This ICCV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

INT2: Interactive Trajectory Prediction at Intersections

Zhijie Yan[1,3]*, Pengfei Li[1], Zheng Fu[1,4], Shaocong Xu[1,5], Yongliang Shi[1], Xiaoxue Chen[1],
Yuhang Zheng[1,3], Yang Li[1], Tianyu Liu[1,6], Chuxuan Li[1], Nairui Luo[2], Xu Gao[2],
Yilun Chen[1], Zuoxu Wang[3], Yifeng Shi[2], Pengfei Huang[1], Zhengxiao Han[1,7], Jirui Yuan[1],
Jiangtao Gong[1], Guyue Zhou[1], Hang Zhao[8], Hao Zhao[1]†
[1]AIR, THU, [2]Baidu Inc., [3]SMEA, BUAA
[4]SVM, THU, [5]XMU, [6]ECE, HKUST, [7]BUCT, [8]IIIS, THU
yanzhijie@buaa.edu.cn, zhaohao@air.tsinghua.edu.cn

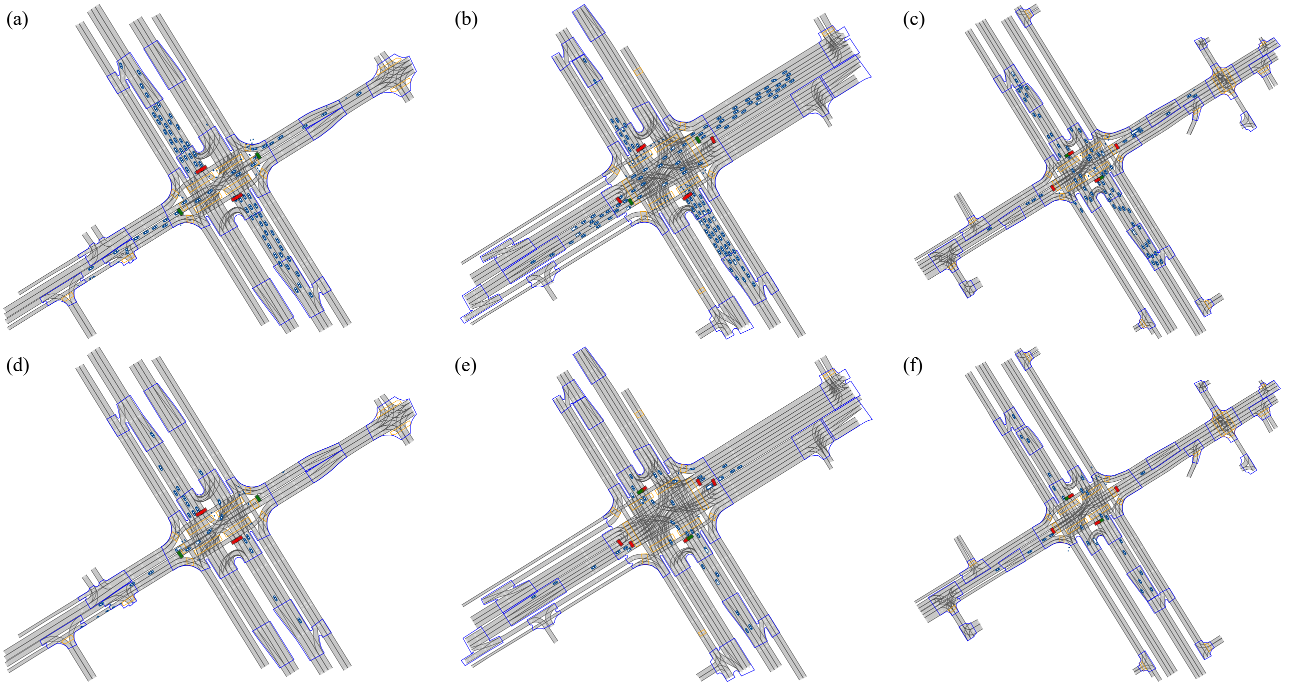


Figure 1. A comparison between rush-hour (a, b, c) and non-rush-hour (d, e, f) traffic at the same intersection in our INT2 dataset. They are naturally treated as two different trajectory prediction domains.

Abstract

*Motion forecasting is an important component in autonomous driving systems. One of the most challenging problems in motion forecasting is interactive trajectory prediction, whose goal is to jointly forecasts the future trajectories of interacting agents. To this end, we present a large-scale interactive trajectory prediction dataset named **INT2** for **INT**eractive trajectory prediction at **INT**ersections. INT2 includes 612,000 scenes, each lasting 1 minute, containing up to 10,200 hours of data. The agent trajectories are auto-labeled by a high-performance offline temporal detection and fusion algorithm, whose quality is further inspected by human judges. Vectorized semantic maps and traffic light information are also included in INT2. Additionally, the dataset poses an interesting domain mismatch challenge. For each intersection, we treat rush-hour and non-rush-hour segments as different domains. We benchmark the best open-sourced interactive trajectory prediction method on INT2 and Waymo Open Motion, under in-domain and cross-domain settings. The dataset, code and models are publicly available at https://github.com/AIR-DISCOVER/INT2.*

* Work done while at AIR, THU.

† Corresponding author.

Sponsored by Baidu Inc. through Apollo-AIR Joint Research Center.

1. Introduction

Autonomous driving is one of the most developed application fields of computer vision, with many heavily studied sub-tasks like 3D detection [36, 37, 50], segmentation [44, 47] and localization [2, 34]. More recently, the research community has directed its focus towards the burgeoning problem of motion forecasting, which seeks to augment down-stream decision making mechanisms by providing enhanced insights into the future trajectories of objects.

Motion forecasting is indeed an old problem in the computer vision community. Conventional datasets focus on forecasting pedestrian trajectory in campus [23, 32, 43]. Whilst methods developed on these legacy datasets can potentially be applied to autonomous driving scenarios, they treat each agent separately or say they are **not** interactive. If we forecast the future trajectories of two interacting agents separately, they may crash into each other, which makes little sense for down-stream decision making modules.

To this end, interactive trajectory prediction is recently formalized by the Waymo Open Motion dataset [7] which benchmarks the trajectory prediction accuracy of interacting agents in diverse scenarios. In this paper, we contribute another large-scale interactive trajectory prediction dataset to the community, with a focus on intersections. Our dataset is named as **INT2**, which is short for **INT**eractive trajectory prediction at **INT**ersections. INT2 has the following features:

(1) **High quality.** INT2 is captured at 16 different intersections by a multi-sensor system and an offline detection and tracking algorithm stack. The multi-sensor system consists of several RGB cameras and LiDARs mounted on poles. Sensors are calibrated routinely. The offline algorithm stack applies state-of-the-art 3D detection and tracking algorithms on raw data without the concern of latency, and fuses the results with production-level rules.

(2) **Large scale.** INT2 contains about 612,000 scenes with each 1 minute long, counting up to 10,200 hours. By contrast, Waymo Open Motion has about 100,000 scenes with every 20 seconds long, counting up to 570 hours. Apart from the total scale, traffic in each 1-minute segment can cover one complete traffic light cycle.

