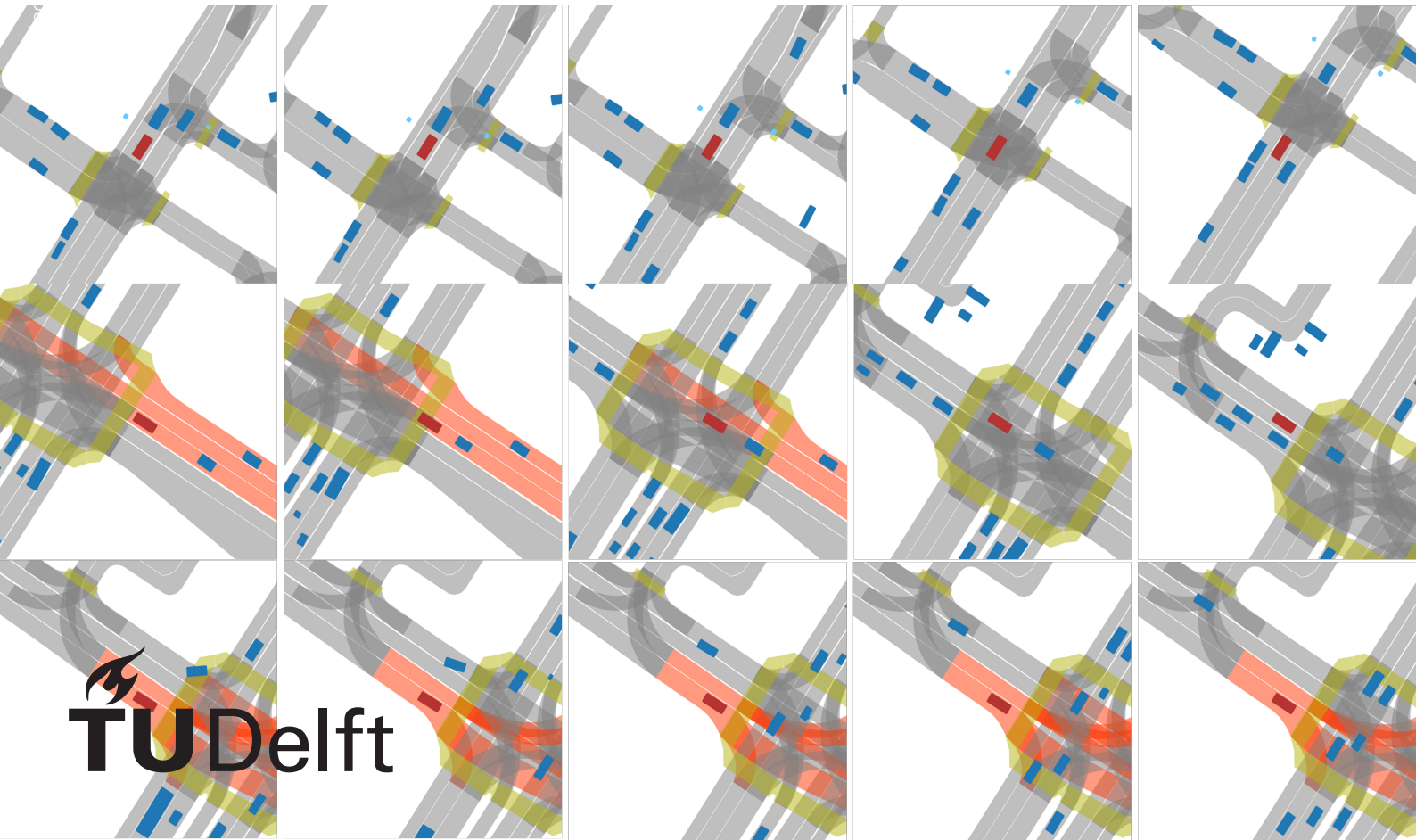
Learning from Demonstrations of Critical Driving Behaviours Using Driver's Risk Field

MSc Thesis

Yurui Du



TU Delft

Learning from Demonstrations of Critical Driving Behaviours Using Driver's Risk Field

MSc Thesis

by

Yurui Du

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Thursday October 20, 2022 at 13:00 PM.

Student number: 5217849
Project duration: October 15, 2021 – August 15, 2022
Thesis committee:

Prof. dr. ir. J. Kober,	TU Delft, supervisor
Dr. T. D. Son,	Siemens, supervisor
F. S. Acerbo,	Siemens, daily supervisor

TU Delft

Abstract

Motion planning and decision-making for autonomous vehicles (AVs) are keys to the AV application stack. In recent years, deep learning (DL) methods are widely studied in both academia and industry to provide AVs' motion planning and decision-making solutions due to their capacity to approximate complex mapping between given input and output (e.g. from raw sensor input to steering actions for wheels and throttle). However, these deep learning models are often criticised for issues such as sample inefficiency and low generalisation to safety-critical traffic scenarios. To address these issues, this work proposed to build a model-based multi-agent traffic simulator to efficiently train and validate imitation learning models in critical scenarios with adversarial agents controlled by driver's risk field (DRF). We demonstrate that our approach helps to promote driving safety in critical scenarios and outperforms Lyft Urban Driver (current state-of-the-art) even with >30 times less training resource.

Acknowledgements

I carried out this work during my internship-thesis project at Siemens Digital Industries Software, Leuven, Belgium. I would like to thank Dr. Son Tong and Flavia Acerbo, my supervisors at Siemens for their excellent suggestions and guidance throughout my stay in Leuven. Furthermore, I would like to express my gratitude for Prof. Kober, my supervisor at TU Delft, who showed great interest in the research topic and my progress even before my internship started and provided insightful feedback in our meetings. The support from my supervisors always inspired and motivated me to push my work to a higher level.

Paper

Learning from Demonstrations of Critical Driving Behaviours Using Driver's Risk Field

Yurui Du[1,2], Flavia Sofia Acerbo[2], Jens Kober[1], Tong Duy Son[2]

***Abstract*—In recent years, imitation learning (IL) has been widely used in industry as the core of autonomous vehicle (AV) planning modules. However, previous work on IL planners shows sample inefficiency and low generalisation in safety-critical scenarios, on which they are rarely tested. As a result, IL planners can reach a performance plateau where adding more training data ceases to improve the learnt policy. First, our work presents an IL model using spline coefficient parameterisation and offline expert queries to enhance safety and training efficiency. Then, we expose the weakness of the learnt IL policy by synthetically generating critical scenarios through optimisation of parameters of the driver's risk field (DRF), a parametric human driving behaviour model implemented in a multi-agent traffic simulator based on the Lyft Prediction Dataset. To continuously improve the learnt policy, we retrain the IL model with augmented data. Thanks to the expressivity and interpretability of the DRF, the desired driving behaviours can be encoded and aggregated to the original training data. Our work constitutes a full development cycle that can efficiently and continuously improve the learnt IL policies in closed-loop. Finally, we show that our IL planner developed with 30 times less training resource still has superior performance compared to the previous state-of-the-art.**

***Index Terms*—imitation learning, autonomous driving, critical scenario generation, model-based multi-agent simulator**

I. Introduction

Today, autonomous vehicles (AVs) worldwide are undergoing extensive road tests in the real world, and some have already been put in active service. However, Level 4+ autonomous driving still remains a significant challenge due to the "long tail" of real-world driving events, meaning AVs can be unsafe in rarely occurring traffic scenarios [1]. In the AV application stack, the motion planning module is the key to solving this bottleneck as it determines the AV's driving policy. In recent years, imitation learning (IL) has been widely used as the core planner by learning from large-scale driving datasets of expert demonstrations [2]–[6]. Academic and industrial research has produced state-of-the-art IL-based AV applications in various real-world traffic scenarios, such as unsigned rural roads [2], highways [3], and urban driving [4]–[6].

Despite the growing use of IL as the planner in AVs' planning module, we observed that IL models often require an excessive amount of training resource in order to achieve capable, but sometimes unsafe driving behaviours [5], [6]. To enhance training efficiency and driving safety, we utilise the spline parameterisation for the IL model's predicted trajectory as proposed by our previous work [7] and an offline expert query approach for IL [6].

However, validation of IL models under simulated critical traffic scenarios is largely missing in published research. Most IL models are validated with log-replay data, where the traffic agents' trajectories are logged, and dynamics between traffic agents are not considered. To address this problem, recent research proposed various methods [8]–[11] to build reactive simulations with traffic agents that respond to others. However, these simulated traffic scenarios are close to the scenarios in the training distribution. Therefore, they are not representative of critical traffic scenarios. Critical scenarios can also be manually designed by human experts, but this approach scales poorly. For example [12], by assigning waypoints and multiple available actions for each agent to choose from during the rollout of their policies, the most adversarial configuration can be found using search algorithms. However, the time complexity of this approach grows exponentially with the number of traffic agents, designed waypoints and actions. For this reason, this approach is not scalable to generate highly complex urban traffic scenarios with multiple traffic agents. Furthermore, the diverse driving styles of traffic agents in the real world are not considered. Therefore, the obtained critical traffic scenarios cannot satisfactorily represent the diversity and complexity of real-world driving.

To generate realistic, complex critical scenarios that can help us discover weak driving policies in validation, we utilise driver's risk field (DRF) [13], a parametric model that explains human driving behaviours using the driver's subjective perceived risk of the environment. Compared to other driver models, DRF is a unified theory that models all human driving behaviours, allowing us to represent different driving behaviours by tuning very few parameters without switching between different models. We use DRF as traffic agents in a model-based multi-agent simulator based on the Lyft Prediction Dataset. By optimising parameters of DRF, critical traffic scenarios with realistic and diverse agents can be generated on a large scale.

Another bottleneck for IL is that increasing size of dataset does not necessarily improves IL models' robustness and safety [6]. This may indicate that IL models can reach a performance plateau during training and stop learning from normal traffic data. To continuously improve pre-trained IL models, we present a novel and flexible data augmentation method in which the DRF is exploited to encode desired

*This work was carried out within the thesis internship of Yurui Du at Siemens.

[1]Department of Cognitive Robotics, Delft University of Technology, Delft, the Netherlands, y.du-7@student.tudelft.nl, j.kober@tudelft.nl

[2]Siemens Digital Industries Software, Leuven, Belgium, {flavia.acerbo, son.tong}@siemens.com

driving behaviours in the original training data to improve poorly trained IL policies exposed in the validation results from critical scenarios.

Our contributions are three-fold, as summarised in Fig. 1:

1) An IL method combining spline coefficient parameterisation with the closed-loop offline expert query approach for efficient training. We demonstrate its superior performance over existing methods by validating it in large urban driving datasets and our generated critical traffic scenarios.
2) Scalable generation of realistic and critical traffic scenarios in an interactive DRF-based traffic simulator to test ego driving policies in validation.
3) A novel data augmentation method leveraging DRF and original expert demonstrations based on validation results from critical scenarios to continuously help our IL model learn critical driving behaviours to enhance driving policies, which further improves its safety in both recorded scenarios from logged data and our generated critical scenarios.

II. Related Works

A. Imitation learning

Compared to optimisation-based motion planners, IL is most attractive for its scalability to integrate new functionalities by learning from expert demonstrations rather than optimising human engineered objective functions. With the availability of large-scale driving datasets, IL is becoming a popular method for motion planning in AV industry. However, one major challenge of IL is the distributional shift, which is often caused by the compounding error in the sequential decision-making process, such as motion planning for AVs. It leads the ego vehicle to unfamiliar scenarios that are not included in the training distribution. Eventually, the behaviour of the ego vehicle becomes completely unpredictable and unsafe due to large deviations from the demonstration.

In practice, many approaches have been proposed to mitigate the distributional shift and significantly improve IL performance. While these approaches may seem very different, they mostly mitigate the distributional shift by providing corrective actions during training so the ego vehicles learns to recover from earlier deviations in the sequential decision-making process. One approach [5] is to leverage simple behaviour cloning with data augmentation by adding perturbation noise to provide more robust driving policies. Similarly, another approach [3] tries to directly label perturbed camera images with corrective actions to avoid drifting. However, these approaches generally depend on empirical experiences to engineer noise mechanisms before training. A more theoretically satisfying approach is the dataset aggregation (DAgger) [19], [6], which generates the training distribution of corrective actions on the run and guarantees an ideal linear regret bound to mitigate the distributional shift. However, it also exerts a heavier computation burden. To improve training efficiency, spline parameterisation, a powerful representation of predicted trajectories for IL has been proposed [7].

B. Critical scenario generation

Prior to our work, critical scenario generation has been studied in [12], in which critical scenario generation is approached by optimising a cost-to-go function that maximises the number of collisions and minimises the distance between traffic agents. However, the excessive manual labour required and the heavy computation burden greatly impair its scalability. Other works on traffic scenario generation and reactive simulation with interactive agents mostly used generative methods such as latent variable models [9], autoregressive models [10], and generative adversarial imitation learning (GAIL) [8], in order to capture the possibility of multiple futures. However, these works mainly focused on generating similar traffic scenarios or agents with similar driving policies as demonstrated in the original dataset. Therefore, the generated scenarios are not critical scenarios that are purposefully designed to challenge weak driving policies in validation.

C. Realistic traffic agent modelling

The aforementioned critical scenario generation is implemented in a model-based multi-agent traffic simulator for better scalability. To incorporate traffic agents with realistic human driving behaviours in a simulator, the choice of traffic agent models is essential. Over the decades, numerous models have been proposed to explain human driving behaviours, which can be categorised into learning-based and knowledge-based ones. For learning-based approaches, the driver model relies on a large amount of data to learn policies that behave like human driving. Some literature also refers to this kind of models as non-parametric models because its model structure is not fixed and should be determined from the data [14]. Whereas for the knowledge-based models, sometimes also known as parametric models, the driver behaviour model is often built from prior expert knowledge to capture human driving features. The prior expert knowledge is often formulated in mathematically analytical forms, in which the parameters can be identified by fitting the model to the given data.

