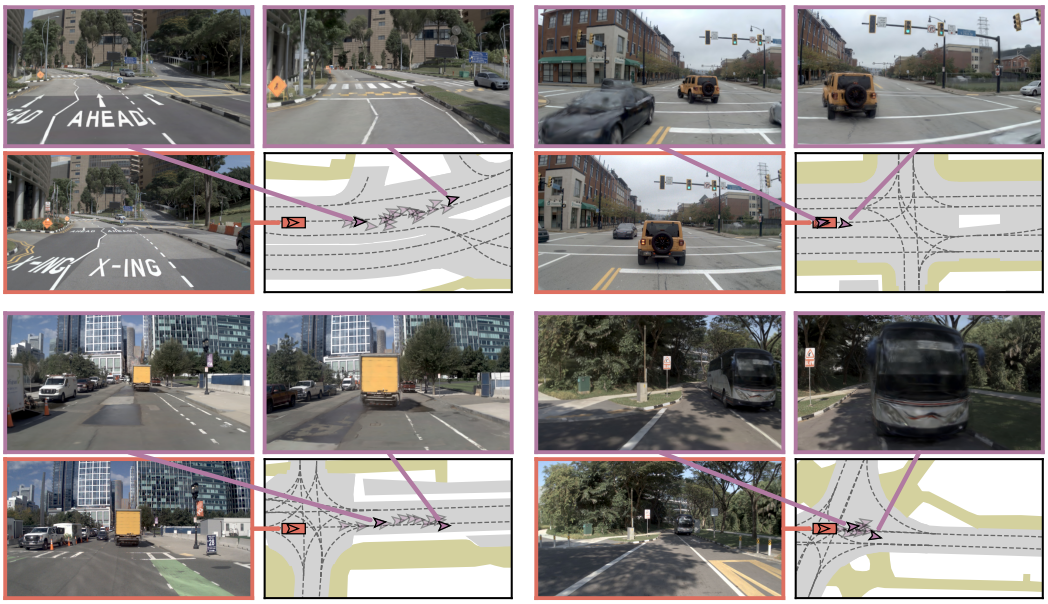

Figure 2: **Example scenes.** We show the poses and front-view camera images for the initial real-world observation (➤) and pre-generated synthetic observations (➤) in four scenes.

expert. We then apply rejection sampling to remove any remaining start points that violate the multiplicative EPDMS constraints (NC, DAC, DDC and TLC). Finally, we discard scenes from the neural reconstruction pipeline if fewer than five valid synthetic observations remain after filtering.

Neural Reconstruction and Rendering. We employ a state-of-the-art dynamic scene reconstruction approach to achieve high-fidelity neural rendering. Specifically, we use a modified version of Multi-Traversal Gaussian Splatting (MTGS) [9]. As in MTGS, we model scene dynamics with a scene graph [45]. However, unlike MTGS, which uses multiple nearby driving traversals for jointly optimizing a 3D scene representation, we use only a single traversal. This significantly expands the pool usable data, as only a subset of our dataset includes multiple co-located traversals. To reduce localization noise, we calculate accurate initial camera pose estimates via LiDAR registration [46] and bundle adjustment [47], followed by camera pose optimization during the MTGS training process [48]. Before reconstruction, we filter out scenes affected by significant sensor failures (water droplets or flares). After reconstruction, we apply a semi-automatic filtering step to discard reconstructed scenes of low visual quality (details in supplementary material).

Score Aggregation. We score each synthetic observation using the EPDMS from Eq. (1) to obtain Stage 2 scores $\{s_2^i\}$. To compute the final score s_{combined}, we define two aggregation functions: $s_{\text{combined}} = \mathcal{A}_1(s_1, s_2)$, where $s_2 = \mathcal{A}_2(\{s_2^i\}, \{x^i\}, \hat{x})$. $\mathcal{A}_1$ fuses the Stage 1 score s_1 with an aggregated Stage 2 result s_2. $\mathcal{A}_2$, in turn, aggregates $\{s_2^i\}$ based on their initial positions $\{x^i\}$, which denote the start points of the i-th Stage 2 scenario. $\hat{x}$ is the ego agent's endpoint reached at the end of the Stage 1 simulation. In our experiments, we conduct an empirical study on different aggregation functions. Based on our findings, we instantiate $\mathcal{A}_1$ as a simple product, and $\mathcal{A}_2$ as a Gaussian-weighted average with kernel variance σ^2:

$$s_{\text{combined}} = s_1 s_2, \quad s_2 = \sum_i \hat{w}^i s_2^i, \quad \hat{w}^i = \frac{w^i}{\sum_i w^i}, \quad w^i = \exp\left(-\frac{\|x^i - \hat{x}\|^2}{2\sigma^2}\right) \tag{2}$$