(3) **Rich information.** INT2 has rich information in several terms. INT2 provides vectorized road maps at all 16 intersections, which is a common input to state-of-the-art trajectory prediction algorithms. INT2 provides 3D agent boxes for small and large vehicles, pedestrians and cyclists. INT provides temporal traffic light information and links to the lanes they have control over.

Besides the aforementioned three standard merits, INT2 is suitable for the study of domain mismatch problem thanks to the fact that it is captured at intersections.

Firstly, domain mismatch is as important for interactive trajectory prediction as for other sensing problems. Since trajectory prediction models are usually used in an iterative manner in practice, minor errors caused by domain mismatch would explode over several rounds. For example, if we predict the future trajectory for 8 seconds and then use the predicted trajectory for another 8 seconds, errors caused by domain mismatch would get larger.

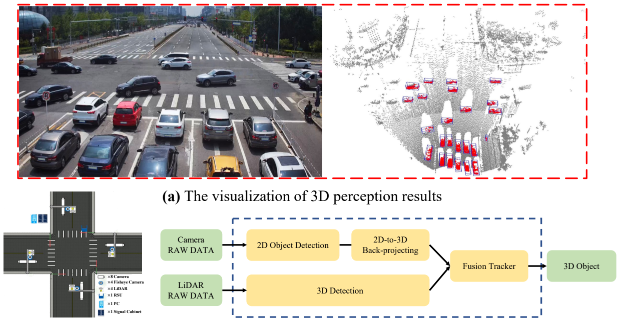
Secondly, it is difficult to properly define domains for the trajectory prediction problem. We ask readers to think about this problem: given two different traffic scenarios, how could we tell whether they are sampled from the same domain or not? Since state-of-the-art trajectory prediction methods all exploit vectorized maps as input, the ever-changing map information in existing datasets makes the definition of domains challenging. In our INT2 dataset, by contrast, the definition of domains becomes natural as trajectories are captured at fixed intersections so that we can use rush-hour and non-rush-hour data as two clearly different domains (Fig. 1), while bypassing the impact of maps.

For initial benchmarking, we provide in-domain and cross-domain interactive trajectory prediction results using the state-of-the-art method M2I [39], pointing to interesting and practical challenges. Baseline training framework and pre-trained models will be released along with the dataset.

2. Related Work

Autonomous Driving Dataset. The key techniques for a self-driving car include solving tasks like 3D construction, scene understanding, segmentation, motion prediction, etc[17, 24, 25, 42, 49]. In past decades, a variety of datasets and benchmarks, have been released to push forward the development of Autonomous Driving. Among them, KITTI[10] and Torontocity[40] are proposed to evaluate visual/LiDAR reconstruction, segmentation, optical flow, stereo and road segmentation tasks. For a better understanding of complex scenes, Cityscapes [6] is to train and test approaches for pixel-level and instance-level semantic segmentation, while Panoptic nuscenes[8] is to evaluate LiDAR panoptic segmentation and tracking. Besides, Waymo adopts panoramic video for Panoptic Segmentation to reason about their surroundings in terms of semantic and geometry properties[29]. Compared with KITTI and Cityscapes, ApolloScape[15, 16] contains much larger and richer labeling including holistic semantic dense point cloud, stereo, per-pixel semantic labeling, lanemark labeling, instance segmentation, 3D car instance, high accurate location for every frame in various driving videos. Since no single type of sensor is sufficient for perception, the first multimodal dataset nuScenes[3] carry full autonomous vehicle sensors (cameras, radars and lidar) suited for detection and tracking.

On top of the scene understanding task, high-quality motion data rich in both interactions and annotations to develop motion planning models are necessary. Several



(a) The visualization of 3D perception results

(b) Schematic diagram of data acquisition system

(c) The overall perception pipeline

Figure 2. Data Acquisition System. (a) On the left is the original scene, and on the right is the result of 3D perception. (b) Shows the schematic diagram of our data collection system. (c) Illustrates the data collection process.

datasets[4, 14, 38] have been developed for motion forecasting in large-scale real-world urban driving environments. As predicting individual object motion is not sufficient, [7] introduces the most diverse interactive motion dataset and provides specific labels for interacting objects suitable for developing joint prediction models. What's more, vehicle-infrastructure cooperation is considered an effective paradigm for autonomous driving, and DAIR-V2X[45] is the first large-scale, multi-modality, multi-view dataset from real scenarios to accelerate computer vision research for Vehicle-Infrastructure Cooperative Autonomous Driving (VICAD).

Trajectory Prediction. Understanding agent behavior is critical for autonomous moving platforms. Social LSTM [1] is proposed to model human movement and predict their future trajectories in crowded scenes. Human motion is inherently multi-modal in dynamic scenes, so GCN[13, 30] is adopted to model motion histories and predicts socially plausible future behavior. On top of this, DESIRE[22] accounts for scene context as well as the interactions among the agents. [28] focuses on long-term trajectory forecasting and proposes a scene-compliant trajectory prediction network to resolve it. [11] leverages graph representations of the High Definition Map and sparse projections to generate the future position probability distribution of an agent.

Forecasting the trajectory of multiple agents is necessary for planning in dynamic traffic environments. Depending on the abundant semantic labels, lanes and spatial locality information of map, [5, 9, 26, 33] predict the trajectories of multiple agents by leveraging graph-structured models. TNT[48] and DenseTNT[12] generate state sequences conditioned on targets for trajectory prediction of vehicles and pedestrians. Due to its attention mechanism, Transformer-based multi-agent models[27, 31, 35, 46] can naturally model the interaction between different agents. Specifically, Scene Transformer[31] provides a multi-agent model for predicting the behavior of all agents. AgentFormer[46] leverages a novel agent-aware attention mechanism to simultaneously model the time and social dimensions. Motion transformer[35] models trajectory prediction as the joint optimization of global intention localization and local movement refinement, and incorporates spatial intention priors by adopting learnable motion query pairs. Besides, M2I[39] models pairs of interaction agents as pairs of influencers and reactors, then leverages a marginal prediction model and a conditional prediction model to predict trajectories for the influencers and reactors, respectively. In this work, We provide a rich set of cross-domain multi-agent interaction data in traffic scenarios based on M2I.

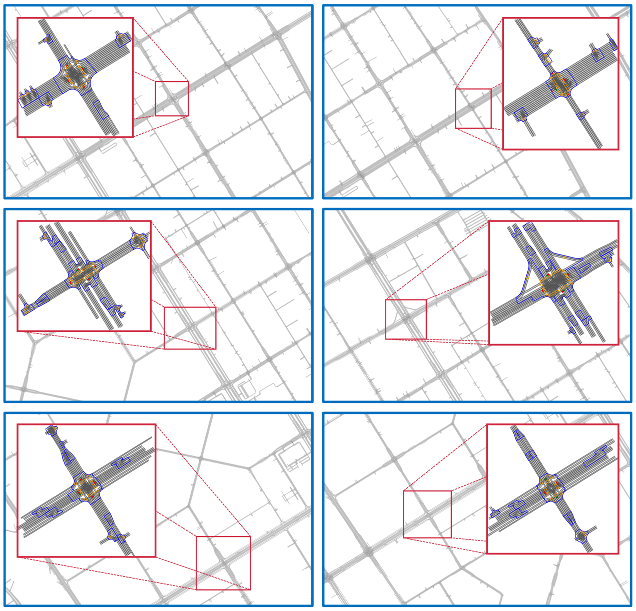


Figure 3. An illustration of the vectorized maps of our intersections, with more figures to be found in the supplementary material.

