

# Hierarchical prediction uncertainty-aware motion planning for autonomous driving in lane-changing scenarios

Ruoyu Yao<sup>a</sup>, Xiaotong Sun<sup>a,b,\*</sup>

<sup>a</sup> Intelligent Transportation Thrust, Systems Hub, The Hong Kong University of Science and Technology (Guangzhou), Nansha, Guangzhou, 511458, Guangdong, China

<sup>b</sup> Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Hong Kong Special Administrative Region of China



## ARTICLE INFO

### Keywords:

Autonomous driving  
Lane change  
Behavior prediction  
Uncertainty issues  
Optimization-based planning

## ABSTRACT

Autonomous vehicles (AVs) are expected to achieve safe and efficient interactions with surrounding dynamic objects. Multi-lane driving scenarios, however, intensify the complexity of AV navigation, given the uncertainties associated with neighboring vehicles' lane-changing intentions and the subsequent travel trajectories. Deep learning has demonstrated effectiveness in unraveling complex motion patterns, enabling stochastic predictions of intentions and trajectories. Nonetheless, reduced performance of deep-learning-based prediction may be observed in unseen driving environments owing to their "black-box" nature, potentially leading to incorrect decision-making in AV navigation in these environments. To address these challenges, this paper proposes a comprehensive AV planning framework that integrates hierarchical behavior prediction via deep learning with motion planning based on dynamic programming. A set of safety criteria is introduced within the motion planning module to accommodate hierarchical uncertainties in behavior patterns, adjustable based on the reliability of the prediction model and eschewing rigid distributional assumptions. An improved constrained iterative linear quadratic regulator is devised to handle the corresponding non-convex constraints and to offer efficient online solutions for AV navigation. Evaluations conducted with the INTERACTION and HighD datasets demonstrate the effectiveness of uncertainty-aware planning in enhancing AV safety performance.

## 1. Introduction

Safe and efficient interactions with surrounding dynamic objects are essential for developing autonomous driving systems. When autonomous vehicles (AVs) drive on multi-lane roadways, their smooth two-dimensional driving can be particularly threatened by neighboring vehicles' lane-changing behaviors (Song and Li, 2021). For one thing, lane changes are a typical cause of traffic accidents, with about 17% of total crashes resulting from sudden lane changes (Shawky, 2020). For another, inappropriate lane changes can induce capacity drops and congestion, which eventually deteriorate the overall traffic flow efficiency (Jin, 2010). In this regard, successfully navigating AVs during complex driving scenarios involving lane changes, whether by themselves or their neighbors, requires understanding neighboring vehicles' behavior patterns, anticipating future risks, and generating a conflict-free path for themselves.

\* Corresponding author at: Intelligent Transportation Thrust, Systems Hub, The Hong Kong University of Science and Technology (Guangzhou), Nansha, Guangzhou, 511458, Guangdong, China.

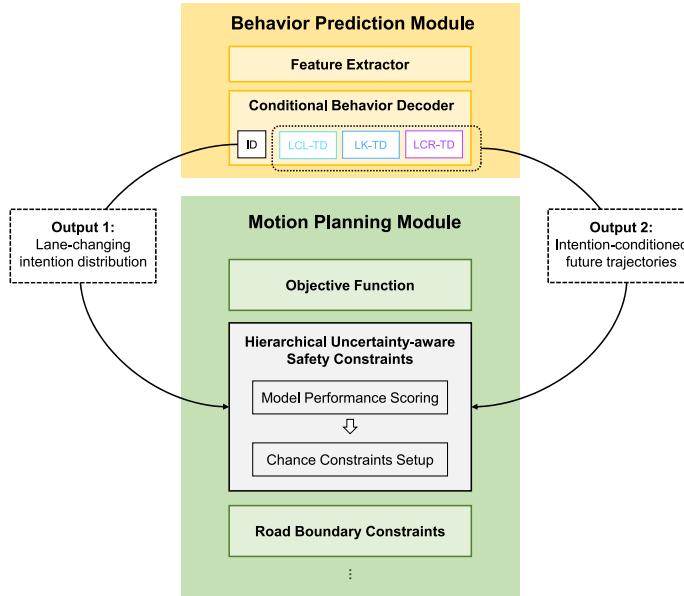
E-mail address: [xtsun@ust.hk](mailto:xtsun@ust.hk) (X. Sun).

In the framework of modern AV operational systems, the aforementioned processes are tackled sequentially by the behavior prediction and motion planning modules (Claussmann et al., 2019). Behavior prediction infers the intention and/or future movements of neighboring road users, as expressed by the probability distribution over upcoming maneuvers and vehicle trajectories. Deep-learning methods (Lee et al., 2017; Gu et al., 2021; Gao et al., 2023; Lan et al., 2023) have been widely adopted, which remarkably improve the accuracy of behavior prediction. Informed by the anticipated behaviors, the motion planning module identifies the potential conflict between AVs and their surrounding objects, and subsequently navigates AVs from initial positions to desired destinations safely, efficiently, and smoothly. Typical methods include optimization (Dolgov et al., 2010; Ziegler et al., 2014), sampling (Kuwata et al., 2009; hwan Jeon et al., 2013), and searching (Likhachev and Ferguson, 2009; McNaughton et al., 2011). Recent advancements in end-to-end learning also enabled a unified operational framework, allowing prediction and planning modules to be tackled in an integrated approach (Zeng et al., 2020; Cui et al., 2021; Hu et al., 2023). End-to-end learning methods can attain remarkable performance in real-time inference, adaptability and perception-awareness. However, challenges related to reduced interpretability and modularity still need to be addressed to enhance their trustworthiness. Regardless of the implemented approaches, safety remains the primary concern, often addressed through constraints that ensure collision avoidance (Chen et al., 2017, 2019; Zeng et al., 2021) and compliance with traffic rules (Liu et al., 2023). When prioritizing the computational efficiency of path planning, many of these studies assume that the movement of obstacles is deterministic, which is reasonable if behavioral predictions are perfectly accurate. However, the inherent uncertainties in behavior prediction challenge this assumption.

The uncertainty of behavior prediction primarily originates from two perspectives: the randomness in motions of the subjects under consideration, referred to as *motion uncertainty*, and concerns regarding the reliability of the prediction model itself, known as *model uncertainty* (Tang et al., 2022; Shao et al., 2023). The challenge intensifies when attempting to predict lane-changing behaviors, as they involve both the continuous longitudinal maneuvers and the lateral ones that usually occur at only a few moments in the forecasting horizon. Traditional microscopic traffic flow models discern lane-changing intentions by examining individual vehicles' rationality. Adopted models include rule-based (Gipps, 1986), utility-based (Toledo et al., 2003; Kesting et al., 2007; Sun and Elefteriadou, 2014), and game theory-based (Ali et al., 2019), producing a probability of changing lanes or staying at the current lanes under studied driving scenarios. As vehicle trajectory data becomes more accessible and widespread, machine learning becomes a dominant method in lane-changing prediction. Popular prediction approaches are based on Long Short-Term Memory (LSTM) (Deo and Trivedi, 2018a,b; Xie et al., 2019), Graph Neural Networks (GNN) (Wu and Liu, 2022), and Transformer (Chen et al., 2022; Gao et al., 2023). Compared to traditional ration-based inference, learning-based predictions also provide possible trajectories in the future conditioned on the anticipated intention to change lanes or stay. This leads to a multi-layer, or termed *multimodal* in deep-learning community (Mozaffari et al., 2020; Chen et al., 2021), predicted behavior. Specifically, the multimodal predictions articulate distributions representing potential lane-changing intentions and demonstrate multiple sampled trajectories conditioned on these intentions. If categorical lateral lane changes are not explicitly accounted for in the prediction model, previous studies suggest that the prediction model may become less reliable when lane changes occur (Tang et al., 2022).

In the literature, the risks of motion uncertainty on path planning efficacy have been noticed, leading to the development of uncertainty-aware motion planners to address these challenges (Aoude et al., 2013; Hardy and Campbell, 2013; Wang et al., 2020; Chen et al., 2021; Han et al., 2022; Zhou et al., 2023). Based on probabilistic predictions of surrounding vehicles' future trajectories, uncertainty-aware motion planners generate feasible paths using chance constraints to bound collision probability at each step along the path. While most studies assume that trajectories are sampled from a Gaussian distribution, Wang et al. (2020) and Han et al. (2022) also explored scenarios with non-Gaussian trajectories. Nonetheless, these studies did not focus on the multi-lane traffic scenarios, leaving the uncertainty associated with lane-change intentions unaddressed. Comparatively, model uncertainty, which is a focal study area in model performance monitoring in the realm of machine learning (Feng et al., 2021), has received less attention in the design of uncertainty-aware motion planners. Among the few exceptions, Ding et al. (2021), Tang et al. (2022), and Li et al. (2023), employed Bayesian Deep Neural Network, deep ensemble (Lakshminarayanan et al., 2017) and dropout approximation (Gal and Ghahramani, 2016), respectively, to quantify model uncertainty. However, as many previous studies focus on general traffic scenario without articulating more safety-critical situations involving lane changes in the prediction module, a comprehensive approach to simultaneously tackling multi-modal motion uncertainty and deep-learning model uncertainty is still absent.

To address the aforementioned challenges and ensure safe autonomous driving during lane-changing scenarios, we propose a novel AV planning framework as illustrated in Fig. 1. The framework comprises a behavior prediction module and a motion planning module. The behavior prediction module employs LSTM (Hochreiter and Schmidhuber, 1997) to encode relevant sequential features and Graph Attention Networks (GAT) (Veličković et al., 2017) to extract multi-agent interaction patterns. Following that, the conditional behavior decoders produce a probability distribution reflecting a nearby vehicle's lane-changing intentions, which are lane-keeping, left or right lane-changes, along with multiple sample trajectories conditioned on each possible intention. The ego vehicle's motion planning is then modeled as an optimization problem that minimizes the total cost over the planning horizon, subject to the vehicle kinematic model and a set of driving viability constraints. Using behavioral prediction results as inputs, the safety constraints are formulated as chance constraints to bound the collision probability between the ego vehicle and its neighbors. To capture the multi-layer motion uncertainty and model uncertainty, the analytical expression of the collision probability progressively incorporates the means and variances in predicted behaviors, and an explicit score that monitors the prediction model performance. This introduces the concept of *hierarchical uncertainty-aware safety constraints*. Subsequently, the optimization problem is reformulated as a dynamic program and is solved by an extension of the Constrained Iterative Linear Quadratic Regulator (CILQR) proposed by Chen et al. (2019). The algorithm, which we name Hierarchical Uncertainty-aware CILQR (HU-CILQR), is capable of dealing with non-convex safety constraints as well as non-Gaussian prediction uncertainties. Our contributions are concluded as follows:



**Fig. 1.** An illustration of the system framework. Lane changing intention distribution and intention-conditioned future trajectories are respectively sent by the Intention Decoder (ID) and three separate Trajectory Decoders (TD) conditioned on Lane-Change to the Left (LCL), Lane-Change to the Right (LCR) and Lane-Keeping (LK), serving as the key information transmitted from the Behavior Prediction Module to the Motion Planning Module.

- We present a novel AV planning framework that tackles uncertainty issues in the lane-changing behavior prediction, where uncertainties from trajectories, intentions, and the prediction model are hierarchically integrated in formulating the safety constraints following general non-Gaussian distributional assumptions (Wang et al., 2020; Han et al., 2022).
- Considering the motion and model uncertainties of learning-based predictions, we introduce an expected, a robust, and an adaptive planning schemes. The first two schemes correspond to regular and conservative driving modes, depending on how the motion uncertainty influence the safety constraints. The third one dynamically adjusts between the previous two, weighted by the trustworthiness of the multi-modal predictive performance.
- The conventional CILQR algorithm is improved to provide both feasible solution search and hard-constraint prevention by adopting soft and hard barrier functions in separate optimization stages.
- Using INTERACTION and HighD datasets, we demonstrate the safety performance of uncertainty-aware motion planning through real-world data in both mandatory and discretionary lane changes.

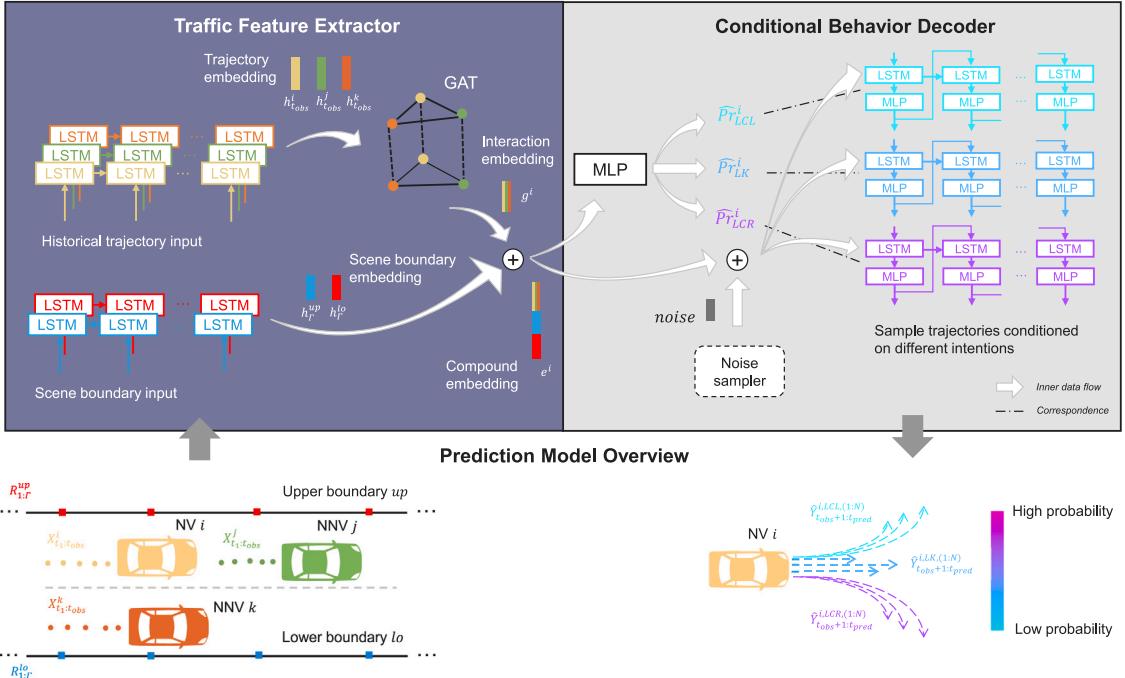
The remainder of the paper is organized as follows. Section 2 introduces learning-based behavior prediction, followed by a detailed description of the motion planning model in Section 3. The algorithm developed to solve the planning problem is introduced in Section 4. Section 5 provides the details of experiments, including both the evaluations of predictions and planning. Section 6 finally concludes the paper.

## 2. Learning-based behavior prediction

The task for the behavior prediction adopts a hierarchical structure for lane changes, with the top layer providing lane-changing intentions at the current moment and the bottom layer comprising sampled trajectories conditioned on these intentions. For clarity, we refer to the autonomous vehicle implementing our method as AV. The vehicles adjacent to the AV, whose behaviors should be predicted by the AV, are named neighboring vehicles (NVs). When predicting an NV's behavior, the vehicles adjacent to the NV are called neighbor's neighboring vehicles (NNVs). The overall architecture of our deep learning-based behavior prediction model is shown in Fig. 2, which is composed of a *traffic feature extractor* and a *conditional behavior decoder*. We name the model Conditional Scene-aware Interactive Graph Attention Network (Cond-SIGAT).

### 2.1. Traffic feature extractor

To predict an NV's lane-changing intention and trajectory, the model needs to extract relevant features from its driving environment. Here, we presume that the NV's future behavior is influenced by its historical motion, its interaction with NNVs, and the proximity of nearby road boundaries. The feature extraction aligns with the procedures in our previous work on driving behavior modeling (Yao et al., 2021), with a key distinction being the encoding of road boundaries to represent the scene semantics of multi-lane roadways, a facet not addressed in the previous work.



**Fig. 2.** Behavior prediction model Cond-SIGAT.

### 2.1.1. Extraction of historical motion features

Historical motion is a key source of information for learning-based behavior prediction (Alahi et al., 2016; Huang et al., 2018; Zhang et al., 2019). We define the NV's input vector at each historical time step  $t$ ,  $X_t^i$ , by the following elements:

$$X_t^i \equiv (\Delta x_t^i, \Delta y_t^i, v_t^i, \zeta^i, v^i).$$

The first two elements in the tuple,  $\Delta x_t^i$  and  $\Delta y_t^i$ , are the two-dimensional coordinates of NV  $i$  in a relative coordination system, whose origin is defined as NV  $i$ 's absolute position at the last observation time-step  $t_{obs}$  to create an ego-centric distance measurement.  $v_t^i$  is the velocity of the vehicle. NV  $i$ 's length and width are represented by  $\zeta^i$  and  $v^i$ , respectively. The input vector corresponding to any NNV  $j$  is defined in a similar way, where the relative position ( $\Delta x_t^j, \Delta y_t^j$ ) is also represented by the coordinates in the NV-centric frame, allowing the network to correctly understand the spatial relationships between an NV and NNVs.

Following this, LSTM networks are employed to derive temporal features from the historical motion sequence. Essentially, the LSTM network recursively updates the hidden state  $h_t^i$  of each NV  $i$  at every time-step  $t$ :

$$h_t^i = LSTM(X_t^i, h_{t-1}^i; W_{MoEnc}), \quad (1)$$

Here,  $W_{MoEnc}$  represents the set of trainable parameters within the motion-encoding LSTM. Through recursive state updates, the temporal motion feature of the NV can be summarized in the last hidden state  $h_{t_{obs}}^i$ . Concurrently, the feature extraction procedure for any arbitrary NNV  $j$  utilizes an LSTM with identical parameters to  $W_{MoEnc}$ , producing a comparable embedded motion feature  $h_{t_{obs}}^j$ .

