Graph-based Topology Reasoning for Driving Scenes

Anonymous authors
Paper under double-blind review

Abstract

Understanding the road genome is essential to realize autonomous driving. This highly intelligent problem contains two aspects - the connection relationship of lanes, and the assignment relationship between lanes and traffic elements, where a comprehensive topology reasoning method is vacant. On one hand, previous map learning techniques struggle in deriving lane connectivity with segmentation or laneline paradigms; or prior lane topology-oriented approaches focus on centerline detection and neglect the interaction modeling. On the other hand, the traffic element to lane assignment problem is limited in the image domain while the mapping task lies in 3D space, leaving the construction of correspondence between image and 3D views as an unexplored challenge. To address these issues, we present **TopoNet**, the first end-to-end framework capable of abstracting traffic knowledge beyond conventional perception tasks. To capture the driving scene topology, we introduce three key designs: (1) an embedding module to incorporate semantic knowledge from 2D elements into a unified feature space; (2) a curated scene graph neural network to model relationships and enable feature interaction inside the network; (3) instead of transmitting messages arbitrarily, a scene knowledge graph is devised to differentiate prior knowledge from various types of the road genome. We evaluate TopoNet on the challenging scene understanding benchmark, OpenLane-V2, where our approach outperforms all previous works by a great margin on all perceptual and topological metrics. The code will be released.

1 Introduction

Imagine that an autonomous vehicle is navigating towards a complex intersection and planning to go straight: it is wondering which one of the lanes in front to drive into and which traffic signal to follow. This high-level intellectual problem requires the agent not only to perceive lane position accurately, but also to understand the topology relationship from sensor inputs. Specifically, the map topology in driving scene includes: (1) the **lane topology graph** comprising centerlines as well as their connectivity, (2) and the **assignment relationship** between lanes and traffic elements (e.g., traffic lights, traffic boards, and road markers). As illustrated in Fig. 1, they altogether build a topological structure that provides explicit navigation signals for downstream tasks such as motion prediction and planning (Bansal et al., 2018; Chai et al., 2020).

Conventional autonomous driving datasets (Caesar et al., 2020; Wilson et al., 2021) include lane topology implicitly in the High-Definition (HD) map, which is designed for map storage but not being learned by neural networks. Various formulations are proposed to serve as substitutes to HD maps, such as 2D and 3D laneline detection (Pan et al., 2018; Garnett et al., 2019; Guo et al., 2020; Tabelini et al., 2021; Chen et al., 2022), bird's-eye-view (BEV) map element detection by segmentation (Pan et al., 2020; Roddick & Cipolla, 2020; Li et al., 2022a; Xu et al., 2023), and vectorization (Liu et al., 2023a; Liao et al., 2023a;b). To derive lane connectivity, a "tabula-rasa" resolution is to directly average two neighboring lanelines to get centerlines and then connect as a graph, based on the instance-wise laneline representations. Yet, it demands complicated hand-crafted rules and heavy post-processings. Another approach is to supervise the perception frameworks with relationship labels. Recent studies, STSU (Can et al., 2021) and TPLR (Can et al., 2022a), employ a Transformer-based architecture for lane instances and an additional MLP for connectivity. But still, they suffer from difficulty in finding useful information without explicit relationship modeling.

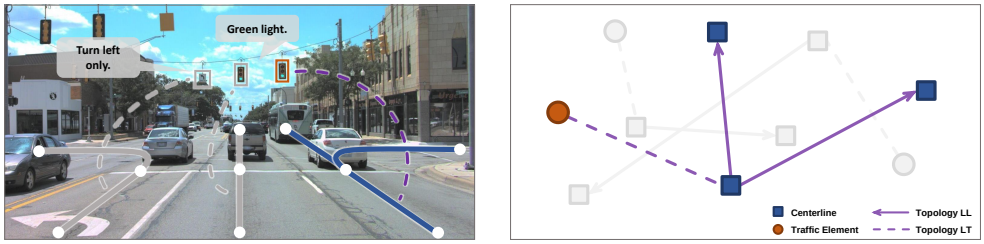

Figure 1: **Topology relationship of driving scenes.** While driving into an intersection, the self-driving vehicle has to reason about the correct lane and traffic information for downstream navigation. We advocate, and present TopoNet, to directly achieve topology understanding on the heterogeneous graph. "Topology LL" and "Topology LT" represent the relationship among lane centerlines and the relationship between lane centerlines and traffic elements respectively.

