End-to-End Interactive Prediction and Planning with Optical Flow Distillation for Autonomous Driving

Hengli Wang[1], Peide Cai[1], Rui Fan[2], Yuxiang Sun[3], and Ming Liu[1]
[1] The Hong Kong University of Science and Technology
[2] University of California San Diego
[3] The Hong Kong Polytechnic University
{hwangdf, pcaiaa}@connect.ust.hk, rfan@ucsd.edu, sun.yuxiang@outlook.com, eelium@ust.hk

Abstract

With the recent advancement of deep learning technology, data-driven approaches for autonomous car prediction and planning have achieved extraordinary performance. Nevertheless, most of these approaches follow a non-interactive prediction and planning paradigm, hypothesizing that a vehicle's behaviors do not affect others. The approaches based on such a non-interactive philosophy typically perform acceptably in sparse traffic scenarios but can easily fail in dense traffic scenarios. Therefore, we propose an end-to-end interactive neural motion planner (INMP) for autonomous driving in this paper. Given a set of past surrounding-view images and a high definition map, our INMP first generates a feature map in bird's-eye-view space, which is then processed to detect other agents and perform interactive prediction and planning jointly. Also, we adopt an optical flow distillation paradigm, which can effectively improve the network performance while still maintaining its real-time inference speed. Extensive experiments on the nuScenes dataset and in the closed-loop Carla simulation environment demonstrate the effectiveness and efficiency of our INMP for the detection, prediction, and planning tasks. Our project page is at sites.google.com/view/inmp-ofd.

1. Introduction

Autonomous driving aims at safely and efficiently maneuvering self-driving vehicles (SDVs) from a starting point to a target point with the input of sensor data and a pre-built map [15, 7, 4, 26]. Most existing approaches are designed to follow a "perception-prediction-planning" paradigm, as shown in Fig. 1 (a), where the perception module detects other agents from the sensor data, the prediction module estimates possible future trajectories of the detected agents, and the planning module generates a safe trajectory to drive the SDV towards the given target location based on the output from the perception and prediction modules [13]. The approaches designed under this paradigm are **non-interactive**, as the planning module is assumed to have no effects on the results of the prediction module. This implies that each SDV is a passive agent, and its behavior does not affect the other agents. Based on this philosophy, these non-interactive approaches typically perform acceptably in sparse traffic scenarios but can easily fail in dense traffic scenarios [25]. For instance, Fig. 1 (b) illustrates a dense traffic scenario where the non-interactive SDV tries to merge into the left lane. Since the estimated future trajectories of other agents can cover most of the road in such a dense scenario, it can be challenging for the SDV to plan

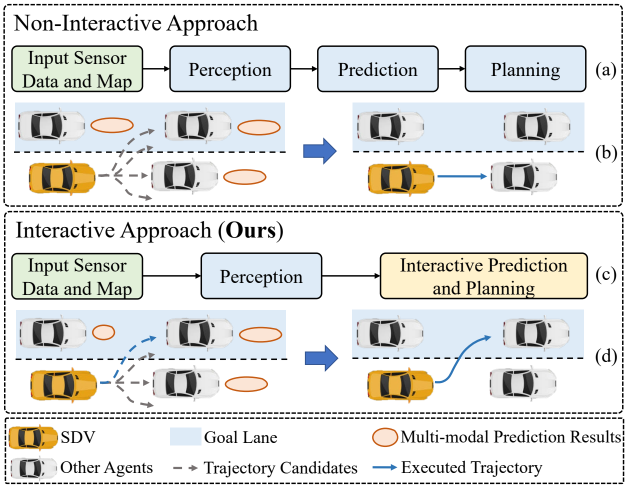
Non-Interactive Approach
Input Sensor Data and Map
Perception
Prediction
Planning
(a)
(b)
Interactive Approach (Ours)
Input Sensor Data and Map
Perception
Interactive Prediction and Planning
(c)
(d)
SDV
Goal Lane
Multi-modal Prediction Results
Other Agents
Trajectory Candidates
Executed Trajectory

Figure 1. An illustration of non-interactive and our proposed interactive approaches, where (a) and (c) show the corresponding frameworks; and (b) and (d) show the corresponding driving performance in a dense traffic scenario. Specifically, the non-interactive SDV can struggle merging into the left lane, while our interactive SDV can perform a satisfactory lane merge by reasoning about how other agents will react to its behaviors.

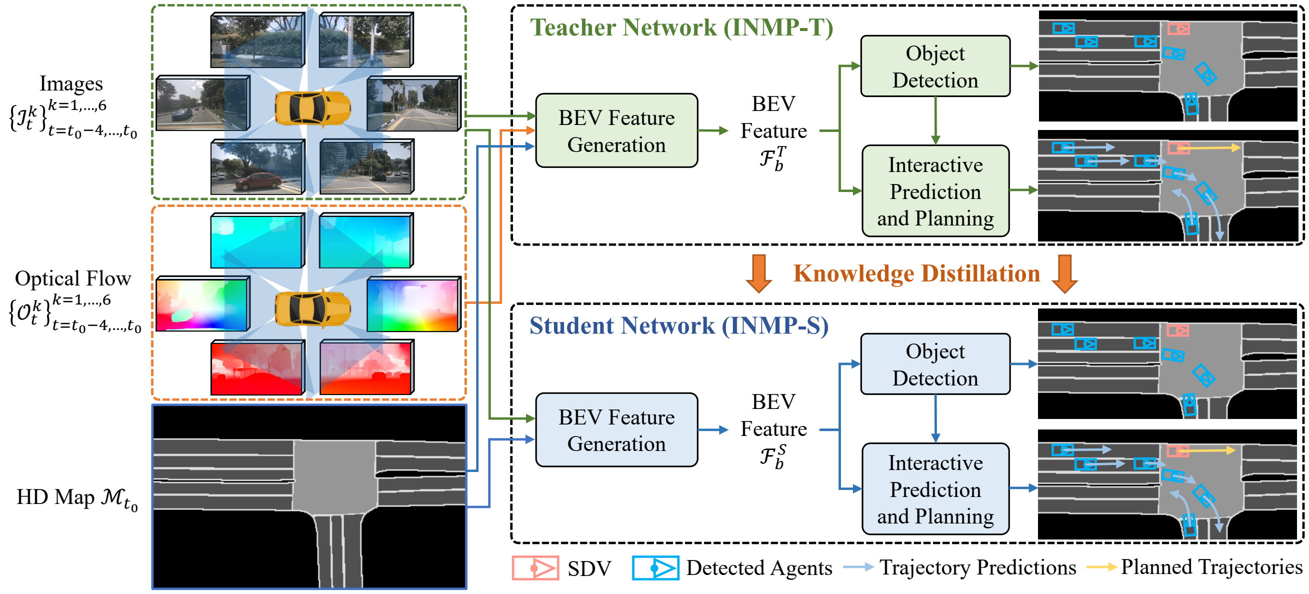
Images $\{\mathcal{I}_t^k\}_{t=t_0-4,\ldots,t_0}^{k=1,\ldots,6}$
Optical Flow $\{\mathcal{O}_t^k\}_{t=t_0-4,\ldots,t_0}^{k=1,\ldots,6}$
HD Map $\mathcal{M}_{t_0}$
Teacher Network (INMP-T)
BEV Feature Generation
BEV Feature $\mathcal{F}_b^T$
Object Detection
Interactive Prediction and Planning
Knowledge Distillation
Student Network (INMP-S)
BEV Feature Generation
BEV Feature $\mathcal{F}_b^S$
Object Detection
Interactive Prediction and Planning
SDV
Detected Agents
Trajectory Predictions
Planned Trajectories

