

Learning to Drive by Imitating Surrounding Vehicles

Yasin Sonmez, Hanna Krasowski and Murat Arcaç

Abstract—Imitation learning is a promising approach for training autonomous vehicles (AV) to navigate complex traffic environments by mimicking expert driver behaviors. While existing imitation learning frameworks focus on leveraging expert demonstrations, they often overlook the potential of additional complex driving data from surrounding traffic participants. In this paper, we study a data augmentation strategy that leverages the observed trajectories of nearby vehicles, captured by the AV’s sensors, as additional demonstrations. We introduce a simple vehicle-selection sampling and filtering strategy that prioritizes informative and diverse driving behaviors, contributing to a richer dataset for training. We evaluate this idea with a representative learning-based planner on a large real-world dataset and demonstrate improved performance in complex driving scenarios. Specifically, the approach reduces collision rates and improves safety metrics compared to the baseline. Notably, even when using only 10 percent of the original dataset, the method matches or exceeds the performance of the full dataset. Through ablations, we analyze selection criteria and show that naive random selection can degrade performance. Our findings highlight the value of leveraging diverse real-world trajectory data in imitation learning and provide insights into data augmentation strategies for autonomous driving.

I. INTRODUCTION

By learning from expert demonstrations, imitation learning enables autonomous vehicles (AVs) to develop policies that mimic human-like driving behavior. Recently, imitation learning models [1], [2] have started to outperform traditional rule-based methods [3] on benchmarks with large-scale real-world data such as nuPlan [4], indicating the increasing viability of imitation learning for real-world deployment. However, imitation learning also suffers from three major challenges. First, recent studies demonstrate that imitation learning models can learn shortcuts from data [5], leading to undesired behaviors. For example, it has been demonstrated that models with historical AV motion data excel in open-loop evaluation but underperform in closed-loop metrics, likely due to learning shortcuts [6]. Second, imitation learning suffers from the distribution shift problem, where the training and test sets have different distributions due to the nature of the application, such as learning from data collected in one location and deploying the model elsewhere. To address this challenge, several studies suggest that imitation learning benefits from reinforcement learning refinements [7]. Lastly, imitation learning suffers from causal confusion [8] when a model learns spurious correlations instead of true causal relationships between actions and outcomes. Since

imitation learning relies on mimicking expert demonstrations, the model may pick up on irrelevant features or unintended cues that correlate with successful behavior but do not actually cause it.

Addressing these challenges through effective data augmentation, model architecture, and loss choices is essential for improving real-world performance. As such, we have to maximize the utility of the available data. Despite the availability of large datasets, simulators, and benchmarks (e.g., [4], [9], [10]), effectively utilizing this data for imitation learning remains a challenge. Different datasets capture driving information at varying levels of abstraction, ranging from object-level annotations to raw sensor images. Furthermore, recent studies have indicated that simply increasing the volume of training data does not necessarily result in improved model performance. For example, [11] highlights that more data may not always translate to better outcomes, suggesting that other factors, such as data quality and relevance, play a more significant role in model effectiveness.

In this paper, we study a data augmentation technique that enhances imitation learning for autonomous driving by using trajectories beyond those of the AV in driving datasets. Building on the idea of learning from other vehicles in a scene, we analyze this strategy in the object-based planning setting. We introduce simple selection criteria that prioritize informative and diverse driving trajectories from observed vehicles and validate the approach through extensive ablation studies. The main contributions of this work are:

- We demonstrate the effectiveness of learning from surrounding vehicles on a large real-world object-based planning dataset (nuPlan), unlike prior studies that are limited to simulation environments. Our approach leverages real human driving behaviors from surrounding vehicles rather than simulated trajectories, providing more realistic and diverse training signals.
- We conduct extensive ablations on vehicle-selection criteria and analyze how different selection strategies affect learning performance, addressing a key limitation of prior research that assumes all vehicles are useful for learning.

II. RELATED WORK

Datasets for autonomous driving: Although traffic data is tedious and costly to collect, there is an increasing amount of open-source datasets for autonomous driving research. These datasets can be broadly categorized into perception (i.e., sensor-based [12], [13], [14]) and motion planning (i.e., object-based [15], [16], [4], [17]) datasets. Since imitation learning is commonly used for motion planning tasks, we

Project website <https://yasinsonmez.github.io/papers/learning-to-drive/>

¹All authors are with University of California, Berkeley, CA, USA {yasin-sonmez, krasowski, arcak}@berkeley.edu

