2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

TOFG: A Unified and Fine-Grained Environment Representation in Autonomous Driving

Zihao Wen[1*], Yifan Zhang[1*], Xinhong Chen[1], and Jianping Wang[1]

Abstract— **In autonomous driving, an accurate understanding of environment, e.g., the vehicle-to-vehicle and vehicle-to-lane interactions, plays a critical role in many driving tasks such as trajectory prediction and motion planning. Environment information comes from high-definition (HD) map and historical trajectories of vehicles. Due to the heterogeneity of the map data and trajectory data, many data-driven models for trajectory prediction and motion planning extract vehicle-to-vehicle and vehicle-to-lane interactions in a separate and sequential manner. However, such a manner may capture biased interpretation of interactions, causing lower prediction and planning accuracy. Moreover, separate extraction leads to a complicated model structure and hence the overall efficiency and scalability are sacrificed. To address the above issues, we propose an environment representation, Temporal Occupancy Flow Graph (TOFG). Specifically, the occupancy flow-based representation unifies the map information and vehicle trajectories into a homogeneous data format and enables a consistent prediction. The temporal dependencies among vehicles can help capture the change of occupancy flow timely to further promote model performance. To demonstrate that TOFG is capable of simplifying the model architecture, we incorporate TOFG with a simple graph attention (GAT) based neural network and propose TOFG-GAT, which can be used for both trajectory prediction and motion planning. Experiment results show that TOFG-GAT achieves better or competitive performance than all the SOTA baselines with less training time.**

I. Introduction

Autonomous driving has gained rapid development in recent years. In a typical autonomous driving system, there are three indispensable modules [1], [2], i.e., perception, planning, and control modules. The planning module contains two major tasks, namely, trajectory prediction of surrounding vehicles and motion planning of the ego vehicle. Both trajectory prediction and motion planning take the HD map and historical trajectories of vehicles as the environment inputs. Such two types of data are necessary to capture interactions among vehicles and lanes, which largely determine the performance of the aforementioned tasks [3]–[5].

Given the great capability of capturing relations among agents, Graph Neural Network (GNN) [6] based models have been widely used for trajectory prediction [7]–[10] and motion planning tasks [11]–[13]. Most of the GNN-based models adopt attention mechanisms [14] to capture the interactions among agents, which helps improve the model performance. There are two main types of interaction that are commonly considered in the literature, namely, vehicle-to-vehicle interactions [15]–[18] and vehicle-to-lane interactions [19]–[21]. These two types of interactions are extracted from the sequential historical trajectories and the graph structured HD map. In the literature, different attention layers are designed manually to represent different types of interactions [13], [22]. The manually designed attention order deems to have some impact on the output of the model since different directions and orders of data flow lead to different results. Such approaches create the following issues.

- Firstly, it is hard to design a one-size-fits-all attention order manually, which can work well in all driving scenarios.
- Secondly, designing dedicated attention layers for every type of interactions results in a complex network architecture, which decreases the computation efficiency and sacrifices the scalability of the model.
- Thirdly, due to the already complex attention mechanism design, most current GNN models adopt a coarse-grained map structure to simplify the graph for easy training. However, such a manner may lead to the loss of attention in some regions and further cause the decrease of the overall model performance.

To address the above issues, both vehicle-to-vehicle and vehicle-to-lane interactions need to be captured with fine-grained map information simultaneously. To this end, a unified and fine-grained representation that expresses the HD map and the trajectory of surrounding vehicles in an isomorphic way is urged to be explored.

The recently proposed Occupancy Flow Field (OFF) [23] is a potential solution that has been utilized to provide a unified representation of the environment for predicting traffic from a macroscopic view. OFF, however, neglects the microscopic interaction and trajectory of a single vehicle. In this paper, we adopt and adapt OFF to construct a **Temporal Occupancy Flow Graph (TOFG)** representation that unifies the HD map information and historical trajectories of vehicles in an isomorphic data format. Specifically, the information of lanes, vehicles, and other road information are represented as nodes in the TOFG. Spatial and temporal edges are then created among them to simultaneously capture the vehicle-to-vehicle and vehicle-to-lane interactions in a consistent manner for more accurate prediction results. Such a unified representation can simplify the GNN structure and further help to improve computation efficiency and scalabil-

*The authors contributed equally to this work.

[1]Zihao Wen, Yifan Zhang, Xinhong Chen, and Jianping Wang are with Department of Computer Science, City University of Hong Kong, Hong Kong, China, and City University of Hong Kong Shenzhen Research Institute, Shenzhen, China (Email: zihaowen2-c@my.cityu.edu.hk, {yif.zhang, xinhong.chen, jianwang}@cityu.edu.hk).

