Proceedings of the 2024 IEEE
International Conference on Robotics and Biomimetics
December 10-14, 2024, Bangkok, Thailand

Enhance Planning with Physics-informed Safety Controller for End-to-end Autonomous Driving

Hang Zhou[1,3], Haichao Liu[1,3], Hongliang Lu[2], Jun Ma[1,3] and Yiding Ji[1,3]

Abstract— **Recent years have seen a growing research interest in applications of Deep Neural Networks (DNN) on autonomous vehicle technology. The trend started with perception and prediction a few years ago and it is gradually being applied to motion planning tasks. Despite the performance of networks improve over time, DNN planners inherit the natural drawbacks of Deep Learning. Learning-based planners have limitations in achieving perfect accuracy on the training dataset and network performance can be affected by out-of-distribution problem. In this paper, we propose FusionAssurance, a novel trajectory-based end-to-end driving fusion framework which combines physics-informed control for safety assurance. By incorporating Potential Field into Model Predictive Control, FusionAssurance is capable of navigating through scenarios that are not included in the training dataset and scenarios where neural network fail to generalize. The effectiveness of the approach is demonstrated by extensive experiments under various scenarios on the CARLA benchmark.**

I. Introduction

Autonomous Vehicle Technology (AVT) is a rapidly developing field that aims to facilitate safe and efficient navigation through complex and dynamic traffic environments. One of the key challenges in this area is to design intelligent systems that can perceive, predict, and plan for various driving scenarios based on sensory data from cameras, radar, lidar and other devices. Deep Neural Network (DNN) is a powerful class of machine learning model that can learn high-level feature representations from large-scale data, and has demonstrated remarkable performance in autonomous driving domains such as perception, prediction and planning.

The modular AVT architecture in Fig.1(a) is explainable [1] and entails mass deployment [2]. However, recent trends started to recast modular AVT architecture into a single network. End-to-end autonomous driving in Fig.1(b) can potentially learn more complex and nuanced behaviors that are difficult to capture by hand-crafted rules or algorithms in modular systems. For example, end-to-end systems learn how to handle rare or unexpected situations such as road hazards, pedestrians and traffic violations, by observing human drivers and mimicking their actions. End-to-end autonomous driving reduces the computational cost and latency of processing the sensor input. This improves the responsiveness and safety of the vehicle, especially in high-speed or dynamic scenarios. End-to-end autonomous driving also avoids error propagation and inconsistency issues that may arise from integrating multiple modules with different assumptions and representations [3].

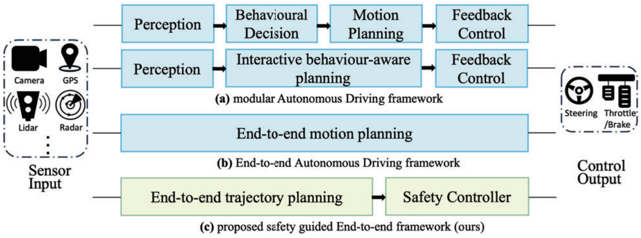

Perception
Behavioural Decision
Motion Planning
Feedback Control
Camera
GPS
Lidar
Radar
Sensor Input
Perception
Interactive behaviour-aware planning
Feedback Control
(a) modular Autonomous Driving framework
Steering
Throttle /Brake
Control Output
End-to-end motion planning
(b) End-to-end Autonomous Driving framework
End-to-end trajectory planning
Safety Controller
(c) proposed safety guided End-to-end framework (ours)

Fig. 1. Popular frameworks of Autonomous Driving [4]. Green shaded is the proposed framework pipeline of FusionAssurance. Compared to other frameworks, FusionAssurance is less modular and its DNN output can be guided by safety controller.

However, DNN-based planner does not guarantee reliability, safety and optimality due to out-of-distribution [5] and generalization issues. Since learning-based motion planner has limitations in achieving perfect accuracy, it also has natural drawback in generalizing existing known scenarios. On the other hand, out-of-distribution problem is a challenge in machine learning that occurs when the test data differs from the training data in some way. This can lead to poor performance, overconfidence or incorrect predictions by the machine learning model. Real-world driving scenarios are dynamic, complex and therefore often unpredictable. The performance of data-centric motion planner is contingent on the quality and quantity of the training data. The network may not perform well in situations that are absent from the training data or simulations.

To address these issues, we propose an innovative mapless trajectory-based end-to-end driving framework which takes advantage of Model Predictive Control and Potential Field for safety assurance. The main contributions are as follows:

- The proposed framework enables integrated decision-making and control of lane keeping, adaptive cruise control and overtaking for data-centric motion planners.
- The proposed method allows the accommodation of diverse driving behaviors by tuning parameters in the safety control module without the need to retrain the entire neural network.
- The proposed framework allows agents to navigate through unseen dynamic and complex scenarios where

[1] Robotics and Autonomous Systems Thrust, Systems Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China hzhou269@connect.hkust-gz.edu.cn; hliu369@connect.hkust-gz.edu.cn; jun.ma@ust.hk; jiyiding@hkust-gz.edu.cn

[2] Intelligent Transportation Thrust, Systems Hub, The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China hlu592@connect.hkust-gz.edu.cn

[3] Department of Electronic and Computer Engineering, School of Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China

979-8-3315-0964-4/24/$31.00 © 2024 IEEE

DNN planner failed to generate feasible trajectories. Our method surpasses preceding methodologies and has the best performance on CARLA 42 routes benchmark.

II. Related Work

End-to-end Autonomous Driving can be generally categorised as trajectory-based method and direct control-based method. Trajectory-based models are usually combined with a controller (e.g. PID) to execute the planned motion. Direct control-based network has its control actions directly optimised as network output. Although the two approaches differ in their methodologies, they can benefit from each other in the aspects of feature extraction and sensor fusion.