3. The INT2 Dataset

3.1. Data acquisition system

Our INT2 dataset was collected at 16 intersections in one of the largest cities in the world, using a sensor system consisting of 8 groups of cameras, 4 groups of Fisheye cameras, and 4 groups of LiDARs, as shown in Fig. 2b. Of the 8 cameras, 4 are oriented toward the intersection and 4 are oriented toward the entrance lanes. Additionally, 4 groups of LiDARs are oriented towards the intersection, while the 4 Fisheye cameras are oriented towards the ground, covering the area between the entrance lane and the intersection. The cameras and Fisheye cameras use an RGB color space, with a 25Hz sampling frequency, 1920 × 1080 resolution, and a JPEG compression coding scheme. The LiDARs are 300-line LiDAR with a 10Hz sampling frequency, 100° horizontal FOV, and -30° to 10° vertical FOV, with a visual range of

Table 1. Comparison between our INT2 dataset and several related trajectory prediction datasets.

	# unique tracks	Avg track length	Time horizon	# segments	Segment duration	Total time	# object types	Boxes	Offline perception	Interactions	Traffic signal states
Lyft [18]	53.4M	1.8 s	5 s	170k	25 s	1118 h	3	2D	-	-	✓
NuSc [3]	4.3k	-	6 s	1k	20 s	5.5 h	1	3D	-	-	-
Argo [4]	11.7M	2.48 s	3 s	324k	5 s	320 h	1	-	-	-	-
Inter [20]	40k	19.8 s	3 s	-	-	16.5 h	1	2D	✓	✓	-
Waymo [29]	7.64M	7.04 s	8 s	104k	20 s	574 h	3	3D	✓	✓	✓
Argo2 [41]	13.9M	5.16 s	6 s	250k	-	763 h	5	3D	✓	✓	-
INT2	**106.8M**	**25.58 s**	**8 s**	**612k**	**60 s**	**10200 h**	3	3D	✓	✓	✓

Table 2. Agent-agent and agent-boundary collision rate at 16 intersections.

Scenario ID	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**Mean**
Agent-agent	0.0071	0.0114	0.0120	0.0091	0.0508	0.0180	0.0106	0.0069	0.0127	0.0253	0.0151	0.0088	0.0134	0.0128	0.0088	0.0063	0.0142
Agent-boundary	0.2395	0.1960	0.1896	0.1203	0.1612	0.1578	0.2666	0.1983	0.3025	0.1568	0.2185	0.1779	0.1907	0.2868	0.2741	0.1819	0.2074

280 m and an accuracy of ±3cm. Since there are multiple sensors and our dataset spans almost one year, transformation matrices between sensors may change due to various mechanical factors. For this, we calibrate the sensor system using human-annotated correspondence routinely during the whole time span of INT2, guaranteeing the quality of post-fusion of detection results from several sources.

The data collected by sensors is processed into structured data and sent to the roadside communication computing unit (RSCU) for computational processing. The perception pipeline takes input from different sensors and undergoes 2D object detection, 2D-to-3D back-projecting, and 3D fusion tracking to generate the fused 3D perception results. Specifically, we use a two-stage 3D monocular detection backbone for camera data, and PointPillars [21] for LiDAR data to generate 3D perception results, as shown in Fig. 2a. The 3D fusion tracking module fuses 3D perception results (including obstacle position, size, orientation, covariance, et. al.) from different sensors using a data association algorithm based on the Kalman filter and probability distribution. The fused results are generated in the world coordinate system. The overall pipeline is depicted in Fig. 2c.

We note that all our ground truth generation algorithms (deep learning based or not) are run in an offline mode without the concern of runtime latency. So we choose an ensemble of detectors with large backbones and long iteration steps for linking algorithms. The quality of our temporal track data is recognized by a hybrid team of industrial and academic experts. Meanwhile, our dataset consists of vectorized maps as shown in Fig. 3. Note that this kind of map information is an important input for modern (interactive) trajectory prediction methods as it can serve as a prior that enforce the predicted trajectories to be constrained on lanes. Blue rectangles can be zoomed into a full resolution in the electronic version. These maps are generated by smashing existing HD maps for these areas to the ground plane. Our dataset provides lane boundaries that are critical to modern vectorized map protocols. To clarify, due to data security reasons, we are unable to release the whole map shown in Fig. 3 but will release local vectorized maps at intersections, which can be used for training and evaluation of state-of-the-art trajectory prediction algorithms.

3.2. Comparisons with related datasets

In Tab. 1, we compare INT2 with several published large-scale trajectory prediction datasets, highlighting several advantages over counterparts. INT2 contains a total of 106.8 million unique tracks, which is over 10 times larger than the de facto standard interactive trajectory forecasting dataset Waymo Open Motion. This large scale is also reflected in the total time and segment number metrics. And we note that all of these data are of the same quality as they are generated by the same offline perception system.

Another unique feature is that every segment in INT2 lasts for 60 seconds, which is longer than its counterparts by 2 or 3 times. This seems to be marginal but we would like to highlight that reaching 60 seconds means many segments see a whole traffic light cycle in our dataset, which is critical to the future study of interactions at intersections. This is also reflected by the average agent track length, which reaches 25.58 seconds and is much longer than Waymo Open Motion. Finally, for the trajectory forecasting horizon, we follow the practice of Waymo Open Motion and use 9.1 seconds as the default setting.

Our INT2 dataset involves three kinds of agents: vehicles, pedestrians, and cyclists. We also provide traffic light signals and state signals over time. And an advantage of INT2 over Waymo Open Motion is that we also provide the control relationship between traffic lights and lanes. We argue that the traffic light has seen limited usage in existing works mainly because of the lack of this relationship. We believe this additional information can bring research interests from the community. Finally, we define interactive scenarios as described in the Sec.4.

3.3. Dataset statistics

The duration of each scene in our dataset is approximately 1 minute, captured using 10 Hz sampling. Similar to the Waymo Open Motion Dataset, each state includes object bounding boxes (3D center point, heading, size, and

