2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

InterCoop: Spatio-Temporal Interaction Aware Cooperative Perception for Networked Vehicles

Wentao Wang[1], Haoran Xu[1,2], and Guang Tan[1,*]

***Abstract*— In autonomous driving, cooperative perception through vehicle-to-vehicle (V2V) communication is considered crucial for enhancing traffic safety and efficiency. However, existing methods often simplify the handling of perception data from multiple vehicles. In these approaches, the ego-vehicle aggregates observations from all neighboring connected cooperative vehicles (CCV), without considering the interactions between the vehicles or making differentiated use of the acquired sensing data. This approach can result in suboptimal performance due to the increase of noise and large transmission delay. In this paper, we introduce a novel approach to cooperative perception. By fusing both the road topology and trajectory histories of neighboring CCVs, our model learns an interaction score for each CCV. These scores prioritize vehicles that are most relevant to the current driving scenario, offering valuable guidance for *selective fusion* of sensor data, thereby enhancing driving decision-making. The proposed method is validated through experiments conducted on the CARLA simulator. Results demonstrate that our approach surpasses existing methods in terms of performance and robustness.**

I. INTRODUCTION

With the assistance of Vehicle-to-vehicle (V2V) communication technologies, connected cooperative vehicles (CCVs) can access shared observations and state information from nearby vehicles, thereby enhancing their perception capabilities and driving safety [1], [2], [3]. From the perspective of an ego-vehicle, a naive way to maximize its field of view is by gathering and aggregating sensor data from all vehicles within its communication range. However, in practice, communication bandwidth is often limited, which can result in longer transmission delays when dealing with high-volume sensor data. Moreover, redundant data can introduce additional noise that may affect the decision-making process while driving [4]. Therefore, how to select the vehicles most relevant to driving safety for cooperation poses a major challenge.

Fig. 1 shows a typical traffic scenario with potential risk. In this scenario, the ego-vehicle plans to turn left under the control of a driving algorithm. The white truck obstructs the ego-vehicles line of sight, making it unable to see the blue vehicle approaching from behind the truck. While all CCVs numbered 1 to 4 can communicate with the ego-vehicle, only CCV1 and CCV2 have the ability to jointly detect the impending collision. If the ego-vehicle, constrained by limited bandwidth, chooses to communicate with only the two closest neighboring vehicles, CCV3 and CCV4, it would expose itself to a significant risk of collision.

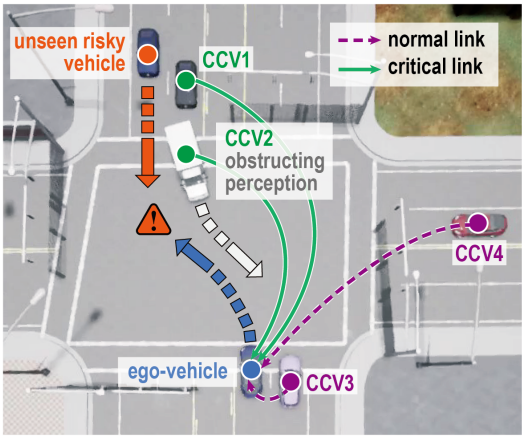


Fig. 1: A traffic scenario in which cooperative perception helps reduce collision risk. In the top-left corner, a vehicle is crossing the intersection in a downward direction, unobserved by the ego-vehicle and lacking a communication link with other vehicles. CCV1 and CCV2 collectively identify the potential risk with their sensors, and notify the ego-vehicle, enabling it to apply early braking. However, the cooperative perception from CCV3 and CCV4 are irrelevant for the particular traffic hazard.

Existing V2V cooperation methods [5], [6], [7], [8] mostly focus on expanding perception for tasks like 3D detection or motion forecasting, without considering how to actively select CCVs for cooperation based on an understanding of inter-vehicle interaction. Non-selective fusion may impair decision-making performance due to excessive redundant information from CCVs and a lack of discriminability in the fusion process. If we can take into account agent interaction, cooperative perception can become more efficient by prioritizing vehicles that are most relevant to the current driving scenario. Our approach to capturing the interaction is by extracting spatio-temporal information between the ego-vehicle and surrounding environment, which includes the nearby road network and neighboring vehicles.