4 Results

4.1 How well-aligned is pseudo-simulation with closed-loop evaluation?

Benchmark. To evaluate how well pseudo-simulation aligns with closed-loop simulation, we conduct a correlation analysis with the nuPlan simulator [4]. nuPlan supports fully reactive rollouts for

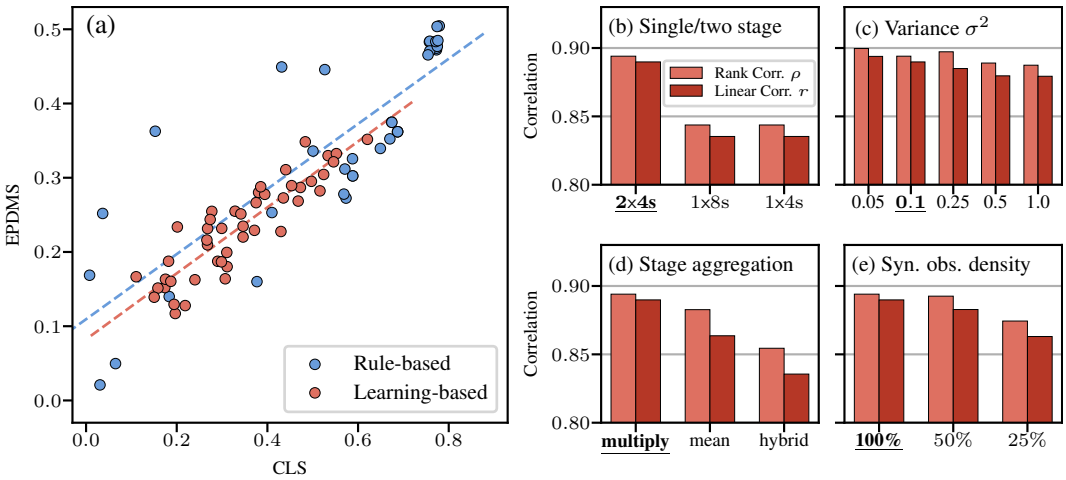

Figure 3: **Correlations.** (a) Correlation between the default pseudo-simulation metric (EPDMS) and the closed-loop score (CLS) for a set of 37 rule-based and 46 learned planners. We further compare (b) single (1x) vs. two stage (2x) evaluation, (c) Gaussian weight variances, (d) Stage 1 and 2 aggregation methods, and (e) synthetic observation densities. Defaults in **bold-underline**.

privileged planners with access to ground-truth perception and HD maps. We include a total of 83 planners, comprising both rule-based and learned models, to represent a wide range of behaviors and performance levels. For rule-based methods, we use 10 constant kinematics baselines, 15 IDM planners [43], and 15 PDM-Closed variants [12]. For learned approaches, we evaluate 22 PlanCNN [49] models with varying input modalities and 24 Urban Driver [50] models differing in architecture and training configurations. For these experiments, we use a reduced version of EPDMS, excluding the TLC, LK, and EC metrics, because nuPlan does not support these for closed-loop evaluation.

We measure the alignment between EPDMS and nuPlan's closed-loop score (CLS) using Pearson's linear (r) and Spearman's rank (ρ) correlation coefficients, as well as the coefficient of determination (R^2). Since R^2 is calculated by fitting a linear model between EPDMS and CLS, it is equivalent to the square of Pearson's correlation coefficient here ($R^2 = r^2$). This assumes that an ideal pseudo-simulation metric should show a linear relationship with closed-loop scores, requiring no adjustments for scale or bias. We evaluate each planner on a filtered subset of nuPlan, described in detail in the supplementary material, to collect both closed-loop and pseudo-simulation scores. This subset includes 244 initial observations (Stage 1) and 4164 synthetic observations (Stage 2).