This work was partially supported by Hong Kong Research Grant Council under GRF 11200220, Science and Technology Innovation Committee Foundation of Shenzhen under Grant No. JCYJ20200109143223052.

ity. Moreover, the simplification of TOFG also leaves room to enable a fine-grained map structure. Our main contributions are summarized as follows:

- We propose TOFG to completely characterize the heterogeneous environment in a unified way and simultaneously capture vehicle-to-vehicle and vehicle-to-lane interactions. Both the historical trajectories of surrounding vehicles and the HD maps are encoded with fine-grained representations to include the temporal and spatial environment information effectively and efficiently.
- We apply a simple Graph Attention (GAT) based neural network with TOFG, referred to as **TOFG-GAT**, to showcase that our proposed TOFG is capable of simplifying the model architecture without sacrificing performance. Specifically, compared with several SOTA models, the extensive experiment results show that TOFG-GAT can achieve better performance in both trajectory prediction and motion planning with **68.57%** fewer parameters and **33.69%** less training time.
- We visualize the attention map of TOFG-GAT and other baselines. The results show that TOFG-GAT produces a more effective and rational driving interaction logic of the model.

The rest of the paper is organized as follows. In Section II, we use an example to motivate the necessity of unified representation and then present the architecture of our proposed TOFG for environment representation. We then elaborate on the details of the TOFG-GAT for trajectory prediction and motion planning tasks in Section III. We conduct extensive experiments in Section IV to demonstrate the superiority of our proposed TOFG in representing environments. Finally, we conclude this paper in Section V.

II. Temporal Occupancy Flow Graph

In this section, we present our unified environment representation TOFG. To begin with, we use a motivation example to illustrate the limitations of separately extracting vehicle-to-vehicle and vehicle-to-lane interactions. Then, we introduce how a TOFG can be constructed based on the input of HD maps and historical vehicle trajectories in details.

A. Motivation Example

To motivate the necessity of unified representation, we take LaneGCN [13] as an example, which uses three GATs, namely Actor-to-Lane (A2L), Lane-to-Actor (L2A), and Actor-to-Actor (A2A), to capture the interactions among vehicles and lanes. Specifically, A2L aggregates vehicle information to lanes, L2A passes fused traffic information to vehicles, and A2A handles the interaction between vehicles. We use a simple overtaking driving scenario to demonstrate that the order of attention layers does have impacts on capturing the interaction and the prediction accuracy, as shown in Figure 1.

In Figure 1, the red ego vehicle is overtaking its blue neighbor vehicle, and L is a lane node representing the lane segment in front of the neighbor vehicle. Consider two different sequential orders of LaneGCN's attention layers:

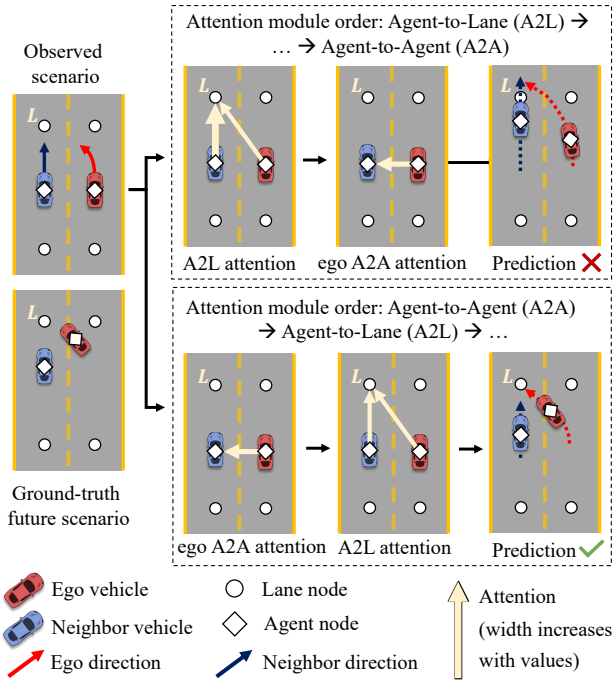


Fig. 1. A failure case of LaneGCN. Manually designed attention order could not handle vehicle-to-vehicle and vehicle-to-lane interactions comprehensively.

the first one is <A2L, L2A, A2A>, which is the same as the order in [13], and the second one is <A2A, A2L, L2A>. In the first case, A2L is computed first, so the neighbor vehicle has a higher probability of appearing at L since it is closer. Then, L2A passes such information of L to two vehicles. In the following A2A, the red vehicle may abort overtaking since it has received the information about the blue vehicle from L. Thus, in A2A, the red vehicle node sends weak lane overtaking intention to the blue vehicle node, leading to wrong prediction results. In the second case, A2A is computed first, and thus only the current information between vehicles is exchanged. When A2L is performed, the red vehicle will have no information that the blue vehicle may occupy node L. Hence, the red vehicle may still have a higher probability of appearing at node L. In the following L2A, the blue vehicle receives the information that the red vehicle will occupy L, and thus it may give the road. Based on the above two cases, either order of attention modules could not handle the scenario comprehensively since they do not compute vehicle interaction and lane occupation simultaneously. Generally speaking, a fixed attention order brings biased understanding of the environment, leading to a decrease of prediction accuracy. To verify the existence of above issues, we conduct experiments to compare the prediction accuracy of the aforementioned two LaneGCN models with different attention layer orders. Please refer to Section IV-A for details.