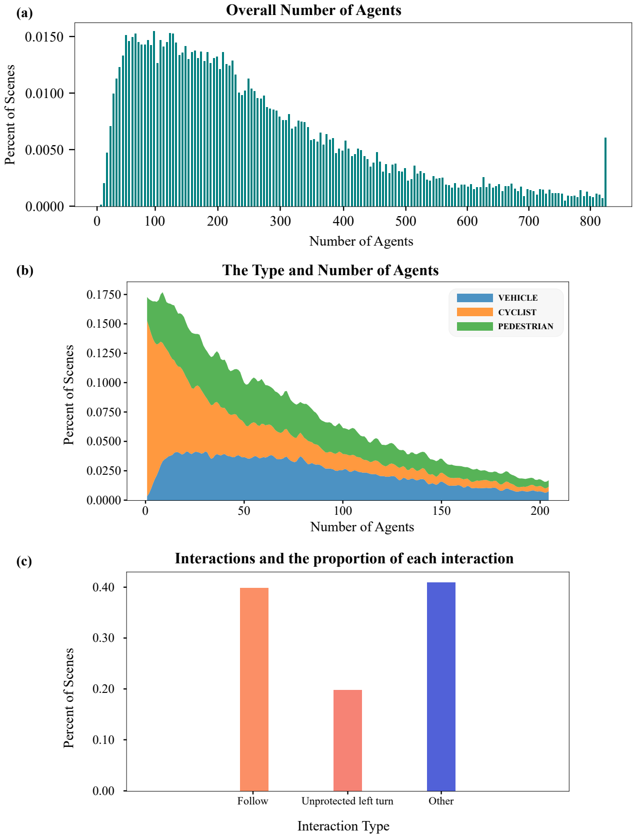


Figure 4. **Our dataset contains many agents including pedestrians, cyclists, and various interaction types. a**. 78% of the segments have more than 100 agents, and 33% segments have more than 300 agents. **b**. All segments have both vehicles and pedestrians, and 95% of the scenes have both vehicles and cyclists. **c**.The main types of interactions are included in the INT2.

velocity components in the x- and y-axis). Our dataset also provides precise map information, such as lane boundaries, lane center lines, crosswalks, junctions, and stop lines. Additionally, we offer intersection-specific traffic lights with real-time status information and corresponding controlled roads. We create 9.1 second scenes for validation of the latest research methods and the original more than 9.1 second segments are also provided for research requiring longer time frames.

Our dataset includes diverse interaction types in heavy traffic scenarios. Fig. 4a illustrates the distribution of segments with varying numbers of agents. The majority of segments contain more than 100 agents and approximately one-third of the scenes have more than 300 agents. In Fig. 4b, we present the proportion of different agent categories for different agent counts in all scenes. We can observe that the number of cyclists is generally distributed between 1 and 50, while the scenes with different numbers of vehicles and pedestrians have similar ratios.

Additionally, Fig. 4c displays the main interaction types and their respective proportions in our dataset. Notably, most of the intersections are left-turn protection intersections, so a large number of interactions occur between the front and rear vehicles. We filtered out numerous neighboring-fellowing and long-distance following the same path interactions while retaining a considerable amount of left-turn-through interactions. Overall, our dataset contained approximately 40% following interactions data (most of which were nearing following and long-distance following), around 20% left turn-straight interactions, and about 40% other interactions (such as u-turns, merge and overtaking, etc.).

To give readers a better understanding of our dataset, We report collision rates so that they can function as baselines for potential trajectory generation (instead of trajectory forecasting) applications. Generated trajectories should be as collision-free as possible. We show the collision rate of our dataset on agent-agent and agent-boundary, where the agents include vehicles, cyclists and pedestrians, as shown in Tab. 2

4. Definition of interaction

In this section, we outline our methodology for defining interactions. We propose an algorithm designed to efficiently extract interactions of research value from an extensive dataset. The algorithm, shown in Fig. 5, consists of three steps: Sec.4.1 filtering out non-interacting pairs based on a spatiotemporal distance threshold, Sec.4.2 normalizing and calculating direction, and Sec.4.3 removing non-compliant pairs and human inspection. Finally, we visualize the main types of interactions in the data at Sec.4.4.

4.1. Filter out non-interacting pairs based on spatiotemporal distance threshold.

To simplify the interactive relations of the complex scenes in our dataset, following [19, 39], we focus on pairwise interaction and define three relation types: pass, yield, and none. A pair of agents are considered interactive when their nearest distance is close enough, otherwise, it will be removed and the relation will be assigned as none. The first agent that arrives at the closest position is taken as influencer (pass) and the other one is reactor (yield).

Formally, for two agent trajectories y_1 and y_2, the closest distance d_I during T step is:

$$d_I = \min_{1\leq\tau_1\leq T} \min_{1\leq\tau_2\leq T} \|y_1^{\tau_1} - y_2^{\tau_2}\|_2 \tag{1}$$

Considering the dimension of the agents, we define the spatiotemporal distance threshold ϵ_d to be:

$$\epsilon_d = \frac{\sqrt{(l_1^2 + w_1^2)}}{2} + \frac{\sqrt{(l_2^2 + w_2^2)}}{2} + k_d, \tag{2}$$

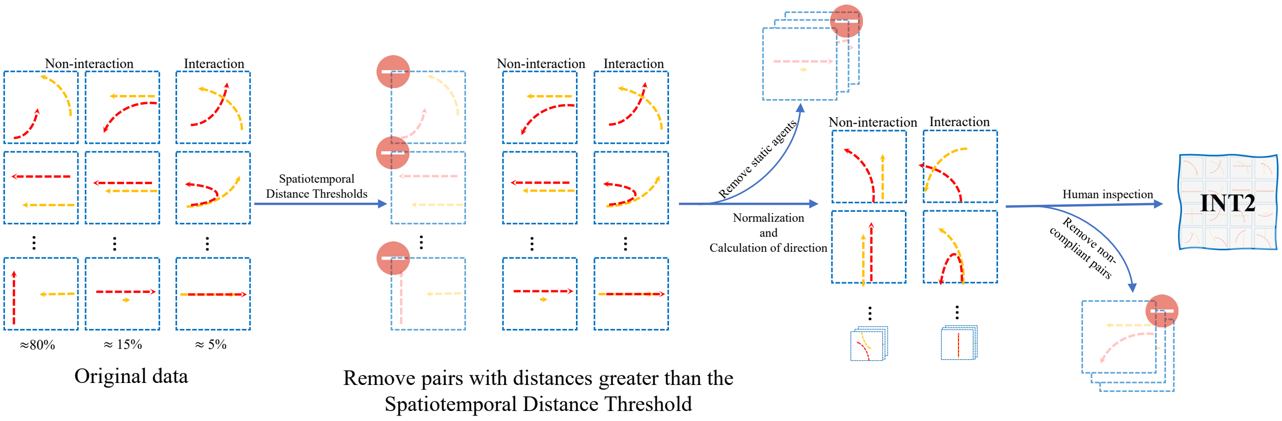

Non-interaction
Interaction
≈80%
≈ 15%
≈ 5%
Original data
Spatiotemporal Distance Thresholds
Non-interaction
Interaction
Remove pairs with distances greater than the Spatiotemporal Distance Threshold
Remove static agents
Normalization and Calculation of direction
Non-interaction
Interaction
Human inspection
INT2
Remove non-compliant pairs

Figure 5. **The illustration of interaction definement pipeline**. Initially, the pairs of trajectories that exhibit spatiotemporal distances above a predefined threshold are discarded, alongside the exclusion of static objects. The remaining trajectory pairs are subjected to normalization and direction calculation, and pairs that do not comply with the pre-established criteria are eliminated according to specific rules. Ultimately, the acquisition of the final data is conducted through a meticulous process of manual review.

where l_1, l_2, w_1, w_2 are the lengths and widths of the agents, $k_d > 0$ is the predefined maximum distance of interaction. Then the time steps when the agents arrive at the closest position can be calculated as:

$$\begin{aligned} t_1 &= \arg \min_{1\le\tau_1\le T} \min_{1\le\tau_2\le T} \|y_1^{\tau_1} - y_2^{\tau_2}\|_2 \\ t_2 &= \arg \min_{1\le\tau_2\le T} \min_{1\le\tau_1\le T} \|y_1^{\tau_1} - y_2^{\tau_2}\|_2 \end{aligned} \tag{3}$$

When $t_1 < t_2$, agent 1 is taken as influencer, agent 2 is defined as reactor, and vice versa.