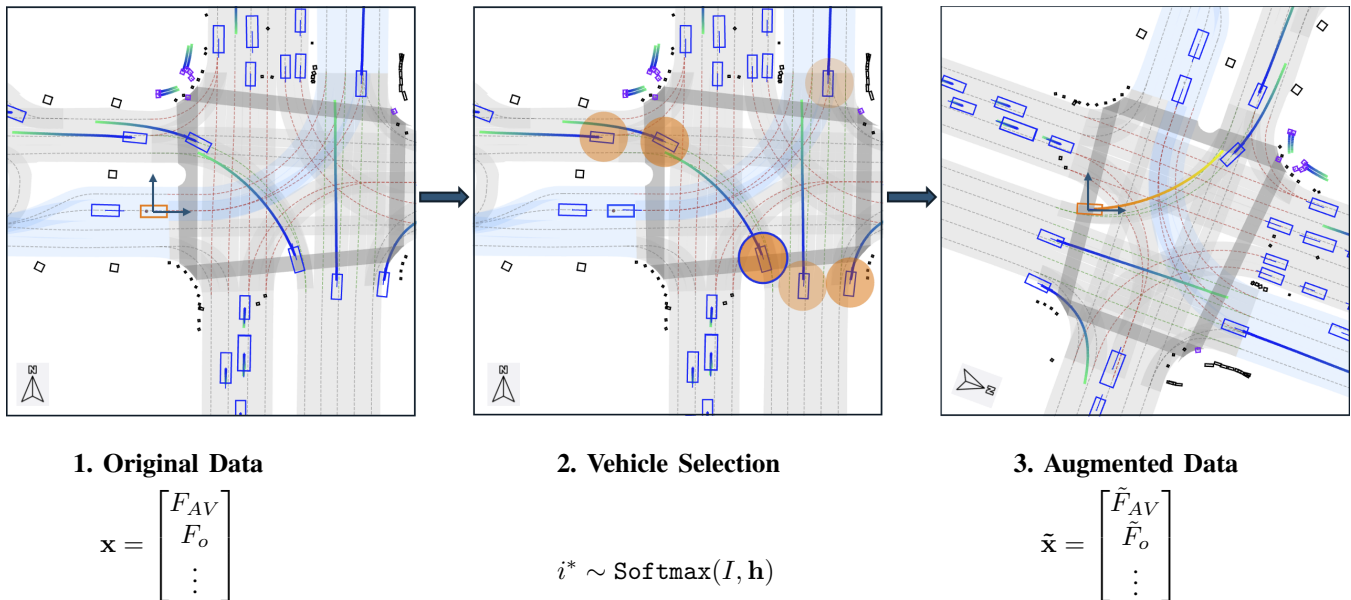


Fig. 1. Data augmentation framework illustrated for a traffic scenario: (1) The expert driver remains stationary at a red light while surrounding vehicles follow diverse trajectories. (2) A suitable surrounding vehicle i^* (shown with a blue circle) is sampled from the weighted categorical distribution defined in Equation 5, where darker circles represent higher selection probabilities and \mathbf{h} is the vector of sum of absolute heading deviations for all vehicles I . (3) A reference frame transformation generates features from the perspective of the new ego vehicle.

focus on object-based datasets, which have the additional advantage of allowing for interpretable and complete data augmentation. The object-based dataset NGSIM [18] is one of the earlier large datasets initially published in 2006. Since then many new datasets with increasing size and traffic complexity have been published. In particular, the nuPlan benchmark [19], [4] consists of real-world autonomous driving datasets and evaluation frameworks. nuPlan offers a comprehensive dataset for both prediction and planning, with 1282 hours of driving data from four cities, and introduces a taxonomy of driving scenarios. Due to these features, nuPlan has been used to compare various planning approaches in the literature such as [20], [3], [21].

Data augmentation for autonomous driving: Although above datasets are of increasing size, data augmentation can significantly enhance their value. For example, [22] develops context-aware data augmentation for imitation learning that is based on a variational autoencoder. PLUTO [1] employs contrastive imitation learning to address distribution shift by applying both positive and negative data augmentations, where positive augmentations agree with the ground truth and negative augmentations intentionally disagree.

Recent studies have also started exploring leveraging trajectories of surrounding vehicles for data augmentation. Chen et al. [23] propose learning from all vehicles observed by the ego-vehicle in a simulator, using simulated agents with simple trajectories to increase sample efficiency. Zhang et al. [24] present a "Learning by Watching" approach that converts observations into Bird's-Eye-View representations and infers actions from observed vehicles. However, these approaches suffer from several limitations: (1) they are restricted to simulation environments where surrounding

vehicles exhibit simplified, rule-based behaviors that may not capture real human driving complexity; (2) they lack intelligent vehicle selection mechanisms, using all observed vehicles without considering that many surrounding vehicles may exhibit uninformative or harmful behaviors; (3) they lack evaluation on large-scale real-world datasets, making it difficult to assess practical effectiveness.

Imitation learning for autonomous driving: The two main learning approaches for autonomous driving are reinforcement learning and imitation learning. Reinforcement learning usually relies on a realistic simulation environment and significant reward-shaping to achieve performant driving policies [25]. Imitation learning is usually easier to tune but requires a diverse and large dataset of driving trajectories to achieve expert-like driving behavior [26], [27]. Leading companies in autonomous driving, such as Tesla and Waymo [28], [7], as well as open-source projects like OpenPilot [29], leverage imitation learning to train models by mimicking expert driving behavior. A recent example is the work by [2], who propose a transformer-based Diffusion Planner for closed-loop planning, capable of modeling multi-modal driving behavior. Another notable framework in imitation learning is PLUTO [1], which introduces key innovations for more efficient driving behavior generation: a longitudinal-lateral aware transformer architecture, contrastive learning to mitigate causal confusion and distribution shift, and ego-related data augmentation. In this study, we use PLUTO as a baseline and perform ablation studies on the nuPlan dataset.

III. METHODOLOGY

A. Problem Formulation

Our method is a general data augmentation approach applicable to object-based planning frameworks. Below, we

briefly formulate a generic object-based planning problem that the planner used in our experiments also follows.

We consider an AV, N_A dynamic agents, N_O static obstacles, a high-definition map M , and traffic context information C (e.g., traffic light status). Each agent i at time t has a state $\mathbf{s}_i^t = (\mathbf{p}_i^t, \theta_i^t, \mathbf{v}_i^t, \mathbf{b}_i^t, I_i^t)$, where $\mathbf{p}_i^t \in \mathbb{R}^2$ and $\theta_i^t \in \mathbb{R}$ denote position and heading, $\mathbf{v}_i^t \in \mathbb{R}^2$ represents velocity, and $\mathbf{b}_i^t \in \mathbb{R}^2$ and $I_i^t \in \{0, 1\}$ correspond to the bounding box dimensions and observation status, respectively. The feature set for dynamic agents is denoted as $\mathcal{A} = A_{0:N_A}$, where A_0 represents the AV, and the static obstacle set is $\mathcal{O} = O_{1:N_O}$. The future state of agent a at time t is denoted as \mathbf{y}_a^t , with historical and future horizons of T_H and T_F , respectively. The planner generates N_T multi-modal planning trajectories for the AV along with predictions for each dynamic agent. The final trajectory τ^* is selected via a scoring module S , which integrates these outputs with the scene context. The overall formulation is given as:

$$(\mathbf{T}_0, \pi_0), \mathbf{P}_{1:N_A} = f(\mathcal{A}, \mathcal{O}, M, C \mid \phi) \quad (1)$$

$$(\tau^*, \pi^*) = \arg \max_{(\tau, \pi) \in (\mathbf{T}_0, \pi_0)} S(\tau, \pi, \mathbf{P}_{1:N_A}, \mathcal{O}, M, C), \quad (2)$$

where f represents the planning model, ϕ are the model parameters, $(\mathbf{T}_0, \pi_0) = \{(\mathbf{y}_{0,i}^{1:T_F}, \pi_i) \mid i = 1, \dots, N_T\}$ are the generated planning trajectories with confidence scores, and $\mathbf{P}_{1:N_A} = \{\mathbf{y}_a^{1:T_F} \mid a = 1, \dots, N_A\}$ are the predicted future states of traffic participants.

