Parting with Misconceptions about Learning-based Vehicle Motion Planning

Anonymous Author(s)
Affiliation
Address
email

Abstract: The release of nuPlan marks a new era in vehicle motion planning research, offering the first large-scale real-world dataset and evaluation schemes requiring both precise short-term planning and long-horizon ego-forecasting. Existing systems struggle to simultaneously meet both requirements. Indeed, we find that these tasks are fundamentally misaligned and should be addressed independently. We further assess the current state of closed-loop planning in the field, revealing the limitations of learning-based methods in complex real-world scenarios and the value of simple rule-based priors such as centerline selection through lane graph search algorithms. More surprisingly, for the open-loop sub-task, we observe that the best results are achieved when using only this centerline as scene context (i.e., ignoring all information regarding the map and other agents). Combining these insights, we propose an extremely simple and efficient planner which outperforms an extensive set of competitors, winning the nuPlan Planning Challenge 2023.

Keywords: Motion Planning, Autonomous Driving, Data-driven Simulation

1 Introduction

Despite learning-based systems' success in vehicle motion planning research [1, 2, 3, 4, 5], a lack of standardized large-scale datasets for benchmarking holds back their transfer from research to applications [6, 7, 8]. The recent release of the nuPlan dataset and simulator [9], a collection of 1300 hours of real-world vehicle motion data, has changed this, enabling the development of a new generation of learned motion planners, which promise reduced manual design effort and improved scalability. Equipped with this new benchmark, we perform the first rigorous empirical analysis on a large-scale, open-source, and data-driven simulator for vehicle motion planning, including a comprehensive set of state-of-the-art (SoTA) planners [10, 11, 12] using the official metrics. Our analysis yields several surprising findings:

Open- and closed-loop evaluation are misaligned. Most learned planners are trained through the supervised learning task of forecasting the ego vehicle's future motion conditioned on a given goal location. We refer to this setting as ego-forecasting [2, 3, 13, 14]. In nuPlan, planners can be evaluated in two ways: by directly measuring ego-forecasting accuracy using distance-based metrics in an open-loop evaluation or by assessing driving-relevant closed-loop metrics such as progress and collision rates in a simulated setting, termed closed-loop evaluation. Our primary contribution lies in uncovering the performance trade-off between the open-loop and closed-loop evaluation schemes. While previous work on the simplistic CARLA simulator [15] has shown that open- and closed-loop evaluation can have little correlation [16], our results indicate a *negative correlation* exists when using nuPlan's metrics. Notably, learned planners excel at ego-forecasting but struggle to make safe closed-loop plans. Rule-based planners exhibit the opposite trend.

Rule-based planning generalizes. We surprisingly find that an established rule-based planning baseline from over twenty years ago [17] surpasses *all SoTA learning-based methods* in terms of

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.

closed-loop evaluation metrics on our benchmark. This contradicts the prevalent motivating claim used in most research on learned planners that rule-based planning faces difficulties in generalization. This was previously only verified on simpler benchmarks [4, 10, 11]. As a result, most current work on learned planning only compares to other learned methods, ignoring rule-based baselines [3, 5, 18].

A centerline is all you need for ego-forecasting. We implement a naïve learned planning baseline which does not incorporate any input about other agents in the scene and merely extrapolates the ego state conditioned on a centerline representation of the desired route. This baseline *sets the new SoTA* for open-loop evaluation on our benchmark. It does not require intricate scene representations (e.g. lane graphs, vectorized maps, rasterized maps, tokenized objects), which have been the central subject of inquiry in previous work [10, 11, 12]. None of these prior studies considered a simple centerline-only representation as a baseline, perhaps due to its extraordinary simplicity.

Long-horizon prediction adds little value. Using the insights gained, we propose novel rule-based and learned planners that achieve SoTA results on the open- and closed-loop evaluation metrics, respectively. We maintain simple input representations so that both of these planners demonstrate significant efficiency. Finally, we introduce a corrective strategy that involves predicting learned offsets to a rule-based plan resulting in a hybrid planner. While this hybrid planning strategy significantly boosts performance in open-loop evaluation compared to the closed-loop planner, we find that it yields no added benefits for closed-loop evaluation due to the incorporated learned component. This outcome suggests that accurate long-horizon prediction, often considered crucial in complex planning scenarios [19, 20], offers little to no additional value for closed-loop planning.

A preliminary version of our approach won the inaugural nuPlan challenge. Given its simplicity, it provides a robust starting point for future motion planning research on nuPlan. We also propose a standardized experimental protocol with fixed training and validation sets, ensuring sufficient data while maintaining fast experimental cycles. We verify that all metrics are well-aligned between the official test set and our validation set based on our submissions to the online leaderboard. All our code, data splits, and models will be publicly released.

2 Related Work

Rule-based planning. Rule-based planners offer a structured, interpretable decision-making framework [17, 21, 22, 23, 24, 25, 26, 27, 28]. They employ explicit rules to determine an autonomous vehicle's behavior (e.g., apply a hard brake when an object is straight ahead). A seminal approach in rule-based planning is the Intelligent Driver Model (`IDM` [17]), which is designed to follow a leading vehicle in traffic while maintaining a safe distance. The desired acceleration along the centerline is determined based on a given target speed and the speed of the leading vehicle so that a safe time gap to the lead vehicle can be ensured given the acceleration and deceleration capabilities of the ego vehicle. Its output is analytically determined and employs several constraints (i.e., hyperparameters). There exist extensions of `IDM` [29] which focus on enabling lane changes on highways. However, this is not the goal of our work. Instead, we extend `IDM` by executing multiple policies with different hyperparameters, and scoring them to select the best option.

