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

Mitigating Causal Confusion in Vector-Based Behavior Cloning for Safer Autonomous Planning

Jiayu Guo[1], Mingyue Feng[2], Pengfei Zhu[2], Jinsheng Dou[2], Di Feng[2], Chengjun Li[2], Ru Wan[2] and Jian Pu[*,1]

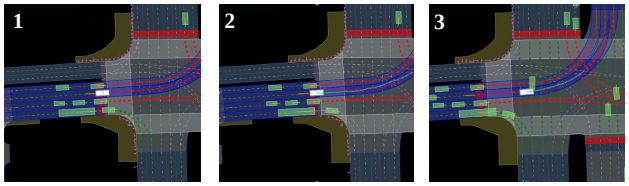
***Abstract*— The utilization of vector-based deep learning techniques has great prospects in the realm of autonomous driving, particularly in the domains of prediction and planning tasks. However, the application of vector-based backbones for prediction and planning tasks may lead to the occurrence of causal confusion. Previous studies have explored the phenomenon of causal confusion, with a specific emphasis on the context of visual imitation learning. As for the vector-based model, we observe that the states of surrounding vehicles can be a nuisance shortcut. In our work, an off-policy approach is proposed to alleviate the issue by incorporating de-confounding supervision. Additionally, to better capture the environmental cues, such as route and traffic lights, in vectorized representation, a decoder utilizing iterative route fusion is devised. By incorporating auxiliary supervision and employing a dedicated decoder, we demonstrate the effectiveness of our methods in reducing causal confusion and improving performance in planning tasks through reactive and nonreactive closed-loop simulations on the nuPlan dataset.**

I. INTRODUCTION

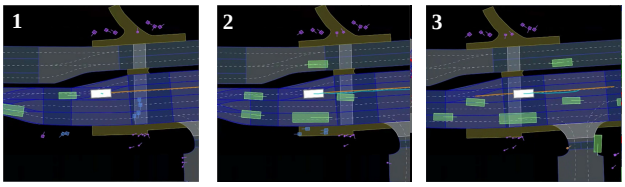
The utilization of deep learning (DL) techniques in vehicle planning shows great potential for autonomous driving. There is a growing interest in DL planners with prediction, taking as input the mid-representations [1] such as the output of perception systems or HD maps. In this field, vector-based representation has been widely employed [1]–[6].

However, for the vector-based method, causal confusion [7] from nearby vehicles may occur if the planning model simply shares the same backbone [8] with the prediction task. In Fig.1, there are two examples of causal confusion brought on by neighboring vehicles. In brief, the situation in the first row is a vehicle that starts erroneously when it detects a moving car, and the second involves a vehicle that does not start until there is a forwarding car nearby. Both instances show that the neural networks build a spurious correlation between the speed of the ego and nearby cars while neglecting the actual factors that should determine vehicle behaviors.

Previous researches on causal confusion in DL planner concentrate on End-to-End (E2E) imitation learning [9]–[11], denoted as inertia problem [12]. The inertia problem arises from the adoption of an imitation policy that erroneously associates current planning with the ego history and subsequently extrapolates actions from history, ultimately resulting in poor performance during deployment. To address causal confusion, removing confounding variables (e.g., historical information) [13], [14], or changing the distribution of data and increasing the proportion of informative samples [11], [15] are considerable.



(a) Ego car started before the red light when observing a right-turn car and crashed.



(b) Ego car kept stationary until a car moved forward.

Fig. 1: Scenarios where nearby cars cause causal confusion. Closed-loop Simulations are conducted with the ego car using a planned neural network and other agents utilizing log-replay. The ego car is represented by a white box, and nearby vehicles are in green. A gray line represents planned driving by the neural network, and an orange line represents expert demonstration.

In contrast to ego history, the speed of nearby vehicles is rarely reported as a potential confounding input. However, there exists a significant number of trivial scenarios [12], where there is a noticeable correlation between the surrounding vehicle states and ego future planning, such as following the car in front or stopping at a red light. In the vector-based model, the velocities of surrounding cars are explicitly represented as low-dimension vectors, instead of being latently encoded in high-dimension features of multi-frame inputs in the E2E model. It is more prone to produce spurious correlation when correlated low-dimension features are exposed, and therefore, surrounding vehicle states should be taken into account as causes of causal confusion in addition to ego history.

Our work aims to mitigate the issues of causal confusion inside the vector-based model while maintaining the integrity of the scenario representation. Our intuition is that by implementing de-confounding supervision and dedicated scenario encoding, the neural network becomes more attuned

*Corresponding author, jianpu@fudan.edu.cn
[1]Institute of Science and Technology for Brain-Inspired Intelligence, Fudan University jyguo20@fudan.edu.cn
[2]Mogo Auto Intelligence and Telematics Information Technology Company Ltd. {fengmingyue, zhupengfei, doujinsheng,james, wanru}@zhidaoauto.com, fd940056938@163.com

to environmental cues of the driving intents, instead of simply mimicking surrounding behaviors. Our contributions can be summarized as follows:

- To alleviate causal confusion in the vector-based models, an off-policy solution named De-Confounding Supervision (DCS) is proposed by employing a trajectory supervision on de-confounding features.
- Different candidates of DCS labels are compared to demonstrate that the ground truth (GT) or expert demonstration leads to substantial improvement.
- A trajectory decoder with Iterative Route Fusion (IRF) is developed to enhance the learning of traffic light regulations and route information, reducing erroneous starting before red light and the off-road risks regarding accumulating errors in closed-loop simulation.