B. Learning From Surrounding Traffic

Many real-world driving scenarios are inherently dynamic, with multiple interacting agents. While some situations may involve routine behaviors such as lane-keeping or waiting at a red light, even in these types of scenarios usually there is at least one vehicle exhibiting complex or “interesting” behaviors in the surrounding traffic. Examples include lane changes, turning at intersections, yielding to pedestrians, or reacting to bicycles. These nuanced interactions provide a rich source of data for understanding diverse driving behaviors and the decision-making processes of road users. Learning from surrounding traffic method capitalizes on this by augmenting the imitation learning dataset with estimated trajectories of selected agents from the surrounding traffic.

The main advantage is increased data diversity along two dimensions. First, incorporating trajectories from surrounding vehicles expands the support of the training distribution by adding non-routine maneuvers (turns, lane changes, yielding), varied driver styles and vehicle dynamics, and exposure to multi-agent interactions. This reduces overfitting, improves generalization [30], [31], and improves anticipation in interaction-heavy scenes [32]. Second, prioritizing dynamic and contextually rich cases rather than routine AV-only behavior has been shown to improve performance [11].

The distribution of the sum of absolute heading angle deviations in nuPlan dataset, illustrated in Figure 2, further supports the need for increased data diversity. The blue histogram indicates that the majority of observed vehicle data consists of minimal deviation movements, such as

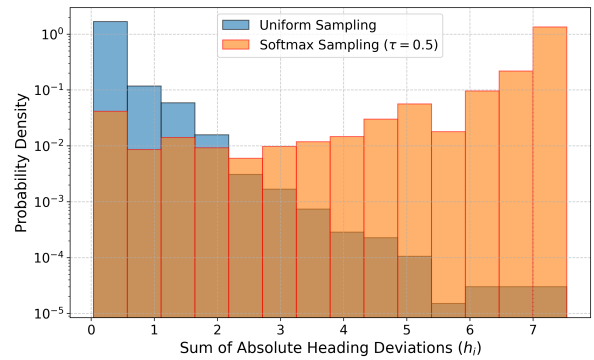


Fig. 2. Histogram of the sum of absolute heading deviations h_i for observed vehicles in the dataset. The blue plot represents the original data, while the orange plot corresponds to vehicles sampled using the softmax distribution defined in Equation 5. The histogram is presented on a logarithmic scale to account for the large differences in probabilities.

lane-keeping, simple acceleration, and deceleration. This demonstrates that routine behaviors dominate the dataset, highlighting the importance of our approach to incorporate diverse motion patterns from dynamic surrounding vehicles.

C. Vehicle Selection Criteria

Most simulation scenarios have many vehicles present, but it is infeasible to augment the dataset with all of them. Many of these vehicles may exhibit little to no interesting behavior, and including them would unnecessarily increase the computation and contribute little to the learning process. Prior research has shown that treating all training data equally can lead to suboptimal performance, especially in safety-critical situations, and that prioritizing more informative samples can improve robustness while reducing data requirements. For instance, [11] demonstrated that an imitation-learning-based planner trained on only 10% of a dataset, carefully curated using a trained scenario difficulty predictor, performed as well as one trained on the full dataset while significantly reducing collisions and improving route adherence. Therefore, vehicle selection criteria are essential to ensure the dataset contains the most beneficial scenarios.

Let s index a scene with agent set $\mathcal{I}_s = \{0, 1, \dots, N_A\}$ (with 0 for the AV), time indices $t \in \{0, 1, \dots, T_H - 1\}$, map M , and context C . We first define the per-scene pool of eligible non-AV agents via filtering

$$\hat{\mathcal{V}}_s = \left\{ i \in \mathcal{I}_s \setminus \{0\} : \begin{array}{ll} I_i^t = 1 & \forall t, \\ \|\mathbf{p}_i^t - \mathbf{p}_0^t\|_2 \leq r & \forall t, \\ \mathbf{p}_i^t \in \text{Drivable}(M) & \forall t \end{array} \right\}, \quad (3)$$

where $I_i^t \in \{0, 1\}$ indicates observability of the agent in that timestep, $r = 50$ m, and $\text{Drivable}(M)$ denotes the drivable set induced by the map. This filtering ensures data quality and consistency by: (i) requiring full observability across all timesteps to avoid partial sensing artifacts; (ii) constraining proximity to the AV for reliable measurements; and (iii) ensuring drivability to focus on meaningful driving interactions rather than parked vehicles.

a) *Weighted sampling based on heading deviations:*

Our vehicle selection strategy is based on the observation that vehicles performing complex maneuvers such as turns, lane changes, and parking operations exhibit significant changes in their heading direction, while routine behaviors like lane-keeping and straight-line driving result in minimal heading variations. As shown in Figure 2, the distribution of heading deviations reveals that most vehicles in the dataset exhibit low deviation values, corresponding to routine driving behaviors that dominate the training data. By prioritizing vehicles with higher heading deviations, we effectively balance the dataset toward more informative and diverse driving patterns that are crucial for learning robust driving policies. While more sophisticated selection criteria based on jerk, acceleration patterns, or driver expertise could be explored, we deliberately use heading deviation as a simple, interpretable, and reasonably effective proxy for dynamic behavior that captures the characteristics of complex driving interactions.

After filtering according to Equation 3, for the remaining pool of vehicles, we assign a weight h_i based on their sum of absolute heading angle deviation over time:

$$h_i = \sum_t |\theta_i^t - \theta_i^{t-1}|, \quad i \in \hat{\mathcal{V}}_s. \quad (4)$$

We then define a softmax probability distribution over eligible agents $\hat{\mathcal{V}}_s$ to sample more informative vehicles:

$$p_\tau(i | s) = \frac{\exp(h_i/\tau)}{\sum_{j \in \hat{\mathcal{V}}_s} \exp(h_j/\tau)}, \quad (5)$$

where h_i represents the weight for vehicle i , $p_\tau(i | s) = \text{Softmax}(i | s)$ is its assigned probability, and τ is the temperature parameter that controls the sharpness of the weight distribution. A lower τ makes the selection more focused on vehicles with higher deviations, while a higher τ results in a more uniform weighting across all vehicles.