### 2.1.2. Extraction of vehicle interaction features

We model the interactions between the NV and its NNVs via the *message-passing mechanism* in a GAT, where every vehicle is treated as a node in the graph, and the interaction between each pair of vehicles is represented by an edge connecting the nodes. Each node is marked by its node state, containing information that represents a vehicle's real-time influence on its neighbors, while each edge associates with an attention score, referring to the relative importance of the interaction between two connected nodes. The GAT is comprised of multiple self-attention layers stacking together. At each layer  $l$  with  $l > 1$ , node states and attention scores are updated in a reciprocal manner, using information from the previous layer  $l - 1$ . The exact formulations are provided as follows:

$$g_l^i = \sum_{j \in \mathcal{N}^i} a_l^{i,j} \cdot W_l \cdot g_{l-1}^j, \quad (2)$$

$$\begin{aligned} a_l^{i,j} &= SoftMax \left( LeakyReLU \left( \alpha_l^\top \cdot [W_l \cdot g_{l-1}^i \parallel W_l \cdot g_{l-1}^j] \right) \right) \\ &= \frac{\exp \left( LeakyReLU \left( \alpha_l^\top \cdot [W_l \cdot g_{l-1}^i \parallel W_l \cdot g_{l-1}^j] \right) \right)}{\sum_{k \in \mathcal{N}^i} \exp \left( LeakyReLU \left( \alpha_l^\top \cdot [W_l \cdot g_{l-1}^i \parallel W_l \cdot g_{l-1}^k] \right) \right)} \end{aligned} \quad (3)$$

In particular,  $g_i^l$  represents the node state at vehicle  $i$  at layer  $l$ , which is an amalgamation of the linear transformation of its neighbors' node states from the previous layer,  $g_{l-1}^i$ , weighted by the attention scores,  $a_l^{i,j}$ , between the neighbors in the neighborhood set  $\mathcal{N}^i$  and itself in the current layer. A learnable transformation matrix  $W_l$  is employed to unify the dimensions of different node states, enhancing the GAT's expressive capacity. The attention score  $a_l^{i,j}$ , which quantifies the importance of interaction between vehicle  $i$  and its neighbor  $j$  at layer  $l$ , is modeled following the approach proposed by Veličković et al. (2017). This approach encapsulates node states of vehicle  $i$  and  $j$  at the previous layer,  $g_{l-1}^i$  and  $g_{l-1}^j$ , to measure the interaction between vehicle  $i$  and  $j$ . For calculation, a learnable linear transformation vector  $\alpha_l$  is applied to the concatenation of  $W_l \cdot g_{l-1}^i$  and  $W_l \cdot g_{l-1}^j$ , producing a one-dimensional coefficient. This allows the use of an activation function, LeakyReLU<sup>1</sup> (Maas et al., 2013), to introduce nonlinearity, followed by a softmax function for normalization.

In the algorithm implementation, we employ a *multi-head attention mechanism* to enhance the expressive power of the network and the robustness of the extracted features (Veličković et al., 2017; Vaswani et al., 2017). Instead of using a single attention head,  $H$  independent self-attentions are concatenated to update the node state recursively:

$$g_l^i = \left\| \sum_{\pi=1}^H a_{l,\pi}^{i,j} \cdot W_{l,\pi} \cdot g_{l-1}^j \right\| \quad (4)$$

The node states at the first self-attention layer are initiated using the last hidden states from the motion-encoding LSTMs, that is,  $g_1 = \{h_{t_{obs}}^1, h_{t_{obs}}^2, \dots, h_{t_{obs}}^{|\mathcal{N}^i|}\}$ . This process produces the final output of a  $L$ -layer GAT  $g_L = \{g_L^1, g_L^2, \dots, g_L^{|\mathcal{N}^i|}\}$ . The final output vector of NV  $i$ , denoted as  $\hat{g}_L^i$ , is produced by augmenting  $g_L^i$  with the last hidden state  $h_{t_{obs}}^i$  via a residual connection (Vaswani et al., 2017).

### 2.1.3. Extraction of road boundary features

Boundary information is mainly used for depicting the feasible driving area, preventing the model from making unreasonable predictions (e.g. lane-change to the right when the vehicle is at the rightmost lane). We utilize another LSTM with different parameters, namely the *scene-encoding LSTM*, to process a road boundary around the NV. The intuition behind the operation is that road boundary points are also sequential data that can be addressed by recurrent neural networks. As shown in the bottom-left part of Fig. 2, a total of  $\Gamma$  local sample points are drawn from the upper or lower boundary line to approximate its geometric shape, and point  $\gamma$ 's coordinates represented in the NV-centric frame are used as the input to an LSTM cell:

$$R_\gamma^{up} = (\Delta x_\gamma^{up}, \Delta y_\gamma^{up}) \quad (5a)$$

$$R_\gamma^{lo} = (\Delta x_\gamma^{lo}, \Delta y_\gamma^{lo}) \quad (5b)$$

The scene-encoding LSTMs operate in the same way as that of Eq. (1), with a different set of trainable parameters denoted by  $W_{ScEnc}$ :

$$h_\gamma^{up} = LSTM(R_\gamma^{up}, h_{\gamma-1}^{up}; W_{ScEnc}) \quad (6a)$$

$$h_\gamma^{lo} = LSTM(R_\gamma^{lo}, h_{\gamma-1}^{lo}; W_{ScEnc}) \quad (6b)$$

The outputs from the last cells are denoted as  $h_\Gamma^{up}$  and  $h_\Gamma^{lo}$ .

Ultimately, all outputs from the traffic feature extractor,  $\hat{g}_L^i$ ,  $h_\Gamma^{up}$  and  $h_\Gamma^{lo}$ , are concatenated into a compound representation of NV  $i$ -centric traffic features, denoted as  $e^i$ . This presentation is subsequently forwarded to the conditional behavior decoder for generating the behavior prediction.

## 2.2. Conditional behavior decoder

In the realization of hierarchical behavior prediction, any lower-level trajectory output is contingent upon a higher-level intention class. In this regard, an *intention-branched* structure (Codevilla et al., 2018) is adopted in the decoding network (see Fig. 2). The higher-level information is embedded in the structure through a switching logic during the training process, enabling the establishment of strong conditional relations between trajectories and intentions. To accomplish the stochastic prediction at the lowest level of motion execution, multiple sample trajectories are further provided within each intention class, acknowledging the trajectory uncertainty.

### 2.2.1. Prediction over lane-changing intentions

For the intention prediction task, we build a two-layer Multi-Layer Perceptron (MLP) with activation functions to implement the classifier, formulated as follows:

$$(\hat{Pr}_{LK}^i, \hat{Pr}_{LCL}^i, \hat{Pr}_{LCR}^i) = SoftMax(W_{int2} \cdot ReLU(W_{int1} \cdot e^i)) \quad (7a)$$

Here, the input  $e^i$  is derived from the previous traffic feature extractor, and the learnable weights of the MLP are signified by  $W_{int1}$  and  $W_{int2}$ . The nonlinear activation ReLU is applied to the linear transformation of  $e^i$  parameterized by  $W_{int1}$  first. It is then followed

<sup>1</sup> For  $x \in \mathbb{R}$ ,  $LeakyReLU(x) = x$  if  $x \geq 0$ , and  $0.01x$ , if  $x < 0$ .

by another linear transformation parameterized by  $W_{int2}$  and a SoftMax function, generating the intention distributions  $\hat{Pr}_{LK}^i$ ,  $\hat{Pr}_{LCL}^i$ , and  $\hat{Pr}_{LCR}^i$ , which are the predicted probabilities of lane-keeping ( $LK$ ), lane-changing to the left ( $LCL$ ), and lane-changing to the right ( $LCR$ ), respectively.

### 2.2.2. Prediction over intention-conditioned trajectories

For the trajectory prediction task, we concatenate the compound embedding  $e^i$  with a noise term, as proposed by Gupta et al. (2018). This treatment allows the generation of multiple results that represent the trajectory uncertainty. For each intention class  $I \in \{LK, LCL, LCR\}$ , an LSTM network undertakes the trajectory generation procedure, with the concatenated vector serving as the first hidden state, which is mathematically formulated as the following:

$$o_t^{i,I} = LSTM(\hat{Y}_{t-1}^{i,I}, o_{t-1}^{i,I}; W_{MoDec}^I) \quad (8a)$$

$$\hat{Y}_t^{i,I} = W_{traj}^I \cdot o_t^{i,I} \quad (8b)$$

$$\hat{Y}_t^{i,I} = P_{t_{obs}}^i + \sum_{\tau=t_{obs}+1}^t \hat{Y}_\tau^{i,I} \quad (8c)$$

In this framework, Eq. (8a) defines  $o_t^{i,I}$  as the hidden state of the motion-decoding LSTM, which is parameterized with  $W_{MoDec}^I$ . According to Eq. (8b), the predicted displacement  $\hat{Y}_t^{i,I}$  is derived from the hidden state  $o_t^{i,I}$  using a MLP with a weight  $W_{traj}^I$ . Following this, Eq. (8c) provides the computation for the predicted absolute position  $\hat{Y}_t^{i,I}$  at the current step  $t$ . This prediction is achieved by summing the predicted displacements for each time-step from  $t_{obs}$  onward, starting from the last observed absolute position  $P_{t_{obs}}^i$ . Regarding the special case that the first LSTM cell has no previous displacement prediction output, the input is given by the actual displacement at time-step  $t_{obs}$ .

By running the three separate trajectory decoders multiple times, we draw various sample trajectories conditioned on different intentions. This approach enriches the information on the NV's stochastic future behavior for the downstream AV motion planning task, without relying on strict distributional assumptions.

### 2.3. Loss function and training approach

Training a model with stochastic inferences can be relatively unstable. To ensure the success of our model training, we split the entire training process into two steps. In the first step, we remove the noise sampler from the original model and train the simplified variant that gives a unique trajectory output  $\hat{Y}_{t_{obs}+1:t_{pred}}^{i,I}$  conditioned on each intention. This simplified model is named as Cond-SIGAT-Base. Considering the multi-task property of Cond-SIGAT-Base, we apply a synthesized loss function to balance performance for intention, denoted as  $L_{int}$ , and trajectory predictions referring as  $L_{traj}$ , whose formulations are provided as follows:

$$L_{int} = - \sum_{I \in \{LK, LCL, LCR\}} GT(I, i) \cdot \log \hat{Pr}_I^i \quad (9a)$$

$$L_{traj} = \frac{1}{t_{pred} - t_{obs}} \sum_{I \in \{LK, LCL, LCR\}} GT(I, i) \sum_{t=t_{obs}+1}^{t_{pred}} \left\| \hat{Y}_t^{i,I} - Y_t^i \right\|_2^2 \quad (9b)$$

$$L_{overall} = w_{int} L_{int} + w_{traj} L_{traj} \quad (9c)$$

In both loss functions stated in Eq. (9a)–(9b), a binary indicator,  $GT(\cdot)$ , is imposed to determine whether intention  $I$  is true for NV  $i$ , functioning as a form of switching logic. Logarithmic functions of intention prediction are adopted to measure the intention prediction loss, and the average normalized distance between the predicted trajectory  $\hat{Y}_t^{i,I}$  and the ground-truth trajectory  $Y_t^i$  during the prediction period from  $t_{obs}$  to  $t_{pred}$  is used to evaluate the trajectory prediction loss. Under the switching logic, the calculated trajectory loss is confined to the trajectory decoder that aligns with the verified ground-truth intention, thereby ensuring a dedicated association between each trajectory decoder and its respective intention. The overall loss function,  $L_{overall}$ , is then formulated by linearly integrating the two loss functions, with respective weights  $w_{int}$  and  $w_{traj}$ .

In the second step, the trained parameters of Cond-SIGAT-Base are used to initialize the traffic feature extractor and lane-changing intention decoder of our original model. These parameters are then fixed to preserve the capacity of feature extraction and intention prediction. The following training process only involves trajectory decoders with the noise term, which is implemented by drawing multiple samples from the decoder and applying a variant of loss function  $L_{traj}$ , either an average of  $N$  samples (Average-of-N), or the minimum of  $N$  samples (Minimum-of-N) (Fan et al., 2017; Gupta et al., 2018). The two variants are given by:

$$L_{traj}^{AoN} = \frac{1}{t_{pred} - t_{obs}} \sum_{I \in \{LK, LCL, LCR\}} GT(I, i) \cdot \frac{1}{N} \sum_{\chi=1}^N \sum_{t=t_{obs}+1}^{t_{pred}} \left\| \hat{Y}_t^{i,I,\chi} - Y_t^i \right\|_2^2 \quad (10a)$$

$$L_{traj}^{MoN} = \frac{1}{t_{pred} - t_{obs}} \sum_{I \in \{LK, LCL, LCR\}} GT(I, i) \cdot \arg \min_{\chi \in \{1, \dots, N\}} \sum_{t=t_{obs}+1}^{t_{pred}} \left\| \hat{Y}_t^{i,I,\chi} - Y_t^i \right\|_2^2 \quad (10b)$$

With  $N$  samples, the predicted trajectory point in the  $\chi$ -th trial is denoted by  $\hat{Y}_t^{i,I,\chi}$ , and the total sample trajectories conditioned on an intention is denoted by  $\hat{Y}_{t_{obs}+1:t_{pred}}^{i,I,(1:N)}$ . We name our models trained by these as Cond-SIGAT-AoN and Cond-SIGAT-MoN respectively. Both of the models are investigated in Section 5: Experiments.

### 3. Uncertainty-aware motion planning

In this section, we elaborate on the real-time motion planning for the ego AV, which is formulated as a discrete-time finite-horizon optimization problem, where behavior predictions are incorporated as parameter inputs. We first present the objective for motion planning, followed by the standard model constraints considering the vehicle kinematics and the limits of controls and states. Then, we present the safety constraint in detail, highlighting the chance-constraint-based safety criterion that addresses motion uncertainty and model uncertainty simultaneously.

Motion planning aspires to keep an AV moving along a designated trajectory throughout the planning horizon, under a desired speed as smooth as possible. At each time-step  $t$  in the planning horizon  $[t_{sta}, t_{plan}]$ , the variables include the ego AV's coordinates,  $x_t$  and  $y_t$ , the velocity  $v_t$ , the yaw angle  $\theta_t$ , the acceleration  $a_t$  and angular velocity  $\delta_t$ . To ensure the correct integration of behavior prediction outcomes, the planning horizon  $[t_{sta}, t_{plan}]$  should be aligned with the prediction horizon  $[t_{obs}, t_{pred}]$ . The objective of the motion planning problem is then mathematically formulated as

$$\begin{aligned} \min J_{x_t, y_t, v_t, \theta_t, a_t, \delta_t} = & w_1 \left[ (x_{t_{plan}} - x_{t_{plan}}^{wpt})^2 + (y_{t_{plan}} - y_{t_{plan}}^{wpt})^2 \right] + w_2 (v_{t_{plan}} - v^d)^2 \\ & + \sum_{t=t_{sta}}^{t_{plan}-1} \left\{ w_1 \left[ (x_t - x_t^{wpt})^2 + (y_t - y_t^{wpt})^2 \right] + w_2 (v_t - v^d)^2 + w_3 a_t^2 + w_4 \delta_t^2 \right\}. \end{aligned} \quad (11)$$

This objective is composed of four key components. The first component quantifies the penalty when the AV's trajectory deviates from its reference waypoints, as presented by  $x_t^{wpt}$  and  $y_t^{wpt}$ . The second component captures the cost when the AV deviates from its desired velocity  $v^d$ . The third and fourth components, meanwhile, evaluate the smoothness of the driving experience. Each of these components is assigned a specific weight,  $w_1$ ,  $w_2$ ,  $w_3$ , and  $w_4$ , respectively, to balance their contributions to the overall objective.

#### 3.1. Standard model constraints

We then introduce the standard model constraints for AV motion planning, including the vehicle's kinematic model, control limits, and road boundary constraints.

##### 3.1.1. Vehicle kinematic model

Under the discrete system, the vehicle kinematics are characterized by the transition from its current state to the next state. At each time step, we define the vehicle state  $S_t$  by its position  $x_t$ ,  $y_t$ , velocity  $v_t$ , and yaw angle  $\theta_t$ , and define its action  $A_t$  by acceleration  $a_t$  and angular velocity  $\delta_t$ . The following vehicle kinematic model is introduced to describe the state transition:

$$\begin{cases} x_{t+1} = x_t + \cos \theta_t (v_t T_s + \frac{1}{2} a_t T_s^2) \\ y_{t+1} = y_t + \sin \theta_t (v_t T_s + \frac{1}{2} a_t T_s^2), \quad \forall t \in \{t_{sta}, \dots, t_{plan} - 1\}, \\ v_{t+1} = v_t + a_t T_s \\ \theta_{t+1} = \theta_t + \delta_t T_s \end{cases} \quad (12)$$

Here,  $T_s$  denotes the system time resolution. The initial condition of the system is provided by  $[x_{t_{sta}} = x(0), y_{t_{sta}} = y(0), v_{t_{sta}} = v(0), \theta_{t_{sta}} = \theta(0)]$ , with  $x(0)$ ,  $y(0)$ ,  $v(0)$ , and  $\theta(0)$  denoting the initial state of the vehicle.

##### 3.1.2. Control limit constraints

The vehicle's actions are bounded by the limits of engine force, braking force, and steering wheel, providing inequality constraints:

$$\begin{cases} a_{min} < a_t < a_{max} \\ -\delta_{lim} < \delta_t < \delta_{lim} \end{cases} \quad \forall t \in \{t_{sta}, \dots, t_{plan} - 1\}. \quad (13)$$

Notably,  $a_{max}$  and  $a_{min}$  are the maximum acceleration and minimum acceleration respectively, and  $\delta_{lim}$  represents the angular velocity limit.

##### 3.1.3. Road boundary constraints

As a vehicle must not travel beyond the boundaries of a road, we apply piece-wise linear functions to represent the road boundaries on both sides, where multiple line segments are adopted to approximate the shape of the boundary to satisfy the applicability to a curved road, as shown in Fig. 3. The road boundary constraint for the vehicle's location at each time step is expressed by:

$$\begin{cases} \frac{\kappa_t^{up} x_t + b_t^{up} - y_t}{\sqrt{1 + (\kappa_t^{up})^2}} - \Lambda > 0 \\ \frac{\kappa_t^{lo} x_t + b_t^{lo} - y_t}{\sqrt{1 + (\kappa_t^{lo})^2}} + \Lambda < 0 \end{cases} \quad \forall t \in \{t_{sta}, \dots, t_{plan}\} \quad (14)$$

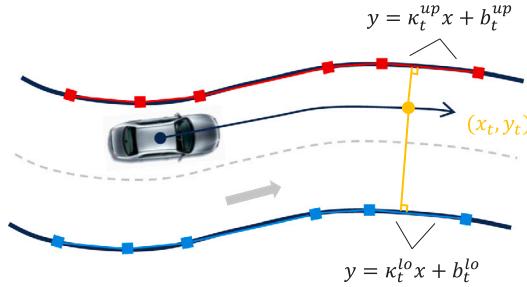


Fig. 3. Road boundary constraint.