Direct control-based end-to-end models are usually achieved by reinforcement Learning (RL) [6]–[8] or imitation learning (IL) [9]–[13]. Latent DRL [6] creates a latent representation as an embedding space and applies RL to learn the latent observation. GRI [7] presents RL algorithm which efficiently both expert demonstration and environment exploration under predicted maps and traffic condition. CIL [10] and CILRS [9] are early IL methods that use a conditional architecture with different network branches for different navigation commands. LBC [11] and Roach [13] train agents with privileged information and later imitation learn the privileged agents. MILE [12] applies Model-based Imitation Learning technique which learns a driving policy, mapping, detection and traffic rules in bird's eye view (BEV) concurrently.

Transfuser [14], [15] has a trajectory-based end-to-end multi-modal transformer to learn surrounding scene understanding in terms of depth estimation, road segmentation, map and obstacle detection. It has a PID controller with formal rules for safety mechanism. LAV [16] applies technique of recording trajectory history of all surrounding vehicles and the ego vehicle, which significantly enlarges the diversity and quantity of the training dataset. IVMP [17] proposed an end-to-end motion planning approach which deploys network-predicted semantic map for better uncertain object handling and trajectory prediction. To take advantage of both trajectory-based and control-based networks, TCP [18] designs a single network that produces control signals and waypoints simultaneously. An adaptive ensemble network is used to merge the two outputs for refinement.

Trajectory-based approaches have the advantage of producing more interpretable results for neural network output than control-based approaches, but they also have the drawback of controller error which means that the actual motion may deviate from the desired trajectory [19], [20]. In comparison, our approach offers the benefit of interpretability and overcomes the drawback of trajectory-based methods by converting the control tracking error into intelligent decision-making and control with the use of safety controller.

Safety controller has been well studied in modern control theory and applications [21]–[23]. Rasekhipour et al. [24] proposed a MPC which models road boundary, crossable and non-crossable obstacles as Potential Function to find optimal paths and perform obstacle avoidance in predefined scenarios. Liu et al. [25] further implemented a Potential Field based MPC on the CARLA simulator. The controller takes the ground truth localization, map and perception information as input and incorporates the traffic control signal as potential functions to achieve integrated decision-making and control. Their work operates under the assumption that ground truth perception and map information are given. Our approach aims to solve the uncertainty produced by raw sensor inference through neural networks.

Safety in learning-based planning remains to be a critical problem. Wei et al. [26] proposed attention-based safety mechanism in neural network by gating backbone features for direct motion optimisation and coupling safe actions with perception. Kalaria et al. [27] presented a RL method that leverages IL and control barrier functions to ensure safety. InterFuser [28] brought up the concept of safety controller for end-to-end autonomous driving. Like other trajectory-based methods, InterFuser uses transformer to decode feature from multi-sensors and output obstacles and waypoints. InterFuser further deploys a simple safety controller to keep distance with target objects which lie in its predicted waypoints. In contrast, our approach incorporates all detected obstacles as input and determines the optimal control output through a global optimization process.

One other related work is DIPP [29]. An end-to-end planning network is trained under the guidance of a MPC controller to learn better from its expert dataset. Our approach also use MPC which serves as an additional safety layer for neural network. Users can tune the safety controller parameter for different driving behaviors without retraining the entire network. Since FusionAssurance applies to any neural network planner, DIPP can also adapt our framework to provide extra safety for its neural network.

III. Methodology

The general structure of the proposed safe autonomous driving framework consists of two parts. The first is a transformer-based network for perception and planning. The second part is physics-informed safety controller which consists of Model Predictive Controller with Potential Field Functions. Safety controller takes the output of neural transformer and generates safe low-level control actions.

A. Network Model

The proposed neural network follows DETR's [30] general structure to fuse multiple sensor data. Specifically, Transformer is used to fuse three RGB cameras' input (I_{front}, I_{left}, I_{right}) and Lidar sensor input (I_{lidar}) together. Each source is followed up with backbone to extract complex features and then fed into the Transformer Encoder for feature fusion in BEV. The learned feature is then decoded into waypoints, BEV obstacle map, probability of junction and traffic control signal.

Camera streams take pretrained ResNet [31] as backbone to convert images to feature embedding. Feature resolution changes to $\frac{H}{8} \times \frac{W}{8} \times 256$ from original image resolution $H \times W \times 3$. Lidar stream on the other hand has different

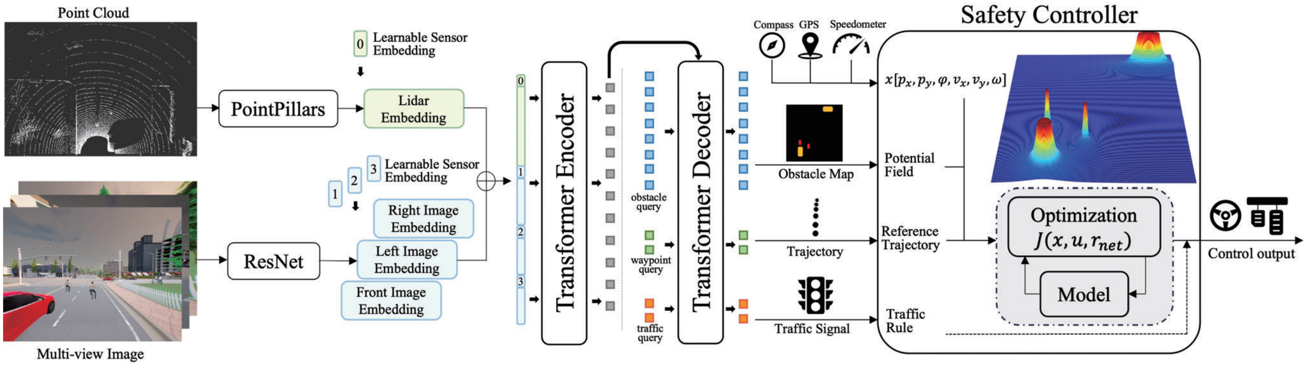


Fig. 2. Overall structure of FussionAssurance consists of two parts. 1) a transformer-based network that integrates the target location, multi-camera data, and lidar data to generate the predicted waypoint, BEV obstacle map, traffic light state, and junction probability for autonomous driving. 2) a safety controller that comprises Model Predictive Controller with Potential Functions. The safety controller takes the output of the neural network and sensor data of GPS, Compass and Speedometer to produce safe and optimal low-level control actions.