We explore two different sampling approaches for vehicle selection. The first approach, *per-scene sampling*, selects N_s agents from each scene independently using the softmax distribution in Equation 5:

$$\mathcal{I}_s = \{i_1, \dots, i_{N_s}\} \sim \text{SWR}(p_\tau(\cdot | s), N_s), \quad (6)$$

where SWR denotes sampling without replacement. This strategy ensures diversity across different traffic scenarios by forcing selection from each scene. The second approach, *per-ego sampling*, selects one agent per ego-vehicle scenario across the entire dataset; for N_e ego scenarios in the dataset, yielding N_e selected agents, rather than sampling scene by scene. While per-ego sampling may select globally highest-scoring agents, it risks over-representing certain scenarios and reducing overall diversity.

b) *Driving agent filters:* Many eligible agents provide little learning signal (e.g., near-stationary) or reflect undesired behavior. To bias sampling toward informative and reliable candidates, we compute three per-agent quantities (displacement, comfort violations, and TTC violations) over $t \in \{0, \dots, T_H - 1\}$ from the states $\{\mathbf{p}_i^t, \theta_i^t, \mathbf{v}_i^t\}$ (Section III) and they are used to prune the candidates before sampling.

Displacement encourages agents that actually move and interact with the scene:

$$d_i = \|\mathbf{p}_i^{T_H-1} - \mathbf{p}_i^0\|_2. \quad (7)$$

Comfort violations filters highly jerky or unstable motion that is often noisy or undesirable to imitate. Using \mathbf{v}_i^t in i 's heading-aligned frame, obtain accelerations $a_{x,i}(t), a_{y,i}(t)$, jerks $\dot{a}_{x,i}(t), \dot{a}_{y,i}(t)$, and yaw rate/acceleration $\dot{\theta}_i(t), \ddot{\theta}_i(t)$. We define the comfort violation count as

$$v_i^{\text{comf}}(t) = \mathbb{1}\{|a_{x,i}(t)| > \alpha_x \wedge |a_{y,i}(t)| > \alpha_y \quad (8)$$

$$\wedge |\dot{a}_{x,i}(t)| > \beta_x \wedge |\dot{a}_{y,i}(t)| > \beta_y \quad (9)$$

$$\wedge |\dot{\theta}_i(t)| > \gamma_1 \wedge |\ddot{\theta}_i(t)| > \gamma_2\}, \quad (10)$$

$$V_i^{\text{comf}} = \sum_{t=0}^{T_H-1} v_i^{\text{comf}}(t), \quad (11)$$

TTC violations detect agents frequently entering imminent-collision geometry. We use an in-lane, longitudinal approximation along agent i 's heading. Let g_0 denote the front-to-rear gap between agent i 's front bumper and agent j 's rear bumper. Let u_i and u_j be the agents' longitudinal speeds along i 's heading, and define the closing speed $\Delta u = u_i - u_j$. The in-lane TTC and its violation count are

$$\text{ttc}_{i,j}(t) = \begin{cases} \frac{g_0}{\max(\epsilon, \Delta u)} & g_0 > 0, \Delta u > 0, \\ +\infty & \text{otherwise,} \end{cases} \quad (12)$$

$$V_i^{\text{ttc}} = \sum_{t=0}^{T_H-1} \mathbb{1}\{\min_{j \neq i} \text{ttc}_{i,j}(t) < \theta_{\text{TTC}}\}. \quad (13)$$

Let $\mathcal{F} \subseteq \{\text{disp}, \text{comf}, \text{ttc}\}$ denote the set of active filters. The filtered pool used for sampling is

$$\tilde{\mathcal{V}}_s(\mathcal{F}) = \left\{ i \in \hat{\mathcal{V}}_s : \begin{aligned} & (\text{disp} \in \mathcal{F} \Rightarrow d_i \geq d_{\min}) \\ & \wedge (\text{comf} \in \mathcal{F} \Rightarrow V_i^{\text{comf}} \leq \kappa_{\text{comf}}) \\ & \wedge (\text{ttc} \in \mathcal{F} \Rightarrow V_i^{\text{ttc}} \leq \kappa_{\text{ttc}}) \end{aligned} \right\}. \quad (14)$$

Thresholds ($d_{\min}, \kappa_{\text{comf}}, \kappa_{\text{ttc}}$) and the choice of \mathcal{F} are specified in Section IV-B. By default, we sample from $\hat{\mathcal{V}}_s$; in filtering ablations we replace it with $\tilde{\mathcal{V}}_s(\mathcal{F})$.

To motivate our filtering design, we compare violation rates for the AV versus other traffic participants. Figure 3 shows that ego trajectories exhibit fewer TTC and comfort violations than surrounding vehicles, either because the ego is a trained expert driver or because non-ego trajectories are noisier due to partial observability and estimation artifacts. This motivates applying simple, robust criteria to prioritize informative and reliable non-ego trajectories.

D. Dataset Augmentation

For each selected agent $i \in \mathcal{I}_s$, let $T_i \in \text{SE}(2)$ denote the rigid transform matrix mapping world coordinates to agent- i 's egocentric frame. We apply this transform to all agent spatial components such as position, velocity, and orientation. Denote the original input feature vector by \mathbf{x}_s ; the augmented feature vector is