Results. First, we create a scatter plot comparing the 8-second closed-loop scores from the nuPlan simulator against the results of our 2×4 second pseudo-simulation, which aims to approximate these closed-loop scores. As shown in Fig. 3 (a), pseudo-simulation exhibits strong correlation with closed-loop results across a broad range of planners, particularly among learned planners.

In Fig. 3 (b), we compare single-stage open-loop simulation (at 4 and 8 seconds) to our two-stage pseudo-simulation variant. The two-stage setup achieves significantly higher alignment, reaching a Pearson correlation of $r = 0.89$ (corresponding to $R^2 = 0.8$), compared to $r = 0.83$ ($R^2 = 0.7$) for the single-stage baselines. Furthermore, compared to standard reactive closed-loop evaluation, our pseudo-simulation method exposes a wider range of potential failures. This typically results in lower average EPDMS values compared to CLS values. By injecting synthetic deviations, pseudo-simulation effectively reveals edge cases that might not be encountered during standard testing.

Within Stage 2, we assess the impact of the weighting used to combine scores across synthetic viewpoints. Fig. 3 (c) shows the correlation with closed-loop scores for different kernel variances. We observe that smaller variances lead to improved results. $\sigma^2 = 0.05$ and our default configuration of $\sigma^2 = 0.1$ give the highest correlations. In additional experiments (included in the supplementary material), we find that other approaches, such as simple averaging, k-nearest neighbors (k-NN), and hybrid k-NN/Gaussian weighting are less effective than our default configuration.

To combine metrics across stages, we compare multiplicative aggregation, aggregation by the arithmetic mean, and a hybrid strategy where penalty metrics (e.g., collision and drivable area compliance) are multiplied while the remaining terms are averaged. Fig. 3 (d) summarizes the correlation of each strategy with closed-loop scores. Multiplicative aggregation shows both higher linear and rank correlation than the other approaches. This outcome is likely because most subscores are binary (i.e., 0 or 1). Consequently, multiplication appears to be a more suitable method for estimating the overall score for an 8-second interval based on two 4-second segments.

Finally, we examine the effect of limiting the number of synthetic views in Stage 2. Fig. 3 (e) reports correlation values when using 100%, 50%, and 25% of our available synthetic viewpoints. At 100% density, each scenario contains 12 synthetic observations on average in Stage 2 for each real observation in Stage 1, resulting in 13 planner inferences per scenario. In comparison, closed-loop simulation in nuPlan requires 80 planner inferences per scenario, corresponding to an 8-second rollout at 10Hz. This is 6× higher than pseudo-simulation. While subsampling reduces the number of synthetic views, the correlation to closed-loop scores remains strong. Even when using only 25% density, i.e., approximately three Stage 2 observations per scene, the correlation remains above 0.85. This indicates that pseudo-simulation maintains reliability even with reduced observation coverage.

4.2 What new challenges and insights does our leaderboard provide?

Benchmark. Our public NAVSIM v2 leaderboard features challenging driving scenarios (e.g. unprotected turns and dense traffic, see Fig. 2). It uses a subset of nuPlan that we refer to as `navhard`, involving 450 Stage 1 and 5462 Stage 2 observations. Further details are in the supplementary.

We select four baseline planners with varying input modalities. These include the Constant Velocity (CV) and Ego-history MLP from [8], Latent TransFuser (LTF) [11], the strongest public image-only planner for nuPlan, and, PDM-Closed (PDM-C) [12], the best privileged planner on nuPlan.

Metric	Stage	CV [8]	MLP [8]	LTF [11]	*PDM-C [12]*
NC ↑	S1	88.8	93.2	**96.2**	*94.4*
	S2	**83.2**	77.2	77.7	*88.1*
DAC ↑	S1	42.8	55.7	**79.5**	*98.8*
	S2	59.1	51.9	**70.2**	*90.6*
DDC ↑	S1	70.6	86.6	**99.1**	*100*
	S2	76.5	74.4	**84.2**	*96.3*
TLC ↑	S1	99.3	99.3	**99.5**	*99.5*
	S2	98.0	**98.2**	98.0	*98.5*
EP ↑	S1	77.5	81.2	**84.1**	*100*
	S2	71.3	77.1	**85.1**	*100*
TTC ↑	S1	87.3	92.2	**95.1**	*93.5*
	S2	**81.1**	75.0	75.6	*83.1*
LK ↑	S1	78.6	83.5	**94.2**	*99.3*
	S2	**47.9**	40.8	45.4	*73.7*
HC ↑	S1	97.1	**97.5**	**97.5**	*87.7*
	S2	97.1	**97.8**	95.7	*91.5*
EC ↑	S1	60.4	77.7	**79.1**	*36.0*
	S2	61.9	**79.8**	75.9	*25.4*
EPDMS ↑		10.9	12.7	**23.1**	*51.3*