We present a Spatio-Temporal Interaction Aware Cooperative (InterCoop) perception method for the multi-vehicle scenario considered. InterCoop can extract the geometry information of the road network, as well as the temporal information of the trajectories, in order to enhance the ego-vehicle's understanding of the environment. The spatio-temporal awareness is then used to derive *interaction scores* between the vehicles. Based on this, the ego-vehicle

The authors are with Shenzhen Campus of Sun Yat-sen University, China. Haoran Xu is also with the Peng Cheng Laboratory, Shenzhen 518055, China. Email: (Wentao Wang) wangwt66@mail2.sysu.edu.cn, (Haoran Xu) xuhr9@mail2.sysu.edu.cn.
*Correspondence: (Guang Tan) tanguang@mail.sysu.edu.cn

can identify the critical neighbors for data sharing. The main contributions of this paper are three-fold:

- We propose a method to extract the spatio-temporal information of neighboring CCVs and scene topology. It generates a set of interaction scores for cooperative perception fusion.
- We design an end-to-end cooperative driving model based on V2V communication. The model learns feature representations and communication policies that can enable the ego-vehicle to effectively utilize shared information to improve driving performance.
- Comprehensive experiments are conducted on the proposed CARLA autopilot benchmark to demonstrate the superiority and robustness of our proposed method in various scenarios compared with the state-of-the-art methods.

II. Related Work

A. End-to-end Autonomous Driving

In end-to-end autonomous driving, the final control signals are directly generated based on sensor input, bypassing the intermediate tasks [9], [10], [11]. The end-to-end approach is divided into two main branches: imitation learning and reinforcement learning (RL). The imitation learning method learns policy by imitating the collected expert driving data [12], while RL generalizes a policy by self-exploration and exploitation. In RL, an agent can optimize itself automatically without expert data [13], [14]. RL is widely known to be sample inefficient that millions of interactions are usually needed even for simple problem settings [15]. In this paper, we follow the end-to-end paradigm to learn driving policy.

B. Road Network and Trajectory Representation

Road network and trajectory representation learning play a crucial role in traffic systems, as the acquired representations can be used in a wide range of downstream tasks including autonomous driving [16], [17], [18], [19]. A road network is typically represented by a graph, which contains the topological and auxiliary contextual information of the traffic scene, while a trajectory is a sequence of consecutive road segments, which includes mobility information. In prior works [20], [21], representations were learned separately from the graph structure of road networks and the sequence information of trajectories. These methods ignored the relationship between the road network and trajectory. The recent study [17] has demonstrated that combining both road network and trajectory can generate better representation than learning each type of data independently. To improve on this, some two-stage methods [22], [23] have been proposed. They first learn representation for road segments, and then use it as a basis for learning trajectory representations. Different from these methods, we aim to learn the interactions between the ego- and neighboring vehicles. We design a unified self-attention mechanism, where the weights represent measurable interaction scores.

C. V2V Cooperative Perception

V2V cooperative perception enables the ego-vehicle to utilize shared information from neighboring vehicles through wireless communication, significantly overcoming the limitations of individual vehicles. Early studies [24], [25] on cooperative perception typically extended the sensing range of the vehicles by exchanging raw sensor information with neighboring vehicles. The complete sensor measurements are kept but a large bandwidth is required. This potentially increases the communication delay and affects real-time performance of perception.

To trade off the bandwidth and fusion effectiveness, intermediate fusion has been widely investigated, where intermediate features are transmitted through V2V channels [7]. SyncNet [26] proposed a latency compensation mechanism, which actively adapts asynchronous perceptual features from multiple agents to the same time stamp. [27], [28] employed a handshake mechanism to determine which two agents were selected to share information. In this work, we leverage the road network and trajectory information to enhance understanding of scene topology enable and more efficient fusion.

III. End-to-end V2V cooperative driving

A. Main Architecture Design

We consider common driving scenarios in which there are multiple nearby CCVs within the communication range of the ego-vehicle. The objective of this architecture is to directly acquire a control policy for the ego-vehicle utilizing V2V communication. The control policy's input signals are encoded primary perspective observations, denoted as O_{ego}, from the ego-vehicle, as well as fusion features, denoted as O_{fusion}, from the neighboring CCVs. The architecture is trained end-to-end. The overall architecture comprises four key components: 1) Point encoder, 2) Map and vehicle history encoding, 3) Attentive Relation-based Scorer, and 4) Control policy learning.

B. Point Encoder

Following the method proposed in [29], we utilize a transformer-based architecture to extract concise point-based representations from point clouds. By employing this architecture, the raw LiDAR inputs can be encoded as spatial-aware representations on a per-vehicle basis, facilitating perception fusion. These representations are compact and can be efficiently over wireless channels.