$$\tilde{\mathbf{x}}_{s,i} = f_{T_i}(\mathbf{x}_s), \quad (15)$$

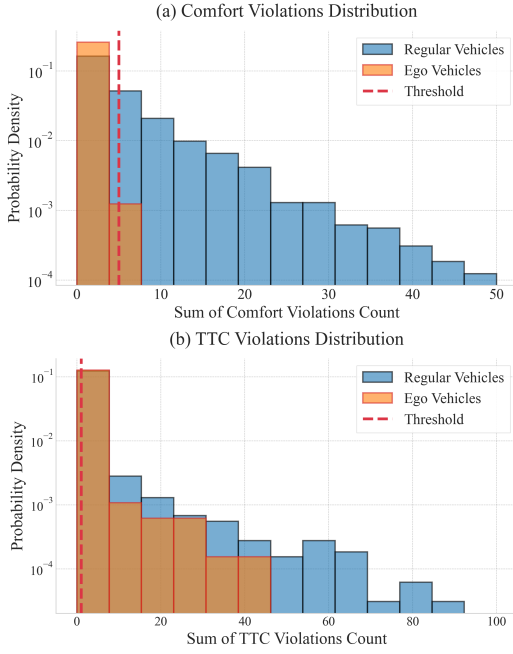


Fig. 3. Comparison of TTC and Comfort violation counts for the ego vehicle versus surrounding vehicles across scenes. Ego trajectories tend to have fewer violations, while non-ego trajectories show higher rates due to suboptimal behaviors and greater measurement noise.

where f_{T_i} applies the rigid transform T_i to the spatial components of \mathbf{x}_s . Over the dataset $\mathcal{D} = \{\mathbf{x}_s\}_{s=1}^N$, the augmented dataset is

$$\mathcal{D}_{\text{aug}} = \mathcal{D} \cup \bigcup_{s=1}^N \{\tilde{\mathbf{x}}_{s,i} : i \in \mathcal{I}_s\}. \quad (16)$$

IV. EXPERIMENTS

We choose the PLUTO planner [1] as the baseline, as it is a high-performance learning-based planner that uses object-based datasets. We selected PLUTO due to its recency, strong closed-loop performance on nuPlan, and compatibility with object-based data. While our method can be applied to other similar object-based planning methods such as Diffusion Planner [2], we focused on PLUTO due to computational constraints. Each training and evaluation cycle requires approximately 6 hours for pre-processing, 48 hours for model training on 4 A5000 GPUs, and 6 hours for simulation rollouts. Nevertheless, our data augmentation technique operates independently of PLUTO’s internal components (e.g., confidence scores, trajectory scoring, goal specification, model architecture details). For completeness, the PLUTO planner operates based on:

Agent History Encoding: PLUTO uses the agent state \mathbf{s}_i^t defined in Section III. The trajectories of agents are captured by computing differences between consecutive timesteps, resulting in a feature matrix $\mathbf{F}_A \in \mathbb{R}^{N_A \times (T_H - 1) \times 8}$.

Static Obstacles Encoding: Static obstacles in the drivable area are encoded as $\mathbf{o}_i = (\mathbf{p}_i, \theta_i, \mathbf{b}_i)$, producing a feature matrix $\mathbf{F}_O \in \mathbb{R}^{N_O \times 5}$.

AV’s State Encoding: To mitigate that, imitation learning frequently learns performance-degrading shortcuts [6], [33], only the current state of the AV is used as input features without using the history. These include the AV’s position, heading angle, velocity, acceleration, and steering angle, represented as $\mathbf{F}_{AV} \in \mathbb{R}^{1 \times 8}$.

Vectorized Map Encoding: The map consists of N_p polylines, each undergoing an initial subsampling step to standardize the number of points. Feature vectors are then computed for each polyline point. Specifically, for the i -th point of a polyline, the feature vector consists of $(\mathbf{p}_i - \mathbf{p}_0, \mathbf{p}_i - \mathbf{p}_{i-1}, \mathbf{p}_i - \mathbf{p}_i^{\text{left}}, \mathbf{p}_i - \mathbf{p}_i^{\text{right}})$ where \mathbf{p}_0 is the initial point of the polyline, and $\mathbf{p}_i^{\text{left}}$ and $\mathbf{p}_i^{\text{right}}$ represent the left and right lane boundary points, respectively. The final representation of the polyline features is $\mathbf{F}_P \in \mathbb{R}^{N_P \times n_p \times 8}$, where n_p is the number of points per polyline.

Scene Encoding: To capture interactions between dynamic agents, static obstacles, polylines, and the autonomous vehicle, all described encodings are concatenated and processed through transformer encoders, with Fourier-based positional embeddings and learnable semantic attributes compensating for the loss of global positional information.

Trajectory Planning and Post-processing: The model generates multimodal trajectories with confidence scores and employs a rule-based post-processing module to ensure safe and robust selection. Forward simulation, utilizing a linear quadratic regulator for trajectory tracking and a kinematic bicycle model for state updates, assesses rollouts based on metrics such as driving comfort, and time-to-collision. The final trajectory $\tau^* = \arg \max_{(\tau, \pi) \in (\mathcal{T}_0, \pi_0)} S(\tau, \pi, \mathbf{P}_{1:N_A}, \mathcal{O}, M, C)$ combines learning-based confidence with rule-based evaluations.

A. Training and Evaluation

We train the baseline PLUTO planner using varying numbers of scenarios extracted from the nuPlan dataset. For each scenario, we generated N_s additional scenarios from surrounding vehicles. However, in some cases, no suitable vehicles were consistently observable across all time steps, and augmentation was not applied then. All models were trained to convergence, monitored using validation error. We used the hyperparameters of the PLUTO implementation.

For evaluation, we use the test14-hard benchmark, which was curated by executing 100 scenarios for each of 14 scenario types and selecting the 20 lowest-performing instances per type using the rule-based planner PDM-Closed [3]. We do not repeat experiments on val14, a uniformly sampled nuPlan scenarios dataset, due to the strong correlation between learning-based methods’ performance on test14-hard and val14, and substantially higher computational costs.

