INT2: Interactive Trajectory Prediction at Intersections

Zhijie Yan1,3*, Pengfei Li1, Zheng Fu1,4, Shaocong Xu1,5, Yongliang Shi1, Xiaoxue Chen1, Yuhang Zheng1,3, Yang Li1, Tianyu Liu1,6, Chuxuan Li1, Nairui Luo2, Xu Gao2, Yilun Chen1, Zuoxu Wang3, Yifeng Shi2, Pengfei Huang1, Zhengxiao Han1,7, Jirui Yuan1, Jiangtao Gong1, Guyue Zhou1, Hang Zhao8, Hao Zhao†

1AIR, THU, 2Baidu Inc., 3SMEA, BUAA
4SVM, THU, 5XMU, 6ECE, HKUST, 7BUCT, 8IIIS, 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 INteractive trajectory prediction at INTERsections. 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 [16, 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 downstream 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 downstream 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 INTeractive trajectory prediction at INTERsections. 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, lane mark 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
datasets\cite{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, \cite{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\cite{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 \cite{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\cite{13, 30} is adopted to model motion histories and predicts socially plausible future behavior. On top of this, DESIRE\cite{22} accounts for scene context as well as the interactions among the agents. \cite{28} focuses on long-term trajectory forecasting and proposes a scene-compliant trajectory prediction network to resolve it. \cite{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, \cite{5, 9, 26, 33} predict the trajectories of multiple agents by leveraging graph-structured models. TNT\cite{48} and DenseTNT\cite{12} generate state sequences conditioned on targets for trajectory prediction of vehicles and pedestrians. Due to its attention mechanism, Transformer-based multi-agent models\cite{27, 31, 35, 46} can naturally model the interaction between different agents. Specifically, Scene Transformer\cite{31} provides a multi-agent model for predicting the behavior of all agents. AgentFormer\cite{46} leverages a novel agent-aware attention mechanism to simultaneously model the time and social dimensions. Motion transformer\cite{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\cite{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.

### 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. 2a. 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 \times 1080 resolution, and a JPEG compression coding scheme. The LiDARs are 300-line LiDAR with a 10Hz sampling frequency, 100$^\circ$ horizontal FOV, and -30$^\circ$ to 10$^\circ$ vertical FOV, with a visual range of...
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. 2.

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 enforces 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 been 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
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. 4 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. 4, 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. 4 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-following 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:

\[
  d_I = \min_{1 \leq t_1 \leq T} \min_{1 \leq t_2 \leq T} \| y_1(t_1) - y_2(t_2) \|_2
\]

We consider the dimension of the agents, we define the spatiotemporal distance threshold \( \epsilon_d \) to be:

\[
  \epsilon_d = \sqrt{\frac{l_1^2 + w_1^2}{2}} + \sqrt{\frac{l_2^2 + w_2^2}{2}} + k_d,
\]
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:

\[
\begin{align*}
    t_1 &= \arg \min_{1 \leq \tau_1 \leq T} \min_{1 \leq \tau_2 \leq T} \| y^1_{\tau_1} - y^2_{\tau_2} \|_2 \\
    t_2 &= \arg \min_{1 \leq \tau_1 \leq T} \min_{1 \leq \tau_2 \leq T} \| y^1_{\tau_1} - y^2_{\tau_2} \|_2
\end{align*}
\] (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 \( \epsilon_d \). 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 \( \lambda_r \) 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 \( A \).

\[
\begin{bmatrix}
    x' \\
    y' \\
    1
\end{bmatrix} = A \begin{bmatrix}
    x \\
    y \\
    1
\end{bmatrix}, \text{ where}
\] (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 \( \theta_0 \) 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 $\hat{y}_{n,t}$ 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 $\hat{y}_{n,t}$:

$$\text{minADE} = \frac{1}{T \times N} \min_{n=1}^{N} \sum_{t=1}^{T} ||\bar{y}_{n,t} - \hat{y}_{n,t}||_2$$

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

8542
Table 3. Trajectory prediction results on the INT2/Waymo Dataset. The in-domain and cross-domain validation results are shown.