C. Map and Vehicle Trajectory Encoding

To effectively capture the scene's topology features, we utilize the local map and historical state information of vehicles. Since the road network can be represented as a directed graph, we employ a GAT-based network [30], which has been used for a wide variety of tasks [31]. Let us assume a road network $G(V, E)$, where V is the initial embedded vectors of the road segment, and E the adjacency matrix. We group the waypoints from the same map and aggregate their features using max-pooling. The resulting map feature is then

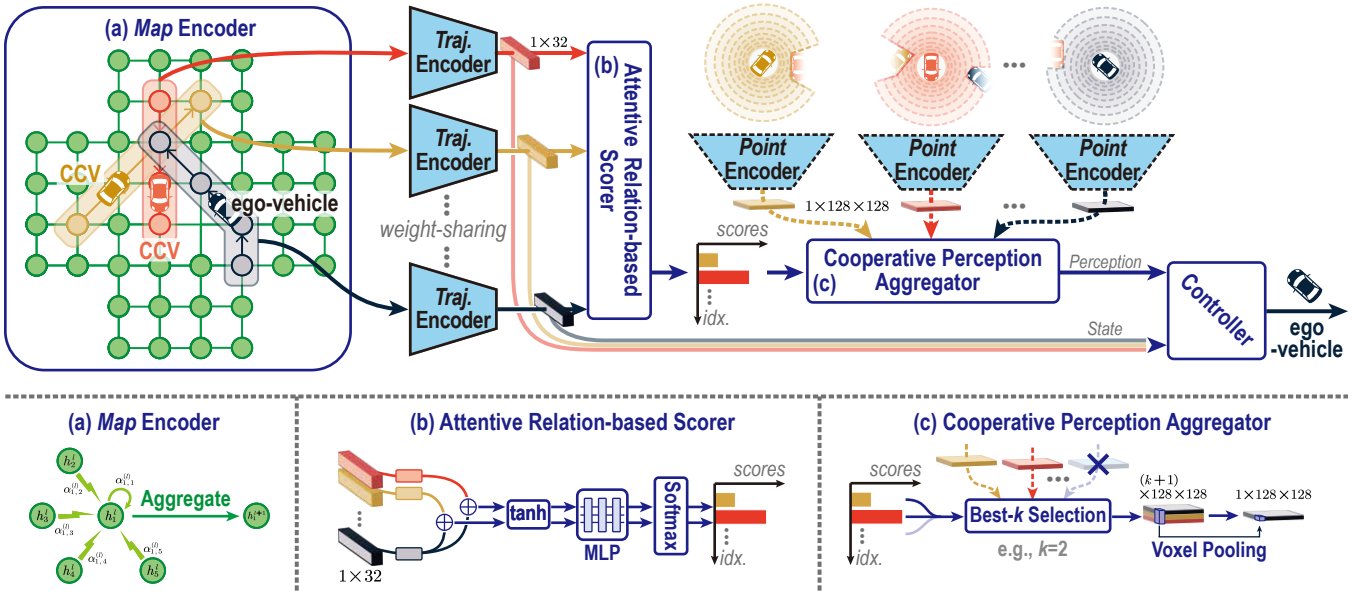


Fig. 2: InterCoop pipeline. We begin by extracting a scene representation of the road network using the *Map* Encoder. The vehicular trajectories are obtained based on the road representation, which is further encoded into compact features for the Attentive Relation-based Scorer (ARS). Within ARS, we attentively assign distinct scores to the neighboring CCVs. To aggregate information, we introduce the Cooperative Perception Aggregator (CPA) to combine the point features of the best-k neighboring CCVs and the ego-vehicle. Finally, the controller takes into account the neighboring states and cooperative perception to apply safer and more efficient actions.

reshaped into intermediate representations. Specifically, we consider road features such as position, yaw, and length as input.

The attentive interaction weight between segments i and j is formally defined as

$$e_{ij} = (W_i h_i + W_j h_j) W_a^\top, \tag{1}$$

$$\alpha_{ij} = \frac{\exp(LeakyReLU(e_{ij}))}{\sum_{k \in N_k} \exp(LeakyReLU(e_{ik}))}, \tag{2}$$

where h_i and h_j are representations of road segments i and j, respectively, and W_i, W_j, W_a are learnable weight parameters. Then, we can further extract the aggregated features of the road segment i as

$$\widetilde{h}_i^{l+1} = ELU(\sum_{j \in N_i} \alpha_{ij} W_b h_j^l), \tag{3}$$

where W_b is a learnable matrix and *ELU* is the exponential linear unit activation function. Given the GAT encoder, the representation of road segments can be learned. The above process is depicted in Map Encoder shown in Fig. 2.