To enhance collision avoidance without becoming overly conservative, predictive rule-based planners anticipate future environmental states, enabling informed and contingent driving decisions. This forecasting can either be agent-centric [30, 31, 32], where trajectories are determined for each actor, or environment-centric [4, 33, 34, 20, 35, 36], involving occupancy or cost maps. Additionally, forecasting can be conditioned on the ego-plan, modeling the ego vehicle's influence on the scene's future [37, 19, 38, 39]. We employ an agent-centric forecasting module that is considerably simpler than existing methods, allowing for its use as a starting point in the newly released nuPlan framework.

Ego-forecasting. Unlike predictive planning, ego-forecasting methods use observational data to directly determine the future trajectory. Ego-forecasting approaches include both end-to-end methods that utilize LiDAR scans [40, 41], RGB images [42, 43, 44, 45, 14, 46] or both [13, 47, 5], as well as modular methods involving lower-dimensional inputs like bird's eye view (BEV) grids or

state vectors [26, 48, 11, 49, 50]. A concurrent study introduces a naive MLP inputting the current dynamic state, yielding competitive ego-forecasting results on the nuScenes dataset [51] with no scene context input [52]. Our findings complement these results, differing by evaluating long-term (8s) ego-forecasting in the challenging 2023 nuPlan challenge scenario test distribution [9]. We show that in this setting, completely removing scene context (as in [52]) is harmful, whereas a simple centerline representation of the context is sufficient for strong open-loop performance.

3 Ego-forecasting and Planning are Misaligned

In this section, we provide the relevant background regarding the data-driven simulator nuPlan [9]. We then describe two baselines used in a preliminary experiment to demonstrate that although ego-forecasting and planning are often considered related tasks, they are not well-aligned given their definitions on nuPlan. Improvements in one task can often lead to degradation in the other.

nuPlan. Simulators like nuPlan enable rapid prototyping and testing of motion planners, facilitating swift iteration of ideas. nuPlan constructs a simulated environment as closely as possible to a real-world driving setting through data-driven simulation [53, 54, 55, 56, 57, 58]. This method extracts road maps, traffic patterns, and object properties (positions, orientations, and speeds) from a pre-recorded dataset consisting of 1,300 hours of real-world driving. These elements are then used to initialize scenarios, which are 15-second simulations employed to assess open-loop and closed-loop driving performance. In the open-loop simulation, the entire log is merely replayed (for both the ego vehicle and other actors). Conversely, in closed-loop simulation, the ego vehicle operates under the control of the planner being tested. There are two versions of closed-loop simulation: non-reactive, where all other actors are replayed along their original trajectory, and reactive, where other vehicles employ an `IDM` planner [17], which we describe in more detail in the following.

Metrics. nuPlan offers three official evaluation metrics: open-loop score (OLS), closed-loop score non-reactive (CLS-NR), and closed-loop score reactive (CLS-R). Although CLS-NR and CLS-R are computed identically, they differ in background traffic behavior. Each score is a weighted combination of established metrics, scaled within a range of 0-100. A high open-loop score necessitates low displacement and heading errors between the planner's and recorded trajectories over an extended period (8 seconds). The closed-loop score mandates that the generated plan remains on the road, adheres to the traffic's direction, avoids collisions where the planner is at fault, makes progress, maintains comfort, and observes the speed limit, all of which rely on an accurate short-term trajectory. For additional information on the metrics' composition, please refer to [59].

Intelligent Driver Model. The simple planning baseline `IDM` [17] serves not only as the mechanism for simulating other actors in the CLS-R evaluation within nuPlan, but also as a baseline for the ego-vehicle's planning. The nuPlan map is provided as a graph, with centerline segments functioning as nodes. After choosing a set of such nodes to follow via a graph search algorithm, `IDM` infers a longitudinal trajectory along the selected centerline. Given the current longitudinal position x, velocity v, and distance to the leading vehicle s along the centerline, it iteratively applies the following policy to calculate a longitudinal acceleration:

$$\frac{\mathrm{d}v}{\mathrm{d}t} = a \left(1 - \left(\frac{v}{v_0} \right)^{\delta} - \left(\frac{s^*}{s} \right)^2 \right). \tag{1}$$

The acceleration limit a, target speed v_0, safety margin s^*, and exponent δ are manually selected. Intuitively, the policy uses an acceleration a unless the velocity is already close to v_0 or the leading vehicle is at a distance of only s^*. Additional details and our exact hyper-parameter choices can be found in the supplementary material.

