Supplementary Material for:
Pseudo-Simulation for Autonomous Driving

Wei Cao*[5,6] **Marcel Hallgarten***[1,5,7] **Tianyu Li***[4]
Daniel Dauner[1,2] **Xunjiang Gu**[7] **Caojun Wang**[4] **Yakov Miron**[5]
Marco Aiello[6] **Hongyang Li**[4] **Igor Gilitschenski**[7,8] **Boris Ivanovic**[3]
Marco Pavone[3,9] **Andreas Geiger**[1,2] **Kashyap Chitta**[1,2,3]

[1]University of Tübingen [2]Tübingen AI Center [3]NVIDIA Research
[4]OpenDriveLab at Shanghai AI Lab [5]Robert Bosch GmbH [6]University of Stuttgart
[7]University of Toronto [8]Vector Institute [9]Stanford University

https://github.com/autonomousvision/navsim

1 Metrics

1.1 Metric Formulation

Our metric, the Extended Predictive Driver Model Score (EPDMS), builds on the PDMS formulation introduced in NAVSIM [1] and later adapted in Hydra-MDP++ [2], and condenses an agent's driving performance into a single aggregate score in $[0, 1]$.

$$\begin{aligned}
\texttt{EPDMS} &= \prod_{m \in \mathcal{M}_{\text{pen}}} \text{filter}_m(\text{agent}, \text{human}) \cdot \frac{\sum_{m \in \mathcal{M}_{\text{avg}}} w_m \cdot \text{filter}_m(\text{agent}, \text{human})}{\sum_{m \in \mathcal{M}_{\text{avg}}} w_m} \\
&= \underbrace{\text{filter}_{\texttt{NC}} \times \text{filter}_{\texttt{DAC}} \times \text{filter}_{\texttt{DDC}} \times \text{filter}_{\texttt{TLC}}}_{\text{penalty terms}} \\
&\quad \times \underbrace{\frac{1}{16}(5 \cdot \text{filter}_{\texttt{TTC}} + 5 \cdot \text{filter}_{\texttt{EP}} + 2 \cdot \text{filter}_{\texttt{LK}} + 2 \cdot \text{filter}_{\texttt{HC}} + 2 \cdot \text{filter}_{\texttt{EC}})}_{\text{weighted average terms}}
\end{aligned} \tag{1}$$

Where $\text{filter}_m(\text{agent}, \text{human})$ is defined as:

$$\text{filter}_m(\text{agent}, \text{human}) = \begin{cases} 1.0 & \text{if } m(\text{human}) = 0 \\ m(\text{agent}) & \text{otherwise} \end{cases} \tag{2}$$

Among the nine sub-metrics, NC, DAC, EP, and TTC are inherited directly from PDMS [1]. The metrics EC and TLC follow the definitions described in [2] and are reimplemented without modification. The metric LK also draws inspiration from [2], but is adapted in our work with a modified violation condition. The metric DDC is adapted from the formulation used in nuPlan [3], reimplemented with changes to suit our evaluation setting. In addition, we adjust the weights of both LK and EC to 2, in contrast to their configurations from [2], since these are generally more challenging, yet also less critical. The final sub-metric, HC, is a novel contribution introduced in this work. The remainder of this section provides detailed definitions for each sub-score.

1.2 Subscores Inherited from PDMS in NAVSIM [1]

No at-fault Collisions (NC) flags any collision initiated by the ego vehicle, distinguishing between impacts involving vulnerable road users and those involving static objects. *Drivable Area Compliance (DAC)* checks whether the ego vehicle remains within legally drivable regions, including lanes, intersections, and parking areas, throughout its trajectory. *Ego Progress (EP)* measures forward progress toward the navigation goal as a fraction of a safe upper-bound distance computed from a reference planner. *Time to Collision (TTC)* tracks the minimum predicted time to contact with any obstacle, enforcing a preset safety margin at each simulation step. Implementation details for these metrics follow [1] and are provided in their supplementary material.

1.3 Subscores Inherited from EPDMS in Hydra-MDP++ [2]

While our metric shares the same name (EPDMS) as that introduced in Hydra-MDP++[2], the two formulations differ in both sub-score weighting and aggregation. In particular, since it is not publicly available, we reimplement several subscores from [2], apply modifications to others, and adjust the final aggregation weight to better align with our pseudo-simulation setup. The subscores `EC` and `TLC` are reimplemented directly based on their original definitions. The metrics `LK` and `DDC` draw inspiration from prior work but are modified in our work to suit the evaluation context. For each subscore described below, we indicate whether it is a direct reimplementation or includes changes relative to the original specification.

1.3.1 Direct Reimplementations

This section describes metrics that we reimplement based on the descriptions in Hydra-MDP++ [2] without modification.

Extended Comfort (EC). The `EC` score checks that the ego-vehicle's predicted motion remains smooth across adjacent time steps. After generating a new trajectory at time step `t+1`, we overlap it with the trajectory from time step `t` and compute the root-mean-square (RMS) change in four ride-quality signals: linear acceleration d_A, linear jerk d_J, yaw rate d_Y^R, and yaw acceleration d_Y^A, over their common horizon. If all RMS deltas remain below predefined comfort thresholds, the transition is considered seamless and we assign $\texttt{score}_{\texttt{EC}} = 1$, otherwise $\texttt{score}_{\texttt{EC}} = 0$.

$$d_A = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(a_{\text{current},t} - a_{\text{preceding},t})^2}, \tag{3}$$

$$d_J = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(j_{\text{current},t} - j_{\text{preceding},t})^2}, \tag{4}$$

$$d_Y^R = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y^r_{\text{current},t} - y^r_{\text{preceding},t})^2}, \tag{5}$$

$$d_Y^A = \sqrt{\frac{1}{T}\sum_{t=1}^{T}(y^a_{\text{current},t} - y^a_{\text{preceding},t})^2}. \tag{6}$$

Using thresholds of $\tau_A = 0.7$ m/s^2, $\tau_J = 0.5$ m/s^3, $\tau_Y^R = 0.1$ rad/s, and $\tau_Y^A = 0.1$ rad/s^2, `EC` penalizes large kinematic changes between successive plans, enforcing temporal consistency and protecting passengers from abrupt motion.