Table 2: `navhard` **leaderboard.**

Results. Table 2 presents the detailed subscores for each planner, broken down by Stage 1 (original observations) and Stage 2 (synthetic observations). In terms of overall performance, PDM-Closed achieves the highest combined EPDMS of 51.3, followed by LTF (23.1), MLP (12.7), and CV (10.9). Comparing performance across stages, we observe a general drop in subscores from Stage 1 to Stage 2 for all methods (particularly for LTF), suggesting their sensitivity to the distribution shifts introduced in Stage 2. Notably, while PDM-Closed excels in most metrics, it exhibits lower performance in comfort metrics such as HC and EC. Our evaluation on `navhard` reveals this specific failure mode of PDM-Closed, highlighting a trade-off that was overlooked in prior benchmarks. We host `navhard` as a public leaderboard.

4.3 Does the proposed neural rendering yield sufficient visual fidelity?

Benchmark. To assess the fidelity of our synthetic observations, we evaluate the impact of our neural rendering on the downstream perception and planning performance of a pre-trained model using

Data	Stage	Perception mIoU ↑	Planning EPDMS ↑
Real	S1	46.0	62.3
Syn.	S1	37.6	61.0
Syn.	S2	36.9	44.2

(a) **Synthetic data quality for downstream tasks.**

Method	LPIPS ↓
Street Gaussians [51]	0.354
Ours (w/o pose opt.)	0.322
Ours	0.253

(b) **NVS ablation study.**

Table 3: **Evaluation of synthetic observations and Novel View Synthesis (NVS).** (a) An end-to-end planner, LTF [11], is trained on real data and evaluated on both real and synthetic `navhard` views, measuring BEV perception (mIoU) and planning (EPDMS) quality. (b) NVS quality is evaluated across several methods using LPIPS on 8 scenes with 10Hz-alternating training and test views.

the `navhard` dataset. Specifically, we employ the LTF model from Table 2. Although primarily an end-to-end planner, LTF outputs intermediate Bird's Eye View (BEV) segmentations and, crucially, was trained only on real-world data. This allows us to measure the domain gap introduced by our rendering: we evaluate LTF's performance on synthetic data and compare it to its performance on real data. We use mean Intersection over Union (mIoU) over the drivable area, walkway, and vehicle classes output by LTF to evaluate BEV perception, and the EPDMS metric to evaluate planning.

Results. We present our findings in Table 3a. First, we evaluate perception performance using the LTF model. Comparing Stage 1 to Stage 2 views, we observe a drop in mIoU from 46.0 to 37.6. Despite this degradation in segmentation quality, planning performance remains largely stable, with EPDMS decreasing only slightly from 62.3 to 61.0. While mIoU captures semantic segmentation fidelity, it does not directly reflect planner-relevant errors [52, 53]. For our data, the observed reduction in mIoU does not appear to impair semantic cues needed for planning. This suggests that our synthetic observations preserve the most critical information.

Next, we evaluate performance under perturbed synthetic Stage 2 inputs. Here, mIoU drops marginally from 37.6 to 36.9, while EPDMS declines substantially to 44.2. This larger drop is consistent with trends observed previously (Section 4.2), where non-privileged planners showed greater sensitivity to deviations from expert trajectories. The small change in mIoU between the synthetic settings of Stage 1 and Stage 2, compared to the greater drop in EPDMS, suggests that the observed planning degradation is primarily driven by the planner's sensitivity to the distribution shift, rather than by perception inaccuracies stemming from rendering artifacts.