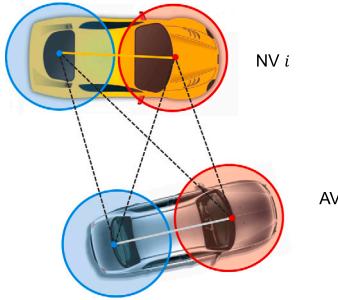


Fig. 4. Double circle collision model.

where  $\kappa_t^{up}$  and  $b_t^{up}$  stand for the slope and intercept of the upper boundary segment corresponding to the planned position at time step  $t$ , and  $\kappa_t^{lo}$  and  $b_t^{lo}$  are those of the lower boundary segment. A distance constant,  $\Lambda$ , is added to buffer the space between the vehicle and the road boundaries, accounting for its size and orientation.

### 3.2. Hierarchical uncertainty-aware safety constraints

In this subsection, we first outline the safety criteria upon which we develop uncertainty-aware chance constraints to address the hierarchical motion uncertainties from NVs' intention and trajectory, employing either an expected or a robust scheme. Furthermore, we introduce a performance measure for assessing model uncertainty, which serves as a weight to adaptively adopt the expected and robust schemes, aiming to dynamically balance safety and travel efficiency in AV navigation.

#### 3.2.1. Definition of safety criteria

We first present the basic safety criteria, which essentially preserve a safe distance between two moving vehicles. For modeling purposes, the shapes of both AV and NVs are approximated via a double circle model illustrated in Fig. 4. The front circle and rear circle with the same radius are located at the front axle's center and rear axle's center respectively. The model serves as a simple and effective approach to represent the geometry of a normal car. For larger vehicles, such as buses or trucks, the approach can be extended by using additional circles with co-linear centers to capture their shapes. However, these details are not addressed in this work, as they fall outside the scope of our focus. Our safety criteria are expressed by:

$$\left\{ \begin{array}{l} ((F_x)_t - (\hat{F}_x)_t^i)^2 + ((F_y)_t - (\hat{F}_y)_t^i)^2 - s_{safe}^2 > 0 \\ ((F_x)_t - (\hat{B}_x)_t^i)^2 + ((F_y)_t - (\hat{B}_y)_t^i)^2 - s_{safe}^2 > 0 \\ ((B_x)_t - (\hat{F}_x)_t^i)^2 + ((B_y)_t - (\hat{F}_y)_t^i)^2 - s_{safe}^2 > 0 \\ ((B_x)_t - (\hat{B}_x)_t^i)^2 + ((B_y)_t - (\hat{B}_y)_t^i)^2 - s_{safe}^2 > 0 \end{array} \right. \quad \forall t \in \{t_{sta}, \dots, t_{plan}\}, i \in \{1, \dots, |M|\} \quad (15)$$

Here, we assume that the set of NVs is  $M$ . The coordinates of AV's front circle center and rear circle center are denoted by  $((F_x)_t, (F_y)_t)$  and  $((B_x)_t, (B_y)_t)$ , which can be calculated from AV's body center location, yaw angle, and wheelbase  $L$ :

$$((F_x)_t, (F_y)_t) = \left( x_t + \frac{L}{2} \cos \theta_t, y_t + \frac{L}{2} \sin \theta_t \right), ((B_x)_t, (B_y)_t) = \left( x_t - \frac{L}{2} \cos \theta_t, y_t - \frac{L}{2} \sin \theta_t \right).$$

For NV  $i \in M$ , its front circle and rear circle centers  $((\hat{F}_x)_t^i, (\hat{F}_y)_t^i)$  and  $((\hat{B}_x)_t^i, (\hat{B}_y)_t^i)$  are defined similarly, using coordinates derived from behavior prediction. The safety distance  $s_{safe}$  is a given constant, which is required to be no less than the sum of the circle

radii of AV and NV  $i$ . When NV's future behavior can be perfectly and deterministically predicted, Eq. (15) ensures that the AV will not collide with any of its NVs. For simplicity, it can be uniformly expressed by

$$\mathbf{H}_t^{i,\eta} > 0, \quad \eta = 1, \dots, 4,$$

with each  $\mathbf{H}_t^{i,\eta}, \forall \eta$  manifesting a quadratic function of  $x_t, y_t$  and  $x_t^i, y_t^i$ , coordinates for both AV and NV  $i$ .

### 3.2.2. Chance constraints under hierarchical uncertainties

Utilizing the  $\mathbf{H}_t^{i,\eta}$  functions generated above, we first provide an inequality, which guarantees the collision probability between AV and any NVs at each time step can be bounded under an acceptable risk level  $\epsilon$ .

**Proposition 3.1** (Bounded Collision Probability). *The collision probability between an ego AV and an arbitrary NV  $i$  is bounded by a uniform risk level  $\epsilon$  if the following set of constraints are satisfied:*

$$\begin{cases} \frac{\sigma_{\mathbf{H}_t^{i,\eta}}^2}{\mu_{\mathbf{H}_t^{i,\eta}}^2 + \sigma_{\mathbf{H}_t^{i,\eta}}^2} < \epsilon \\ \mu_{\mathbf{H}_t^{i,\eta}} > 0 \end{cases} \quad \forall t \in \{t_{sta}, \dots, t_{plan}\}, i \in \{1, \dots, |M|\}, \eta \in \{1, \dots, 4\}, \quad (16)$$

with  $\mu_{\mathbf{H}_t^{i,\eta}}$  and  $\sigma_{\mathbf{H}_t^{i,\eta}}^2$  being the mean and variance of  $\mathbf{H}_t^{i,\eta}$ , both of which are assumed to be finite.

The proof is given in Appendix B. In the uncertain-aware motion planning, the safety criteria stated in Eq. (15) can be replaced by Eq. (16) to achieve robust safety performance. Two key parameters,  $\mu_{\mathbf{H}_t^{i,\eta}}$  and  $\sigma_{\mathbf{H}_t^{i,\eta}}^2$ , can be approximated by moments estimations of  $\mathbf{H}_t^{i,\eta}$ , denoted as  $\mathbb{E}(\cdot^\alpha)$  with  $\alpha = 1, 2$ , respectively. Depending on the degrees that motion and model uncertainties are utilized, three estimation approaches are proposed as follows.

**Definition 3.1** (The Expected Scheme). If  $\mu_{\mathbf{H}_t^{i,\eta}}$  and  $\sigma_{\mathbf{H}_t^{i,\eta}}^2$  in constraint (16) are estimated by

$$\mathbb{E}^{exp}((\mathbf{H}_t^{i,\eta})^\alpha) \equiv \sum_{I \in \{LK, LCL, LCR\}} \hat{P}_{r_I}^i \cdot \mathbb{E}((\mathbf{H}_t^{i,\eta})^\alpha | I), \quad \alpha = 1, 2, \quad (17)$$

the motion planning with uncertainty-awareness is said to follow the **expected scheme**.

The expected scheme is similar to the one adopted by Wang et al. (2020), which accounts for all conceivable intentions of NVs' behaviors, preventing AV motion planning from being over-conservative owing to the unlikely NV intentions. However, it may fail to ensure safety in certain edge cases when a non-negligible inconsistency between the predicted and the real intention distributions occur. Hence, a conservative moments estimation is introduced as the second scheme.

**Definition 3.2** (The Robust Scheme). Let  $lane_t^{AV}$  and  $lane_t^i$  represent the lane indexes where AV and NV  $i$  locate at time-step  $t$ , respectively, and the binary indicator  $I^{wst}(lane_t^{AV}, lane_t^i)$  denote the intention that produces the worst case in which an NV is driven towards occupying the AV's lane. If  $\mu_{\mathbf{H}_t^{i,\eta}}$  and  $\sigma_{\mathbf{H}_t^{i,\eta}}^2$  in constraint (16) are estimated by

$$\mathbb{E}^{rhs}((\mathbf{H}_t^{i,\eta})^\alpha) \equiv \mathbb{E}((\mathbf{H}_t^{i,\eta})^\alpha | I^{wst}(lane_t^{AV}, lane_t^i)), \quad \alpha = 1, 2. \quad (18)$$

then the motion planning with uncertainty-awareness is said to follow the **robust scheme**.

Compared to the expected scheme, the robust scheme can guarantee safe motion planning to the largest degree, but it also can be excessively conservative since the AV's motion would always be restricted by the future intention that induces close vehicle interactions. However, both schemes do not capture model uncertainty, whose effects and a countermeasure will be elaborated in the next subsection.

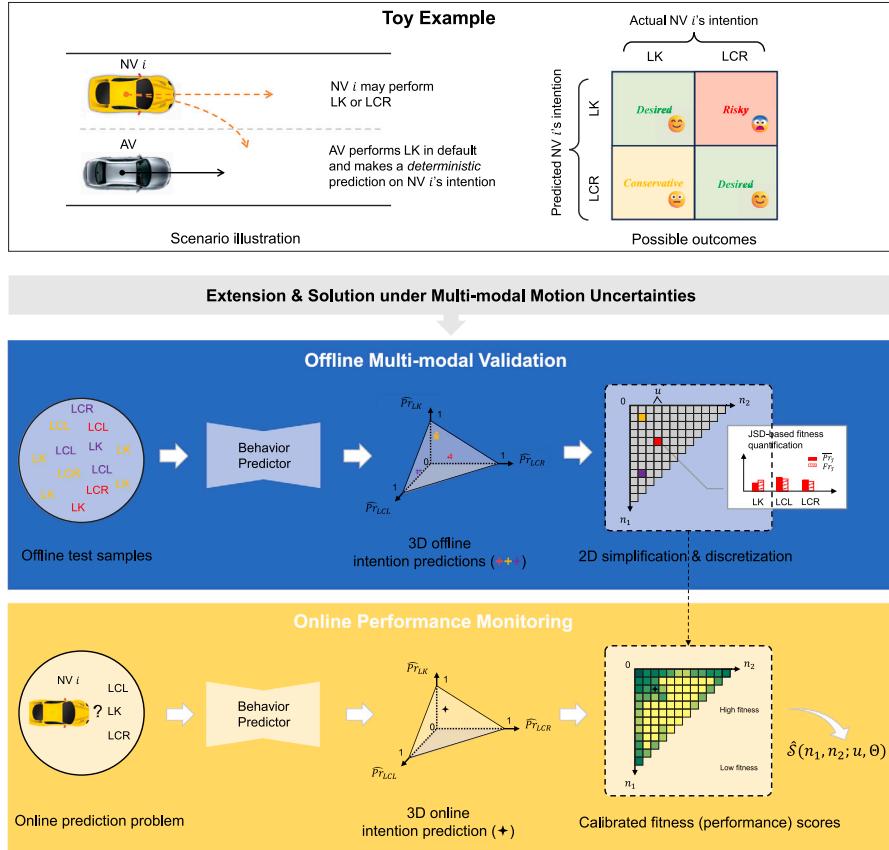
### 3.2.3. The adaptive treatment of model uncertainty

Model uncertainty can have two types of effects on motion planning. Illustrating with the example in Fig. 5, where an AV performs behavior prediction to determine if an NV in the adjacent lane will remain in its current lane, two errors occur: *false positives*, where the predicted intention is lane-keeping (LK) but the actual action is lane-changing to the right (LCR), and *false negatives*, where the prediction is LCR but the actual intention is LK. The case of false positives would result in risky scenarios such as collisions or near-collisions, and the case of false negatives may lead to overly cautious deceleration of the AV and possible slow-down of upstream traffic. Without addressing model uncertainty, the case of false positives are likely to appear under the expected scheme, while the robust scheme leans towards to create cases of false negatives.

In this regard, the third scheme, namely adaptive scheme, is proposed. It utilized a normalized performance score to assess the goodness of fit between predicted and actual intention probabilities and dynamically balances the preference over the expected and robust schemes, such that a high score leans towards the expected scheme, while a lower one weighs more on the robust scheme.

**Definition 3.3** (The Adaptive Scheme). Let  $\hat{\mathcal{V}}^i = (\hat{P}_{r_{LK}}^i, \hat{P}_{r_{LCL}}^i, \hat{P}_{r_{LCR}}^i)$  and  $\mathcal{V}^i = (Pr_{r_{LK}}^i, Pr_{r_{LCL}}^i, Pr_{r_{LCR}}^i)$  be the predicted and actual intention probabilities, and  $S(\hat{\mathcal{V}}^i, \mathcal{V}^i)$  the normalized performance score. If  $\mu_{\mathbf{H}_t^{i,\eta}}$  and  $\sigma_{\mathbf{H}_t^{i,\eta}}^2$  in constraint (16) are estimated by

$$\mathbb{E}^{ada}((\mathbf{H}_t^{i,\eta})^\alpha) = S(\hat{\mathcal{V}}^i, \mathcal{V}^i) \cdot \mathbb{E}^{exp}((\mathbf{H}_t^{i,\eta})^\alpha) + (1 - S(\hat{\mathcal{V}}^i, \mathcal{V}^i)) \cdot \mathbb{E}^{rhs}((\mathbf{H}_t^{i,\eta})^\alpha), \quad \alpha = 1, 2, \quad (19)$$



**Fig. 5.** Illustration of adaptive scheme.

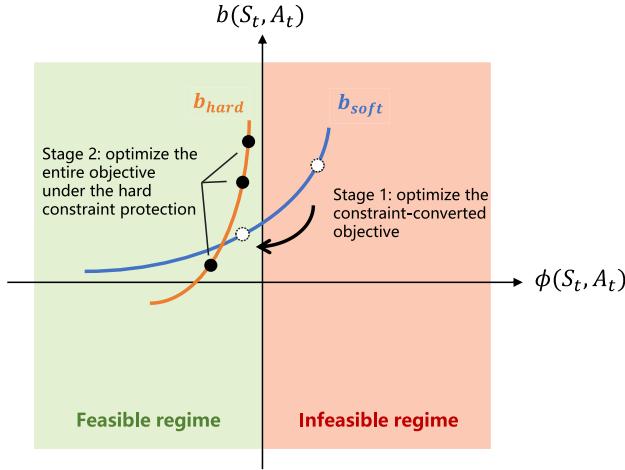
then the motion planning with uncertainty-awareness is said to follow the **adaptive scheme**.

The normalized performance score  $S$  can be constructed by Jensen–Shannon divergence (Lin, 1991), which evaluates the similarity between  $\hat{\mathcal{V}}^i$  and  $\mathcal{V}^i$ ,  $\forall i \in \{1, \dots, |M|\}$ . Though conceptually straightforward, the practical application can be challenging since it is near impossible to observe the actual intention probabilities  $\mathcal{V}^i$  for online fitness checking. The limitation persists even in an *ex post* manner (Fridovich-Keil et al., 2020), since only a single maneuver is taken and observable in a real-world event.

To address this challenge, we propose an online performance monitoring method supported by offline multi-modal validation, whose procedure is shown in Fig. 5. The method begins with offline multi-modal validation shown in the blue block, where the behavior prediction model is applied to offline test samples. The three-dimensional are then grouped into a finite number of categories, visualized as the two-dimensional matrix on the right. For each category, the frequencies of offline samples' actual intentions are calculated. Subsequently, the comparisons between predicted and actual intentions are stored as JS divergence-based performance scores, which can be easily accessed during online performance monitoring, depicted in the yellow block, whenever the online motion planning module generates a prediction. For further mathematical details, readers are encouraged to refer to Appendix C.

#### 4. Online solution via two-stage CILQR

Motion planning with hierarchical uncertainty-aware safety constraints fundamentally constitutes a non-convex, nonlinear optimal control problem. Directly applying nonlinear programming solvers (Wächter and Biegler, 2006) can suffer from excessive computation time due to difficulties in satisfying the non-convex constraints (Liu et al., 2017; Han et al., 2023). In light of this, we employ CILQR for flexibly integrating various constraints and yielding an effective and efficient solution. Building on the foundational Linear Quadratic Regulator (LQR), which solves optimal control problems with quadratic costs and linear dynamics, CILQR is a control policy that adapts to the nonlinear and non-convex constraints rising from vehicle dynamics and safety criteria (Chen et al., 2017, 2019). Here, we first outline the standard control policy, the Iterative Linear Quadratic Regulator (ILQR), which is compatible with our discrete dynamic system setting. We then introduce our CILQR policy, which is extended from the previous one to expedite the solution procedure for our problem.



**Fig. 6.** Illustration of the two-stage optimization.

Let  $S_{t+1} = h(S_t, A_t)$  stand for the dynamic transition of the vehicle state described in Eq. (12). The Bellman equation is expressed as

$$V_t(S_t) = \min_{A_t} [Q(S_t, A_t)] = \min_{A_t} [C(S_t, A_t) + V_{t+1}(h(S_t, A_t))] \quad (20a)$$

$$V_{t_{plan}}(S_{t_{plan}}) = C(S_{t_{plan}}) \quad (20b)$$

where  $V_t(\cdot)$  represents the value function at time-step  $t$ , and  $C(S_t, A_t)$  stands for the direct cost at the time-step  $t$ , whose formulation can be traced from the objective stated in Eq. (11). At the final state, the value function equals the cost. For simplicity, we use  $\phi(S_t, A_t) \leq 0$  to conclude the optimization constraints stated in Eqs. (12)–(13) and (16). The standard ILQR (Tassa et al., 2012) does not consider the inequality constraints. It initiates the solution process with a nominal trajectory  $(\bar{S}_{t_{sta}:t_{plan}}, \bar{A}_{t_{sta}:(t_{plan}-1)})$  through backward induction. The second-order derivative is used to find a better trajectory that produces a lower value. The details of standard ILQR are outlined in Appendix D.

When the set of inequality constraints  $\phi(S_t, A_t) \leq 0$  is introduced to the dynamic program, CILQR suggests that transforming the constraints into cost terms through function composition with a logarithmic barrier function, that is,  $-\log(-\phi(S_t, A_t))$ . This method, which has been employed in previous studies on motion planning such as Chen et al. (2019), transforms the problem into constraint-free and is sufficient to deal with most situations where deterministic predictions are applied. Nonetheless, as uncertainty-aware planning imposes a more complex constrained searching space, the nominal trajectory for initialization may fail to generate feasible solutions under the traditional method.

In light of this, we propose a two-stage extended CILQR policy that integrates two separate barrier functions. These functions preserve the flexibility in choosing initial trajectory while guaranteeing the feasibility of the optimal solution. The core concept of the policy is illustrated in Fig. 6, with its procedure being elaborated as follows:

#### Step 1: Nominal trajectory acquisition

To start the planning algorithm, we provide an initial control sequence composed of  $(t_{plan} - t_{sta})$  successive actions, according to some heuristics. The nominal trajectory is then calculated by performing a forward simulation in which the vehicle's current state and initial control sequence are substituted into the vehicle kinematic model defined in Eq. (12).

#### Step 2: Feasibility query

We check the feasibility of the nominal trajectory. If it strictly follows all the inequality constraints defined, the algorithm jumps to Step 4; otherwise, the infeasible nominal trajectory is sent to Step 3.