After obtaining the road representations, we convert road representations into trajectory representations and incorporate temporal information. We select road segments contained in each trajectory to construct a spatio-temporal sequence. Specifically, for each vehicle, we use the encoded road segment features incorporating temporal information to denote its trajectory instead of the single sequence of waypoints [19]. Thus the fused embeddings of the road segment can be denoted as $x_i = Concat(r_i, t_i)$, where r_i is the road segment representations. Then we can obtain the initial representation of the trajectory:

$$X_i = \{x_1, ..., x_t\}. \tag{4}$$

The historical state sequences are encoded using a shared Long Short-Term Memory (LSTM) encoder [32]:

$$h_i^t = \text{LSTM}(h_i^{t-1}, X_i^t, W, b), \tag{5}$$

where h denotes the hidden state and the weights are shared for all vehicles to encode the temporal information of each one in the scene.

D. Attentive Relation-based Scorer

Instead of broadcasting all information throughout the entire network, a more efficient approach of utilizing shared information is to select key vehicles from which to obtain data.

After the unified road network and trajectory encoding, the encoded features include the spatial and temporal information about scene and the vehicles. The ego vehicle needs to pay attention to specific neighboring vehicles, depending on their past trajectories and the ego's observation. The intuition is, after obtaining the scene topology information, an ego-vehicle is able to know where there is a lack of insufficient information. Thereby, the perception perspective of the neighboring CCVs can be estimated based on the spatio-temporal encoder. Fusing the shared data from CCVs in specific areas can significantly improve the ego-vehicle's control decisions. Therefore, we propose an Attentive Relation-based Scorer (ARS) module, as depicted in Fig. 2, to effectively discover best-k neighboring CCVs for cooperative driving.

Specifically, CCVs first broadcast their encoded historical information within the communication range, and the ego-vehicle receives information and attentively computes the interaction scores with them. Then, the ego-vehicle selects the most critical k neighboring vehicles according to the scores. To accomplish this, we utilize an additive attention layer [33] to compute the interaction scores between vehicles:

$$s_{ij} = \Phi(u_i, k_j), \tag{6}$$

$$\Phi = W_p^T tanh(W_k u_i + W_q k_j), \tag{7}$$

where u_i, k_j are the state representations of the ego and neighboring vehicles, respectively. W_p, W_k and W_q are learnable parameters. The additive attention mechanism allows for the query and key to be vectors with different sizes.

Based on the above cross-attention mechanism across all queries and keys, we can derive the score vector:

$$S_{ego} = \sigma\{s_{i1}, s_{i2}, ..., s_{iN}\}, \tag{8}$$

where σ is the Softmax activation function. Then, the selected critical agents with the n highest scores will send perception information to the ego-vehicle.

The ego-vehicle combines the perception information received from neighboring CCVs with its observations. We aggregate two types of features: shared perception information and transmitted agent historical states. To accomplish this, we use two separate representation aggregators.

Firstly, Cooperative Perception Aggregator (CPA) is proposed as shown in Fig. 2. The points in the supporter agents' perpetual perspective will be spatially transformed into the ego-vehicle's coordinates. We use a voxel-pooling layer to map the close points into the same voxel grids for representations. This reduces data redundancy and preserves sufficient spatial information. Then a point transformer block is adopted to fuse the perception information received from selected CCVs' view. Secondly, all of the trajectories of the CCVs will be aggregated into a fixed-shape tensor, to form the input tensor of historical agent states for the Controller module. Considering the historical states of each CCV have been encoded as a spatio-temporal sequence, we concatenate them and employ MLPs to capture inherent features in the high-dimensional latent space. Then we utilize a max-pooling layer to aggregate the representations.

E. Control Policy Learning

The control module employs a fully-connected neural network that is specifically designed to make control decisions based on the received messages. The output control decision is a combination of throttle, brake, and steering, which are denoted as T, B, and S, respectively.

We train our model via imitation learning with the expert policy. To eliminate the weakness of relying too heavily on expert data in imitation learning, we employ DAgger [34] to suit our specific experimental conditions. Specifically, the training process for the control policy has two steps: behavior cloning and data aggregation. Firstly, the policy π is trained in a supervised manner using the collected dataset of state-action pairs:

$$\pi = \arg\min_{\pi} E_{(s^*, a^* \sim P^*)}[\mathcal{L}(a^*, \pi(s^*))], \tag{9}$$

where P^* represents the state distribution provided by expert policy π^* and the loss function $\mathcal{L}$ is calculated by the $L1$ losses between the output signals and the expert policy's control signals:

$$\mathcal{L} = \underbrace{|T_{out} - T_{Exp.}|}_{throttle} + \underbrace{|B_{out} - B_{Exp.}|}_{brake} + \underbrace{|S_{out} - S_{Exp.}|}_{steering}. \tag{10}$$

Then, by following the convention [35], we adopt a modified DAgger as an on-policy sampling approach to select critical states and utilize a replay buffer to minimize the uncertainty associated with the learned model.