Figure 2. An overview of our INMP, which takes a set of past surrounding-view images and an HD map as input to jointly 1) detect other agents and 2) perform interactive prediction and planning. We also adopt an optical flow distillation paradigm, where the teacher network adopts a similar architecture to the student network but takes optical flow as an additional type of input. We then distill the knowledge from the teacher network to the student network, which can effectively improve the student network performance while still maintaining its real-time inference speed.

a feasible trajectory to the left lane, resulting in a long wait for the SDV and causing inconvenience to the other agents behind it.

To address this problem, the SDV needs to be modeled as an active agent, capable of reasoning about how other agents will react to its behaviors. In this way, the prediction module becomes correlated to the planning module, and they can be formulated as a single **interactive** prediction and planning module, as shown in Fig. 1 (c). An iterative SDV is now capable of considering the possible reactions of other agents when safely merging into the left lane in the same dense traffic scenario, as shown in Fig. 1 (d). Researchers have already developed some interactive prediction and planning approaches from different perspectives, such as game-theoretic planning [14] and reinforcement learning [35]. However, these approaches mainly depend on manually designed models or rewards, which may not be able to accurately model real-world agent dynamics or human-like driving behaviors. Therefore, there is a strong motivation to develop a general interactive prediction and planning approach for autonomous driving.

In this paper, we propose an end-to-end **I**nteractive **N**eural **M**otion **P**lanner (INMP), for autonomous driving. Our INMP, illustrated in Fig. 2, takes a set of past surrounding-view images and a high definition (HD) map as input to jointly 1) detect other agents and 2) perform interactive prediction and planning for the SDV. Specifically, we first lift these images into three dimensions (3-D), and then combine them with the HD map to generate a feature map in bird's-eye-view (BEV) space. This BEV feature map $\mathcal{F}_b$ is then processed to 1) detect other agents via a single-shot detection header and 2) jointly estimate the future trajectories of the detected agents and produce safe motion plans for the SDV via an interactive prediction and planning model with a joint probability distribution and a set of learnable costs. This paradigm enables the SDV to reason about how other agents will react to its behaviors, and thus can effectively improve the driving performance. Please note that the whole pipeline is differentiable, enabling end-to-end learning from raw sensor data to the outputs. In addition, we follow [40] and adopt a similar optical flow distillation paradigm to further improve the performance. Specifically, we refer to the above network as the student network (INMP-S) and additionally develop a teacher network (INMP-T), which adopts a similar architecture to the student network but further takes optical flow as an additional type of input. Optical flow can provide explicit motion information, leading to significant performance improvement for the teacher network. However, the computation of the optical flow seriously hinders the whole pipeline to achieve real-time performance [38, 41]. We then distill the knowledge from the teacher network to the student network, which can effectively enhance the student network performance while still maintaining its real-time inference speed. We demonstrate the effectiveness and efficiency of our approach on the popular nuScenes dataset [2] and in the closed-loop Carla simulation environment [9]. Our INMP can achieve competitive performance on the detection, prediction, and planning tasks. Moreover, the adopted optical flow distillation paradigm enables our student network to achieve a much faster inference speed than the teacher network with similar driving performance.

2. Related Work

2.1. Trajectory Prediction

Trajectory prediction aims to estimate the future trajectories of the agents based on their past states. The major challenges of this task are modeling the interactions between different agents and generating accurate multi-modal trajectory predictions. Traditional approaches generally achieve it based on manually designed models, *e.g.*, the Kalman filter [22]. With the advancement of deep learning techniques, many data-driven approaches have achieved impressive performance in this field. These approaches typically use the past states to learn a latent representation for each agent, and model the interactions between different agents based on their latent representations [1, 21, 16]. Recently, some researchers have developed a new paradigm that takes the raw sensor data as input to jointly perform object detection and trajectory prediction [46, 47, 24]. These approaches usually use LiDARs, since trajectory prediction is often performed in BEV space and the point clouds provided by LiDARs meet this requirement inherently. Considering that images can provide more semantic information than point clouds and cameras are much cheaper than LiDARs, we take images as input in our approach. Extensive experiments have demonstrated that the proposed vision-based approach can achieve competitive performance compared with previous LiDAR-based approaches, as presented in Section 4.

2.2. Motion Planning

The goal of motion planning is to generate a trajectory to drive the SDV towards its given destination safely and efficiently. Traditional approaches generally sample a large set of candidate trajectories based on the input perception and prediction results [29, 11, 12, 42, 43], and then use a cost function to select the executed trajectory, which has the minimal cost [34]. Recently, many end-to-end approaches that directly map the raw sensor data to the planned trajectories or control commands have been proposed [15, 7, 4]. These approaches are optimized jointly from data, and thus can compensate the adverse effects caused by the accumulated errors in traditional approaches [23]. However, the end-to-end approaches are often criticized for their lack of interpretability, which makes these approaches hard to explain the generated behaviors and further leads to their limited applications in practice. To address it, some approaches have adopted the multi-task learning paradigm, jointly conducting detection, prediction, and planning tasks [46, 47, 33]. The generated intermediate results, *i.e.*, detection and planning results, can effectively help people understand why the model can produce specific motion planning results. However, these approaches typically follow the non-interactive prediction and planning paradigm, and can easily fail in dense traffic scenarios, as mentioned above.

Recently, some researches have proposed the end-to-end interactive paradigm [31, 37, 25]. These approaches typically take the point clouds provided by LiDARs as input, and utilize joint probability distribution models to perform interactive prediction and planning. In this paper, we follow this paradigm and explore its feasibility and effectiveness when images are given as input.

2.3. Knowledge Distillation

Knowledge distillation aims at leveraging the dark knowledge of a teacher network to improve the performance of a student network with fewer parameters. This paradigm was first proposed in [18] for image classification. After that, researchers have presented more effective and efficient knowledge distillation techniques [32, 45]. Specifically, [32] proposed hint training (HT), which aims at training the intermediate representation of the student network such that it can mimic the latent representation of the teacher network. [45] defined attention maps for two networks and then forced the student network to mimic the attention maps of the teacher network. Knowledge distillation has been adopted in many other applications, *e.g.*, object detection [6] and semantic segmentation [17], to improve their performance. In this paper, we follow [40] and adopt a similar optical flow distillation paradigm for autonomous driving. Different from [40], we utilize this technique for the detection, prediction, and planning tasks jointly. In addition, [40] adopts a non-interactive paradigm, while our INMP can perform interactive prediction and planning.