II. Related Work

A. Learning-based Planning Model

Plenty of studies have been conducted on vision-based End-to-End (E2E) planning models that output control signals directly using either a monocular camera [12], [16] or a multicamera system [17]. Lidar scans can also be used to improve performance [18]. In this field, both IL [17], [19] and RL [20] methods have been studied, and numerous researchers have combined them to overcome their own drawbacks and enhance performance [21], [22].

In addition to the E2E model [12], [18], which utilizes raw sensor data, there has been a growing trend towards the adoption of models that incorporate the output of the perception system and a high-definition (HD) map in modular autonomous stacks [1], [2], [4]. It involves encoding the environment context around the ego car as vector graphs [1], [6] or bird's eye view (BEV) raster images [3]. Behavior Cloning (BC) [23] is a widely employed technique in the field of planning tasks, primarily due to its ease of use and the abundance of extensive driving datasets.

With the advancement of trajectory prediction models [24], [25], an increasing amount of research has begun to focus on neural networks integrated with prediction and planning [2], [6]. The training process for unified models, which can handle both prediction and planning tasks, can be viewed as a form of multi-task supervised learning [6], [26]. It is believed that auxiliary supervision, like nearby agents' behavior prediction, can facilitate ego planning [4], [18]. As one kind of successful application, post-optimization methods using forecasting and planning outputs to avoid collision are employed to make planning more robust [5], [6]. In this field, capturing scenario context with vector representations has gained popularity [27]. However, previous works have rarely explored whether the vectorized encoding of surrounding vehicles negatively impacts planning.

B. Causal Confusion

Causal confusion refers to a spurious correlation between model input and output. In autonomous driving, the inertia problem [12] is brought up that the neural networks may employ a shortcut by extrapolating from historical information, especially when the scenes and labels change smoothly. Relevantly, [28] recognizes that the historical information of ego is nuisance input which results in spurious correlation between historical and future decisions, and [10] understands the inertia problem as a manifestation of shortcut learning [29].

To tackle this problem, a 50% dropout on past ego poses is introduced during training in [3] to force the BEV-based model to consider other environmental cues. More works [12], [13], [30] directly abandon the ego history to avoid causal confusion. In addition, [10] and [7] ensemble the multi-frame network with a single-frame one, incorporating task-specific domain knowledge in a manner of auxiliary supervision.

Another approach to address the inertia problem is increasing the importance or proportion of complex scenarios in the training dataset. [11] defines the moments as keyframes when the future states cannot be predicted by historical states alone and increases the weight of these samples. DAgger [31] and DAgger-like [15] algorithms aggregate new scenarios from failure cases after querying expert to update data distribution.

Previous investigations on the subject of causal confusion have primarily concentrated on the ego history in visual or BEV-based image networks. The work most similar to ours is [9], which also addresses causal confusion through the auxiliary supervision. However, a key disparity exists in our interest of vector-based approach compared to their end-to-end visual imitation learning. Also, the confounding variables that are taken into account in [9] merely pertain to ego history, instead of the environmental states that are explored in our research.

III. Method

Mitigating causal confusion entails two aspects: firstly, reducing dependence on confounders, and secondly, enhancing the capability of model to extract environmental cues. In Sec.III-A, the vector-based graph backbone is introduced to establish a comprehensive understanding as shown in Fig.2. Sec.III-B outlines the architecture of LSTM with iterative route fusion (LSTM-IRF), capable of robust feature extraction. Sec.III-C details the extraction of the De-Confounding (DC) ego feature and discuss the selection of appropriate labels for training DC features. Post-optimization is in Sec.III-D and Sec.III-E introduces the training loss. Our work primarily focuses on enhancing the trajectory learning efficiency of vector-based imitated models. As a result, non-neural network modules, such as emergency braking or driving initialization from standstill, are not incorporated.

A. Preliminary: Vector-based Graph Backbone

Vector representation is commonly used for scenario encoding in trajectory prediction within ego-centric coordinates [27]. However, an alternative approach is proposed by encoding all agents in the scenario based on their local reference frame [8], aiming to enhance data efficiency in trajectory learning. Our model adopts the same strategy, opting to share