IV. Experiments

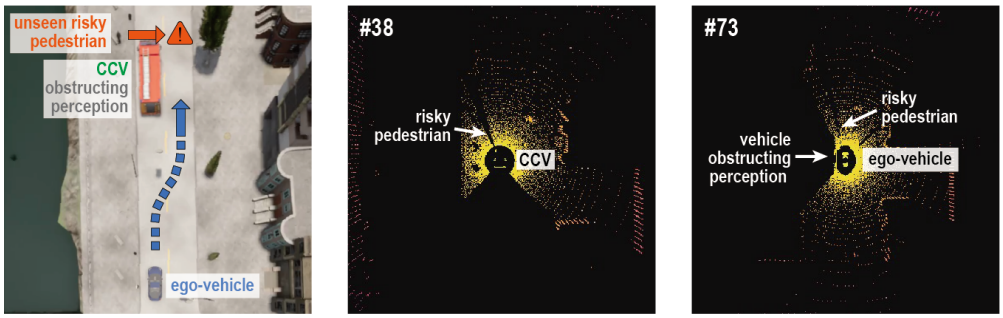
A. Experimental Setup

1) Driving Scenario Configuration: To assess the effectiveness of our approach, we created two common and challenging driving scenarios utilizing the CARLA simulator [36], as illustrated in Fig. 3. Two typical scenarios have been carefully chosen from the crash avoidance research conducted by the US National Highway Traffic Safety Administration [37], both of which involve potential collisions that can be mitigated through cooperative perception methods.

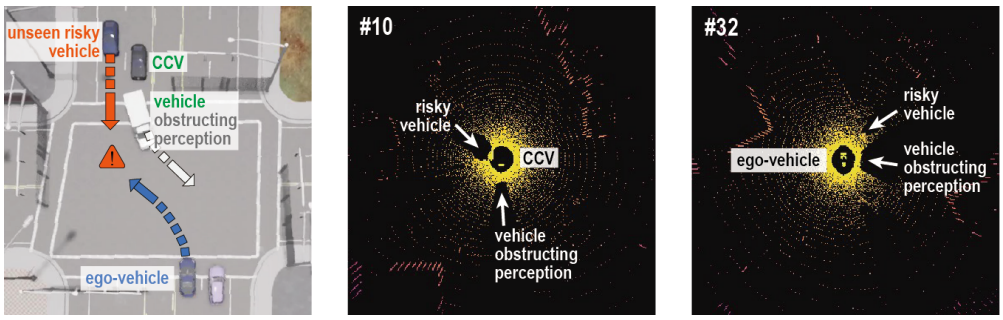
- Jaywalking. Fig. 4a depicts a scenario where the ego-vehicle is driving on a one-way road with two lanes, while one of the lanes is blocked by a parked CCV. At the same time, a pedestrian carelessly crosses the road from the ego's blind spot.
- Intersection. Fig. 4b depicts a scenario where the ego-vehicle turns left without noticing the vehicle driving downwards the crossroad. This is due to the presence of a white vehicle in front of it, obstructing the ego-vehicle's view.

2) Metrics: We assess the decision-making performance in terms of safety, stability and completion.

- Collision Rate (CR) measures the percentage of generated traces where the ego-vehicle collides with any other objects. It is typically used to evaluate driving safety.
- Hazard Frequency (HF) [25] measures the frequency of events where the Time-to-Collision (TTC) is less than a predefined threshold (TTC_t). The TTC represents the estimated time it would take for two vehicles to collide if they were to continue with their current control signals. The threshold TTC_i differentiates between safe and unsafe events. HF is used to evaluate stability.
- Success Rate (SR) measures the percentage of successful completions among all evaluated traces. Successful completion of the scenario is defined as the ego-vehicle reaching a designated target location in a permissible time without collision or prolonged stagnation.



(a) The potential risk in the narrow lane drop scenario. At time #38, CCV can detect the pedestrian and inform the ego-vehicle to apply early brake (second column). However, if the ego-vehicle only detects the pedestrian at time #73, it may not have sufficient time to avoid a collision (third column).



(b) The potential risk in the arterial intersection scenario. At time #10, CCV can detect a pre-colliding vehicle and inform the ego-vehicle to apply brake in advance. However, when the ego-vehicle reaches to the middle of the intersection at time #32, both vehicles detect each other, but it is too late to prevent the collision.

