SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2026, Vol. 69, Iss. 5, 152103:1–152103:12
https://doi.org/10.1007/s11432-025-4815-9

Graph-based topology reasoning for driving scenes

Tianyu LI[1,2,3†], Li CHEN[3†], Huijie WANG[2†], Yang LI[5], Jiazhi YANG[5], Xiangwei GENG[5], Hang XU[4], Chunjing XU[4], Junchi YAN[2], Ping LUO[3] & Hongyang LI[2,3*]

[1]*College of Computer Science and Artificial Intelligence, Fudan University, Shanghai 200438, China*
[2]*Shanghai Innovation Institute, Shanghai 200231, China*
[3]*School of Computing and Data Science, The University of Hong Kong, Hong Kong 999077, China*
[4]*Huawei, Shenzhen 518129, China*
[5]*Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China*

Received 18 February 2025/Revised 18 June 2025/Accepted 29 September 2025/Published online 23 April 2026

Abstract Understanding the road structure is essential for achieving autonomous driving. This intricate topic contains two fundamental components: the interconnections between lanes and the associations between lanes and traffic elements (e.g., traffic lights), where a comprehensive topology reasoning method is still absent. On one hand, existing map learning techniques face challenges in deriving lane connectivity using segmentation or laneline-based representations; or prior approaches focus on centerline detection while neglecting interaction modeling. On the other hand, the topic of assigning traffic elements to lanes is limited in the image domain, leaving the construction of the correspondence between image and 3D views an unexplored challenge. To address these issues, we present TopoNet, an end-to-end topology reasoning network for analyzing driving scenes. To capture the topology of driving environments effectively, we introduce three key designs: (1) an embedding module that integrates semantic knowledge from 2D elements into a unified feature space; (2) a curated scene graph neural network that models relationships and facilitates feature interactions within the network; (3) a scene knowledge graph devised to differentiate prior knowledge from various types of the scene topology avoiding arbitrary message transmission. We evaluate TopoNet on the challenging scene understanding benchmark, OpenLane-V2, where our approach outperforms all previous studies by a significant margin across all perceptual and topological metrics. The code is released at https://github.com/OpenDriveLab/TopoNet.

Keywords autonomous driving, topology reasoning, lane perception, traffic element recognition, graph

Citation Li T Y, Chen L, Wang H J, et al. Graph-based topology reasoning for driving scenes. Sci China Inf Sci, 2026, 69(5): 152103, https://doi.org/10.1007/s11432-025-4815-9

1 Introduction

Imagine an autonomous vehicle navigating towards a complex intersection and planning to go straight ahead: it faces uncertainty when choosing the appropriate lane to enter and determining which traffic signal to adhere to. This sophisticated challenge necessitates the agent to not only perceive lane positions, but also understand the topological relationships from sensor inputs. Specifically, the topology in driving scenes includes: (1) the lane topology graph comprising centerlines and their connectivity; (2) the assignment relationships between lanes and traffic elements such as traffic lights and road markers. As depicted in Figure 1, they collectively form a topological structure that furnishes explicit navigation cues essential for downstream tasks like motion prediction and planning [1, 2].

Conventional driving datasets [3] typically incorporate lane topology implicitly within high-definition (HD) maps, which are primarily designed for data storage but not for learning. Various formulations have been proposed to substitute HD maps, such as 2D and 3D laneline detection [4, 5], bird's-eye-view (BEV) map element detection through segmentation [6, 7] and vectorization [8, 9]. To derive lane connectivity, a naïve strategy is to directly average the coordinates of two neighboring lanelines to get lane centerlines, and then construct a lane graph based on the centerline instances. Yet, it requires complicated hand-crafted rules and extensive post-processing. An alternative approach is to supervise perception networks with relationship labels. Recent studies [10] employ a Transformer-based architecture for lane instance prediction and an additional multi-layer perceptron (MLP) to learn connectivity. Nevertheless, they suffer from extracting valuable information without explicit modeling of relationships.

* Corresponding author (email: hongyang@hku.hk)
† These authors contributed equally to this work.

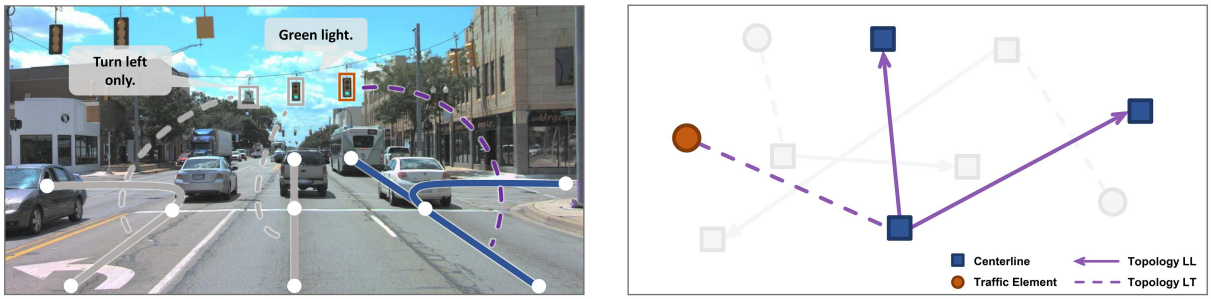

Figure 1 Topology relationships of a driving scene. In an intersection, an autonomous vehicle has to reason about the correct lane and traffic information for subsequent navigation. We advocate and present TopoNet to directly achieve topology understanding on the heterogeneous graph. "Topology LL" and "Topology LT" represent the relationship among lane centerlines and the associations between lane centerlines and traffic elements, respectively.

Moreover, the problem of assigning relationships between traffic elements and lanes based on sensor inputs remains unexplored. Previous work [11] attempted to associate the ground truth representations of lanelines and traffic lights in the image domain (in the perspective view, PV). However, integrating traffic elements and lanes within a heterogeneous graph (Figure 1) presents a distinct set of challenges. One key obstacle is that traffic elements are typically described as bounding boxes in PV, whereas lanes are depicted in 3D or BEV space. Besides, spatial locations remain less important for traffic elements as the semantic meanings are essential, but positional clues for lanes are crucial for self-driving vehicles.

To address these issues, we present a topology reasoning network (TopoNet), which predicts the driving scene topology in an end-to-end manner. As an attempt to reason about scene topology in a single network, TopoNet comprises a shared feature extractor, and two detection branches for traffic elements and centerlines, respectively. Motivated by the Transformer-based detection algorithms [12, 13], we employ instance queries to extract local features via the deformable attention, which confines the attention region and accelerates convergence. Since the clues for locating a specific centerline instance might lie in its neighboring elements and related traffic elements' features, a scene graph neural network (SGNN) is devised to facilitate message passing among instance-level embeddings. Furthermore, we propose a scene knowledge graph to capture prior topological knowledge from entities of different types. Specifically, a series of graph neural networks (GNNs) is developed based on the categories of traffic elements and the centerline connectivity relationships (i.e., predecessor, ego, successor). Updated queries are decoded to yield perception results and topology relationships. With the proposed designs, we deploy TopoNet on the large-scale topology reasoning benchmark, OpenLane-V2 [14]. TopoNet outperforms state-of-the-art (SOTA) approaches by 15%–84% in centerline perception, 38%–270% in topology reasoning tasks, and 37% in terms of the overall perception and reasoning metric. Ablations are conducted to verify the effectiveness of our framework.