To tackle the above issues, we propose Occupancy Flow Graph (OFG) and Temporal Occupancy Flow Graph (TOFG), two vectorized representations combining HD maps and vehicle trajectory information, which enables capturing more accurate interactions simultaneously.

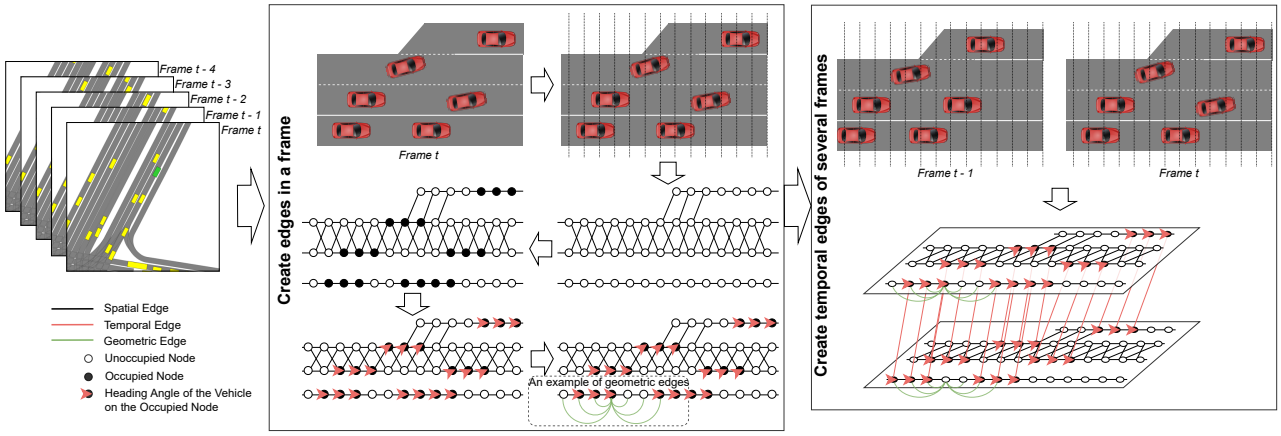


Fig. 2. TOFG construction process. (1) Extract lane centerlines from raw map data and cut the lane centerlines into fine-grained lane segments. (2) Build lane graph based on extracted lane segments and their connections. (3) Build OFG for frame t. Compute occupancy flows given vehicles' trajectories and construct vehicle interaction edges and multi-scale geometric edges. (4) Connect OFGs in consecutive frames with temporal edges.

B. Occupancy Flow Graph (OFG)

Given the lane information provided by HD maps, an OFG is constructed based on the lane graph structure. Current HD maps are built based on the geometry of lanes and lane connectors, i.e., ramps or crossroads. We first extract the geometry of the center line for each lane and connections among lanes, which are usually represented by polylines. Each center line can be represented as a sequence of lane segments, denoted as $L = (l_1, l_2, ...l_{N_{\text{seg}}})$ with N_{seg} being the number of segments. Treating each lane segment as a graph node, all these lane segments form a lane graph. The lane segments are short for defining the detailed and fine curves or polygons of lanes. These short lane segments serving as nodes allow the OFG to capture a more fine-grained map structure. To trade-off between the precision and complexity of the OFG, we set the average length of these lane segments as 0.3 meters.

We use the above lane segments and lane graph to represent the environment and model the obstacles in it. We expand each lane segment $l = [(x_1, y_1) \rightarrow (x_2, y_2)]$ to a rectangle R_l according to the lane width. A lane segment is occupied if its expanded rectangle R_l intersects with the bounding box of a vehicle. If more than one vehicle's bounding box intersects with R_l, then the one closest to or containing the centroid $(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2})$ of R_l is considered to be the occupant.

Inspired by [23] which proposes OFF to extract the motion and future movement of occupant vehicles, we further extend OFF to graphs to effectively extract interactions and integrate more informative features. With the lane graph structure described above, we could build an OFG as $G_{\text{OF}} = \{\mathcal{V}, E\}$, where $\mathcal{V} = \{u_1, u_2, ..., u_n\}$ is the set of n nodes, and E is the set of edges. For clearer distinction, we consider the following features to construct the node in the graph.

Lane segment features: Each lane segment l starting from (x_1, y_1) to (x_2, y_2), is represented by its midpoint coordinates $(\frac{x_1+x_2}{2}, \frac{y_1+y_2}{2})$ and its vector form $(x_2 - x_1, y_2 - y_1)$.

Occupant vehicle features: Each vehicle that occupies lane segments includes an occupancy value $O \in \{0, 1\}$ and an occupancy flow vector $(-v_x, -v_y, \theta, \omega)$, where $v = (v_x, v_y)$ is the velocity of this vehicle, θ and ω are the yaw and yaw rate of the occupant vehicle, respectively. Here we follow the backward flow representation in [23] and use negative velocity.

The above features are used to form the node of the graph. The edges that connect the nodes are defined as follows.

Geometric edges: Two adjacent lane segments are connected using an edge following the geometric edges of the lane graph extracted from the HD map.