3. Methodology

Fig. 2 illustrates the overview of the proposed approach. Our INMP first generates a BEV feature map $\mathcal{F}_b$, as introduced in Section 3.1. $\mathcal{F}_b$ is then processed to 1) detect other agents and 2) perform interactive prediction and planning, as presented in Section 3.2 and Section 3.3, respectively. After that, Section 3.4 elaborates the proposed optical flow distillation paradigm. Finally, we introduce the training phase in Section 3.5.

3.1. BEV Feature Map Generation

Let $\mathcal{I}_t^k \in \mathbb{R}^{H \times W \times 3}$ denote the input RGB image, where $t = t_0 - 4, \ldots, t_0$ denotes the timestamp of the past five frames; and $k = 1, \ldots, 6$ denotes the six cameras used in our experiments. The six cameras with known extrinsic and intrinsic parameters roughly point in the forward,

forward-left, forward-right, backward, backward-left, and backward-right directions respectively. We also take the HD map $\mathcal{M}_{t_0}$ that contains the road, lane and intersection information as input, since it can provide a strong prior about the driving scenario. Then, given all images in the past five frames $\{\mathcal{I}_t^k\}_{t=t_0-4,\ldots,t_0}^{k=1,\ldots,6}$ and the current HD map $\mathcal{M}_{t_0}$, we aim to generate a BEV feature map $\mathcal{F}_b$, as presented in Fig. 3. $\mathcal{F}_b$ plays an important role in the following object detection and interactive prediction and planning.

Considering that images are located in perspective-view space, we first conduct monocular depth estimation for each $\mathcal{I}_t^k$, which builds a bridge between perspective-view space and BEV space. To achieve it, we follow [30] and generate contextual features at all possible depths for each pixel. Specifically, we associate each pixel with a set of $|\mathcal{D}|$ discrete depths, where $\mathcal{D} = \{d_0 + \Delta d, \ldots, d_0 + |\mathcal{D}|\Delta d\}$. Then, we use the known intrinsic parameters to produce a point cloud $\mathcal{P}_t^k$ that contains $H \cdot W \cdot |\mathcal{D}|$ 3-D points for each $\mathcal{I}_t^k$. To obtain the contextual feature for each point in $\mathcal{P}_t^k$, we first use an image backbone to generate a contextual feature $\mathbf{f} \in \mathbb{R}^C$ and a distribution π over the discrete depth set $\mathcal{D}$ for each pixel $\mathbf{p}$. Afterwards, the contextual feature $\mathbf{f}_d \in \mathbb{R}^C$ for point $\mathbf{p}_d$ is computed as a combination of the feature for the corresponding pixel and the discrete depth inference:

$$\mathbf{f}_d = \pi_d \cdot \mathbf{f}, \tag{1}$$

where $d \in \mathcal{D}$ refers to any discrete depth in $\mathcal{D}$.

For the teacher network, we incorporate optical flow information into $\mathcal{P}_t^k$ to enhance the network's capability to model dynamic relationships for performance improvement. Specifically, we use an existing optical flow estimation network [38] to compute the backward optical flow $\mathcal{O}_t^k \in \mathbb{R}^{H\times W\times 2}$:

$$\mathcal{I}_t^k(u, v) = \mathcal{I}_{t-1}^k\left(u + \mathcal{O}_t^k(u, v, 1), v + \mathcal{O}_t^k(u, v, 2)\right). \tag{2}$$

$\mathcal{O}_t^k$ can be regarded as containing the explicit past motion information from $\mathcal{I}_{t-1}^k$ to $\mathcal{I}_t^k$. Then, we use a flow backbone to produce a contextual feature $\mathbf{f}' \in \mathbb{R}^C$ for each pixel $\mathbf{p}$. After that, we concatenate $\mathbf{f}'$ with $\mathbf{f}$ and produce a new feature. Please note that the new feature is still denoted as $\mathbf{f}$ for notational simplicity. However, $\mathbf{f}$ in the teacher network contains the explicit motion information provided by the optical flow while $\mathbf{f}$ in the student network does not. We then use (1) to compute a contextual feature $\mathbf{f}_d \in \mathbb{R}^C$ for every point $\mathbf{p}_d$ in the teacher network.

Then, we can use the known extrinsic parameters to aggregate $\{\mathcal{P}_t^k\}^{k=1,\ldots,6}$ into a large point cloud $\mathcal{P}_t$ for each timestamp t. After that, we follow [20] to convert $\mathcal{P}_t$ into "pillars", which refer to voxels with infinite height. To be specific, we assign each point to its nearest pillar and use pooling operation to construct a feature map $\mathcal{F}_t$, which contains the information in BEV space and can be processed by convolutional layers. We also use a map backbone to produce a feature map $\mathcal{F}_m$, which is then concatenated with the features of all five past frames $\{\mathcal{F}_t\}_{t=t_0,\ldots,t=t_0-4}$ and processed by convolutional layers to generate the BEV feature map $\mathcal{F}_b$, as shown in Fig. 3.

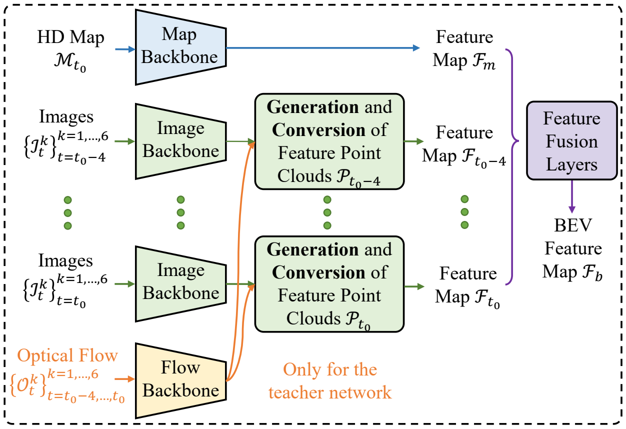
HD Map $\mathcal{M}_{t_0}$
Map Backbone
Feature Map $\mathcal{F}_m$
Images $\{\mathcal{I}_t^k\}_{t=t_0-4}^{k=1,\ldots,6}$
Image Backbone
Generation and Conversion of Feature Point Clouds $\mathcal{P}_{t_0-4}$
Feature Map $\mathcal{F}_{t_0-4}$
Feature Fusion Layers
BEV Feature Map $\mathcal{F}_b$
Images $\{\mathcal{I}_t^k\}_{t=t_0}^{k=1,\ldots,6}$
Image Backbone
Generation and Conversion of Feature Point Clouds $\mathcal{P}_{t_0}$
Feature Map $\mathcal{F}_{t_0}$
Optical Flow $\{\mathcal{O}_t^k\}_{t=t_0-4,\ldots,t_0}^{k=1,\ldots,6}$
Flow Backbone
Only for the teacher network

Figure 3. An illustration of the BEV feature map generation. We lift the input images into 3-D, and combine them with the HD map to generate a BEV feature map. The teacher network also uses optical flow to enhance its capability to model dynamic relationships for performance improvement.