Although learning-based models in theory can mimic any human driving behaviour if given enough data, their non-interpretable parameters make the decision-making process a blackbox, which can be detrimental for safety-critical applications requiring traffic agents to reasonably interact with the ego vehicle in a well-controlled manner. Another shortcoming of learning-based models is that we cannot easily and intuitively change their driving behaviours due to their non-interpretable parameters. For example, if we need to model cautious and sporty driving behaviours, a huge amount of data from cautious and sporty drivers is required to train both models. Moreover, as diverse behaviours for different traffic agents need to be modelled individually to make realistic simulations, it will be extremely expensive to train learning-based models for all driving behaviours. For the reasons mentioned above, learning-based models are not suitable for creating realistic scenarios with many diverse traffic agents.

Compared to learning-based models, building knowledge-based human driving behaviour models requires a considerably

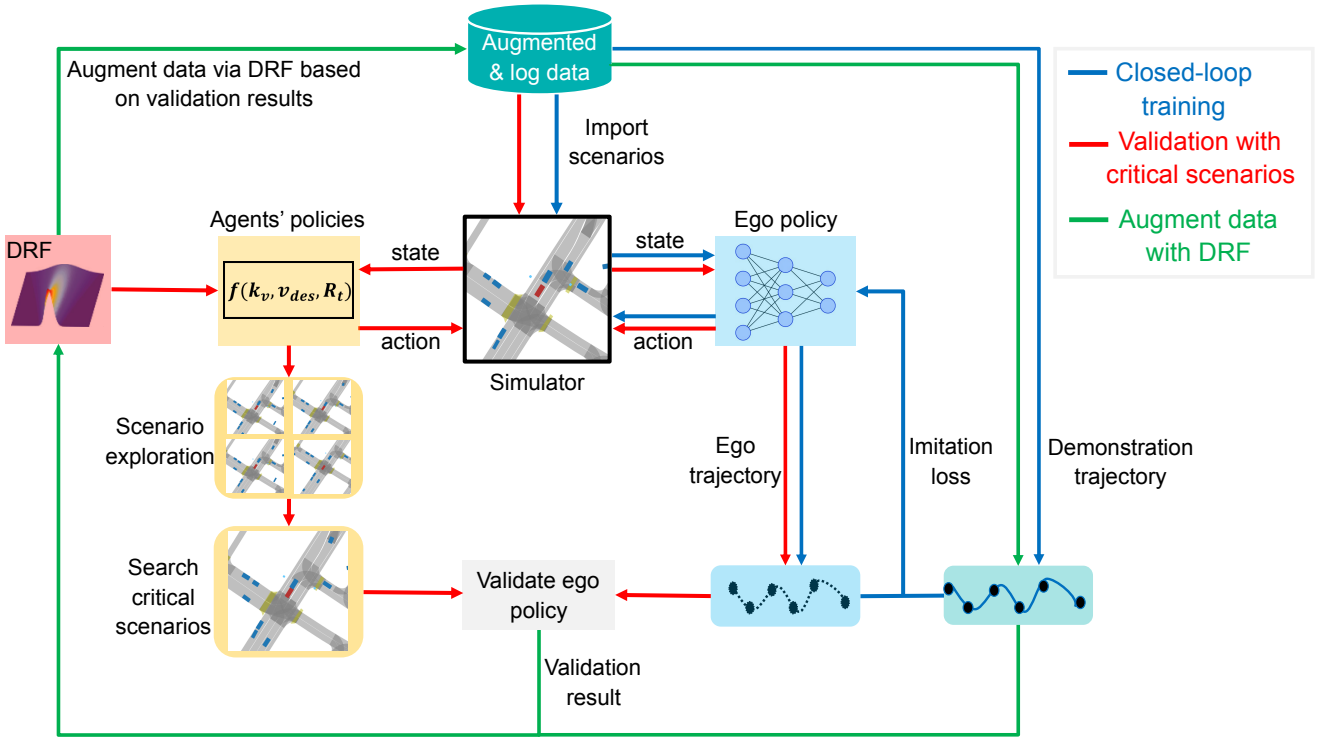


Fig. 1: Overview of our three-part work that respectively addresses a safe, efficient IL method, generation of critical scenarios for validation, and data augmentation encoding desired driving behaviours via DRF. All three parts constitute a development cycle that allows us to continuously improve IL policies in closed-loop.

smaller amount of parameters and these parameters often have clear physical and mathematical meaning, making it easier to interpret and control behaviours of the model. However, most knowledge-based human behaviour driving models are highly scene-specific, assuming the model works only for a specific driving scenario, such as car following in free roads [15] or multi-lane highways [16]. In addition, such models often have strong built-in assumptions connecting certain driving behaviours to specific driving scenarios. For example, time to collision (TTC) is widely used to describe ego vehicle approaching obstacles [17]. Time headway (THW), on the other hand, is specially put forward for car following [18]. These fragmented methods to model driving different behaviours are by nature flawed because real driving scenarios are highly complex. Therefore, it is difficult to identify all possible traffic scenarios and design smooth transitions for them.

For these reasons, the driver's risk field (DRF) [13], a parametric human driving behaviour model, is especially suitable for realistic agent modelling because:

1) It provides the driver's subjective view of the driving risk in any given scenario.
2) It can explain diverse driving behaviours with a unified theory.
3) It has interpretable parameters tunable to mimic diverse driving behaviours.

III. Methods

In this section, we first specify the formulation of our IL method for the ego vehicle. Then, the parametric modelling of other traffic agents using DRF is discussed. We propose our method to generate critical scenarios with DRF agents that act adversarially to challenge the IL policy in validation. Finally, we present a novel data augmentation method that encodes demonstrations of critical driving behaviours to purposefully improve weak IL policies exposed in critical scenarios.

A. Efficient IL with spline coefficient parameterisation and closed-loop offline expert query

IL is a supervised learning method that aims to directly mimic driving behaviours from expert demonstrations. In the context of IL, the expert policy is defined as $a_t^* = \pi^*(s_t)$, i.e., the mapping from an agent's states to its actions. To learn such a mapping, the naive IL is to collect a dataset of state-action pairs $D^* = \{(s_1, a_1)...(s_n, a_n)\}$ from expert demonstrations and learn the policy directly via supervised learning. The parameter of the policy network is denoted by θ, which can be learnt by minimising the loss function L:

$$\hat{\theta} = \arg\min_{\theta} \sum_{(s_i,a_i)\in D^*} L(\pi_\theta(s_i), a_i^*)\,. \tag{1}$$

In this work, $L(\pi_\theta(s_i), a_i^*) = ||\pi_\theta(s_i) - a_i^*||_1$ is the L1 distance between the learner's action $\pi_\theta(s_i)$ and the expert action a_i^*.

To mitigate the distributional shift caused by the naive IL from Eq. (1), we adopt a similar approach to the one proposed in [6], which is itself very similar to DAgger [19], but with better computational efficiency owing to the use of an offline synthetic expert query rather than an active expert policy to aggregate training datasets. This offline expert query approach is achieved by a closed-loop training scheme. Assuming that the dataset D^* consists of N expert trajectories and each trajectory τ has the length of T steps, namely $D^* = \{\tau_i\}_{i=1}^{N}$, $\tau_i = \{(s_j, a_j)\}_{j=1}^{T}$, we first sample the ego vehicle's current policy for K steps, which will lead the ego vehicle to unfamiliar scenarios due to the distributional shift. Then, the current policy is updated by minimising the above loss function in the remaining $T-K$ steps so the ego vehicle learns how to recover from mistakes caused by the distributional shift. The optimisation objective can be rewritten as a discounted cumulative expected loss with a discount factor of γ:

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_{\tau\sim\pi^*} \sum_{t=K}^{T} \gamma^{T-K} L(\pi_\theta(s_t), a_t^*)\,. \tag{2}$$

Similarly to [7], we use spline coefficients, also known as control points, to parameterise trajectories in the dataset D^* instead of using discrete waypoints for more stable long-horizon predictions and smoother trajectories. The details of the spline parameterisation used in this work is provided in Appendix C. We also show that this parameterisation greatly improves the training efficiency in Sec. IV-C.

The states s_t and actions a_t of the original expert trajectories are both denoted as a 3D vector (x, y, θ) in the SE_2 space. The corresponding spline coefficients in all three directions can be expressed as a matrix $\mathrm{A}_{3\times n}$. By replacing a_t^* in Eq. (2) with $\mathrm{A}_{3\times n}^*$, our objective function can be written as:

$$\hat{\theta} = \arg\min_{\theta} \mathbb{E}_{\tau\sim\pi^*} \sum_{t=K}^{T} \gamma^{T-K} L(\pi_\theta(s_t), \mathrm{A}_{3\times n}^*)\,. \tag{3}$$

B. Parametric agent modelling using DRF

The DRF model [13] builds the driver's subjective view of its surrounding environment as a 2D Gaussian distribution along the predicted path. The perceived risk is derived from DRF representing the driver's subjective view of the driving risk in traffic. It is a function of the ego vehicle's current velocity and steering angle $P_{risk}(v, \delta)$. Then, based on the risk threshold theory, the future velocity and steering angle are obtained by solving an optimisation problem to keep the perceived risk below the assigned threshold. Please refer to Appendix. A for detailed formulations of DRF, where typical values of DRF parameters and principles of parameter tuning for different driving behaviours are disclosed.

C. Critical scenario generation

In this part, we detail how to generate critical traffic scenarios with agents that follow DRF policies controlling their velocity profiles along their original trajectories. The agents are designed to react adversarially to the ego vehicle's driving policy by optimising the DRF parameters of agents. The traffic scenarios are initialised based on real-world urban driving data to improve the complexity and realism of the generated scenarios.

We assume $s_t = \{x_t, y_t, \theta_t\}$ to be the state vector of the ego vehicle's pose at time t. This vector includes the 2D position and orientation of the vehicle w.r.t. the ego-centric reference frame at $t = 0$. Let $Z_t = \{z_t^i\}_{i=1}^{M}$ be the state vector that consists of the pose of all other M agent vehicles closest to the ego vehicle, where z_t^i is the state vector of the i^{th} agent vehicle's pose. Let us assume that $a_t = \pi_\theta(s_t)$ is the IL policy of the ego vehicle and $u_t^i = \pi_{\phi^i}(z_t^i)$ is the parametric policy of the i^{th} agent parameterised by ϕ^i, and also the dynamics model of the state of the ego vehicle $s_{t+1} = f(s_t, a_t)$ and of the state of the agent $z_{t+1}^i = h(z_t^i, u_t)$. We can obtain ϕ^i for each agent's parametric policy leading to critical traffic scenarios by optimising the following objective:

$$\Phi^* = \arg\min_{\Phi} J(\theta, \Phi)\,, \tag{4}$$

where $\Phi = \{\phi_i^*\}_{i=1}^{M}$ and $J(\theta, \Phi)$ is the cost-to-go function computed from the scenario via unrolling all vehicles' policies. The cost is computed from the L1 distance between the ego vehicle and other vehicles and the total number of accidents (collisions, off-road incidents) to encourage the formation of dense traffic and collisions:

$$J(\theta, \Phi) = \mathbb{E}_{s_t, Z_t} \sum_{t=0}^{T} L1(s_t, Z_t) - L_{accidents}\,. \tag{5}$$

To ease the computation burden, we assume that each agent can either drive aggressively or cautiously, with aggressive and cautious driving behaviours represented by different values of parameters of DRF. For every scenario, a total of M agents are controlled by DRF, meaning that there are 2^M different combinations of agents' parameters of DRFs that lead to 2^M possible futures. Therefore, the optimal parameter combination corresponding to the most critical traffic scenarios can be obtained with an exhaustive search algorithm. To scalably generate critical scenarios, Simcenter HEEDS, a high-performance, global parameter optimization software, is used to optimise the parameters of DRF. The algorithm of generating critical scenarios is shown in Alg. 1.

D. Data augmentation for desired driving behaviours

Improving the performance of IL models for AVs is difficult, as adding more training data does not guarantee better performance. To address this problem, here we propose to augment the expert demonstrations by altering the ego vehicle's velocity profiles of the expert trajectories with DRF. This method offers great flexibility to encode the desired driving behaviours we wish the IL model to learn. The DRF model ensures the new learnt policy is still applicable to the previous dataset because the DRF-augmented data distribution is similar to human demonstrations.

Other data augmentation methods for IL planning models, such as perturbing the original trajectory with noise [5],

Algorithm 1 Generate critical scenarios in a model-based multi-agent simulator with DRF

$\theta \leftarrow \theta_0$ // ego vehicle's policy
for $i = 1, ..., N$ **do**
  // for each traffic scenario
  // Φ^i are agents' parameters of their DRF policies in the i^{th} scenario
  // Exhaustively search 2^M combinations of agents' DRF parameters $\{\Phi^i_j\}^{2^M}_{j=1}$ to get the optimal combination of agents' parameters leading to the critical sceanario
  $\Phi^i* = \arg\min_{\Phi_j} J(\theta, \Phi^i_j)$
  // J is computed via Eq. (5) by unrolling agents' policies
  // Validate ego policy θ in scenario i with M DRF agents parameterised by Φ^i*
end for
return validation results from critical scenarios

or requiring an expert policy during training [7], although they can significantly improve IL performance, they do not guarantee further improvement by retraining with more data. Furthermore, since they cannot be used to learn desired driving behaviours that purposefully improve previous weak IL policies, the performance usually worsens in critical scenarios where other agents act adversarially. By comparison, our data augmentation method encoding desired driving behaviours can be used to continuously improve poorly trained policies exposed in critical scenarios by learning from DRF-augmented demonstrations.