Multi-scale geometric edges: In order to capture long range dependencies, we extend normal geometric edges and propose the multi-scale geometric edges. They are a union of all n-scale geometric edges, where $n = \{1, 2, ..., N_{\text{scale}}\}$, and an n-scale geometric edge connects the current node with the node n hops away. In this paper, we choose $N_{\text{scale}} = 4$.

Vehicle interaction edges: We design vehicle interaction edges to integrate the vehicle-to-vehicle interactions in our OFG. We assume that there would be interactions between every two vehicles when the distance between them is less than a threshold, which is empirically set to 100 meters in this paper. For two interacting vehicles, the vehicle interaction edges connect the nodes between the bounding boxes of the two vehicles. Specifically, if vehicle A occupies a set of nodes $\mathcal{V}_A = \{u_1, u_2, \cdots, u_m\}$ and vehicle B occupies $\mathcal{V}_B = \{u'_1, u'_2, \cdots, u'_n\}$, where $m \leq n$, the set of the vehicle interaction edges E_{AB} are the injection from $\mathcal{V}_A$ to $\mathcal{V}_B$ without loss of generality, namely $E_{AB} = \{(u_1, u'_1), (u_2, u'_2), ..., (u_m, u'_m)\}$.

The creation of an OFG given a frame of the data is illustrated in the first part of Figure 2.

C. Temporal Occupancy Flow Graph (TOFG)

Existing literature has proved that the historical trajectories of a moving object contribute to its future action prediction [24]. Moreover, human drivers, even autonomous vehicles, have reaction time due to the time lag of perception and information processing so that a driving behavior may be resulted from some events seconds ago. Therefore, a single OFG cannot effectively handle such hysteretic time dependencies between perceived environment and observed driving decisions. To this end, we further propose Temporal

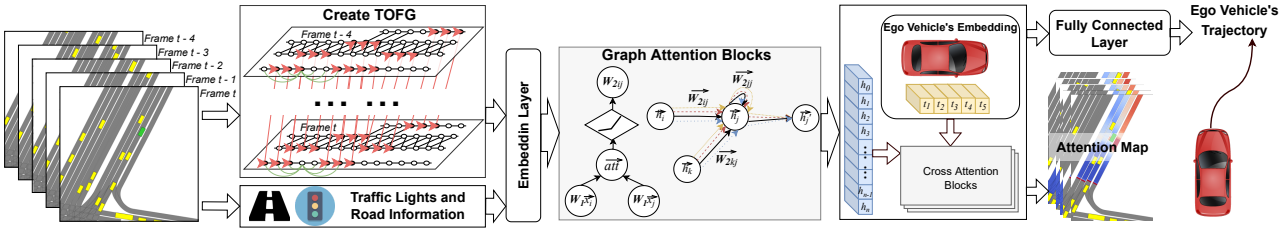


Fig. 3. Architecture of TOFG-GAT.

Occupancy Flow Graph (TOFG) G_{TOF}, the combination of multiple OFGs $G_{\mathrm{OF}}^{(t)}$ in a given time horizon.

The **temporal edges** are introduced to connect the nodes between two OFGs in consecutive frames so that the temporal information could be propagated. For a vehicle which occupies $\mathcal{V}_t = \{u_1^{(t)}, u_2^{(t)}, ..., u_m^{(t)}\}$ at time t, if it exists in the previous OFG at time $t-1$ and occupies $\mathcal{V}_{t-1} = \{u_1^{(t-1)}, u_2^{(t-1)}, ..., u_n^{(t-1)}\}$, we conduct the following two operations to construct the temporal edges. First, we transform the coordinates of the occupied nodes to relative coordinates using the relative frame of the vehicle. Then, temporal edges are built to connect all occupied nodes at time t to their closest nodes at time $t-1$ in the relative frame in a similar way to vehicle interaction edges. The creation of TOFG from several consecutive OFGs is shown in the right part of Figure 2.

III. GAT Model with TOFG

Our proposed TOFG can be applied in many tasks, among which two representatives are motion planning and trajectory prediction. By unifying the map information and vehicles' trajectories into a homogeneous graph, TOFG enables the downstream models to encode consistent interaction among vehicles and lanes using a simplified model structure without sacrificing accuracy. To facilitate the subsequent demonstration of these merits, we propose **TOFG-GAT** for the downstream tasks and introduce the detailed model design in the rest of this section.

A. Model Design

Considering that the attention mechanism performs a great capability of capturing the relations among nodes, we apply a GAT-based model with our proposed TOFG, i.e., **TOFG-GAT** to predict the trajectory of the ego vehicle. As shown in Figure 3, our model consists of three components. The first component is TOFG construction based on trajectory and HD map inputs. The second component is a GAT network. To enrich semantic information, we embed some other road information with TOFG, which includes the following features: (1) the traffic light status, which could be one of {red, yellow, green, none}, and (2) the on-route status of lane segments, which could be "on route" or "off route". Given the above inputs, the GAT network is responsible for extracting spatial and temporal interaction features from the environment. Note that the graph attention layers adopted here are slightly different from the one in [6]. Given a node i in TOFG, the features from its adjacent nodes j are aggregated as Equation 1.

$$h_i' = h_i + \sum_j \phi((h_i||h_j)W_1)W_2 \tag{1}$$

where h_i is the embedding of node u_i, W_1, W_2 are trainable weight matrices, ϕ is the combination of layer normalization and ReLU, symbol $||$ is concatenation.