Traffic Light Compliance (TLC). The `TLC` score ensures that the ego-vehicle respects traffic-light phases and enters intersections only on a valid green signal. In our implementation, each active red light is represented by a polygon tagged with a dedicated red-light token. At every simulation step, the metric checks whether any corner of the ego-vehicle's bounding box intersects one of these red-light polygons while the signal is red. A single violation at any point along the trajectory marks the proposal as non-compliant, assigning $\texttt{score}_{\texttt{TLC}} = 0$; if no red-light encroachments occur throughout the entire planning horizon, the metric assigns full credit with $\texttt{score}_{\texttt{TLC}} = 1$.

1.3.2 Subscore Modified from Hydra-MDP++

This section includes metrics that are based on Hydra-MDP++ [2], but are modified in our work to suit the pseudo-simulation setup better.

Lane Keeping (LK). This metric is motivated by [2], but we introduce a modification to the violation condition. The LK score checks whether the ego-vehicle follows the centerline of its current lane and avoids lingering between adjacent lanes, while also discouraging hesitant "half-commit" lane-change probes. Both behaviors are considered unsafe and discouraged in real-world traffic. In our implementation, we sample the lateral offset of the ego-vehicle's geometric center from the closest lane centerline at each simulation step. A violation is recorded only if the offset exceeds a fixed threshold (d = 0.5 m) continuously for more than 2 seconds, in contrast to [2], where any instantaneous deviation is penalized. Brief deviations are tolerated to account for decisive lane changes and intersection maneuvers. If no such sustained deviation occurs during the episode, the metric assigns $\texttt{score}_{\texttt{LK}} = 1$, or $\texttt{score}_{\texttt{LK}} = 0$ otherwise.

1.4 Subscore Adapted from nuPlan [3]

Driving Direction Compliance (DDC). The ego-vehicle must follow the legal lane direction and avoid traveling in oncoming lanes outside of intersections. In our implementation, we track the ego-vehicle's forward progress whenever its center is flagged as being in oncoming traffic and not within an intersection. Over a sliding horizon of 1 second, we accumulate the distance traveled against the intended traffic flow and record the maximum observed value, denoted as P_{oncoming}. We define the compliance and violation thresholds as $\tau_{\text{compliance}} = 2.0$ m and $\tau_{\text{violation}} = 6.0$ m, respectively. Compared to the original nuPlan implementation, we exclude intersections from the evaluation, as vehicles frequently cross between different lanes during turning or merging maneuvers. We apply the same exclusion in the Lane Keeping (LK) metric for consistency.

The corresponding score is thus calculated as:

$$\texttt{score}_{\texttt{DDC}} = \begin{cases} 1, & P_{\text{oncoming}} < \tau_{\text{compliance}}, \\ 0.5, & \tau_{\text{compliance}} \leq P_{\text{oncoming}} < \tau_{\text{violation}}, \\ 0, & \text{otherwise}, \end{cases}$$

1.5 New Subscore

This subscore is novel in this work and was not introduced in previous works.

History Comfort (HC). To obtain a realistic assessment of ride comfort, we prepend the planner's predicted trajectory with a short segment of historical motion from the human driver, using a fixed padding length of 1.5 seconds. The resulting continuous trajectory is then evaluated using the same comfort metric adopted in the nuPlan framework [3]. We compute ride-quality statistics and compare them against predefined human-derived thresholds. If all statistics remain within their respective limits, the episode is deemed comfortable and we assign $\texttt{score}_{\texttt{HC}} = 1$, or $\texttt{score}_{\texttt{HC}} = 0$ otherwise.

2 Datasets and Leaderboard

Our experiments are based on OpenScene [4], a downsampled redistribution of the nuPlan [3] dataset containing 120 hours of annotated urban driving at 2Hz. Each sample includes eight 1920×1080 camera views. Up to three past frames may be included, providing 1.5 seconds of history at 2Hz.

navhard Leaderboard. To support external benchmarking, we host a public leaderboard on Hugging Face using the `navhard` split. It is a filtered subset of OpenScene designed to support closed-loop and pseudo-simulation benchmarking. `navhard` includes 450 curated Stage 1 scenes selected semi-automatically for evaluation diversity, combining manual selection and failure mining for state-of-the-art planners. Along with the 450 real scenarios, the set includes 5462 pre-generated synthetic Stage 2 scenarios. Importantly, submissions to the leaderboard consist of predicted trajectories for each test frame and are evaluated server-side using the official EPDMS metrics. Therefore, unlike closed-loop leaderboards which require participants to submit entire models for evaluation, our leaderboard is much easier to scale, requiring only the submission of predictions.

navhard Correlation Subset. For our correlation analysis, we evaluate a diverse range of planners on the `navhard correlation subset`, a dedicated evaluation split subsampled from `navhard`. This subset includes 244 Stage 1 observations and 4164 Stage 2 synthetic observations, and is used in all correlation experiments described in Section 4.1. The subset was selected to ensure compatibility with both our pseudo-simulation and nuPlan [3], as certain scenes in the full `navhard` split are not uniformly supported across the two evaluation tools.

Neural Reconstruction. To pre-render images for novel view synthesis in Stage 2, we reconstruct the scene at 10 Hz using all available camera inputs at full resolution. For each selected scene, we collect images from a 4-second history to an 8-second future window relative to the current time. If the trajectory covered within this time window is shorter than 50 meters, we instead select images captured within a 50-meter spatial range. Subsequently, we manually filtered out scenes that could degrade reconstruction quality, including those affected by direct sunlight causing lens flare, water droplets on the camera surface, and highly reflective environments such as wet road surfaces. During reconstruction, all images are undistorted to a pinhole camera model. At inference time, we first render images under the pinhole model and then reapply the original distortion to simulate the characteristics of the real cameras. To avoid artifacts from the ego vehicle, regions corresponding to the ego car in the training images are masked out and excluded during rendering. When moving the ego vehicle and surrounding agents, we determine their 6-DOF poses by first estimating the local road plane from the nearest point on their original trajectories. After reconstruction, we apply a semi-automatic filtering step to discard reconstructed scenes with low visual quality. Specifically, we remove scenes with a PSNR below 27.0 or an LPIPS above 0.22. Subsequently, after rendering the stage-2 scenes from novel viewpoints, we manually filter out those exhibiting severe reconstruction artifacts, which are likely caused by erroneous pose registration.

3 Additional Results

3.1 Additional Visualizations of `navhard` scenes

In Fig. 1, we show a set of example scenes sampled from the `navhard` data split. In addition, Fig. 2 visualizes all eight surround-view camera renderings for a single synthetic pose. Our modified implementation of MTGS [5] renders photo-realistic camera observations even at pre-generated synthetic poses that deviate significantly from the original human-driven trajectory. While some characteristic artifacts of 3D Gaussian Splatting are still present, we find the synthetic views sufficiently realistic for our evaluation setting.