data representation and PointPillars [32] is applied for feature extraction and data resolution reduction. Lidar data consists of three-dimensional (x, y, z) points that are stored without any rule-based order, which implies that adjacent points in visualization may have their data located far apart in the point cloud. Therefore, point cloud data requires preprocessing before being fed into a convolutional neural network. Specifically, a simplified PointNet is employed to transform 3D lidar data into 2D pillar format. Point cloud data therefore can be fed into standard 2D Convolutional Neural Network. Lidar stream network takes point cloud ranged $[-l_y, 0, l_y, l_x]$ as input and assigns it into pillars sized $0.25m \times 0.25m$. The feature resolution would be $8l_y \times 4l_x \times 64$ after PointNet and $l_y \times \frac{l_x}{2} \times 512$ after 3-layer CNN.

Following the implementation of ViLT [33], single-modal spatial features are linearized and attached with modal-type embedding. Each modal embedding is then concatenated into a one-dimensional token embedding for sensor fusion feature learning. Sinusoidal positional encoding is used to capture the sequential or spatial relationships between feature tokens. Standard Visual Transformer [34] encoder with self-attention layer and feedforward layer are used. The self-attention layer enables the encoder to learn the relationships among the sensor tokens in the input sequence, by computing a weighted average of all the tokens based on their similarity. The feedforward layer applies a non-linear transformation to each token individually to enhance its representation. The output of each sub-layer is normalized and added to the sub-layer input to preserve the information from the previous layer. The Transformer has global attention for multiple modalities and captures the global context of the scene.

The decoder receives the tokens from the RGB images as values and keys, and the tokens from the LiDAR points as queries of size $H * W$, which are used to produce features in bird's eye view. Moreover, the decoder also takes two other types of queries for predicting traffic signals and waypoints w. Following the approach of InterFuser, a two-layer multilayer perceptron (MLP) is deployed as the traffic sign classifier, which determines the state of the traffic light and the presence of a stop sign ahead. A single-layer gated recurrent unit (GRU) [35] is used to generate successive waypoints w_{t_s} in an auto-regressive manner, which is conditioned on the goal location of the ego vehicle. Where t_s denotes the number of the predicted time steps.

B. Physics-informed Safety Controller

The proposed physic-informed controller is based on Model Predictive Control and Potential Field Function. The safety controller takes the output of transformer and decodes them into trajectory waypoints, obstacle map, probability of junction, red light and stop sign.

1) Vehicle Dynamics: In order to fully utilise the potential of MPC, we leverage a simple but accurate model for the vehicle dynamics which was proposed in [36].

$$\begin{aligned} \boldsymbol{x}_{k+1} &= f(\boldsymbol{x}_k, \boldsymbol{u}_k) \\ &= \begin{bmatrix} p_x(k) + \left(v_x(k)\cos\varphi(k) - v_y(k)\sin\varphi(k)\right)\Delta t \\ p_y(k) + \left(v_y(k)\cos\varphi(k) + v_x(k)\sin\varphi(k)\right)\Delta t \\ \varphi(k) + \omega(k)\Delta t \\ v_x(k) + a(k)\Delta t \\ \frac{m v_x(k) v_y(k) + l\omega(k)\Delta t - k_f\delta(k)v_x(k)\Delta t - m v_x(k)^2\omega(k)\Delta t}{m v_x(k) - (k_f + k_r)\Delta t} \\ \frac{I_z v_x(k)\omega(t) + l v_y(k)\Delta t - l_f k_f \delta(k) v_x(k)\Delta t}{I_z v_x(k) - \left(l_f^2 k_f + l_r^2 k_r\right)\Delta t} \end{bmatrix} \end{aligned} \tag{1}$$

The above model demonstrates high numerical stability and robustness in urban driving task scenarios which require frequent changes of velocity. The state vector of the dynamics model is $\boldsymbol{x} = [p_x, p_y, \varphi, v_x, v_y, \omega]^T$, $[p_x, p_y]^T$ is the vehicle position in the global map coordinate, and $[v_x, v_y]^T$ denotes the velocity in the ego vehicle coordinate. Moreover, ϕ indicates the ego vehicle orientation with respect to the map coordinate, and ω indicates the yaw rate. The control input of the system is $\boldsymbol{u} = [a, \delta]^T$, where a and δ indicate the acceleration and steering angle respectively. m is the vehicle mass, l_f and l_r represent the distance from the mass center to the front and rear axle, k_f and k_r to denote the cornering stiffness of the front and rear wheels, and I_z is the inertia polar moment. We define $l = l_f k_f - l_r k_r$ for simplicity.