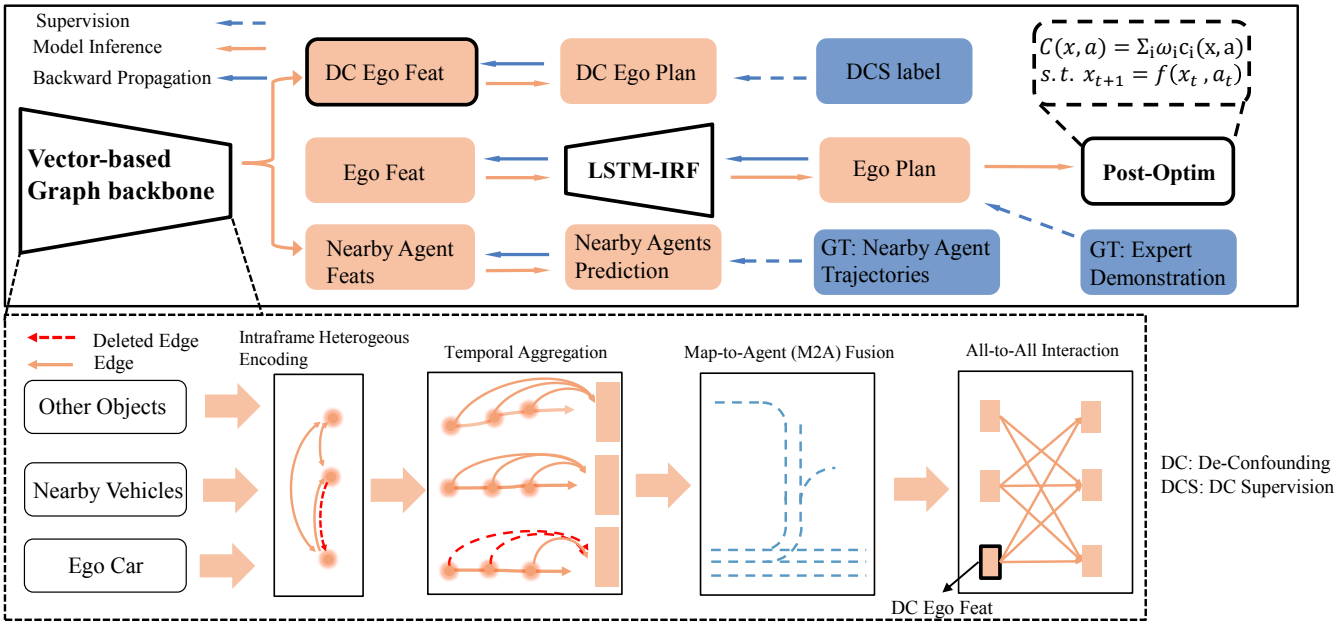


Fig. 2: Overview of the model architecture. The model comprises a vector-based graph backbone, an LSTM-IRF planner decoder, and a post-optimization module. The backbone produces ego features, de-confounding (DC) ego features, and nearby agent features. These features are supervised by distinct signals. The MLP decoder leverages nearby agent features to produce trajectory predictions. The DC ego feature is supervised by the de-confounding supervision (DCS) label. The ego feature is fed into an LSTM with iterative route fusion (LSTM-IRF) to generate ego planning. Finally, a post-optimization step refines the planning trajectory.

this backbone for both planning and prediction tasks. To effectively capture the interaction among different types of road participants, we employ a single heterogeneous graph neural network with four stages.

1) Intra-frame Heterogeous Encoding: Intra-frame heterogeneous fusion is conducted among the traffic participants in the same timestamp, and an gated attention is applied for feature aggregation like [8].

2) Temporal Aggregation: Temporal aggregation aims to capture and fuse the dynamic information of each traffic participant by temporal-wise attention, aggregated by one additional learnable embedding.

3) M2A Fusion: The graph establishes unidirectional edges from lane segments to traffic agents. The lane attributes like traffic lights and route states are encoded in node features.

4) All-to-All Interaction: This module utilizes an ALL-to-ALL graph to represent global interaction between traffic participants. A skip connection is established across the module.

B. LSTM-IRF

Unlike the prediction task performed with no knowledge of their destinations, the planning task can be enhanced by route information. In this module, we represent the route as a sequence of lane centerlines $p_i^{(r)}$ and their direction $v_i^{(r)}$, where the lane centerlines on the route are preserved, and the rest are masked. LSTM with Iterative Rout Fusion (LSTM-IRF) is employed to aggregate the route information and ego information and infer the planning waypoints iteratively. In each time t, the decoder generate route features r^t by graph attention:

$$r^t = MLP(x^t) + \Sigma_{i \in N_t} \alpha_i^t v_i^t$$

where N_t is the set of route centerlines close to planned position x^t. α_i^t and v_i^t are attention and value embedding built from route at time t.

Given route feature r^t, memory embedding h^t and c^t, the decoder reasons about the positions in T moments as follows:

$$f^t, h^{(t+T)}, c^{(t+T)} = LSTM(Linear(x^t) + r^t, h^t, c^t)$$
$$x^{(t+1:t+T)} = MLP(cat(x^t, f^t, z))$$

where x^t and z are planned position at time t and ego features in Fig.2 .

In order to enhance comprehension of traffic light rules, it is proposed to mask the route centerlines in red state and their successive route lanes. Upon the transition of a red traffic light to green, a notable change of input will occur where the route inputs to IRF will shift from being empty in the front to including route centerlines, which is easier to learn by the model.

C. De-Confounding Supervision

It is assumed that causal confusion sources from not only the ego histories but also nearby cars. De-confounding supervision (DCS) is introduced as an auxiliary task that assists in learning other environmental cues, preventing neural networks from overemphasizing the features of ego histories and nearby cars.