3.2 Qualitative Analysis of Subscore Violations

In this section, we present qualitative examples illustrating common failure modes for the reimplemented and modified subscores. Each example highlights a scenario where a specific subscore is violated. Red agents denote the ego vehicle, blue agents represent other traffic participants, and the red dashed line shows the predicted trajectory of the ego vehicle. These visualizations provide insight into the behavioral assumptions behind each metric and help contextualize their failure cases.

Extended Comfort (EC). Fig. 3 illustrates a representative failure case where consecutive predictions exhibit significant discontinuities. As shown in Fig. 4, the profiles for acceleration, jerk, yaw rate, and yaw acceleration differ substantially from the preceding prediction. The root-mean-square (RMS) deltas between overlapping trajectories exceed the predefined comfort thresholds. These abrupt changes from one planning step to the next can introduce sudden jolts, reduce ride comfort, and undermine confidence in the reliability of the autonomous driving system.

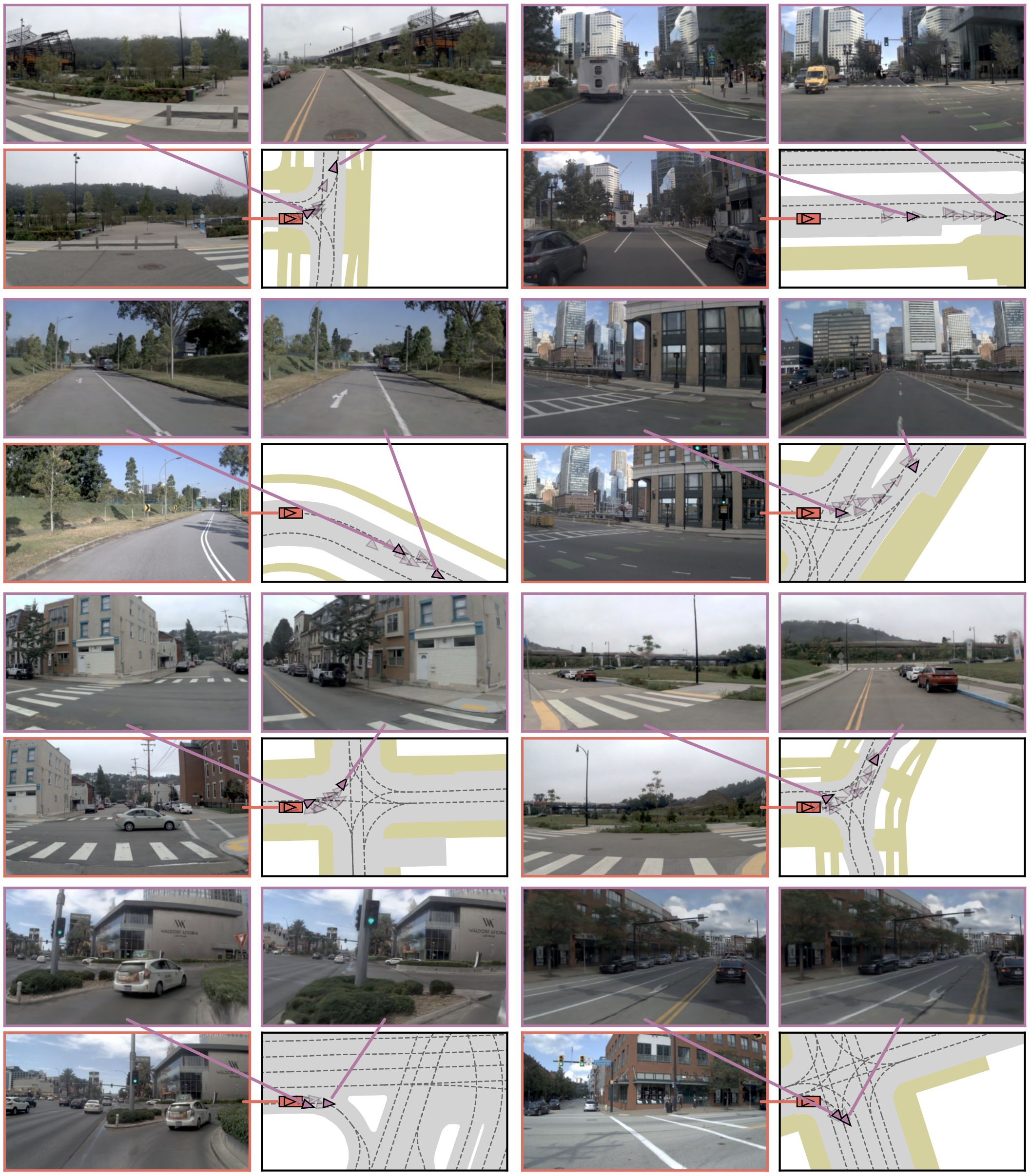

Figure 1: **Uncurated `navhard` scenes.** We show several randomly sampled scenes from the `navhard` split. We visualize the poses and front-view camera images for the initial real-world observation (⮞) and pre-generated synthetic observations (⮞).

Lane Keeping (LK). Fig. 5 shows a typical Lane Keeping failure case where the ego-vehicle deviates from the lane centerline for an extended duration. According to the metric definition, this constitutes a violation, as the lateral offset exceeds the 0.5 m threshold continuously for more than 2 seconds, resulting in $\texttt{score}_{\texttt{LK}} = 0$. This example highlights the importance of consistent lane positioning, since prolonged deviation can lead to unsafe encroachment into adjacent lanes or create ambiguity for other road users.