2) Potential Field Function: The cost function of the MPC incorporates repulsive potential field functions for the detected objects, which enables obstacle avoidance capability. In order to avoid surrounding obstacles smoothly and computationally efficiently, elliptic functions are used to describe the PF of obstacles where obstacles' length and width are linear to ellipse vertex and co-vertex. Having a higher obstacle PF value lengthwise helps ego vehicle to perform obstacle avoidance and ACC, and relatively smaller PF value crosswise helps close fleets which move in similar orientations not to interfere with each other. The main traffic participants on the road are vehicles, cyclists and pedestrians. Cyclists and pedestrians have smaller physical dimensions but higher vulnerability, therefore bigger PF gain K_o is assigned to these two classes.

$$F_O = \sum_{i=0}^{n} \frac{K_o}{\frac{(p_{x_{rot}} - p_{i,x})^2}{a^2} + \frac{(p_{y_{rot}} - p_{i,y})^2}{b^2}} \tag{2}$$

$$\begin{bmatrix} p_{x_{rot}} \\ p_{y_{rot}} \end{bmatrix} = \begin{bmatrix} p_{i,x} \\ p_{i,y} \end{bmatrix} + \begin{bmatrix} p_x - p_{i,x} & 0 \\ 0 & p_y - p_{i,y} \end{bmatrix} \begin{bmatrix} cos(\theta) & sin(\theta) \\ -sin(\theta) & cos(\theta) \end{bmatrix} \tag{3}$$

where K_o controls the intensity of the obstacle repulsive force, which varies under different scenarios. a and b are the half-length of the major and minor axis, which defines the shape of the ellipse. Specific values of obstacle's length and width are assigned to a and b to represent the physical size of the obstacle. $p_{i,x}$ and $p_{i,y}$ denote ith obstacle's position. θ is the orientation of the target obstacle. $p_{x_{rot}}$ and $p_{y_{rot}}$ are the rotated position of the ego vehicle to align with the ellipse.

An extra potential function is applied to the target front obstacle to enhance Adaptive Cruise Control capability. The problem is simplified only to consider the euclidean distance between the ego vehicle and the target front vehicle.

$$F_C = \frac{K_c \times u_{k[0]} \times x_{k[2]}}{D_{safety} + 0.001} \tag{4}$$

where K_c denotes the gain of the cost function, $u_{k[0]}$ is the throttle control and $x_{k[2]}$ is the current speed of the ego vehicle. D_{safety} denotes the euclidean distance between the ego vehicle and obstacle which lies in its moving trajectory. The cost function penalizes acceleration and high speed when there are obstacles in the front. By having the front obstacle cost function, it helps agent to slow down without trying to perform unnecessary overtaking actions for most simple scenarios.

3) Model Predictive Control Formulation: The cost function is formalised to follow reference trajectory, reach target velocity and improve driving comfort in the receding horizon manner.

$$\begin{aligned} J(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{r}_{net}) = &\sum_{k=0}^{N} \|\boldsymbol{x}_{\text{ref},k} - \boldsymbol{x}_k\|^2 + \sum_{k=1}^{N} \|\boldsymbol{u}_k - \boldsymbol{u}_{k-1}\|^2 \\ &+ \sum_{k=0}^{N} \|\boldsymbol{u}_k\|^2 + F_O(\boldsymbol{r}_{net}, \boldsymbol{x}) + F_C(\boldsymbol{r}_{net}, \boldsymbol{x}) \end{aligned} \tag{5}$$

where $\boldsymbol{r}_{net}$ is the transformed output of the neural network, N is the length of the receding horizon, $\boldsymbol{x}_{\text{ref},k} \in \mathbb{R}^6$ is the reference trajectory point generated by spline interpolation based on the network waypoint prediction at the time step k. $F_O(\boldsymbol{r}_{net}, \boldsymbol{x})$ and $F_C(\boldsymbol{r}_{net}, \boldsymbol{x})$ are the artificial potential field functions generated according to the perception information. Therefore, the MPC controller is constructed as:

$$\begin{aligned} \min_{\boldsymbol{x},\boldsymbol{u}} \quad & J(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{r}_{net}) \\ \text{s.t.} \quad & \boldsymbol{x}_{k+1} = f(\boldsymbol{x}_k, \boldsymbol{u}_k), \forall k \in \{0, 1, ..., N\} \\ & -\boldsymbol{u}_{\min} \preceq \boldsymbol{u}_k \preceq \boldsymbol{u}_{\max}, \forall k \in \{0, 1, ..., N\} \\ & -\boldsymbol{x}_{\min} \preceq \boldsymbol{x}_k \preceq \boldsymbol{x}_{\max}, \forall k \in \{0, 1, ..., N\} \end{aligned} \tag{6}$$

The first constraint corresponds to the vehicle dynamics given by equation (1). The second and third constraints impose the bounds on the control inputs and system state variables, which are derived from the physical properties of the chosen vehicle model.

C. Neural Network and Safety Controller Integration

The output of the neural network is expressed in the ego vehicle coordinate system, whereas the physics-informed controller requires an input that is expressed in the global map coordinate system. Therefore, a coordinate transformation needs to be performed for the waypoints and the obstacles. The traffic information includes the probability of encountering a red light, a stop junction, and being on-road for the agent. Braking will be applied to the agent if the red light probability is above a certain threshold and the throttle will be disabled. Stop junction probability helps to agent to slow down during junction. On-road probability P_{on_road} tells if the agent is on the road or at the junction. P_{on_road} is used to vary the gain of obstacle potential field and weight position and yaw angle tracking terms in the cost function.