The nuPlan framework provides a comprehensive evaluation score for each simulation, incorporating key metrics such as (1) **No Ego At-Fault Collisions**, where only AV-initiated collisions are considered; (2) **TTC (Time-to-Collision) Compliance**, ensuring time-to-collision remains above a threshold; (3) **Drivable Area Compliance**, requiring the AV to stay within road boundaries; (4) **Comfort**, assessed

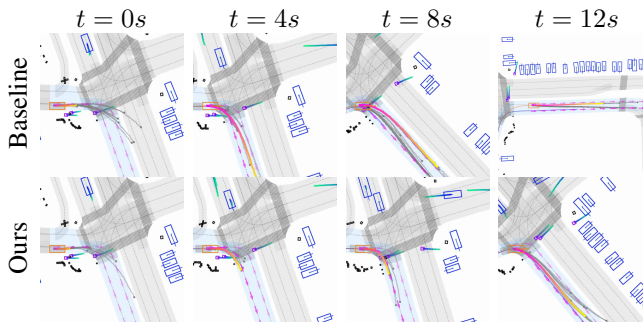


Fig. 4. Comparison in a pedestrian-yielding right-turn scenario. The baseline fails to yield to crossing pedestrians and collides, while our method waits for pedestrians to cross and completes the turn safely.

via acceleration, jerk, and yaw dynamics within empirical thresholds; (5) **Progress**, measured as the AV’s traveled distance relative to the expert driver. We use the non-reactive closed-loop score as our performance evaluation metric.¹

B. Results

We conducted experiments for datasets with 1K, 10K, and 100K scenarios. As shown in Table I, our data augmentation method consistently outperforms the baseline across all dataset sizes, achieving performance improvements of 11.7%, 20.1%, and 3.4%, respectively. A deeper analysis of individual metrics reveals the most significant enhancements in collision and time-to-collision, indicating that our method substantially reduces the likelihood of collisions and enhances safety. Notably, even when using only 10% of the original dataset, our augmented approach achieves better performance than using the full dataset, outperforming the baseline in terms of both collision rate and TTC. This highlights the effectiveness of our data augmentation strategy in improving both safety and model performance.

This improvement is exemplified in a challenging scenario depicted in Figure 4, where a vehicle performs a right turn at an intersection with pedestrians crossing. The baseline model fails to yield to pedestrians and collides with them, demonstrating poor safety awareness. In contrast, our augmented model demonstrates proper pedestrian yielding behavior, waiting for pedestrians to cross safely before completing the turn. Such safety-conscious and interesting behaviors are more prevalent in our augmented dataset due to our sampling and filtering method, which prioritizes vehicles exhibiting dynamic and contextually rich driving patterns.

C. Ablation Studies

Effect of Dataset Size: We conducted ablation studies on dataset size as shown in Figure 5. The results indicate that increasing the dataset size improves performance for both the baseline and our proposed method. Notably, the data augmentation method consistently enhances performance across all dataset sizes, with the most significant gains observed at 10K scenarios. This trend aligns with the expectation that

¹Thresholds used in our filters follow nuPlan’s implementations at https://github.com/motional/nuplan-devkit/blob/master/docs/metrics_description.md.

larger datasets generally lead to better generalization and improved model robustness.

Impact of the Number of Selected Vehicles N_s : To analyze the effect of augmenting with multiple vehicles, we experimented with selecting 1 or 2 additional vehicles per AV scenario and compared the results against the baseline (no augmentation). In the 1K scenario setting, performance improved when augmenting with 2 vehicles, whereas in the 10K scenario dataset, the best performance was achieved with $N_s = 1$. This suggests that while data augmentation is beneficial, excessive augmentation beyond a certain point does not provide further improvements. The results indicate that in low-data regimes, increasing the amount of augmented data is advantageous, but as dataset size grows, additional augmentation yields diminishing improvements.

Influence of the Temperature Parameter τ : The temperature parameter τ in the softmax distribution controls sampling bias toward vehicles with higher heading deviations. Lower τ focuses on dynamic vehicles, while higher τ leads to more uniform sampling. We tested $\tau = 0.1$, $\tau = 0.5$, and uniform sampling ($\tau \rightarrow \infty$). In the 1K dataset, $\tau = 0.5$ outperformed others and even exceeded the 10K baseline only using 10% of it, highlighting the benefit of selective augmentation with limited data. However, uniform sampling degraded performance. For 10K and 100K datasets, all τ values performed similarly, though higher τ was slightly better for 10K and $\tau = 0.5$ for 100K. This suggests τ selection is crucial in low-data regimes.

Per-scene vs Per-ego Sampling: We compared two sampling strategies for vehicle selection across 10K scenarios, as shown in Figure 6. Per-scene sampling outperforms per-ego sampling, which is surprising given that per-ego sampling can theoretically select the best vehicles from the entire dataset. However, per-scene sampling introduces more diverse scenarios to the dataset, which could be the primary driver of the observed performance improvement.

Effect of Filtering Strategies: We evaluated different filter combinations (displacement, TTC, and comfort) on both 1K and 10K scenarios using per-scene sampling, as shown in Figure 6. For these ablations we use $d_{\min} = 3$ m, $\kappa_{\text{comf}} = 5$, and $\kappa_{\text{ttc}} = 0$ violations. Displacement filters consistently improve performance in both dataset sizes, while TTC filters provide additional benefits, particularly in the 10K setting, where displacement + TTC filters achieve performance comparable to the 100K baseline using only 10% of the data. In the 1K setting, displacement alone provides the best performance, significantly surpassing the 10K baseline while using only 10% of that dataset. Interestingly, comfort filters generally reduce performance, which might be related to reduced diversity or suboptimal learning of the safety-comfort tradeoff.

When examining individual metrics in Table I, we observe that displacement filters improve Progress scores and TTC filters enhance TTC metric performance as expected. These results demonstrate that our filtering approach effectively targets specific driving behaviors and safety metrics.

TABLE I
SELECTED PERFORMANCE RESULTS ACROSS DATASET SIZES AND FILTERING STRATEGIES.

Scenarios	Sampling	Filter	Score	Collisions	TTC	Drivable	Comfort	Progress
1K		Baseline	58.60	83.02	76.23	92.45	70.94	71.20
	Per-scene N_s	No filter ($N_s=1, \tau=0.5$)	60.31	82.76	75.48	92.34	68.20	74.66
	Per-scene N_s	+ Disp	65.49	84.51	74.90	93.73	73.73	76.98
	Per-scene N_s	+ Disp + TTC	64.17	87.02	78.63	93.51	70.61	73.96
	Per-scene N_s	+ Disp + TTC + Comfort	56.07	80.47	73.83	91.41	68.36	74.50
10K		Baseline	61.95	83.90	74.91	93.26	76.03	78.72
	Per-scene N_s	No filter ($N_s=1, \tau=0.5$)	72.17	91.29	82.58	95.08	80.68	80.61
	Per-scene N_s	+ Disp	73.98	92.80	84.44	94.55	82.49	81.77
	Per-scene N_s	+ Disp + TTC	74.41	92.34	84.29	96.55	80.84	81.58
	Per-scene N_s	+ Disp + TTC + Comfort	73.69	91.70	86.04	94.72	81.13	80.43
100K		Baseline	74.81	91.23	83.96	97.01	86.57	78.64
	Per-scene N_s	No filter ($N_s=1, \tau=0.5$)	77.38	93.75	84.09	96.97	87.88	80.47

Note: Bold values indicate the highest score in each column for each dataset size.