We have noticed that our model suffers more from rear collisions than from other kinds of violations. Most rear collisions occurred due to the passiveness of the ego vehicle. Passiveness is a common case of causal confusion [8] in IL. To mitigate this problem, we use the DRF model with aggressive parameters to augment the original expert demonstrations in scenarios where the rear vehicle is approaching the ego vehicle as shown in Alg. 2. Please note more flexible augmentation can be achieved by specifying different DRFs and traffic scenario conditions. In Sec. IV-E, we show that the IL model retrained with the augmented data learns a more robust policy and drives less passively, which reduces rear collisions in validation with both recorded scenarios from the logged data and our generated critical scenarios.

IV. Experiments

In this section, we evaluate the three contributions of this paper. In particular, we are interested in: the impact of spline parameterisation on the training efficiency of IL models; the ability of generated critical scenarios to help detect poorly trained policies; and the ability of IL models to learn desired driving behaviours via retraining with DRF-augmented demonstrations.

These three aspects are evaluated in Sec. IV-C, IV-D and IV-E, respectively. The details of the models and scenarios used for validation are listed as follows:

Algorithm 2 Data augmentation encoding desired driving behaviours with DRF

// Dataset D has N expert trajectories corresponding to N scenarios, each trajectory consists of T steps.
$D := \{\tau_i\}^N_{i=1}$, $\tau_i = \{(s_j, a_j)\}^T_{j=1}$
// DRF with aggressive parameters
$DRF \leftarrow DRF_{agg}$
for $i = 1, ..., N$ **do**
  // for each traffic scenario
  if rear vehicle exists **then**
    // Conditional data augmentation
    for $t = 1, ..., T-1$ **do**
      // for each timestep
      $a^{DRF}_t = DRF(s_t)$
      $s^{DRF}_{t+1} = f(s_t, a^{DRF}_t)$ // Update next state
      $s_{t+1} = s^{DRF}_{t+1}$
    end for
    // Aggregate dataset D
    $D \leftarrow \{(s^{DRF}_j, a^{DRF}_j)\}^T_{j=1}$
  end if
end for
// Retrain ego IL policy with aggregated dataset D

1) **Data-efficient IL**
   Models: Our IL model trained 30h with original data with 1 NVIDIA RTX A4000 laptop GPU (our 30h IL model), Lyft Urban Driver [6] trained 30h with original data with 32 Tesla V100 GPUs (Lyft Urban Driver).
   Scenarios: 2500 log-replay scenarios.
2) **Generation of critical traffic scenarios**
   Models: Our 30h model, Lyft Urban Driver.
   Scenarios: 1250 log-replay scenarios, 1250 critical scenarios.
3) **Data augmentation for desired driving behaviours**
   Models: Our 30h model, our IL model trained 30h with original data + 2h retraining with augmented data (our 30h model + 2h retraining), Lyft Urban Driver.
   Scenarios: 1250 log-replay scenarios, 1250 critical scenarios.

For a better benchmark to demonstrate the advantage of using the spline parameterisation to improve training efficiency, our IL model is adapted from Lyft Urban Driver [6], the original work that proposed the closed-loop IL with offline expert query, by adding the spline parameterisation to it, with more details explained in Appendix. B.

A. Data

We use the Lyft Prediction Dataset to train and validate our IL models. This is an urban driving dataset with diverse and complex traffic scenarios. Specifically, we use the 112h training dataset and randomly choose 2500 four-second scenarios from the validation dataset for evaluation. Both training and validation datasets are provided by Lyft.

Both log-replay and generated critical scenarios are used in validation. In log-replay scenarios, the other agents are

TABLE I: Metrics for the baseline and our model from 2500 log-replay scenarios.

Models	**Collision**			**Imitation**	**Aggressive driving**
	Front	Rear	Side	Off-road	
Urban Driver	1	3	**0**	4	140
Ours	**0**	2	**0**	**0**	**109**

following their original trajectories. While in critical ones, the other agents are reactive and following the DRF policy, which controls their velocity profiles along their original trajectories.

B. Metrics

We evaluate all models in closed-loop, meaning that the IL policy takes full control throughout the entire duration of each scenario. For each scenario, we check the following metrics to keep track of the number of violations and events to compare the performance of different IL models.

Safety metrics

· Collisions: Record this violation if the ego vehicle collides with other traffic agents.

Imitation metrics

· Off-road events: Record this violation if the ego vehicle deviates from its ground-truth trajectory by more than 4m in the lateral direction.

Subjective risk metrics

· Aggressive driving: Record this event if the perceived risk (as specified in Sec. III-B) of the ego vehicle is larger than 10^5. This is a comprehensive metric that large risk values can mean a very close distance to other vehicles, which makes the driver feel at risk.

C. Data efficient IL with spline coefficient parameterisation

In this experiment, we evaluate the impact of spline parameterisation on the training efficiency by comparing IL models trained with the original training dataset. The baseline model is Lyft Urban Driver, which is trained 30h with 32 Tesla V100 GPUs. Ours is also trained 30h, but with only 1 NVIDIA RTX A4000 laptop GPU. In Table I, we show that even with $>$30 times less training resource, our model outperforms Urban Driver in all metrics, indicating better performance in safety and imitation. Also, our model has a less aggressive driving style compared to Lyft Urban Driver.

Fig. 2 and Fig. 3 present more qualitative results that compare the performance of our model and Lyft Urban Driver. We show that our model drives safer and obeys traffic rules.

From Fig. 2, we observe that Lyft Urban Driver fails to accelerate in time and does not react to the approaching vehicle from behind, which leads to a rear collision. In comparison, our model makes the correct decision to speed up and drives through the intersection safely.

From Fig. 3, Lyft Urban Driver fails to stop at an intersection when the traffic light is red, leading to a front collision. Whereas our model obeys traffic rules and stops in the same scenario.

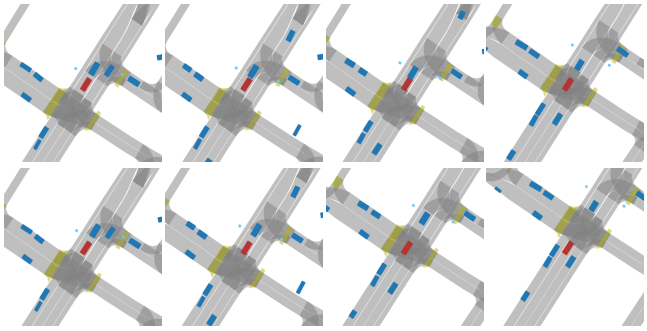


Fig. 2: Each row is a 4-second traffic scenario that consists of five images consecutive in time displaying the rollout of the entire scenario. Red rectangle is the ego vehicle controlled by IL policies, and blue rectangles are other traffic agents from Lyft data log. Top row: Lyft Urban Driver has from rear collision due to passive driving behaviour at an intersection. Bottom row: Our IL policy accelerates the ego vehicle in time given the same scenario.

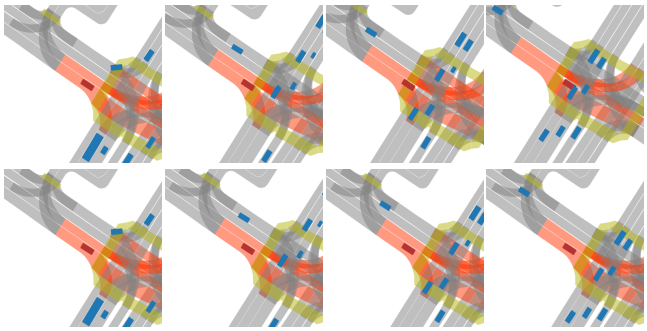


Fig. 3: Top row: Lyft Urban Driver runs red traffic light and has front collision. Bottom row: Our IL model stops at red light in the same scenario.

D. Critical traffic scenario generation

In this experiment, we compare our generated critical traffic scenarios with reactive adversarial agents to original traffic scenarios with log-replay agents by unrolling our ego IL policy and Lyft Urban Driver in both kinds of scenarios and comparing their performance. It is shown in TABLE II that our generated critical scenarios are more challenging for the IL models (both Lyft Urban Driver and ours) to handle, as the number of collisions increases in critical scenarios.

Additionally, more aggressive driving is observed in critical scenarios. This is because the distance between agents is smaller in critical scenarios. Therefore, the ego vehicle subjectively "feels" more at risk driving in critical scenarios.

Fig. 4 qualitatively illustrates how our critical scenarios expose the weakness of our IL policy. On the top row, we see the ego vehicle drives passively and is not reactive enough to the approaching rear vehicle but no violation is reported as no collision occurs, making the problem difficult to be noticed. This problem is often referred to as passiveness due to causal confusion [20]. By comparison, our critical scenario easily

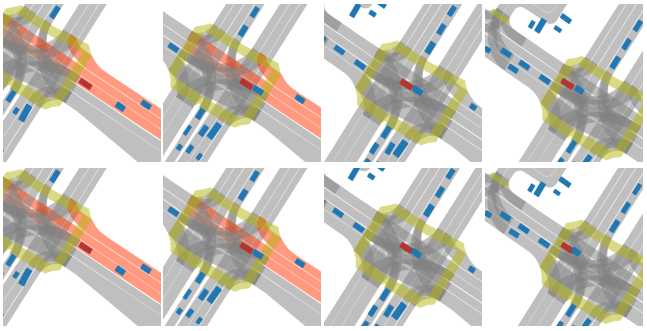


Fig. 4: Red rectangle is the ego vehicle controlled by our IL policy, and blue rectangles are other traffic agents controlled by DRFs leading to critical scenarios. Top row: Validation of our IL model using original Lyft data log, where the ego vehicle drives passively, leading to a near-collision scenario but is not detected. Bottom row: Validation of our IL model using our generated critical scenario, where the ego vehicle has a rear collision due to passive driving.

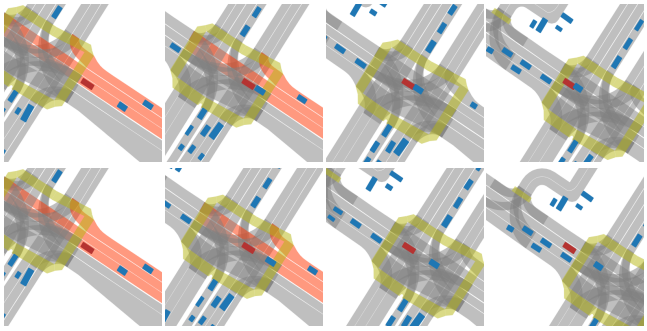


Fig. 5: Red rectangle is the ego vehicle controlled by our IL policy, and blue rectangles are other traffic agents controlled by DRFs leading to critical scenarios. Top row: Validation of our IL model using the same critical scenario from Fig. 4, where the ego vehicle has a rear collision due to passiveness and inaction to the approaching rear vehicle. Bottom row: Validation of our IL model retrained with augmented data demonstrating less passive driving behaviours using the same generated critical scenario, where the ego vehicle correctly responds to the approaching rear vehicle and keeps safe longitudinal distance due to reduced passiveness.

TABLE II: Metrics for the baseline and our (retrained) model from 1250 log-replay and critical scenarios.

Scenarios	**Models**	**Collision**			**Imitation**	**Aggressive driving**
		Front	Rear	Side	Off-road	
Log-replay	Urban Driver	**0**	**1**	**0**	**0**	71
	Ours	**0**	**1**	**0**	**0**	**56**
	Ours(Re)	**0**	**1**	**0**	**0**	100
Critical	Urban Driver	**0**	8	**0**	6	71
	Ours	**0**	7	1	**0**	**60**
	Ours(Re)	**0**	**5**	**0**	**0**	111

reveals the passiveness of our IL policy by a rear collision with an aggressive rear agent.

For this particular case, some may argue that introducing another metric that marks close distance between vehicles below a set threshold as a violation can also help to expose passiveness. However, this threshold is difficult to define because close distance between vehicles is very common in dense urban traffic, where lots of false positives are likely to be reported. Therefore, we argue that our generated critical scenarios make it easier to disclose weak driving policies.

E. Data augmentation for desired driving behaviours

From previous validation results from Fig. 4, we observed passiveness and inaction to approaching rear vehicles from our IL policy due to causal confusion. This is a common mistake of IL models for autonomous driving [21]. To mitigate passiveness of the ego vehicle, we retrain the IL model with augmented data where the ego vehicle drives slightly faster when the rear vehicles are approaching.

In this experiment, we aim to demonstrate the ability of IL models to learn desire driving behaviours from DRF-augmented demonstrations by comparing the performance of our retrained IL model (30h training + 2h retraining), our IL model (30h training), and Lyft Urban Driver in both log-replay and critical traffic scenarios.

In TABLE II, we see that our retrained IL model performs better in critical scenarios and equally in log-replay scenarios regarding the collision and imitation metrics.

The reason for the significant increase in aggressive driving from our retrained model is that the retrained model learns more aggressive driving, so it will slightly pick up speed if followed by a rear agent. Therefore, the distance between itself and front vehicles is shorter, leading to higher subjective perceived risk. However, no front collisions occur, indicating that the ego vehicle learns to drive less passively to avoid rear collisions without compromising other safety metrics. Also, our retrained model performs well in both critical and log-replay scenarios, meaning the policy learned via retraining is robust enough to handle both normal and critical scenarios with adversarial agents. More importantly, we show that driving styles of IL models can be properly customised using DRF without compromising driving safety.

Fig. 5 presents validation results comparing our IL model before and after retraining with DRF-augmented data to alleviate passiveness. The shown traffic scenario is the same as the one we generated from Fig. 4. On the top row, we see that our IL model before retraining has a rear collision due to passive driving. However, as shown in the bottom row, our retrained model speeds up in time and safely passes the intersection without noticeable sign of passiveness, indicating it has learnt the desired driving policy that drives slightly more aggressively if followed by a rear vehicle.