Moreover, the relationship assignment problem between traffic elements and lanes from sensor inputs remains mostly unexplored. Langenberg et al. (2019) tried to associate the ground truth of lanelines and traffic lights in the image domain (perspective view, PV). However, integrating traffic elements and lanes in a heterogenous graph (Fig. 1) is a different story. A reason is that traffic elements are described as bounding boxes in PV, while lanes are characterized as curves in 3D or BEV space. Meanwhile, spatial locations remain less important for traffic elements as their semantic meanings are essential, but positional clues of lanes are crucial for autonomous 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 two branches with a shared feature extractor, for traffic elements and centerlines respectively. Motivated by the Transformer-based detection algorithms (Carion et al., 2020; Zhu et al., 2020), we employ instance queries to extract local features via the deformable attention mechanism, which restricts the attention region and accelerates convergence. Since the clues for locating a specific centerline instance could be encoded in its neighbors and corresponding traffic elements, a Scene Graph Neural Network (SGNN) is devised to transmit messages 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 GNNs are developed based on categories of traffic elements and the centerline connectivity relationship (i.e., predecessor, ego, successor). Updated queries are ultimately decoded as the perception results and driving scene topology. With the proposed designs, we substantiate TopoNet on the large-scale topology reasoning benchmark for HD mapping, OpenLane-V2 (Wang et al., 2023). TopoNet outperforms state-of-the-art approaches by 15-84% for centerline perception, and achieves times of performance in terms of the challenging topology reasoning task. Ablations are conducted to verify the effectiveness of our framework.

2 Related Work

2.1 Lane Graph Learning

Lane Graph Learning has received abundant attention due to its pivotal role in autonomous driving. Prior works investigate generating road graphs (He et al., 2020; Bandara et al., 2022) or spatially denser lane graphs (Homayounfar et al., 2019; Zürn et al., 2021; He & Balakrishnan, 2022; Büchner et al., 2023) from aerial images. However, roads in aerial images are often occluded by trees and buildings, leading to inaccurate results. Recently, there has been a growing focus on producing lane graphs directly from vehicle-mounted sensor data. STSU (Can et al., 2021) proposes a DETR-like neural network to detect centerlines and then derive their connectivity by a successive MLP module. Based on STSU, Can et al. (2022a) introduce additional minimal cycle queries to ensure proper order of overlapping lines. CenterLineDet (Xu et al., 2023) regards centerlines as vertices and designs a graph-updating model trained by imitation learning. LaneGAP (Liao et al., 2024) proposes a path-wise modeling to represent the lane graph. It is also worth noticing that Tesla proposes the "language of lanes" to represent the lane graph as a sentence (Tesla, 2022). The attention-based model recursively predicts lane tokens and their connectivity. In this work, we focus on

explicitly modeling the centerline connectivity inside the network to enhance feature learning and indulging traffic elements in constructing the full driving scene graph.