Fig. 3: The proposed scenarios highlight the importance of cooperative driving.

3) Dataset: Following the convention [9], we employ an expert agent to generate an initial training set. Specifically, our dataset is generated through simulation utilizing an oracle driving policy. This policy leverages the precise privileged information about the environment to accurately predict their future trajectories. In our setting, we use the Town01 and Town03 maps of the CARLA simulator. The A* search algorithm is employed to determine the actions for the ego-vehicle at each time step, ensuring its safe and collision-free navigation towards the desired goal. Additionally, we construct the road graph using map data of the scenario provided by CARLA. We first sample waypoints from the center trajectory of the lanes at intervals of 3m. Then we can obtain a list of the road segments which includes several waypoints. Since the map also provides the linkage and direction information, we can connect the adjacent segment and then obtain the graph of road segments.

4) Implementation Details: All experiments are conducted on the NVIDIA GeForce RTX 3080 GPU. The policy training consists of two stages: behavior cloning and dataset aggregation. We initially train the model using expert policies and then utilize Dagger [35] for interaction with the environment. To prioritize driving safety, we sample the deviation in brake as the policy since we have observed inadequate braking in most failure cases. To mitigate overfitting, we construct a replay buffer with a size of 5, where the proportion of expert and on-policy samples is initialized at $\beta = 0.8$ and decreases during training. To ensure fairness, we adhere to the baseline method and conduct driving simulations for each constructed scenario using 27 different attribute configurations. Each simulation configuration is repeated 3 times to evaluate all methods consistently.

All neighboring vehicles send the spatio-temporal features encoded by the local trajectory encoder to the ego-vehicle. The ego-vehicle computes the interaction scores and selects best-k ($k = 3$) crucial neighbors to connect. The point clouds are encoded as compact features with the size of $[128, 128]$, which preserve the position information. We aggregate both the received historical states and shared perception information.

B. Ablation Study

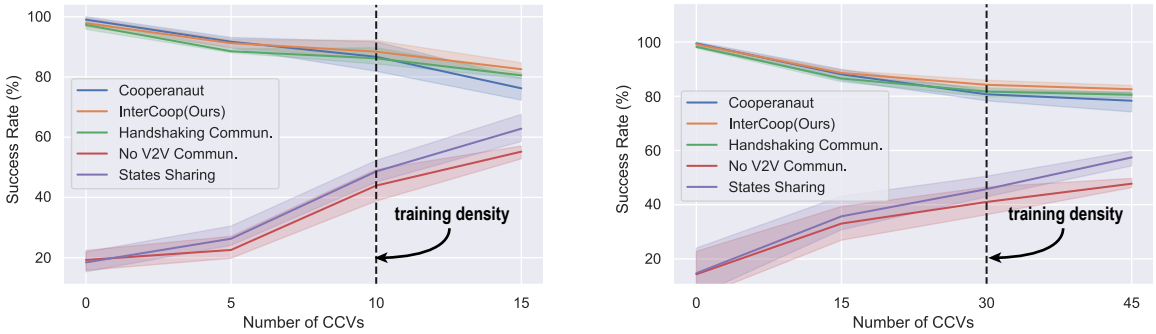
This section presents the empirical evaluations of the proposed models in the three common but dangerous scenarios.

1) Overall Performance: We consider the following baseline methods for comparison:

- No V2V Communication: The non-communication baseline makes control decisions only based on the onboard LiDAR data. This model uses the same point encoder as our model.
- States Sharing: The States Sharing baseline makes control based on the received neighboring CCVs' state information and onboard LiDAR data. This model shares the same module as our model but does not include cooperative perception.

Models	Jaywalking			Intersection			Average		
	CR↓	HF↓	SR↑	CR↓	HF↓	SR↑	CR↓	HF↓	SR↑
No V2V Commun.	38.6	11.7	43.9	55.6	15.2	40.5	47.1	13.4	42.2
States Sharing.	36.1	10.8	48.7	46.7	12.9	44.1	41.4	11.9	46.4
Coopernaut [9]	7.4	5.3	86.7	18.1	8.4	80.7	12.8	6.9	83.7
Handshaking Commun. [27]	7.1	4.5	86.2	16.4	8.1	81.3	11.8	6.2	84.0
InterCoop (Ours)	6.3	3.6	88.4	14.5	6.7	84.2	10.5	5.2	86.2

TABLE I: Quantitative results of different models for driving policies. (The best results are highlighted in gray background.)



(a) Robustness capability in Jaywalking.

(b) Robustness capability in Intersection.

Fig. 4: Robustness capability in different scenarios with various traffic densities.