Centerline-conditioned ego-forecasting. We now propose the Predictive Driver Model (Open), i.e., `PDM-Open`, which is a straightforward multi-layer perceptron (MLP) designed to predict future trajectories. The inputs to this MLP are the centerline ($\mathbf{c}$) extracted by `IDM` and the ego history ($\mathbf{h}$). To accommodate the high speeds (reaching up to 15 m/s) and ego-forecasting horizons (extending to 8 seconds) observed in nuPlan, the centerline is sampled with a resolution of 1 meter up to a length

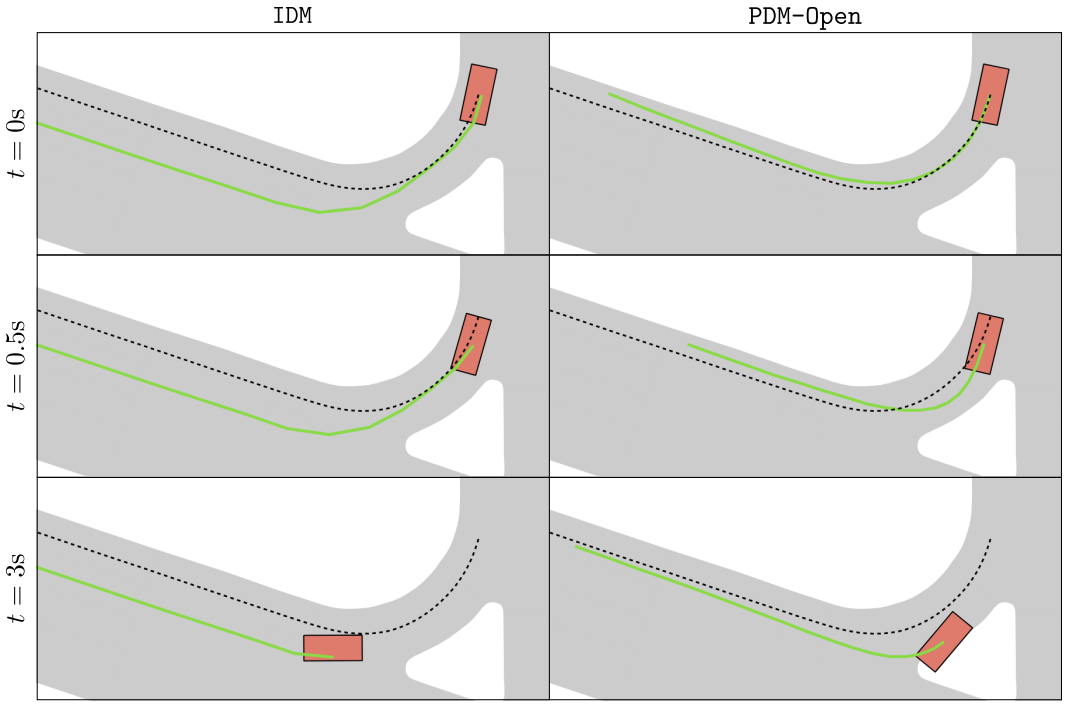

Figure 1: **Planning vs. ego-forecasting.** We present a nuPlan scenario, highlighting the driveable area in grey and the original human trajectory as a dashed black line. In each snapshot, we display the ego agent with its prediction. (Top) observe the significant displacement between the IDM prediction (constrained to a rule-based centerline) and the human trajectory, resulting in low open-loop scores. (Mid + bot) after 0.5 seconds of simulation, the learned PDM-Open planner extrapolates its own errors while making predictions and eventually veers off-road, leading to suboptimal closed-loop scores.

of 120 meters. Meanwhile, the ego history incorporates the positions, velocities, and accelerations of the vehicle over the previous two seconds, sampled at a rate of 5Hz. Both $\mathbf{c}$ and $\mathbf{h}$ are linearly projected to feature vectors of size 512, concatenated, and input to the MLP $\phi_{\texttt{Open}}$ which has two 512-dimensional hidden layers. The output is the future waypoints for an 8-second horizon, spaced 0.5 seconds apart, expressed as $\mathbf{w}_{\texttt{Open}} = \phi_{\texttt{Open}}(\mathbf{c}, \mathbf{h})$. The model is trained using an L_1 loss on our training dataset of 177k samples (elaborated upon in Section 4). By design, PDM-Open is considerably simpler than existing learned planners [10, 12].

OLS vs. CLS. In Table 1, we benchmark the IDM and PDM-Open baselines using the nuPlan metrics. We present two IDM variants with different maximum acceleration values (the default $a = 1.0\text{ms}^{-2}$ and $a = 0.1\text{ms}^{-2}$) and four PDM-Open variants based on witholding different inputs. While IDM demonstrates strong closed-loop performance, PDM-Open excels in open-loop. Reducing IDM's acceleration improves OLS but negatively impacts CLS. For PDM-Open, adding the centerline significantly contributes to ego-forecasting performance, with minor enhancements when adding a 2-second state history. However, including the ego-history leads to a drop in CLS. A clear trade-off between CLS and OLS indicates a misalignment between the goals of ego-forecasting and planning. This sort of inverse correlation on nuPlan is unanticipated, considering the increasing use of ego-forecasting in current planning literature [3, 10, 12, 11].

Method	Centerline	History	CLS-R ↑	CLS-NR ↑	OLS ↑
IDM [17] $a = 1.0$	✓	-	**77**	**76**	38
$a = 0.1$	✓	-	54	66	48
PDM-Open	-	-	51	50	69
	-	✓	38	34	72
	✓	-	54	53	85
	✓	✓	54	50	**86**

Table 1: **OLS-CLS Tradeoff.** Baseline scores on nuPlan with different inputs.