The third component is a cross-attention model to calculate the ego vehicle's attention score to every node of TOFGs in order to plan its future trajectory. We only use one multi-head attention layer [14] with $N_{\mathrm{head}} = 4$ so that the attention scores can be summarized in one attention map, which can provide comprehensive information. The computing process of the multi-head cross-attention layer is defined in Equation 2.

$$\begin{aligned} &\mathrm{Attention}(Q,K,V) = \mathrm{softmax}(\frac{QK^T}{\sqrt{d_k}})V \\ &\mathrm{head}_i = \mathrm{Attention}(Q(h_{\mathrm{ego}}), K(h_{\mathrm{tofg}}), V(h_{\mathrm{tofg}})) \\ &y_{\mathrm{att}} = (\|_{i=1}^{N_{\mathrm{head}}} \mathrm{head}_i)W^{att} \end{aligned} \tag{2}$$

where h_{ego} and h_{tofg} are the embedding of the ego vehicle and the embedding matrix of the TOFG nodes, respectively; Q, K, and V are query, key, and value networks, respectively, and d_k is the output dimension of Q and K. and W^{att} is a trainable weight matrix. The output of the cross-attention module y_{att} is fed into a Muti-Layer Perceptron (MLP) layer to generate a discrete trajectory for several future time steps.

B. Training Loss

We follow the classic imitation learning setting [25] and use imitation loss as the loss function for our model. Denote $\overline{\pi} = \{(\overline{x}_1, \overline{y}_1), (\overline{x}_2, \overline{y}_2), \cdots, (\overline{x}_T, \overline{y}_T)\}$ as the trajectory planned by our model and $\pi = \{(x_1, y_1), (x_2, y_2), \cdots, (x_T, y_T)\}$ as the ground truth trajectory of the vehicle in the dataset. The imitation loss is defined as $L_{\mathrm{imitation}} = \sum_{t=1}^{T} \sqrt{(\overline{x}_t - x_t)^2 + (\overline{y}_t - y_t)^2}$.

IV. Experiment

To demonstrate the superiority of TOFG-GAT in the tasks of trajectory prediction and motion planning, we compare it with the following baseline models: LaneGCN [13], mmTransformer [8], and HiVT [7]. Note that we slightly change the output layer of the original HiVT to fit in our experiment settings. All the three baseline models are constructed based on GNNs while different attention mechanisms are adopted to capture the interactions among vehicles and lanes.

To train and evaluate both the proposed model and the baselines, NuPlan [26] mini dataset is utilized. NuPlan

provides 1500 hours of human driving data from 4 cities across the US and Asia with diverse traffic patterns, and the mini dataset is a teaser version of the full dataset and contains about 2.5% of all data (around 40 hours). We choose trajectory prediction and motion planning tasks to illustrate the performance of our TOFG-GAT. We randomly extract 5000 scenario samples from NuPlan mini dataset and split them into training (70%), validation (15%), and testing set (15%). Each of the models is trained on the training set for 60 epochs. We use Adam optimizer with a learning rate of 1×10^{-5} and set batch size as 3 due to the limitation of GPU memory. We select the checkpoint that performs the best in validation from training epochs for testing.

A. Trajectory Prediction

We perform a single-agent trajectory prediction task to illustrate our motivation example numerically and the superiority of TOFG-GAT. The problem of the single-agent trajectory prediction is to predict a 6-second future trajectory of ego vehicle $\hat{\pi}_0 = (\hat{s}_0^{t+1}, \hat{s}_0^{t+2}, ..., \hat{s}_0^{t+H})$, given the 1.5-second historical trajectory of ego and surrounding vehicles $\Pi = (\pi_0, \pi_1, ..., \pi_m)$ and map information $\mathcal{M}$, where $\pi_i = (s_i^{t-T}, s_i^{t-T+1}, ..., s_i^t)$ is the historical trajectory of vehicle i, $s_i^t = (x_i^t, y_i^t, \theta_i^t)$ is the state of vehicle i at time t, and H and T are the numbers of time steps in prediction horizon and past time horizon, respectively. Here, we set $H = 12$ and $T = 5$ following NuPlan's setting. We use the following four metrics to evaluate the performance: average displacement error (ADE), average heading error (AHE), final displacement error (FDE), and final heading error (FHE). They are formally defined as Equation 3.

$$\begin{aligned} \text{ADE} &= \frac{1}{H}\sum_{i=0}^{H} \|\hat{s}_0^{t+i} - s_0^{t+i}\|_2, \ \text{FDE} = \|\hat{s}_0^{t+H} - s_0^{t+H}\|_2 \\ \text{AHE} &= \frac{1}{H}\sum_{i=0}^{H} \left|\hat{\theta}_0^{t+i} - \theta_0^{t+i}\right|, \ \text{FHE} = \left|\hat{\theta}_0^{t+H} - \theta_0^{t+H}\right| \end{aligned} \tag{3}$$