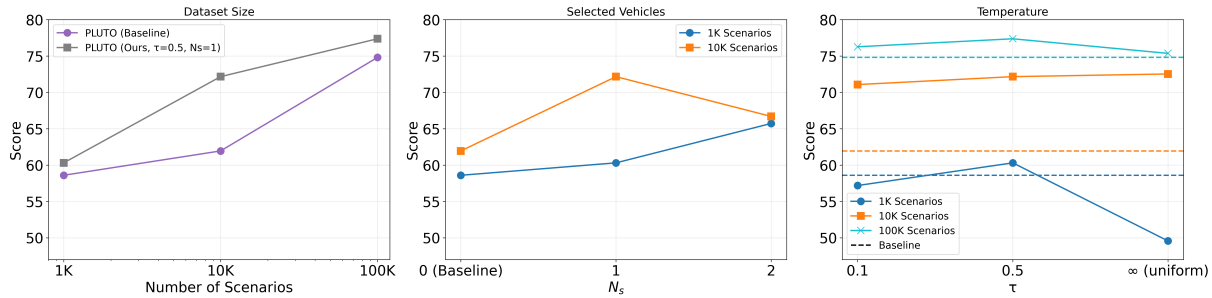


Fig. 5. Ablation experiments 1. (1) Dataset size. (2) Number of selected vehicles N_s . (3) Temperature parameter τ .

V. DISCUSSION

Our approach shows clear performance improvements for imitation learning in autonomous driving with small datasets by selecting suitable trajectories from the surrounding traffic as additional data. We observe that the heading deviation per-

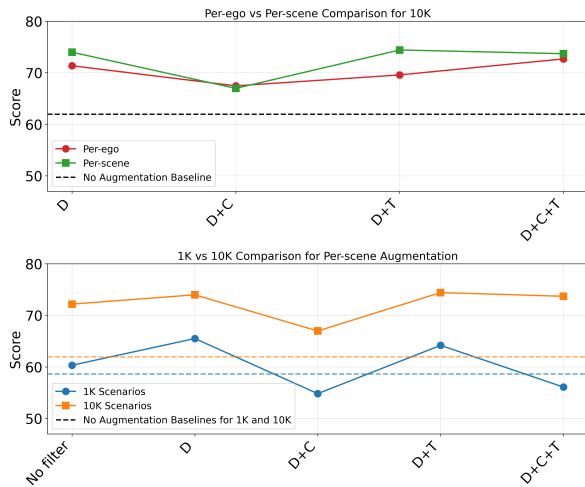


Fig. 6. Ablation experiments 2. Top: Per-scene vs per-ego sampling comparison for 10K scenarios. Bottom: Different filter combinations (D=displacement, T=TTC, C=comfort) on 1K and 10K scenarios using per-scene sampling.

scene sampling and displacement and TTC filtering improve performance, likely due to the increased diversity and informativeness of the training trajectories. While our selection criteria are effective and easy to compute for the nuPlan dataset, future work should investigate if more nuanced metrics for selecting trajectories, such as temporal logic-based filtering [34], can additionally improve the performance. Additionally, although we demonstrated the effectiveness of our approach on the highly diverse nuPlan dataset, further investigation is needed across other datasets and planning algorithms to assess the generalizability of the observed performance gains. Applying our method to perception-based datasets would also require an additional preprocessing step to generate object-level representations [35], which may introduce new challenges related to sensor noise and partial observability. Lastly, the benefits of our method might vary depending on the baseline model architecture and the specific augmentation parameters, such as the vehicle sampling strategy and softmax temperature. Exploring adaptive augmentation policies and dataset curation strategies could further improve robustness and scalability in future work.

VI. CONCLUSION

We propose a data augmentation and filtering strategy using surrounding traffic participants for imitation learning of

object-based path planning for autonomous driving. Our approach introduces vehicle selection criteria that are efficient to compute and align with the expert trajectory distribution. We evaluated our method using the PLUTO planner and the nuPlan dataset, demonstrating that the augmentation strategy consistently improves performance for small to medium-sized expert data sets. Our ablations show that using heading deviation as a sampling strategy and displacement and TTC as selection criteria leads to safe driving behaviors, enriching the imitation learning data set. Beyond autonomous driving, the underlying principle of leveraging observed agent interactions for improved decision-making could extend to other robotics tasks and less structured multi-agent environments, such as aerial traffic or maritime navigation, where large-scale expert demonstrations are difficult to obtain.

ACKNOWLEDGMENT

We thank Chris Ostafew for valuable discussions. This work was supported by the NSF under Grant CNS-2111688.