Our method effectively filters out irrelevant object pairs from a vast amount of data by selecting pairs that have interacted within a time segment over 1 minute and meeting a spatiotemporal distance threshold of more than ϵ_d. We eliminate interaction pairs with lengths shorter than 91 frames to exclude pairs that may not exhibit meaningful interactions. Then, we identify an influencer and reactor among the remaining pairs by assuming that they engage in interactions. This crucial recognition process eliminates more than 80% non-interaction pairs.

4.2. Normalization and calculation of direction

In the previous step, we filtered out pairs that is far in distance. However, in reality, some agent pairs may be close in distance but not interact, such as the so-called interaction between a long bus and cars in its neighboring lane. To address this issue, we established rules to filter non-interacting pairs more strictly in three steps:

First, we remove static agents that may remain stationary for long periods due to traffic lights. We consider interactions such as overtaking and lane changing caused by these objects to have no actual interaction significance, as they are equivalent to obstacles or buildings on the road. Therefore, we remove all pairs of vehicles with a displacement less than the threshold λ_x within 91 frames.

Secondly, to better capture the properties and characteristics of trajectories, we judge the spatiotemporal relative positions of the trajectories of the influencer and reactor. We obtain the coordinates x' and y' of the influencer and the reactor by applying the transformation matrix A.

$$\begin{bmatrix} \mathrm{x}' \\ \mathrm{y}' \\ 1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}, \text{where} \tag{4}$$

$$A = \begin{bmatrix} cos\theta_0 & -sin\theta_0 & -x_0 cos\theta_0 + y_0 sin\theta_0 \\ sin\theta_0 & cos\theta_0 & -x_0 sin\theta_0 - y_0 cos\theta_0 \\ 0 & 0 & 1 \end{bmatrix}, \tag{5}$$

where x_0, y_0, and θ_0 respectively represent the initial coordinates and yaw of the influencer's frame, x and y represent the coordinates of the interaction pair at all time points in the time series. At this time, the trajectory of the influencer starts from the origin with an initial direction towards the positive x-axis, and the relative position between the trajectory of the reactor and the influencer remains unchanged. Therefore, the relative positional relationship between the influencer and the reactor can be easily obtained.

Finally, the trajectory characteristics, such as left-turn, right-turn, straight, straight and turn, u-turn, etc., are derived by computing the curvature of the path, displacements, and changes in trajectory speed and orientation. To filter out the behaviors that lack interaction significance, we rely on prescribed rules obtained from multiple experiments and interactive studies.

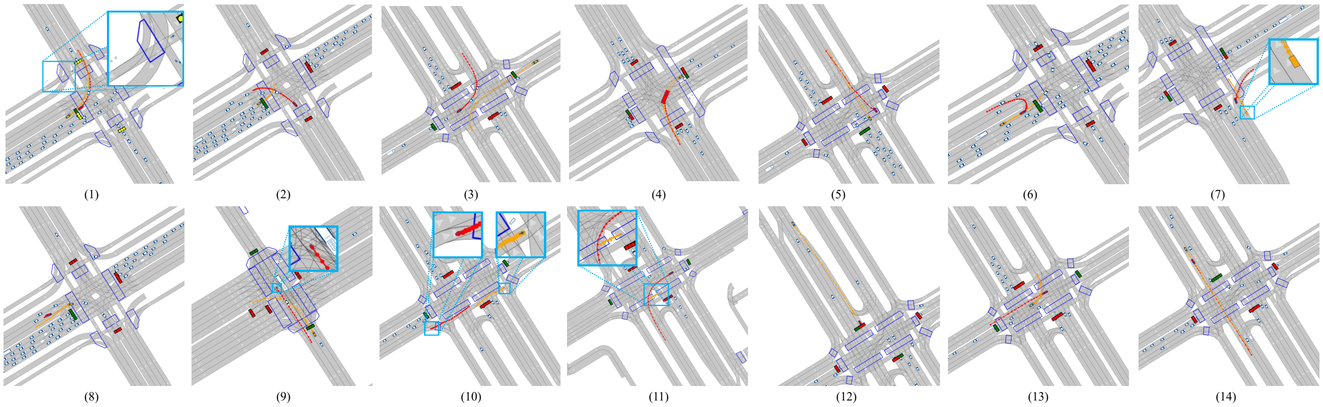


Figure 6. **Diverse interaction samples**. We present some of the main interactions that exist at intersections, including those between vehicles, cyclists, and pedestrians. In each panel, the red and yellow trajectories depict the future behavior of two agents (Vehicle, Cyclist, and Pedestrian). Panels (1) and (4) show the yellow car following the red car either in the same or a different lane while making an unprotected left-turn. In panel (2), the red car makes an unprotected left-turn while the yellow car makes a perpendicular left-turn. In panel (3), both the red car and the yellow car make left-turns in opposite directions. In panel (5), the red car merges into the lane where the yellow car goes straight. In panel (6), the red car makes a u-turn while the yellow car waits and then makes a left-turn. In panel (7), the red car makes a right-turn while the yellow motorcyclist waits and then goes straight. In panel (8), the yellow car overtakes the red car. Panel (9) shows the red motorcyclist going straight while the yellow car changes lanes and turns left. In panel (10), the red motorcyclist with a yellow cyclist and the red bus with a yellow car is driving toward each other. In panel (11), the red car turns left and the yellow pedestrian goes straight. In panel (12), the yellow car is forced to stop by the red car, and this interaction may be caused by the traffic light, so it is removed from our dataset. In panel (13), the red car goes straight while the yellow car makes a left turn in the opposite direction. In panel (14), the yellow car follows the red car on the same road straight.

4.3. Remove non-compliant pairs and Human inspection

Following the two filtering steps described above, we observe that almost all of the selected pairs showed subjective interactions. To mitigate potential inaccuracies resulting from non-compliance with the prescribed rules, we manually exclude less significant interactions. We also observe that a majority of the interactions involved the reactor following influencer on the same lane or neighbor lane. Therefore, we include only a subset of these interactions in our final dataset.

4.4. Interactions visualization

The presented samples in Fig.6 showcase various intersection scenarios. These samples display a range of interaction types involving influencers (red agents) and reactors (yellow agents) at intersections. Crosswalks (blue polygons) and pedestrians (white squares) are also shown, along with cyclists or motorcyclists (blue bike icons) as seen in the enlarged part of Fig.6(1).

5. Benchmarking

5.1. Dataset split

Domain mismatch. Our dataset presents the challenge of domain mismatch. We calculate the number of interactions between vehicles, cyclists, and pedestrians in each scene based on the interaction rules. Then, we designate the top 40% of the interaction counts in each segment as rush-hour data, which predominantly occurs from 7 to 9 am in the morning and 5 to 8 pm in the evening. The remaining data represents non-rush-hour instances.