- Coopernaut [9]: Coopernaut is an end-to-end autonomous driving framework that relies on V2V cooperative perception. Each CCV encodes its local point cloud and transmits it to the ego-vehicle for aggregation. To address bandwidth limitations, the method randomly selects three CCVs to establish connections in dense traffic scenarios.
- Handshaking Communication [27]: Who2comm employs a handshaking mechanism to learn driving policies. This approach is designed for collaborative object detection and forecasting. The ego-vehicle encodes the local point cloud and sends a request message to neighboring CCVs. The ego-vehicle selects the best-k candidates based on matching scores received from the CCVs. The requested size is fixed at 32, and the received data is encoded point data. The control module then utilizes this data to generate control decisions.

Table I reports the performance in terms of all metrics on two scenarios. The No V2V Commun has performed poorly with the highest CR and lowest SR, and the States Sharing method yields better performance, indicating that the proposed spatio-temporal encoder provides critical information about the traffic situation including road networks and other CCVs. However, only state sharing cannot overcome the limitation of the blind spots. All three cooperative perception models achieved reliable SR scores as well as lower CR and HF. Both of Handshaking Commun. and our model outperform the Coopernaut baseline in two scenarios. Thus we can confirm the reasonable selection of crucial CCVs enables a more effective use of cooperative perception.

2) Robustness Analysis: Furthermore, we evaluate the performance of our proposed method in different scenarios with various traffic densities. As shown in Fig. 4, V2V techniques may not yield substantial improvements in denser traffic conditions due to their vulnerability to increased stochasticity resulting from the variability of incoming messages from neighboring vehicles. We observe that the performance of the Coopernaut deteriorates and fluctuates under high traffic densities due to increased uncertainty, which hinders the obtaining of crucial information. In contrast, our method exhibits superior robustness across different scenarios with varying traffic densities.

V. Conclusion

In this paper, we presented a new cooperative driving framework that leverages the V2V communication. We extract the spatial topology of the map, as well as the temporal information of the trajectories to capture spatio-temporal features. Then an attention mechanism is applied to learn the interaction between neighbors and output measurable scores. Thus the ego-vehicle can receive information from crucial vehicles for fusion. The driving policy is trained end-to-end. Experimental results demonstrate the robustness of our method to traffic density in common potential risky scenarios.

VI. Acknowledgements

This work was supported in part by Shenzhen Basic Research Fund under grant JCYJ20200109142217397.

References

[1] Q. Yang, S. Fu, H. Wang, and H. Fang, "Machine-learning-enabled cooperative perception for connected autonomous vehicles: Challenges and opportunities," *IEEE Network*, vol. 35, no. 3, pp. 96–101, 2021.

[2] M. Vitelli, Y. Chang, Y. Ye, A. Ferreira, M. Wołczyk, B. Osiński, M. Niendorf, H. Grimmett, Q. Huang, A. Jain, *et al.*, "Safetynet: Safe planning for real-world self-driving vehicles using machine-learned policies," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 897–904.

[3] Y. Bian, Y. Zheng, W. Ren, S. E. Li, J. Wang, and K. Li, "Reducing time headway for platooning of connected vehicles via v2v communication," *Transportation Research Part C: Emerging Technologies*, vol. 102, pp. 87–105, 2019.

[4] F. Zhou and Y. Chai, "Near-sensor and in-sensor computing," *Nature Electronics*, vol. 3, no. 11, pp. 664–671, 2020.

[5] R. Xu, H. Xiang, Z. Tu, X. Xia, M.-H. Yang, and J. Ma, "V2x-vit: Vehicle-to-everything cooperative perception with vision transformer," in *European conference on computer vision*. Springer, 2022, pp. 107–124.

[6] T.-H. Wang, S. Manivasagam, M. Liang, B. Yang, W. Zeng, and R. Urtasun, "V2vnet: Vehicle-to-vehicle communication for joint perception and prediction," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*. Springer, 2020, pp. 605–621.

[7] Y. Li, S. Ren, P. Wu, S. Chen, C. Feng, and W. Zhang, "Learning distilled collaboration graph for multi-agent perception," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 541–29 552, 2021.

[8] Y. Hu, S. Fang, Z. Lei, Y. Zhong, and S. Chen, "Where2comm: Communication-efficient collaborative perception via spatial confidence maps," *Advances in neural information processing systems*, vol. 35, pp. 4874–4886, 2022.

[9] J. Cui, H. Qiu, D. Chen, P. Stone, and Y. Zhu, "Coopernaut: End-to-end driving with cooperative perception for networked vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 252–17 262.