In Fig. 1, we illustrate the misalignment between the OLS and CLS metrics. In the depicted scenario, the rule-based IDM selects a different lane in comparison to the human driver. However, it maintains its position on the road throughout the simulation. This results in a high CLS yet a low OLS. Conversely,

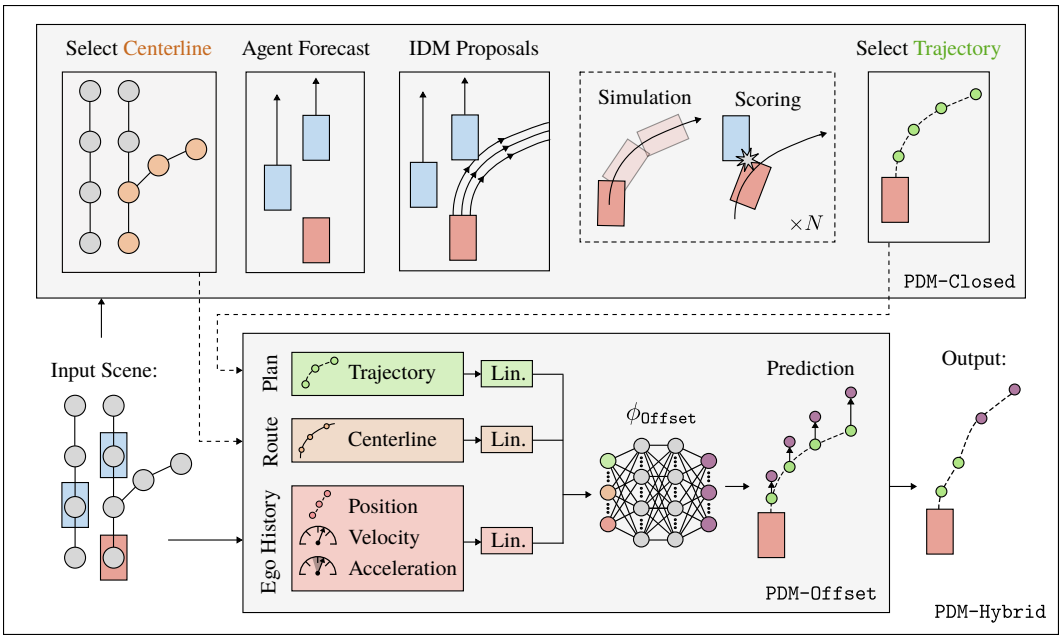

Figure 2: **Architecture.** `PDM-Closed` selects a centerline, forecasts the environment, and creates varying trajectory proposals, which are simulated and scored for trajectory selection. The `PDM-Hybrid` module predicts offsets using the `PDM-Closed` centerline, trajectory, and ego history, correcting only long-term waypoints and thereby limiting the learned model's influence in closed-loop simulation.

the learned `PDM-Open` generates predictions along the lane chosen by the human driver, thereby obtaining a high OLS. Nonetheless, as errors accumulate in its short-term predictions during the simulation [60, 61], the model's trajectory veers off the drivable area, culminating in a subpar CLS.

Improving closed-loop driving. We now extend `IDM` by incorporating several concepts from model predictive control, including forecasting, proposals, simulation, scoring, and selection, as illustrated in Fig. 2 (top). We call this model `PDM-Closed`. Note that as a first step, we still require a graph search to find a sequence of lanes along the route and extract their centerline, as in the `IDM` planner.

Forecasting. In nuPlan, the simulator provides an orientation vector and speed for each dynamic agent such as a vehicle or pedestrian. By linearly extrapolating these values, we can approximate the future positions of the agents up to the forecasting horizon F of 8 seconds at 10Hz.

Proposals. In the process of calibrating the `IDM` planner, we observed a trade-off when selecting a single value for the target speed hyperparameter (v_0), which either yielded aggressive driving behavior or insufficient progress across various scenarios. Consequently, we generate a set of trajectory proposals by implementing `IDM` policies at five distinct target speeds, namely, $\{20\%, 40\%, 60\%, 80\%, 100\%\}$ of the designated speed limit. For each target speed, we also incorporate proposals with three lateral centerline offsets ($\pm$1m and 0m), thereby producing 15 proposals in total. To circumvent computational demands in subsequent stages, the proposals have a reduced horizon of H steps, which corresponds to 4 seconds at a 10Hz.

Simulation. Trajectories in nuPlan are simulated by iteratively retrieving actions from a LQR controller [62] and propagating the ego vehicle with a kinematic bicycle model [63, 64]. We simulate the proposals with the same parameters and a faster re-implementation of this two-stage pipeline. Thereby, the proposals are evaluated based on the expected movement in closed-loop.

Scoring. Each simulated proposal is scored to favor traffic-rule compliance, progress, and comfort. By considering proposals with lateral and longitudinal variety, the planner can avoid collisions with agent forecasts and correct drift that may arise when the controller fails to accurately track the intended trajectory. Furthermore, our scoring function closely resembles the nuPlan evaluation metrics. We direct the reader to the supplementary material for additional details.