Ablation Study. Additionally, we evaluate novel view synthesis fidelity using the LPIPS metric [54] on 8 `navhard` scenes, where lower scores indicate higher perceptual similarity [55]. Here, the training and test viewpoints were sampled at alternating 10Hz intervals from the expert trajectory to ensure disjoint inputs and outputs for evaluation. As shown in Table 3b, the baseline Street Gaussians method [51] obtains an LPIPS of 0.354. Our MTGS-based variant [9] without optimizations improves this score to 0.322, while our full method incorporating LiDAR registration, bundle adjustment, and pose optimization achieves the best LPIPS of 0.253. Combining these results with the perception and planning evaluations in Table 3a, we believe our neural rendering pipeline provides sufficient visual fidelity to approximate planning evaluation as in closed-loop settings.

5 Conclusion

We introduce pseudo-simulation, a new evaluation paradigm which demonstrates a high correlation to computationally expensive closed-loop simulations. Our experiments show how it better captures crucial aspects of AV evaluation like error recovery than open-loop evaluation. Pseudo-simulation offers significant potential impacts for AV development. It enables more efficient iteration cycles, promotes system robustness by rigorously testing sensitivity to perturbations, and ultimately enhances safety through more comprehensive evaluations. We hope our public `navhard` benchmark, featuring pre-rendered data and standardized metrics via an online leaderboard, can foster community adoption of pseudo-simulation for standardized comparisons of AV systems.

Limitations and Future Work

While pseudo-simulation demonstrates strong correlation with closed-loop evaluation and offers advantages over existing paradigms, we acknowledge several limitations:

Correlation with Real-World Deployment. Our current validation focuses on establishing correlation with established simulation benchmarks. We do not yet demonstrate or claim direct correlation with performance metrics from real-world vehicle deployment. Bridging this gap between simulation-based evaluation and predicting real-world outcomes remains an important direction for future investigation. Rather than replacing real-world validation, frameworks to augment real-world evaluations with simulation can be applied more effectively with our work [56]

Pre-Processing Computational Cost. The current pipeline relies on a per-scene optimization process (based on MTGS) to generate the synthetic views, requiring approximately 1-2 hours per scene on current hardware. While manageable for our dataset scale (under 1000 scenes), this computational cost limits scalability for extremely large datasets. Exploring recent advancements in potentially faster, feedforward 3D scene representation and rendering methods could offer a path towards significantly reducing this overhead in the future [57, 58].

Rendering Fidelity and Evaluation. Despite achieving excellent quantitative results on rendering fidelity (LPIPS) and downstream task performance (mIoU, EPDMS), some visual artifacts may persist in the generated synthetic views. Our evaluation primarily focuses on algorithmic metrics. Future work may also benefit from incorporating human perceptual studies to gain a more comprehensive understanding of perceived realism and the potential impact of any remaining artifacts. Furthermore, combining neural rendering techniques like ours with state-of-the-art generative diffusion models might offer possibilities for enhancing rendering quality [59, 60, 61, 62].

Background Traffic Realism. The current approach utilizes relatively simple, rule-based traffic models for background agents within the synthetic observations [63]. This results in these agents strictly following road-centerline paths during Stage 2 evaluation. In future work, we aim to incorporate more sophisticated, potentially learned, traffic models that can adapt background agent behavior dynamically based on the ego agent's actions [64]. Another possible extension is adversarial background traffic designed to further emphasize the need for robustness [65]. These extensions could enable the evaluation of more complex, interactive scenarios and improve evaluation fidelity without compromising the scalability of the pseudo-simulation approach.

Human-flag Filtering. Our filtering strategy disregards rule violations also committed by human experts. While this helps reduce false positives, it could also risk overlooking important failure and edge cases, since human driving is not always a gold standard for safety. Future work could further refine the human-flag filtering and explore this trade-off to ensure more reliable evaluation.

Metric Design Choices. We choose multiplicative aggregation because most sub-scores are binary-valued, and multiplication captures compounding failures, e.g., a collision should significantly impact the final score. Our Gaussian weighting is selected for its strong empirical performance with minimal assumptions. Exploring more principled formulations for aggregation and weighting remains an interesting future direction.