2.2 HD Map Perception

With the trending popularity of BEV perception (Philion & Fidler, 2020; Li et al., 2022b; Zhou & Krähenbühl, 2022; Hu et al., 2023; Gao et al., 2023; Liao et al., 2023b), recent works focus on learning HD Maps with segmentation and vectorized methods. Map segmentation aims at predicting the semantic meaning of each BEV grid, such as lanelines, pedestrian crossings, and drivable areas. These works differentiate from each other mainly in the perspective view to BEV transform module, i.e., IPM-based (Xie et al., 2022; Can et al., 2022b), depth-based (Hu et al., 2022; Liu et al., 2023b), or Transformer-based (Li et al., 2022b; Jiang et al., 2023). Though dense segmentation provides pixel-level information, it cannot touch down the complex relationship of overlapping elements. Li et al. (2022a) handles the problem by grouping and vectorizing the segmented map with complicated post-processings. VectorMapNet (Liu et al., 2023a) proposes to directly represent each map element as a sequence of points, which uses coarse key points to decode laneline locations sequentially. MapTR (Liao et al., 2023a) further explores a unified permutation-based modeling approach for the sequence of points to eliminate the modeling ambiguity and improve performance and efficiency. In fact, since vectorization also enriches the direction information for lanelines, vectorization-based methods could be easily adapted to centerline perception by alternating the supervision. Recently, InstaGraM (Shin et al., 2023) constructs map elements as a graph by predicting vertices first and then utilizing a GNN module to detect edges. Its GNN produces all vertex features simultaneously, leading to the lack of instance-level interaction. Contrary to the aforementioned approaches, we leverage instance-wise feature transmission with a graph neural network, to extract prominent prediction hints from other elements in the topology graph.

2.3 Driving Scene Understanding

Driving Scene Understanding mainly indicates summarizing positional relationships of elements in outdoor environments beyond perception (Tian et al., 2020; Mylavarapu et al., 2020b; Zipfl & Zöllner, 2022; Malawade et al., 2022a). Previous works focus on utilizing the relationships of 2D bounding boxes for motion prediction (Li et al., 2020; Mylavarapu et al., 2020a;b; Fang et al., 2023) and risk assessment (Yu et al., 2021; Malawade et al., 2022b). In the industrial context, Mobileye presents an optimization-based method to automatically construct lane topology and traffic light-to-lane relationships based on their internal data (Mobileye, 2022). In the academy, Langenberg et al. (2019) address the traffic light to lane assignment (TL2LA) problem with a convolutional network by taking heterogeneous metadata as additional inputs. In contrast, TopoNet takes RGB images only and additionally reasons about the topology for lane entities besides TL2LA. We instantiate TopoNet on the large-scale driving scene understanding benchmark, which covers complicated urban scenarios.

2.4 Graph Neural Network

Graph Neural Network and its variants, such as graph convolutional network (GCN) (Kipf & Welling, 2017), GraphSAGE (Hamilton et al., 2017), and GAT (Veličković et al., 2018), are widely adopted to aggregate features of vertices and extract information from graph (Scarselli et al., 2008). Witnessing the impressive achievements of GNN in various fields (e.g., recommendation system and video understanding) (Guo & Wang, 2020; Mohamed et al., 2020; Chang et al., 2021; Pradhyumna & Shreya, 2021), researchers in the autonomous driving community attempt to utilize it to process unstructured data. Weng et al. (2020; 2021) introduce GNN to capture interactions among agents for 3D multi-object tracking. LaneGCN (Liang et al., 2020) constructs a lane graph from HD map, while others (Jia et al., 2022; 2023; Fang et al., 2023) model the relationship of moving agents and lanelines as a graph to improve the trajectory forecasting performance. Inspired by prior works, we design a GNN for the driving scene understanding task to enhance feature interaction and introduce a class-specific knowledge graph to better incorporate semantic information.