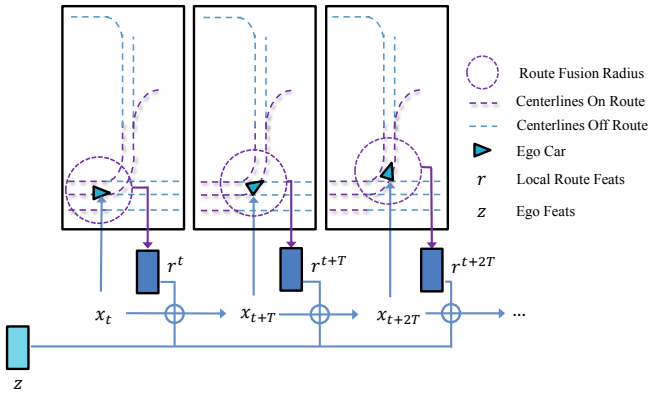


Fig. 3: LSTM with Iterative Route Fusion (LSTM-IRF).

1) Building DC features: In intra-frame heterogeous encoding, the edges of graph message passing are strictly controlled, where only messages from ego to nearby vehicles are being passed, but not vice versa. This stage ensures that the ego feature is visible to nearby cars, thereby enhancing subsequent trajectory predictions. However, surrounding cars remain invisible to the ego, preserving its independence from surrounding vehicle states until all-to-all interaction. Specifically, to avoid inertia issues [12] in subsequent planning processes, the histories of ego cars are masked when temporal aggregation is performed. After M2A fusion, DC ego features are constructed, totally isolated from potential confounders. The ego features learns to interact only in All-to-All interaction encoding, being aware of surrounding vehicles.

2) Selecting supervision labels: Intuitively, the ideal candidate labels supervising the de-confounding feature should indicate the potential driving intention of the ego. To obtain such trajectories, the non-learning algorithm Predictive Driver Model (PDM) [32] ignoring nearby vehicles is utilized to generate the reference trajectories as supervising labels (Ref-DCS). Pedestrians, bicyclists, and traffic cones will still influence the driving decisions made by PDM, except the surrounding vehicles. Another candidate for DCS signal is the ground truth (GT) or expert demonstration (GT-DCS). Comparisons between these candidates is shown in the experimental part.

D. Post Optimization

To prevent from covering up the issues we are concerned about, post optimization is only with comfortable objectives and kinematic constraints, ensuring it does not influence behavioral decision-making. The kinematic model we use is the unicycle model, and the numerical method for integration is fourth-order Runge-Kutta (RK4) with $\frac{3}{8}$ rule whose total accumulated error is $O(\Delta t^4)$ [33] to reduce the numerical cumulative error:

$$\begin{aligned} \min_{u,a} C(u,a;y) = &\sum_t (||x_t - y_t||_2) + \sum_t (\omega_1 a_t^2 + \omega_2 u_t^2) \\ s.t. \quad & x_{t+1} = f_{RK4}(x_t, a_t, u_t) \\ & x_0 = y_0 \end{aligned}$$

where a is acceleration, u is steering rate, y is the output of neural network, x is the trajectory passed to the controller and ω_i are objective weights. The optimization program is solved by Ipopt [34].

E. Loss Functions

As the multimode prediction designed in the motion prediction task, the regression loss on the trajectory and the classification loss of the best trajectory are combined to form our prediction/planning loss. For trajectory learning, we apply a winner-takes-all strategy [8] that we only optimize the trajectory closest to ground truth at the end of the horizon as shown below:

$$L_{reg} = \frac{1}{T_h} \sum_t ||x_{j^*}^{(t)} - x_{gt}^{(t)}||_1$$

where $j^* = \underset{j}{argmin} ||x_j^{t_{end}} - x_{gt}^{t_{end}}||$, and t_{end} is the end of planning or prediction horizon.

For the classification loss, the label q_j for possibility p_j of prediction/planning output x_j is calculated based on the final displacement between prediction and ground truth:

$$L_{conf} = -\frac{1}{M} \sum_j q_j log(p_j)$$

where $q_j = softmax(-a||x_j^{t_{end}} - x_{gt}^{t_{end}}||_1)$.

Additionally, to improve the accuracy of trajectory with the highest probability, denoted as $x_{\bar{j}}$ where $\bar{j} = \underset{j}{argmax}\, p_j$, $x_{\bar{j}}$ is also optimized in each learning step:

$$L_{highest} = \frac{1}{T_h} \sum_t ||x_{\bar{j}}^{(t)} - x_{gt}^{(t)}||_1$$

The loss functions for plan/prediction tasks are as follows:

$$L_{plan/pred} = L_{ref} + L_{conf} + L_{highest}$$

Therefore, the total loss of our method is formed as:

$$L = \alpha_1 L_{plan} + \alpha_2 L_{pred} + \alpha_3 L_{DCS}.$$

IV. Experiments

A. Dataset

All experiments are conducted on the nuPlan [37] dataset and its simulator, and it contains approximately 1300 hours of driving data from Las Vegas, Boston, Pittsburgh, and Singapore, with high complexity and various scenarios. The evaluation of our method utilizes the Val14 benchmark [32] since the nuPlan challenge has been closed. This benchmark consists of 1,118 scenarios for 14 scenario types. In the Reactive Closed-Loop (R-CL) simulation, all other vehicles are utilizing an IDM planner. In the context of Non-Reactive Closed-Loop (NR-CL) simulation, background traffic vehicles are log replay. In CL simulation, the trajectory from our model is fed into a two-stage controller of an LQR tracker and a kinematic model to generate control signals.