Driving Direction Compliance (DDC). Fig. 6 presents two clear Driving Direction Compliance failure cases where the ego-vehicle violates directional constraints by traveling against the intended traffic flow. In the left example, the vehicle enters oncoming lanes while navigating a curved road segment. The right example depicts a more hazardous situation involving proximity to another agent. In both cases, the ego-vehicle's accumulated distance traveled against traffic exceeds the violation

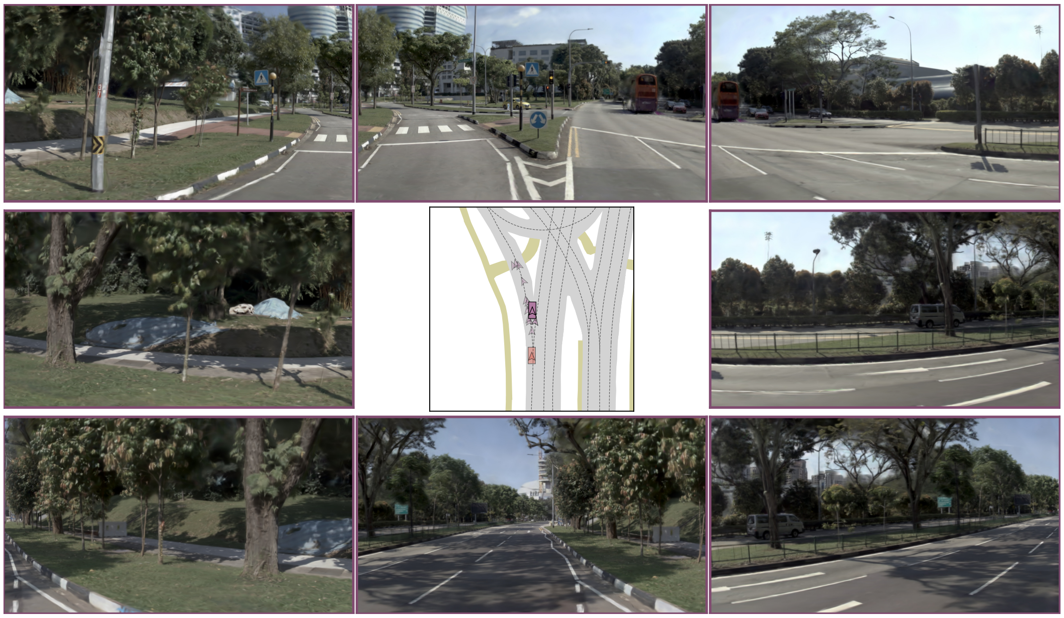

Figure 2: **Surround-view Synthetic Observation** The depicted sample from the `navhard` split shows all eight surround-view images at the pre-generated synthetic pose (➤).

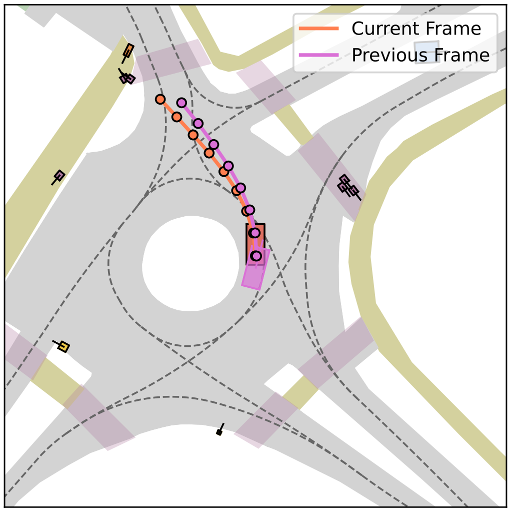

Figure 3: **BEV of Extended Comfort (EC) Failure Case.**

threshold, resulting in $\texttt{score}_{\texttt{DDC}} = 0$. These failures highlight the importance of directionality compliance, as such behavior in real-world settings introduces serious risk of head-on collisions.

Traffic Light Compliance (TLC). Fig. 7 illustrates a typical traffic light violation, where the ego-vehicle's predicted trajectory intersects a red-light polygon while the signal is active. Such violations pose a serious safety risk in real-world driving, increasing the likelihood of collisions with cross-traffic, disrupting traffic flow, and potentially leading to severe accidents.

History Comfort (HC). In our analysis shown in Fig. 8, we concatenate the historical human-driven trajectory with the model's predictions to enable continuous evaluation. As illustrated in Fig. 9, most comfort metrics remain within acceptable bounds, but the yaw acceleration exceeds the human-derived threshold precisely at the transition point from human control to autonomous behavior. This results in $\texttt{score}_{\texttt{HC}} = 0$, highlighting the difficulty of achieving seamless handovers between human and machine control.