Method	Rep.	CLS-R ↑	CLS-NR ↑	OLS ↑	Time ↓
`Urban Driver` [10]	Polygon	44	45	76	64
`GC-PGP` [12]	Graph	54	57	82	100
`PlanCNN` [11]	Raster	72	73	64	43
`IDM` [17]	Centerline	77	76	38	27
`PDM-Open`	Centerline	54	50	**86**	**7**
`PDM-Closed`	Centerline	**92**	**93**	44	91
`PDM-Hybrid`	Centerline	**92**	**93**	84	96
`PDM-Hybrid`*	Graph	**92**	**93**	84	172
Log Replay	*GT*	*80*	*94*	*100*	-

Table 2: **Val14 benchmark.** We show the closed-loop score reactive/non-reactive (CLS-R/CLS-NR), open loop score (OLS) and runtime in ms for several planners. We specify the input representation (Rep.) used by each planner. `PDM-Hybrid` accomplishes strong ego-forecasting (OLS) and planning (CLS). *This is a preliminary version of `PDM-Hybrid` that combined `PDM-Closed` with `GC-PGP` [12], and was used in our online leaderboard submission (Table 3)

Trajectory selection. Finally, `PDM-Closed` selects the highest-scoring proposal which is extended to the complete forecasting horizon F with the corresponding `IDM` policy. If the best trajectory is expected to collide within 2 seconds, the output is overwritten with an emergency brake maneuver.

Enhancing long-horizon accuracy. To integrate the accurate ego-forecasting capabilities of `PDM-Open` with the precise short-term actions of `PDM-Closed`, we now propose a hybrid version of `PDM`, i.e., `PDM-Hybrid`. Specifically, `PDM-Hybrid` uses a learned module `PDM-Offset` to predict offsets to waypoints from `PDM-Closed`, as shown in Fig. 2 (bottom).

In practice, the LQR controller used in nuPlan relies exclusively on the first 2 seconds of the trajectory when determining actions in closed-loop. Therefore, applying the correction only to long-term waypoints (i.e., beyond 2 seconds by default, which we refer to as the correction horizon C) allows `PDM-Hybrid` to maintain closed-loop planning performance. The final planner output waypoints (up to the forecasting horizon F) $\{\mathbf{w}^t_{\texttt{Hybrid}}\}^F_{t=0}$ are given by:

$$\mathbf{w}^t_{\texttt{Hybrid}} = \mathbf{w}^t_{\texttt{Closed}} + \mathbb{1}_{[t>C]}\phi^t_{\texttt{Offset}}(\mathbf{w}_{\texttt{Closed}}, \mathbf{c}, \mathbf{h}). \tag{2}$$

Where $\mathbf{c}$ and $\mathbf{h}$ are the centerline and history (identical to the inputs of `PDM-Open`). $\{\mathbf{w}^t_{\texttt{Closed}}\}^F_{t=0}$ are the `PDM-Closed` waypoints added to the hybrid approach, and $\phi_{\texttt{Offset}}$ is an MLP. Its architecture is identical to $\phi_{\texttt{Open}}$ except for an extra linear projection to accommodate $\mathbf{w}_{\texttt{Closed}}$ as an additional input.

It is important to note that `PDM-Hybrid` is designed with high modularity, enabling the substitution of individual components with alternative options when diverse requirements emerge. For example, we show results with a different open-loop module in the supplementary material. Given its overall simplicity, one interesting approach to explore involves incorporating modular yet differentiable algorithms as components, as seen in [30]. Exploring the integration of these modules within unified multi-task architectures is another interesting direction. We reserve such exploration for future work.

4 Experiments

We now outline our proposed benchmark and highlight the driving performance of our approach.

Val14 benchmark. We offer standardized data splits for training and evaluation. Training uses all 70 scenario types from nuPlan, restricted to a maximum of 4k scenarios per type, resulting in ∼177k training scenarios. For evaluation, we use 100 scenarios of the 14 scenario types considered by the leaderboard, totaling 1,118 scenarios. Despite minor imbalance (all 14 types do not have 100 available scenarios), our validation split aligns with the online leaderboard evaluation (Table 2 and Table 3), confirming the suitability of our Val14 benchmark as a proxy for the online test set.

Baselines. We include several additional SoTA approaches adopting ego-forecasting for planning in our study. `Urban Driver` [10] encodes polygons with PointNet layers and predicts trajectories with

a linear layer after a multi-head attention block. GC-PGP [12] clusters trajectory proposals based on route-constrained lane-graph traversals before returning the most likely cluster center. PlanCNN [11] predicts waypoints using CNN from rasterized grid features without ego-state input. It shares several similarities to ChauffeurNet [8], a seminal work in the field. A preliminary version of PDM-Hybrid, which won the nuPlan competition, used GC-PGP as its ego-forecasting component, and we include this as a baseline. A full description of this version is provided in the supplementary.

Results. Our results are presented in Table 2. Intriguingly, PlanCNN achieves the best CLS among learned planners, possibly due to its design choice of removing ego state from input, trading OLS for enhanced CLS. Contrary to the community's growing preference for graph- and vector-based scene representations in prediction and planning [65, 11, 66, 67], these results show no clear advantage in adopting these methods for the closed-loop task, with the raster-based PlanCNN also offering a lower runtime. Surprisingly, the simplest rule-based approach in our study, IDM, outperforms the best learned planner, PlanCNN. Moreover, we observe PDM-Closed's advantages over IDM in terms of CLS: an improvement from 76-77 to 92-93 as a result of the ideas from Section 3.

Notably, the centerline representation serves as a highly valuable prior for achieving the SoTA OLS of 86 with PDM-Open in a runtime of only 7ms. Next, despite PDM-Closed's unsatisfactory 44 OLS, PDM-Hybrid successfully combines PDM-Closed with PDM-Open. Both the centerline and graph versions of PDM-Hybrid achieve identical scores in our evaluation. However, the final centerline version, using PDM-Open instead of GC-PGP, is more efficient during inference.