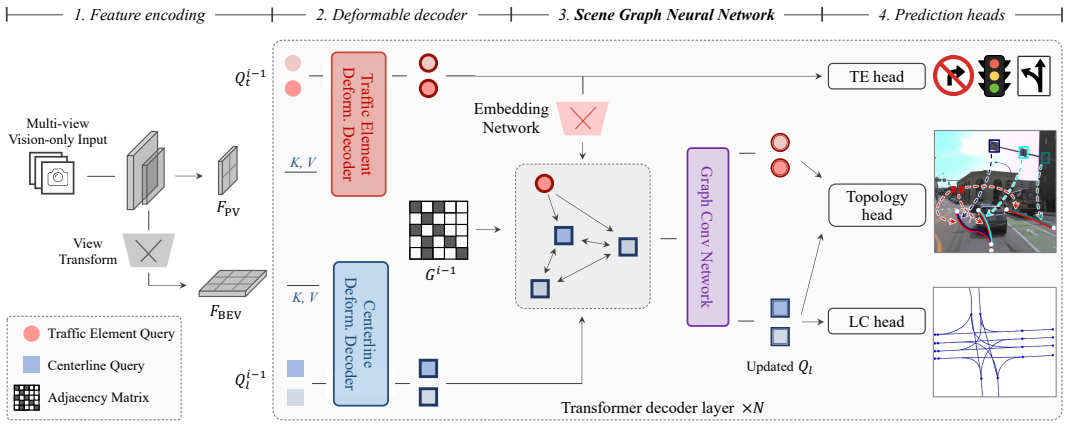

Figure 2: **Systematic diagram of TopoNet**. TopoNet addresses the crucial problem of topology reasoning for driving scenes in an end-to-end fashion. It consists of four stages, with the last three being compacted in a Transformer decoder architecture. TopoNet handles traffic elements and centerlines as two parallel branches at the Deformable decoder. Various types of instance queries (red, blue) 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. We abbreviate traffic elements and lane centerlines as "TE" and "LC" in this paper, respectively.

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 centerline is described as an ordered list of points. We denote it as $v_l = [p_0, ..., p_{n-1}]$, 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 respectively. Traffic elements 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 entities, the topology relationships are built. The connectivity of directed lanes establishes a map-like network on which vehicles can drive and is denoted as the lane graph (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 $(V_l \cup V_t, E_{lt})$ describes the correspondence between lanes and traffic elements. It can be seen as a bipartite graph that positive edges only exist between V_l and V_t. Both edge sets are required to be predicted in the task of topology reasoning.

3.2 Overview

Fig. 2 illustrates the overall architecture of the proposed **TopoNet**. Given multi-view images as input, the feature extractor generates multi-scale image features, including the front-view feature F_{PV}, and then convert them into a BEV feature F_{BEV} through a view transform module. Two independent decoders with the same deformable attention architecture (Zhu et al., 2020) consume F_{PV} and F_{BEV} to produce instance-level embeddings Q_t and Q_l separately. The proposed **Scene Graph Neural Network (SGNN)** then refines centerline queries Q_l in positional and topological aspects. Note that the decoder and SGNN layers are stacked iteratively to obtain local and global features in a sequential fashion. Finally, the task-specific heads take the refined queries to produce prediction 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 conventional perception works (Carion et al., 2020; Wu et al., 2022). However, being discrimi-

native is not enough to recognize correct topology relationships. The reason is that it takes a pair of instance queries as input to determine their relationship, in which feature embeddings are actually not independent. Meanwhile, adopting the local feature aggregation scheme of point-wise queries (Liu et al., 2023a; Liao et al., 2023a) for centerline perception is inadequate. Specifically, a key difference between centerlines and physical map elements is that centerlines naturally encode lane topology and traffic rules, which cannot be inferred from local 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 has several designs and merits compared to previous works. (1) It adopts an embedding network to extract TE knowledge within a unified feature space. (2) It models all entities in a frame as vertices in a graph, and strengthens interconnection among perceived instances to learn their inherent relationships with a graph neural network. (3) Alongside the graph structure, SGNN incorporates prior topology knowledge with a scene knowledge graph.