The driving score in nuPlan is a weighted aggregation of multiple metrics. To provide a comprehensive understanding

TABLE I: Driving scores of each scenario category in *Val14* [32] benchmark. *BLV*:Behand Long Vehicle. *ST*:Stationary in Traffic. *SL*: Stopping with Lead. *HMS*:High Magnitude Speed. *NMV*:Near Multiple Vehicles. *LMS*: Low Magnitude Speed. *CL*: Change Lane. *FLL*:Follow Lane with Lead. *STLT*: Straight Traffic Light Trasversing. *HLA*: High Lateral Speeds. *LT*: Left Turn. *RT*: Right Turn. *TPD*: Trasversing Pickup Dropoff. *WPC*: Waiting for Pedestrian to Cross.

CL Type	Backbone	Decoder	Normal									Hard					Avg
			BLV	ST	SL	HMS	NMV	LMS	CL	FLL	STLT	HLA	LT	RT	TPD	WPC	
Nonreactive	w/o DCS	LSTM	90.0	**98.1**	92.1	88.9	86.8	82.3	79.0	84.6	77.1	65.2	46.9	40.3	54.1	60.1	73.1
	w/o DCS	LSTM-IRF	85.7	97.3	93.0	89.7	86.6	82.8	75.7	83.0	**91.1**	69.6	72.1	72.4	58.9	65.6	80.2
	Ref-DCS	LSTM-IRF	91.1	96.6	**95.4**	90.1	**88.7**	79.6	81.0	93.9	86.3	67.9	60.2	71.5	**66.9**	66.8	79.9
	GT-DCS	LSTM-IRF	**92.9**	97.0	93.3	**91.4**	87.9	**85.2**	**88.3**	**96.5**	90.3	**74.9**	**70.6**	**77.0**	65.5	**67.3**	**83.0**
Reactive	w/o DCS	LSTM	90.1	**96.3**	87.3	83.0	76.4	79.4	79.2	72.9	66.5	62.0	40.8	36.2	50.6	55.4	70.4
	w/o DCS	LSTM-IRF	85.7	95.3	88.4	84.5	79.0	73.3	75.1	71.1	78.1	65.7	**61.4**	70.0	63.3	66.1	75.4
	Ref-DCS	LSTM-IRF	91.4	93.8	93.4	**88.3**	82.1	75.2	79.7	82.2	74.1	63.2	49.9	64.3	62.4	63.5	74.6
	GT-DCS	LSTM-IRF	**92.8**	91.8	**93.9**	86.7	**82.2**	**80.0**	**81.0**	**86.2**	**81.8**	**73.5**	60.6	**73.0**	**64.6**	**69.7**	**78.7**

TABLE II: Driving scores of learning methods on **Val14** between NR-CL and R-CL and latency of neural network. † denotes the results obtained from [32] and * is based on implementation of nuPlan codebase. Our methods employ LSTM-IRF as decoders.

Input	Method	NR-CL	R-CL	Lat.(ms)
vector	GC-PGP† [35]	57	54	-
raster	PlanCNN† [4]	73	72	-
point	Urban Driver* [1]	47.5	46	45
point	PlanTF [36]	**84.83**	76.72	60
vector	w/o DCS	80.2	75.4	50
vector	Ref-DCS	79.9	74.6	52
vector	GT-DCS w/o DCF	81.9	76.6	50
vector	GT-DCS(ours)	83.0	**78.7**	52

of the driving score, [37] is recommended for more details. Several metric components are used for illustration. *No-Prog.* refers to the ratio of scenarios where the planner does not make progress, whereas expert demonstration does. *Col.* refers to the collision rate and *Off-Rd.* refers to the rate of violating drivable areas. In the NR-CL test, the ego vehicle may cheat by following surrounding vehicles like in Fig.1b, which are themselves following the ego vehicle in the logs. Since it will not happen in the R-CL test, as a manifestation of causal confusion, the Cheating by Following Log (*CFL*) rate is measured by *the difference of No-Prog.* between R-CL and NR-CL tests.

B. Implementation Details

The maximum number of adjacent agents is 10, the map collection radius is 50 meters, and the route fusion radius is 20 meters. A Cosine Annealing scheduler is utilized, where the initial learning rate is 5×10^{-4}, $\eta_{min} = 5 \times 10^{-6}$ and $T_{max} = 24$. The optimizer is AdamaW, and the weight decay is 5×10^{-4}. The model is trained over 24 epochs with a maximum of 12,000 learning steps per epoch with loss weights $\alpha_1 = \alpha_3 = 1$, $\alpha_2 = 0.5$, and the batch size is 64. The model training and simulation are carried out on a workstation with an i7-13700k and GeForce RTX 3090Ti.