3.2. Object Detection

Given the BEV feature map $\mathcal{F}_b$, we first detect other agents via a single-shot detection header. Specifically, following [27], we apply two convolutional layers on $\mathcal{F}_b$ separately, one for classifying the class of a location, the other one for regressing the position offset, size, heading angle, and velocity of each agent. After that, we use a non-maximum suppression (NMS) operation [28] to obtain the bounding boxes and velocities of all other agents, which are then utilized to perform interactive prediction and planning.

The training loss for object detection $\mathcal{L}_O$ is defined as a summation of a classification loss $\mathcal{L}_{OC}$ and a regression loss $\mathcal{L}_{OR}$, *i.e.*, $\mathcal{L}_O = \mathcal{L}_{OC} + \mathcal{L}_{OR}$. To be specific, in $\mathcal{L}_{OC}$, we use a cross entropy classification loss and assign the label of each anchor based on its intersection over union (IoU) with any agent as follows:

$$\mathcal{L}_{OC}(\widehat{\mathcal{C}}, \mathcal{C}) = H\left(\widehat{\mathcal{C}}, \mathcal{C}\right), \tag{3}$$

where $H(\cdot, \cdot)$ denotes the cross entropy; and $\widehat{\mathcal{C}}$ and $\mathcal{C}$ denote the ground-truth and the predicted classification distribution, respectively. In our experiments, we detect two kinds of agents, *i.e.*, vehicles and pedestrians. In addition, $\mathcal{L}_{OR}$ is defined as a smooth L_1 loss between the regression ground truth $\widehat{\mathcal{S}}$ and regression predictions $\mathcal{S}$ as follows:

$$\mathcal{L}_{OR}(\widehat{\mathcal{S}}, \mathcal{S}) = \sum_k SL_1\left(\widehat{\mathcal{S}_k}, \mathcal{S}_k\right), \tag{4}$$

where $SL_1(\cdot,\cdot)$ denotes the smooth L_1 loss; and the regression state set contains the position offsets in two dimensions, the width and height of the bounding box, the sine and cosine value of the orientation angle, and the velocities in two dimensions.

3.3. Interactive Prediction and Planning

Given the object detection results, we then focus on generating $\mathcal{T} = \{\tau_0, \tau_1, \ldots, \tau_N\}$, which contains the planned trajectory of the SDV τ_0 and the trajectory predictions of N detected agents $\mathcal{T}_r = \{\tau_1, \ldots, \tau_N\}$. Considering that performing prediction and planning in continuous space can consume much computational resources, we follow [46, 47, 25] and sample trajectories in a discrete space, which contains K possible candidate trajectories for each trajectory $\tau \in \mathcal{T}$. The adopted trajectory sampler takes the past trajectories of each agent as input, and generates a set of straight lines, circular curves and euler spirals as candidate trajectories. Now, the generation of each trajectory $\tau \in \mathcal{T}$ is transformed to a classification problem.

To achieve it, we first define a joint probability distribution over the prediction and planning results $\mathcal{T}$ conditioned on the environmental context $\mathcal{X}$ as follows:

$$p(\mathcal{T}|\mathcal{X};\mathbf{w}) = \frac{1}{\mathcal{N}} \exp\left(-\mathcal{E}(\mathcal{T},\mathcal{X};\mathbf{w})\right), \tag{5}$$

where $\mathcal{N}$ is a normalizer; $\mathcal{X}$ includes the BEV feature map $\mathcal{F}_b$ and the past trajectories of each agent; $\mathcal{E}(\mathcal{T},\mathcal{X};\mathbf{w})$ denotes the defined joint energy of the prediction and planning results $\mathcal{T}$; and $\mathbf{w}$ denotes the parameters of the model. To be specific, $\mathcal{E}(\mathcal{T},\mathcal{X};\mathbf{w})$ is defined as a summation of an agent-specific term $\mathcal{E}_a$, a safety term $\mathcal{E}_s$ and a goal-directed term $\mathcal{E}_g$ as follows:

$$\mathcal{E}(\mathcal{T},\mathcal{X};\mathbf{w}) = \sum_{i=0}^{N} \mathcal{E}_a(\tau_i,\mathcal{X};\mathbf{w}) + \sum_{i,j} \mathcal{E}_s(\tau_i,\tau_j) + \mathcal{E}_g(\tau_0), \tag{6}$$

where $\mathcal{E}_a$ is used to evaluate all K candidate trajectories for each agent; $\mathcal{E}_s$ is designed to penalize the occurrence of dangerous cases such as collision; and $\mathcal{E}_g$ is utilized to encourage the SDV to follow the input high-level route. Specifically, the BEV feature map $\mathcal{F}_b$ is combined with the candidate trajectories and then processed via a multi-layer perceptron (MLP) to produce a $K \times (N+1)$ matrix of evaluation scores for $\mathcal{E}_a$. Moreover, we follow [34] and define $\mathcal{E}_s$ as a summation of a collision term and a safety distance violation term. The former will present 1 if the collision between a pair of future trajectories happens and 0 if not; while the latter is defined as a squared penalty within the safety distance of each agent's bounding box, scaled by the velocity of the SDV. Additionally, $\mathcal{E}_g$ is defined as the average projected distance between the planned trajectory of the SDV τ_0 and the input high-level route.

Then, we can determine the planned trajectory of the SDV τ_0 by selecting the candidate trajectory with the minimal cost of a defined cost function f_I as follows:

$$\tau_0^* = \arg\min_{\tau_0} f_I(\mathcal{T},\mathcal{X};\mathbf{w}). \tag{7}$$

Compare with the non-interactive approaches, our INMP enables the SDV to reason about how other agents will react to its behaviors, *i.e.*, considering the trajectory predictions of the other agents $\mathcal{T}_r$ conditioned on τ_0 in the planning objective. Based on this philosophy, the cost function f_I is defined as an expectation of the joint energy over the trajectory prediction distribution of the other agents conditioned on the planned trajectory of the SDV, as follows:

$$f_I(\mathcal{T},\mathcal{X};\mathbf{w}) = \mathbb{E}_{\mathcal{T}_r \sim p(\mathcal{T}_r|\tau_0,\mathcal{X};\mathbf{w})}\left[\mathcal{E}(\mathcal{T},\mathcal{X};\mathbf{w})\right]. \tag{8}$$