First, we verify our motivation by training the aforementioned two LaneGCN models in Section II-A: the vanilla LaneGCN (attention layer order:<A2L, L2A, A2A>) and the modified LaneGCN (attention layer order: <A2A, A2L, L2A>). Since the lane change is a more complicated and challenging scenario, different from the conventional training procedures, we extract 5000 lane-changing scenarios from NuPlan mini dataset for a better illustration of the impact of attention order. The extracted scenarios are divided into training (70%), validation (15%), and testing set (15%). Both models are trained on the training set for 30 epochs and tested on the testing set. The result of the verification experiment is shown in Table I. We can see that except for AHE, the overall performance of the modified LaneGCN is better than the original one, verifying our motivation that there is no one-size-fits-all attention order.

Next, we compare the performance of TOFG-GAT and the baselines on the trajectory prediction task. The training setting of the prediction task follows the aforementioned conventional one in the beginning of this section. The result of the trajectory prediction task using our model and baselines are shown in Table II. It can be seen from the table that TOFG-GAT performs the best in ADE, AHE, and FHE. And our FDE is preceded only by mmTransformer. Specifically, our model has the smallest heading errors among baseline models, implying that our model performs better in terms of both position and heading orientation.

TABLE I

TRAJECTORY PREDICTION PERFORMANCE COMPARISON ACHIEVED BY TWO DIFFERENT LANEGCNS.

Metric	Vanilla LaneGCN	Modified LaneGCN
ADE (m)	2.348	**2.235**
AHE (rad)	**0.0851**	0.0889
FDE (m)	5.650	**5.231**
FHE (rad)	0.1353	**0.1324**

TABLE II

TRAJECTORY PREDICTION PERFORMANCE COMPARISON ACHIEVED BY TOFG-GAT AND BASELINES.

Metric	TOFG-GAT	LaneGCN	mmTrans	HiVT
ADE (m)	**1.818**	2.258	1.836	3.5571
AHE (rad)	**0.0463**	0.0721	0.1133	0.1014
FDE (m)	4.538	5.063	**4.210**	8.4658
FHE (rad)	**0.0818**	0.1094	0.1421	0.1340
Num parameters	**542K**	1.9M	1.6M	1.7M

B. Motion Planning

To compare our model with the baselines in the motion planning task, we adopt close loop simulation, which provides a more in-depth evaluation since compounding errors during simulation largely affect future observations and could significantly diverge from the ground truth. In the close loop simulation, each model is required to plan a trajectory $\hat{\pi}^t$ every k ms for controlling the ego vehicle given historical trajectory of ego and surrounding vehicles Π and map information $\mathcal{M}$ at time t. We assume that the ego vehicle can perfectly follow $\hat{\pi}^t$ and move to the corresponding position s^{t+k} on $\hat{\pi}$ at time step $t+k$. As for surrounding vehicles, we simply replay their driving behaviors from the dataset. For safety concerns, we add an extra auto-correction mechanism for ego vehicle to avoid collision or driving out of the road.

We randomly select 50 simulation scenarios from NuPlan mini dataset. Each simulation lasts for 20 seconds. For each scenario, there is a goal state $s_{\text{goal}} = (x_{\text{goal}}, y_{\text{goal}}, \theta_{\text{goal}})$, which is the true final state of ego vehicle. Given the trajectory $\pi_{\text{v}} = (\overline{s}^0, ..., \overline{s}^{T_s})$ of ego vehicle planned by the motion planning model and the ground truth one $\pi_{\text{expert}} = (s^0, ..., s^{T_s})$, where T_s is the duration time of the simulation. We use the following metrics to evaluate motion planning.

(1) L2 distance to expert trajectory: This metric evaluates the trajectory difference between expert trajectory and vehicle trajectory. We also include the heading error in this metric, with a weight factor $w_\theta = 2.5$. The metric could be formulated as $M_{L2} = \sum_{t=0}^{T_s}(\|(\overline{x}^t - x^t, \overline{y}^t - y^t)\|_2 + w_\theta \left|\hat{\theta}^t - \theta^t\right|)$.

(2) Distance to goal: This metric is the distance from each planned waypoint to the simulation goal, and a smaller value

TABLE III

MOTION PLANNING PERFORMANCE COMPARISON ACHIEVED BY TOFG-GAT AND BASELINES.