The encoder and decoder embedding size is 128, and the layer number of each attention module is 2, with eight heads and a 0.1 dropout rate. During the training phase, 1,500,000 scenarios from the training and validation sets are randomly sampled. The samples are collected at least one second apart to guarantee that the training samples are diverse. Our model leverages 2s of the past to plan for 4s in the future with 2 Hz of history and 5 Hz of the future.

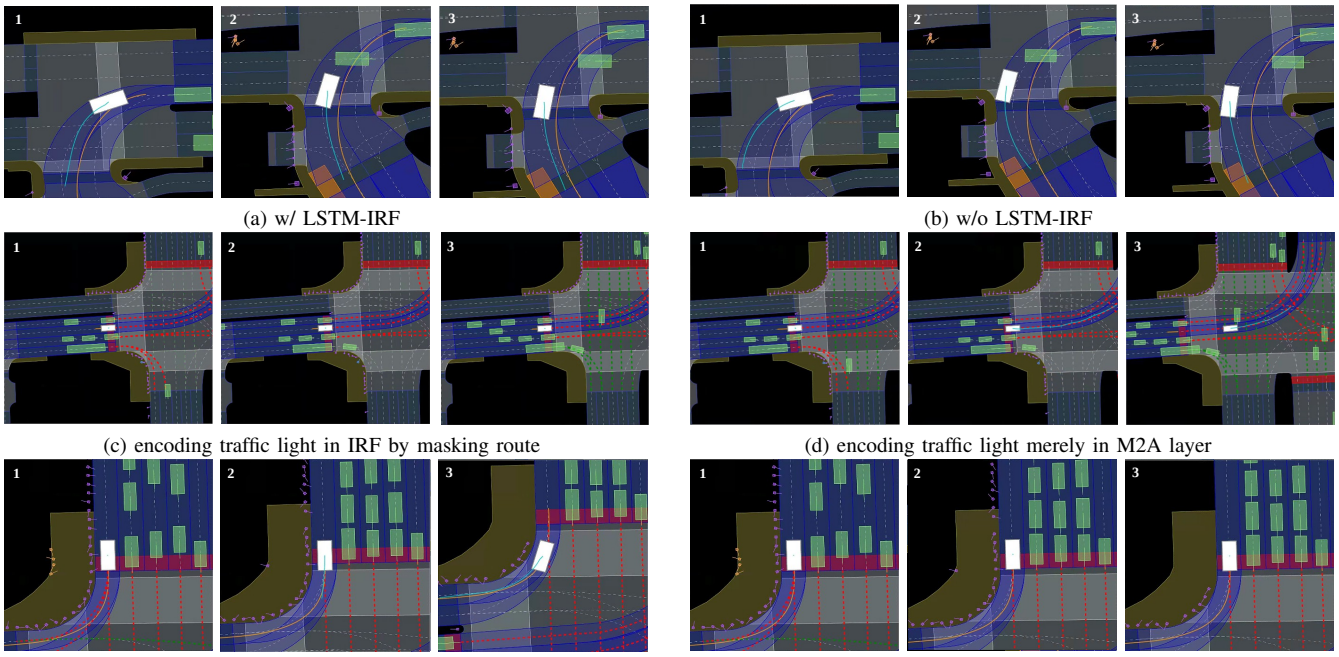
C. Simulation Results

The Val14 benchmark results are shown in Tab.II, which compares our methods with other representative methods in terms of NR-CL and R-CL scores. Only learning-based methods are compared without any explicit emergency braking or initiation of driving. Urban Driver [1] is a learning-based planner using point-based representations encoded by PointNet and an attention mechanism for global reasoning. PlanCNN [4] uses CNNs to encode a rasterized BEV scenario for waypoint planning. GC-PGP [35] employs a lane graph framework to generate plans conditioned on target route points. PlanTF [36] incorporates an RL control adapter that is trained jointly with the transformer-based trajectory planner.

The proposed method demonstrates superior performance compared to the learning-based baselines on Val14, especially on R-CL tests. Specifically, using ground truth (GT) as the de-confounding supervision (DCS) signal yields the highest driving scores on R-CL tests. In contrast to PlanTF, which solely addresses historical confounders, our method exhibits less disparity in performance between R-CL and NR-CL. If the *deleted edges* of surrounding vehicles are reinstated (GT-DCS w/o DCF), performance gains are less substantial than on de-confounding features, which implies the necessity of removing surrounding vehicle states when performing DCS. Note that all these experiments are conducted with the consideration of removing confounders of ego histories.

D. Ablation Study

1) The Effectiveness of DCS: Tab.I illustrates that by appropriate superving signals, DCS has the potential to enhance model performance across most categories of scenarios. In the NR-CL tests, the GT-DCS achieves the maximum score for 9 different scene types, while the Ref-DCS obtains a score of 3, resulting in a total of 12 distinct scene types. In the R-CL tests, there are similar results. The selection of the DCS label carries considerable significance, and Ref-DCS does not yield improvement comparable to the GT-DCS.