#### Step 3: Soft CILQR loop

As the nominal trajectory cannot satisfy all constraints, we perform a Soft CILQR loop aiming at altering the infeasible trajectory into a feasible one as demonstrated in Fig. 6. Every inequality constraint is converted into a cost term via an exponential barrier function:

$$b_{soft}(S_t, A_t) = \sum_{j=1}^n q_1 \cdot \exp(q_2 \cdot \phi_j(S_t, A_t)) \quad (21)$$

with  $q_1$  and  $q_2$  being the tunable parameters, and  $n$  being the number of inequality constraints. Subsequently, we replace the cost  $C(S_t, A_t)$  in the Bellman equation by  $b_{soft}(S_t, A_t)$ , and solve the revised dynamic program by the standard ILQR. Once a feasible trajectory is found, the Soft CILQR loop terminates, and the algorithm goes to Step 4. Step 3 is referred to as the *first-stage optimization problem*.

#### Step 4: Hard CILQR loop

Starting with the strictly feasible nominal trajectory, the remaining issue is to further reduce the overall cost in the feasible region. We then march into the *the second-stage optimization problem*, where the direct cost  $C(S_t, A_t)$  in Bellman equation incorporates cost terms in the objective defined in Eq. (11) and the logarithm barrier function

$$b_{hard}(S_t, A_t) = -\frac{1}{v} \cdot \sum_{j=1}^n \log(-\phi_j(S_t, A_t)) \quad (22)$$

In this function, the parameter  $v$  can be dynamically increased to accelerate the convergence, where a detailed explanation is given by Chen et al. (2019). The second-stage optimization problem is then solved by a standard ILQR, which guarantees that all inequality constraints will not be violated (Shimizu et al., 2020).

**Remark 4.1.** It is worth noting that the traditional ILQR implements the optimized trajectory over the entire time window  $[t_{sta}, t_{plan}]$ . However, the proposed method can also be adapted to the model predictive control (MPC) framework, where only the optimized trajectory from  $t_{sta}$  to  $t_{sta} + n$ , with  $0 < n < t_{plan} - t_{sta}$ , is implemented. This allows for continuous updates to account for the AV's possible interactions with NVs. Subsequently, behavioral prediction of NVs and motion planning of AVs from  $t_{sta} + n + 1$  to  $t_{sta} + n + t_{plan} - t_{sta}$  is performed, with only the trajectory from  $t_{sta} + n + 1$  to  $t_{sta} + 2n + 1$  is implemented. This procedure is conducted iteratively, until the entire motion planning task is completed. Integrating MPC with CILQR can offer more flexibility and potentially enhance safety in AV motion planning, though it may introduce additional computational burden.

In the end, the solution procedure is summarized in the Algorithm 1:

---

#### Algorithm 1 The Two-stage Constrained Iterative LQR Algorithm

---

- 1: **Given:** current state:  $S(0)$ ; heuristic control:  $A_{t_{sta}:(t_{plan}-1)}^h$ ; barrier function parameters:  $q_1 > 0, q_2 > 0, v_0 > 0, \mu > 1$ ; regularization related parameters:  $\lambda_0 > 0, \lambda_{max} > 0, \Delta_1 < 1, \Delta_2 > 1$ .
- 2: **Return:** the optimized trajectory:  $(S_{t_{sta}:t_{plan}}^*, A_{t_{sta}:(t_{plan}-1)}^*)$ .
- 3:
- 4: ——Start of Nominal Trajectory Acquisition——
- 5: Compute the state sequence  $S_{t_{sta}:t_{plan}}^h$  using Eq. (12) with  $A_{t_{sta}:(t_{plan}-1)}^h$  and  $S(0)$ .
- 6: Obtain a nominal trajectory:  $\tilde{S}_{t_{sta}:t_{plan}} := S_{t_{sta}:t_{plan}}^h, \tilde{A}_{t_{sta}:(t_{plan}-1)} := A_{t_{sta}:(t_{plan}-1)}^h$ .
- 7: ——End of Nominal Trajectory Acquisition——
- 8:
- 9: ——Start of Feasibility Query——
- 10: if  $(\tilde{S}_{t_{sta}:t_{plan}}, \tilde{A}_{t_{sta}:(t_{plan}-1)})$  is strictly feasible then
- 11: Call Hard CILQR Loop.
- 12: else
- 13: Call Soft CILQR loop.
- 14: end if
- 15: ——End of Feasibility Query——
- 16:
- 17: ——Start of Soft CILQR Loop——
- 18: Initialize variables:  $S_{t_{sta}:t_{plan}} := \tilde{S}_{t_{sta}:t_{plan}}, A_{t_{sta}:(t_{plan}-1)} := \tilde{A}_{t_{sta}:(t_{plan}-1)}$ .
- 19: Initialize the parameter:  $\lambda := \lambda_0$ .
- 20: Convert inequality constraints using exponential barrier function:  $\phi(S_t, A_t) < 0 \rightarrow b_{soft}(S_t, A_t) := \sum_{j=1}^n q_1 \cdot \exp(q_2 \cdot \phi_j(S_t, A_t))$ .
- 21: Compute the cost that comprises all constraint-converted terms:  $B := b_{soft}(S_t, A_t)$ .
- 22: while  $(S_{t_{sta}:t_{plan}}, A_{t_{sta}:(t_{plan}-1)})$  is not feasible do
- 23: Compute  $(\tilde{S}_{t_{sta}:t_{plan}}, \tilde{A}_{t_{sta}:(t_{plan}-1)})$  along with the cost  $\tilde{B}$  using an ILQR iteration with the regularization parameter  $\lambda$  (see Appendix The Standard ILQR for details).
- 24: if  $\tilde{B} < B$  then
- 25: Update variables:  $S_{t_{sta}:t_{plan}} := \tilde{S}_{t_{sta}:t_{plan}}, A_{t_{sta}:(t_{plan}-1)} := \tilde{A}_{t_{sta}:(t_{plan}-1)}$
- 26: Update the cost:  $B := \tilde{B}$
- 27: Decrease the regularization parameter:  $\lambda := \Delta_1 \lambda$
- 28: else
- 29: Increase the regularization parameter:  $\lambda := \Delta_2 \lambda$
- 30: if  $\lambda > \lambda_{max}$  then
- 31: Break. {Limit of exploration reached.}
- 32: end if
- 33: end if
- 34: end while
- 35: Obtain a feasible nominal trajectory:  $\tilde{S}_{t_{sta}:t_{plan}} := S_{t_{sta}:t_{plan}}, \tilde{A}_{t_{sta}:(t_{plan}-1)} := A_{t_{sta}:(t_{plan}-1)}$ .
- 36: ——End of Soft CILQR Loop——
- 37:

```

38: ——Start of Hard CILQR Loop——
39: Initialize variables:  $S_{t_{sta}:t_{plan}} := \bar{S}_{t_{sta}:t_{plan}}$ ,  $A_{t_{sta}:(t_{plan}-1)} := \bar{A}_{t_{sta}:(t_{plan}-1)}$ 
40: Initialize parameters:  $\nu := v_0$ ,  $\lambda := \lambda_0$ .
41: Compute the original cost  $C$  according to Eq. (11).
42: Convert inequality constraints using logarithmic barrier function:  $\phi(S_t, A_t) < 0 \rightarrow b_{hard}(S_t, A_t) := -\frac{1}{\nu} \cdot \sum_{j=1}^n \log(-\phi_j(S_t, A_t))$ .
43: Add constraints-converted terms to the original cost:  $C_+ := C + b_{hard}(S_t, A_t)$ .
44: while  $C_+$  is not convergent do
45:   while  $C_+$  is not convergent do
46:     Compute  $(\tilde{S}_{t_{sta}:t_{plan}}, \tilde{A}_{t_{sta}:(t_{plan}-1)})$  along with the cost  $\tilde{C}_+$  using an ILQR iteration with the regularization parameter  $\lambda$  (see
      Appendix The Standard ILQR for details).
47:     if  $\tilde{C}_+ < C_+$  then
48:       Update variables:  $S_{t_{sta}:t_{plan}} := \tilde{S}_{t_{sta}:t_{plan}}$ ,  $A_{t_{sta}:(t_{plan}-1)} := \tilde{A}_{t_{sta}:(t_{plan}-1)}$ 
49:       Update the cost:  $C_+ := \tilde{C}_+$ 
50:       Decrease the regularization parameter:  $\lambda := \Delta_1 \lambda$ 
51:     else
52:       Increase the regularization parameter:  $\lambda := \Delta_2 \lambda$ 
53:       if  $\lambda > \lambda_{max}$  then
54:         Break. {Limit of exploration reached.}
55:       end if
56:     end if
57:   end while
58:   Increase the outer loop parameter:  $\nu := \mu \nu$ 
59: end while
60: Obtain an optimized trajectory:  $S_{t_{sta}:t_{plan}}^* := S_{t_{sta}:t_{plan}}$ ,  $A_{t_{sta}:(t_{plan}-1)}^* := A_{t_{sta}:(t_{plan}-1)}$ .
61: ——End of Hard CILQR Loop——

```

## 5. Experiments

### 5.1. Data preparation

To train and evaluate our approach, we extract data from the open-source INTERACTION ([Zhan et al., 2019](#)) and HighD ([Krajewski et al., 2018](#)) datasets. As illustrated in [Fig. 7](#), a two-lane merging section, *DR\_DEU\_Merging\_MT*, is selected from INTERACTION dataset to investigate mandatory lane-changing behaviors, where only two kinds of intention need to be considered due to the leftward forced merging direction. In total, 10,000 training samples and 6796 test samples are extracted from the scenario. Regarding the HighD dataset, we extract data from files no.1 to no.20, mainly to explore discretionary lane changes on multi-lane highways. A total of 280,704 samples have been collected and randomly divided into two halves for training and evaluation.

Regarding the selection of neighboring vehicles, existing studies usually adopted a standard related to distance ([Gao et al., 2020](#)) or topology ([Xie et al., 2019](#)) to finish the procedures. Following their definitions, the specific standard used for identifying neighboring relations in our data processing is:

We categorize a vehicle as an NV on condition that:

- The vehicle is located at the same lane or adjacent lanes of the AV's;
- The vehicle is deemed alongside the AV. Otherwise, it serves as the vehicle that is longitudinally closest to the AV in the preceding or following direction at its lane;
- The vehicle's longitudinal distance to the AV is less than a predefined threshold according to the requirement of motion planning. The threshold is empirically set as 15 m in this study.

We categorize a vehicle as an NNV on condition that:

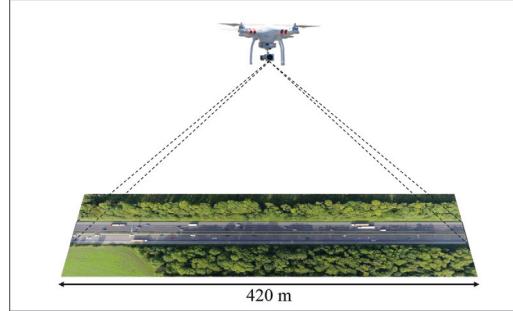
- The vehicle is located at the same lane or adjacent lanes of the NV's;
- The vehicle is deemed alongside the NV. Otherwise, it serves as the vehicle that is longitudinally closest to the NV in the preceding or following direction at its lane;
- The vehicle's longitudinal distance to the NV is less than a predefined threshold according to the requirement of behavior prediction. The threshold is empirically given by 25 m in INTERACTION and follows the original setting in HighD.<sup>2</sup>

With the above standard, a maximum of 8 vehicles may be identified as NVs for the AV, or NNVs for an NV. This closely follows the 8-neighbor definition by [Krajewski et al. \(2018\)](#), encompassing possible roles of: 1. preceding vehicle, 2. following vehicle, 3. left alongside vehicle, 4. left preceding vehicle, 5. left following vehicle, 6. right alongside vehicle, 7. right preceding vehicle, and 8. right following vehicle.

<sup>2</sup> The distance threshold for considering NNVs is relatively larger, which has the DL model learn to scale across diverse scenarios. The HighD dataset provides original labels of neighboring relations.



Two-lane merging section of INTERACTION dataset  
(Zhan et al., 2019)



Multi-lane highway of HighD dataset  
(Krajewski et al., 2018)

**Fig. 7.** Illustration of dataset collection sites.

**Table 1**

Model structural hyper-parameter setting.

Component	Layer/Operation	Data format	Shape
Trajectory encoder	LSTM @ $W_{MoEnc}$	num. vehicles × num. observed time-steps × unit input size	(none, 10, 5)
	Output	num. vehicles × LSTM hidden size	(none, 32)
GAT	Attentive layer 1	num. vehicles × layer-1 node input size	(none, 32)
	Multi-head concat.	num. heads × num. vehicles × layer-1 embedding size	(4, none, 16)
	Attentive layer 2	num. vehicles × layer-2 node input size	(none, 64)
	Output	num. vehicles × layer-2 embedding size	(none, 32)
Scene encoder	LSTM @ $W_{ScEnc}$	num. side points × unit input size	(none, 2)
	Hidden state concat.	num. boundaries × num. side points × LSTM hidden size	(2, none, 6)
	Output	scene embedding size	(12)
Intention decoder	Dense layer 1	compound embedding size	(44)
	Dense layer 2	MLP hidden size	(64)
	Output	num. intention classes	(3)
Noise generator	–	noise size	(8)
Trajectory decoder	LSTM @ $W_{MoDec}^T$	num. prediction time-steps × unit input size	(20, 2)
	MLPs @ $W_{traj}^T$	num. prediction time-steps × LSTM hidden size	(20, 52)
	Output (under $T$ )	num. prediction time-steps × coordinates dimension	(20, 2)

## 5.2. Evaluation of behavior prediction

We implement our behavior prediction model using Python 3.10 with PyTorch 2.0.0 on an NVIDIA GeForce RTX3090. The hyper-parameters used in the model are detailed in [Table 1](#), where the observation range of the selected datasets determines the number of vehicles and the number of roadside boundary points. The model processes input from 10 historical time-steps and outputs the predictions over the intention and future trajectories in 20 time-steps. The time-step sizes of INTERACTION dataset and HighD dataset are set to be 0.20 s and 0.24 s, respectively. We first train our Cond-SIGAT-Base using the loss function described in [Eq. \(9\)](#) for 300 epochs with a learning rate of 3e-3 on the INTERACTION training set, and for 60 epochs with a learning rate of 5e-4 on the HighD training set, applying a batch size of 128 in both cases. The final model, either incorporating AoN or MoN loss as per [Eq. \(10\)](#), is further trained for 100 and 20 epochs on the respective training sets to predict 20 trajectory samples, denoted by the suffix “-AoN(20)” or “-MoN(20)”. For this final model training on both the training sets, we set the learning rate as 5e-3 and the batch size as 128.

The effectiveness of our behavior prediction method is assessed through its performance of intention and trajectory prediction on the test datasets. For intention prediction, we label the intention of a vehicle as “lane change” if its future ground-truth in the prediction horizon crosses the lane marking. Depending on the processed time resolutions of the two datasets, the advance time of intention prediction is up to 4 s in INTERACTION and 4.8 s in HighD. The intention prediction performance is evaluated by

**Table 2**  
Lane-changing intention prediction performance.

Dataset	Metric	LSTM-Int	IGAT-Int	SIGAT-Int	Cond-SIGAT
INTERACTION	Accuracy	98.18%	98.64%	<b>99.54%</b>	99.37%
	Precision <sub>LK</sub>	98.78%	99.19%	<b>99.78%</b>	99.54%
	Precision <sub>LCL</sub>	88.72%	90.48%	96.08%	<b>96.87%</b>
	Recall <sub>LK</sub>	99.28%	99.36%	99.73%	<b>99.79%</b>
	Recall <sub>LCL</sub>	82.13%	88.17%	<b>96.75%</b>	93.27%
HighD	Accuracy	79.96%	85.32%	<b>91.59%</b>	91.22%
	Precision <sub>LK</sub>	80.13%	87.51%	92.69%	<b>93.79%</b>
	Precision <sub>LCL</sub>	82.04%	79.85%	<b>88.19%</b>	85.88%
	Precision <sub>LCR</sub>	78.02%	82.32%	<b>90.70%</b>	87.76%
	Recall <sub>LK</sub>	92.30%	89.95%	<b>94.14%</b>	92.38%
	Recall <sub>LCL</sub>	51.71%	73.90%	86.57%	<b>88.10%</b>
	Recall <sub>LCR</sub>	64.33%	80.10%	87.72%	<b>90.06%</b>

typical classification metrics such as accuracy, precision, and recall. These metrics, which are widely adopted in deep-learning-based behavior prediction, are consistent across all versions of Cond-SIGAT, due to the training technique explained in Section 2.3. For evaluating trajectory prediction accuracy, we employ metrics that include minimum Average Displacement Error (minADE) and minimum Final Displacement Error (minFDE), as introduced by Gu et al. (2021). Lower values of these metrics indicate better performance.

The performance of our approach is compared with the following baselines that do not incorporate the intention-branched structure. Instead, these approaches either produce intention or trajectory exclusively, but not both in an integrated manner.

- Prediction on intentions:

- LSTM-Int: a simple network that applies a motion-encoding LSTM to extract NV's historical motion features and an intention-generating MLP to predict lane-changing intentions.
- IGAT-Int: an interaction-aware network applying motion-encoding LSTMs to extract NV's and NNVs' historical motion features and a GAT to model their interactions, with an intention-generating MLP to predict lane-changing intentions.
- SIGAT-Int: a network comprised of our complete traffic feature extractor and an intention-generating MLP, which can capture multiple categories of features to predict lane-changing intentions.

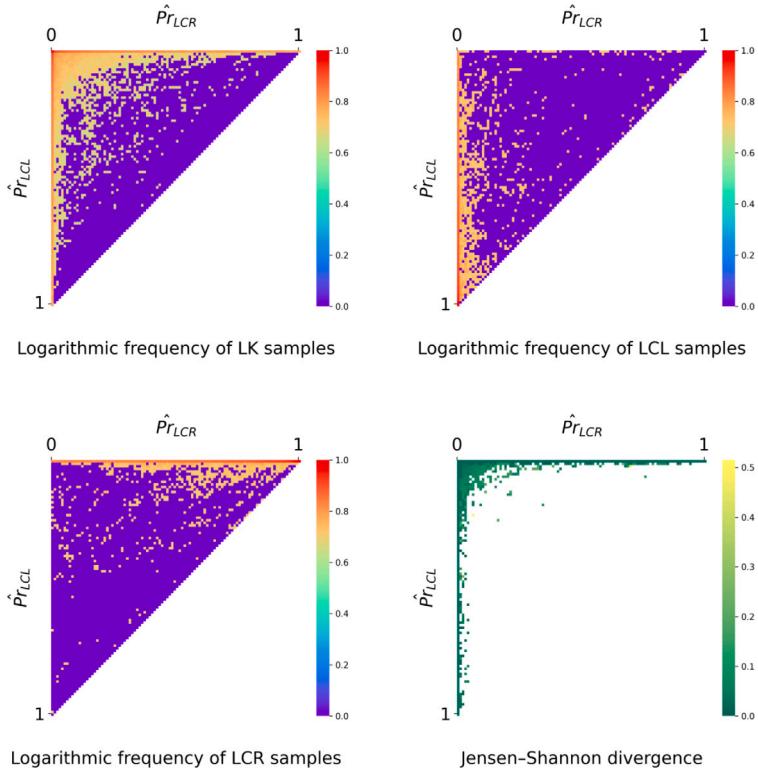
- Prediction on trajectories:

- LSTM-Traj: a seq2seq-based encoder-decoder network, which applies a motion-encoding LSTM to extract the NV's historical motion features and a single LSTM-based decoder for trajectory prediction.
- IGAT-Traj: an interaction-aware network applying motion-encoding LSTMs to extract NV's and NNVs' historical motion features and a GAT to model their interactions, with a single LSTM-based decoder for trajectory prediction.
- SIGAT-Traj: a network comprised of our complete traffic feature extractor and a single LSTM-based decoder, which can capture multiple categories of features for trajectory prediction.