V. Conclusions

In this paper, we have demonstrated the potential of incorporating the DRF, a parametric human driving behaviour model, in a multi-agent traffic simulator to build a full development

cycle that can continuously improve the performance of IL models. With the expressivity and interpretability of the DRF, we can generate critical scenarios with DRF-based agents that are parameterised to act adversarially to the ego IL policy. These generated critical scenarios are proven to be more challenging for the ego IL policy to handle than the recorded scenarios from the logged data. Moreover, weak policies are more easily detected from the validation with critical scenarios. To enhance the weak policy, we use DRF to encode desired driving behaviours to augment the expert demonstrations. By retraining the IL model with augmented data, the IL model achieves safer driving. The IL policy learnt via retraining is also more robust as it is applicable to both critical scenarios with adversarial agents and recorded scenarios from logged data. We also proposed to improve (re)training efficiency by adding the spline parameterisation to Lyft Urban Driver. We show that our model, even developed with 30 times less training resource, outperforms Lyft Urban Driver.

ACKNOWLEDGEMENT

This work is part of FOCETA project that has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 956123.

REFERENCES

[1] Ashesh Jain, Luca Del Pero, Hugo Grimmett, and Peter Ondruska. Autonomy 2.0: Why is self-driving always 5 years away? 2021.

[2] Dean Pomerleau. Alvinn: An autonomous land vehicle in a neural network. In D.S. Touretzky, editor, Proceedings of (NeurIPS) Neural Information Processing Systems, pages 305 – 313. Morgan Kaufmann, December 1989

[3] M. Bojarski et al., "End to End Learning for Self-Driving Cars." arXiv, Apr. 25, 2016. Accessed: Aug. 24, 2022. [Online]. Available: http://arxiv.org/abs/1604.07316

[4] J. Hawke et al., "Urban Driving with Conditional Imitation Learning," in 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, May 2020, pp. 251–257. doi: 10.1109/ICRA40945.2020.9197408.

[5] M. Bansal, A. Krizhevsky, and A. Ogale, "ChauffeurNet: Learning to Drive by Imitating the Best and Synthesizing the Worst," Jun. 2019. doi: 10.15607/RSS.2019.XV.031.

[6] O. Scheel, L. Bergamini, M. Wołczyk, B. Osiński, and P. Ondruska, "Urban Driver: Learning to Drive from Real-world Demonstrations Using Policy Gradients." arXiv, Sep. 27, 2021. Accessed: Aug. 24, 2022. [Online]. Available: http://arxiv.org/abs/2109.13333

[7] F. S. Acerbo, H. Van der Auweraer, and T. Duy Son, "Safe and Computational Efficient Imitation Learning for Autonomous Vehicle Driving," in 2020 American Control Conference (ACC), Denver, CO, USA, Jul. 2020, pp. 647–652. doi: 10.23919/ACC45564.2020.9147256.

[8] L. Bergamini et al., "SimNet: Learning Reactive Self-driving Simulations from Real-world Observations," in 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China, May 2021, pp. 5119–5125. doi: 10.1109/ICRA48506.2021.9561666.

[9] S. Suo, S. Regalado, S. Casas, and R. Urtasun, "TrafficSim: Learning to Simulate Realistic Multi-Agent Behaviors," in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, Jun. 2021, pp. 10395–10404. doi: 10.1109/CVPR46437.2021.01026.

[10] S. Tan, K. Wong, S. Wang, S. Manivasagam, M. Ren, and R. Urtasun, "SceneGen: Learning to Generate Realistic Traffic Scenes," in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Nashville, TN, USA, Jun. 2021, pp. 892–901. doi: 10.1109/CVPR46437.2021.00095.

[11] M. Igl et al., "Symphony: Learning Realistic and Diverse Agents for Autonomous Driving Simulation," in 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, May 2022, pp. 2445–2451. doi: 10.1109/ICRA46639.2022.9811990.

[12] Y. Abeysirigoonawardena, F. Shkurti, and G. Dudek, "Generating Adversarial Driving Scenarios in High-Fidelity Simulators," in 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, May 2019, pp. 8271–8277. doi: 10.1109/ICRA.2019.8793740.

[13] S. Kolekar, J. de Winter, and D. Abbink, "Human-like driving behaviour emerges from a risk-based driver model," Nat Commun, vol. 11, no. 1, p. 4850, Dec. 2020, doi: 10.1038/s41467-020-18353-4.

[14] S. Lefevre, C. Sun, R. Bajcsy, and C. Laugier, "Comparison of parametric and non-parametric approaches for vehicle speed prediction," in 2014 American Control Conference, Portland, OR, USA, Jun. 2014, pp. 3494–3499. doi: 10.1109/ACC.2014.6858871.

[15] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," Phys. Rev. E, vol. 62, no. 2, pp. 1805–1824, Aug. 2000, doi: 10.1103/PhysRevE.62.1805.

[16] A. Kesting, M. Treiber, and D. Helbing, "General Lane-Changing Model MOBIL for Car-Following Models," Transportation Research Record, vol. 1999, no. 1, pp. 86–94, Jan. 2007, doi: 10.3141/1999-10.

[17] D. N. Lee, "A Theory of Visual Control of Braking Based on Information about Time-to-Collision," Perception, vol. 5, no. 4, pp. 437–459, Dec. 1976, doi: 10.1068/p050437.

[18] W. Van Winsum and H. Godthelp, "Speed Choice and Steering Behavior in Curve Driving," Hum Factors, vol. 38, no. 3, pp. 434–441, Sep. 1996, doi: 10.1518/001872096778701926.

[19] S. Ross, G. J. Gordon, and J. A. Bagnell, "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning." arXiv, Mar. 16, 2011. Accessed: Aug. 24, 2022. [Online]. Available: http://arxiv.org/abs/1011.0686

[20] P. de Haan, D. Jayaraman, and S. Levine, "Causal Confusion in Imitation Learning." arXiv, Nov. 04, 2019. Accessed: Aug. 28, 2022. [Online]. Available: http://arxiv.org/abs/1905.11979

[21] M. Vitelli et al., "SafetyNet: Safe Planning for Real-World Self-Driving Vehicles Using Machine-Learned Policies," in 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, May 2022, pp. 897–904. doi: 10.1109/ICRA46639.2022.9811576.

APPENDIX A
DRF TRAFFIC AGENT MODELLING AND VALIDATION

Our DRF-based traffic agent model is adapted from [13] with modifications to the mathematical formulation of DRF and its corresponding controllers. In the next sections, we present the modelling details of our DRF-based traffic agent as well as validation results using the Lyft Prediction Dataset.

A. Formulation of DRF-based traffic agent models

The DRF-based traffic agent model consists of the subjective DRF map of the driver and the objective cost map of the environment. The subjective DRF map of the driver is modelled as a 2D Gaussian distribution along the predicted trajectory during the period of look-ahead time t_{la} of the ego vehicle. The equations of the predicted path and the corresponding Gaussian at each cross section along the predicted path are defined as:

$$R_{car} = \frac{L}{\tan\delta}\,, \tag{6}$$

$$d_{la} = vt_{la} + d_s\,, \tag{7}$$

$$G(x,y) = a(s)\exp\left(-\frac{\sqrt{(x-x_c)^2+(y-y_c)^2}-R_{car}}{2\sigma^2}\right), \tag{8}$$

$$a(s) = p(s-d_{la})^2\,, \tag{9}$$

$$\sigma_i = (m+k_i|\delta|)s + c\,, \quad i = 1(\text{inner}), 2(\text{outer})\,. \tag{10}$$

R_{car} is the radius of the ego vehicle's predicted trajectory computed from the its wheel-base L and steering angle δ. d_{la} is the look-ahead distance computed from the vehicle velocity v, look-ahead time t_{la}, and the safety distance d_s. Please note that d_s is not in the original formulation [13], and our reasons to add d_s is given in later parts of this appendix. Eq. (9) and (10) computes the height and width of the 2D Gaussian distribution as a function of the arc length s along the predicted trajectory, with $s = vt$. The height of the DRF is modelled as a parabola with the parameter p denoting the steepness of the parabola. The width of the DRF is modelled to increase linearly with s. c is the width of the DRF at the current position of the ego vehicle (where $s = 0$), which is equal to car width / 4 ($\pm 2\sigma$ covers 95% of Gaussian distribution). m defines the slope of widening of the DRF. k_1, k_2 respectively defines the inner and outer boundaries of the DRF, with its width changing proportionally to the absolute value of the steering angle. Intuitively, larger values of DRF parameters $p, d_s, t_{la}, c, m, k_1, k_2$ increases the perceptive field of the DRF, meaning that given the same traffic scenario, the driver with larger values of parameters of the DRF perceives higher risk compared to the driver with smaller parameter values.

The DRF-based traffic agent model consists of the subjective DRF map of the driver and the objective cost map of the environment. The values of the Gaussian distribution $G(x, y)$ indicate the probability that the ego vehicle is in (x, y) in the next step. The mathematical formulations of the DRF [13] are given in Sec. III-B. The objective cost map of the environment $C(x, y)$ models the consequence of the ego vehicle being in (x, y) at the next timestep. In our work, we assign penalties of 2500 for obstacles in (x, y) and penalties of 500 for non-drivable areas.

In short, the subjective DRF map of the driver quantified by seven parameters $(p, d_s, m, c, t_{la}, k_1, k_2)$ is only dependent on the driver's state of mind, not the environment. Whereas the objective cost map of the environment is independent from the subjective view of the driver and hence is the same for everyone.

The perceived risk of the driver is the convolution product of the subjective DRF map of the driver and the objective cost map of the environment, as shown in Fig 6. In other words, the perceived risk is computed by the consequence of an event multiplying the probability of the event occurring, or the occurring event's expected consequence. As many potential events can occur in the environment, the perceived risk is the summation of all events' expected consequence.

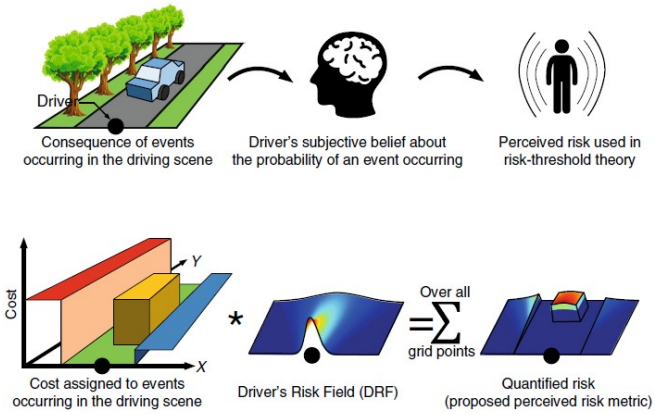


Fig. 6: An illustration of driver's perceived risk from [13]. Top row: qualitative formulation of the perceived risk. Bottom row: quantification of the perceived risk

In [13], the perceived risk is derived as a function of the current velocity and steering angle of the ego vehicle $P_{risk}(v, \delta)$. Then, based on the risk threshold theory, the future velocity and steering angle are optimised to keep the perceived risk below the assigned threshold. We adopt similar approaches to compute future actions of the DRF model. However, the optimisation here is only used to compute the next velocity of the ego vehicle and assume that it follows the ground-truth trajectory, i.e., assuming steering angle is known. We did such simplification for the following considerations:

1) Avoid high-level decisions: The Lyft Prediction Dataset is an urban driving dataset with complex high-level decisions such as go straight/turn left/turn right at an intersection. These high-level decisions depend on global route planning and cannot be modelled by DRF. Therefore, assuming that ego vehicle follows the ground-truth trajectory and only leaving the lower-level velocity for the DRF model to control is a more reasonable approach.
2) Underdefinition for complex urban driving: The quantified DRF model and perceived risk theory are proposed to explain human driving behaviours in an unlimited, natural setting. In urban traffic, drivers are limited by many traffic rules and conditions, such as red lights, one-way signs, and different speed limit signs. Making DRF applicable to urban traffic is a difficult task that requires incorporating hard-coded rules with the existing DRF model, which is not in the scope of this paper.