(a) w/ LSTM-IRF (b) w/o LSTM-IRF

(c) encoding traffic light in IRF by masking route (d) encoding traffic light merely in M2A layer

(e) w/ GT-DCS (f) w/ Ref-DCS

Fig. 4: Case studies of NR-CL simulation tests. Fig.4a and Fig.4b illustrate one left-turn scenario where the absence of LSTM-IRF leads to scraping the curb. Fig.4c and Fig.4d depict a traffic intersection, where with LSTM-IRF, the ego car keeps stationary when the traffic light is red, showing less causal confusion from nearby vehicles. Fig.4e and Fig.4f illustrate a scenario involving a right turn. With DCS, the ego car successfully transitions from a stationary position to a state of motion next to static vehicles, as opposed to with Ref-DCS.

TABLE III: Ablation study of DCS.

Backbone	NR-CL		R-CL		CFL
	No-Prog.	Col.	No-Prog.	Col.	
w/o DCS	4.56%	8.69%	7.88%	10.3%	3.32%
Ref-DCS	5.99%	**6.80%**	9.48%	**9.75%**	3.49%
GT-DCS	**2.77%**	7.97%	**5.02%**	10.1%	**2.25%**

TABLE IV: Ablation study of IRF in turning scenarios.

Decoder	NR-CL		R-CL	
	No-Prog.	Off-Rd.	No-Prog.	Off-Rd.
LSTM w/o IRF	22.39%	19.27%	29.68%	24.55%
LSTM w/ IRF	**8.63%**	**6.09%**	**16.7%**	**7.10%**

As shown in Tab.III, GT-DCS has been found to significantly decrease No-Prog. rate both in NR-CL and R-CL tests while not leading to increased Collision rate. This suggests that utilizing the model with GT-DCS can facilitate safer and more driving progress. Additionally, the rate of CFL has also experienced a decrease, which implies that GT-DCS can help in starting without leading vehicles. The aforementioned findings indicate that GT-DCS has the ability to strengthen the attention for driving conditions rather than the behaviors of surrounding vehicles. On the contrary, the model with Ref-DCS is more conservative and has a higher CFL rate, waiting until following. Ref-DCS demonstrates a significantly lower collision rate, potentially related to its significantly higher No-Prog rate. This observation might also be implicitly correlated with the fact that Ref is typically targeted at a lower velocity compared to expert demonstrations. Fig.4e depicts a successful example of model with GT-DCS starting with static surrounding vehicles, as opposite to Ref-DCS.

2) The Effectiveness of IRF: IRF is advantageous in enhancing the acquisition of curvature of lane centerlines for the planning decoder. As shown in Fig.4a and Fig.4b, the use of IRF facilitates accurate trajectory planning on curved routes and leads to decreased accumulating errors and safe driving. In Tab.I, compared with LSTM, LSTM-IRF greatly improves the performance in unprotected turning scenarios in Val14. In Tab.IV, the equipped IRF mitigates off-road problems as well as improves driving efficiency. The use of IRF can effectively address the issue of cause confusion depicted in Fig.1a since IRF provides a more explicit encoding of the traffic light states. In Fig.4c, the impact of red light surpasses the influence exerted by adjacent moving vehicles since the masking route strategy better highlights the undrivable areas than implicit one-hot encoding.

V. CONCLUSIONS

Our work elucidates the presence and detrimental effects of causal confusion arising in the context of vector-based behavior cloning. It proposes an off-policy DCS approach and novel context encoding that place the model's attention on driving intents within environments, rather than only imitating nearby cars. Numerous simulations conducted on nuPlan provide evidence of efficacy of our methods in addressing causal confusion for safer autonomous driving.

REFERENCES

[1] O. Scheel, L. Bergamini, M. Wolczyk, B. Osiński, and P. Ondruska, "Urban driver: Learning to drive from real-world demonstrations using policy gradients," in *Conference on Robot Learning (CoRL)*. PMLR, 2022, pp. 718–728.

[2] Y. Hu, J. Yang, L. Chen, K. Li, C. Sima, X. Zhu, S. Chai, S. Du, T. Lin, W. Wang, *et al.*, "Planning-oriented autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023, pp. 17 853–17 862.

[3] M. Bansal, A. Krizhevsky, and A. Ogale, "Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst," *arXiv preprint arXiv:1812.03079*, 2018.

[4] K. Renz, K. Chitta, O.-B. Mercea, A. S. Koepke, Z. Akata, and A. Geiger, "Plant: Explainable planning transformers via object-level representations," in *Conference on Robot Learning (CoRL)*. PMLR, 2023, pp. 459–470.

[5] P. Karkus, B. Ivanovic, S. Mannor, and M. Pavone, "Diffstack: A differentiable and modular control stack for autonomous vehicles," in *Conference on Robot Learning (CoRL)*. PMLR, 2023, pp. 2170–2180.

[6] 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 (TNNLS)*, 2023.

[7] C. Wen, J. Lin, T. Darrell, D. Jayaraman, and Y. Gao, "Fighting copycat agents in behavioral cloning from observation histories," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33, pp. 2564–2575, 2020.

[8] Z. Zhou, L. Ye, J. Wang, K. Wu, and K. Lu, "Hivt: Hierarchical vector transformer for multi-agent motion prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 8823–8833.

[9] C. Wen, J. Qian, J. Lin, J. Teng, D. Jayaraman, and Y. Gao, "Fighting fire with fire: Avoiding dnn shortcuts through priming," in *International Conference on Machine Learning (ICML)*. PMLR, 2022, pp. 23 723–23 750.