By substituting (6) into (8), we can have

$$\begin{aligned} f_I = \mathcal{E}_a(\tau_0) + \mathcal{E}_g(\tau_0) + \mathbb{E}_{\mathcal{T}_r \sim p(\mathcal{T}_r|\tau_0,\mathcal{X};\mathbf{w})}[\sum_{i=1}^{N} \mathcal{E}_a(\tau_i) \\ + \sum_{i=1}^{N} \mathcal{E}_s(\tau_0,\tau_i) + \sum_{i=1,j=1}^{N,N} \mathcal{E}_s(\tau_i,\tau_j)]. \end{aligned} \tag{9}$$

where f_I is short for $f_I(\mathcal{T},\mathcal{X};\mathbf{w})$; and $\mathcal{E}_a(\tau_i)$ is short for $\mathcal{E}_a(\tau_i,\mathcal{X};\mathbf{w})$. Since [25] have shown that excluding the interaction term between the other agents, *i.e.*, $\sum_{i=1,j=1}^{N,N} \mathcal{E}_s(\tau_i,\tau_j)$, does not lead to significant performance degradation, we follow it and also exclude this term in (9) for computational efficiency. Then, for any given τ_0, we can compute the expectation directly by simplifying the terms into the marginal probabilities as follows:

$$\begin{aligned} f_I = \mathcal{E}_a(\tau_0) + \mathcal{E}_g(\tau_0) + \sum_{i,\tau_i} p(\tau_i \mid \tau_0,\mathcal{X};\mathbf{w})\, \mathcal{E}_a(\tau_i) \\ + \sum_{i,\tau_i} p(\tau_i \mid \tau_0,\mathcal{X};\mathbf{w})\, \mathcal{E}_s(\tau_0,\tau_i). \end{aligned} \tag{10}$$

Now, we can successfully perform the proposed interactive prediction and planning, *i.e.*, estimating the future trajectories of the detected agents and producing safe motion plans for the SDV jointly, by computing these marginal probabilities via loopy belief propagation (LBP) [44] efficiently.

Considering that the generation of each trajectory $\tau \in \mathcal{T}$ is transformed to a classification problem, we define the training loss for interactive prediction and planning $\mathcal{L}_P$ as follows:

$$\mathcal{L}_P(\widehat{\mathcal{T}},\mathcal{T}) = \sum_{i} H\left(p(\widehat{\tau_i}), p(\tau_i)\right) + \sum_{i,j} H\left(p(\widehat{\tau_i},\widehat{\tau_j}), p(\tau_i,\tau_j)\right), \tag{11}$$

where $\widehat{\mathcal{T}} = \{\widehat{\tau_0}, \widehat{\tau_1}, \ldots, \widehat{\tau_N}\}$ denotes the prediction and planning ground truth; and $p(\cdot)$ and $p(\cdot,\cdot)$ denote the marginal probabilities. Please note that we follow [25] and define $U(\widehat{\tau}_i)$ as a set of the trajectories close to $\widehat{\tau}_i$. In (11), we only compute the loss for $\tau_i \notin U(\widehat{\tau}_i)$, since we do not want to penalize the trajectory close to the ground truth.

3.4. Optical Flow Distillation Paradigm

As mentioned previously, optical flow can provide explicit motion information, leading to significant performance improvement for the teacher network. However, the computation of the optical flow seriously hinders the whole pipeline to achieve real-time performance. We then follow [40] and distill the knowledge from the teacher network to the student network, which can effectively enhance the student network while still maintaining its real-time performance. The distillation loss $\mathcal{L}_D$ is defined as follows:

$$\mathcal{L}_D = \lambda_{DO}\mathcal{L}_{DO} + \lambda_{DP}\mathcal{L}_{DP} + \lambda_{DF}\mathcal{L}_{DF}, \tag{12}$$

where $\mathcal{L}_{DO}$, $\mathcal{L}_{DP}$ and $\mathcal{L}_{DF}$ denote the distillation loss for object detection, interactive prediction and planning, and the BEV feature map $\mathcal{F}_b$, respectively; and λ_{DO}, λ_{DP} and λ_{DF} are the hyperparameters that scale the three loss terms.

Similar to $\mathcal{L}_O$, $\mathcal{L}_{DO}$ is defined as a summation of a classification distillation loss $\mathcal{L}_{DOC}$ and a regression loss $\mathcal{L}_{DOR}$, *i.e.*, $\mathcal{L}_{DO} = \mathcal{L}_{DOC} + \mathcal{L}_{DOR}$. We follow [18] and define $\mathcal{L}_{DOC}$ as follows:

$$\mathcal{L}_{DOC} = \mathcal{L}_{OC}(C^T, C^S) = H\left(C^T, C^S\right), \tag{13}$$

where C^T and C^S denote the predicted classification distributions of the teacher and student networks, respectively. Different from $\widehat{C}$ in (3) that can only provide hard information, C^T can provide useful soft information to effectively improve the student network. In addition, inspired by [6], we design $\mathcal{L}_{DOR}$ as follows:

$$\mathcal{L}_{DOR} = \sum_k \begin{cases} SL_1\left(\mathcal{S}_k^T, \mathcal{S}_k^S\right), \text{if} \|\widehat{\mathcal{S}_k} - \mathcal{S}_k^S\|_1 > \|\widehat{\mathcal{S}_k} - \mathcal{S}_k^T\|_1, \\ 0, \qquad \text{otherwise.} \end{cases} \tag{14}$$

where $\|\cdot\|_1$ denotes the L_1 norm; and $\mathcal{S}^T$ and $\mathcal{S}^S$ denote the regression predictions of the teacher and student networks, respectively. $\mathcal{L}_{DOR}$ encourages the student network to be close or better than the teacher network, but does not push the student once it reaches the teacher's performance.

Moreover, we define $\mathcal{L}_{DP}$ as follows:

$$\mathcal{L}_{DP} = \begin{cases} \mathcal{L}_P(\mathcal{T}^T, \mathcal{T}^S), \text{if} \sum_i D(\widehat{\tau}_i, \tau_i^{S*}) > \sum_i D(\widehat{\tau}_i, \tau_i^{T*}), \\ 0, \qquad \text{otherwise.} \end{cases} \tag{15}$$

where $\mathcal{T}^T$ and $\mathcal{T}^S$ denote the prediction and planning results of the teacher and student networks, respectively; $\widehat{\tau}_i$ denotes the trajectory ground truth; τ_i^{T*} and τ_i^{S*} denote the trajectories of the teacher and student networks with the minimal cost of f_I, respectively; and $D(\cdot,\cdot)$ measures the average projected distance between two trajectories. Similar to $\mathcal{L}_{DOR}$, $\mathcal{L}_{DP}$ also encourages the student network to perform better than the teacher, network but does not push the student too much.