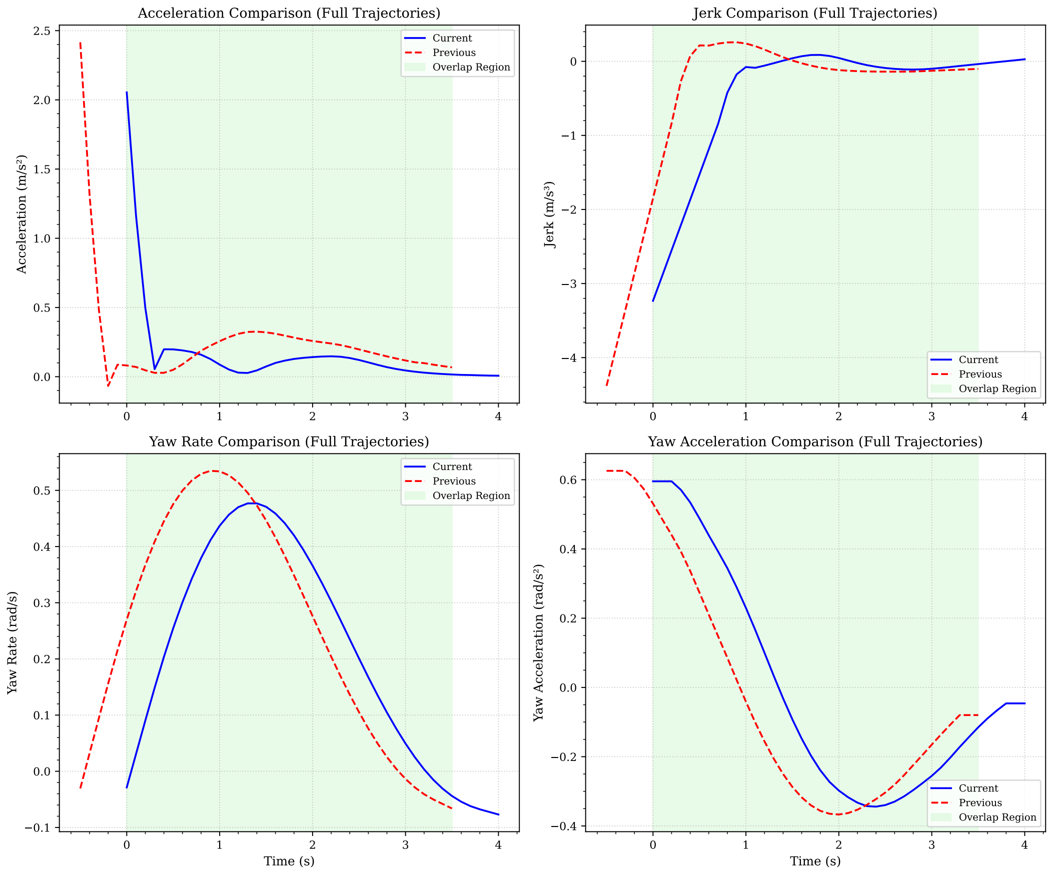

Figure 4: **Statistical Comparison of Extended Comfort (EC) Failure Case.**

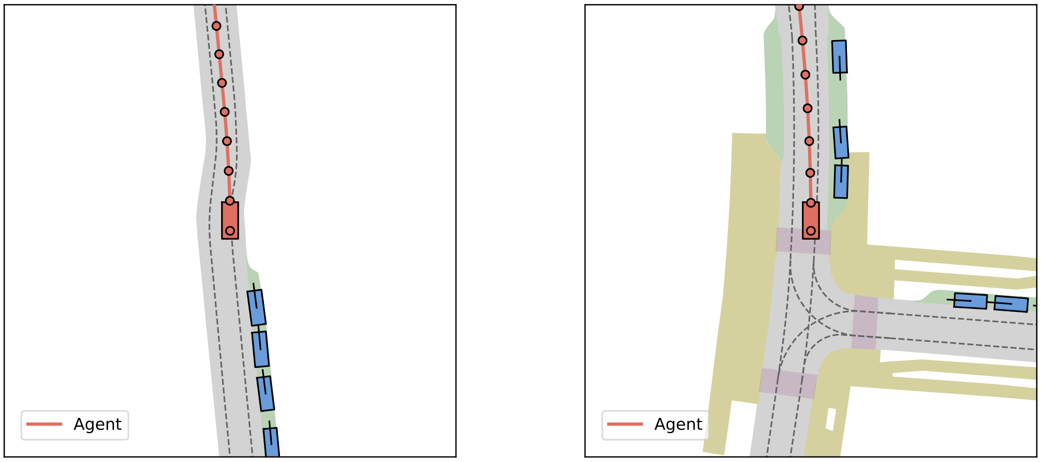

Figure 5: **BEV of Lane Keeping (LK) Failure Cases.**

3.3 Human Flag

To mitigate unwarranted penalties caused by annotation noise or contextually valid maneuvers, we introduce a human-flag filter: if the human expert's trajectory for a given scene exhibits the same rule violation as the model's prediction, the corresponding penalty is omitted. This mechanism preserves

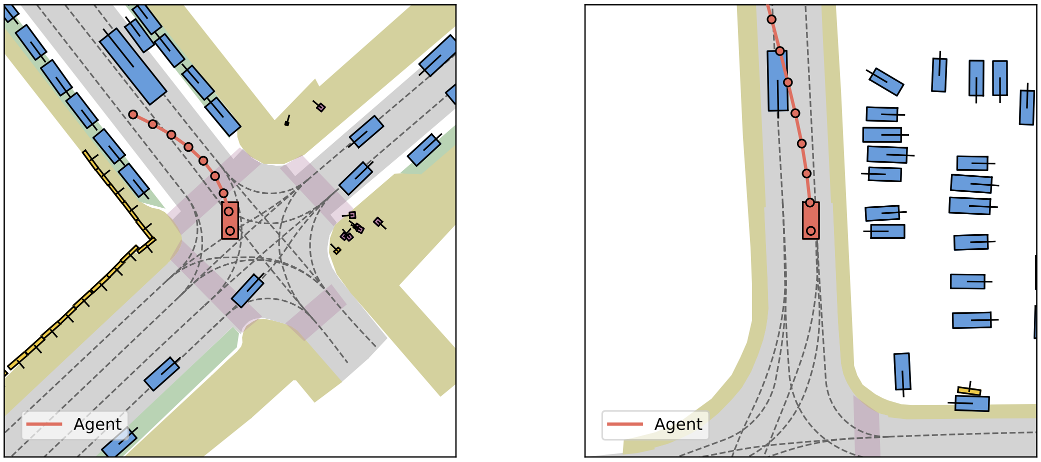

Figure 6: **BEV of Driving Direction Compliance (DDC) Failure Cases.**

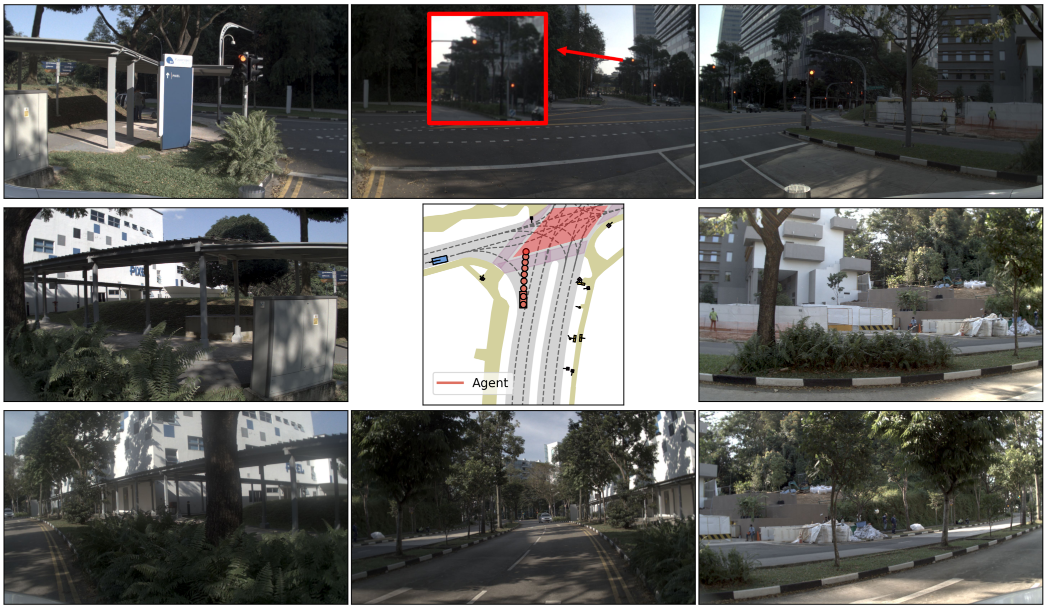

Figure 7: **Traffic Light Compliance (TLC) Failure Case.**

enforcement of safety constraints while exempting legitimate behaviors. The following case studies demonstrate the importance of the human-flag filter in maintaining evaluation integrity.