$$K_o = \frac{K}{P_{on\_road} + 0.5} \tag{7}$$

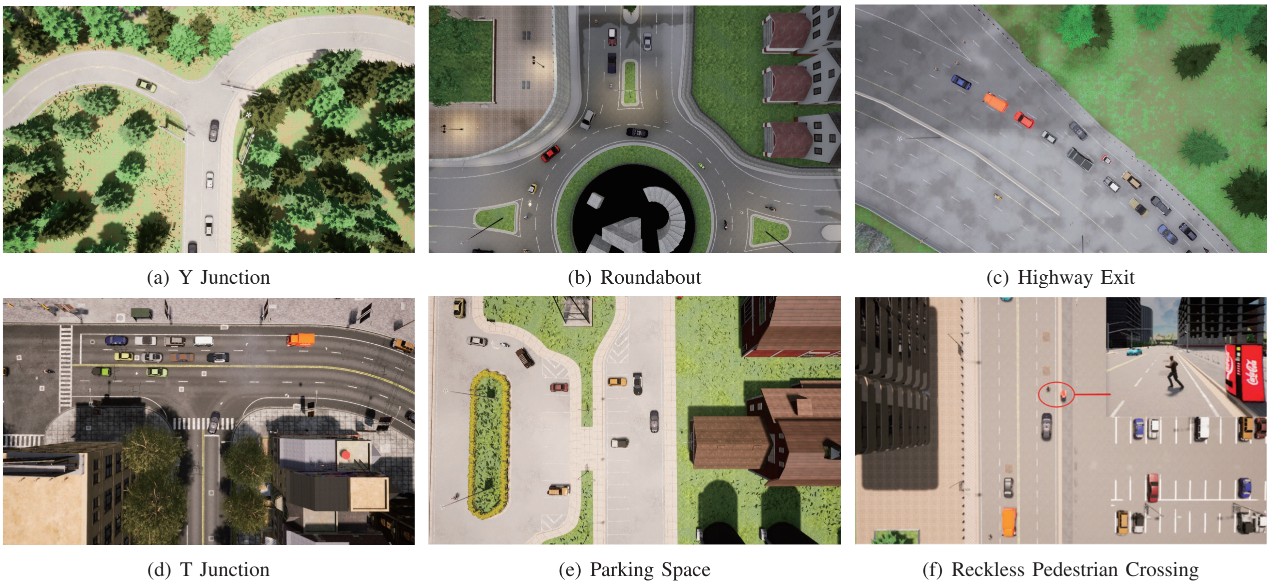
$$w_{p_x,p_y,\varphi} = w \times (P_{on\_road} + 0.5) \tag{8}$$

where $w_{p_x,p_y,\varphi}$ is the weight of position, yaw angle following. P_{on_road} has value close to 0 when neural network predicts that the agent is not on-road, which leads to equation 7 to amplify the potential function gain K_o and equation 8 to reduce the weight of position and yaw angle tracking. The above implementation results in MPC controller not precisely following the predicted trajectory and allowing overtaking actions when there are close obstacles at junction.

The overall design is therefore able to 1) precisely track the trajectory for most of the time and 2) perform its own decision-making of overtaking or obstacle avoidance when neural network-generated trajectory is not optimal.

IV. Experiments

This section presents a comprehensive overview of the experimental setup, encompassing data collection, benchmarking and metrics for evaluation. The proposed method undergoes rigorous testing across a spectrum of intricate and dynamic scenarios which include those depicted in Fig. 3. Furthermore, Section IV-B aims to demonstrate the efficacy



(a) Y Junction

(b) Roundabout

(c) Highway Exit

(d) T Junction

(e) Parking Space

(f) Reckless Pedestrian Crossing

Fig. 3. Test Examples of FusionAssurance completing various scenarios under different weather conditions.

of the proposed methods in selected corner cases where former end-to-end driving algorithms encounter problems.

A. Environment, Data Collection and Benchmark

Autonomous driving framework and experiments are conducted on the open-source simulator CARLA version 0.9.10.1. A dataset of 1M frames was collected from an agent that followed rule-based policy with access to the privileged information of CARLA as groundtruth. Data was collected on all available 8 towns and 21 weather conditions at 2 Hz rate. Routes, dynamic objects and adversarial scenarios were randomly generated as provided in [28] to increase the diversity of the collected data. Each frame of dataset contain information of Front, Left and Right Camera Images, Lidar Point Cloud, GPS coordinates and corresponding groundtruth object detection labels generated by simulator.

Evaluations and experiments were conducted on CARLA 42 routes benchmark as its convenience of visualization and variety of scenarios. MPC controller uses optimiser heavily and its performance highly depends on the CPU power, thus visualization is important for hyperparameter tuning. The benchmark requires the intelligent agent to follow predefined routes without colliding or breaking traffic rules in the presence of adversarial events. The benchmark randomly generates target points and a list of goal locations in global map coordinates for each run. The proposed method utilizes these goal locations to guide the agent to drive without manually setting high-level navigational commands. Three metrics are introduced by the CARLA Leaderboard [37] to evaluate our method, which are the route completion ratio (RC), infraction score (IS) and driving score (DS). The driving score is the product of the infraction score and route completion ratio. It is overall a more comprehensive metric to measure agent planning and safety control capability.

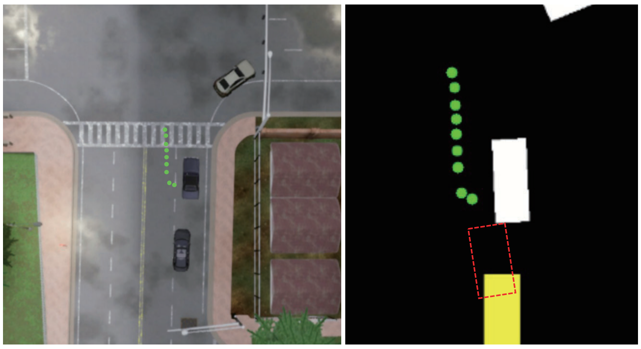
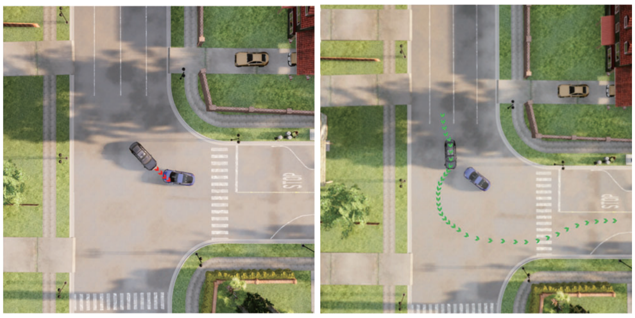


Fig. 4. Case when unfeasible trajectory is generated by neural network.