Considering that $\mathcal{F}_b$ of the teacher network incorporates the explicit motion information provided by the optical flow while $\mathcal{F}_b$ of the student network does not, we follow HT [32] and further design $\mathcal{L}_{DF}$ as:

$$\mathcal{L}_{DF} = \|\mathcal{F}_b^T - \mathcal{F}_b^S\|_1, \tag{16}$$

where $\mathcal{F}_b^T$ and $\mathcal{F}_b^S$ denote the BEV feature maps $\mathcal{F}_b$ of the teacher and student networks, respectively. $\mathcal{L}_{DF}$ encourages the student network to mimic the BEV feature map $\mathcal{F}_b$ of the teacher network.

3.5. Training Phase

In the training phase, we first use the following teacher training loss $\mathcal{L}^T$ to train the teacher network:

$$\mathcal{L}^T = \lambda_O\mathcal{L}_O^T + \lambda_P\mathcal{L}_P^T, \tag{17}$$

where $\mathcal{L}_O^T = \mathcal{L}_{OC}(\widehat{C}, C^T) + \mathcal{L}_{OR}(\widehat{\mathcal{S}}, \mathcal{S}^T)$; $\mathcal{L}_P^T = \mathcal{L}_P(\widehat{\mathcal{T}}, \mathcal{T}^T)$; and λ_O and λ_P are the hyperparameters that scale the two loss terms.

After that, we utilize the following student training loss $\mathcal{L}^S$ to train the student network based on the trained teacher network:

$$\mathcal{L}^S = \lambda_O\mathcal{L}_O^S + \lambda_P\mathcal{L}_P^S + \lambda_D\mathcal{L}_D, \tag{18}$$

where $\mathcal{L}_O^S = \mathcal{L}_{OC}(\widehat{C}, C^S) + \mathcal{L}_{OR}(\widehat{\mathcal{S}}, \mathcal{S}^S)$; $\mathcal{L}_P^T = \mathcal{L}_P(\widehat{\mathcal{T}}, \mathcal{T}^S)$; and λ_O, λ_P and λ_D are the hyperparameters that scale the three loss terms.

4. Experimental Results and Discussions

4.1. Datasets and Implementation Details

In our experiments, we first evaluate the performance of our approach for object detection and trajectory prediction on the nuScenes dataset [2], which contains around 1000 human driving sequences. The dataset is split into a training, a validation and a test set that consists of 18072, 8019 and 8033 samples, respectively. The best-performing networks are selected on the validation set and evaluated on the test set. We also conduct closed-loop evaluation in the Carla simulation environment [9]. Specifically, we first collect a large-scale driving dataset on different maps with different weather and illumination conditions, *e.g.*, clear, rainy, daytime and sunset. We also set random roaming pedestrians and vehicles, which are controlled by the Carla simulator [9]. The dataset is split into a training set with 200K samples and a validation set with 50K samples. Finally,

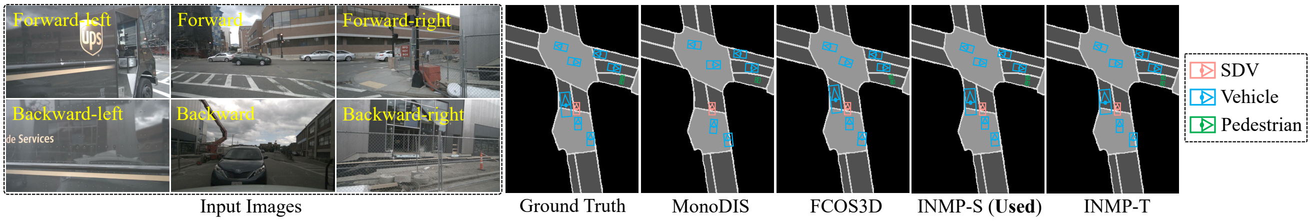

Figure 4. Object detection results of MonoDIS [36], FCOS3D [8], our INMP-S, and our INMP-T on the nuScenes dataset [2]. It is evident that our INMP-T and INMP-S can produce more accurate results than other approaches.

Table 1. Object detection results (%) on the nuScenes dataset [2]. Best results are shown in the bold type.

Approach	$\text{AP}_{vehicle}$	$\text{AP}_{pedestrian}$	mAP
MonoDIS [36]	45.67	36.70	41.19
CenterNet [10]	52.06	37.85	44.96
FCOS3D [8]	51.34	39.21	45.28
INMP-S-ND	49.79	38.95	44.37
INMP-S (**Used**)	52.58	**40.63**	46.61
INMP-T	**53.81**	40.34	**47.08**

Table 2. Trajectory prediction results (m) on the nuScenes dataset [2]. Best results are shown in the bold type.

Approach	Type	L_2	minMSD
NMP [46]	LiDAR-based	2.36	3.22
ESP [31]	LiDAR-based	2.15	2.93
DSDNet [47]	LiDAR-based	2.04	2.65
INMP-S-ND	Vision-based	2.29	3.10
INMP-S (**Used**)	Vision-based	2.07	2.68
INMP-T	Vision-based	**1.95**	**2.59**

each network is evaluated thoroughly with 1800 episodes (around $1000km$) in a closed-loop manner.

For the implementation details, we adopt EfficientNet-B0 [39] as the map, flow and image backbones. In addition, our INMP takes the information of past $2s$ as input and performs interactive prediction and planning for the future $4s$. We adopt the Adam optimizer [19] with an initial learning rate of 10^{-4} to train our INMP-T and INMP-S on two NVIDIA GeForce RTX 2080 Ti GPUs. Moreover, we train the student network without the proposed optical flow distillation paradigm, referred to as INMP-S-ND, for performance comparison.

4.2. Object Detection Results

We follow the nuScenes benchmark [2] and adopt the average precision (AP) at the $1m$ distance threshold as our evaluation metric. We compute the AP for vehicles and pedestrians respectively, and also compute its mean value (mAP) across the two classes. The evaluation results are shown in Table 1. It is evident that the three variants of our INMP all achieve competitive performance compared to the existing vision-based approaches, and our INMP-T achieves the best performance. In addition, our INMP-S presents a better performance than INMP-S-ND and a similar performance to INMP-T thanks to the adopted optical flow distillation paradigm. The qualitative results in Fig. 4 also confirm the above conclusions. Moreover, we adopt INMP-S in practice due to its real-time inference speed, as analyzed in Section 4.4.

4.3. Trajectory Prediction Results

We use the L_2 distance at $t = 4s$ [47] and the minMSD (5 agents and $K = 12$) [31] for performance comparison between trajectory prediction approaches. These two metrics both measure the distance between the trajectory prediction and the ground truth for the correctly detected agents, and the evaluation results are presented in Table 2. We can see that the conclusions in Section 4.2 also hold for the trajectory prediction task. Excitingly, our INMP-S and INMP-T can even achieve competitive performance compared to existing LiDAR-based approaches, which strongly demonstrates that our energy-based model can effectively generate accurate trajectory predictions.