Case 1: Label Noise. In some scenes, traffic light states are incorrectly annotated due to occlusion. As shown in Fig. 10, a right turn signal is blocked by a truck, resulting in a mislabeled red light in the dataset. Manual inspection of the surrounding video frames confirms that the light was in fact green. Consequently, the human expert's trajectory (shown in green) appears to violate the Traffic Light Compliance (TLC) criterion, an artifact of label noise rather than true noncompliance. By applying the human flag filter, we omit the penalty for the ego vehicle in this case, since the expert demonstrates the same apparent violation. This example highlights how the filter uses expert behavior to suppress penalties introduced by annotation errors.

Case 2: Legitimate Maneuver. In certain edge cases, human drivers carry out contextually valid maneuvers that still trigger violations under our defined metrics. For example, the Time to Collision (TTC) metric computes the minimum predicted time to contact with any obstacle, enforcing a fixed

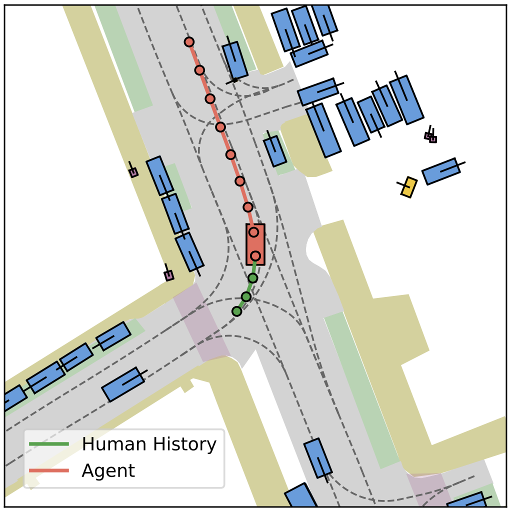

Figure 8: **BEV of History Comfort (HC) Failure Case.**

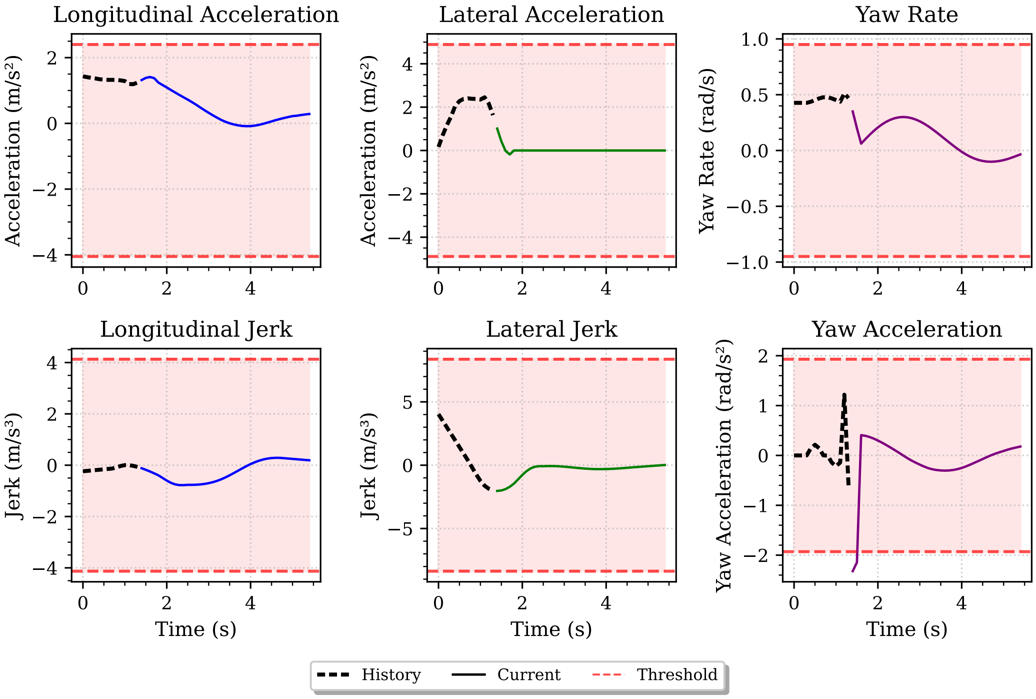

Figure 9: **Statistical Comparison of History Comfort (HC) Failure Case.**

safety threshold at every simulation step. In the scenario shown in Fig. 11, the ego vehicle is navigating a narrow turn with a stationary black car near its intended path. Although the maneuver is safe and commonly executed by human drivers, the predicted trajectory briefly passes close to the obstacle, causing the TTC to drop below the threshold. This results in a TTC score of zero, despite the absence of actual risk. Penalizing the ego agent in this setting would therefore be misleading. By applying the human flag filter, we exempt such behavior from penalty, using expert trajectories to distinguish unsafe motion from valid human driving.