B. Effectiveness Analysis on Corner Cases

The CARLA leaderboard algorithms have demonstrated their performance on the vast majority of scenarios. However, there are still some corner cases that previous work failed to pass. Our proposed method completes complex scenarios (Figure 3) and corner cases that previous algorithms could not handle. This section presents several corner cases that the previous state-of-the-art algorithms failed to pass, and illustrates the benefit of our framework. Some of these corner cases occurred to the rule-based agent during data collection, and were consequently learned by the neural agents that adopted a similar policy.

1) Case 1(Fig. 4): This case shows an anomalous predicted trajectory and it occurs when the neural network is not able to generalize for specific kind of scenarios. It can be seen from the figure that the first available waypoint is 4 meters in front of the ego vehicle and the target vehicle is also $3.5m$ in the front. Collision might occur for InterFuser's safety controller as the target vehicle is not on ego vehicle's trajectory and the agent would not slow down. The InterFuser's safety controller simply is not aware of



(a) no physics-informed controller (b) with physics-informed controller

Fig. 5. The left figure shows the agent getting into a Deadlock situation by the neural network planner. The right figure shows the guided trajectory of FusionAssurance under the same planning network.

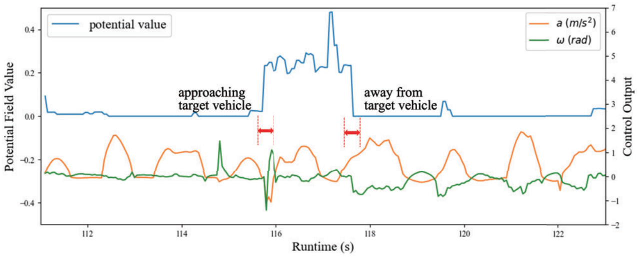
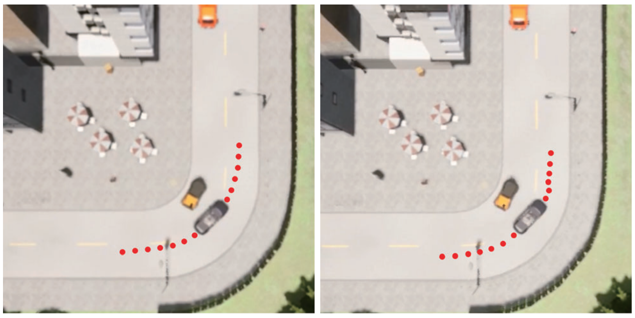


Fig. 6. Obstacles Potential Field value and control output for Case 2.

the existence of the target vehcile under such corner case. Interfuser would directly drives towards the nearest waypoint and red-dash bounding box is where the collision might happen. Our proposed physics-informed safety controller employs Potential Functions and Model Predictive Control to generate a smooth and collision-free trajectory for the agent, which allows the agent to navigate safely through the scenario.

2) Case 2(Fig. 5): This case is at a T-junction where an ego vehicle is driving into an intersection and oncoming cars have the right of way. Despite the two vehicles sharing different paths ideally, the rule-based target agent and trained ego agent drive towards each other as the target vehicle takes a wide turn and the ego vehicle takes a short turn. Two vehicles result in a deadlock situation where two agents stop and give way to each other but none of them are intelligent enough to overtake another.

This is a typical case caused by a low-quality training dataset where the recorded data perform a bad example for the neural network. Our proposed physics-informed safety controller overcomes this issue with the designed obstacle potential field function and performs its own overtaking decision. The physics-informed controller detects the potential field of the vehicle from a far distance and incorporates it in its receding horizon optimization. In Fig. 6, when the target vehicle approaches the ego vehicle at time 115.5s, the obstacle potential value starts to rise. the ego vehicle rapidly decelerates and steers away from the target vehicle. The ego vehicle continuously plans a new trajectory and returns to the planned trajectory as the target vehicle diminishes at time 117.5s. The ego vehicle thus manages to execute overtaking decision and control by taking a wider turn than the original predicted trajectory.



(a) MPC(w_{p_x,p_y}=15) (b) MPC(w_{p_x,p_y}=15)+PF(K_o=60)

Fig. 7. The trajectory of the agent with different parameters. (a) and (b) shows the waypoints of agent using MPC and agent using MPC with PF.

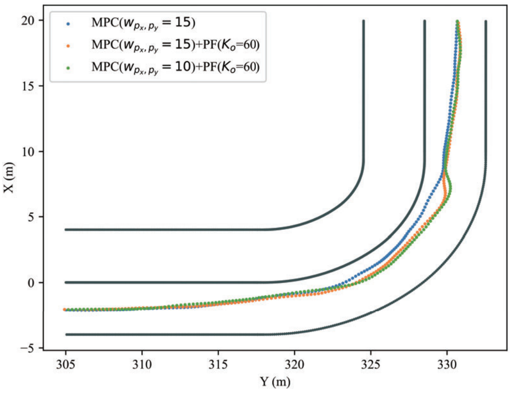


Fig. 8. Trajectory comparison of three sets of parameters under Case 3.

3) Case 3(Fig. 7): This case aims to analysis the effect of different safety controller parameters on driving behavior. The case happens when agent is taking a corner at single carriageway. The MPC controller exhibits a corner-cutting behavior where a portion of the ego vehicle crosses into the opposite lane while following the trajectory. This behavior poses a safety risk if there is an oncoming target vehicle that makes a large turn.

With the assistance of Potential Field Function, the agent keeps the target vehicle's position in motion consideration. As shown in Figure. 8, agent with Potential Field Function has $0.6m$ larger safety distance when it is closest to the target vehicle. By lowering the position tracking parameter w_{p_x,p_y} to 10, agent has further $0.1m$ safety distance. Users can also adjust the agent's driving behavior by tuning the values of target speed, weights for the position tracking w_{p_x,p_y} and gain of obstacle potential function K_o. In general, smaller value of w_{p_x,p_y} and higher value of K_o tends to lead to more overtaking actions.