Finally, the privileged approach of outputting the ground-truth ego future trajectory (log replay) fails to achieve a perfect CLS, in part due to the nuPlan framework's LQR controller occasionally drifting from the provided trajectory. PDM-Hybrid compensates for this by evaluating proposals based on the expected controller outcome, causing it to match/outperform log replay in closed-loop evaluation.

Challenge. The 2023 nuPlan challenge saw the preliminary (graph) version of PDM-Hybrid rank first out of 25 participating teams. The leaderboard considers the mean of CLS-R, CLS-NR, and OLS. While open-loop performance lagged slightly, closed-loop performance excelled, resulting in an overall SoTA score. Unfortunately, due to the closure of the leaderboard, our final (centerline) version of PDM-Hybrid that replaces GC-PGP with the simpler PDM-Open module could not be benchmarked.

Method	CLS-R ↑	CLS-NR ↑	OLS ↑	Score ↑
PDM-Hybrid*	**93**	**93**	83	**90**
hoplan	89	88	85	87
pegasus_multi_path	82	85	**88**	85
Urban Driver [10]	68	70	86	75
IDM [17]	72	75	29	59

Table 3: **2023 nuPlan Challenge.**

Importantly, near identical scores were recorded for our submission on both our Val14 benchmark (Table 2) and the official leaderboard (Table 3). Note that the Urban Driver and IDM results on the leaderboard are provided by the nuPlan team, so they likely use different training data and hyper-parameters than our implementations from Table 2.

Ablation Study. We delve into our design choices through an ablation study in Table 4. Table 4a displays PDM-Hybrid's closed-loop score reactive (CLS-R) and open-loop score (OLS) with varied correction horizons (C) from 0s to 3s. Applying the waypoint correction to all waypoints (i.e., $C = 0$), outperforms PDM-Open in OLS (87 vs. 86, see Table 2) but leads to a substantial drop in CLS-R compared to the default value of $C = 2$. On the other hand, a noticeable OLS decline occurs when initiating corrections deeper into the trajectory (e.g., $C = 3$), with minimal impact on CLS-R.

For PDM-Closed (Table 4b), we compare CLS-R and runtime (ms) with the base planner across three scenarios: removing lateral centerline offsets ("lat."), longitudinal IDM proposals ("lon."), and environment forecasting ("cast."). Our analysis reveals that eliminating proposals diminishes CLS-R effectiveness but accelerates runtimes. Performance significantly drops when excluding the forecasting used for creating and evaluating proposals. However, the runtime remains nearly identical, showing the effectiveness of the simple forecasting mechanism.

C	CLS-R ↑	OLS ↑
0.0s	58	**87**
2.0s	**92**	84
2.5s	**92**	84
3.0s	**92**	72

(a) `PDM-Hybrid`

Method	CLS-R ↑	Time ↓
Base	**92**	91
No lat.	89	**55**
No lon.	88	64
No cast.	86	90

(b) `PDM-Closed`

Method	OLS ↑
Baseline	**86**
Shorter centerline	84
Coarser centerline	**86**
Smaller MLP	84

(c) `PDM-Open`

Table 4: **Ablation Study.** We show the closed-loop score reactive (CLS-R), open loop score (OLS) and runtime in ms. We investigate (a) varying correction horizons for `PDM-Hybrid`, (b) ignoring sub-modules of `PDM-Closed`, and (c) the effects of input and architecture choices on `PDM-Open`. The default configuration, highlighted in gray, achieves the best trade-offs.

As for `PDM-Open` (Table 4c), we test three variations: a shorter centerline (30m vs. 120m), a coarser centerline (every 10m vs. 1m), and a smaller MLP with a reduced hidden dimension (from 512 to 256). Both a smaller MLP and a reduced centerline length lead to performance degradation, but the impact remains relatively minor compared to disregarding the centerline altogether (Table 1, OLS=72). Meanwhile, the impact of a coarser centerline is negligible.

5 Discussion

We identify prevalent misconceptions in learning-based vehicle motion planning and provide evidence in our paper that challenges these notions. We also introduce `PDM-Hybrid` which surpasses a comprehensive set of competitors on nuPlan and claimed victory in the 2023 nuPlan challenge.

Although rule-based planning is often criticized for its limited generalization, our results demonstrate strong performance in the closed-loop nuPlan task which best resembles real-world evaluation. Notably, open-loop success in part requires a *trade-off in closed-loop performance*. Consequently, imitation-trained ego-forecasting methods fare poorly in closed-loop. This suggests that rule-based planners remain promising and warrant further exploration. At the same time, given their poor performance out-of-the-box, there is room for improvement in imitation-based methods on nuPlan.

Integrating the strengths of closed-loop planning and open-loop ego-forecasting, we present a hybrid model. However, this does not enhance closed-loop driving performance; instead, it boosts open-loop performance *while executing identical driving maneuvers*. We conclude that considering open-loop ego-forecasting as a prerequisite for achieving long-term planning goals is misleading.

Acknowledging the potential importance of long-term ego-forecasting for interpretability and assessing human-like behavior, we propose focusing this evaluation on the short horizon (e.g., 2 seconds) relevant for closed-loop driving. The current nuPlan OLS definition, requiring a unimodal 8-second ego-forecast, may only be useful for *alternate applications* like setting goals for background agents in data-driven traffic simulations or allocating computational resources better, e.g. to prioritize perception or prediction in areas the ego-vehicle is expected to traverse. We discourage the use of open-loop metrics as a primary indicator of planning performance [18].