The detailed structure of the DRF-based traffic agent model is shown in Fig. 7. Fig. 7a presents the overall control structure of the DRF-based traffic agent: The agent leverages the cost map of the driving scenario, and the vehicle states (position: x_k, y_k; heading: ϕ_k; and speed: v_k) at the k^{th} timestep to generate the steering angle δ_{k+1} and velocity v_{k+1} for the $(k+1)^{th}$ timestep. Fig. 7b shows the structure of the DRF-based traffic agent model: The DRF block computes the driver's subjective view of the environment using the current state of the ego vehicle. The perceived risk is computed from

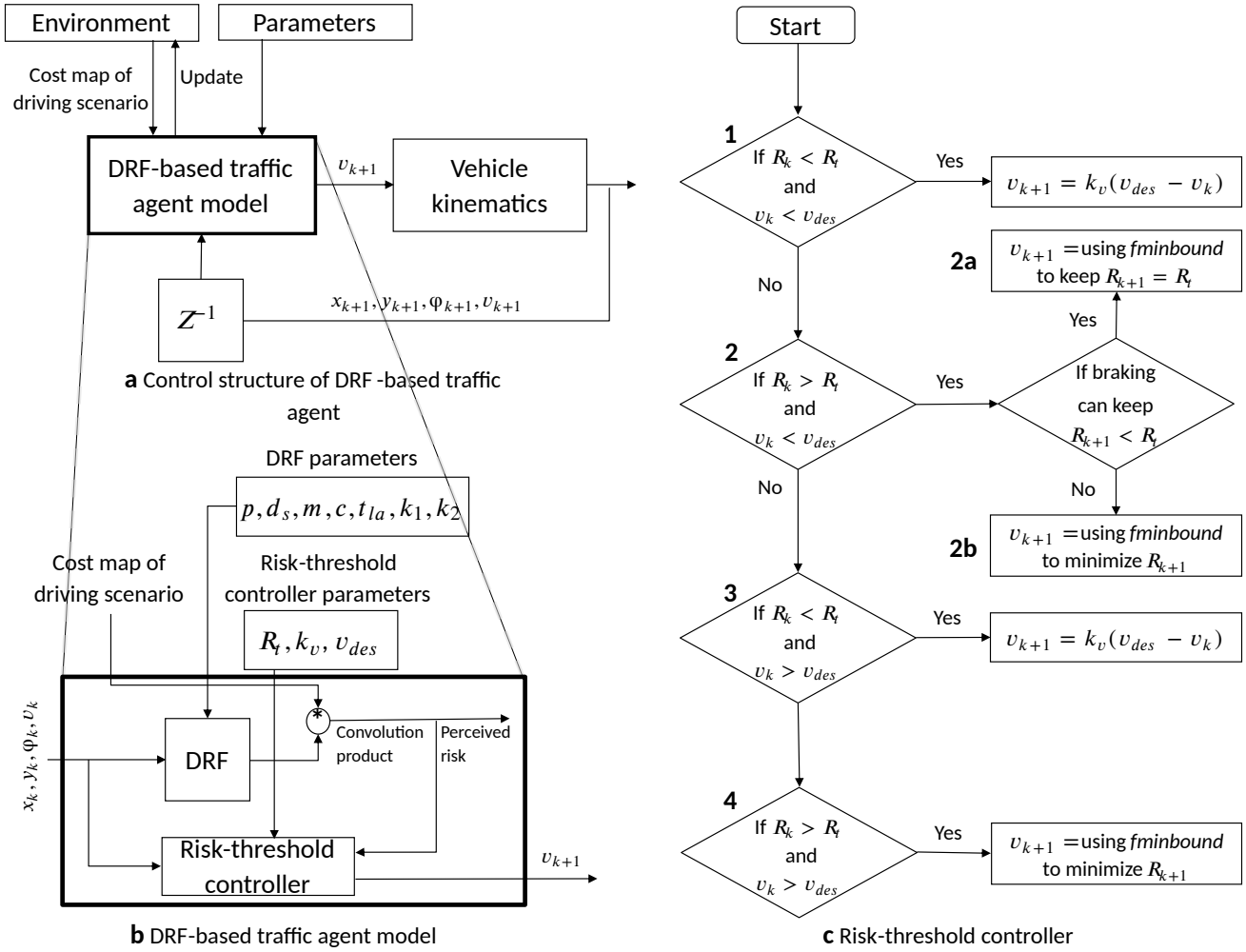


Fig. 7: The structure of the DRF-based traffic agent model adapted from [13].

the convolution product of the DRF and objective cost map. The perceived risk and the state of the ego vehicle are inputs to the risk-threshold controller in Fig. 7c. takes in the computed perceived risk, and the ego vehicle states to generate the velocity v_{k+1} for next timestep. The risk-threshold controller tries to keep the perceived risk below the given risk threshold R_t. At each time step k, It compares the input perceived risk R_k to risk threshold R_t, and velocity v_k to the assigned desired velocity v_{des}. This results in four distinct cases of inequality for the controller to operate on:

1) $R_k < R_t$ and $v_k < v_{des}$: This condition means that the perceived risk is lower than the risk threshold and the velocity is lower than the desired velocity. Therefore, no special action is needed to keep the perceived risk below the threshold and the ego vehicle should accelerate to reach the desired velocity. In this work, a simple P-controller is used with k_v representing how aggressively the driver accelerates. Hence, $v_{k+1} = v_k + k_v(v_{des} - v_k)$.
2) $R_k > R_t$ and $v_k < v_{des}$: In this case, the perceived risk is higher than the risk threshold, while the desired velocity is not achieved. Therefore, we need to check if the perceived risk can be reduced to the threshold. The check is carried out using `fminbound` to find a velocity v_{op} in $(v_k - a_{max}dt, v_k + a_{max}dt)$ that can reduce the perceived risk to the threshold, where a_{max} is the maximum acceleration of the vehicle ego and dt is the duration of a timestep. In this work, $a_{max} = 4m/s^2$ and $dt = 0.1s$. This search leads to two possible outcomes for the controller to handle accordingly.
   2a) If v_{op} exists, the driver should try to reach v_{op}. Hence, $v_{k+1} = v_k + k_v(v_{op} - v_k)$.
   2b) If v_{op} does not exist, the driver should decelerate to the velocity v_{min} that minimises the perceived risk. Hence, $v_{k+1} = v_k + k_v(v_{min} - v_k)$.
3) $R_k < R_t$ and $v_k > v_{des}$: In this case, the perceived risk is safely below the threshold, but the velocity exceeds the desired velocity. Therefore, the driver should slow down to approach the desired velocity. $v_{k+1} = v_k + k_v(v_{des} - v_k)$.
4) $R_k > R_t$ and $v_k > v_{des}$: In this case, both the perceived

risk and the velocity exceeds the assigned goal. Therefore, the driver should slow down to the velocity v_{min} that minimises the perceived risk. $v_{k+1} = v_k + k_v(v_{min} - v_k)$.

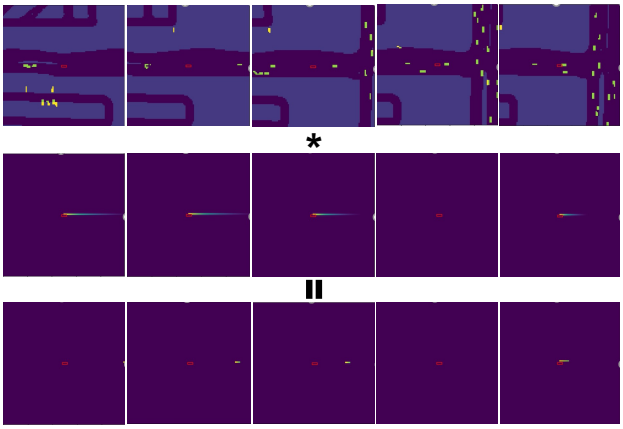


Fig. 8: The traffic scenario with the original DRF-based model [13] (With its location marked by the red box in the centre, please be aware the red box is not included in the actual maps.) and other agents from the log-replay (yellow boxes). Top row: The objective cost map the environment. Middle row: The subjective DRF map of the DRF-based model. Bottom row: The perceived risk map computed from the convolution product of the objective cost map and the subjective DRF map. Brighter colours indicate higher values in the maps while darker colours denote lower values.

It is worth noting that the safety distance d_s is added in Eq. (7) so that the look-ahead distance $d_{la} = vt_{la} + d_s$. The rationale behind this change is given as follows: Let us assume that the look-ahead distance is $d_{la} = vt_{la}$. When the ego vehicle's velocity v is approaching 0, the predicted trajectory length $s = vt, 0 < t < t_{la}$, along with look-ahead distance $d_{la} = vt_{la}$ would also approach 0. As a result, the height of the DRF, $a(s) = p(s - d_{la})^2$ would also be 0, meaning that the subjective DRF map would consist of zeros and the perceived risk computed from the convolution product of the subjective DRF map and the objective environment cost map would also be zero. In this sense, the driver cannot perceive the risk of the environment. Therefore, the vehicle will start to accelerate and will never be able to stop. This loophole can be detrimental in scenarios where the ego vehicle must brake and stop to avoid front collisions, as shown in Fig. 8. From the top row showing the objective cost map of the environment, we see that the DRF-based model (red box in the centre) approaches an intersection, gradually decelerates but fails to stop and hits the front vehicle. The reason for this rogue behaviour is presented in the middle and bottom rows. The middle row shows the subjective DRF map of the DRF-based model. From the first to the fourth map, we see that the driver's DRF diminishes due to decreasing speed and becomes almost dormant. This also explains changes in the perceived risk map shown in the bottom row, where the driver approaches the front vehicle and decelerates to keep the perceived risk below the threshold. However, when its velocity reaches 0, no risk is perceived due to the dormant subjective DRF map. Hence, the DRF-based model would "think" that it is safe to pick up speed again according to the control mechanism we discussed in Fig. 7 and ends up colliding with the front vehicle.

By adding a safety distance d_s to Eq. (7), the subjective DRF map would never be dormant even if the velocity becomes 0. Therefore, the driver would always be able to perceive the risk in the environment. In Fig. 9, we show the modified DRF-based model brakes and stops reasonably well in the same scenario.

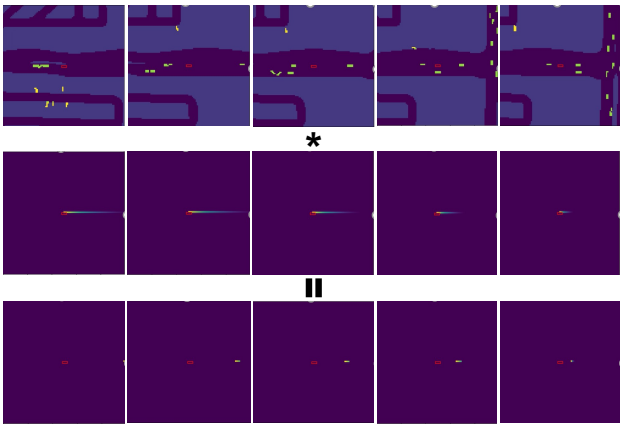


Fig. 9: The same traffic scenario as in Fig. 8 with our modified DRF-based model. Top row: The objective cost map the environment. Middle row: The subjective DRF map of the DRF-based model. Bottom row: The perceived risk map computed from the convolution product of the objective cost map and the subjective DRF map. The modified DRF-based model operates normally because the driver is able to perceive the risk in the environment at low speeds.

B. Validation of DRF-based traffic agent models

Since that the DRF-based model has not been validated with real urban driving data before, quantitative validation results are presented in this section. We start by identifying typical values for the parameters of the DRF-based model. Then, the DRF-based model parameterise by the identified typical values are validated with the Lyft Prediction Dataset. The process of the identification and validation is shown in Fig. 10. Here, a step-by-step explanation for this process is given as follows:

Step 1: 100 scenarios without intersections are selected because high-level decision-making, such as turning left/right, is not considered in the formulation of DRF. We divided the selected scenarios into 10 identification scenarios and 90 validation scenarios. As the DRF model's capability for basic manoeuvers such as lane keeping and braking is extremely important to verify before using it to develop the model-based multi-agent simulator, we evaluate the DRF model in 100 lane-keeping and 100 braking scenarios, respectively.

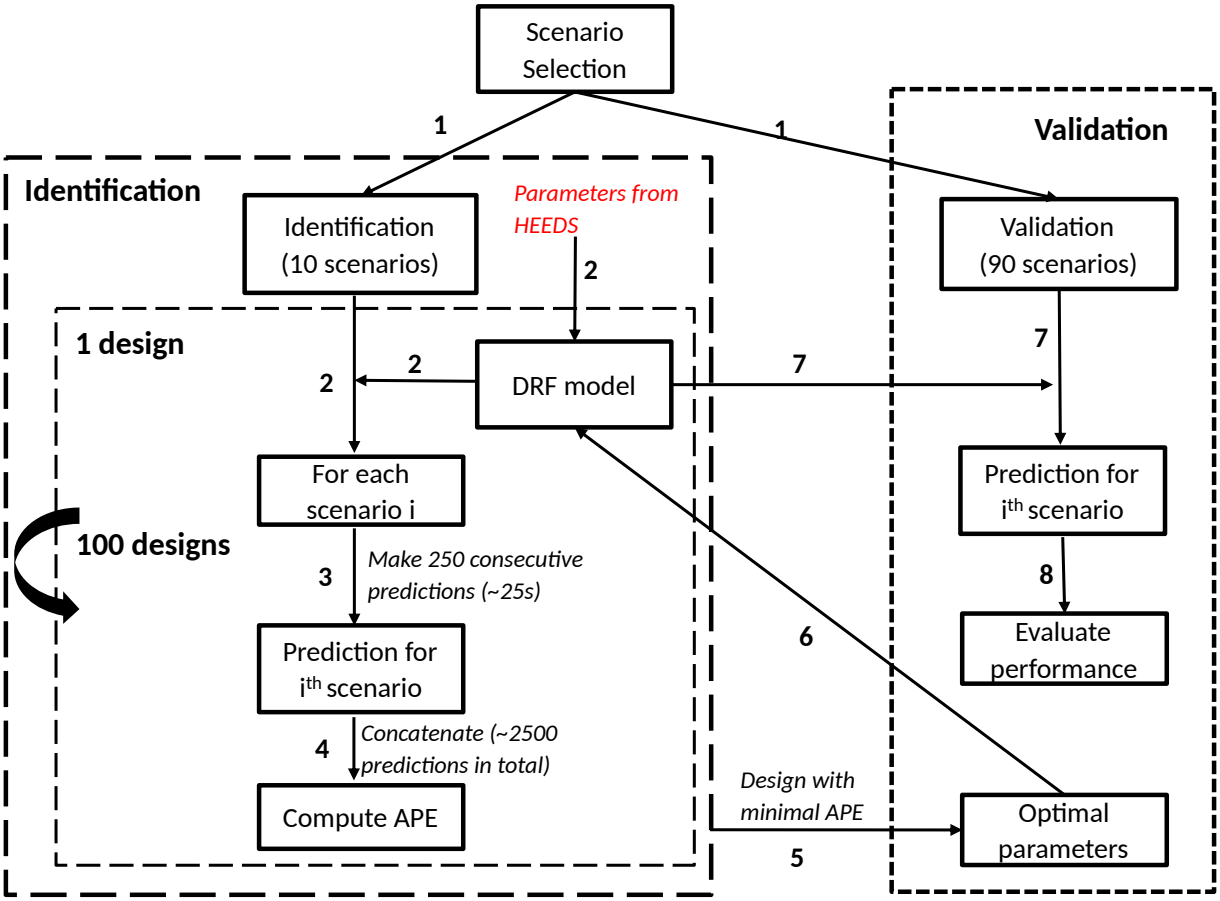


Fig. 10: The DRF parameters' identification and validation process.