Metric		TOFG-GAT	LaneGCN	mmTrans	HiVT
M_{L2} (lower-the-better)	max	50.7751	53.7209	60.6808	**50.0980**
	mean	23.7583	27.3519	28.7099	**23.1862**
	max (yaw)	52.1662	55.6454	62.9381	**51.6376**
	mean (yaw)	24.7023	28.7969	30.3691	**24.1930**
$M_{\text{dist2goal}}$ (lower-the-better)	max	**215.0541**	217.4206	215.7068	215.2509
	min	**118.7215**	140.1732	190.6343	151.3336
	mean	**166.5199**	179.3244	202.9253	184.6144
$M_{\text{prog2goal}}$ (higher-the-better)	absolute	**89.0107**	66.3626	19.1848	59.1334
	relative	**0.8671**	0.8312	0.5415	0.7435
M_{prog2exp} (higher-the-better)	total	**81.0641**	66.1283	26.1031	58.4340
	ratio	**0.7844**	0.6887	0.3508	0.6121
Inference time (s)		**0.0122**	0.0250	0.0125	0.0164

is better. We take the maximum, minimum, and average of this metric for comparison, i.e., $M_{\text{dist2goal}} = \Phi(\{\|(\overline{x}^t - x_{\text{goal}}, \overline{y}^t - y_{\text{goal}})\|_2\}_t), t = 0, 1, ..., T_s$, where Φ is min, max, or avg.

(3) Distance progress to goal: This metric measures the progress of the ego vehicle toward the goal position. It is calculated using $M_{\text{prog2goal}} = \|(\overline{x}^0 - x_{\text{goal}}, \overline{y}^0 - y_{\text{goal}})\|_2 - \|(\overline{x}^{T_s} - x_{\text{goal}}, \overline{y}^{T_s} - y_{\text{goal}})\|_2$.

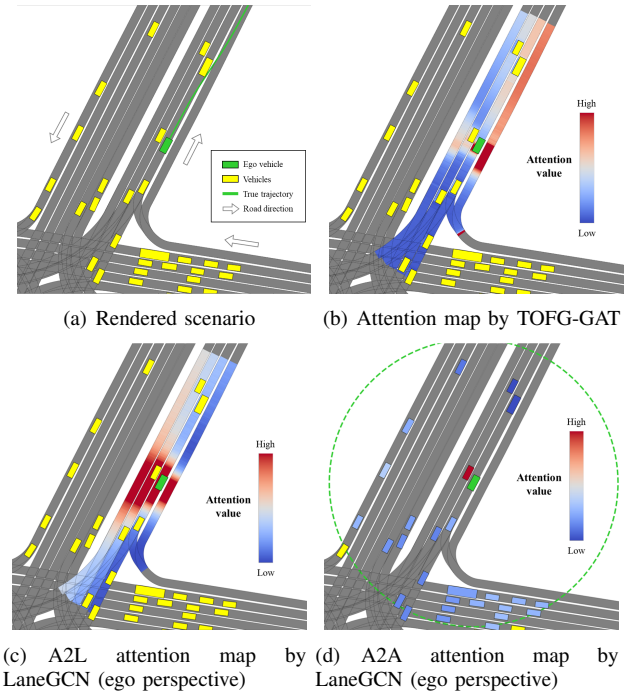
(4) Distance progress along expert route: This metric accumulates ego vehicle's progress distance along the expert route, denoted as $M_{prog2exp}$, and a higher value is better. Ego vehicle is on the expert route if it is on the same roadblock of expert trajectory or the skipped road block connected to the next roadblock of expert trajectory.

The result of motion planning is shown in Table III. TOFG-GAT has an overall best performance in terms of these four metrics. It is only slightly worse than HiVT in L2 distance. The reason behind this is that the planning accuracy of the first few frames of HiVT is slightly better than ours. The superiority of our model in the other three metrics demonstrates that TOFG-GAT has a better understanding of the environment and the intention of the ego vehicle since our representation can better capture inter-vehicle interactions and vehicle-lane interactions. Additionally, TOFG-GAT has the shortest inference time among the four models.

C. Attention Map Visualization and Analysis

Finally, we analyze the attention map extracted from TOFG-GAT and LaneGCN. Figure 4(a) shows a lane-changing scenario extracted from NuPlan dataset: ego vehicle marked with green is changing to the lane on its right. The last frame's attention map of the cross-attention layer from TOFG-GAT and the ego vehicle's attention maps from A2L and A2A module of LaneGCN are shown using heat maps in Figure 4(b), 4(c), and 4(d), respectively.

In Figure 4(b), the attention values on the target lane are higher than those on other lanes. Also, the attention values on the lane segments around neighbor vehicles, which



(a) Rendered scenario (b) Attention map by TOFG-GAT

(c) A2L attention map by LaneGCN (ego perspective) (d) A2A attention map by LaneGCN (ego perspective)

Fig. 4. Visualization of attention maps from TOFG-GAT and LaneGCN. The attentions of TOFG-GAT points out the target lane and places near the vehicles in interactions, while the attentions of LaneGCN are basically related to distance between ego and the attention subject.

are likely to have interactions with the ego vehicle, are higher than others vehicles. Such pattern allows our model to make a similar choice to the ground truth while keeping ego vehicle from colliding with others. The attention maps from LaneGCN are less informative than ours, as shown in Figure 4(c) and 4(d). The A2L attention is basically a monotonic increase function to distance between ego vehicle and the lane segment. Similarly, in A2A attention map, the attention value seems to be higher when the distance between the ego vehicle and the other vehicle become larger. Moreover, the vehicles interacting with the ego vehicle do not get larger attention value. The above case studies well demonstrates that our model can obtain more informative attention than baseline models.