2 Related work

Lane graph learning has gained significant attention due to its pivotal role in autonomous driving. STSU [10] proposes a detection transformer (DETR)-like network to detect centerlines and then derive their connectivity by a subsequent MLP module. LaneGAP [15] proposes a path-wise modeling approach to represent the lane graph. It is also worth noting that Tesla proposes the concept of the "language of lanes" to depict the lane graph as a sentence [16]. The attention-based model recursively predicts lane tokens and their connectivity. In this work, we focus on explicitly modeling the centerline connectivity within the network to enhance feature learning and incorporate traffic elements in the construction of a driving scene graph.

HD map perception. With the trend of BEV perception [17], recent studies focus on learning HD maps with segmentation and vectorized methods. Map segmentation predicts the classification of each BEV grid, such as lanelines and pedestrian crossings [18]. Though dense segmentation provides detailed pixel-level semantic information, it falls short in capturing the complex relationships between overlapping elements. Li et al. [6] addressed the problem by grouping and vectorizing the segmented map with sophisticated post-processing. VectorMapNet [8] proposes directly representing each map element as a sequence of points and using coarse key points to decode laneline locations sequentially. MapTR [9] further explores unified permutation-based modeling to eliminate ambiguities in point sequence ordering. In fact, vectorization-based methods could be easily adapted for centerline perception by adjusting the supervision since they have enriched the direction information for lanelines. Shin et al. [19] constructed map elements as a graph by initially predicting vertices and then utilizing a GNN module to detect edges. However,

its GNN produces all vertex features simultaneously, limiting instance-level interactions. Contrary to them, we leverage instance-wise feature transmission within the GNN, enabling the extraction of significant prediction cues from other elements in the topology graph.

Driving scene understanding. The concept of driving scene understanding primarily involves the comprehension of the spatial relationships among elements within outdoor environments, extending beyond mere detection [20,21]. Previous studies focus on utilizing the relationships of 2D bounding boxes for motion prediction [22,23] and risk assessment [24]. In the industrial context, Mobileye presents an optimization-based method to construct lane topology and relationships between traffic lights and lanes automatically based on their proprietary data sources [25]. In academia, Langenberg et al. [11] addressed the traffic light to lane assignment (TL2LA) problem with a convolutional network by taking heterogeneous metadata as additional inputs. In contrast, TopoNet takes only RGB images as inputs and additionally reasons about lane entities' topology. We train and evaluate TopoNet on the large-scale scene understanding benchmark, which covers complicated urban scenarios.

GNN. GNN and its variants, such as graph convolutional network (GCN) [26] and GAT [27], are widely adopted to aggregate features of vertices and extract information from a graph. Witnessing the impressive achievements of GNN in various fields (e.g., recommendation system and video understanding) [28, 29], researchers in the driving community try utilizing it to process unstructured data. LaneGCN [30] constructs a lane graph from HD map, while others [31, 32] model the relationship of moving agents and lanelines as a graph to improve the trajectory forecasting performance. Inspired by prior studies, we design a GNN for the driving scene understanding task to enhance feature interaction and introduce a class-specific knowledge graph to better integrate semantic information.

3 TopoNet

3.1 Problem formulation

Given multi-view images, the goal of TopoNet lies in two perspectives-perceiving entities and reasoning their relationships. As an instance-level representation is preferable for topology reasoning, a directed lane centerline (LC) is described by an ordered list of points. We denote it as $v_l = [p_0, \dots, p_{n-1}]$, where $p = (x, y, z) \in \mathbb{R}^3$ describes a point's coordinate in 3D space, p_0 and p_{n-1} are the starting and ending point. Traffic elements (TE) are represented as 2D bounding boxes in different classes on the front-view images. All existing lanes V_l and traffic elements V_t within a predefined range are required to be detected.

On the perceived instances, the topology relationships are built. The connectivity of directed lanes establishes a map-like network on which vehicles can drive. We denote the lane graph as $G_{ll} = (V_l, E_{ll})$, where the edge set $E_{ll} \subseteq V_l \times V_l$ is asymmetric. An entry (i, j) in E_{ll} is positive if and only if the ending point of the lane v_i is connected to the starting point of v_j. The graph $G_{lt} = (V_l \cup V_t, E_{lt})$ describes the correspondence between lanes and traffic elements. This graph can be interpreted as a bipartite structure, where positive edges only exist between V_l and V_t. Given the instance set V_l and V_t, the connectivity of predicted graph G_{ll} and G_{lt} is represented by the adjacency matrices A_{ll} and A_{lt}. These matrices are required to be predicted in the task of topology reasoning.

3.2 Overview

Figure 2 illustrates the overall architecture of the proposed TopoNet. Given multi-view images, the feature extractor generates multi-scale features, including a front-view feature F_{PV} and a BEV feature F_{BEV}. Two independent decoders with the same deformable attention mechanism [13] take F_{PV} and F_{BEV} to update instance-level embeddings Q_t and Q_l, respectively. The proposed SGNN then refines centerline queries Q_l in positional and topological aspects. The decoders and SGNN layers are stacked iteratively for N layers to facilitate local and global feature interactions. Task-specific heads employ the refined queries to get final results. Next, we elaborate on the proposed SGNN.

3.3 Scene graph neural network

A representative embedding (or query) provides ideal instance-wise detection or segmentation results, as discussed in previous perception studies [12, 33]. However, being discriminative is not enough to recognize correct topology relationships. The reason is that it takes a pair of instance queries to determine their relationship, in which feature embeddings are actually not independent. Meanwhile, adopting the local feature aggregation scheme of point-wise queries [8, 9] for centerline perception is inadequate. Specifically, a key difference between centerlines and physical map elements is that centerlines *per se* encode lane topology and traffic rules, which cannot be inferred from local

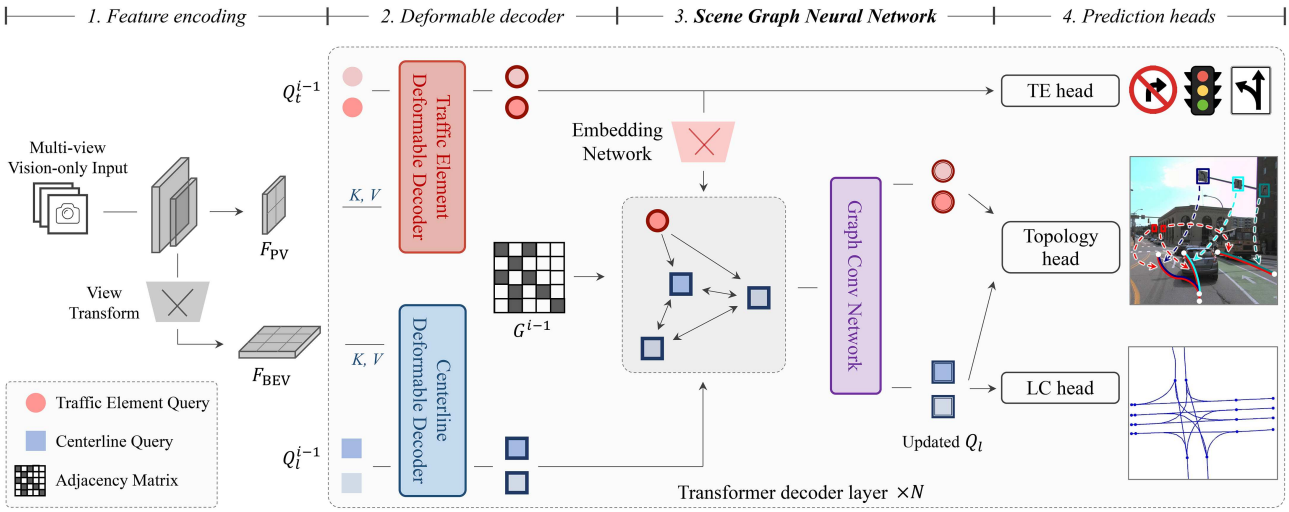

Figure 2 Systematic diagram of TopoNet. TopoNet addresses the crucial problem of topology reasoning for driving scenes in an end-to-end fashion. It consists of four stages, with the latter three compacted in a Transformer decoder architecture. TopoNet handles traffic elements and centerlines as two parallel branches at the deformable decoder. Various types of instance queries then interact, exchange messages, acquire and aggregate prominent knowledge in the proposed scene graph neural network. The explicit relationship modeling inside the network serves as a favorable scheme for feature learning and topology prediction. In this paper, we abbreviate traffic elements and lane centerlines as "TE" and "LC", respectively.