Step 2: Simcenter HEEDS, a industrial parameter optimisation software, is used to identify the typical values of DRF parameters because it can efficiently search the parameter space with the SHERPA algorithm. In each design, HEEDS would parameterise the DRF model, except that we provide the baseline values based on [13] for the first design.

Step 3: For each identification scenario, we run the DRF model in closed-loop. Each scenario has a duration of 25 seconds, where the DRF model must make 250 consecutive predictions regarding its future position.

Step 4: We concatenate all predictions that the DRF model made in 10 identification scenarios. The average position error (APE) is computed by comparing the predicted positions to the ground-truth positions from the dataset.

Step 5: HEEDS would conduct 100 designs, i.e., for each design, repeat Step 2 to Step 5 by parameterising the DRF model with different parameters. Then, choose the parameters from the design with the minimal APE as the identified optimal parameters to best fit the driving behaviours demonstrated in the Lyft Prediction Dataset.

Step 6: Parameterise the DRF model with the identified optimal parameters for validation.

TABLE III: Identified parameters of the DRF-based model

Identified DRF Parameters						
p	m	t_{la}	d_s	c	k_1	k_2
0.06	0.001	4	12	0.5	0	1.12
Identified Risk-Threshold Controller Parameters						
k_v		v_{des}		R_t		
0.025		13.5		9000		

Step 7: For each of the 90 validation scenarios, we run the DRF in closed-loop and evaluate its performance.

Step 8: The performance for every validation scenario is evaluated by comparing the predicted velocity profile to the ground-truth velocity profile. Other metrics such as APE and the perceived risk is also evaluated.

The identified parameters for the lane-keeping and braking scenarios are presented in TABLE. III.

More quantitative results from the identification and validation of DRF parameters in lane-keeping scenarios are presented in Fig. 11. Fig. 11 shows the overall performance of the DRF model parameterised by the identified optimal parameters. From Fig. 11a, we observe that the velocity profiles given by the DRF mostly fit the ground-truth velocity profiles. It is worth noting that the numbered scenarios are not

necessarily connected in time and space. Therefore, the DRF model is initialised at the start of every individual scenario. We can also see the driver's perceived risk at every scenario in the bottom row of Fig. 11a and Fig. 11b. Only very few spikes of the perceived risk are observed, indicating the DRF model can quickly keep the perceived risk below the assigned threshold to achieve safe driving. Here, we provide a detailed analysis of scenario 30 in Fig. 11a, in which there are large changes in velocity and a large perceived risk appears.

As can be seen in Fig. 12b, the perceived risk is very high (approximately 25000, compared to the risk threshold of 9000) throughout almost the entire journey, indicating a very high risk of collision. The high perceived risk makes sense, as we can see from Fig. 12a that the DRF model is driving in a relatively dense urban traffic scenario with leading vehicles in front of it. This means that the driver should be more cautious while driving.

It is also interesting to see a change in velocity profile (decelerate-accelerate-decelerate) in Fig. 12b. The first deceleration stage occurs because the DRF model is too close to the leading vehicle at the beginning, so it decelerates to reduce the perceived risk. Note how the trend of the velocity profile fits the perceived risk in order to keep the perceived risk below a threshold of 9000. When the distance between the leading vehicle becomes larger, the DRF model perceives less risk and picks up speed again. At this stage, there are many oscillations in both the velocity and perceived risk profile, indicating that the DRF model is concentrating on the motion of the leading vehicle and ready to take actions to prevent collisions. This explains the accelerating stage where the DRF model picks up speed but remains highly cautious to keep a safe distance and velocity. This intention is shown in the oscillating perceived risk profile, meaning that the DRF model makes frequent and minor adjustments to the vehicle to maintain this safety-critical state of car-following. Then the leading car stops near an intersection. The ego vehicle brakes to avoid collision when the distance from the leading vehicle becomes smaller than the safety distance.

The average position error (APE) and the final position error (FPE) computed from 12c are 5.66m and 3.34m, respectively. Though the APE and FPE shows that the DRF model still needs improving to more realistically model human driving behaviours, the overall performance still matches our intuition and the perceived risk truthfully represents the actual driving risk. Considering the complexity of this urban driving scenario, the DRF model exhibiting human-like driving behaviour demonstrates its ability to mimic human driving in dense urban traffic.

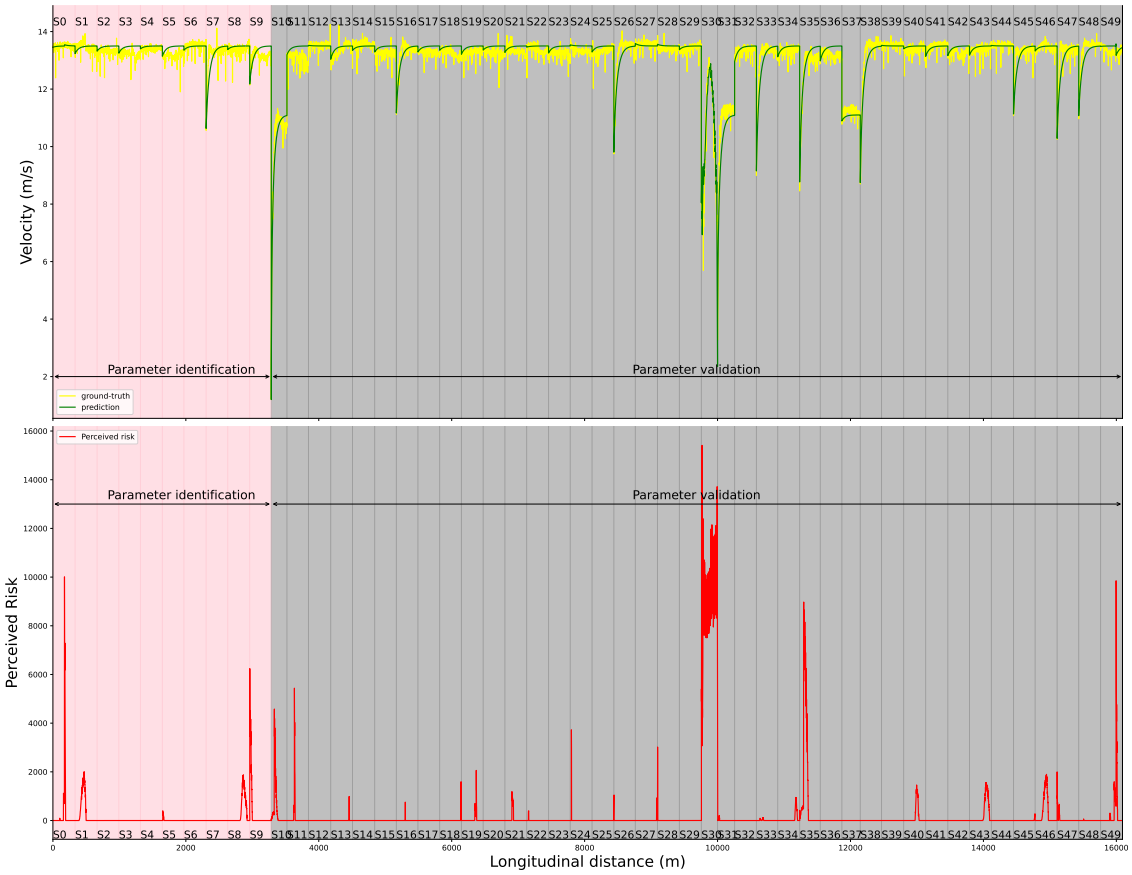
Fig. 13 shows the overall performance of the DRF model parameterised by the optimal parameters identified in the braking scenarios. From Fig. 13a, we observe that the DRF model often continues to slowly speed up when the ground-truth velocity profiles begins to decelerate and the DRF model gives many hard braking behaviours. We can also see the driver's perceived risk is really high for most scenarios, indicating that the distance between vehicles is really small, so the driver feels at risk. Moreover, the perceived risk is observed to rise quickly before abruptly drops in almost every scenario. We can deduce that the DRF model first approaches the front vehicle before quickly applying hard brakes to keep the perceived risk below the assigned threshold to achieve safe driving. Here, we provide a detailed analysis of scenario 0 in Fig. 13a, in which large changes in both velocity and perceived risk are observed.

As can be seen in Fig. 14b, the perceived risk is low most of the time until the last 30m. This fits our observation from the objective map in Fig. 14a, where the DRF model does not see any leading vehicle most of the time. However, it hard brakes near the end where it detects very high risk as the lead vehicle stops at an intersection. With an APE of 5.68m and an FPE of only 0.39m computed from Fig. 14c, the DRF model shows its ability to ensure safety in braking scenarios.

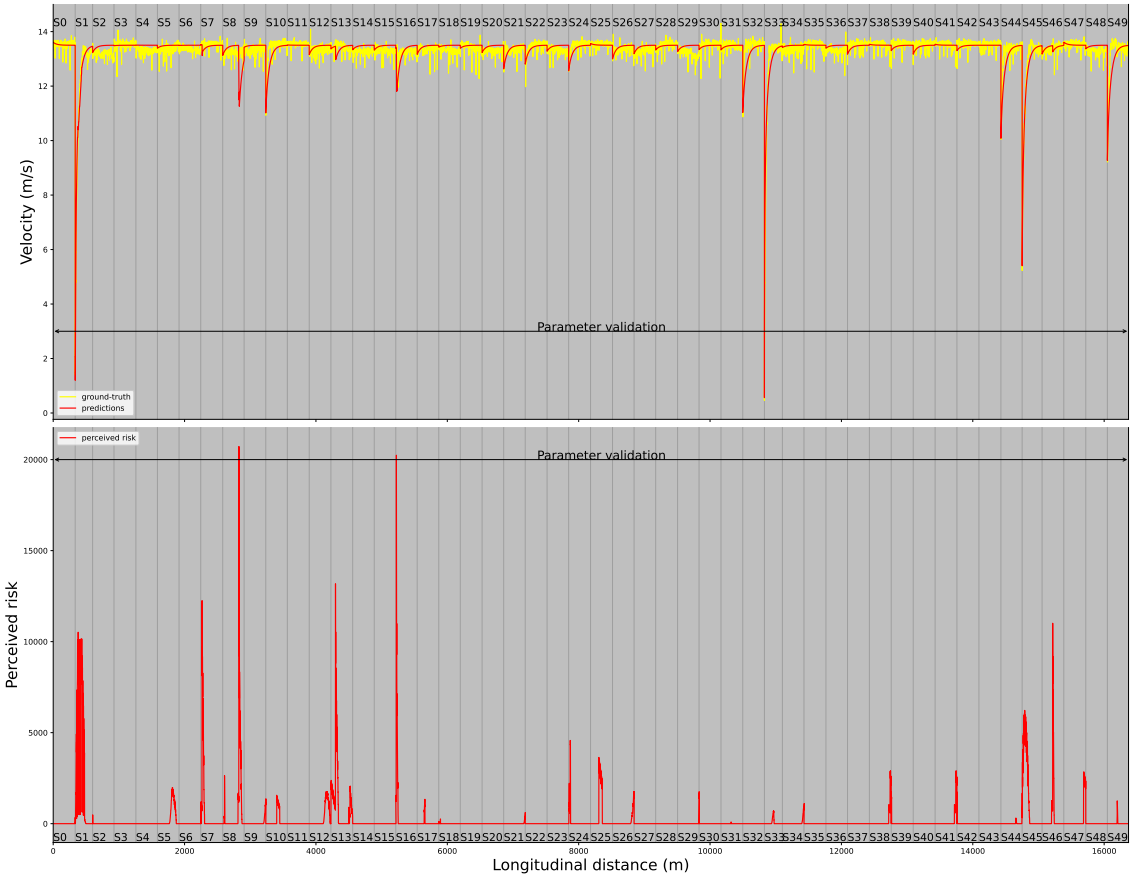
This scenario shows that the DRF model cannot capture the gradual braking manoeuvres of the human driver (transition from mild braking in the beginning to harder braking in the end). The human driver in the ground truth data in Fig. 14b can start the decelerating process even approximately 150m from the stopping location, while the DRF model can only perform hard braking in the final 25m. This deficiency of the DRF model can be traced back to its mathematical formulation where it is unable to detect obstacles 50m away from it because the subjective DRF map diminishes exponentially with longer distance away from it, as can be seen in Eq. (8). Therefore, we argue that the formulation of the DRF should be further improved to properly model urban driving behaviours.

Based on the validation results, we can conclude that the identified parameters can provide capable performance in basic urban traffic despite some deficiencies in its theoretical formulation. Here, other values for the parameters representing different driving behaviours other than the one demonstrated by the Lyft Prediction Dataset are also intuitively proposed. Instead of presenting specific values of the parameters, a connection between the values and represented driving styles is proposed so that interested readers can intuitively tune the DRF parameters to model different driving styles.