Acknowledgments

This work was supported by the ERC Starting Grant LEGO-3D (850533), the DFG EXC number 2064/1 - project number 390727645, the German Federal Ministry of Education and Research: Tübingen AI Center, FKZ: 01IS18039A and the German Federal Ministry for Economic Affairs and Climate Action within the project NXT GEN AI METHODS. We thank the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Daniel Dauner and Kashyap Chitta. We also thank HuggingFace for hosting our evaluation servers, the team members of OpenDriveLab for their organizational support, Maxim Dolgov for helpful discussions, as well as Napat Karnchanachari and the team from Motional for open-sourcing their dataset and providing us the private test split used in the 2025 NAVSIM Challenge.

References

[1] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li. End-to-end autonomous driving: Challenges and frontiers. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2023.

[2] M. Buehler, K. Iagnemma, and S. Singh. *The DARPA urban challenge: autonomous vehicles in city traffic*. Springer, 2009.

[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.

[4] N. Karnchanachari, D. Geromichalos, K. Seang Tan, N. Li, C. Eriksen, S. Yaghoubi, N. Mehdipour, G. Bernasconi, W. Kit Fong, Y. Guo, and H. Caesar. Towards learning-based planning: The nuPlan benchmark for real-world autonomous driving. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2024.

[5] Q. Li, Z. Peng, L. Feng, Q. Zhang, Z. Xue, and B. Zhou. Metadrive: Composing diverse driving scenarios for generalizable reinforcement learning. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2022.

[6] X. Weng, B. Ivanovic, Y. Wang, Y. Wang, and M. Pavone. Para-drive: Parallelized architecture for real-time autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[7] 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? In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[8] D. Dauner, M. Hallgarten, T. Li, X. Weng, Z. Huang, Z. Yang, H. Li, I. Gilitschenski, B. Ivanovic, M. Pavone, A. Geiger, and K. Chitta. Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[9] T. Li, Y. Qiu, Z. Wu, C. Lindström, P. Su, M. Nießner, and H. Li. MTGS: Multi-traversal gaussian splatting. *arXiv.org*, 2503.12552, 2025.

[10] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 2023.

[11] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger. TransFuser: Imitation with transformer-based sensor fusion for autonomous driving. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 2023.

[12] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta. Parting with misconceptions about learning-based vehicle motion planning. In *Proc. Conf. on Robot Learning (CoRL)*, 2023.

[13] S. Pitis, E. Creager, and A. Garg. Counterfactual data augmentation using locally factored dynamics. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:3976–3990, 2020.

[14] S. Pitis, E. Creager, A. Mandlekar, and A. Garg. Mocoda: Model-based counterfactual data augmentation. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:18143–18156, 2022.

[15] H. Chen, R. Xia, and J. Yu. Reinforced counterfactual data augmentation for dual sentiment classification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 269–278. Association for Computational Linguistics, 2021.

[16] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, et al. End to end learning for self-driving cars. *arXiv.org*, 1604.07316, 2016.

[17] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by cheating. In *Proc. Conf. on Robot Learning (CoRL)*, 2020.

[18] F. Codevilla, E. Santana, A. M. López, and A. Gaidon. Exploring the limitations of behavior cloning for autonomous driving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[19] A. Prakash, A. Behl, E. Ohn-Bar, K. Chitta, and A. Geiger. Exploring data aggregation in policy learning for vision-based urban autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[20] B. Jaeger, K. Chitta, and A. Geiger. Hidden biases of end-to-end driving models. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2023.

[21] E. Ma, L. Zhou, T. Tang, Z. Zhang, D. Han, J. Jiang, K. Zhan, P. Jia, X. Lang, H. Sun, et al. Unleashing generalization of end-to-end autonomous driving with controllable long video generation. *arXiv.org*, 2406.01349, 2024.

[22] J. Zimmerlin, J. Beißwenger, B. Jaeger, A. Geiger, and K. Chitta. Hidden biases of end-to-end driving datasets. *arXiv.org*, 2412.09602, 2024.

[23] K. Renz, L. Chen, E. Arani, and O. Sinavski. Simlingo: Vision-only closed-loop autonomous driving with language-action alignment. *arXiv.org*, 2503.09594, 2025.