Limitations. While we significantly improve upon the established `IDM` model, `PDM` still does not execute lane-change maneuvers. Lane change attempts often lead to collisions when the ego-vehicle is between two lanes, resulting in a high penalty as per the nuPlan metrics. `PDM` relies on HD maps and precise offboard perception [68, 9] that may be unavailable in real-world driving situations. Moreover, our experiments, aside from the held-out test set, have not specifically evaluated the model's generalization capabilities when encountering distributional shifts, such as unseen towns or novel scenario types. They were all conducted on a single simulator, nuPlan. Therefore, it is important to recognize the limitations inherent in nuPlan's data-driven simulation approach. When a planner advances more rapidly than the human driving log, objects materialize abruptly in front of the ego-vehicle during simulation. For CLS-NR, vehicles move independently as observed in reality, disregarding the ego agent, leading to excessively aggressive behavior. Conversely, CLS-R background agents rely on `IDM` and adhere strictly to the centerline, leading to unrealistically passive behavior. We see high value in developing a more refined reactive environment for future work.

References

[1] F. Codevilla, M. Miiller, A. López, V. Koltun, and A. Dosovitskiy. End-to-end driving via conditional imitation learning. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2018.

[2] 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.

[3] N. Rhinehart, R. McAllister, K. Kitani, and S. Levine. PRECOG: prediction conditioned on goals in visual multi-agent settings. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2019.

[4] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun. End-to-end interpretable neural motion planner. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[5] 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)*, 2022.

[6] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba. End to end learning for self-driving cars. *arXiv.org*, 1604.07316, 2016.

[7] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J. M. Allen, V. D. Lam, A. Bewley, and A. Shah. Learning to drive in a day. *arXiv.org*, abs/1807.00412, 2018.

[8] M. Bansal, A. Krizhevsky, and A. S. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. In *Proc. Robotics: Science and Systems (RSS)*, 2019.

[9] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. M. Wolff, A. H. Lang, L. Fletcher, O. Beijbom, and S. Omari. nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2021.

[10] 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.

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

[12] M. Hallgarten, M. Stoll, and A. Zell. From Prediction to Planning With Goal Conditioned Lane Graph Traversals. *arXiv.org*, 2302.07753, 2023.

[13] A. Prakash, K. Chitta, and A. Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[14] K. Chitta, A. Prakash, and A. Geiger. Neat: Neural attention fields for end-to-end autonomous driving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[15] 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.

[16] 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.

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

[18] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, L. Lu, X. Jia, Q. Liu, J. Dai, Y. Qiao, and H. Li. Planning-oriented autonomous driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[19] N. Rhinehart, J. He, C. Packer, M. A. Wright, R. McAllister, J. E. Gonzalez, and S. Levine. Contingencies from observations: Tractable contingency planning with learned behavior models. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2021.

[20] A. Cui, A. Sadat, S. Casas, R. Liao, and R. Urtasun. Lookout: Diverse multi-future prediction and planning for self-driving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, 2021.

[21] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. M. Oakley, M. Palatucci, V. R. Pratt, P. Stang, S. Strohband, C. Dupont, L. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. R. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. V. Nefian, and P. Mahoney. Stanley: The robot that won the DARPA grand challenge. *Journal of Field Robotics (JFR)*, 23 (9):661–692, 2006.

[22] A. Bacha, C. Bauman, R. Faruque, M. Fleming, C. Terwelp, C. Reinholtz, D. Hong, A. Wicks, T. Alberi, D. Anderson, et al. Odin: Team victortango's entry in the darpa urban challenge. *Journal of Field Robotics (JFR)*, 25(8), 2008.

[23] J. J. Leonard, J. P. How, S. J. Teller, M. Berger, S. Campbell, G. A. Fiore, L. Fletcher, E. Frazzoli, A. S. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. R. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. E. Antone, R. Galejs, S. Krishnamurthy, and J. Williams. A perception-driven autonomous urban vehicle. *Journal of Field Robotics (JFR)*, 25(10):727–774, 2008.

[24] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, et al. Autonomous driving in urban environments: Boss and the urban challenge. *Journal of Field Robotics (JFR)*, 25(8), 2008.

[25] C. Chen, A. Seff, A. L. Kornhauser, and J. Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proc. of the IEEE International Conf. on Computer Vision (ICCV)*, pages 2722–2730, 2015.

[26] A. Sauer, N. Savinov, and A. Geiger. Conditional affordance learning for driving in urban environments. In *Proc. Conf. on Robot Learning (CoRL)*, 2018.

[27] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong. Baidu apollo EM motion planner. *arXiv.org*, 1807.08048, 2018.

[28] A. Sadat, M. Ren, A. Pokrovsky, Y. Lin, E. Yumer, and R. Urtasun. Jointly learnable behavior and trajectory planning for self-driving vehicles. In *Proc. IEEE International Conf. on Intelligent Robots and Systems (IROS)*, 2019.

[29] A. Kesting, M. Treiber, and D. Helbing. General lane-changing model mobil for car-following models. *Transportation Research Record*, 1999(1), 2007.

[30] P. Karkus, B. Ivanovic, S. Mannor, and M. Pavone. Diffstack: A differentiable and modular control stack for autonomous vehicles. In *Proc. Conf. on Robot Learning (CoRL)*, 2022.

[31] M. H. Danesh, P. Cai, and D. Hsu. Leader: Learning attention over driving behaviors for planning under uncertainty. In *Proc. Conf. on Robot Learning (CoRL)*, 2022.