1) p (steepness of the parabola defining the height of the DRF): Larger p leads to larger values of $G(x, y)$ in the subjective DRF map modelled as a 2d Gaussian distribution, which represents that the driver deems the probability of the event happening at location (x, y) is higher. To put it more bluntly, the driver with higher p scares easily and drives cautiously.
2) d_s (safety distance): Larger d_s makes the driver to keep larger distance away from other vehicles and leads to cautious driving behaviours.
3) m (slope of widening): Larger m increase the width of the driver's perceptive field, meaning the driver takes a wider view of the road, representing more cautious driving.
4) c (width of the DRF at the ego vehicle's current position): Similar to m, larger c increases width of the perceptive field and leads to more cautious driving.
5) t_{la} (look-ahead time): larger t_{la} means the driver has better sight and can perceive risk at longer distances.

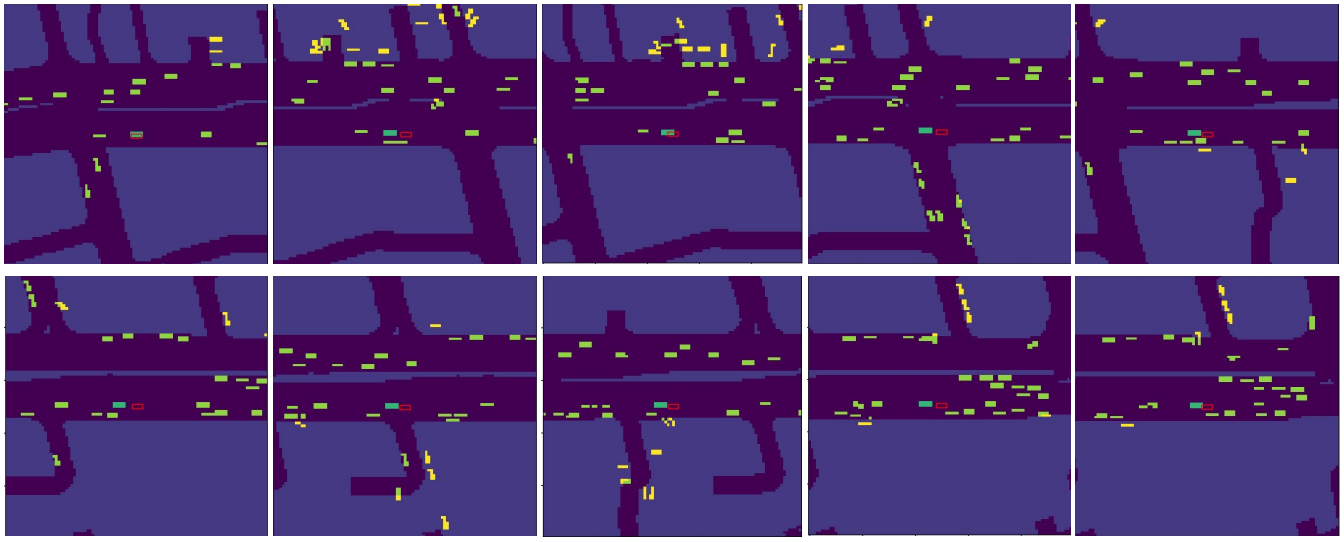


(a) Quantitative results for lane keeping: 10 identification scenarios and 40 validation scenarios (part 1).

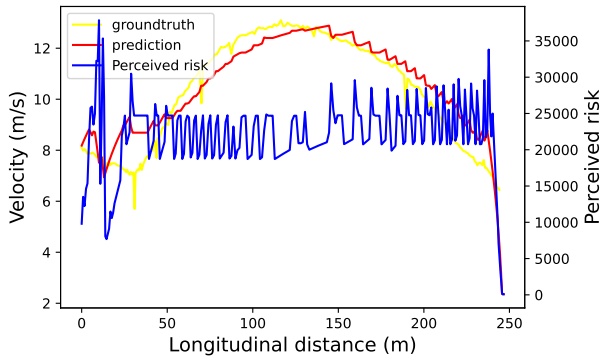


(b) Quantitative results for lane keeping: the remaining 50 validation scenarios (part 2).

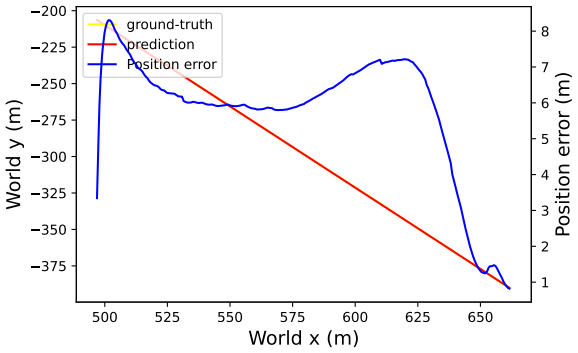
Fig. 11: Quantitative results of the identification and validation of DRF parameters in lane-keeping scenarios.



(a) Traffic scenario 16 in Fig. 11a with a duration of 25 seconds. Ten figures are sequential in the order first from left to right, then from top to bottom. The red box in the center marks the DRF model's position, with the green box marking its corresponding ground-truth position. The yellow boxes are other traffic agents from the logged data.



(b) The DRF model's velocity-longitudinal distance and perceived risk.



(c) The y-x world coordinates of the DRF model's trajectory and position error.

Fig. 12: Detailed qualitative and quantitative results of validating the DRF model with scenario 30 in Fig. 11a.

6) k_1, k_2 (Gain of DRF's inner and outer boundaries while turning): These two parameters qualify the curve cutting feature of the DRF model and are less related to driving styles.

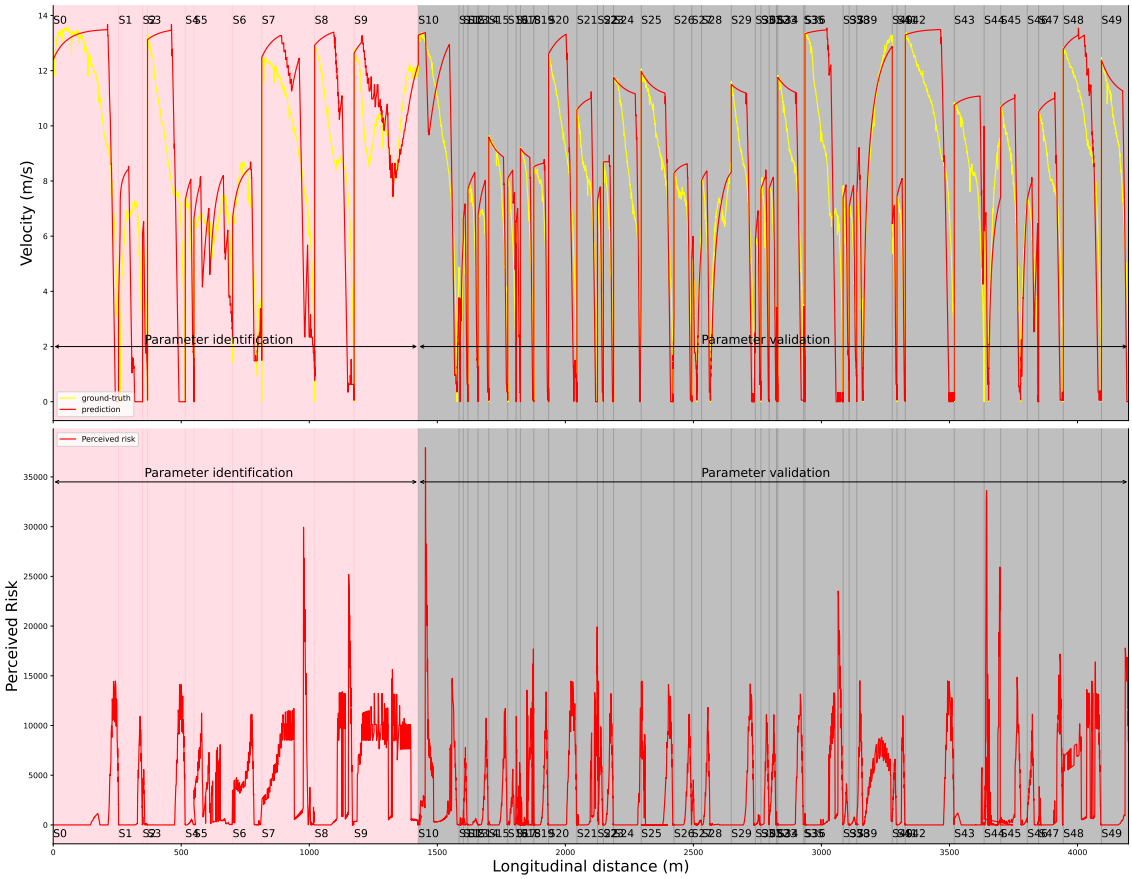
The risk-threshold controller is also related to the DRF model's driving styles:

1) R_t (risk threshold): The lower risk threshold requires more effort from the driver to keep the perceived risk below the threshold. Therefore, lower R_t means more cautious driving styles.
2) k_t (Gain of the velocity P-controller): Lower gain means slower acceleration and deceleration, corresponding to cautious driving.
3) v_{des} (Desired velocity of the driver): Intuitively, lower desired speed leads to more cautious driving.

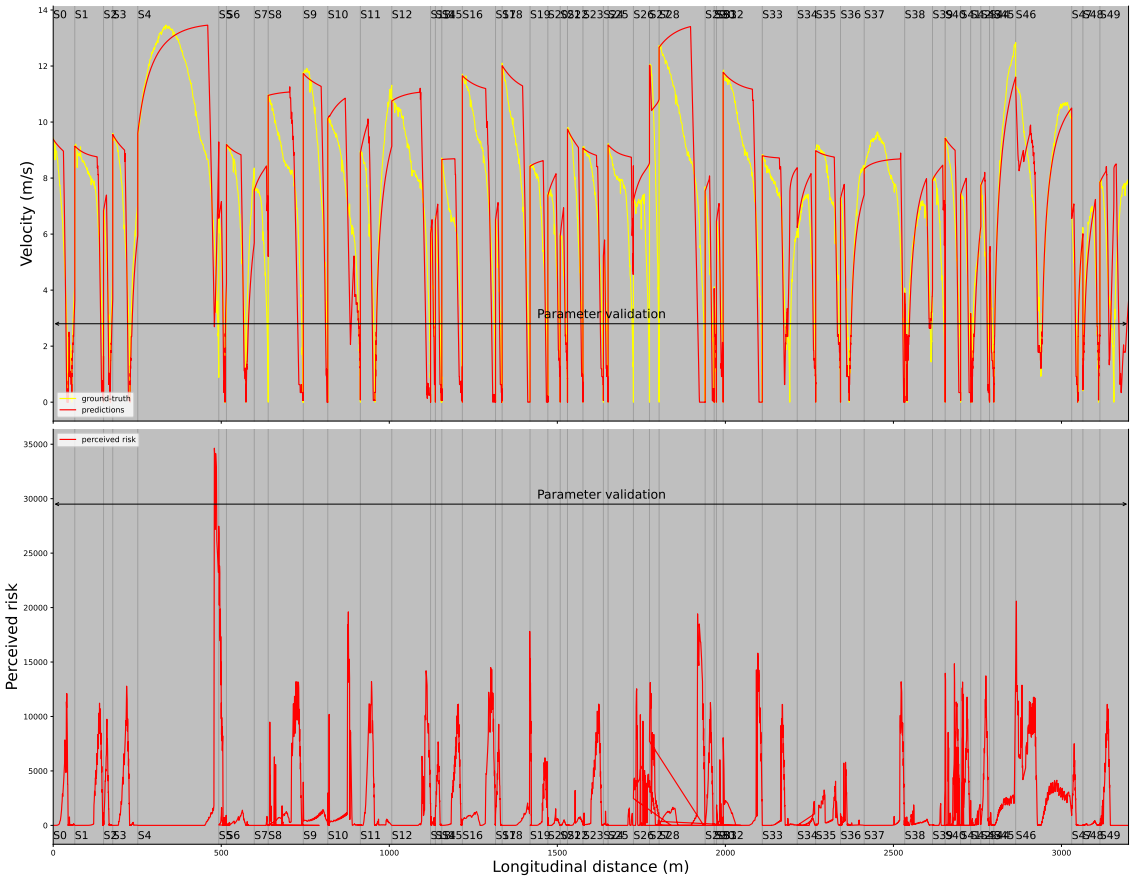
APPENDIX B
IL MODEL ARCHITECTURE AND CLOSED-LOOP TRAINING

Our IL model is adapted from Lyft Urban Driver [6], whose architecture is shown in Fig. 15. Here, we provide an intuitive explanation for the network architecture.