C. Comparison with State of the Art

The evaluations are done on the CARLA 42 routes benchmark. CARLA 42 routes benchmark encompasses a diverse set of 42 routes across 6 distinct towns. This comprehensive testing suite combines 7 weather patterns with 6 daylight

TABLE I
COMPARISON WITH STATE-OF-THE-ART ON CARLA 42 ROUTES BENCHMARK.

Method	Route Comple↑	Infraction Rate↑	Driving Score↑
InterFuser (reproduced) [28]	93.827	0.865	81.084
Latent Transfuser (pretrained) [14]	87.359	0.756	63.570
Transfuser++ (pretrained) [38]	91.989	**0.982**	90.203
TCP (pretrained) [18]	90.946	0.912	82.991
FusionAssurance (ours)	**99.966**	0.907	**90.683**

Note: Infraction rate does not consider travelled distance. Driving score is more comprehensive for measuring infraction per unit distance. [IV-E]

TABLE II
ABLATION STUDY ON THE NETWORK STRUCTURE AND SAFETY CONTROLLER.

Method	Route Comple↑	Infraction Rate↑	Driving Score↑
FusionAssur(histogram) PID	93.827	0.865	81.084
FusionAssur(histogram) MPC+PF	97.177	0.857	83.184
FusionAssur(pointpillar) PID	96.970	0.840	81.435
FusionAssur(pointpillar) MPC	97.839	0.844	82.576
FusionAssur(pointpillar) MPC+PF	99.966	0.907	90.683

scenarios to ensure each route presents a unique driving experience.. We take the state-of-the-art algorithms on CARLA online leaderboard - **InterFuser** [28], **TransFuser** [14], [15], **TransFuser++** [38] and **TCP** [18] for comparison. The CARLA benchmark imposes substantial penalties for infractions and therefore most prior work adopt conservative driving behaviors that prioritize avoiding infractions over completing routes. The conservative driving behaviors often lead to Deadlock situations where stuck agents are not intelligent enough to overtake another.

As shown in Table I, **InterFuser(reproduced)** and FusionAssurance use the exact same training data for performance comparison. FusionAssurance outperforms InterFuser by 9.6% in driving score with higher route complement ratio 99.966% and fewer infractions with 0.907. **TransFuser++** has the best infraction score but their driving behavior is conservative. Similar to other previous work, TransFuser++ stops for all potential obstacles without much overtaking. It often gets into Deadlock situation and its route complement ratio is 9% less than our proposed method. Our proposed method has the highest driving score and route complement ratio. This is mainly due to adaptive cruise control, obstacle avoidance and overtaking capability provided by safety controller.

D. Ablation Study

The effectiveness of each module is studied in this section. In Table II, FusionAssurance(histogram) PID uses histogram for lidar feature extraction and rule-based PID for vehicle motion control. PointPillars preserves more 3D point cloud features compared to the 2d lidar histogram. This helps the network to understand the surroundings, especially the detection of small-sized objects and the differentiation of objects with different heights. Under the same controller setting, pointpillars feature extraction methods have up to 7.5% driving score increase.

With the replacement of PID controller by the MPC controller, the method is able to achieve a higher overall driving score and higher route complement rate both by 1%. The MPC controller optimizes not only the nearest waypoints, but also a horizon of future waypoints under the specified vehicle dynamics. This indicates MPC controller exhibits better performance in trajectory tracking and thus achieves a higher route completion ratio.

By replacing the controller from PID to our proposed physics-informed safety controller(MPC+PF), FusionAssurance(histogram) and FusionAssurance(pointpillars) gain an increase in driving score by 2.1% and 9.2% repectively. The full version FusionAssurance which deploys pointpillars, MPC and PF has a significant performance boost on all three metrics.

E. Discussions

The CARLA infraction rate does not take travelled distance into account. Agents which travel short distance and get into Deadlock situation can still get high infraction rate. Driving score which is the product of route complement and infraction rate is therefore a more comprehensive metric for measuring infraction rate per unit distance. Therefore FusionAssurance achieves the best performance on infraction per unit distance.

V. CONCLUSIONS

In this paper, we propose FusionAssurance, an integrated end-to-end autonomous driving framework with safety controller. This framework employs a neural network for perception and trajectory planning, and Model Predictive Control with Potential Field for enhanced safety control. The safety of the neural network planning task is ensured by the low-level control with the assistance of designed Potential Functions. The overall framework also makes integrated overtaking decisions to compensate for the suboptimal network planning results and to circumvent potential deadlock. With same training dataset, FusionAssurance outperforms baseline - InterFuser by 9.6% on CARLA 42 routes benchmark. Additionally, the physics-informed safety controller can adapt any trajectory-based end-to-end neural network to boost the overall performance.

Potential future work includes adding lane detection to the neural network and using lane information as Potential Field to reduce the out-of-lane rate. Another future improvement is to develop static obstacle detection ability for the network to further prevent infraction.

ACKNOWLEDGMENT

This work is partially supported by the National Natural Science Foundation of China under grants 62303389 and 62373289, Guangdong Basic and Applied Basic Research Funding under grants 2022A151511076 and 2024A1515012586, Guangzhou Basic and Applied Basic

Research Scheme under grants 2023A04J1067, together with Guangzhou-HKUST(GZ) Joint Funding Program under grants 2023A03J0678, 2023A03J0011, 2024A03J0618 and 2024A03J0680.

REFERENCES

[1] C. Chen, A. Seff, A. Kornhauser, and J. Xiao, "Deepdriving: Learning affordance for direct perception in autonomous driving," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 2722–2730, 2015.

[2] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020.

[3] A. Tampuu, T. Matiisen, M. Semikin, D. Fishman, and N. Muhammad, "A survey of end-to-end driving: Architectures and training methods," *IEEE Transactions on Neural Networks and Learning Systems (NeurIPS)*, vol. 33, pp. 1364–1384, apr 2022.