features alone. Therefore, we aim to simultaneously acquire perception and reasoning results by modeling not only discriminative instance-level representations but also inter-entity relationships.

To this end, we present SGNN, which distinguishes itself by several designs and merits compared to previous studies. (1) It adopts an embedding network to extract TE knowledge in a unified feature space. (2) It models all entities in a frame as vertices in a graph, and enhances interconnection among perceived instances to learn their inherent relationships with a GNN. (3) Alongside the graph structure, SGNN incorporates prior topology knowledge with a scene knowledge graph.

3.3.1 *Embedding network*

As traffic elements are annotated in PV, it is hard to harness their positional information in the 3D feature space. However, their semantic meaning has a great effect. For instance, a road sign indicating the prohibition of a left turn usually corresponds to lanes that lay in the middle of the road. This predefined knowledge is beneficial for locating corresponding lanes. We introduce an embedding network to extract semantic information and transform it into a unified feature space to match with centerlines. This process is formulated as $\widetilde{Q}_t^i = \texttt{embedding}^i(Q_t^i)$, where i denotes the i-th decoder layer. Note that the queries $\widetilde{Q}_t^i$ remain intact in the SGNN and are not used in traffic element detection. This is intended since imagining traffic elements from centerlines is relatively challenging. Besides, it would be hard to predict traffic elements' attributes in the image feature space if their features have been transformed into the lane centerline feature space and further updated through interactions.

3.3.2 *Feature propagation in GNN*

In this part, we introduce how topological relationships are modeled and how knowledge from different queries is exchanged. Using GNN, relations are conveniently formulated as edges in a graph where entities are seen as vertices. However, it is nontrivial in driving scenarios, as there is no explicit constraint or prior knowledge of the topology structure. A possible way is to construct a fully connected graph (V, E), with $V = V_l \cup V_t$ and $E \subseteq V \times V$. However, this inevitably increases computational cost and introduces unnecessary information transmission, such as between two traffic elements that are placed subjectively by humans. Instead, we use $G_{ll} = (V_l, V_l \times V_l)$ for lane graph estimation and $G_{lt} = (V_l \cup V_t, V_l \times V_t)$ representing the predicted TE to LC assignments, to guide the information transmission.

In graphs G_{ll} and G_{lt}, lane queries Q_l are refined by the connected neighbors and corresponding traffic elements. The semantic gap still exists since Q_l and Q_t represent different kinds of objects. We introduce an adapter layer to combine this heterogeneous information into the information gain denoted as R. The overall process in an SGNN

layer is formulated as

$$\begin{aligned} Q_l^{i'} &= \texttt{SGNN}_{ll}^i\big(Q_l^i, G_{ll}^{i-1}\big), \\ Q_l^{i''} &= \texttt{SGNN}_{lt}^i\big(Q_l^i, \widetilde{Q}_t^i, G_{lt}^{i-1}\big), \\ R^i &= \texttt{downsample}^i\Big(\texttt{ReLU}\big(\texttt{concat}(Q_l^{i'}, Q_l^{i''})\big)\Big), \\ \widetilde{Q}_l^i &= Q_l^i + R^i. \end{aligned} \tag{1}$$

3.3.3 *Vanilla scene graph*

Given the directed lane graph G_{ll}^{i-1} predicted by the previous layer, our goal is to construct a weight matrix T_{ll}^i that controls the flow of messages in the current layer. In this directed graph, messages typically propagate in a single direction, such as from a centerline to its successor. However, the spatial position of a lane can serve as a good indication of the locations of neighboring lanes, which suggests that a bidirectional information exchange could be advantageous. To facilitate this, we augment the weighted adjacency matrix A_{ll}^{i-1} of G_{ll}^{i-1} by incorporating backward edges to construct T_{ll}^i, thereby enabling message exchange between two connected centerlines. The process can be formulated as

$$T_{ll}^i = \beta_{ll} \cdot \big(A_{ll}^{i-1} + \texttt{transpose}(A_{ll}^{i-1})\big) + I, \tag{2}$$

where $T_{ll}^0 = I$ and I denotes the identical mapping for self-loop, β_{ll} is a hyperparameter to control the ratio of features propagated between nodes.

In the bipartite graph G_{lt}, where only the correspondence between lanes and traffic elements is presented, we utilize features of traffic elements to refine centerline embeddings

$$T_{lt}^i = \beta_{lt} \cdot A_{lt}^{i-1}, \tag{3}$$

where $T_{lt}^0 = O$ is a matrix in which all entries are zero.

After obtaining the weight matrices, SGNN utilizes the GCN [26] to perform feature propagation among queries

$$\begin{aligned} Q_l^{i'} &= \texttt{GCN}_{ll}^i\big(Q_l^i, T_{ll}^i\big), \\ Q_l^{i''} &= \texttt{GCN}_{lt}^i\big(Q_l^i, \widetilde{Q}_t^i, T_{lt}^i\big). \end{aligned} \tag{4}$$

3.3.4 *Scene knowledge graph*

Though the vanilla scene graph enables feature propagation with GCN layers and treats nodes differently based on their connectivity, the semantic meaning of vertices remains unused. For example, a traffic element indicating to go straight is not equally important as that indicating a red light. To incorporate categorical prior, we design the scene knowledge graph to treat vertices in different classes differently. Figure 3 illustrates an example process of updating a centerline query LC_1 on the given knowledge graph.

On the graph G_{lt}, we use $\mathbf{W}_{lt}^i \in \mathbb{R}^{|C_t| \times F_l \times F_t}$ to denote the learnable weights, where C_t describes the attribute set of traffic elements, F_l and F_t are the number of feature channel of LC and TE queries, respectively. A centerline query with index x aggregates information from its corresponding traffic elements based on their classification scores. The expanded formula for the message propagation is

$$\begin{aligned} K_{lt}^i &= A_{lt}^{i-1}, \\ Q_{l_{(x)}}^{i''} &= \sum_{y \in N(x)} \sum_{c_t \in C_t} \beta_{lt} \cdot S_{t_{(c_t,y)}}^i K_{lt_{(x,y)}}^i \mathbf{W}_{lt_{(c_t)}}^i \widetilde{Q}_{t_{(y)}}^i, \end{aligned} \tag{5}$$

where $N(x)$ outputs the indices of all neighbors of the vertex with index x, and $S_t^i \in \mathbb{R}^{|C_t| \times |Q_t^i|}$ represents the classification scores predicted by the TE branch.