Training and Validation set. We randomly select 28,000 segments, comprising 360,000 Vehicle-Vehicle, 100,000 Vehicle-Cyclist, and 100,000 Vehicle-Pedestrian interaction scenarios. We allocate 70% of these segments as the training set and 30% as the Validation set.

5.2. Metrics

Following [7], we evaluate and compare performances using minimum Average Displacement Error (minADE), minimum Final Displacement Error (minFDE), Miss Rate (MR), and mean Average Precision (mAP).

For N road agents, our model predicts T future waypoints $y_{n,t}$ whose groundtruth is $\bar{y}_{n,t}$ where $n \in \{1, ..., N\}$ and $t \in \{1, ..., T\}$.

minADE. Calculate L2 norm between $\bar{y}_{n,t}$ and $y_{n,t}$:

$$\text{minADE} = \frac{1}{T \times N} \min \sum_{n=1}^{N} \sum_{t=1}^{T} ||\bar{y}_{n,t} - y_{n,t}||_2 \qquad (6)$$

minFDE. Evaluating the minADE at the final time step T.

Table 3. **Trajectory prediction results on the INT2/Waymo Dataset**. The in-domain and cross-domain validation results are shown.

Interaction Type	Train Set	Valid Set	Marginal				Conditional			
			minADE ↓	mFDE ↓	MR ↓	mAP ↑	minADE ↓	mFDE ↓	MR ↓	mAP ↑
	Waymo	Waymo	1.70	3.45	0.23	0.3	N/A	5.49	0.55	0.18
Vehicle-Vehicle (V)	Rush-hour	Rush-hour	2.21	4.70	0.33	0.21	1.72	3.36	0.27	0.18
	Non-rush-hour	**Rush-hour**	**2.34**	**4.84**	**0.34**	**0.19**	**1.80**	**3.53**	**0.28**	**0.18**
	Non-rush-hour	Non-rush-hour	2.54	5.38	0.37	0.20	1.87	3.71	0.29	0.18
	Rush-hour	**Non-rush-hour**	**2.49**	**5.44**	**0.37**	**0.19**	**1.87**	**3.77**	**0.30**	**0.18**
Vehicle-Cyclist (V / C)	Rush-hour	Rush-hour	2.94 / 3.34	6.22 / 6.94	0.48 / 0.60	0.10 / 0.08	2.85 / 2.95	5.92 / 5.88	0.51 / 0.56	0.05 / 0.06
	Non-rush-hour	**Rush-hour**	**2.92 / 3.30**	**6.19 / 6.83**	**0.48 / 0.59**	**0.11 / 0.07**	**2.88 / 3.19**	**6.04 / 6.52**	**0.52 / 0.60**	**0.06 / 0.05**
	Non-rush-hour	Non-rush-hour	3.09 / 3.51	6.65 / 7.47	0.47 / 0.58	0.13 / 0.07	3.02 / 3.28	6.20 / 6.82	0.49 / 0.59	0.05 / 0.04
	Rush-hour	**Non-rush-hour**	**3.22 / 3.75**	**6.94 / 8.01**	**0.50 / 0.62**	**0.10 / 0.06**	**3.10 / 3.18**	**6.35 / 6.50**	**0.50 / 0.56**	**0.06 / 0.04**
Vehicle-Pedestrian (V / P)	Rush-hour	Rush-hour	4.30 / 1.22	10.12 / 2.05	0.68 / 0.22	0.10 / 0.35	2.64 / 1.53	5.27 / 2.87	0.47 / 0.34	0.11 / 0.13
	Non-rush-hour	**Rush-hour**	**4.07 / 1.30**	**9.60 / 2.30**	**0.68 / 0.27**	**0.05 / 0.24**	**2.90 / 1.63**	**5.85 / 3.15**	**0.51 / 0.37**	**0.10 / 0.11**
	Non-rush-hour	Non-rush-hour	4.30 / 1.28	10.09 / 2.23	0.69 / 0.26	0.08 / 0.27	2.77 / 1.70	5.41 / 3.34	0.47 / 0.37	0.09 / 0.12
	Rush-hour	**Non-rush-hour**	**5.22 / 1.44**	**12.33 / 2.56**	**0.78 / 0.32**	**0.05 / 0.17**	**2.77 / 1.58**	**5.50 / 3.04**	**0.47 / 0.35**	**0.09 / 0.12**

$$\text{minFDE} = \frac{1}{N} \min \sum_{n=1}^{N} ||\bar{y}_{n,T} - y_{n,T}||_2 \tag{7}$$

Miss Rate (MR). A predicted waypoint is a miss or match to a target waypoint. It is a match if the differences in x and y coordinates between prediction and target waypoints are both smaller than the thresholds λ_x and λ_y.

$$\bar{y}_{n,t} - y_{n,t} = (\Delta x_{n,t}, \Delta y_{n,t}) \tag{8}$$

$$M(\bar{y}_{n,t}, y_{n,t}) = (|\Delta x_{n,t}| < \lambda_x) \wedge (|\Delta y_{n,t}| < \lambda_y) \tag{9}$$

where λ_x and λ_y are dynamic thresholds that depend on the velocities and time:

$$(\lambda_x, \lambda_y) = (\lambda_t^x \phi(v_x), \lambda_t^y \phi(v_y)) \tag{10}$$

where v_x and v_y are the velocity in x and y directions respectively; and

$$\phi(v) = \frac{1 + \max(0, \min(1, \frac{v-l}{L-l}))}{2} \tag{11}$$

where L and l are equals 11 m/s and 1.4 m/s, respectively.

For all of the K pairs of predicted and groundtruth waypoints:

$$\text{MR} = 1 - \frac{\sum_{k=1}^{K} \mathbb{1}[M(\bar{y}_k, y_k)]}{K} \tag{12}$$

Mean average precision (mAP). Mean Average Precision (mAP) is the mean value of Average Precisions (APs):

$$\text{mAP} = \overline{AP}_i \tag{13}$$

where $i \in$ {forward, left, right, turn left, turn right, left u-turn, right u-turn, no movement}. The AP calculates the area beneath the precision-recall curve by employing confidence score thresholds e_k and utilizes MR to distinguish between true positives and false positives. At most one true positive can be assigned to each groundtruth. After assigning the prediction with the highest confidence to one ground truth, other predictions for that groundtruth are all considered false positives.

5.3. Baseline method

We use the state-of-the-art method M2I [39] as the baseline method. M2I consists of three modules: a relation prediction module that predicts whether an agent is an influencer of a reactor (who yields to influencer), a marginal trajectory predictor that predicts the future trajectories independently without considering the potential interactions among agents, and a conditional trajectory predictor that takes both relation and marginal trajectory into consideration and predicts the trajectory for reactors.

We train the M2I models under 6 different settings: vehicle-vehicle rush-hour, vehicle-vehicle non-rush-hour, vehicle-cyclist rush-hour, vehicle-cyclist non-rush-hour, vehicle-pedestrian rush-hour and vehicle-pedestrian non-rush-hour. In each setting, we need to train three components proposed in [39]: relation predictor, marginal trajectory predictor, and conditional trajectory predictor.