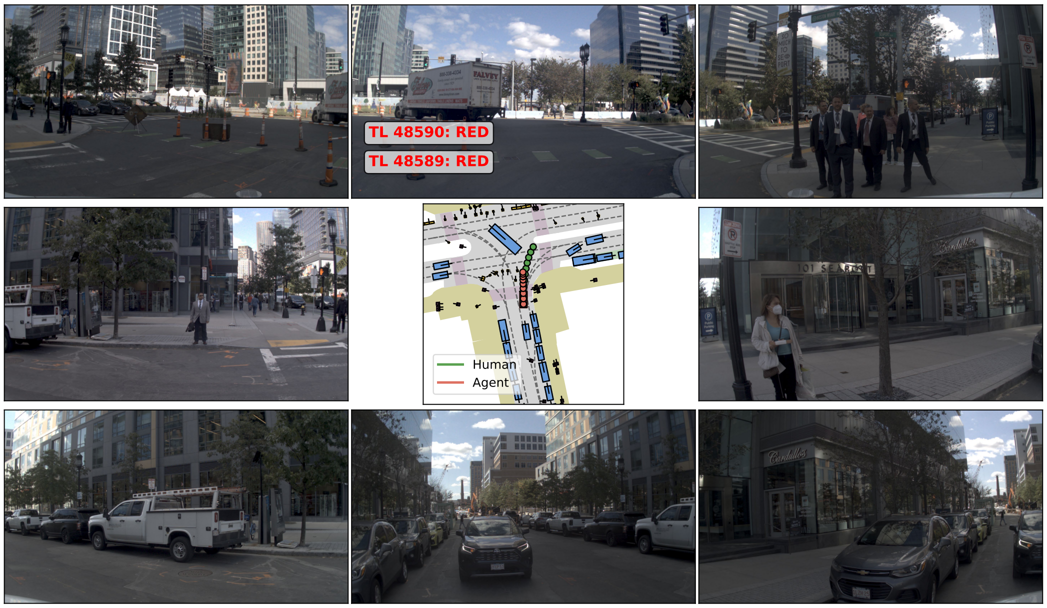

Figure 10: **Case 1: Human Flag in Traffic Light Compliance (TLC).**

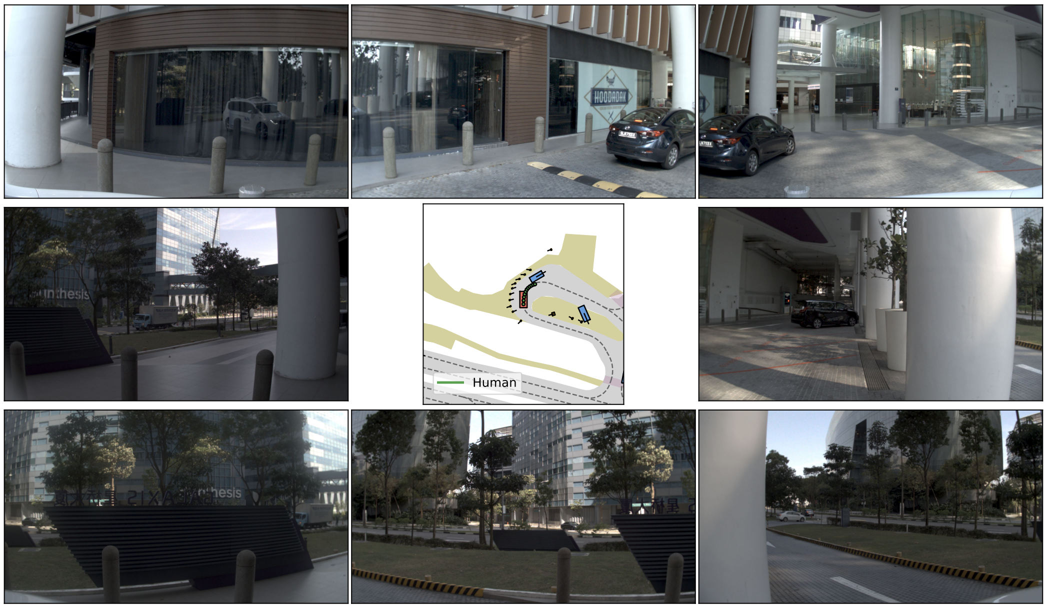

Figure 11: **Case 2: Human Flag in Time to Collision (TTC).**

3.4 Ablation Studies on Correlation Analysis

We attempt different aggregation approaches to weight the importance of each Stage 2 score across different synthetic viewpoints.

k-NN Aggregation. The pure kNN approach first calculates Euclidean distances, $d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$, between Stage 1 endpoints and Stage 2 start points. The algorithm selects `k` nearest neighbors before applying exponential decay weighting, $w_i = e^{-d_i}$. Weights are normalized to ensure $\sum_{i=1}^{k} w_i = 1$.

Fig. 12 summarizes the correlation performance across different values of k, evaluated over a diverse set of rule-based and learning-based planners. Our method achieves an R^2 of 0.80, while the best-performing kNN configuration (k = 5) yields an R^2 of 0.75. We observe that both rank and linear correlation coefficients improve consistently with increasing k, with results reported for $k = 1$, $k = 3$, and $k = 5$. This suggests that larger neighborhoods help capture more relevant information, though the marginal gains diminish.

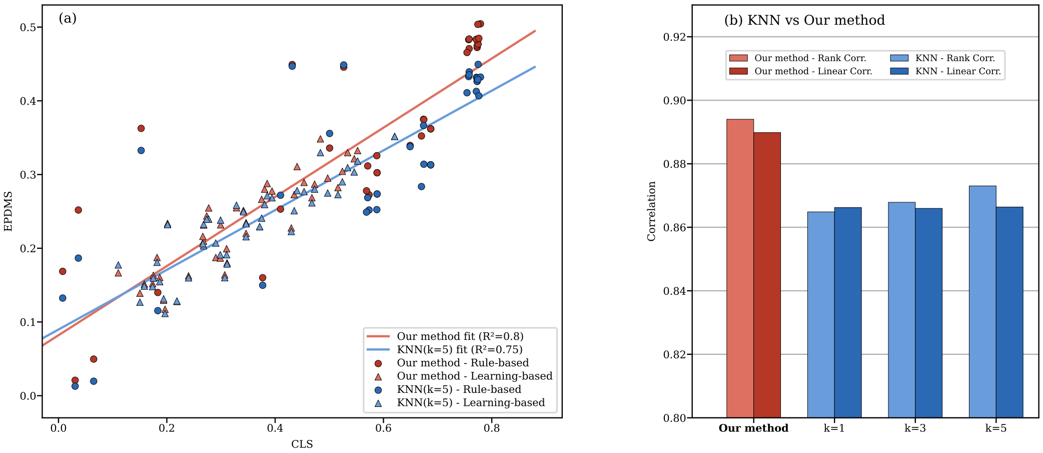

Figure 12: **kNN correlation analysis.**

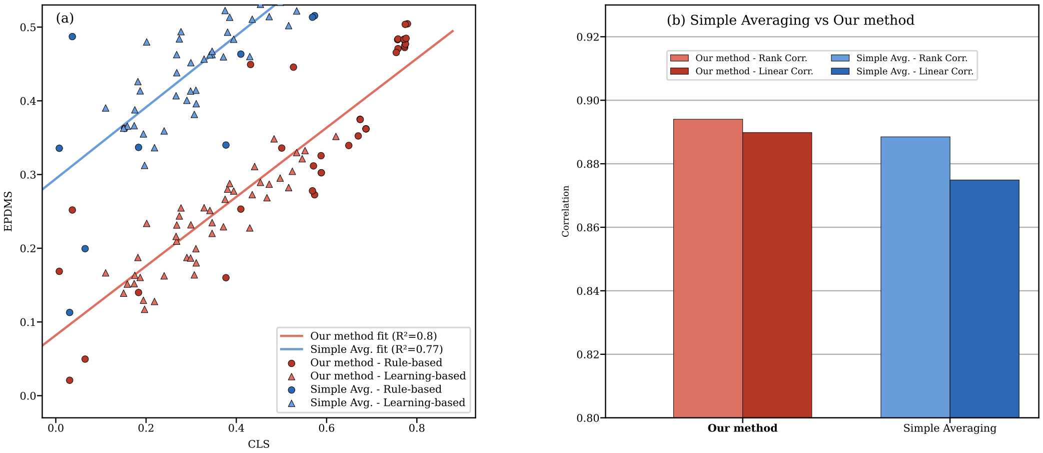

Figure 13: **Simple Averaging correlation analysis.**

Simple Averaging. The simple averaging method computes weights as the arithmetic mean across all Stage 2 scenarios, without distance-based weighting. This approach involves direct calculation of average scores, without any spatial filtering or neighbor selection.