Although all centerlines fall into the same category, the directed connection nature, namely predecessor and successor, still poses an impact on the process of feature propagation. Thus, we formulate the learnable weight matrix for the lane graph as $\mathbf{W}_{ll}^i \in \mathbb{R}^{|C_l| \times F_l \times F_l}$, where $C_l = \{\text{successor}, \text{predecessor}, \text{self-loop}\}$. The centerline queries are further updated by

$$\begin{aligned} K_{ll}^i &= \texttt{stack}\big(A_{ll}^{i-1}, \texttt{transpose}(A_{ll}^{i-1}), I\big), \\ Q_{l_{(x)}}^{i'} &= \sum_{y \in N(x)} \sum_{c_l \in C_l} \beta_{ll} \cdot K_{ll_{(c_l,x,y)}}^i \mathbf{W}_{ll_{(c_l)}}^i Q_{l_{(y)}}^i. \end{aligned} \tag{6}$$

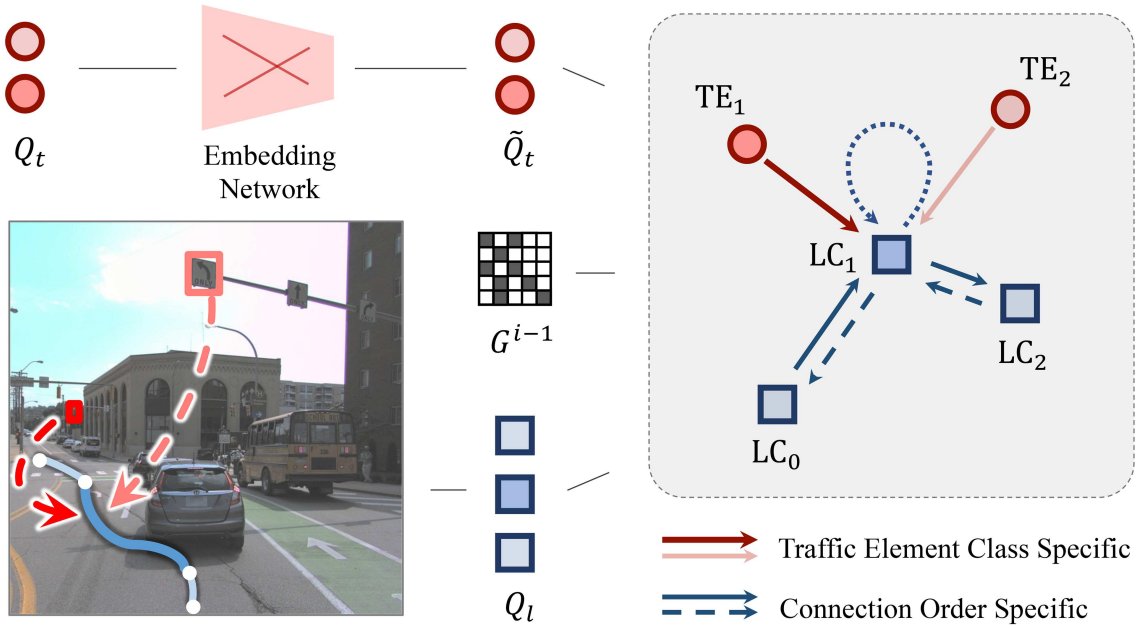

Figure 3 Scene knowledge graph illustration. For the centerline colored blue in the left case, the related weight matrices in the graph are categorically independent. Different traffic elements and lane-directed connections bring different information to the centerline, which is encoded as a scene knowledge graph on the right.

3.4 Learning

We employ multiple losses to train TopoNet in an end-to-end manner. As depicted in Figure 2, all heads utilize queries to generate perception and reasoning results. Nevertheless, they are not entirely independent, as the topology head requires matching results from perception heads. Similar to Transformer-based networks [12, 13], the supervision is applied on each decoder layer to optimize the query feature iteratively. The overall loss of the proposed model is $\mathcal{L} = \mathcal{L}_{det_{\mathrm{TE}}} + \mathcal{L}_{det_{\mathrm{LC}}} + \mathcal{L}_{top}$.

Perception. Following the head design in DETR [12], the TE head predicts 2D bounding boxes with classification scores. The LC head produces 11 ordered 3D points and a confidence score from each centerline query $\tilde{q}_l \in \widetilde{Q}_l$. The ground truth coordinate of centerlines is normalized based on the predefined BEV range. The Hungarian algorithm is utilized to match ground truth and predictions, with matching cost the same as the loss function. Then task-specific losses $\mathcal{L}_{det_{\mathrm{TE}}}$ and $\mathcal{L}_{det_{\mathrm{LC}}}$ are applied accordingly.

Reasoning. The topology head reasons pairwise relationships on the given embeddings to predict A_{ll} and A_{lt}. Similar to STSU [10], for a pair of instances, we use two MLP layers to reduce the feature dimension for each instance. Then the concatenated feature is sent into another MLP with a sigmoid activation to predict their relationship. Based on the matching results from perception heads, the ground truth of each pair of embeddings is assigned. In the LC head, we adopt embeddings generated in the SGNN module, i.e., the refined queries $\widetilde{Q}_l$ for lanes and the semantic embeddings $\widetilde{Q}_t$ for traffic elements. Due to the sparsity of the graph, focal loss is deployed in $\mathcal{L}_{top}$ to deal with the imbalance in sample distribution.

4 Experiments

4.1 Implementation details

Feature encoding. We adopt an ImageNet pre-trained ResNet-50 [34], with a feature pyramid network (FPN) [35] to obtain multi-scale image features. Then we adopt a view Transformer with 3 encoder layers proposed in BEVFormer [7]. Note that we do not use temporal information, and thus the temporal self-attention layer in the BEVFormer encoder is replaced by a deformable attention [13] layer. The size of BEV grids is set to 200 × 100, with four different height levels of −1.5, −0.5, +0.5 and +1.5 m relative to the ground.

Deformable decoder. For the decoder, we utilize the decoder layer in deformable DETR [13]. The dimension of initial queries $q = [q_p, q_o] \in Q$ is set to 256, where q_p is utilized to generate the initial reference point, and q_o is the initial object query. The query number for centerlines and traffic elements is set to 200 and 100. The reference

points will remain unchanged across different layers.

Scene graph neural network. We utilize a simplified version of GCN [26] as our GNN layer. Given an input matrix $Q^i \in \mathbb{R}^{N\times C}$, with N representing the number of nodes and C denoting the number of channels, the output of the operation is

$$Q^{i'} = \sigma\Big(T^i Q^i \mathbf{W}^i\Big), \tag{7}$$

where $\mathbf{W}^i \in \mathbb{R}^{C\times C}$ is the learnable weight matrix, $T^i \in \mathbb{R}^{N'\times N}$ describes the adjacency matrix with N' output nodes, and $\sigma(\cdot)$ is the activation function. Note that the matrix T is inferred without gradients during training. For the traffic element, an embedding network is employed before each GNN layer. The embedding network is a two-layer MLP, in which the output channels are 512 and 256. Between the MLP layers, a ReLU activation function and a dropout layer are included. β_{ll} and β_{lt} are set to 0.6.