[10] 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*, vol. 33, no. 4, pp. 1364–1384, 2020.

[11] Y. Xiao, F. Codevilla, A. Gurram, O. Urfalioglu, and A. M. López, "Multimodal end-to-end autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 1, pp. 537–547, 2020.

[12] 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)*. IEEE, 2018, pp. 4693–4700.

[13] A. Kendall, J. Hawke, D. Janz, P. Mazur, D. Reda, J.-M. Allen, V.-D. Lam, A. Bewley, and A. Shah, "Learning to drive in a day," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 8248–8254.

[14] X. Liang, T. Wang, L. Yang, and E. Xing, "Cirl: Controllable imitative reinforcement learning for vision-based self-driving," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 584–599.

[15] T. Yang, H. Tang, C. Bai, J. Liu, J. Hao, Z. Meng, P. Liu, and Z. Wang, "Exploration in deep reinforcement learning: a comprehensive survey," *arXiv preprint arXiv:2109.06668*, 2021.

[16] S. Gong, H. Zhou, F. Xue, C. Fang, Y. Li, and Y. Zhou, "Fastroadseg: Fast monocular road segmentation network," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 11, pp. 21 505–21 514, 2022.

[17] J. Jiang, D. Pan, H. Ren, X. Jiang, C. Li, and J. Wang, "Self-supervised trajectory representation learning with temporal regularities and travel semantics," in *2023 IEEE 39th International Conference on Data Engineering (ICDE)*. IEEE, 2023, pp. 843–855.

[18] C. Luo, L. Sun, D. Dabiri, and A. Yuille, "Probabilistic multi-modal trajectory prediction with lane attention for autonomous vehicles," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 2370–2376.

[19] T.-Y. Fu and W.-C. Lee, "Trembr: Exploring road networks for trajectory representation learning," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 1, pp. 1–25, 2020.

[20] N. Wu, X. W. Zhao, J. Wang, and D. Pan, "Learning effective road network representation with hierarchical graph neural networks," in *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*, 2020, pp. 6–14.

[21] T. S. Jepsen, C. S. Jensen, and T. D. Nielsen, "Graph convolutional networks for road networks," in *Proceedings of the 27th ACM SIGSPATIAL international conference on advances in geographic information systems*, 2019, pp. 460–463.

[22] Y. Chen, X. Li, G. Cong, Z. Bao, C. Long, Y. Liu, A. K. Chandran, and R. Ellison, "Robust road network representation learning: When traffic patterns meet traveling semantics," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021, pp. 211–220.

[23] P. Han, J. Wang, D. Yao, S. Shang, and X. Zhang, "A graph-based approach for trajectory similarity computation in spatial networks," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 556–564.

[24] Q. Chen, S. Tang, Q. Yang, and S. Fu, "Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2019, pp. 514–524.

[25] R. Xu, H. Xiang, X. Han, X. Xia, Z. Meng, C.-J. Chen, C. Correa-Jullian, and J. Ma, "The opencda open-source ecosystem for cooperative driving automation research," *IEEE Transactions on Intelligent Vehicles*, 2023.

[26] Z. Lei, S. Ren, Y. Hu, W. Zhang, and S. Chen, "Latency-aware collaborative perception," in *European Conference on Computer Vision*. Springer, 2022, pp. 316–332.

[27] Y.-C. Liu, J. Tian, C.-Y. Ma, N. Glaser, C.-W. Kuo, and Z. Kira, "Who2com: Collaborative perception via learnable handshake communication," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6876–6883.

[28] Y.-C. Liu, J. Tian, N. Glaser, and Z. Kira, "When2com: Multi-agent perception via communication graph grouping," in *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, 2020, pp. 4106–4115.

[29] H. Zhao, L. Jiang, J. Jia, P. H. Torr, and V. Koltun, "Point transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 16 259–16 268.

[30] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *arXiv preprint arXiv:1710.10903*, 2017.

[31] Y. Dong, Q. Liu, B. Du, and L. Zhang, "Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 31, pp. 1559–1572, 2022.

[32] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 961–971.

[33] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[34] S. Ross, G. Gordon, and D. Bagnell, "A reduction of imitation learning and structured prediction to no-regret online learning," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[35] A. Prakash, A. Behl, E. Ohn-Bar, K. Chitta, and A. Geiger, "Exploring data aggregation in policy learning for vision-based urban autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 763–11 773.

[36] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[37] W. G. Najm, J. D. Smith, M. Yanagisawa, *et al.*, "Pre-crash scenario typology for crash avoidance research," United States. National Highway Traffic Safety Administration, Tech. Rep., 2007.