[10] C.-C. Chuang, D. Yang, C. Wen, and Y. Gao, "Resolving copycat problems in visual imitation learning via residual action prediction," in *European Conference on Computer Vision (ECCV)*. Springer, 2022, pp. 392–409.

[11] C. Wen, J. Lin, J. Qian, Y. Gao, and D. Jayaraman, "Keyframe-focused visual imitation learning," in *International Conference on Machine Learning (ICML)*. PMLR, 2021, pp. 11 123–11 133.

[12] F. Codevilla, E. Santana, A. M. López, and A. Gaidon, "Exploring the limitations of behavior cloning for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 9329–9338.

[13] 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," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, pp. 6119–6132, 2022.

[14] K. Chitta, A. Prakash, B. Jaeger, Z. Yu, K. Renz, and A. Geiger, "Transfuser: Imitation with transformer-based sensor fusion for autonomous driving," *IEEE Transactions on Pattern Analysis & Machine Intelligence (TPAMI)*, no. 01, pp. 1–18, 2022.

[15] 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 (CVPR)*, 2020, pp. 11 763–11 773.

[16] D. Chen, B. Zhou, V. Koltun, and P. Krähenbühl, "Learning by cheating," in *Conference on Robot Learning (CoRL)*. PMLR, 2020, pp. 66–75.

[17] J. Hawke, R. Shen, C. Gurau, S. Sharma, D. Reda, N. Nikolov, P. Mazur, S. Micklethwaite, N. Griffiths, A. Shah, *et al.*, "Urban driving with conditional imitation learning," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 251–257.

[18] D. Chen and P. Krähenbühl, "Learning from all vehicles," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022, pp. 17 222–17 231.

[19] L. Chen, L. Platinsky, S. Speichert, B. Osiński, O. Scheel, Y. Ye, H. Grimmett, L. Del Pero, and P. Ondruska, "What data do we need for training an av motion planner?" in *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2021, pp. 1066–1072.

[20] 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.

[21] T. Wang and D. E. Chang, "Improved reinforcement learning through imitation learning pretraining towards image-based autonomous driving," in *2019 19th international conference on control, automation and systems (ICCAS)*. IEEE, 2019, pp. 1306–1310.

[22] H. Liu, Z. Huang, J. Wu, and C. Lv, "Improved deep reinforcement learning with expert demonstrations for urban autonomous driving," in *2022 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2022, pp. 921–928.

[23] F. Torabi, G. Warnell, and P. Stone, "Behavioral cloning from observation," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, ser. IJCAI'18. AAAI Press, 2018, p. 4950–4957.

[24] J. Gu, C. Sun, and H. Zhao, "Densetnt: End-to-end trajectory prediction from dense goal sets," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 15 303–15 312.

[25] Z. Huang, X. Mo, and C. Lv, "Multi-modal motion prediction with transformer-based neural network for autonomous driving," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2605–2611.

[26] Y. Hu, K. Li, P. Liang, J. Qian, Z. Yang, H. Zhang, W. Shao, Z. Ding, W. Xu, and Q. Liu, "Imitation with spatial-temporal heatmap: 2nd place solution for nuplan challenge," *arXiv preprint arXiv:2306.15700*, 2023.

[27] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid, "Vectornet: Encoding hd maps and agent dynamics from vectorized representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 11 525–11 533.

[28] P. De Haan, D. Jayaraman, and S. Levine, "Causal confusion in imitation learning," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 32, 2019.

[29] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence (NMI)*, vol. 2, no. 11, pp. 665–673, 2020.

[30] L. Chen, P. Wu, K. Chitta, B. Jaeger, A. Geiger, and H. Li, "End-to-end autonomous driving: Challenges and frontiers," *arXiv preprint arXiv:2306.16927*, 2023.

[31] 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 (AISTATS)*. JMLR Workshop and Conference Proceedings, 2011, pp. 627–635.

[32] D. Dauner, M. Hallgarten, A. Geiger, and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," *arXiv preprint arXiv:2306.07962*, 2023.

[33] J. C. Butcher, *Numerical Methods for Ordinary Differential Equations*, 3rd ed. Hoboken, New Jersey: John Wiley & Sons, 2016. [Online]. Available: https://onlinelibrary.wiley.com/doi/book/10.1002/9781119121534

[34] A. Wächter and L. T. Biegler, "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming," *Mathematical programming*, vol. 106, pp. 25–57, 2006.

[35] M. Hallgarten, M. Stoll, and A. Zell, "From prediction to planning with goal conditioned lane graph traversals," *arXiv preprint arXiv:2302.07753*, 2023.

[36] J. Cheng, Y. Chen, X. Mei, B. Yang, B. Li, and M. Liu, "Rethinking imitation-based planner for autonomous driving," *arXiv preprint arXiv:2309.10443*, 2023.

[37] H. Caesar, J. Kabzan, K. S. Tan, W. K. Fong, E. Wolff, A. Lang, L. Fletcher, O. Beijbom, and S. Omari, "nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles," *arXiv preprint arXiv:2106.11810*, 2021.