Loss. $\mathcal{L}_{det_{\mathrm{TE}}}$ includes a classification, a regression, and a intersection over union (IoU) loss that $\mathcal{L}_{det_{\mathrm{TE}}} = \lambda_{cls}\cdot\mathcal{L}_{cls} + \lambda_{reg}\cdot\mathcal{L}_{reg} + \lambda_{iou}\cdot\mathcal{L}_{iou}$. λ_{cls}, λ_{reg}, and λ_{iou} are set to 1.0, 2.5, and 1.0, respectively. $\mathcal{L}_{cls}$ is a focal loss. For centerline detection, $\mathcal{L}_{det_{\mathrm{LC}}}$ comprises a classification and a regression loss that $\mathcal{L}_{det_{\mathrm{LC}}} = \lambda_{cls}\cdot\mathcal{L}_{cls} + \lambda_{reg}\cdot\mathcal{L}_{reg}$, where λ_{cls} and λ_{reg} are 1.5 and 0.025, respectively. Note that the regression loss is calculated on the denormalized 3D coordinates. For topology reasoning, we adopt the same focal loss but different weights on different types of relationships. The loss $\mathcal{L}_{top}$ is defined as $\lambda_{top_{ll}}\cdot\mathcal{L}_{top_{ll}} + \lambda_{top_{lt}}\cdot\mathcal{L}_{top_{lt}}$, where both $\lambda_{top_{ll}}$ and $\lambda_{top_{lt}}$ are 5.0.

Training. We adopt the AdamW optimizer [36] and a cosine annealing schedule with an initial learning rate of 1×10^{-4}. TopoNet is trained for 24 epochs with a total batch size of 8, distributed across 8 NVIDIA A100 GPUs. For data augmentation, 0.5× resizing and color jitter are applied.

Baselines. We adopt three SOTA algorithms which are designed for lane graph estimation or map learning: STSU [10], VectorMapNet [8], and MapTR [9]. We also compare our method with HDMapNet [6] and LaneGAP [15], the map learning method with different lane formulations, on the BEV segmentation metrics. To ensure a fair comparison, we employ the same input resolution, the same ResNet-50 image backbone, and the same FPN neck for extracting features from surrounding images. Additionally, we incorporate a deformable DETR head specifically for traffic elements, aligning all settings with TopoNet. As for topology reasoning, we utilize their own modeling concepts of instance query, and the topology heads for each method are the same MLPs as in TopoNet. All the methods are trained for 24 epochs to ensure a fair comparison.

4.2 Dataset and metrics

We conduct experiments on the OpenLane-V2 benchmark [14]. The dataset contains topological structures in the driving scenes. Ablation studies are conducted on its subset_A.

Dataset. The OpenLane-V2 benchmark includes images from 2000 scenes collected worldwide under different environments. The dataset is split into two subsets, namely subset_A and subset_B, built on top of the Argoverse 2 [37] and nuScenes [3] datasets, respectively. Each subset contains 1000 scenes with multi-view images and annotations at 2 Hz. Lanes are represented as centerlines within a region spanning from -50 to $+50$ m along the x-axis and from -25 to $+25$ m along the y-axis. Traffic elements are annotated as 2D bounding boxes on the front-view images with 13 types of attributes. The topology relationships are provided in the form of adjacency matrices based on the ordering of centerlines and traffic elements.

Perception metrics. The DET score is the typical mean average precision (mAP) for measuring instance-level perception performance. The DET_l score uses Fréchet distance [38] as the similarity measure, which is very sensitive to line direction and local deviation and thus suitable for evaluating directional lane centerlines. The final score is averaged over match thresholds of $\mathbb{T} = \{1.0, 2.0, 3.0\}$. Note that the defined BEV range is relatively large compared to other map detection benchmarks [6, 8], and accurate perception of lanes in the distance is hard. As a result, thresholds $\mathbb{T}$ are relaxed based on the distance between the lane and the ego car. For traffic elements, the DET_t uses IoU as the similarity measure and is averaged over different types of attributes $\mathbb{A}$ of traffic elements.

Reasoning metrics. The TOP score is an mAP metric adapted from the graph domain. Specifically, given a ground truth graph $G = (V, E)$ and a predicted one $\hat{G} = (\hat{V}, \hat{E})$, it builds a set of vertices $\hat{V}'$ by a projection from $\hat{V}$ such that $V = \hat{V}' \subseteq \hat{V}$, where the Fréchet and IoU distances are utilized for similarity measure among lane centerlines and traffic elements respectively. Inside the predicted $\hat{V}'$, two vertices are regarded as connected if the confidence of the edge is greater than 0.5. Then the TOP score is the averaged vertex mAP between (V, E) and $(\hat{V}', \hat{E}')$ over all vertices:

$$\mathrm{TOP} = \frac{1}{|V|}\sum_{v\in V}\frac{\sum_{\hat{n}'\in\hat{N}'(v)} P(\hat{n}')\mathbb{1}(\hat{n}'\in N(v))}{|N(v)|}, \tag{8}$$

Table 1 Comparison with SOTA methods on the OpenLane-V2 benchmark. "*" indicates that topology reasoning evaluation is based on matching results on Chamfer distance. The highest score is in bold, while the second one is underlined.

Data	Method	$\text{DET}_l\uparrow$	$\text{TOP}_{ll}\uparrow$	$\text{DET}_t\uparrow$	$\text{TOP}_{lt}\uparrow$	OLS↑
subset_*A*	STSU [10]	12.7	0.5	43.0	<u>15.1</u>	25.4
	VectorMapNet [8]	11.1	0.4	41.7	5.9	20.8
	MapTR [9]	8.3	0.2	<u>43.5</u>	5.9	20.0
	MapTR* [9]	<u>17.7</u>	<u>1.1</u>	<u>43.5</u>	10.4	<u>26.0</u>
	TopoNet (ours)	**28.5**	**4.1**	**48.1**	**20.8**	**35.6**
subset_*B*	STSU [10]	8.2	0.0	43.9	<u>9.4</u>	21.2
	VectorMapNet [8]	3.5	0.0	49.1	1.4	16.3
	MapTR [9]	8.3	0.1	<u>54.0</u>	3.7	21.1
	MapTR* [9]	<u>15.2</u>	<u>0.5</u>	<u>54.0</u>	6.1	<u>25.2</u>
	TopoNet (ours)	**24.3**	**2.5**	**55.0**	**14.2**	**33.2**

where $N(v)$ denotes the ordered list of neighbors of vertex v ranked by confidence, $\hat{N}'(v)$ denotes the ordered list of predicted neighbors of vertex v, and $P(v)$ is the precision of the i-th vertex v in the ordered list. The TOP_{ll} is for topology among centerlines on graph (V_l, E_{ll}), and the TOP_{lt} for topology between lane centerlines and traffic elements on graph $(V_l \cup V_t, E_{lt})$.

Overall metrics. The OpenLane-V2 score (OLS) summarizes metrics covering different aspects of it:

$$\text{OLS} = \frac{1}{4}\left[\text{DET}_l + \text{DET}_t + f(\text{TOP}_{ll}) + f(\text{TOP}_{lt})\right], \tag{9}$$

where f is the square root function.