For each of the components, we train 30 epochs with an initial learning rate equaling 0.001. We use the Adam optimizer with a weight decay of 0.3. The learning rate is dropped every 5 epochs. Our models are trained to predict future 80 frames (8 s) using 11 past frames (1.1 s).

During inference, we first use the relation predictor to predict the relations between agent pairs. Then, we produce the predicted marginal trajectories by marginal predictor. Finally, we predict the conditional trajectories using relations and marginal trajectories.

5.4. Experiments

Quantitative Results. Trajectory prediction results are shown in Tab. 3. For different interaction types, we show the validation results for each kind of agents separately. The in-domain and cross-domain validation results on Waymo Open Motion and our split dataset are compared.

Firstly, the M2I model performs worse on our dataset than on Waymo Open Motion, which demonstrates the challenges brought by the large scale and diverse interaction of INT2. Secondly, cross-domain validation on INT2 leads to

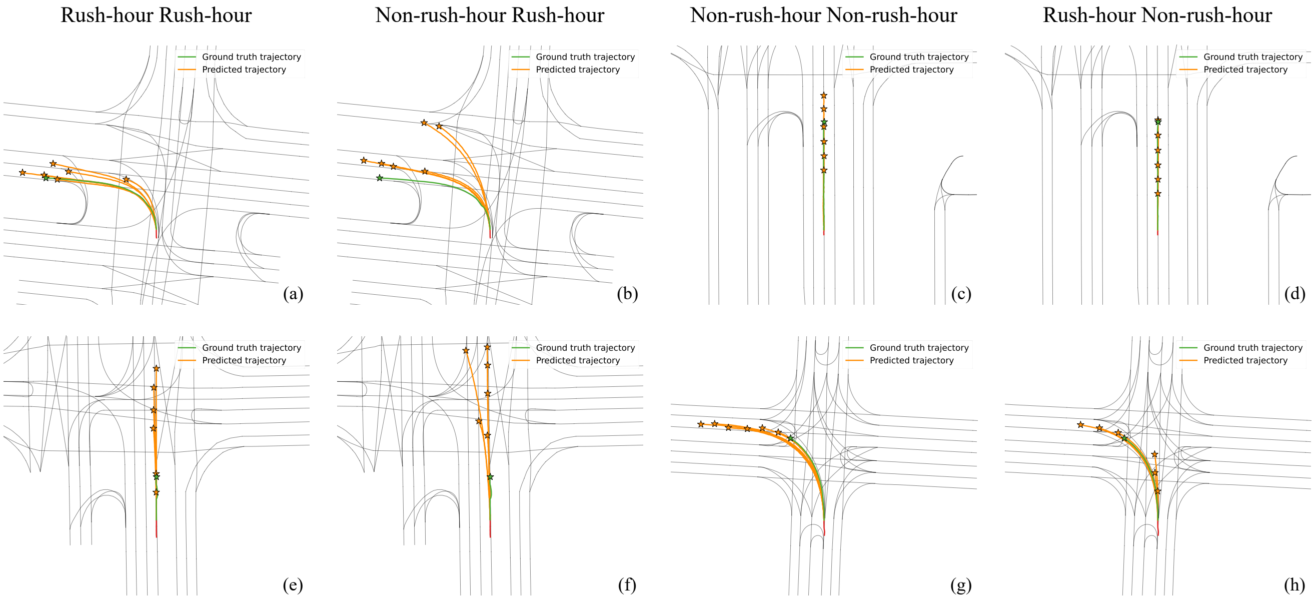


Figure 7. Qualitative results of M2I on our INT2 dataset.

significantly different results. For ease of comparison, the cross-domain validation results are bolded. In the third and fourth rows of Vehicle-Pedestrian, it can be observed that the cross-domain validation results are significantly lower than the in-domain validation results in marginal prediction, though the scenes are the same. But in conditional prediction, the validation results of cross-domain pedestrians are obviously better than the in-domain results. We believe that humans respond differently (more conservatively or aggressively) to interactions during rush-hour periods and non-rush-hour periods when they act in different roles (Vehicle, Cyclist, or Pedestrian) at intersections. How to effectively leverage the different domain features to alleviate the domain mismatch problem while obtaining accurate trajectory prediction results will be an interesting line of research.

Qualitative Results. In Fig. 7, we visualize the prediction results of M2I on the Vehicle-Vehicle interactions of our dataset. From left to right, we show the results of the model training and evaluate on in-domain or cross-domain scenarios (“Non-rush-hour Rush-hour”means the model is trained on non-rush-hour scenarios and validated on rush-hour scenarios). The results are consistent with our quantitative results.

Based on Fig. 7a, e, c, and g, when the training and validation data belong to the same domain, the predicted trajectories align with the ground truth in terms of direction. However, the predicted trajectories of the model trained on non-rush-hour scenarios tend to be longer and more divergent compared to those of the model trained on rush-hour scenarios. We attribute this to the more aggressive driving behavior of humans on non-rush-hour scenes.

From Fig. 7b, f, d, and h, it can be observed that when the training and validation data come from cross-domain, there is a notable deviation between the predicted results and the ground truth. Overall, a consistent pattern emerges: models trained on non-rush-hour scenarios tend to exhibit more aggressive behavior, as demonstrated by Fig. 7b with a greater number of paths and Fig. 7f with higher speeds and more direction choices. Conversely, models trained on rush-hour datasets tend to adopt a more cautious approach, as illustrated by Fig. 7d with slower predicted speeds, and in Fig. 7h, there are partial instances of conservative straight-line trajectories.

6. Conclusion

This paper presents a new interactive trajectory prediction dataset named **INT2**, which is short for **INT**eractive trajectory prediction at **INT**ersections. INT2 has three notable features: high quality, large scale and rich information. The high-quality 3D agent box trajectories are credited to a multi-sensor setup and an offline detection and fusion algorithm stack. The scale of INT2 is not only much larger than Waymo Open Motion but also longer in each segment. INT2 also contains rich information including various agents, vectorized maps and traffic light signals. Capturing data at intersections allows us to bypass the impact of the map and clearly define two domains: rush-hour and non-rush-hour. We systematically evaluate several cross-domain settings using the state-of-the-art interactive trajectory prediction method, pointing to interesting observations.

References

[1] Alexandre Alahi, Kratarth Goel, Vignesh Ramanathan, Alexandre Robicquet, Li Fei-Fei, and Silvio Savarese. Social lstm: Human trajectory prediction in crowded spaces. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 961–971, 2016.

[2] Marcus A Brubaker, Andreas Geiger, and Raquel Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3057–3064, 2013.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[4] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, et al. Argoverse: 3d tracking and forecasting with rich maps. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8748–8757, 2019.

[5] Guangyi Chen, Junlong Li, Jiwen Lu, and Jie Zhou. Human trajectory prediction via counterfactual analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9824–9833, 2021.

[6] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[7] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R Qi, Yin Zhou, et al. Large scale interactive motion forecasting for autonomous driving: The waymo open motion dataset. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9710–9719, 2021.

[8] Whye Kit Fong, Rohit Mohan, Juana Valeria Hurtado, Lubing Zhou, Holger Caesar, Oscar Beijbom, and Abhinav Valada. Panoptic nuscenes: A large-scale benchmark for lidar panoptic segmentation and tracking. *IEEE Robotics and Automation Letters*, 7(2):3795–3802, 2022.