<table>
<thead>
<tr>
<th>Interaction Type</th>
<th>Train Set</th>
<th>Valid Set</th>
<th>Marginal</th>
<th>Conditional</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>minADE</td>
<td>mFDE</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>MR</td>
<td>mAP</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>mAP</td>
<td>MR</td>
</tr>
<tr>
<td>Waymo</td>
<td>Rush-hour</td>
<td>Rush-hour</td>
<td>1.70</td>
<td>3.45</td>
</tr>
<tr>
<td></td>
<td>Non-rush-hour</td>
<td>Rush-hour</td>
<td>2.23</td>
<td>4.70</td>
</tr>
<tr>
<td></td>
<td>Non-rush-hour</td>
<td>Non-rush-hour</td>
<td>2.54</td>
<td>5.38</td>
</tr>
<tr>
<td></td>
<td>Rush-hour</td>
<td>Non-rush-hour</td>
<td>2.49</td>
<td>5.44</td>
</tr>
<tr>
<td></td>
<td>Rush-hour</td>
<td>Rush-hour</td>
<td>2.94</td>
<td>3.34</td>
</tr>
<tr>
<td></td>
<td>Non-rush-hour</td>
<td>Rush-hour</td>
<td>2.92</td>
<td>3.30</td>
</tr>
<tr>
<td></td>
<td>Non-rush-hour</td>
<td>Non-rush-hour</td>
<td>3.09</td>
<td>3.51</td>
</tr>
<tr>
<td></td>
<td>Rush-hour</td>
<td>Non-rush-hour</td>
<td>3.22</td>
<td>3.75</td>
</tr>
<tr>
<td></td>
<td>Rush-hour</td>
<td>Rush-hour</td>
<td>4.30</td>
<td>1.22</td>
</tr>
<tr>
<td></td>
<td>Non-rush-hour</td>
<td>Rush-hour</td>
<td>4.07</td>
<td>1.30</td>
</tr>
<tr>
<td></td>
<td>Non-rush-hour</td>
<td>Non-rush-hour</td>
<td>4.30</td>
<td>1.28</td>
</tr>
<tr>
<td></td>
<td>Rush-hour</td>
<td>Non-rush-hour</td>
<td>5.22</td>
<td>1.44</td>
</tr>
<tr>
<td>Non-rush-hour</td>
<td>Rush-hour</td>
<td>Rush-hour</td>
<td>2.49</td>
<td>3.45</td>
</tr>
<tr>
<td></td>
<td>Non-rush-hour</td>
<td>Rush-hour</td>
<td>2.34</td>
<td>4.84</td>
</tr>
<tr>
<td></td>
<td>Non-rush-hour</td>
<td>Non-rush-hour</td>
<td>2.54</td>
<td>5.38</td>
</tr>
<tr>
<td></td>
<td>Rush-hour</td>
<td>Non-rush-hour</td>
<td>2.49</td>
<td>5.44</td>
</tr>
<tr>
<td></td>
<td>Rush-hour</td>
<td>Rush-hour</td>
<td>2.94</td>
<td>3.34</td>
</tr>
<tr>
<td></td>
<td>Non-rush-hour</td>
<td>Rush-hour</td>
<td>2.92</td>
<td>3.30</td>
</tr>
<tr>
<td></td>
<td>Non-rush-hour</td>
<td>Non-rush-hour</td>
<td>3.09</td>
<td>3.51</td>
</tr>
<tr>
<td></td>
<td>Rush-hour</td>
<td>Non-rush-hour</td>
<td>3.22</td>
<td>3.75</td>
</tr>
<tr>
<td></td>
<td>Rush-hour</td>
<td>Rush-hour</td>
<td>4.30</td>
<td>1.22</td>
</tr>
<tr>
<td></td>
<td>Non-rush-hour</td>
<td>Rush-hour</td>
<td>4.07</td>
<td>1.30</td>
</tr>
<tr>
<td></td>
<td>Non-rush-hour</td>
<td>Non-rush-hour</td>
<td>4.30</td>
<td>1.28</td>
</tr>
<tr>
<td></td>
<td>Rush-hour</td>
<td>Non-rush-hour</td>
<td>5.22</td>
<td>1.44</td>
</tr>
</tbody>
</table>

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
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. [7], 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. [7], 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. [7] with a greater number of paths and Fig. [7] 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. [7] with slower predicted speeds, and in Fig. [7], 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 INTERactive trajectory prediction at INTERsections. 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


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