Fig. 13 shows the correlation results using this method. The R^2 score reaches approximately 0.77, slightly below the 0.80 achieved by our current aggregation method. Both rank and linear correlation coefficients are also consistently lower compared to our approach, indicating that distance-aware aggregation contributes meaningfully to improving alignment between pseudo-simulation and full closed-loop performance.

Hybrid k-NN/Gaussian. The hybrid approach combines neighborhood selection with Gaussian kernel weighting. First, it calculates squared distances, $d^2 = (x_1 - x_2)^2 + (y_1 - y_2)^2$, and selects k nearest neighbors. It then applies Gaussian weighting, $w_i = e^{-d_i^2/(2\sigma^2)}$, to the filtered points, followed by normalization.

Fig. 14 shows the correlation results for this method. We evaluate several combinations of k and σ^2, and find that the setting $k = 3$ and $\sigma^2 = 0.1$ yields the highest correlation, achieving an R^2 of approximately 0.76. While this outperforms simple averaging and offers computational efficiency through neighborhood reduction, it still underperforms compared to our current method, which achieves an R^2 of 0.80.

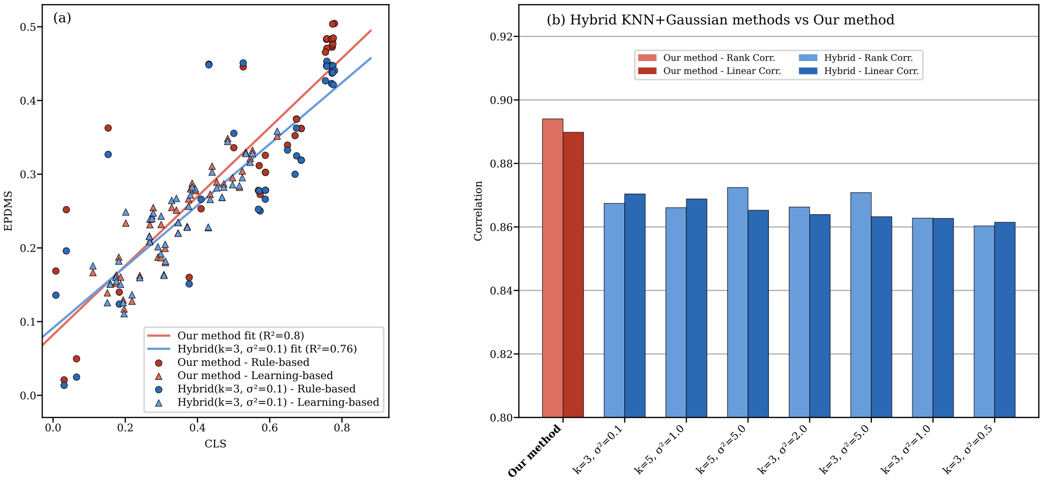

Figure 14: **Hybrid kNN/Gaussian correlation analysis.**

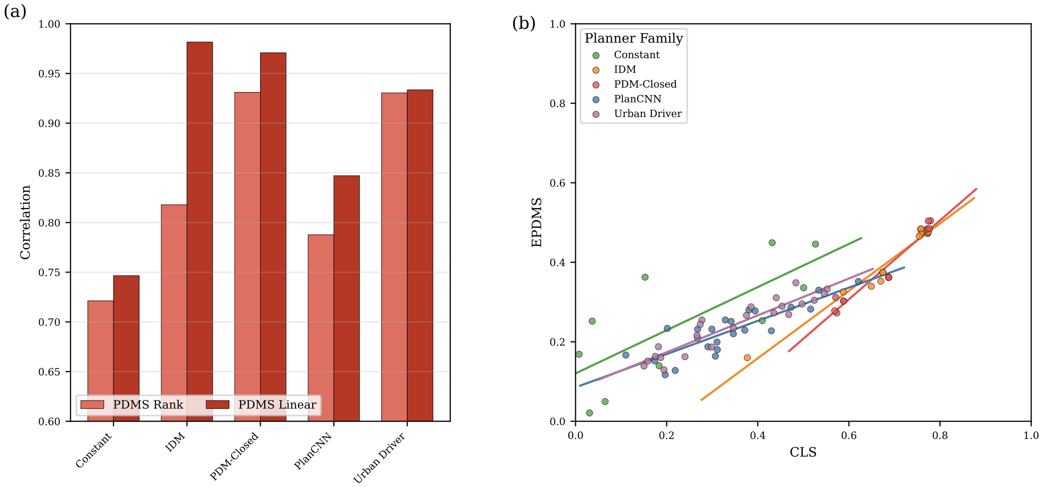

Figure 15: **Fine-Grained Correlations.**

Fine-Grained Correlations. We also examine the correlation performance for each individual planner, covering a diverse set of rule-based and learning-based methods. Specifically, we consider Constant Kinematics, IDM [6], PDM-Closed [7], PlanCNN [8], and Urban Driver [9], each tested under multiple model configurations.

Fig. 15 presents these results. On the left, we report both rank and linear correlation coefficients for each planner. All planners achieve strong positive correlations over 0.7. Fig. 15 (b) visualizes the predicted EPDMS score against the closed-loop score. PDM-Closed and IDM exhibit the strongest alignment, with scatter points tightly concentrated along the diagonal. These results confirm that our metric aligns closely with full closed-loop performance.

References

[1] 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.

[2] 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.

[3] 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.

[4] O. Contributors. Openscene: The largest up-to-date 3d occupancy prediction benchmark in autonomous driving. https://github.com/OpenDriveLab/OpenScene, 2023.

[5] 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.

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

[7] 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.

[8] 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.

[9] 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.