The bird's-eye-view image on the left qualitatively illustrates the traffic scenario that the ego vehicle is in. The input of the network can be extracted from this traffic scenario, which can be categorised into vehicle objects (ego vehicle, agent vehicles) and static objects (lanes, crosswalks). Each object is vectorised before entering the first fully connected layer, where all objects are embedded into a 128-dimensional space. Then, a sinusoidal positional embedding is added to all objects to encode sequential information of all objects. This positional embedding is essential for the multi-head attention (MHA) layer to learn the relationship between objects. Before MHA, there are three PointNet layers with each layer composed of two multilayer perceptrons (MLPs) to learn a descriptor for

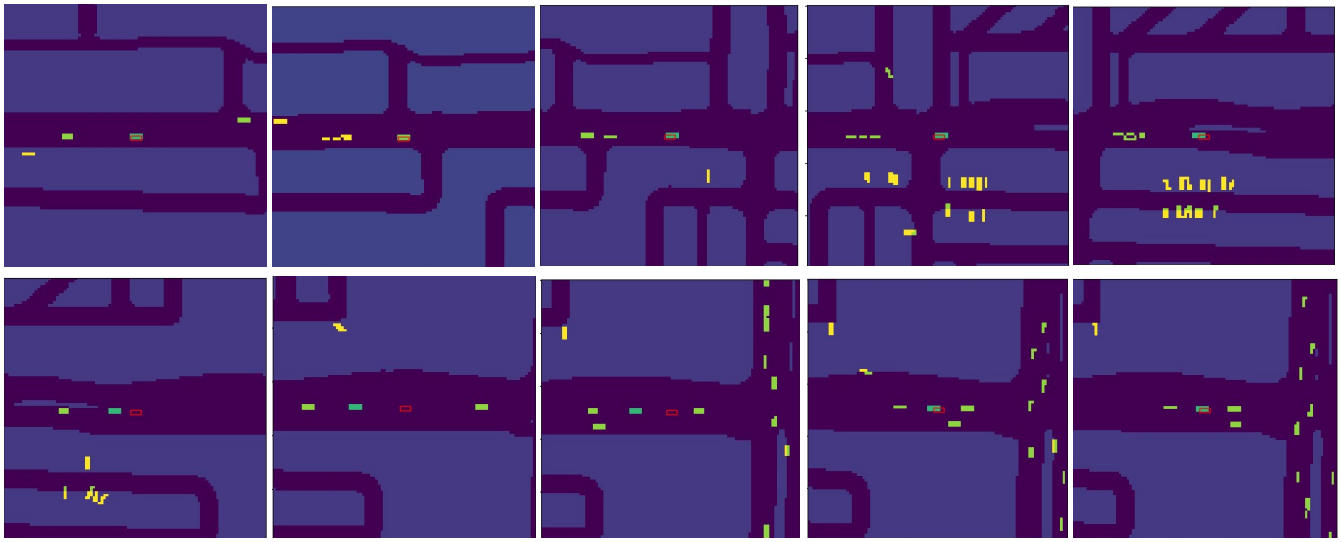


(a) Quantitative results for braking: 10 identification scenarios and 40 validation scenarios (part 1).

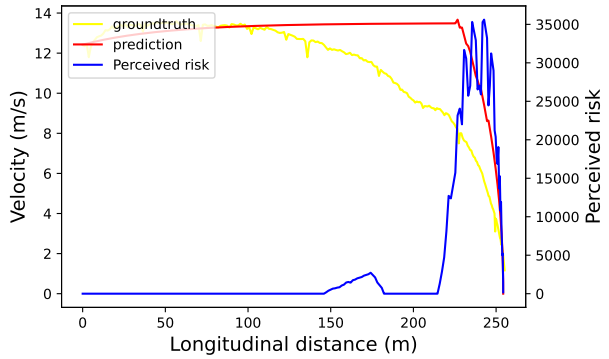


(b) Quantitative results for braking: the remaining 50 validation scenarios (part 2).

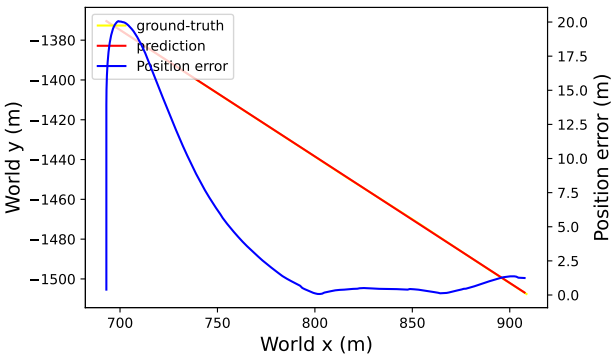
Fig. 13: Quantitative results from the identification and validation of DRF parameters in braking scenarios.



(a) Traffic scenario 1 in Fig. 13a with a duration of 25 seconds. Ten figures are sequential in the order first from left to right, then from top to bottom. The red box in the center marks the DRF model's position, with the green box marking its corresponding ground-truth position. The yellow boxes are other traffic agents from the logged data.



(b) The DRF model's velocity-longitudinal distance and perceived risk.



(c) The y-x world coordinates of the DRF model's trajectory and position error.

Fig. 14: Detailed qualitative and quantitative results of the validation of the DRF model with scenario 0 in Fig. 13a.

each object. The next part is a scaled dot-product attention layer with eight heads. The descriptor of the ego vehicle is used as query, while all descriptors are used as both keys and values. It's worth noting that a global type embedding for all objectives is also added as a key so MHA can be configured to attend based on types of objects. The final fully connected layer projects the output of MHA to the desired feature dimension. In the original work, the final output is the next pose of the ego vehicle. Whereas in our case, the final output is the spline coefficients parameterising the future trajectory. In other words, the only difference between Lyft Urban Driver and our network is the feature dimension of the final output, which leads to a slight difference in closed-loop training as shown in Fig. 16.

The choices of K, the length of policy sampling per scenario, and T, the entire length of each scenario, are most important for the stability and performance of the closed-loop training. The ablation study of [6] proposed that larger K improves performance. This conclusion is only partially true from our experience. Here, we provide a simple qualitative analysis. Since the first K steps would lead the ego vehicle to deviate from demonstrations due to the distributional shift and the last $T - K$ steps are meant for the ego vehicle to learn to recover from previous deviations, the values of T and K need to be jointly selected to achieve a good performance. Our proposed guidelines for selecting T and K are:

1) K should not be either too small or too large. If K approached 0, the closed-loop training scheme would recede to the pure closed-loop training scheme without synthetic expert query, which suffers greatly from the distributional shift. On the other hand, if K approached T, the deviation from demonstrations will be too large for the remaining $T - K$ steps to possibly recover from. Therefore, our recommendation is to start with $K \approx 0.5T$.
2) T should not be either too small or too large. If T approached 0, the closed-loop training scheme would

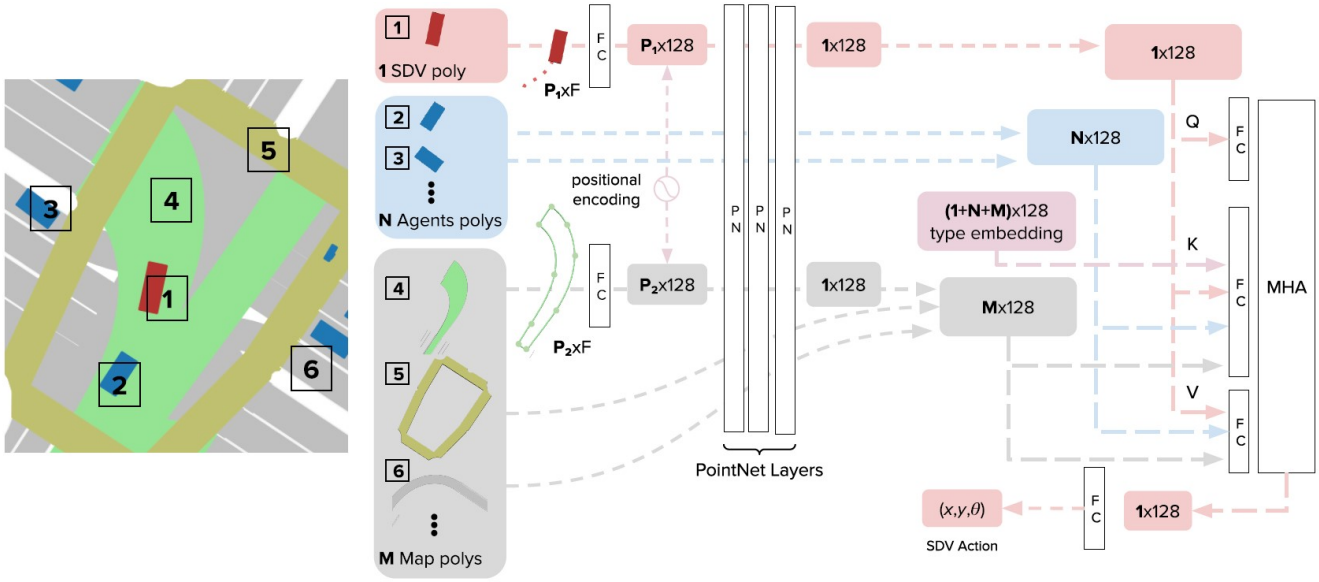
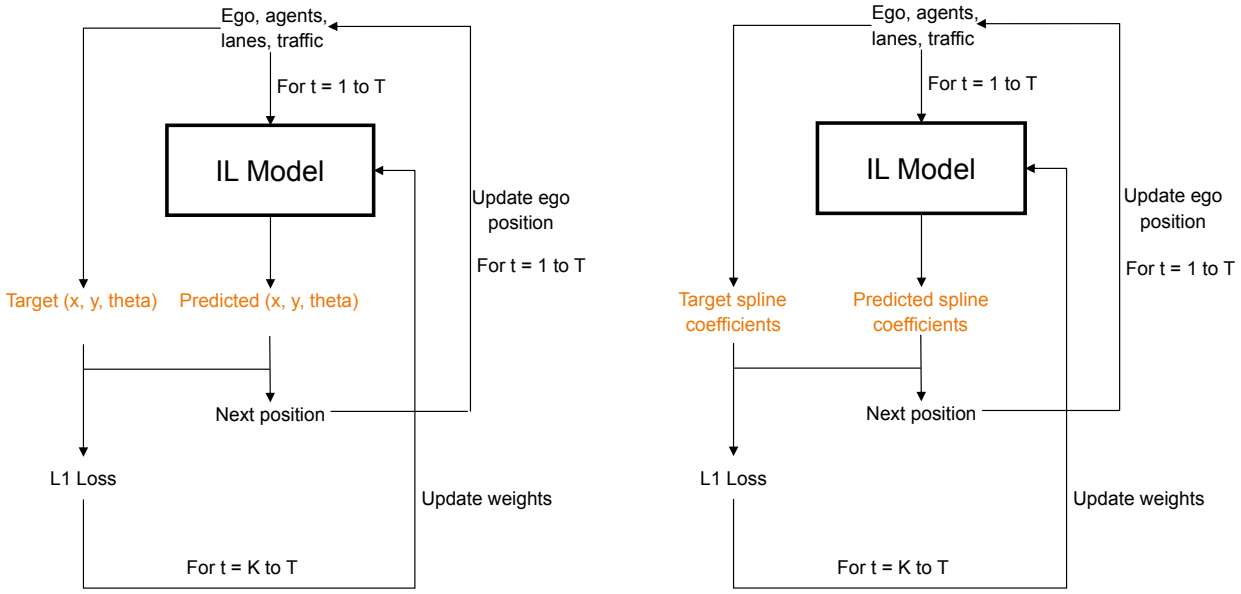


Fig. 15: Architecture of Lyft Urban Driver [6]



(a) The closed-loop training scheme of Lyft Urban Driver.

(b) The closed-loop training scheme of our IL model.

Fig. 16: Comparison of closed-loop training schemes of Lyft Urban Driver and our IL model

recede to the naive IL with open-loop training. However, large T would affect the training efficiency and increase the computation burden. Therefore, our recommendation is to start with $T \approx 20$ and T can be slightly increased if the training resource is sufficient.

In this work, we use $K = 10$ and $T = 20$.

Appendix C
Spline Parameterisation for Predicted Trajectories

A spline is defined piecewise by smooth polynomial functions. A trajectory can be approximated by a spline $s(u)$ as a linear combination of B-splines basis functions $B_i(u)$ with spline coefficients α_i:

$$s(u) = \sum_{i=1}^{n} \alpha_i B_{i,d}(u)\,. \tag{11}$$

Eq. (11) presents three essential components that parameterise B-splines, namely the degree of basis functions d, the number of control points or coefficients n, and the knot vector $U = \{u_i\}_{i=1}^{m}$ containing m knots, with each u_i corresponding to the knot point $s(u_i)$ on the B-spline. These values have a relationship of $m = n + d + 1$. In other words, one value can be determined if given the other two.

Here we detail the parameterisation used in this work. First, we choose to parameterise the future trajectory of the next two timesteps, with each timestep having a duration of 0.1 second. We decide that $n = 3$ so we have one control point for each timestep (including one control point at the beginning). The future trajectory is only 0.2 second long so that it can be sufficiently parameterised by quadratic splines ($d = 2$). Thus, the knot vector has $m = 6$ knots. To get smoother B-splines close to the control polygon defined by the control points, we use the clamped B-spline that is tangential to the control polygon at both ends. This requires that the knot vector's first and last $d+1$ elements are the same. Therefore, we determine that the knot vector $U = \{0, 0, 0, 0.2, 0.2, 0.2\}$.

APPENDIX D
SUPPLEMENTARY RESULTS FROM SEC. IV-C

In this appendix, we provide more qualitative and quantitative results to boost the conclusion from Sec. IV-C that our IL model has better training efficiency and learns safer driving compared to Lyft Urban Driver.

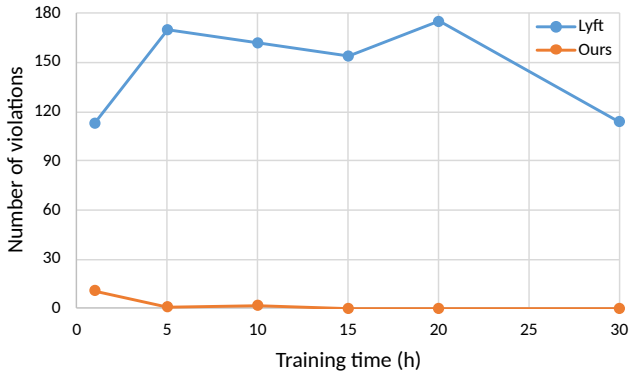


Fig. 17: Quantitative result comparing Lyft Urban Driver to our model with the same training resource

In Fig. 17, we validate Lyft Urban Driver and our model in 250 log-replay scenarios. Both models are trained 30h with 1 NVIDIA RTX A4000 laptop GPU. Our model quickly learns from expert demonstrations with decreasing number of violations in validation. However, when trained with the same weak computing resource, no improvement is observed from Lyft Urban Driver. This result further proves the supremacy of the training efficiency of our model over Lyft Urban Driver.