REFERENCES

- [1] J. Cheng, Y. Chen, and Q. Chen, "Pluto: Pushing the limit of imitation learning-based planning for autonomous driving," 2024. [Online]. Available: <https://arxiv.org/abs/2404.14327>
- [2] Y. Zheng, R. Liang, K. ZHENG, J. Zheng, L. Mao, J. Li, *et al.*, "Diffusion-based planning for autonomous driving with flexible guidance," in *The Thirteenth Conference on Learning Representations*.
- [3] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," in *Conference on Robot Learning*, 2023, pp. 1268–1281.
- [4] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, "nuPlan: A closed-loop ml-based planning benchmark for autonomous vehicles," *arXiv:2106.11810*, 2021.
- [5] B. Jaeger, K. Chitta, and A. Geiger, "Hidden biases of end-to-end driving models," in *Proceedings of the IEEE/CVF Conference on Computer Vision*, 2023, pp. 8240–8249.
- [6] J. Cheng, Y. Chen, X. Mei, B. Yang, B. Li, and M. Liu, "Rethinking imitation-based planners for autonomous driving," in *IEEE Conference on Robotics and Automation (ICRA)*, 2024, pp. 14 123–14 130.
- [7] Y. Lu, J. Fu, G. Tucker, X. Pan, E. Bronstein, R. Roelofs, *et al.*, "Imitation is not enough: Robustifying imitation with reinforcement learning for challenging driving scenarios," in *IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, 2023, pp. 7553–7560.
- [8] P. De Haan, D. Jayaraman, and S. Levine, "Causal confusion in imitation learning," *Advances in neural information processing systems*, vol. 32, 2019.
- [9] C. Gulino, J. Fu, W. Luo, G. Tucker, E. Bronstein, Y. Lu, *et al.*, "Waymax: An accelerated, data-driven simulator for large-scale autonomous driving research," *Advances in Neural Information Processing Systems*, vol. 36, pp. 7730–7742, 2023.
- [10] D. Dauner, M. Hallgarten, T. Li, X. Weng, Z. Huang, Z. Yang, *et al.*, "Navsim: Data-driven non-reactive autonomous vehicle simulation and benchmarking," *Advances in Neural Information Processing Systems*, vol. 37, pp. 28 706–28 719, 2025.
- [11] E. Bronstein, S. Srinivasan, S. Paul, A. Sinha, M. O'Kelly, P. Nikdel, and S. Whiteson, "Embedding synthetic off-policy experience for autonomous driving via zero-shot curricula," in *Proceedings of the 6th Conference on Robot Learning*, 2023, pp. 188–198.
- [12] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research (IJRR)*, 2013.
- [13] M.-F. Chang, J. W. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, *et al.*, "Argoverse: 3D tracking and forecasting with rich maps," in *Conference on Computer Vision and Pattern Recognition*, 2019.
- [14] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, *et al.*, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.
- [15] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD Dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *IEEE Conference on Intelligent Transportation Systems (ITSC)*, 2018, pp. 2118–2125.
- [16] W. Zhan, L. Sun, D. Wang, H. Shi, A. Clausse, M. Naumann, J. Kümmerle, H. Königshof, C. Stiller, A. de La Fortelle, and M. Tomizuka, "INTERACTION Dataset: An INTERNATIONAL, Adversarial and Cooperative motion Dataset in Interactive Driving Scenarios with Semantic Maps," *arXiv:1910.03088*, 2019.
- [17] E. Barmounakis and N. Geroliminis, "On the new era of urban traffic monitoring with massive drone data: The pNEUMA large-scale field experiment," *Transportation Research Part C: Emerging Technologies*, vol. 111, pp. 50–71, 2020.
- [18] U. D. of Transportation Federal Highway Administration, "Next generation simulation (ngsim) vehicle trajectories and supporting data," Provided by ITS DataHub, Tech. Rep., 2016.
- [19] N. Karnchanachari, D. Geromichalos, K. S. Tan, N. Li, C. Eriksen, S. Yaghoubi, *et al.*, "Towards learning-based planning: The nuPlan benchmark for real-world autonomous driving," in *IEEE Conference on Robotics and Automation (ICRA)*, 2024, pp. 629–636.
- [20] Z. Huang, H. Liu, and C. Lv, "Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision*, 2023, pp. 3903–3913.
- [21] S. Sharan, F. Pittaluga, M. Chandraker, *et al.*, "Llm-assist: Enhancing closed-loop planning with language-based reasoning," *arXiv:2401.00125*, 2023.
- [22] K. Guo, Z. Miao, W. Jing, W. Liu, W. Li, D. Hao, and J. Pan, "Lasil: Learner-aware supervised imitation learning for long-term microscopic traffic simulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 15 386–15 395.
- [23] D. Chen and P. Krähénbühl, "Learning from all vehicles," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17 201–17 210, 2022.
- [24] J. Zhang and E. Ohn-Bar, "Learning by watching," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 711–12 721.
- [25] B. R. Kiran, I. Sobh, V. Talpaert, P. Mannion, A. A. A. Sallab, S. Yogamani, and P. Pérez, "Deep reinforcement learning for autonomous driving: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 4909–4926, 2022.
- [26] L. Le Mero, D. Yi, M. Dianati, and A. Mouzakitis, "A survey on imitation learning techniques for end-to-end autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 9, pp. 14 128–14 147, 2022.
- [27] A. O. Ly and M. Akhloofi, "Learning to drive by imitation: An overview of deep behavior cloning methods," *IEEE Transactions on Intelligent Vehicles*, vol. 6, no. 2, pp. 195–209, 2021.
- [28] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," *Robotics: Science and Systems XV*, 2019.
- [29] Comma.ai, "Openpilot: An open-source self-driving car system," 2025. [Online]. Available: <https://github.com/commaai/openpilot>
- [30] K. Wang, B. Kang, J. Shao, and J. Feng, "Improving generalization in reinforcement learning with mixture regularization," *Advances in Neural Information Processing Systems*, vol. 33, pp. 7968–7978, 2020.
- [31] Y. Yu, S. Khadivi, and J. Xu, "Can data diversity enhance learning generalization?" in *Proceedings of the 29th Conference on Computational Linguistics*, 2022, pp. 4933–4945.
- [32] R. Zhang, J. Hou, F. Walter, S. Gu, J. Guan, F. Röhrbein, Y. Du, P. Cai, G. Chen, and A. Knoll, "Multi-agent reinforcement learning for autonomous driving: A survey," *arXiv:2408.09675*, 2024.
- [33] C. Wen, J. Lin, T. Darrell, D. Jayaraman, and Y. Gao, "Fighting copycat agents in behavioral cloning from observation histories," *Advances in Neural Information Processing Systems*, vol. 33, pp. 2564–2575, 2020.
- [34] R. Karagulle, N. Aréchiga, J. DeCastro, and N. Ozay, "Classification of driving behaviors using stl formulas: A comparative study," in *Conference on Formal Modeling and Analysis of Timed Systems*, 2022, pp. 153–162.
- [35] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 47, no. 3, pp. 2020–2036, 2025.