4.4. Closed-loop Evaluation Results

We adopt the success rate (SR) and right lane rate (RL) as our evaluation metrics [5]. SR is defined as the proportion of the successfully finished episodes to the total testing episodes, while RL is defined as the proportion of the period when the SDV drives in the input high-level route to the total driving time. To verify the effectiveness of our INMP, we further develop a non-interactive neural motion planner (NINMP) by using the following cost function: $f_{NI}(\mathcal{T}, \mathcal{X}; \mathbf{w}) = \mathbb{E}_{\mathcal{T}_r \sim p(\mathcal{T}_r | \mathcal{X}; \mathbf{w})} [C(\mathcal{T}, \mathcal{X}; \mathbf{w})]$. Different from (8), f_{NI} considers the trajectory predictions of the other agents $\mathcal{T}_r$ unconditioned on the planned trajectory of the SDV τ_0. Moreover, we record the inference time of each approach on the NVIDIA GeForce RTX 2080 Ti GPU.

Table 3 presents the evaluation results, where it is evident that the conclusions in Section 4.2 also hold for the closed-

Table 3. Closed-loop evaluation results in the Carla simulator [9]. Best results are shown in the bold type.

Approach	SR (%)	RL (%)	Time (s)
CIL [7]	60.72	82.97	0.07
VTP [3]	76.11	80.89	0.08
NINMP-S	80.94	82.31	**0.05**
INMP-S-ND	81.39	85.63	0.06
INMP-S (**Used**)	91.28	93.96	0.06
INMP-T	**92.33**	**95.20**	0.21

Table 4. Closed-loop evaluation results of our INMP-S with some of the loss terms disabled in the Carla simulator [9]. Best results are shown in the bold type.

Variant	$\mathcal{L}_{DO}$	$\mathcal{L}_{DP}$	$\mathcal{L}_{DF}$	SR (%)
(a) INMP-S	–	–	–	81.39
(b) INMP-S	✓	✓	–	89.61
(c) INMP-S	✓	–	✓	83.39
(d) INMP-S	–	✓	✓	87.83
(e) INMP-S (**Used**)	✓	✓	✓	**91.28**

loop autonomous driving task. Our INMP-T achieves the best performance thanks to the explicit motion information provided by the optical flow. Moreover, our INMP-S can present a real-time inference speed with a similar performance to INMP-T, which demonstrates the effectiveness of our optical flow distillation paradigm. This is also the reason why we adopt INMP-S in practice. In addition, our INMP-S outperforms the non-interactive approach, NINMP-S, in terms of both SR and RL. This implies that by considering how other agents will react to the SDV's behaviors in the planning objective, our interactive prediction and planning model can effectively improve the driving performance for the SDV. Fig. 5 presents an example scenario, where the SDV is trying to merge into the left lane. It is evident that the non-interactive SDV (NINMP-S) drifts slowly to the left lane instead of completing a lane merge, while our interactive SDV (INMP-S) can complete a satisfactory lane merge efficiently without bringing safety risk and inconvenience to other agents. All the analysis have demonstrated the effectiveness and efficiency of our proposed approach.

4.5. Ablation Study

We conduct ablation studies to demonstrate the effectiveness of our selection in the loss functions. Specifically, we take INMP-S as the baseline, and test its closed-loop performance with different combinations of loss terms in the Carla simulator [9]. The evaluation results are presented in Table 4, where it can be observed that our paradigm that employs the three loss terms together achieves the best performance for our INMP-S.

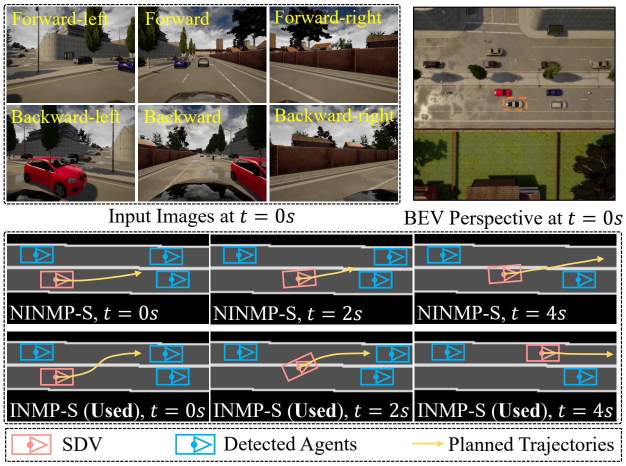

Figure 5. An example scenario in the Carla simulator [9], where the SDV is trying to merge into the left lane. The non-interactive SDV (NINMP-S) drifts slowly to the left lane instead of completing a lane merge, while our interactive SDV (INMP-S) can complete a satisfactory lane merge efficiently without bringing safety risk and inconvenience to other agents. The SDV is marked with an orange dashed box in the BEV perspective.

5. Conclusions

In this paper, we proposed INMP, an end-to-end interactive neural motion planner for autonomous driving. Given a set of past surrounding-view images and a high definition map, our INMP first generated a feature map in bird's-eye-view space, which was then processed to detect other agents and perform interactive prediction and planning jointly. Our interactive prediction and planning paradigm enables the self-driving vehicle to reason about how other agents will react to its behaviors, and thus can significantly improve the driving performance. In addition, we adopted an optical flow distillation paradigm, which can effectively improve the network performance while still maintaining its real-time inference speed. Extensive experiments on the nuScenes dataset and in the closed-loop Carla simulation environment demonstrated the effectiveness and efficiency of our INMP for the detection, prediction, and planning tasks. In the future, we plan to utilize the optical flow distillation paradigm in other tasks related to spatio-temporal information analysis for their performance improvement.

Acknowledgements

This work was supported in part by the Collaborative Research Fund by Research Grants Council Hong Kong under Project C4063-18G, in part by the Department of Science and Technology of Guangdong Province Fund under Project GDST20EG54, and in part by the Zhongshan Municipal Science and Technology Bureau Fund under project ZSST21EG06, awarded to Prof. Ming Liu.

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] 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 (CVPR)*, pages 11621–11631, 2020.

[3] Peide Cai, Yuxiang Sun, Yuying Chen, and Ming Liu. Vision-based trajectory planning via imitation learning for autonomous vehicles. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 2736–2742. IEEE, 2019.

[4] Peide Cai, Yuxiang Sun, Hengli Wang, and Ming Liu. VTGNet: A vision-based trajectory generation network for autonomous vehicles in urban environments. *IEEE Transactions on Intelligent Vehicles*, 2020.

[5] Peide Cai, Sukai Wang, Yuxiang Sun, and Ming Liu. Probabilistic end-to-end vehicle navigation in complex dynamic environments with multimodal sensor fusion. *IEEE Robotics and Automation Letters*, 5(3):4218–4224, 2020.

[6] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 742–751, 2017.

[7] Felipe Codevilla, Matthias Miiller, Antonio López, Vladlen Koltun, and Alexey Dosovitskiy. End-to-end driving via conditional imitation learning. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1–9. IEEE, 2018.