V. CONCLUSION

In this paper, we proposed a unified environment representation TOFG to include both the map information and trajectory of vehicles in a homogeneous graph. The lane segments are partitioned with a smaller length to construct a graph with fine-grained information. Such a unified and fine-grained graph can contribute to simultaneously and accurately capturing the interactions so that the performance of downstream models can be further improved. As the complex attention layers are no longer required to tackle different interactions of heterogeneous data, a simplified model TOFG-GAT with fewer parameters can achieve a competitive and even better performance than that of SOTA models. In future work, we plan to analyze the attention map numerically to summarize a common interaction-aware guideline that can be applied to all types of models of autonomous driving tasks.

REFERENCES

[1] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE access*, vol. 8, pp. 58 443–58 469, 2020.

[2] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.

[3] L. Ye, Z. Wang, X. Chen, J. Wang, K. Wu, and K. Lu, "Gsan: Graph self-attention network for learning spatial–temporal interaction representation in autonomous driving," *IEEE Internet of Things Journal*, vol. 9, no. 12, pp. 9190–9204, 2022.

[4] G. Markkula, R. Madigan, D. Nathanael, E. Portouli, Y. M. Lee, A. Dietrich, J. Billington, A. Schieben, and N. Merat, "Defining interactions: a conceptual framework for understanding interactive behaviour in human and automated road traffic," *Theoretical Issues in Ergonomics Science*, vol. 21, no. 6, pp. 728–752, 2020.

[5] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 1549–15 498.

[6] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," *stat*, vol. 1050, p. 20, 2017.

[7] 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.

[8] Y. Liu, J. Zhang, L. Fang, Q. Jiang, and B. Zhou, "Multimodal motion prediction with stacked transformers," *Computer Vision and Pattern Recognition*, 2021.

[9] 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)*, October 2021, pp. 15 303–15 312.

[10] J. Ngiam, B. Caine, V. Vasudevan, Z. Zhang, H.-T. L. Chiang, J. Ling, R. Roelofs, A. Bewley, C. Liu, A. Venugopal, *et al.*, "Scene transformer: A unified architecture for predicting multiple agent trajectories," *arXiv preprint arXiv:2106.08417*, 2021.

[11] C. Yu and S. Gao, "Reducing collision checking for sampling-based motion planning using graph neural networks," in *Advances in Neural Information Processing Systems*, M. Ranzato, A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, Eds., vol. 34. Curran Associates, Inc., 2021, pp. 4274–4289.

[12] B. Ichter, E. Schmerling, T.-W. E. Lee, and A. Faust, "Learned critical probabilistic roadmaps for robotic motion planning," 2019.

[13] M. Liang, B. Yang, R. Hu, Y. Chen, R. Liao, S. Feng, and R. Urtasun, "Learning lane graph representations for motion forecasting," in *European Conference on Computer Vision*. Springer, 2020, pp. 541–556.

[14] 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.

[15] A. Kuefler, J. Morton, T. Wheeler, and M. Kochenderfer, "Imitating driver behavior with generative adversarial networks," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE Press, 2017, p. 204–211.

[16] N. Deo and M. M. Trivedi, "Multi-modal trajectory prediction of surrounding vehicles with maneuver based lstms," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE Press, 2018, p. 1179–1184.

[17] X. Li, X. Ying, and M. C. Chuah, "Grip: Graph-based interaction-aware trajectory prediction," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 3960–3966.

[18] T. Gilles, S. Sabatini, D. Tsishkou, B. Stanciulescu, and F. Moutarde, "Gohome: Graph-oriented heatmap output for future motion estimation," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 9107–9114.

[19] J. Mercat, T. Gilles, N. E. Zoghby, G. Sandou, D. Beauvois, and G. P. Gil, "Multi-head attention for multi-modal joint vehicle motion forecasting," 2019.

[20] 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)*, June 2020.

[21] M.-F. Chang, J. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[22] W. Schwarting, A. Pierson, J. Alonso-Mora, S. Karaman, and D. Rus, "Social behavior for autonomous vehicles," *Proceedings of the National Academy of Sciences*, vol. 116, no. 50, pp. 24 972–24 978, 2019.

[23] R. Mahjourian, J. Kim, Y. Chai, M. Tan, B. Sapp, and D. Anguelov, "Occupancy flow fields for motion forecasting in autonomous driving," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 5639–5646, 2022.

[24] É. Zablocki, H. Ben-Younes, P. Pérez, and M. Cord, "Explainability of vision-based autonomous driving systems: Review and challenges," *arXiv preprint arXiv:2101.05307*, 2021.

[25] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Computing Surveys (CSUR)*, vol. 50, no. 2, pp. 1–35, 2017.

[26] K. T. e. a. H. Caesar, J. Kabzan, "Nuplan: A closed-loop ml-based planning benchmark for autonomous vehicles," in *CVPR ADP3 workshop*, 2021.