### 5.2.1. Intention prediction performance

Table 2 summarizes the performance of intention prediction. From the comparison of the three baselines which are LSTM-Int, IGAT-Int, and SIGAT-Int, one can see that the integration of more categories of features leads to overall more reliable predictions. SIGAT-Int and Cond-SIGAT show comparable performance, though the accuracy of Cond-SIGAT is marginally lower by 0.17% and 0.37% on the two datasets, respectively. The slight difference arises since Cond-SIGAT is developed as a multi-objective network instead of being dedicated to intention prediction. Moreover, the model shows better performance in predicting mandatory lane changes in the INTERACTION than predicting discretionary lane changes in the HighD dataset. This difference is attributed to two reasons. On one hand, predicting binary intentions is generally easier than predicting multiple intentions. On the other hand, the road boundaries at a forced merging section also provide a clear indication of lane-changing.

To showcase the model uncertainty of behavior prediction and to support the adaptive scheme detailed in Section 3.2.3, we present a visualization of the discretized distribution of samples from HighD test set and their JS divergence in Fig. 8. The discretization resolution is set to 0.01. The first three subplots display the predicted frequencies for LK, LCL, and LCR within each cell, using a color scale where redder hues indicate higher frequencies and purpler hues denote lower ones. Logarithmic scaling is applied to enhance the visibility of frequency differences. These subplots reveal distinct patterns over different intentions: cells with higher frequencies of LCL cluster in the triangle's bottom left, while those with higher frequencies of LCR are found in the upper right, showing strong alignment between the predicted and the actual probabilities. Conversely, cells with a prevalent LK prediction are more scattered, though a concentration trend towards the upper left corner can be detected. The last subplot, excluding cells with fewer than 6 offline samples due to their unverified fitness, depicts JS divergence, indicating improved behavior prediction when one intention's actual probability significantly exceeds the others.



**Fig. 8.** Visualization of lane-changing prediction statistics from HighD dataset.

**Table 3**  
Trajectory prediction performance.

Dataset	Metric	LSTM-Traj	IGAT-Traj	SIGAT-Traj	Cond-SIGAT		
					Base	AoN(20)	MoN(20)
INTERACTION	minADE (m)	0.41	0.32	0.28	0.26	0.24	<b>0.16</b>
	minFDE (m)	1.08	0.73	0.65	0.58	0.50	<b>0.24</b>
HighD	minADE (m)	0.81	0.78	0.74	0.71	0.67	<b>0.63</b>
	minFDE (m)	2.05	1.86	1.77	1.63	1.40	<b>1.01</b>

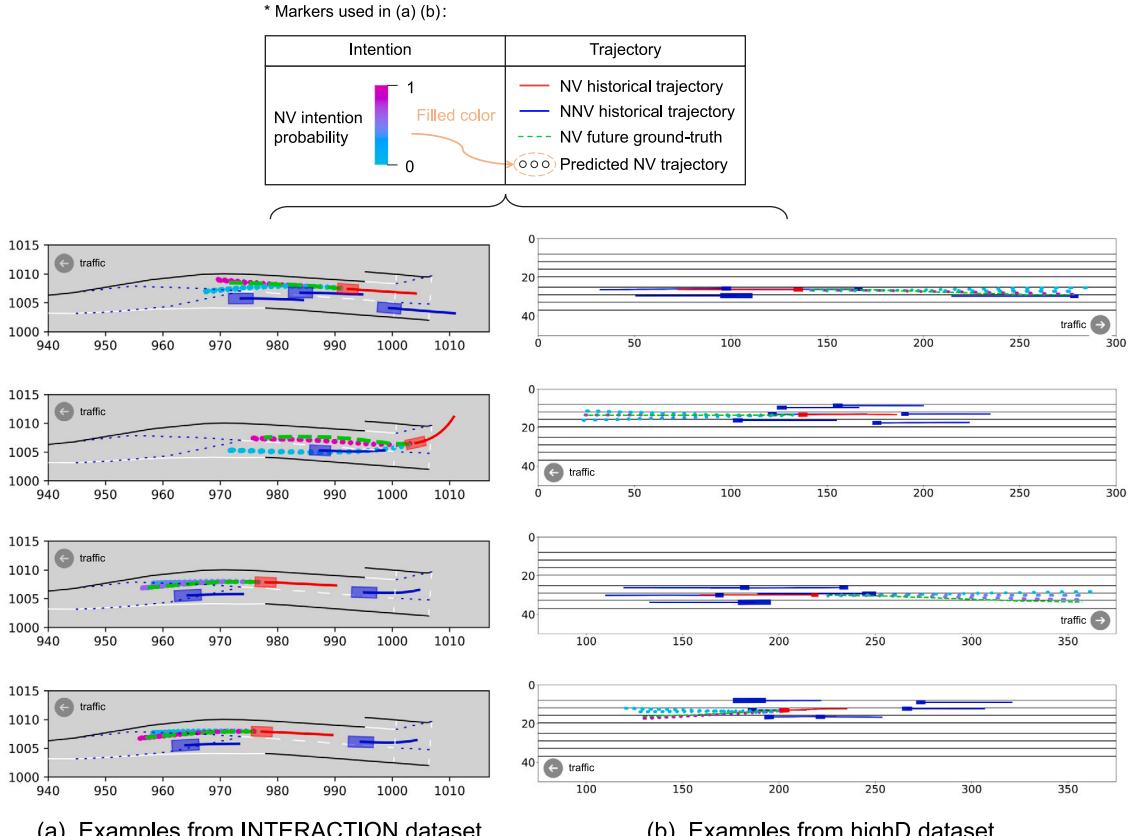
### 5.2.2. Trajectory prediction performance

**Table 3** details the performance regarding the trajectory prediction task. Echoing the findings from intention prediction performance, leveraging multiple traffic features enhances the accuracy of trajectory predictions. Notably, our Cond-SIGAT-Base outperforms all baseline models, despite the complexity of its multi-tasking learning objective. This demonstrates the efficacy of building the intention-branched structure for making multimodal predictions. In Cond-SIGAT-AoN and Cond-SIGAT-MoN with adding noise terms and 20-shot predictions, the minimum errors can be further decreased since the trajectory uncertainty is accounted for. Specifically, our Cond-SIGAT-MoN(20) achieves a minADE lower than 0.2 m and a minFDE lower than 0.25 m in the INTERACTION scenario, showing that the best prediction closely aligns with the ground truth. However, it is also observed that the predictions made under Cond-SIGAT-MoN can be prohibitively diverse owing to the over-optimistic loss function applied in the training. Compared to Cond-SIGAT-MoN(20), Cond-SIGAT-AoN(20) produces a less-varying trajectory under the same intention while preserving the hierarchical multimodal feature through the intention-branched network, better reflecting the property of motion uncertainty. Therefore, it is selected to generate NVs' behaviors for motion planning tasks.

Furthermore, **Fig. 9** depicts representative scenarios where Cond-SIGAT-AoN(20) is applied, highlighting its effective prediction capacity. Predicted NV trajectories are illustrated with dotted, color-coded lines representing different intention probabilities. One can see that our prediction model assigns a higher probability to the ground-truth intention in most cases and is capable of generating realistic trajectory samples in various speed conditions.

### 5.2.3. Analysis of uncertainty related factors

We further investigate the effects of different input factors on the prediction output performance, namely model uncertainty and motion uncertainty. As introduced in Section 2.1, these factors include historical motion ( $h_{t_{obs}}^i$ ), vehicle interaction ( $g_L^i$ ), and traffic



**Fig. 9.** Qualitative results of the behavior prediction.

scene features represented by road boundaries ( $h_{\Gamma}^{up} \parallel h_{\Gamma}^{lo}$ ). To quantify the multi-dimensional uncertainties, entropy of intention prediction  $U_{int}$  and variance of trajectory predictions  $U_{traj}$  are utilized:

$$U_{int}^i = - \sum_{I \in \{LK, LCL, LCR\}} P_I^i \cdot \log \hat{P}_I^i$$

$$U_{traj}^i = \frac{1}{3N(t_{pred} - t_{obs})} \sum_{I \in \{LK, LCL, LCR\}} \sum_{\chi=1}^N \sum_{t=t_{obs}+1}^{t_{pred}} \left\| \hat{Y}_t^{i,I,\chi} - \bar{\hat{Y}}_t^{i,I} \right\|_2^2$$

Here,  $\hat{P}_I^i$  and  $\hat{Y}_t^{i,I,\chi}$  are as defined in Section 2.2, with  $\bar{\hat{Y}}_t^{i,I}$  representing the average of  $N$  samples of  $\hat{Y}_t^{i,I,\chi}$ . In addition, JS-divergence from Eq. (C.6) is employed to capture the model uncertainty for any prediction.

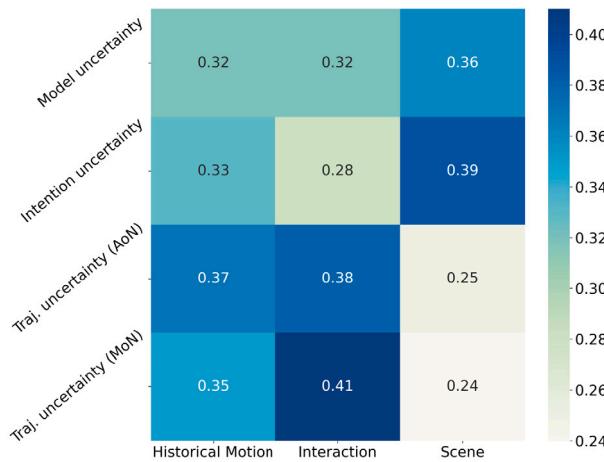
To quantitatively link uncertainty with input factors, we introduce a Relative Effect (RE) metric, extending the classical Mutual Information (MI) measure (Shannon, 1948). This approach quantifies the impact of any input factor on uncertainty measures through normalization. Mathematically, MI and RE for this study are defined as

$$\text{MI}(X_a; Y) = \sum_{x \in X_a} \sum_{y \in Y} \mathbb{P}(x, y) \log \left( \frac{\mathbb{P}(x, y)}{\mathbb{P}(x)\mathbb{P}(y)} \right), \forall X_a \in F$$

$$\text{RE}(X_a; Y) = \frac{1}{\sum_{X_b \in F} \frac{\text{MI}(X_b; Y)}{\min\{\text{h}(X_b), \text{h}(Y)\}}} \cdot \frac{\text{MI}(X_a; Y)}{\min\{\text{h}(X_a), \text{h}(Y)\}}, \forall X_a \in F$$

Here,  $X_a \in F$  represents the embedding vector of an input factor, and  $F$  is the set of all factors. The set  $Y$  corresponds to a type of uncertainty quantification, either intention, trajectory, or model. Parameters  $\text{h}(X_a)$  and  $\text{h}(Y)$  represent the entropies of  $X_a$  and  $Y$ , where  $\min\{\text{h}(X_a), \text{h}(Y)\} \geq \text{MI}(X_a; Y)$  by definition. To mitigate the curse of dimensionality, the normalized MI,  $\frac{\text{MI}(X_a; Y)}{\min\{\text{h}(X_a), \text{h}(Y)\}}$ , is approximated by pairwise processing each element in the embedding and the uncertainty score before taking the average.

The results from HighD experiments are shown in Fig. 10. The REs of different factors display similar patterns across model and intention uncertainties, possibly because changes in model uncertainty often coincide with those in intention uncertainty, as also supported by Fig. 8. The REs of traffic scene futures, 0.36 for model uncertainty and 0.39 for intention uncertainty, are



**Fig. 10.** The Relative Effects (REs) of different factors on uncertainty.

**Table 4**  
Real-time efficiency of prediction models.

Model	Average inference time (ms)	
	INTERACTION	HighD
LSTM-Int	1.22	1.12
IGAT-Int	3.18	3.25
SIGAT-Int	3.60	3.40
LSTM-Traj	3.12	2.83
IGAT-Traj	4.81	4.85
SIGAT-Traj	5.28	5.34
Cond-SIGAT-Base	8.97	8.83
Cond-SIGAT-AoN(20)	10.17	9.89
Cond-SIGAT-MoN(20)	10.03	10.11

relatively higher than other factors, likely due to the boundary restrictions limit a vehicle's possible intentions when it is in a side lane compared to other locations. Regarding trajectory uncertainty under AoN training (see Eq. (10a)), higher REs of 0.37 and 0.38 are observed for historical motion and vehicle interaction, suggesting that the trajectory uncertainty is more closely related to the movements of NVs and NNVs. When MoN loss (Eq. (10b)) is applied, the RE of interaction increases further to 0.41, also indirectly corroborating previous findings that MoN loss captures more diverse interaction patterns (Gupta et al., 2018). Overall, the contributions of different factors to uncertainty show only slight variations, highlighting the importance of synthesizing multiple factors when modeling randomness in behavior prediction.

#### 5.2.4. Computational efficiency of prediction models

The computational efficiency of behavior prediction models is evaluated with the average inference time regarding making prediction for each NV in the test sets, as shown in Table 4. It can be seen that with the model capacity enhanced from different aspects, additional computational costs are required. However, the average inference time used by our final models, Cond-SIGAT-AoN(20) and Cond-SIGAT-MoN(20), is only around 10 ms, illustrating their commendable real-time performance. This is because our models adopt a lightweight architecture while incorporating essential functionalities. Furthermore, since the trajectory sampling process is implemented in batches on the GPU device, the final models exhibit only a marginal increase in computational time relative to Cond-SIGAT-Base.

#### 5.3. Evaluation of motion planning

To evaluate the performance of our motion planners, we conduct two studies using samples from INTERACTION and HighD datasets, respectively. The studies investigate the planner's performance in dealing with both mandatory and discretionary lane changes, encompassing conditions for adopting typical responding strategies such as yielding while keeping lane, courtesy lane change (Knoop et al., 2018; Ali et al., 2020), which serve as representative examples for observing our planner's availability in addressing prediction uncertainties during lane changes. Similar to existing works (Li et al., 2021; Zhou et al., 2023), a virtual AV is spawn in each test using the state of a human-driving vehicle, whose motion is then controlled by the AV motion planner to finish the closed-loop evaluation. We set the planning horizon to be 40 time-steps, with a time-step interval of 0.10 s for INTERACTION and 0.12 s for HighD. A linear interpolation is performed to augment the frequency of predicted trajectory points to satisfy the planning requirement.

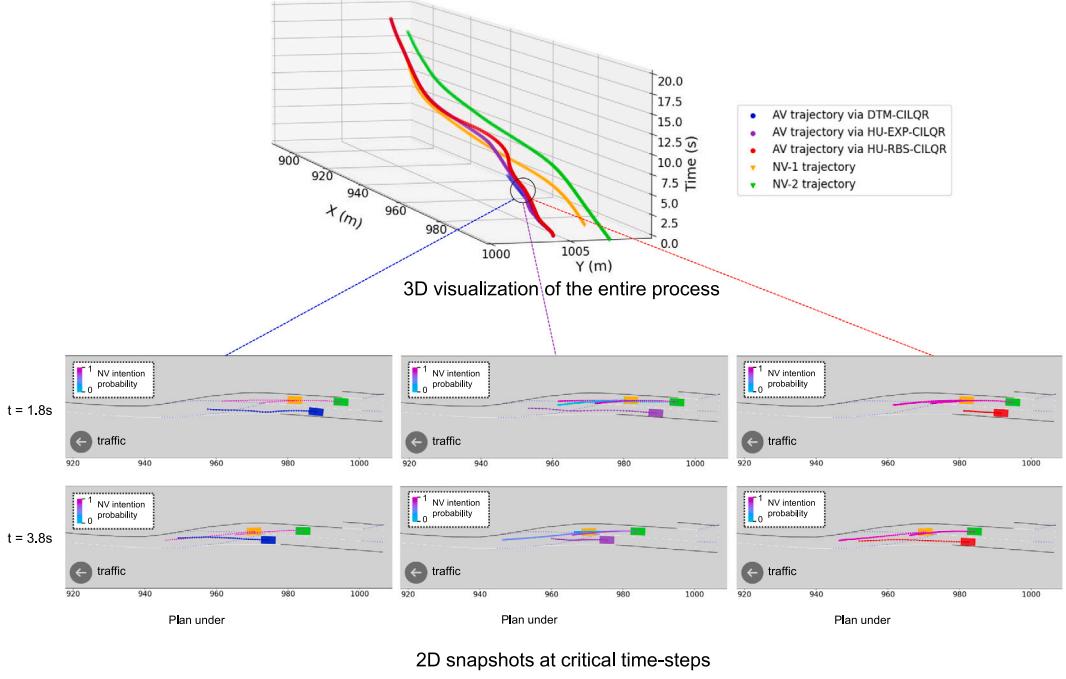


Fig. 11. Trajectory visualization of Case 1.