[24] C. Gulino, J. Fu, W. Luo, G. Tucker, E. Bronstein, Y. Lu, J. Harb, X. Pan, Y. Wang, X. Chen, J. D. Co-Reyes, R. Agarwal, R. Roelofs, Y. Lu, N. Montali, P. Mougin, Z. Yang, B. White, A. Faust, R. McAllister, D. Anguelov, and B. Sapp. Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[25] E. Vinitsky, N. Lichtlé, X. Yang, B. Amos, and J. Foerster. Nocturne: a scalable driving benchmark for bringing multi-agent learning one step closer to the real world. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[26] S. Kazemkhani, A. Pandya, D. Cornelisse, B. Shacklett, and E. Vinitsky. Gpudrive: Data-driven, multi-agent driving simulation at 1 million fps. *arXiv.org*, 2408.01584, 2024.

[27] A. Amini, I. Gilitschenski, J. Phillips, J. Moseyko, R. Banerjee, S. Karaman, and D. Rus. Learning robust control policies for end-to-end autonomous driving from data-driven simulation. *IEEE Robotics and Automation Letters (RA-L)*, 2020.

[28] A. Amini, T.-H. Wang, I. Gilitschenski, W. Schwarting, Z. Liu, S. Han, S. Karaman, and D. Rus. Vista 2.0: An open, data-driven simulator for multimodal sensing and policy learning for autonomous vehicles. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2022.

[29] T.-H. Wang, A. Amini, W. Schwarting, I. Gilitschenski, S. Karaman, and D. Rus. Learning interactive driving policies via data-driven simulation. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2022.

[30] Z. Yang, Y. Chen, J. Wang, S. Manivasagam, W.-C. Ma, A. J. Yang, and R. Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[31] A. Tonderski, C. Lindström, G. Hess, W. Ljungbergh, L. Svensson, and C. Petersson. NeuRAD: Neural rendering for autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[32] W. Ljungbergh, A. Tonderski, J. Johnander, H. Caesar, K. Åström, M. Felsberg, and C. Petersson. Neuroncap: Photorealistic closed-loop safety testing for autonomous driving. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024.

[33] X. Yang, L. Wen, Y. Ma, J. Mei, X. Li, T. Wei, W. Lei, D. Fu, P. Cai, M. Dou, et al. Drivearena: A closed-loop generative simulation platform for autonomous driving. *arXiv.org*, 2408.00415, 2024.

[34] H. Zhou, L. Lin, J. Wang, Y. Lu, D. Bai, B. Liu, Y. Wang, A. Geiger, and Y. Liao. Hugsim: A real-time, photo-realistic and closed-loop simulator for autonomous driving. *arXiv.org*, 2412.01718, 2024.

[35] J. You, X. Jia, Z. Zhang, Y. Zhu, and J. Yan. Bench2drive-r: Turning real world data into reactive closed-loop autonomous driving benchmark by generative model. *arXiv.org*, 2412.09647, 2024.

[36] C. Sima, K. Renz, K. Chitta, L. Chen, H. Zhang, C. Xie, J. Beißwenger, P. Luo, A. Geiger, and H. Li. Drivelm: Driving with graph visual question answering. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024.

[37] F. Codevilla, A. M. Lopez, V. Koltun, and A. Dosovitskiy. On offline evaluation of vision-based driving models. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2018.

[38] J.-T. Zhai, Z. Feng, J. Du, Y. Mao, J.-J. Liu, Z. Tan, Y. Zhang, X. Ye, and J. Wang. Rethinking the open-loop evaluation of end-to-end autonomous driving in nuscenes. *arXiv.org*, 2305.10430, 2023.

[39] 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 Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[40] C. Wen, J. Lin, T. Darrell, D. Jayaraman, and Y. Gao. Fighting copycat agents in behavioral cloning from observation histories. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[41] R. Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[42] N. Lehtomaki, N. Sandell, and M. Athans. Robustness results in linear-quadratic gaussian based multivariable control designs. *IEEE Trans. on Automatic Control (TAC)*, 1981.

[43] M. Treiber, A. Hennecke, and D. Helbing. Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 2000.

[44] K. Li, Z. Li, S. Lan, Y. Xie, Z. Zhang, J. Liu, Z. Wu, Z. Yu, and J. M. Alvarez. Hydra-MDP++: Advancing End-to-End Driving via Expert-Guided Hydra-Distillation. *arXiv.org*, 2503.12820, 2025.