4.3 Main results

In Table 1, we compare the proposed TopoNet to several SOTA methods. TopoNet outperforms all previous algorithms by a large margin. As the SOTA map learning method MapTR ignores the direction of centerlines with the permutation-equivalent modeling [9], we additionally evaluate MapTR based on Chamfer distance matching. However, its performance on DET_l, as well as topology metrics, significantly degenerates from TopoNet under fair comparison. The large performance gap indicates that reasoning the complex topology raises greater challenges upon merely perceiving presented instances, highlighting the effectiveness of TopoNet's design. All methods achieve similar DET_t, since we adopt the same traffic element detection branch. In more detail, TopoNet possesses slightly superior traffic light detection performance, which indicates that its comprehensive framework is capable of performing heterogeneous feature learning between traffic elements and centerlines, thereby enhancing the performance of DET_t and TOP_{lt}. On the other hand, since all methods employ a shared backbone, it should be noted that the convergence of traffic light detection could be influenced by other branches, especially when the model struggles to learn centerlines and topological information with a large loss in the LC head. Therefore, given that all methods have the same TE head, our experimental analysis primarily focuses on centerline detection and topology reasoning. Regarding LC-TE topology reasoning, the performance of TopoNet benefits from its overall superior centerline and traffic element detection performance as well as the proposed SGNN module, in which different entities are modeled differently.

Comparison on centerline perception. To have a fair comparison, we use a unified backbone and PV-to-BEV transformation module for various SOTA methods on the centerline perception task. We keep the topology supervision for STSU, as it was originally designed for detecting centerlines and their topology relationship. Since VectorMapNet and MapTR initially target laneline detection where there is no relationship between visible lanelines, we alter the supervision from laneline to centerline and ignore topology supervision to preserve their original design choice.

To better align with previous studies [8, 9], we also provide $\text{DET}_{l,\text{chamfer}}$ with the Chamfer distance as the similarity measure. It does not take the lane direction into account and is thresholded on $\{0.5, 1.0, 1.5\}$. As shown in Table 2, TopoNet outperforms other methods on all metrics. We also find that the original design of online mapping approaches struggles with managing lane topology and traffic elements. As shown in Tables 1 and 2, when the effect from lane topology and traffic elements is removed, MapTR's performance in centerline detection improves from 17.7 to 21.7 on $\text{DET}_{l,\text{chamfer}}$ score. In contrast, TopoNet's performance in centerline detection decreased by 0.8 points on DET_l due to the removal of the traffic element branch and the lane-traffic element feature interaction in SGNN. This suggests that TopoNet benefits from detecting traffic elements and reasoning

Table 2 Comparison on centerline perception with a unified feature extractor. "Topology" denotes that the network is trained with topology supervision.

Method	Topology	DET_l↑	TOP_{ll}↑	$\text{DET}_{l,\text{chamfer}}$↑	FPS
STSU [10]	✓	14.2	0.6	13.8	**12.8**
VectorMapNet [8]	✗	12.7	–	10.3	1.0
MapTR [9]	✗	10.0	–	21.7	11.5
TopoNet (ours)	✓	**27.7**	**4.6**	**27.4**	10.1

Table 3 Comparison on BEV segmentation. When rendering centerlines on the BEV grids, TopoNet also outperforms the previous approach.

Method	HDMapNet [6]	STSU [10]	VectorMapNet [8]	MapTR [9]	LaneGAP [15]	TopoNet (ours)
mIoU↑	18.3	24.6	18.9	32.1	35.0	**35.1**

Table 4 Ablation on the design of scene graph neural network. "SG" represents the vanilla scene graph, and "SKG" is the enhanced SGNN with the proposed scene knowledge graph.

Method	DET_l↑	TOP_{ll}↑	DET_t↑	TOP_{lt}↑	OLS↑
Baseline	25.7	4.0	47.2	20.6	34.6
+ SG	27.7	3.7	48.0	20.1	35.0
+ SKG	**28.5**	**4.1**	**48.1**	**20.8**	**35.6**

the LT topology, attributable to the effective design of our pipeline. Besides, the FPS of TopoNet is 10.1 on an A100 bare machine. Compared to other methods on the same machine with an aligned input size of 512×676, our method has comparable online efficiency but stronger performance.

Comparison on BEV segmentation. DET_l is defined to evaluate the validity of each point on a single centerline, ensuring a consistent instance representation of lanes. In contrast, IoU focuses on pixel-level accuracy in segmentation formulation. It provides a fair comparison of the overall geometric accuracy across various methods with different lane formulations, such as HDMapNet [6] and LaneGAP [15]. Except for HDMapNet, the vectorized centerline predictions of each method are rendered to BEV with a fixed line width of 0.75 m aligned with the setting in HDMapNet. As shown in Table 3, TopoNet surpasses other methods in terms of IoU. We also conduct a fair comparison with LaneGAP [15], which utilizes path-wise modeling to represent the lane graph. Transforming lane paths into lane pieces in the LaneGAP's post-processing stage necessitates high geometric accuracy, making it unsuitable for evaluation using DET_l. This method achieves a similar performance to TopoNet in terms of IoU. However, we note that piece-wise modeling of TopoNet can effectively capture the precise locations of lane splits or merges, as well as the topology between lanes and traffic elements, making it more suitable for practical applications.

4.4 Ablation study

Effect of design in scene graph neural network. We alternate the proposed network into a baseline without feature propagation by downgrading the SGNN module to an MLP and supervising topology reasoning at the final decoder layer only. As illustrated in Table 4, the proposed SKG outperforms models in other settings, demonstrating its effectiveness for topology understanding. However, directly mixing information without accounting for semantic distinctions in the SG variant leads to performance degradation. This suggests that information from different types of traffic elements, as well as lane predecessors and successors, varies in importance and should be treated accordingly. Compared to the SG version, the scene knowledge graph provides an additional improvement of 0.8% for centerline perception, owing to the predefined semantic prior encoded in the categories of traffic elements. The improvement of traffic element detection and topology reasoning is also consistent. Given that Transformers are widely regarded as a variant of GNN, this also reveals that explicitly designing the feature interaction between queries within a Transformer decoder can further enhance performance, especially when instances have a strong correlation.

Effect on feature propagation. In the "LL only" setting, we set the β_{lt} parameter to 0. Similar to the baseline, we remove the concatenation and down-sampling operations, as well as the traffic element embedding. For "LT only", we set the β_{ll} parameter to 0, while other modules remain intact. Results are reported in Table 5. In the "LL only" setting, the drop on TOP_{lt} demonstrates the importance of the graph G_{lt}. Besides, it can be observed that the performance of DET_l experiences a certain decline under this setting as well. This might result from the lack of traffic element features' guidance for lane centerline detection within intersections. Compared to non-intersection areas, there is a higher number of centerlines within intersections, while they lack distinct lane

Table 5 Ablation on feature propagation in the SGNN. "LL only" denotes spatial information from lane connectivity, and "LT only" includes lane-traffic element relationship.