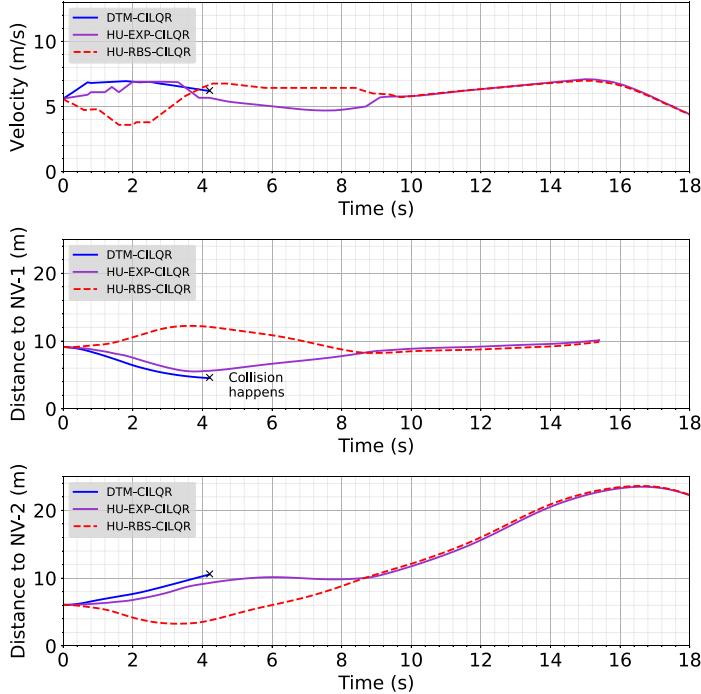


Fig. 12. Illustration of the AV's velocity and distance relations of Case 1.

### 5.3.1. Case 1-response to mandatory lane change

Case 1 is chosen from INTERACTION dataset to analyze the motion planner's capacity in response to mandatory lane changes. Two NVs attempting to change from the merging lane to the main lane are engaged in the case, which will inevitably affect the AV's

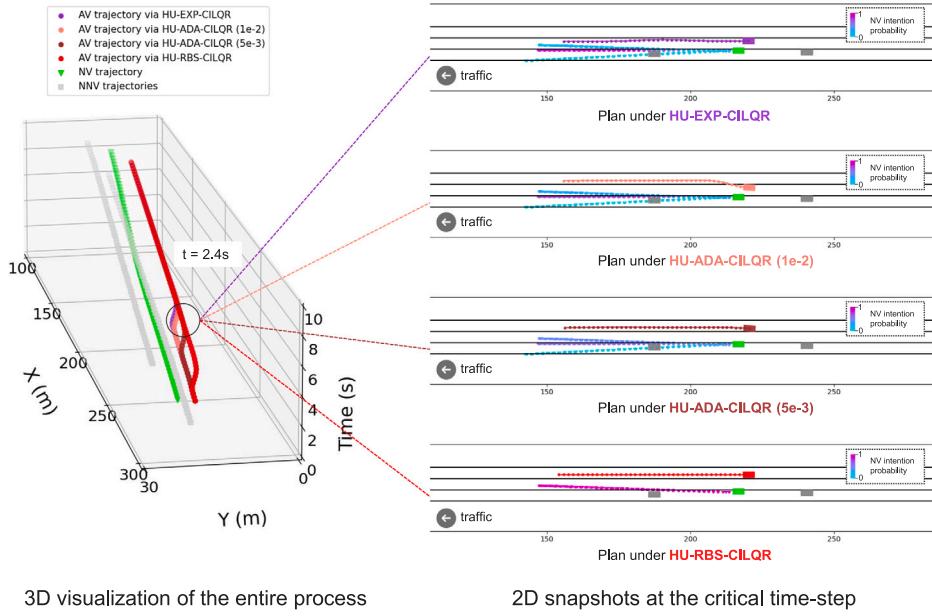


Fig. 13. Trajectory visualization of case 2.

driving on the main lane. The initial states of NV-1 and NV-2 are [991.299 m, 1007.598 m, 5.413 m/s, 3.094 rad] and [1005.223 m, 1007.050 m, 5.566 m/s, 3.101 rad], respectively. The AV's state is initialized by [999.806 m, 1004.30739 m, 5.608 m/s, 3.049 rad]. The following tracks of NVs are from the log-replay of the data. Three motion planners, depending on the schemes of the safety constraints being used, are adopted to generate the AV's trajectory through the two-stage CILQR upon Cond-SIGAT-AoN(20):

- DTM-CILQR: the baseline motion planner under deterministic safety constraints stated in Eq. (15), where the expectation of 20 predicted trajectories under the most probable intention is utilized as parameter inputs.
- HU-EXP-CILQR: hierarchical uncertainty-aware motion planner under the expected scheme, using the weighted-average approach to address multi-modal uncertainties as introduced by Wang et al. (2020).
- HU-RBS-CILQR: hierarchical uncertainty-aware motion planner under the robust scheme, accounting for intentions that drive NVs to the AV's lane.

Notice that the adaptive scheme is not employed here since the lane-changing intentions for Case 1 only include LK and LCL. In addition, key parameters are defined as follows:  $v^d = 9.0 \text{ m/s}$ ,  $w_1 = 2.0$ ,  $w_2 = 0.1$ ,  $w_3 = 1.0$ ,  $w_4 = 3.0$ ,  $s_{safe} = 2.4 \text{ m}$ ,  $\epsilon = 0.01$ . The maximum iteration of a CILQR loop is set to be 100.

Fig. 11 displays the trajectories of AVs computed by the three motion planners, colored in blue (under DTM-CILQR), purple (under HU-EXP-CILQR), and red (under HU-RBS-CILQR), alongside the predicted behaviors of NVs. Using DTM-CILQR, the AV collides with a merging NV at approximately 4 s. The trajectory planned by HU-EXP-CILQR, though closely resembles that of DTM-CILQR, prevents the accident owing to its uncertainty awareness ability. Comparatively, AV under HU-RBS-CILQR adopts a conservative driving style in its initial two seconds, constrained by the merging intentions of two NVs. Beyond this period, the AV drives at a fast speed to overtake NV-2, previously ahead on the adjacent lane. In this way, AV under HU-RBS-CILQR guarantees a safe distance from NVs being as large as possible.

Fig. 12 provides further insights into the distinctions from the velocity and distance to the NVs measured from the AV body center. The acceleration phase of AV under HU-RBS-CILQR during 2.5 s – 4 s can be seen clearly, which may result in increased discomfort for passengers on the AV as a trade-off of its improved safety performance due to the robust scheme.

Additionally, we investigate the effect of intention prediction on the planning results, utilizing a variant model that is trained to directly predict 60 possible trajectories under the AoN loss without making intention prediction. The detailed experimental process and corresponding results is eliminated from the manuscript due to paper length consideration. Readers who have an interest may require the results from the authors.

### 5.3.2. Case 2-response to discretionary lane change

Case 2, reconstructed from HighD dataset, intends to investigate the planner's capability on performing courtesy lane changes in reaction to an NV's discretionary lane change. The AV is positioned in the middle lane of a three-lane highway at the beginning, with a lane-changing NV starting in the leftmost lane. Additionally, two NNVs, acting as the NV's leader and follower vehicles, are present without influencing the AV's decision-making. The initial states of the AV and NV are set at [252.730 m, 13.030 m,

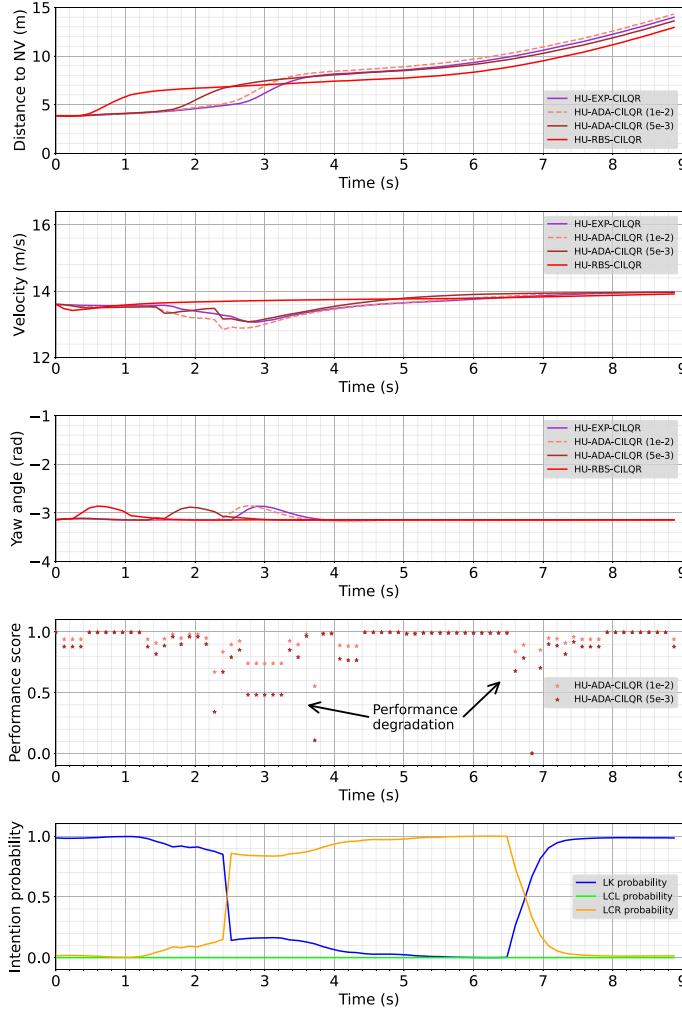


Fig. 14. Illustration of the distance relation, the AV's velocity and yaw angle, the predictive performance scores, and intention predictions of case 2.

13.600 m/s,  $-3.139$  rad] and [250.73 m, 16.33 m, 13.300 m/s,  $-3.132$  rad], respectively. Two reference paths are designated for AV planning based on the center lines of the middle and rightmost lanes. A decision on courtesy lane-changing is made when the cost of switching to the rightmost lane is lower than that of lane-keeping. In addition to the uncertainty-aware planners (HU-EXP-CILQR and HU-RBS-CILQR) introduced in Case 1, two planners adopting safety constraints under the adaptive schemes are introduced:

- HU-ADA-CILQR(1e-2): hierarchical uncertainty-aware motion planner under the adaptive scheme, with a mismatch threshold of  $\Theta = 1e-2$ .
- HU-ADA-CILQR(5e-3): hierarchical uncertainty-aware motion planner under the adaptive scheme, with a mismatch threshold of  $\Theta = 5e-3$ .

The parameters are set as  $v^d = 14.0$  m/s,  $w_1 = 2.0$ ,  $w_2 = 0.8$ ,  $w_3 = 1.0$ ,  $w_4 = 3.0$ . The safety distance is  $s_{safe} = 2.4$  m. The chance-constraint probability is  $\epsilon = 0.05$ . The maximum iteration of a CILQR loop is 100.

The trajectories of the AV and NV are shown in Fig. 13, colored in purple (under HU-EXP-CILQR), salmon (under HU-ADA-CILQR(1e-2)), brown (under HU-ADA-CILQR (5e-3)), red (under HU-RBS-CILQR), and green respectively. We observe that at  $t = 2.4$  s, the NV crosses the lane marking, yet the dominant predicted intention remains lane-keeping, indicating a discrepancy between the prediction and the ground truth. Under HU-EXP-CILQR, where the expected scheme is adopted, the risk of encountering a critical situation may increase due to the trust in predictions and, hence, a lag in response. Contrarily, the two adaptive planners are more proactive in performing a courtesy lane change for collision avoidance, thanks to their ability to dynamically switch between the expected and robust schemes. By setting a lower mismatch threshold ( $\Theta = 5e-3$ ), the resulting trajectory (colored in brown) presents a closer proximity to that under robust scheme (colored in red). Further details of the results are depicted in Fig. 14, including the distance between the AV and NV, the AV's velocity and yaw angle, the predictive performance scores, as well as the

**Table 5**  
Solution time summary.

Metric	Case 1			Case 2			
	DTM	EXP	RBS	EXP	ADA(1e-2)	ADA(5e-3)	RBS
Average solution time (s)	0.090	0.092	0.107	0.125	0.135	0.139	0.154

predicted intention probabilities. One can observe that: (i). Implementing adaptive schemes reduces the duration the AV remains within close proximity to the NV (less than 6 m), compared to the expected scheme; (ii). Implementing adaptive schemes also allows for earlier lane changes with smaller velocity reduction ( $\Theta = 5e-3$ ) or later changes with a slightly greater velocity reduction ( $\Theta = 1e-2$ ), both of which offer safer maneuvers than the expected scheme and less conservative maneuvers than the robust scheme; (iii). The performance score of the prediction model drops during 2.0 s–4.0 s and 6.5 s–7.0 s as an intention class supersedes another in terms of the predicted probability distribution. This is consistent with the analysis in Fig. 8, suggesting planners should exercise increased caution, such as switching to a robust scheme, amidst significant uncertainty in intention predictions.

### 5.3.3. Computational efficiency of planning models

We implemented the two case studies above using Python 3.10 with Numba 0.57 on a 13th Gen Intel(R) Core(TM) i9-13900H 2.60 GHz computing device. The computational efficiency of different planners is summarized in Table 5. In Case 1, the results show that the planner adopting deterministic safety constraints requires the least solution time, whereas the planner built upon the robust scheme consumes the most time. This is because when accounting for the worst case, the optimization algorithm demands a longer search process to yield the desired solution that complies with safety constraints. For Case 2, the computational loads of the adaptive schemes lie between that of the expected and robust schemes due to the weighted combination of the two approaches. Notably, the offline reliability measurement requires little additional time to manage the weighting mechanism based on JS divergence in implementing the online adaptive planning. Overall, the two-stage CILQR can solve the planning problem in approximately 0.1 s within the simulation environment. In future work, more efforts will be devoted to optimizing the programming structure with C++ language to better guarantee real-time performance.

## 6. Conclusion

The uncertainty inherent in the motion of surrounding vehicles and the challenge of predicting it poses significant difficulties for motion planning of autonomous vehicles. The complexities are amplified particularly in scenarios involving lane changes, which require a multimodal behavior prediction on both lane-changing intentions and corresponding trajectories to be performed for safety motion planning. This study develops an integrated framework incorporating behavior prediction and motion planning for implementing autonomous driving in lane-changing scenarios, especially accounting for the motion and model uncertainties associated with the behavior prediction. A deep learning-based model is constructed to predict the lane-changing intention and the intention-conditioned future trajectories, providing hierarchical information about the behavior uncertainty. On this basis, we study three different approaches to formulating the hierarchical uncertainty-aware safety constraints, in which the adaptive scheme is particularly presented to dynamically adjust preferences on the regular and conservative driving modes based on the JS divergence-based performance measure. An extended two-stage CILQR algorithm is proposed to offer an online motion planning solution in a highly uncertain and constrained environment. Through the validation using publicly available datasets, our approach demonstrates satisfactory prediction performance. Two case studies using real-world reconstructed data manifest the capability of our uncertainty-aware motion planner in dealing with both mandatory and discretionary lane changes in neighboring vehicles.

Some extensions can be made to the current study. First, although our study concentrates on the safety-critical lane-changing scenarios, the methodologies introduced can be immigrated to other traffic environments featuring hierarchical behavior uncertainty issues, e.g. intersections with left-turning and straight-moving maneuvers, through modifying and retraining the behavior prediction model. Second, more advanced and higher-order vehicle kinematic models can be further utilized to ensure the smoothness of AV motion under uncertain and dynamic traffic conditions. Besides, further refinement over the safety constraints can be provided to accommodate real-world applications. While a risk constant  $\epsilon$  is adopted for simplicity when building our chance constraints, it could be extended to an adjustable parameter based on the planning horizon and the number of neighboring vehicles. In the meantime, a tighter upper bound than that provided by Cantelli's inequality (Wang et al., 2020) could further improve the safety performance without compromising the integrity of the overall model framework. Lastly, in designing the adaptive planning, we primarily monitored the intention prediction performance owing to its maneuver-level safety significance. Future efforts can be devoted to accounting for the model performance uncertainty in terms of trajectory prediction, gaining more safety margins in a finer degree.

## CRediT authorship contribution statement

**Ruoyu Yao:** Writing – original draft, Visualization, Software, Methodology, Formal analysis. **Xiaotong Sun:** Writing – review & editing, Supervision, Methodology, Funding acquisition, Formal analysis, Conceptualization.

## Acknowledgments

This study is partly supported by the National Science Foundation of China (72201073) and Guangzhou Municipal Science and Technology Project, China (2023A03J0679, 2023A03J0011).

## Appendix A. Nomenclature

See [Table A.1](#).

## Appendix B. Proof of [Proposition 3.1](#)

**Proof.** The initial inequalities necessitate a robust counterpart through the chance-constrained approach, resulting in

$$\mathbb{P}(\mathbf{H}_t^{i,\eta} < 0) < \epsilon \quad \forall t \in \{t_{sta}, \dots, t_{plan}\}, i \in \{1, \dots, |M|\}, \eta \in \{1, \dots, 4\} \quad (\text{B.1})$$

with  $\epsilon$  a small value representing the acceptable risk level.

Directly calculating  $\mathbb{P}(\mathbf{H}_t^{i,\eta} < 0)$  imposes a strong hypothesis on the trajectory distribution, a challenging condition that undermines the model generality. Alternatively, concentration inequalities are introduced to establish an upper bound that reformulates the condition. The canonical expression of Cantelli's inequality ([Haimes and Steuer, 2012](#)) states that for a real-valued random variable  $X$ , given  $\lambda > 0$ , we have:

$$\mathbb{P}(X - \mu_X \leq -\lambda) \leq \frac{\sigma_X^2}{\lambda^2 + \sigma_X^2}$$

Our expression is in the special case when  $\lambda = \mu_X$ . Specifically, Cantelli's inequality ensures that:

$$\mathbb{P}(\mathbf{H}_t^{i,\eta} \leq 0) \leq \frac{\sigma_{\mathbf{H}_t^{i,\eta}}^2}{\mu_{\mathbf{H}_t^{i,\eta}}^2 + \sigma_{\mathbf{H}_t^{i,\eta}}^2}, \quad (\text{B.2a})$$

$$\mu_{\mathbf{H}_t^{i,\eta}} > 0, \quad (\text{B.2b})$$

Accordingly, Eq. [\(B.1\)](#) is valid if Eq. [\(16\)](#) attains.  $\square$

## Appendix C. The online performance monitoring method

To initiate the method, the intention probability distribution is first degraded and partitioned from a three-dimensional plane into a two-dimensional matrix, whose mathematical definition is provided as follows.