3.3.1 Embedding Network

As traffic elements are labeled on the perspective view, it is hard to harness their positional features in the spatial feature space. However, their semantic meaning imposes a great effect. For instance, a road sign indicating the prohibition of 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 that $\widetilde{Q}_t^i = \texttt{embedding}^i\big(Q_t^i\big)$, where i denotes the i-th decoder layer. Note that the queries $\widetilde{Q}_t^i$ remain intact in the SGNN. This is intended since imagining traffic elements from centerlines is relatively challenging. Besides, noting that the traffic element features are filtered and transformed into the spatial feature space by the embedding network, if these features are subsequently updated with adequate feature interactions with lane centerlines, they become unsuitable for predicting their attributes in the image feature space.

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 can be conveniently formulated as edges in a graph where entities are seen as vertices, while it is nontrivial in an open world without any explicit constraint. As there is no prior knowledge of topology structure, a trivial way is to construct a fully connected graph (V, E), where $V = V_l \cup V_t$ and $E \subseteq V \times V$. This inevitably increases computational cost and introduces unnecessary information transmission, such as between two traffic elements that are placed subjectively by humans. Instead, we form two directed graphs to propagate features, namely $G_{ll} = (V_l, V_l \times V_l)$ for lane graph estimation and $G_{lt} = (V_l \cup V_t, V_l \times V_t)$ representing the TE to LC assignments.

In graph G_{ll} and G_{lt}, lane queries Q_l are refined by the connected neighbors and corresponding traffic elements. Due to the fact that Q_l and Q_t represent different objects, the semantic gap still exists. We introduce an adapter layer to combine this heterogeneous information into the information gain denoted as R. The overall process in an SGNN layer can be formulated as follows:

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

3.3.3 Vanilla Scene Graph

Given the adjacency matrix A_{ll}^{i-1}, which is a representation of neighboring relationships in graph G_{ll}^{i-1} from the previous layer, we construct a weight matrix T_{ll}^i to control the flow of messages in the graph. In the directed graph, messages are passed in a single direction, e.g., from a centerline to its successor. However, as the structure of lanes depends on each other, the position of a lane is a good indication of the locations

of its neighbors. Thus, we supplement A_{ll}^{i-1} with a backward adjacency matrix to allow message exchange for two connected centerlines. The matrix T_{ll}^{i} of the i-th layer is calculated by:

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

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 as follows:

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

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

After obtaining the weight matrices, SGNN utilizes the graph convolutional layer (GCN) (Kipf & Welling, 2017) to perform feature propagation among queries:

$$\begin{aligned} Q_{l}^{i'} &= \texttt{GCN}_{ll}^{i}(Q_{l}^{i}, T_{ll}^{i}), \\ Q_{l}^{i''} &= \texttt{GCN}_{lt}^{i}(Q_{l}^{i}, \widetilde{Q}_{t}^{i}, T_{lt}^{i}). \end{aligned} \tag{4}$$

3.3.4 Scene Knowledge Graph

Though GCN enables feature propagation in the built graphs and treats nodes differently based on their connectivity, the semantic meaning of vertices remains unexplored. For example, the information from a traffic element indicating to go straight is not equally important to a red light. To address the issue and incorporate categorical prior, we design the scene knowledge graph to treat vertices in different classes differently. Fig. 3 illustrates an example process of updating a centerline query LC_1 on the given knowledge graph.

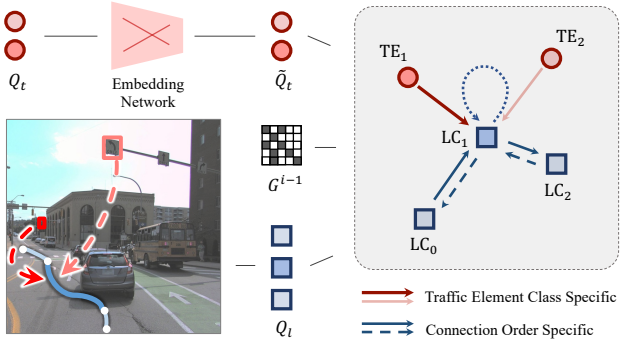

Figure 3: **Scene knowledge graph** illustration. For the centerline colored **blue** in the left case, related weight matrices in the graph are categorically independent. Different traffic elements and lane-directed connections bring different information to the centerline, which is encoded as a scene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

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:

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

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 of traffic element queries.