Method	DET_l↑	TOP_{ll}↑	DET_t↑	TOP_{lt}↑	OLS↑
LL only	27.9	3.8	47.8	20.3	35.1
LT only	27.8	3.9	47.5	20.5	35.1
TopoNet	**28.5**	**4.1**	**48.1**	**20.8**	**35.6**

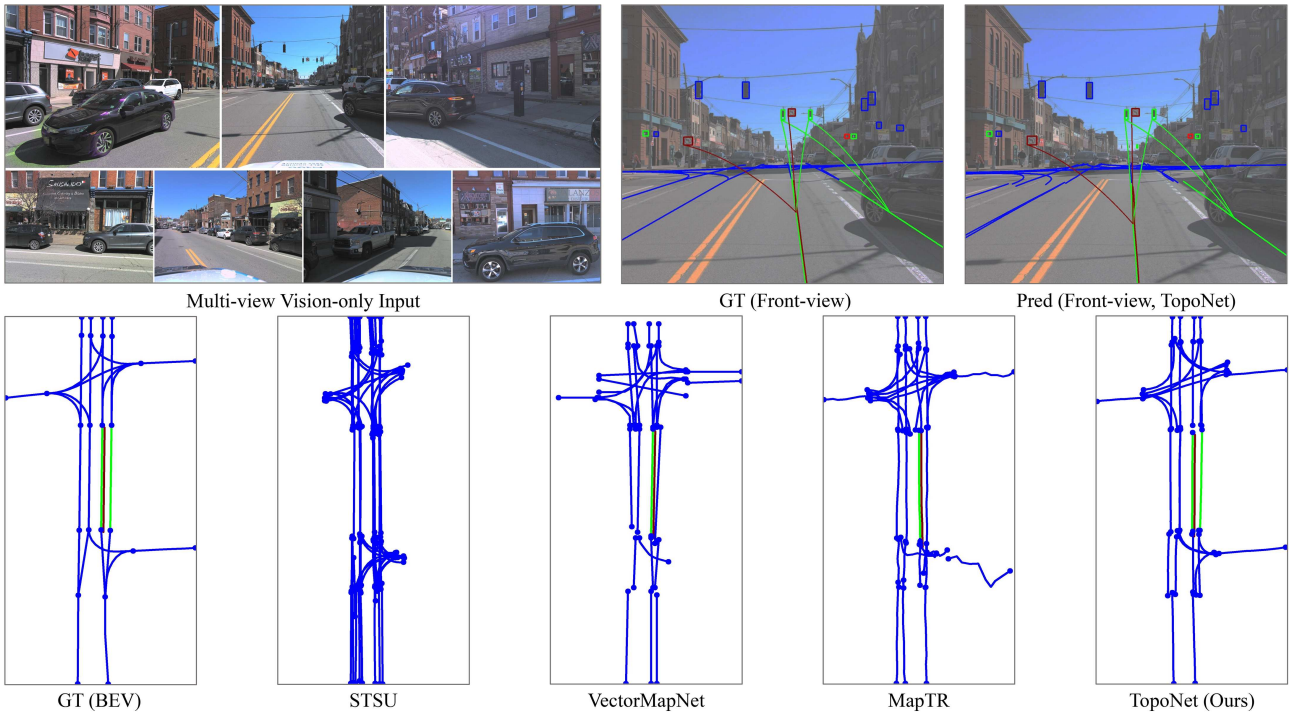

Figure 4 Qualitative results on subset_*A* of the OpenLane-V2 dataset. TopoNet achieves superior lane graph prediction performance compared to other SOTA methods in the complex intersection scenario. It also successfully builds all connections between traffic elements and lanes. Colors denote categories of traffic elements.

marking features and require traffic elements' guidance.

With the "LT only" design, DET_l degenerates when removing the graph G_{ll}, showing the importance of feature propagation between centerline queries. These experiments show that both branches are necessary for achieving satisfactory model performance on the primary task.

4.5 Qualitative analysis

A qualitative comparison on validation set is depicted in Figure 4. We show the raw output of each method, without the post-processing technique in STSU [10], to avoid potential accumulated inaccuracies and misalignment with quantitative evaluation. TopoNet predicts most centerlines correctly and constructs a lane graph in BEV. Yet, prior works fail to output all entities or get confused about their connectivity.

Figure 5 shows a case where a bus occludes the intersection in the front-view image. TopoNet fails to predict lanes and the topology, especially those in the left half of the crossing.

5 Conclusion and future work

In this paper, we discuss abstracting driving scenes as topology relationships of various entities and propose TopoNet, to address the problem. Importantly, our method models feature interactions via the graph neural network architecture and incorporates traffic knowledge in heterogeneous feature spaces with the knowledge graph-based design. Our experiments on the OpenLane-V2 benchmark demonstrate that TopoNet outperforms previous SOTA approaches in perceiving and reasoning about the driving scene topology under complex urban scenarios.

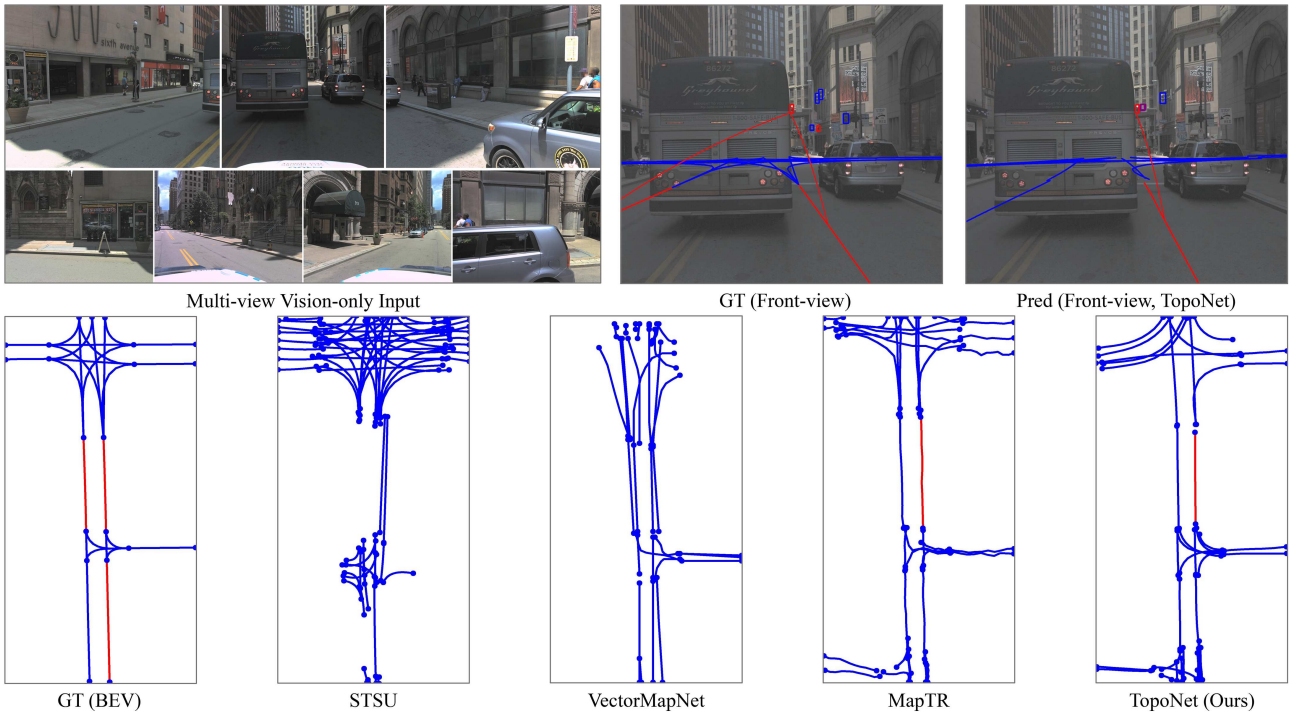

Figure 5 Failure case under large-area occlusion. TopoNet fails to predict centerlines and the lane graph in the intersection with a large bus blocking in front. Note that the relationship between the left lane and the red light is an incorrect annotation where our algorithm reasons about the direction of the left lane and avoids the false positive prediction.

Limitations and future work. Benefiting from the query-based design for feature propagation, TopoNet performs well in outputting positive predictions. However, post-processes such as merging or pruning are still needed to produce clean output, just as other lane topology studies [10]. The topic of incorporating the merging ability with auto-regressive or other mechanisms deserves future exploration. Our initial arXiv version [39] has inspired subsequent investigations; these recent studies explore ideas in [40–45].