**Definition C.1 (2D Intention Matrix).** Let  $\rho$  denote the number of cells along one dimension of the matrix and  $u$  the resolution of the partitioning so that  $\rho = 1/u$ . Therefore, the **2D intention matrix** is represented by a set of cells,  $\mathbb{D}$ , whose formulation is given by

$$\mathbb{D} = \left\{ c(n_1, n_2; u) \mid \rho \in \mathbb{Z}^+, n_1, n_2 \in \mathbb{Z}_0^+ \cap [0, \rho], n_1 + n_2 \leq \rho + 1, u = 1/\rho \right\} \quad (\text{C.1})$$

in which each cell is defined by

$$c(n_1, n_2; u) = ([n_1 u, (n_1 + 1)u), [n_2 u, (n_2 + 1)u)).$$

Using the 2D intention matrix, any intention probability  $\mathbf{Pr} = (Pr_{LK}, Pr_{LCL}, Pr_{LCR})$ , which satisfies  $Pr_{LK} + Pr_{LCL} + Pr_{LCR} = 1$ , can be categorized into a corresponding cell  $c(n_1, n_2; u)$ , if

$$n_1 u \leq Pr_{LCL} \leq (n_1 + 1)u, n_2 u \leq Pr_{LCR} \leq (n_2 + 1)u. \quad (\text{C.2})$$

The average intention probability of a cell,  $\bar{\mathbf{Pr}}$ , can also be derived as

$$\bar{Pr}_{LK}(n_1, n_2; u) = 1 - \frac{2n_1 + 1}{2}u - \frac{2n_2 + 1}{2}u, \bar{Pr}_{LCL}(n_1, n_2; u) = \frac{2n_1 + 1}{2}u, \bar{Pr}_{LCR}(n_1, n_2; u) = \frac{2n_2 + 1}{2}u. \quad (\text{C.3})$$

Consequently, the theoretical normalized performance score,  $S(\hat{\mathcal{V}}^i, \mathcal{V}^i)$ , can be replaced by a proxy  $\hat{S}(n_1, n_2; u, \Theta)$ , which requires the observed intention frequency as a surrogate metric of actually intention prediction. We apply the trained prediction model to generate predictions on the offline test samples that are labeled with one-shot ground-truth intentions. Following the relationship identified in Eq. [\(C.2\)](#), a set of indicators  $\mathbb{I}(n_1, n_2; u)$  is generated, such that if the predicted intention  $\hat{Pr}_{LK}^i, \hat{Pr}_{LCL}^i, \hat{Pr}_{LCR}^i$  corresponds to cell  $c(n_1, n_2; u)$ , the value of  $\mathbb{I}(n_1, n_2; u)$  equals one, otherwise, zero. In addition, for each sample, we also record its true intention by a tuple  $(r'_{LK}, r'_{LCL}, r'_{LCR})$ , for which  $r'_{LK}, r'_{LCL}, r'_{LCR} \in \{0, 1\}$  and  $r'_{LK} + r'_{LCL} + r'_{LCR} = 1$ . With that, we can calculate the observed intention frequency  $\mathbf{Fr}$  by

$$Fr_I(n_1, n_2; u) = \frac{\sum_I r'_I \mathbb{I}(n_1, n_2; u)}{\sum_I \mathbb{I}(n_1, n_2; u)}, I \in \{LK, LCL, LCR\}. \quad (\text{C.4})$$

**Table A.1**  
Variables description table.

Module	Variable	Description
General	$AV$	Autonomous vehicle implementing the behavior prediction and motion planning pipeline.
	$NV$	Neighboring vehicle, whose behavior is predicted by the AV.
	$NNV$	Neighbor's neighboring vehicle, vehicles adjacent to the NV.
Behavior prediction–Traffic feature extractor	$X_t^i$	Input vector for NV $i$ at time step $t$ , consisting of relative position $(\Delta x_t^i, \Delta y_t^i)$ , velocity $v_t^i$ , and vehicle dimensions $\zeta^i$ (length) and $\nu^i$ (width).
	$\Delta x_t^i, \Delta y_t^i$	Two-dimensional coordinates of NV $i$ in a relative coordinate system at time step $t$ .
	$v_t^i$	Velocity of NV $i$ at time step $t$ .
	$\zeta^i, \nu^i$	Length and width of NV $i$ , respectively.
	$h_t^i$	Hidden state of NV $i$ at time step $t$ obtained from the motion-encoding LSTM.
	$W_{MoEnc}$	Set of trainable parameters within the motion-encoding LSTM.
	$\mathcal{N}^i$	Neighborhood of NV $i$ , consisting of NNVs and NV $i$ itself.
	$a_l^{i,j}$	Attention score between NV $i$ and NNV $j$ at layer $l$ in the GAT.
	$a_l, W_l$	Learnable linear transformation vector and matrix, respectively, used in the attention mechanism of the GAT.
	$g_l^i$	Node state of vehicle $i$ at layer $l$ in the GAT.
Behavior prediction–Conditional behavior decoder	$H$	Number of parallel self-attentions in the multi-head attention mechanism.
	$R_\gamma^{up}, R_\gamma^{lo}$	Coordinate vectors of sample points $\gamma$ from the upper and lower boundary lines around the NV, respectively, in the NV-centric frame.
	$\Delta x_\gamma^{up}, \Delta y_\gamma^{up}$	Coordinates of upper boundary sample point $\gamma$ in the NV-centric frame.
	$\Delta x_\gamma^{lo}, \Delta y_\gamma^{lo}$	Coordinates of lower boundary sample point $\gamma$ in the NV-centric frame.
	$h_\gamma^{up}, h_\gamma^{lo}$	Hidden states of sample points $\gamma$ from the upper and lower boundary lines, obtained from the scene-encoding LSTM.
	$W_{ScEnc}$	Set of trainable parameters within the scene-encoding LSTM.
	$e^i$	Compound representation of NV $i$ -centric traffic features.
	$(\hat{P}_{LK}^i, \hat{P}_{LCL}^i, \hat{P}_{LCR}^i)$	Predicted probabilities of lane-keeping, lane-changing to the left, and lane-changing to the right, respectively.
	$W_{int1}, W_{int2}$	Learnable weights of the MLP in the intention prediction task.
	$o_t^{i,I}$	Hidden state of the motion-decoding LSTM for intention class $I$ and NV $i$ at time step $t$ .
Behavior prediction–Loss function	$\hat{Y}_t^{i,I}$	Predicted displacement for NV $i$ under intention $I$ at time step $t$ .
	$\hat{Y}_t^{i,I}$	Predicted absolute position for NV $i$ under intention $I$ at time step $t$ .
	$P_{t_{obs}}^i$	Last observed absolute position of NV $i$ .
	$W_{MoDec}^I, W_{traj}^I$	Parameter sets of the motion-decoding LSTM and MLP for trajectory prediction under intention $I$ , respectively.
	$L_{int}, L_{traj}, L_{overall}$	Loss functions for intention prediction, deterministic trajectory prediction, and the overall synthesized loss, respectively.
	$w_{int}, w_{traj}$	Weights for intention and trajectory prediction losses in the overall loss function.
	$L_{traj}^{AoN}, L_{traj}^{MoN}$	Loss functions for stochastic trajectory prediction in Average-of-N and Minimum-of-N modes, respectively.
	$x_t, y_t$	Coordinates of the ego AV at time step $t$ .
	$v_t$	Velocity of the ego AV at time step $t$ .
	$\theta_t$	Yaw angle of the ego AV at time step $t$ .
Motion planning–Objective function	$a_t$	Acceleration of the ego AV at time step $t$ .
	$\delta_t$	Angular velocity of the ego AV at time step $t$ .
	$[t_{start}, t_{plan}]$	Planning horizon time interval.
	$x_t^{up}, y_t^{up}$	Coordinates of the reference path waypoints at time step $t$ .
	$v^d$	Desired velocity of the ego AV.
	$w_1, w_2, w_3, w_4$	Weights assigned to different components in the objective function.

(continued on next page)

**Table A.1** (continued).

Motion planning–Standard model constraints	$S_t$	State of the vehicle at time step $t$ , including position, velocity, and yaw angle.
	$A_t$	Action of the vehicle at time step $t$ , including acceleration and angular velocity.
	$T_s$	System time resolution.
	$a_{min}, a_{max}$	Minimum and maximum acceleration limits, respectively.
	$\delta_{lim}$	Angular velocity limit.
	$\kappa_i^{up}, b_i^{up}, \kappa_i^{lo}, b_i^{lo}$	Slope and intercept of the upper and lower road boundary segments at time step $t$ .
	$A$	Distance constant for the buffer space between the vehicle and road boundaries.
	$(F_x)_t, (F_y)_t$	Coordinates of the AV's front circle center at time step $t$ .
Motion planning–Safety constraints	$(B_x)_t, (B_y)_t$	Coordinates of the AV's rear circle center at time step $t$ .
	$(\hat{F}_x)_t^i, (\hat{F}_y)_t^i$	Coordinates of NV $i$ 's front circle center predicted at time step $t$ .
	$(\hat{B}_x)_t^i, (\hat{B}_y)_t^i$	Coordinates of NV $i$ 's rear circle center predicted at time step $t$ .
	$M$	The set of NVs.
	$s_{safe}$	Given constant safety distance.
	$L$	Wheelbase of the ego AV.
	$\mathbf{H}_t^{i,\eta}$	Safety criterion function for AV and NV $i$ at time step $t$ and condition $\eta$ .
	$\epsilon$	Acceptable risk level for chance constraints.
	$\mu_{\mathbf{H}_t^{i,\eta}}, \sigma_{\mathbf{H}_t^{i,\eta}}^2$	Mean and variance of the safety criterion function $\mathbf{H}_t^{i,\eta}$ .
	$\mathbb{E}^{exp}((\mathbf{H}_t^{i,\eta})^\alpha)$	Estimated $\alpha$ -order moment of $\mathbf{H}_t^{i,\eta}$ under the expected scheme.
	$\mathbb{E}^{rhs}((\mathbf{H}_t^{i,\eta})^\alpha)$	Estimated $\alpha$ -order moment of $\mathbf{H}_t^{i,\eta}$ under the robust scheme.
	$\mathbb{E}^{ada}((\mathbf{H}_t^{i,\eta})^\alpha)$	Estimated $\alpha$ -order moment of $\mathbf{H}_t^{i,\eta}$ under the adaptive scheme.
CILQR algorithm	$lane_t^{AV}, lane_t^i$	Lane indexes where AV and NV $i$ drive on at time-step $t$ , respectively.
	$I^{wst}(lane_t^{AV}, lane_t^i)$	Intention associated with the most adverse scenario according to the current lanes of AV and NV $i$ .
	$S(\hat{\mathcal{V}}, \mathcal{V}^i)$	Normalized performance score assessing the fitness between predicted and actual intention probabilities for NV $i$ .
	$JSD(p  q)$	Jensen-Shannon divergence measuring the similarity between two distributions $p$ and $q$ .
	$\theta$	Threshold value of JSD representing the maximum mismatch accepted between two distributions.
	$h(S_t, A_t)$	Function describing the dynamic transition based on the vehicle state and action at time $t$ .
	$V_t(S_t)$	Value function at time-step $t$ .
	$Q_t(S_t, A_t)$	Action-value function (Q-function) at time-step $t$ .
	$C(S_t, A_t)$	Cost at the time-step $t$ .
	$\phi(S_t, A_t)$	Set of inequality constraints on the vehicle state and action at time $t$ .
CILQR algorithm	$b_{soft}(S_t, A_t)$	Exponential barrier function used in Soft CILQR loop for converting inequality constraints into cost terms.
	$q_1, q_2$	Tunable parameters in the exponential barrier function.
	$b_{hard}(S_t, A_t)$	Logarithmic barrier function used in Hard CILQR loop for converting inequality constraints into cost terms.
	$v$	Parameter in the logarithmic barrier function, dynamically increased to accelerate convergence.
	$A_{t_{sta}:t_{plan}-1}^h$	Initial heuristic control sequence.
	$S_{t_{sta}:t_{plan}}^h$	State sequence calculated using the vehicle kinematic model with the initial control sequence.
	$\tilde{S}_{t_{sta}:t_{plan}}, \tilde{A}_{t_{sta}:t_{plan}-1}$	Nominal trajectory including state and action sequences.
	$\tilde{S}_{t_{sta}:t_{plan}}, \tilde{A}_{t_{sta}:t_{plan}-1}$	Updated state and action sequences after an ILQR iteration.
	$S_{t_{sta}:t_{plan}}^*, A_{t_{sta}:t_{plan}-1}^*$	Optimized state and action sequences.
	$\delta S_t, \delta A_t$	Perturbed state and action at time step $t$ .
	$\lambda_0, \lambda_{max}$	Initial and maximum values of the regularization parameter, respectively.
	$\lambda$	Regularization parameter used to preserve the Hessian's positive definiteness in the optimization process.
	$\Delta_1, \Delta_2$	Factors for decreasing or increasing the regularization parameter.

(continued on next page)

**Table A.1** (continued).

$\mu$	Factor for increasing the parameter $v$ in the outer loop of Hard CILQR.	
$B, \tilde{B}$	Costs associated with the current and updated trajectories in the Soft CILQR loop, respectively.	
$C_+, \tilde{C}_+$	Costs including original cost and constraints-converted terms for the current and updated trajectories in the Hard CILQR loop, respectively.	
$O_{t,S}, O_{t,A}$	First-order partial derivatives of the Q-function with respect to state and action at time step $t$ .	
$Q_{t,SS}, Q_{t,AA}, Q_{t,AS}, Q_{t,SA}$	Second-order partial derivatives of the Q-function with respect to state and action, used in the quadratic approximation.	
$h_S, h_A$	First-order partial derivatives of the system dynamic function with respect to state and action at time step $t$ .	
$h_{SS}, h_{AA}, h_{AS}, h_{SA}$	Second-order partial derivatives of the system dynamic function with respect to state and action, treated as zeros in the system linearization.	
$k_t, K_t$	Open-loop term (feedforward term) and feedback control gain, respectively, used in optimal control updates.	
$\Delta V_t$	Reduced cumulative costs due to optimization at time step $t$ .	
$V_{t,S}, V_{t,SS}$	First and second-order partial derivatives of the value function with respect to state, used in backward induction.	
$\rho$	Line search parameter used in the forward pass to calculate the updated action-state sequence.	
$\hat{Q}_{t,AA}, \hat{Q}_{t,AS}$	Modified Hessian matrices including the regularization term.	
$I_{\dim(S)}$	Identity matrix aligned with the dimension of the state space, used in regularization.	
Experiment	$U_{int}^i$	Intention uncertainty score for the $i$ th NV.
	$U_{traj}^i$	Trajectory uncertainty score for the $i$ th NV.
	$\hat{Y}_t^{i,I}$	Average predicted position of the $i$ th NV at time-step $t$ conditioned on intention $I$ .
	$MI(X; Y)$	Mutual Information between random variables $X$ and $Y$ .
	$RE(X_a; Y)$	Relative Effect of factor $X_a$ on uncertainty score $Y$ .
	$h(X_a)$	Entropy of the random variable $X_a$ .
	$F$	Set of factors considered for analysis.

As the two sets of distributions are obtained offline, the performance score  $\hat{S}(n_1, n_2; u, \Theta)$  is calculated in an offline fashion, whose definition is given below:

**Definition C.2 (Normalized JSD-Based Performance Score).** Suppose the predicted and observed intention distributions,  $\bar{Pr}$  and  $Fr$ , follow Eqs. (C.3) and (C.4). A positive parameter  $\Theta$  measures the maximum mismatch that can be accepted between two distributions. Then, the **normalized JSD-based performance score** evaluating model uncertainty is given by

$$\hat{S}(n_1, n_2; u, \Theta) = \max\left(\frac{\Theta - JSD(n_1, n_2; u)}{\Theta}, 0\right). \quad (\text{C.5})$$

where **JS divergence** is defined by

$$JSD(n_1, n_2; u) = \frac{1}{2} \sum_{I \in \{LK, LCL, LCR\}} \bar{Pr}_I(n_1, n_2; u) \log\left(\frac{\bar{Pr}_I(n_1, n_2; u)}{\bar{Pr}_I(n_1, n_2; u) + Fr_I(n_1, n_2; u)}\right) + \frac{1}{2} \sum_{I \in \{LK, LCL, LCR\}} Fr_I(n_1, n_2; u) \log\left(\frac{Fr_I(n_1, n_2; u)}{\bar{Pr}_I(n_1, n_2; u) + Fr_I(n_1, n_2; u)}\right) + 1. \quad (\text{C.6})$$

The value of JS divergence lies between 0 and 1, with a lower value indicating higher similarity. Therefore, the score equals one when two distributions are completely the same, and decays as the JS divergence value increases from zero to  $\Theta$ . When the JS divergence value exceeds  $\Theta$ , the score retains at zero, implying that the robust scheme should take full charge when the prediction is regarded as completely untrustworthy. We then denote the set of all scores as  $\mathbb{S}$ . For the real-time implementation, the offline score corresponding to the online prediction can be directly retrieved and used to replace the theoretical value  $S(\hat{V}^i, \mathcal{V}^i)$  in Eq. (19). This process efficiently facilitates adaptive motion planning.