[32] Z. Huang, H. Liu, J. Wu, and C. Lv. Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving. *arXiv.org*, 2207.10422, 2022.

[33] A. Sadat, S. Casas, M. Ren, X. Wu, P. Dhawan, and R. Urtasun. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[34] S. Casas, A. Sadat, and R. Urtasun. Mp3: A unified model to map, perceive, predict and plan. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[35] B. Wei, M. Ren, W. Zeng, M. Liang, B. Yang, and R. Urtasun. Perceive, attend, and drive: Learning spatial attention for safe self-driving. *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2021.

[36] S. Hu, L. Chen, P. Wu, H. Li, J. Yan, and D. Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2022.

[37] H. Song, W. Ding, Y. Chen, S. Shen, M. Y. Wang, and Q. Chen. Pip: Planning-informed trajectory prediction for autonomous driving. In *Proc. of the European Conf. on Computer Vision (ECCV)*, 2020.

[38] Y. Chen, P. Karkus, B. Ivanovic, X. Weng, and M. Pavone. Tree-structured policy planning with learned behavior models. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[39] Z. Huang, H. Liu, and C. Lv. Gameformer: Game-theoretic modeling and learning of transformer-based interactive prediction and planning for autonomous driving. *arXiv.org*, 2303.05760, 2023.

[40] N. Rhinehart, R. McAllister, and S. Levine. Deep imitative models for flexible inference, planning, and control. In *Proc. of the International Conf. on Learning Representations (ICLR)*, 2020.

[41] A. Filos, P. Tigas, R. McAllister, N. Rhinehart, S. Levine, and Y. Gal. Can autonomous vehicles identify, recover from, and adapt to distribution shifts? In *Proc. of the International Conf. on Machine learning (ICML)*, 2020.

[42] H. Xu, Y. Gao, F. Yu, and T. Darrell. End-to-end learning of driving models from large-scale video datasets. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3530–3538, 2017.

[43] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl. Learning by cheating. In *Proc. Conf. on Robot Learning (CoRL)*, 2019.

[44] E. Ohn-Bar, A. Prakash, A. Behl, K. Chitta, and A. Geiger. Learning situational driving. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[45] 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.

[46] P. Wu, X. Jia, L. Chen, J. Yan, H. Li, and Y. Qiao. Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[47] D. Chen and P. Krähenbühl. Learning from all vehicles. In *CVPR*, 2022.

[48] 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.

[49] M. Vitelli, Y. Chang, Y. Ye, A. Ferreira, M. Wołczyk, B. Osiński, M. Niendorf, H. Grimmett, Q. Huang, A. Jain, et al. Safetynet: Safe planning for real-world self-driving vehicles using machine-learned policies. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2022.

[50] S. Pini, C. S. Perone, A. Ahuja, A. S. R. Ferreira, M. Niendorf, and S. Zagoruyko. Safe real-world autonomous driving by learning to predict and plan with a mixture of experts. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[51] 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.

[52] 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.

[53] L. Bergamini, Y. Ye, O. Scheel, L. Chen, C. Hu, L. D. Pero, B. Osinski, H. Grimmett, and P. Ondruska. Simnet: Learning reactive self-driving simulations from real-world observations. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2021.

[54] S. Suo, S. Regalado, S. Casas, and R. Urtasun. Trafficsim: Learning to simulate realistic multi-agent behaviors. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[55] Z. Zhong, D. Rempe, D. Xu, Y. Chen, S. Veer, T. Che, B. Ray, and M. Pavone. Guided conditional diffusion for controllable traffic simulation. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[56] D. Xu, Y. Chen, B. Ivanovic, and M. Pavone. Bits: Bi-level imitation for traffic simulation. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[57] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool. TrafficBots: Towards world models for autonomous driving simulation and motion prediction. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[58] L. Feng, Q. Li, Z. Peng, S. Tan, and B. Zhou. Trafficgen: Learning to generate diverse and realistic traffic scenarios. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[59] Motional. nuplan metrics, 2023. URL https://nuplan-devkit.readthedocs.io/en/latest/metrics_description.html.

[60] S. Ross, G. J. Gordon, and D. Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

[61] 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.

[62] Y. Tassa, N. Mansard, and E. Todorov. Control-limited differential dynamic programming. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2014.

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

[64] P. Polack, F. Altché, B. d'Andréa Novel, and A. de La Fortelle. The kinematic bicycle model: A consistent model for planning feasible trajectories for autonomous vehicles? In *Proc. IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[65] N. Deo, E. M. Wolff, and O. Beijbom. Multimodal Trajectory Prediction Conditioned on Lane-Graph Traversals. In *Proc. Conf. on Robot Learning (CoRL)*, 2021.

[66] N. Nayakanti, R. Al-Rfou, A. Zhou, K. Goel, K. S. Refaat, and B. Sapp. Wayformer: Motion forecasting via simple and efficient attention networks. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[67] A. Cui, S. Casas, K. Wong, S. Suo, and R. Urtasun. Gorela: Go relative for viewpoint-invariant motion forecasting. In *Proc. IEEE International Conf. on Robotics and Automation (ICRA)*, 2023.

[68] C. R. Qi, Y. Zhou, M. Najibi, P. Sun, K. Vo, B. Deng, and D. Anguelov. Offboard 3d object detection from point cloud sequences. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2021.