Acknowledgements This work was supported by National Key R&D Program of China (Grant No. 2022ZD0160104), National Natural Science Foundation of China (Grant No. 62206172), and Shanghai Committee of Science and Technology (Grant No. 23YF1462000).

References

1 Bansal M, Krizhevsky A, Ogale A. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. 2018. ArXiv:1812.03079
2 Chai Y, Sapp B, Bansal M, et al. Multipath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. 2019. ArXiv:1910.05449
3 Caesar H, Bankiti V, Lang A H, et al. Nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 11621–11631
4 Pan X, Shi J, Luo P, et al. Spatial as deep: Spatial CNN for traffic scene understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence, 2018. 7276–7283
5 Chen L, Sima C, Li Y, et al. Persformer: 3D lane detection via perspective transformer and the OpenLane benchmark. In: Proceedings of the European Conference on Computer Vision, 2022. 550–567
6 Li Q, Wang Y, Wang Y, et al. Hdmapnet: An online HD map construction and evaluation framework. In: Proceedings of the International Conference on Robotics and Automation, 2022. 4628–4634
7 Li Z, Wang W, Li H, et al. BEVFormer: Learning bird's-eye-view representation from LiDAR-camera via spatiotemporal transformers. IEEE Trans Pattern Anal Mach Intell, 2024, 47: 2020–2036
8 Liu Y, Yuan T, Wang Y, et al. Vectormapnet: End-to-end vectorized HD map learning. In: Proceedings of the International Conference on Machine Learning, 2023. 22352–22369
9 Liao B, Chen S, Wang X, et al. Maptr: Structured modeling and learning for online vectorized HD map construction. In: Proceedings of the International Conference on Learning Representations, 2023
10 Can Y B, Liniger A, Paudel D P, et al. Structured bird's-eye-view traffic scene understanding from onboard images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021. 15661–15670
11 Langenberg T, Lüddecke T, Wörgötter F. Deep metadata fusion for traffic light to lane assignment. IEEE Robot Autom Lett, 2019, 4: 973–980

12 Carion N, Massa F, Synnaeve G, et al. End-to-end object detection with transformers. In: Proceedings of the European Conference on Computer Vision, 2020. 213–229
13 Zhu X, Su W, Lu L, et al. Deformable detr: Deformable transformers for end-to-end object detection. 2020. ArXiv:2010.04159
14 Wang H, Li T, Li Y, et al. Openlane-v2: A topology reasoning benchmark for unified 3D HD mapping. In: Proceedings of the 37th International Conference on Neural Information Processing Systems, 2024. 18873–18884
15 Liao B, Chen S, Jiang B, et al. Lane graph as path: Continuity-preserving path-wise modeling for online lane graph construction. In: Proceedings of the European Conference on Computer Vision, 2024. 334–351
16 Tesla. Tesla AI Day. 2022. https://www.youtube.com/watch?v=ODSJsviD_SU
17 Li H, Sima C, Dai J, et al. Delving into the devils of bird's-eye-view perception: a review, evaluation and recipe. IEEE Trans Pattern Anal Mach Intell, 2023, 46: 2151–2170
18 Philion J, Fidler S. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. In: Proceedings of the European Conference on Computer Vision, 2020. 194–210
19 Shin J, Jeong H, Rameau F, et al. InstaGraM: Instance-level graph modeling for vectorized HD map learning. IEEE Trans Intell Transp Syst, 2025, 26: 1889–1899
20 Tian Y, Carballo A, Li R, et al. Road scene graph: A semantic graph-based scene representation dataset for intelligent vehicle. 2020. ArXiv:2011.13588
21 Malawade A V, Yu S Y, Hsu B, et al. Roadscene2vec: A tool for extracting and embedding road scene-graphs. Knowl-Based Syst, 2022, 242: 108245
22 Li C, Meng Y, Chan S H, et al. Learning 3D-aware egocentric spatial-temporal interaction via graph convolutional networks. In: Proceedings of the IEEE International Conference on Robotics and Automation, 2020. 8418–8424
23 Mylavarapu S, Sandhu M, Vijayan P, et al. Understanding dynamic scenes using graph convolution networks. In: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, 2020. 8279–8286
24 Malawade A V, Yu S Y, Hsu B, et al. Spatiotemporal scene-graph embedding for autonomous vehicle collision prediction. IEEE Int Things J, 2022, 9: 9379–9388
25 Mobileye. Mobileye under the hood. 2022. https://www.mobileye.com/ces-2022/
26 Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks. 2016. ArXiv:1609.02907
27 Veličković P, Cucurull G, Casanova A, et al. Graph attention networks. 2017. ArXiv:1710.10903
28 Mohamed A, Qian K, Elhoseiny M, et al. Social-stgcnn: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020. 14424–14432
29 Pradhyumna P, Shreya G P. Graph neural network (GNN) in image and video understanding using deep learning for computer vision applications. In: Proceedings of the 2nd International Conference on Electronics and Sustainable Communication Systems, 2021. 1183–1189
30 Liang M, Yang B, Hu R, et al. Learning lane graph representations for motion forecasting. In: Proceedings of the European Conference on Computer Vision, 2020. 541–556
31 Jia X, Wu P, Chen L, et al. HDGT: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding. IEEE Trans Pattern Anal Mach Intell, 2023, 45: 13860–13875
32 Fang J, Zhu C, Zhang P, et al. Heterogeneous trajectory forecasting via risk and scene graph learning. IEEE Trans Intell Transp Syst, 2023, 24: 12078–12091
33 Wu J, Jiang Y, Bai S, et al. Seqformer: Sequential transformer for video instance segmentation. In: Proceedings of the European Conference on Computer Vision, 2022. 553–569
34 He K, Zhang X, Ren S, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016. 770–778
35 Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017. 2117–2125
36 Loshchilov I. Decoupled weight decay regularization. 2017. ArXiv:1711.05101
37 Wilson B, Qi W, Agarwal T, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. 2023. ArXiv:2301.00493
38 Eiter T, Mannila H. Computing Discrete Fréchet Distance. Technical Report CD-TR 94/64. 1994
39 Li T, Chen L, Wang H, et al. Graph-based topology reasoning for driving scenes. 2023. ArXiv:2304.05277
40 Li T, Jia P, Wang B, et al. Lanesegnet: Map learning with lane segment perception for autonomous driving. In: Proceedings of the International Conference on Learning Representations, 2024
41 Wu D, Chang J, Jia F, et al. Topomlp: A simple yet strong pipeline for driving topology reasoning. In: Proceedings of the International Conference on Learning Representations, 2024
42 Ma Z, Liang S, Wen Y, et al. Roadpainter: Points are ideal navigators for topology transformer. In: Proceedings of the European Conference on Computer Vision, 2024. 179–195
43 Fu Y, Liao W, Liu X, et al. Topologic: An interpretable pipeline for lane topology reasoning on driving scenes. In: Proceedings of the 38th International Conference on Neural Information Processing Systems, 2024. 61658–61676
44 Fu Y, Liu X, Li T, et al. Topopoint: Enhance topology reasoning via endpoint detection in autonomous driving. In: Proceedings of the 38th International Conference on Neural Information Processing Systems, 2025
45 Li H, Huang S, Xu L, et al. Ratopo: Improving lane topology reasoning via redundancy assignment. In: Proceedings of the 33rd ACM International Conference on Multimedia, 2025. 777–786