#### Appendix D. The standard ILQR

Standard ILQR optimizes the nominal trajectory by taking second-order Taylor approximations of the perturbed  $Q$ -function. Suppose a perturbation term  $(\delta S_t, \delta A_t)$  is imposed on the incumbent trajectory  $(S_t, A_t)$  for each time-step starting from the  $t_{plan} - 1$

time-step. The perturbed  $Q$ -function, which is the  $Q$ -function difference before and after the perturbation term is introduced, is given by

$$\Delta Q_t(\delta S_t, \delta A_t) = C(S_t + \delta S_t, A_t + \delta A_t) - C(S_t, A_t) + V_{t+1}(h(S_t + \delta S_t, A_t + \delta A_t)) - V_{t+1}(h(S_t, A_t)) \quad (\text{D.1})$$

The dynamic program is optimal if and only if  $Q_t(\delta S_t, \delta A_t)$  is minimized. To find the optimal  $\delta A_t^*$ , we approximate the expression by its second-order Taylor expansion<sup>3</sup>:

$$\Delta Q_t(\delta S_t, \delta A_t) \approx \frac{1}{2} \begin{bmatrix} 1 \\ \delta S_t \\ \delta A_t \end{bmatrix}^T \begin{bmatrix} 0 & (Q_{t,S})^T & (Q_{t,A})^T \\ Q_{t,S} & Q_{t,SS} & Q_{t,SA} \\ Q_{t,A} & Q_{t,AS} & Q_{t,AA} \end{bmatrix} \begin{bmatrix} 1 \\ \delta S_t \\ \delta A_t \end{bmatrix} \quad (\text{D.2})$$

where

$$Q_{t,S} = C_S + (h_S)^T \cdot V_{t+1,S},$$

$$Q_{t,A} = C_A + (h_A)^T \cdot V_{t+1,S},$$

$$Q_{t,SS} = C_{SS} + (h_S)^T \cdot V_{t+1,SS} \cdot h_S + V_{t+1,S} \cdot h_{SS},$$

$$Q_{t,AA} = C_{AA} + (h_A)^T \cdot V_{t+1,SS} \cdot h_A + V_{t+1,S} \cdot h_{AA},$$

$$Q_{t,AS} = C_{AS} + (h_A)^T \cdot V_{t+1,SS} \cdot h_S + V_{t+1,S} \cdot h_{AS},$$

$$Q_{t,SA} = C_{SA} + (h_A)^T \cdot V_{t+1,SS} \cdot h_S + V_{t+1,S} \cdot h_{SA}.$$

The dynamic model is further linearized by ignoring its second-order derivatives, which is,  $h_{SS} = 0, h_{AA} = 0, h_{AS} = h_{SA} = 0$ . The optimal  $\delta A_t^*$  can then be analytically formulated in the close-form:

$$\delta A_t^* = \arg \min_{\delta A_t} Q_t(\delta S_t, \delta A_t) = -(Q_{t,AA})^{-1} Q_{t,A} + -(Q_{t,AA})^{-1} Q_{t,AS} \cdot \delta S_t \quad (\text{D.3})$$

We denote

$$k_t = -(Q_{t,AA})^{-1} Q_{t,A},$$

$$K_t = -(Q_{t,AA})^{-1} Q_{t,AS}.$$

for simplicity, which are respectively referred to as *open-loop term* and *feedback control gain* in optimal control terminology. Incorporating Eq. (D.3) with Eq. (D.2), the value function at time-step  $t$  is updated by

$$\Delta V_t = -\frac{1}{2} Q_{t,A} \cdot Q_{t,AA}^{-1} \cdot Q_{t,A} \quad (\text{D.4a})$$

$$V_t = Q_t - \Delta V_t \quad (\text{D.4b})$$

$$V_{t,S} = Q_{t,S} - Q_{t,A} \cdot Q_{t,AA}^{-1} \cdot Q_{t,AS} \quad (\text{D.4c})$$

$$V_{t,SS} = Q_{t,SS} - Q_{t,SA} \cdot Q_{t,AA}^{-1} \cdot Q_{t,AS} \quad (\text{D.4d})$$

$V_{t,S}$  and  $V_{t,SS}$  calculated by Eq. (D.4c)–(D.4d) are utilized in the backward induction at  $t - 1$  step. Once the backward induction is finished, a forward pass will be executed to generate the updated trajectory:

$$\tilde{S}_1 = S_1 \quad (\text{D.5a})$$

$$\tilde{A}_t = A_t + \rho k_t + K_t(\tilde{S}_t - S_t) \quad (\text{D.5b})$$

$$\tilde{S}_{t+1} = h(\tilde{S}_t, \tilde{A}_t) \quad (\text{D.5c})$$

In forward pass, a line search parameter,  $\rho$ , is used to ensure the feasibility of the update action state. In practice, one can assign different values to  $\rho$  and select the result with the least cost as the output of the iteration.

Other than the above, a regularization parameter can be introduced to preserve the Hessian's positive definiteness. Specifically, the regularization scheme introduced by Tassa et al. (2012) modifies the aforementioned calculation of  $k_t$  and  $K_t$  into:

$$\hat{Q}_{t,AA} = C_{AA} + (h_A)^T \cdot (V_{t+1,SS} + \lambda I_{\dim(S)}) \cdot h_A + V_{t+1,S} \cdot h_{AA},$$

$$\hat{Q}_{t,AS} = C_{AS} + (h_A)^T \cdot (V_{t+1,SS} + \lambda I_{\dim(S)}) \cdot h_S + V_{t+1,S} \cdot h_{AS},$$

$$k_t = -(\hat{Q}_{t,AA})^{-1} \hat{Q}_{t,A},$$

$$K_t = -(\hat{Q}_{t,AA})^{-1} \hat{Q}_{t,AS}.$$

Here  $\lambda$  is the regularization parameter, whose updating procedure in our implementation is elaborated in Algorithm 1.  $I_{\dim(S)}$  denotes the identity matrix aligned with the dimension of the state space.

<sup>3</sup> The term  $X_Y$  denotes the partial derivative of  $X$  with respect to  $Y$  at the current step  $t$ , e.g.,  $C_S = \frac{\partial C}{\partial S_t}$ . And  $X_{YZ}$  denotes the second-order partial derivative of  $X$  over  $Y$  and  $Z$  sequentially, e.g.,  $h_{AS} = \frac{\partial^2 h}{\partial A_t \partial S_t}$ .

## References

- Alahi, A., Goel, K., Ramanathan, V., Robicquet, A., Fei-Fei, L., Savarese, S., 2016. Social lstm: Human trajectory prediction in crowded spaces. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 961–971.
- Ali, Y., Bliemer, M.C., Zheng, Z., Haque, M.M., 2020. Cooperate or not? Exploring drivers interactions and response times to a lane-changing request in a connected environment. *Transp. Res. C* 120, 102816.
- Ali, Y., Zheng, Z., Haque, M.M., Wang, M., 2019. A game theory-based approach for modelling mandatory lane-changing behaviour in a connected environment. *Transp. Res. C* 106, 220–242.
- Aoude, G.S., Luders, B.D., Joseph, J.M., Roy, N., How, J.P., 2013. Probabilistically safe motion planning to avoid dynamic obstacles with uncertain motion patterns. *Auton. Robots* 35, 51–76.
- Chen, J., Shimizu, Y., Sun, L., Tomizuka, M., Zhan, W., 2021. Constrained iterative LQG for real-time chance-constrained Gaussian belief space planning. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 5801–5808.
- Chen, J., Zhan, W., Tomizuka, M., 2017. Constrained iterative lqr for on-road autonomous driving motion planning. In: 2017 IEEE 20th International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 1–7.
- Chen, J., Zhan, W., Tomizuka, M., 2019. Autonomous driving motion planning with constrained iterative LQR. *IEEE Trans. Intell. Veh.* 4 (2), 244–254.
- Chen, X., Zhang, H., Zhao, F., Hu, Y., Tan, C., Yang, J., 2022. Intention-aware vehicle trajectory prediction based on spatial-temporal dynamic attention network for internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* 23 (10), 19471–19483.
- Claussmann, L., Revilloud, M., Gruyer, D., Glaser, S., 2019. A review of motion planning for highway autonomous driving. *IEEE Trans. Intell. Transp. Syst.* 21 (5), 1826–1848.
- Codevilla, F., Müller, M., López, A., Koltun, V., Dosovitskiy, A., 2018. End-to-end driving via conditional imitation learning. In: 2018 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 4693–4700.
- Cui, A., Casas, S., Sadat, A., Liao, R., Urtasun, R., 2021. Lookout: Diverse multi-future prediction and planning for self-driving. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16107–16116.
- Deo, N., Trivedi, M.M., 2018a. Convolutional social pooling for vehicle trajectory prediction. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 1468–1476.
- Deo, N., Trivedi, M.M., 2018b. Multi-modal trajectory prediction of surrounding vehicles with Maneuver based LSTMs. In: 2018 IEEE Intelligent Vehicles Symposium. IV, pp. 1179–1184. <http://dx.doi.org/10.1109/IVS.2018.8500493>.
- Ding, L., Li, D., Liu, B., Lan, W., Bai, B., Hao, Q., Cao, W., Pei, K., 2021. Capture uncertainties in deep neural networks for safe operation of autonomous driving vehicles. In: 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). IEEE, pp. 826–835.
- Dolgov, D., Thrun, S., Montemerlo, M., Diebel, J., 2010. Path planning for autonomous vehicles in unknown semi-structured environments. *Int. J. Robot. Res.* 29 (5), 485–501.
- Fan, H., Su, H., Guibas, L.J., 2017. A point set generation network for 3D object reconstruction from a single image. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR.
- Feng, D., Harakeh, A., Waslander, S.L., Dietmayer, K., 2021. A review and comparative study on probabilistic object detection in autonomous driving. *IEEE Trans. Intell. Transp. Syst.* 23 (8), 9961–9980.
- Fridovich-Keil, D., Bajcsy, A., Fisac, J.F., Herbert, S.L., Wang, S., Dragan, A.D., Tomlin, C.J., 2020. Confidence-aware motion prediction for real-time collision avoidance1. *Int. J. Robot. Res.* 39 (2–3), 250–265.
- Gal, Y., Ghahramani, Z., 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: International Conference on Machine Learning. PMLR, pp. 1050–1059.
- Gao, K., Li, X., Chen, B., Hu, L., Liu, J., Du, R., Li, Y., 2023. Dual transformer based prediction for lane change intentions and trajectories in mixed traffic environment. *IEEE Trans. Intell. Transp. Syst.*
- Gao, J., Sun, C., Zhao, H., Shen, Y., Angelov, D., Li, C., Schmid, C., 2020. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11525–11533.
- Gipps, P.G., 1986. A model for the structure of lane-changing decisions. *Transp. Res. B* 20 (5), 403–414.
- Gu, J., Sun, C., Zhao, H., 2021. Densentnt: End-to-end trajectory prediction from dense goal sets. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 15303–15312.
- Gupta, A., Johnson, J., Fei-Fei, L., Savarese, S., Alahi, A., 2018. Social GAN: Socially acceptable trajectories with generative adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. CVPR.
- Haines, Y.Y., Steuer, R.E., 2012. Research and Practice in Multiple Criteria Decision Making: Proceedings of the XIVth International Conference on Multiple Criteria Decision Making (MCDM) Charlottesville, Virginia, USA, June 8–12, 1998, Vol. 487, Springer Science & Business Media.
- Han, W., Jasour, A., Williams, B., 2022. Non-Gaussian risk bounded trajectory optimization for stochastic nonlinear systems in uncertain environments. In: 2022 International Conference on Robotics and Automation. ICRA, IEEE, pp. 11044–11050.
- Han, R., Wang, S., Wang, S., Zhang, Z., Zhang, Q., Eldar, Y.C., Hao, Q., Pan, J., 2023. RDA: An accelerated collision free motion planner for autonomous navigation in cluttered environments. *IEEE Robot. Autom. Lett.* 8 (3), 1715–1722.
- Hardy, J., Campbell, M., 2013. Contingency planning over probabilistic obstacle predictions for autonomous road vehicles. *IEEE Trans. Robot.* 29 (4), 913–929.
- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Comput.* 9 (8), 1735–1780.
- Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al., 2023. Planning-oriented autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17853–17862.
- Huang, X., Sun, J., Sun, J., 2018. A car-following model considering asymmetric driving behavior based on long short-term memory neural networks. *Transp. Res. C* 95, 346–362.
- hwani Jeon, J., Cowlagi, R.V., Peters, S.C., Karaman, S., Frazzoli, E., Tsotras, P., Iagnemma, K., 2013. Optimal motion planning with the half-car dynamical model for autonomous high-speed driving. In: 2013 American Control Conference. IEEE, pp. 188–193.
- Jin, W.-L., 2010. A kinematic wave theory of lane-changing traffic flow. *Transp. Res. B* 44 (8–9), 1001–1021.
- Kesting, A., Treiber, M., Helbing, D., 2007. General lane-changing model MOBIL for car-following models. *Transp. Res. Rec.* 1999 (1), 86–94.
- Knoop, V., Keyvan-Ekbatani, M., de Baat, M., Taale, H., Hoogendoorn, S., 2018. Lane change behavior on freeways: an online survey using video clips. *J. Adv. Transp.* 2018 (1), 9236028.
- Krajewski, R., Bock, J., Kloeker, L., Eckstein, L., 2018. The hghd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems. In: 2018 21st International Conference on Intelligent Transportation Systems. ITSC, IEEE, pp. 2118–2125.
- Kuwata, Y., Teo, J., Fiore, G., Karaman, S., Frazzoli, E., How, J.P., 2009. Real-time motion planning with applications to autonomous urban driving. *IEEE Trans. Control Syst. Technol.* 17 (5), 1105–1118.
- Lakshminarayanan, B., Pritzel, A., Blundell, C., 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. *Adv. Neural Inf. Process. Syst.* 30.
- Lan, W., Li, D., Hao, Q., Zhao, D., Tian, B., 2023. Implicit scene context-aware interactive trajectory prediction for autonomous driving. *IEEE Trans. Intell. Veh.*

- Lee, N., Choi, W., Vernaza, P., Choy, C.B., Torr, P.H., Chandraker, M., 2017. Desire: Distant future prediction in dynamic scenes with interacting agents. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 336–345.
- Li, D., Liu, B., Huang, Z., Hao, Q., Zhao, D., Tian, B., 2023. Safe motion planning for autonomous vehicles by quantifying uncertainties of deep learning-enabled environment perception. *IEEE Transactions on Intelligent Vehicles*.
- Li, J., Sun, L., Chen, J., Tomizuka, M., Zhan, W., 2021. A safe hierarchical planning framework for complex driving scenarios based on reinforcement learning. In: 2021 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 2660–2666.
- Likhachev, M., Ferguson, D., 2009. Planning long dynamically feasible maneuvers for autonomous vehicles. *Int. J. Robot. Res.* 28 (8), 933–945.
- Lin, J., 1991. Divergence measures based on the Shannon entropy. *IEEE Trans. Inf. Theory* 37 (1), 145–151.
- Liu, H., Chen, K., Li, Y., Huang, Z., Duan, J., Ma, J., 2023. Integrated behavior planning and motion control for autonomous vehicles with traffic rules compliance. In: 2023 IEEE International Conference on Robotics and Biomimetics. ROBIO, IEEE, pp. 1–7.
- Liu, C., Lee, S., Varnhagen, S., Tseng, H.E., 2017. Path planning for autonomous vehicles using model predictive control. In: 2017 IEEE Intelligent Vehicles Symposium. IV, IEEE, pp. 174–179.
- Maas, A.L., Hannun, A.Y., Ng, A.Y., et al., 2013. Rectifier nonlinearities improve neural network acoustic models. In: Proc. Icm. Vol. 30, Atlanta, GA, p. 3.
- McNaughton, M., Urmon, C., Dolan, J.M., Lee, J.-W., 2011. Motion planning for autonomous driving with a conformal spatiotemporal lattice. In: 2011 IEEE International Conference on Robotics and Automation. IEEE, pp. 4889–4895.
- Mozaffari, S., Al-Jarrah, O.Y., Dianati, M., Jennings, P., Mouzakitis, A., 2020. Deep learning-based vehicle behavior prediction for autonomous driving applications: A review. *IEEE Trans. Intell. Transp. Syst.* 23 (1), 33–47.
- Shannon, C.E., 1948. A mathematical theory of communication. *Bell Syst. Tech. J.* 27 (3), 379–423.
- Shao, W., Xu, Y., Peng, L., Li, J., Wang, H., 2023. Failure detection for motion prediction of autonomous driving: An uncertainty perspective. arXiv preprint arXiv:2301.04421.
- Shawky, M., 2020. Factors affecting lane change crashes. *IATSS Res.* 44 (2), 155–161.
- Shimizu, Y., Zhan, W., Sun, L., Chen, J., Kato, S., Tomizuka, M., 2020. Motion planning for autonomous driving with extended constrained iterative lqr. In: Dynamic Systems and Control Conference. 84270, American Society of Mechanical Engineers, V001T12A001.
- Song, R., Li, B., 2021. Surrounding vehicles' lane change maneuver prediction and detection for intelligent vehicles: A comprehensive review. *IEEE Trans. Intell. Transp. Syst.* 23 (7), 6046–6062.
- Sun, D., Elefteriadou, L., 2014. A driver behavior-based lane-changing model for urban arterial streets. *Transp. Sci.* 48 (2), 184–205.
- Tang, X., Yang, K., Wang, H., Wu, J., Qin, Y., Yu, W., Cao, D., 2022. Prediction-uncertainty-aware decision-making for autonomous vehicles. *IEEE Trans. Intell. Veh.* 7 (4), 849–862.
- Tassa, Y., Erez, T., Todorov, E., 2012. Synthesis and stabilization of complex behaviors through online trajectory optimization. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 4906–4913.
- Toledo, T., Koutsopoulos, H.N., Ben-Akiva, M.E., 2003. Modeling integrated lane-changing behavior. *Transp. Res. Rec.* 1857 (1), 30–38.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.U., Polosukhin, I., 2017. Attention is all you need. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Vol. 30, Curran Associates, Inc., URL [https://proceedings.neurips.cc/paper\\_files/paper/2017/file/3f5ee243547dee91fb053c1c4a845aa-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fb053c1c4a845aa-Paper.pdf).
- Veličković, P., Cucurull, G., Casanova, A., Romero, A., Lio, P., Bengio, Y., 2017. Graph attention networks. arXiv preprint arXiv:1710.10903.
- Wächter, A., Biegler, L.T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Math. Program.* 106, 25–57.
- Wang, A., Jasour, A., Williams, B.C., 2020. Non-gaussian chance-constrained trajectory planning for autonomous vehicles under agent uncertainty. *IEEE Robot. Autom. Lett.* 5 (4), 6041–6048.
- Wu, H., Liu, M., 2022. Lane-GNN: Integrating GNN for predicting drivers' lane change intention. In: 2022 IEEE 25th International Conference on Intelligent Transportation Systems. ITSC, pp. 4138–4144. <http://dx.doi.org/10.1109/ITSC55140.2022.9922139>.
- Xie, D.-F., Fang, Z.-Z., Jia, B., He, Z., 2019. A data-driven lane-changing model based on deep learning. *Transp. Res. C* 106, 41–60.
- Yao, R., Zeng, W., Chen, Y., He, Z., 2021. A deep learning framework for modelling left-turning vehicle behaviour considering diagonal-crossing motorcycle conflicts at mixed-flow intersections. *Transp. Res. C* 132, 103415.
- Zeng, W., Wang, S., Liao, R., Chen, Y., Yang, B., Urtasun, R., 2020. Dsdnet: Deep structured self-driving network. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16. Springer, pp. 156–172.
- Zeng, J., Zhang, B., Sreenath, K., 2021. Safety-critical model predictive control with discrete-time control barrier function. In: 2021 American Control Conference. ACC, IEEE, pp. 3882–3889.
- Zhan, W., Sun, L., Wang, D., Shi, H., Clausse, A., Naumann, M., Kummerle, J., Konigshof, H., Stiller, C., de La Fortelle, A., et al., 2019. Interaction dataset: An international, adversarial and cooperative motion dataset in interactive driving scenarios with semantic maps. arXiv preprint arXiv:1910.03088.
- Zhang, X., Sun, J., Qi, X., Sun, J., 2019. Simultaneous modeling of car-following and lane-changing behaviors using deep learning. *Transp. Res. C* 104, 287–304.
- Zhou, J., Olofsson, B., Frisk, E., 2023. Interaction-aware motion planning for autonomous vehicles with multi-modal obstacle uncertainty predictions. *IEEE Trans. Intell. Veh.*
- Ziegler, J., Bender, P., Schreiber, M., Lategahn, H., Strauss, T., Stiller, C., Dang, T., Franke, U., Appenrodt, N., Keller, C.G., et al., 2014. Making bertha drive—an autonomous journey on a historic route. *IEEE Intell. Transp. Syst. Mag.* 6 (2), 8–20.