[9] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533, 2020.

[10] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[11] Thomas Gilles, Stefano Sabatini, Dzmitry Tsishkou, Bogdan Stanciulescu, and Fabien Moutarde. Gohome: Graph-oriented heatmap output for future motion estimation. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 9107–9114. IEEE, 2022.

[12] Junru Gu, Chen Sun, and Hang Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312, 2021.

[13] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social gan: Socially acceptable trajectories with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2255–2264, 2018.

[14] John Houston, Guido Zuidhof, Luca Bergamini, Yawei Ye, Long Chen, Ashesh Jain, Sammy Omari, Vladimir Iglovikov, and Peter Ondruska. One thousand and one hours: Self-driving motion prediction dataset. In *Conference on Robot Learning*, pages 409–418. PMLR, 2021.

[15] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1067–10676, 2018.

[16] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(10):2702–2719, 2020.

[17] Bu Jin, Xinyu Liu, Yupeng Zheng, Pengfei Li, Hao Zhao, Tong Zhang, Yuhang Zheng, Guyue Zhou, and Jingjing Liu. Adapt: Action-aware driving caption transformer. *arXiv preprint arXiv:2302.00673*, 2023.

[18] R Kesten, M Usman, J Houston, T Pandya, K Nadhamuni, A Ferreira, M Yuan, B Low, A Jain, P Ondruska, et al. Lyft level 5 perception dataset 2020, 2019.

[19] Sumit Kumar, Yiming Gu, Jerrick Hoang, Galen Clark Haynes, and Micol Marchetti-Bowick. Interaction-based trajectory prediction over a hybrid traffic graph. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5530–5535. IEEE, 2021.

[20] Julius Kümmerle, Hendrik Königshof, Christoph Stiller, Arnaud de La Fortelle, and Masayoshi Tomizuka. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps.

[21] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[22] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 336–345, 2017.

[23] Alon Lerner, Yiorgos Chrysanthou, and Dani Lischinski. Crowds by example. In *Computer graphics forum*, volume 26, pages 655–664. Wiley Online Library, 2007.

[24] Ding Li, Qichao Zhang, Zhongpu Xia, Kuan Zhang, Menglong Yi, Wenda Jin, and Dongbin Zhao. Planning-inspired

hierarchical trajectory prediction for autonomous driving. *arXiv preprint arXiv:2304.11295*, 2023.

[25] Pengfei Li, Ruowen Zhao, Yongliang Shi, Hao Zhao, Jirui Yuan, Guyue Zhou, and Ya-Qin Zhang. Lode: Locally conditioned eikonal implicit scene completion from sparse lidar. *arXiv preprint arXiv:2302.14052*, 2023.

[26] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, 2020.

[27] Yicheng Liu, Jinghuai Zhang, Liangji Fang, Qinhong Jiang, and Bolei Zhou. Multimodal motion prediction with stacked transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7577–7586, 2021.

[28] Karttikeya Mangalam, Yang An, Harshayu Girase, and Jitendra Malik. From goals, waypoints & paths to long term human trajectory forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15233–15242, 2021.

[29] Jieru Mei, Alex Zihao Zhu, Xinchen Yan, Hang Yan, Siyuan Qiao, Liang-Chieh Chen, and Henrik Kretzschmar. Waymo open dataset: Panoramic video panoptic segmentation. In *European Conference on Computer Vision*, pages 53–72. Springer, 2022.

[30] Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14424–14432, 2020.

[31] Jiquan Ngiam, Benjamin Caine, Vijay Vasudevan, Zhengdong Zhang, Hao-Tien Lewis Chiang, Jeffrey Ling, Rebecca Roelofs, Alex Bewley, Chenxi Liu, Ashish Venugopal, et al. Scene transformer: A unified multi-task model for behavior prediction and planning. *arXiv e-prints*, pages arXiv–2106, 2021.

[32] Stefano Pellegrini, Andreas Ess, Konrad Schindler, and Luc Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *2009 IEEE 12th international conference on computer vision*, pages 261–268. IEEE, 2009.

[33] Tim Salzmann, Boris Ivanovic, Punarjay Chakravarty, and Marco Pavone. Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data. In *European Conference on Computer Vision*, pages 683–700. Springer, 2020.

[34] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5135–5142. IEEE, 2020.

[35] Shaoshuai Shi, Li Jiang, Dengxin Dai, and Bernt Schiele. Motion transformer with global intention localization and local movement refinement. *arXiv preprint arXiv:2209.13508*, 2022.

[36] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 770–779, 2019.

[37] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE transactions on pattern analysis and machine intelligence*, 43(8):2647–2664, 2020.

[38] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[39] Qiao Sun, Xin Huang, Junru Gu, Brian C Williams, and Hang Zhao. M2i: From factored marginal trajectory prediction to interactive prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6543–6552, 2022.

[40] Shenlong Wang, Min Bai, Gellert Mattyus, Hang Chu, Wenjie Luo, Bin Yang, Justin Liang, Joel Cheverie, Sanja Fidler, and Raquel Urtasun. Torontocity: Seeing the world with a million eyes. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 3028–3036, 2017.

[41] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021.

[42] Zirui Wu, Tianyu Liu, Liyi Luo, Zhide Zhong, Jianteng Chen, Hongmin Xiao, Chao Hou, Haozhe Lou, Yuantao Chen, Runyi Yang, Yuxin Huang, Xiaoyu Ye, Zike Yan, Yongliang Shi, Yiyi Liao, and Hao Zhao. Mars: An instance-aware, modular and realistic simulator for autonomous driving. *CICAI*, 2023.

[43] Dan Xie, Sinisa Todorovic, and Song-Chun Zhu. Inferring" dark matter" and" dark energy" from videos. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2224–2231, 2013.

[44] Xu Yan, Jiantao Gao, Jie Li, Ruimao Zhang, Zhen Li, Rui Huang, and Shuguang Cui. Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3101–3109, 2021.

[45] Haibao Yu, Yizhen Luo, Mao Shu, Yiyi Huo, Zebang Yang, Yifeng Shi, Zhenglong Guo, Hanyu Li, Xing Hu, Jirui Yuan, et al. Dair-v2x: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21361–21370, 2022.

[46] Ye Yuan, Xinshuo Weng, Yanglan Ou, and Kris M Kitani. Agentformer: Agent-aware transformers for socio-temporal multi-agent forecasting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9813–9823, 2021.

[47] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An

improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2020.

[48] Hang Zhao, Jiyang Gao, Tian Lan, Chen Sun, Ben Sapp, Balakrishnan Varadarajan, Yue Shen, Yi Shen, Yuning Chai, Cordelia Schmid, et al. Tnt: Target-driven trajectory prediction. In *Conference on Robot Learning*, pages 895–904. PMLR, 2021.

[49] Yupeng Zheng, Chengliang Zhong, Pengfei Li, Huan-ang Gao, Yuhang Zheng, Bu Jin, Ling Wang, Hao Zhao, Guyue Zhou, Qichao Zhang, et al. Steps: Joint self-supervised nighttime image enhancement and depth estimation. *arXiv preprint arXiv:2302.01334*, 2023.

[50] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4490–4499, 2018.