[4] W. Schwarting, J. Alonso-Mora, and D. Rus, "Planning and decision-making for autonomous vehicles," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 1, pp. 187–210, 2018.

[5] D. Hendrycks and K. Gimpel, "A baseline for detecting misclassified and out-of-distribution examples in neural networks," *Proceedings of International Conference on Learning Representations*, pp. 24–26, 2017.

[6] X. Ma, J. Li, M. J. Kochenderfer, D. Isele, and K. Fujimura, "Reinforcement learning for autonomous driving with latent state inference and spatial-temporal relationships," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6064–6071, 2021.

[7] D. Chen, V. Koltun, and P. Krähenbühl, "Gri: General reinforced imitation and its application to camera-based autonomous driving," *Robotics*, vol. 12, no. 5, p. 127, 2023.

[8] D. Chen, V. Koltun, and P. Krähenbühl, "Learning to drive from a world on rails," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1039–1048, 2021.

[9] F. Codevilla, E. Santana, A. Lopez, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9328–9337, 2019.

[10] F. Codevilla, M. Müller, A. López, V. Koltun, and A. Dosovitskiy, "End-to-end driving via conditional imitation learning," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4693–4700, 2018.

[11] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Proceedings of the Conference on Robot Learning*, vol. 100 of *Proceedings of Machine Learning Research*, pp. 66–75, PMLR, 30 Oct–01 Nov 2020.

[12] A. Hu, G. Corrado, N. Griffiths, Z. Murez, C. Gurau, H. Yeo, A. Kendall, R. Cipolla, and J. Shotton, "Model-based imitation learning for urban driving," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 20703–20716, 2022.

[13] Z. Zhang, A. Liniger, D. Dai, F. Yu, and L. Van Gool, "End-to-end urban driving by imitating a reinforcement learning coach," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 15222–15232, 2021.

[14] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *Pattern Analysis and Machine Intelligence (PAMI)*, pp. 12878–12895, 2022.

[15] A. Prakash, K. Chitta, and A. Geiger, "Multi-modal fusion transformer for end-to-end autonomous driving," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 7077–7087, 2021.

[16] D. Chen and P. Krähenbühl, "Learning from all vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17222–17231, 2022.

[17] H. Wang, P. Cai, Y. Sun, L. Wang, and M. Liu, "Learning interpretable end-to-end vision-based motion planning for autonomous driving with optical flow distillation," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 13731–13737, 2021.

[18] P. Wu, X. Jia, L. Chen, J. Yan, H. Li, and Y. Qiao, "Trajectory-guided control prediction for end-to-end autonomous driving: A simple yet strong baseline," in *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6119–6132, 2022.

[19] M. Park, S. Lee, and W. Han, "Development of steering control system for autonomous vehicle using geometry-based path tracking algorithm," *ETRI Journal*, vol. 37, no. 3, pp. 617–625, 2015.

[20] H. Liu, K. Chen, and J. Ma, "Incremental learning-based real-time trajectory prediction for autonomous driving via sparse gaussian process regression," in *2024 IEEE Intelligent Vehicles Symposium (IV)*, pp. 737–743, IEEE, 2024.

[21] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, "Control barrier functions: Theory and applications," in *2019 18th European Control Conference (ECC)*, pp. 3420–3431, 2019.

[22] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," in *Proceedings. 1985 IEEE International Conference on Robotics and Automation (CoRL)*, vol. 2, pp. 500–505, 1985.

[23] Y. Li, X. Tang, K. Chen, C. Zheng, H. Liu, and J. Ma, "Geometry-aware safety-critical local reactive controller for robot navigation in unknown and cluttered environments," *IEEE Robotics and Automation Letters*, vol. 9, no. 4, pp. 3419 – 3426, 2024.

[24] Y. Rasekhipour, A. Khajepour, S.-K. Chen, and B. Litkouhi, "A potential field-based model predictive path-planning controller for autonomous road vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 5, pp. 1255–1267, 2017.

[25] H. Liu, K. Chen, Y. Li, Z. Huang, J. Duan, and J. Ma, "Integrated behavior planning and motion control for autonomous vehicles with traffic rules compliance," in *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 1–7, 2023.

[26] B. Wei, M. Ren, W. Zeng, M. Liang, B. Yang, and R. Urtasun, "Perceive, attend, and drive: Learning spatial attention for safe self-driving," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4875–4881, 2021.

[27] D. Kalaria, Q. Lin, and J. M. Dolan, "Towards safety assured end-to-end vision-based control for autonomous racing," in *22nd IFAC World Congress*, pp. 2767–2773, 2023.

[28] H. Shao, L. Wang, R. Chen, H. Li, and Y. Liu, "Safety-enhanced autonomous driving using interpretable sensor fusion transformer," in *Conference on Robot Learning*, pp. 726–737, 2023.

[29] Z. Huang, H. Liu, J. Wu, and C. Lv, "Differentiable integrated motion prediction and planning with learnable cost function for autonomous driving," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2023.

[30] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision (ECCV)*, pp. 213–229, Springer, 2020.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[32] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12697–12705, 2019.

[33] W. Kim, B. Son, and I. Kim, "Vilt: Vision-and-language transformer without convolution or region supervision," in *Proceedings of the 38th International Conference on Machine Learning*, pp. 5583–5594, 2021.

[34] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *The International Conference on Learning Representations (ICLR)*, 2021.

[35] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 1724–1734, 2014.

[36] Q. Ge, Q. Sun, S. E. Li, S. Zheng, W. Wu, and X. Chen, "Numerically stable dynamic bicycle model for discrete-time control," in *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, pp. 128–134, 2021.

[37] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*, pp. 1–16, 2017.

[38] B. Jaeger, K. Chitta, and A. Geiger, "Hidden biases of end-to-end driving models," *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pp. 8240–8249, 2023.