[8] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3d object detection. https://github.com/open-mmlab/mmdetection3d, 2020.

[9] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. CARLA: An open urban driving simulator. In *Conference on Robot Learning (CoRL)*, 2017.

[10] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6569–6578, 2019.

[11] Rui Fan, Hengli Wang, Peide Cai, and Ming Liu. SNE-RoadSeg: Incorporating surface normal information into semantic segmentation for accurate freespace detection. In *European Conference on Computer Vision*, pages 340–356. Springer, 2020.

[12] Rui Fan, Hengli Wang, Peide Cai, Jin Wu, Mohammud Junaid Bocus, Lei Qiao, and Ming Liu. Learning collision-free space detection from stereo images: Homography matrix brings better data augmentation. *IEEE/ASME Transactions on Mechatronics*, 2021.

[13] Rui Fan, Li Wang, Mohammud Junaid Bocus, and Ioannis Pitas. Computer stereo vision for autonomous driving. *CoRR*, 2020.

[14] Jaime F Fisac, Eli Bronstein, Elis Stefansson, Dorsa Sadigh, S Shankar Sastry, and Anca D Dragan. Hierarchical game-theoretic planning for autonomous vehicles. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9590–9596. IEEE, 2019.

[15] Wei Gao, David Hsu, Wee Sun Lee, Shengmei Shen, and Karthikk Subramanian. Intention-net: Integrating planning and deep learning for goal-directed autonomous navigation. In *Conference on Robot Learning (CoRL)*, 2017.

[16] 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.

[17] Tong He, Chunhua Shen, Zhi Tian, Dong Gong, Changming Sun, and Youliang Yan. Knowledge adaptation for efficient semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 578–587, 2019.

[18] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. In *Advances in Neural Information Processing Systems Workshop (NIPSW)*, 2014.

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, 2014.

[20] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12697–12705, 2019.

[21] 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.

[22] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. A survey on motion prediction and risk assessment for intelligent vehicles. *ROBOMECH journal*, 1(1):1–14, 2014.

[23] Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *The Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

[24] Ming Liang, Bin Yang, Wenyuan Zeng, Yun Chen, Rui Hu, Sergio Casas, and Raquel Urtasun. Pnpnet: End-to-end perception and prediction with tracking in the loop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11553–11562, 2020.

[25] Jerry Liu, Wenyuan Zeng, Raquel Urtasun, and Ersin Yumer. Deep structured reactive planning. *CoRR*, 2021.

[26] Tianyu Liu, Qinghai Liao, Lu Gan, Fulong Ma, Jie Cheng, Xupeng Xie, Zhe Wang, Yingbing Chen, Yilong Zhu, Shuyang Zhang, Zhengyong Chen, Yang Liu, Meng Xie, Yang Yu, Zitong Guo, Guang Li, Peidong Yuan, Dong Han, Yuying Chen, Haoyang Ye, Jianhao Jiao, Peng Yun, Zhenhua Xu, Hengli Wang, Huaiyang Huang, Sukai Wang, Peide Cai,

Yuxiang Sun, Yandong Liu, Lujia Wang, and Ming Liu. The role of the hercules autonomous vehicle during the COVID-19 pandemic: An autonomous logistic vehicle for contactless goods transportation. *IEEE Robotics and Automation Magazine*, 2021.

[27] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[28] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006.

[29] Umar Ozgunalp, Rui Fan, Xiao Ai, and Naim Dahnoun. Multiple lane detection algorithm based on novel dense vanishing point estimation. *IEEE Transactions on Intelligent Transportation Systems*, 18(3):621–632, 2016.

[30] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conference on Computer Vision*, pages 194–210. Springer, 2020.

[31] Nicholas Rhinehart, Rowan McAllister, Kris Kitani, and Sergey Levine. Precog: Prediction conditioned on goals in visual multi-agent settings. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2821–2830, 2019.

[32] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In *International Conference on Learning Representations (ICLR)*, 2015.

[33] Abbas Sadat, Sergio Casas, Mengye Ren, Xinyu Wu, Pranaab Dhawan, and Raquel Urtasun. Perceive, predict, and plan: Safe motion planning through interpretable semantic representations. In *European Conference on Computer Vision*, pages 414–430. Springer, 2020.

[34] Abbas Sadat, Mengye Ren, Andrei Pokrovsky, Yen-Chen Lin, Ersin Yumer, and Raquel Urtasun. Jointly learnable behavior and trajectory planning for self-driving vehicles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[35] Dhruv Mauria Saxena, Sangjae Bae, Alireza Nakhaei, Kikuo Fujimura, and Maxim Likhachev. Driving in dense traffic with model-free reinforcement learning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5385–5392. IEEE, 2020.

[36] Andrea Simonelli, Samuel Rota Bulo, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1991–1999, 2019.

[37] Haoran Song, Wenchao Ding, Yuxuan Chen, Shaojie Shen, Michael Yu Wang, and Qifeng Chen. Pip: Planning-informed trajectory prediction for autonomous driving. In *European Conference on Computer Vision*, pages 598–614. Springer, 2020.

[38] Deqing Sun, Xiaodong Yang, Ming-Yu Liu, and Jan Kautz. PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8934–8943, 2018.

[39] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[40] Hengli Wang, Peide Cai, Yuxiang Sun, Lujia Wang, and Ming Liu. Learning interpretable end-to-end vision-based motion planning for autonomous driving with optical flow distillation. In *2021 International Conference on Robotics and Automation (ICRA)*. IEEE, 2021. to be published.

[41] Hengli Wang, Rui Fan, and Ming Liu. CoT-AMFlow: Adaptive modulation network with co-teaching strategy for unsupervised optical flow estimation. In *Conference on Robot Learning (CoRL)*, 2020.

[42] Hengli Wang, Rui Fan, Yuxiang Sun, and Ming Liu. Applying surface normal information in drivable area and road anomaly detection for ground mobile robots. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020.

[43] Hengli Wang, Rui Fan, Yuxiang Sun, and Ming Liu. Dynamic fusion module evolves drivable area and road anomaly detection: A benchmark and algorithms. *IEEE Transactions on Cybernetics*, 2021.

[44] Jonathan S Yedidia, William T Freeman, Yair Weiss, et al. Generalized belief propagation. In *NIPS*, volume 13, pages 689–695, 2000.

[45] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *International Conference on Learning Representations (ICLR)*, 2017.

[46] Wenyuan Zeng, Wenjie Luo, Simon Suo, Abbas Sadat, Bin Yang, Sergio Casas, and Raquel Urtasun. End-to-end interpretable neural motion planner. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8660–8669, 2019.

[47] Wenyuan Zeng, Shenlong Wang, Renjie Liao, Yun Chen, Bin Yang, and Raquel Urtasun. Dsdnet: Deep structured self-driving network. In *European Conference on Computer Vision*, pages 156–172. Springer, 2020.