[45] J. Ost, F. Mannan, N. Thuerey, J. Knodt, and F. Heide. Neural scene graphs for dynamic scenes. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[46] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss. KISS-ICP: In Defense of Point-to-Point ICP – Simple, Accurate, and Robust Registration If Done the Right Way. *RA-L*, 2023.

[47] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[48] C. Yan, D. Qu, D. Xu, B. Zhao, Z. Wang, D. Wang, and X. Li. GS-SLAM: Dense visual slam with 3d gaussian splatting. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[49] K. Renz, K. Chitta, O.-B. Mercea, S. Koepke, Z. Akata, and A. Geiger. Plant: Explainable planning transformers via object-level representations. In *Proc. Conf. on Robot Learning (CoRL)*, 2022.

[50] O. Scheel, L. Bergamini, M. Wolczyk, B. Osiński, and P. Ondruska. Urban driver: Learning to drive from real-world demonstrations using policy gradients. In *Proc. Conf. on Robot Learning (CoRL)*, 2021.

[51] Y. Yan, H. Lin, C. Zhou, W. Wang, H. Sun, K. Zhan, X. Lang, X. Zhou, and S. Peng. Street Gaussians: Modeling dynamic urban scenes with gaussian splatting. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024.

[52] A. Behl, K. Chitta, A. Prakash, E. Ohn-Bar, and A. Geiger. Label efficient visual abstractions for autonomous driving. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2020.

[53] T. Schreier, K. Renz, A. Geiger, and K. Chitta. On offline evaluation of 3d object detection for autonomous driving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV) Workshops*, 2023.

[54] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[55] C. Lindström, G. Hess, A. Lilja, M. Fatemi, L. Hammarstrand, C. Petersson, and L. Svensson. Are nerfs ready for autonomous driving? towards closing the real-to-simulation gap. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[56] R. Luo, H. Yang, M. Watson, A. Sharma, S. Veer, E. Schmerling, and M. Pavone. Leveraging correlation across test platforms for variance-reduced metric estimation. *arXiv.org*, 2506.20553, 2025.

[57] J. Yang, J. Huang, Y. Chen, Y. Wang, B. Li, Y. You, M. Igl, A. Sharma, P. Karkus, D. Xu, B. Ivanovic, Y. Wang, and M. Pavone. Storm: Spatio-temporal reconstruction model for large-scale outdoor scenes. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2025.

[58] S. Miao, J. Huang, D. Bai, X. Yan, H. Zhou, Y. Wang, B. Liu, A. Geiger, and Y. Liao. Evolsplat: Efficient volume-based gaussian splatting for urban view synthesis. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[59] J. Yang, S. Gao, Y. Qiu, L. Chen, T. Li, B. Dai, K. Chitta, P. Wu, J. Zeng, P. Luo, J. Zhang, A. Geiger, Y. Qiao, and H. Li. Generalized Predictive Model for Autonomous Driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[60] S. Gao, J. Yang, L. Chen, K. Chitta, Y. Qiu, A. Geiger, J. Zhang, and H. Li. Vista: A generalizable driving world model with high fidelity and versatile controllability. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[61] J. Yang, K. Chitta, S. Gao, L. Chen, Y. Shao, X. Jia, H. Li, A. Geiger, X. Yue, and L. Chen. Resim: Reliable world simulation for autonomous drivingend to end learning for self-driving cars. *arXiv.org*, 2506.09981, 2025.

[62] J. Z. Wu, Y. Zhang, H. Turki, X. Ren, J. Gao, M. Z. Shou, S. Fidler, Z. Gojcic, and H. Ling. Difix3d+: Improving 3d reconstructions with single-step diffusion models. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[63] K. Chitta, D. Dauner, and A. Geiger. Sledge: Synthesizing driving environments with generative models and rule-based traffic. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2024.

[64] Z. Zhang, P. Karkus, M. Igl, W. Ding, Y. Chen, B. Ivanovic, and M. Pavone. Closed-loop supervised fine-tuning of tokenized traffic models. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2025.

[65] N. Hanselmann, K. Renz, K. Chitta, A. Bhattacharyya, and A. Geiger. King: Generating safety-critical driving scenarios for robust imitation via kinematics gradients. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.