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. To this end, 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:

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

3.4 Learning

We employ multiple losses to train TopoNet in an end-to-end manner. As depicted in Fig. 2, all heads consume queries to provide perception and reasoning results. Nevertheless, they are not entirely independent, as the topology head requires matching results from perception heads. Similar to Transfomer-based networks (Carion et al., 2020; Zhu et al., 2020), the supervision is applied on each decoder layer to optimize the query feature iteratively. The overall loss of the proposed model is $\mathcal{L} = \mathcal{L}_{det_{\text{TE}}} + \mathcal{L}_{det_{\text{LC}}} + \mathcal{L}_{top}$.

Perception. Following the head design in DETR (Carion et al., 2020), the TE head predicts 2D bounding boxes with classification scores. Note that for predicting traffic elements, we take Q_t instead of $\widetilde{Q}_t$ to preserve their positional information in the perspective view. 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 of centerlines is normalized based on the predefined BEV range. For both heads, the Hungarian algorithm is utilized to generate matchings between ground truth and predictions, with the matching cost the same as the loss function. Then task-specific losses $\mathcal{L}_{det_{\text{TE}}}$ and $\mathcal{L}_{det_{\text{LC}}}$ are applied accordingly. Specifically, for the TE head, we employ the Focal loss (Lin et al., 2017b) for classification, an L1 regression loss, and an IOU loss for localization. Meanwhile, for centerlines, we use Focal loss and L1 loss as the classification and regression loss, respectively.

Reasoning. The topology head reasons pairwise relationships on the given embeddings. Similar to STSU (Can et al., 2021), 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. Different from the TE head, we adopt embeddings from 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 a ResNet-50 (He et al., 2016), which is pre-trained on ImageNet (Deng et al., 2009), with an FPN (Lin et al., 2017a) to obtain multi-scale image features. Following previous works (Zhu et al., 2020; Li et al., 2022b), the output features are from stage $S_{8\times}$, $S_{16\times}$ and $S_{32\times}$ of ResNet-50, where the subscripts $n\times$ indicates the downsampling factor. In the FPN module, the features are transformed into a four-level output with an additional $S_{64\times}$ level. The number of output channels of each level is set to 256. Then we adopt a simplified view transformer with 3 encoder layers proposed in BEVFormer (Li et al., 2022b). 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 (Zhu et al., 2020) layer. The size of BEV grids is set to 200×100, with four different height levels of $\{-1.5m, -0.5m, +0.5m, +1.5m\}$ relative to the ground.

Deformable Decoder. For the decoder, we utilize the decoder layer in Deformable DETR (Zhu et al., 2020) that each decoder layer contains three layers: a self-attention layer with 8 attention heads, a deformable attention layer with 8 attention heads and 4 offset points, and a two-layer feed-forward network with 512 channels in the middle. After each operation, a dropout layer with a ratio of 0.1 and a layer normalization is applied. 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 Graph Convolutional Network (GCN) (Kipf & Welling, 2017) 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 branch, 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. In between the MLP, a ReLU activation function and a dropout layer are included. β_{ll} and β_{lt} are set to 0.6.

Prediction Heads. The prediction head for perception comprises a classification head and a regression head. For the traffic element branch, the classification head is a single-layer MLP, which outputs the sigmoid probability of each class. The regression head is a three-layer MLP with ReLU, which predicts the normalized coordinates of 2D bounding boxes in the form of $\{cx, cy, width, height\}$. For centerline, the classification head consists of a three-layer MLP with LayerNorm and ReLU in between, which predicts the confidence score. The regression head is a three-layer MLP with ReLU, which predicts the normalized point set of 11×3 for a centerline. To predict topology relationships, relationship heads are applied. Given the instance queries $\widetilde{Q}_a$ and $\widetilde{Q}_b$ with 256 feature channels, the topology head first applies a three-layer MLP:

$$\widetilde{Q}'_a = \texttt{MLP}_\texttt{a}(\widetilde{Q}_a),\ \widetilde{Q}'_b = \texttt{MLP}_\texttt{b}(\widetilde{Q}_b), \tag{8}$$

where the number of output channels is 128. For each pair of queries $\tilde{q}'_a \in \widetilde{Q}'_a$ and $\tilde{q}'_b \in \widetilde{Q}'_b$, the output is the confidence of the relationship, with independent `MLP`s for different types of relationships:

$$conf. = \texttt{sigmoid}\Big(\texttt{MLP}_\texttt{top}\big(\texttt{concat}(\tilde{q}'_a, \tilde{q}'_b)\big)\Big). \tag{9}$$

Loss. $\mathcal{L}_{det_{\text{TE}}}$ includes a classification, a regression, and an IoU loss that $\mathcal{L}_{det_{\text{TE}}} = \lambda_{cls} \cdot \mathcal{L}_{cls} + \lambda_{reg} \cdot \mathcal{L}_{reg} + \lambda_{iou} \cdot \mathcal{L}_{iou}$. λ_{cls}, λ_{reg}, and λ_{iou} are set to 1.0, 2.5, and 1.0, respectively. The classification loss $\mathcal{L}_{cls}$ is a Focal loss. Note that the regression loss $\mathcal{L}_{reg}$ is an L1 Loss calculated on a normalized format of $\{cx, cy, width, height\}$, while the IoU loss $\mathcal{L}_{iou}$ is a GIoU loss computed on the denormalized coordinates. For centerline detection, $\mathcal{L}_{det_{\text{LC}}}$ comprises a classification and a regression loss that $\mathcal{L}_{det_{\text{LC}}} = \lambda_{cls} \cdot \mathcal{L}_{cls} + \lambda_{reg} \cdot \mathcal{L}_{reg}$, 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. The resolution of input images is 2048 $\times$ 1550, except for the front-view image, which is 1550 $\times$ 2048 and cropped into 1550 $\times$ 1550. For data augmentation, $\times$0.5 resizing and color jitter are used. We adopt the AdamW optimizer (Loshchilov & Hutter, 2018) and a cosine annealing schedule with an initial learning rate of 1×10^{-4}. TopoNet is trained for 24 epochs with a batch size of 8 with 8 Tesla A100 GPUs.

4.2 Dataset and Metrics

We conduct experiments on the OpenLane-V2 benchmark (Wang et al., 2023). The dataset contains topological structures in the driving scenes, and raises huge challenges for algorithms to perceive and reason about the environment accurately. Ablation studies are conducted on the *subset_A* of OpenLane-V2.

Dataset. Built on top of the Argoverse 2 (Wilson et al., 2021) and nuScenes (Caesar et al., 2020) datasets, the OpenLane-V2 benchmark includes images from 2,000 scenes collected worldwide under different environments. The dataset is split into two subsets, namely *subset_A* and *subset_B*. Each subset contains 1,000 scenes with multi-view images and annotations at $2Hz$. All lanes within $[-50m, +50m]$ along the x-axis and $[-25m, +25m]$ along the y-axis are annotated in the 3D space. Centerlines are described in the form of lists of points. Each list is ordered and comprises 201 points in 3D space. Statistically, about 90% of frames have more than 10 centerlines while about 10% have more than 40. Traffic elements follow the typical labeling style in 2D detection that objects are labeled as 2D bounding boxes on the front-view images. Each element is denoted as a 2D bounding box on the front view image, with its attribute. There are 13 types of attributes, including *unknown*, *red*, *green*, *yellow*, *go_straight*, *turn_left*, *turn_right*, *no_left_turn*, *no_right_turn*, *u_turn*, *no_u_turn*, *slight_left*, and *slight_right*. The topology relationships are provided in the form of adjacency matrices based on the ordering of centerlines and traffic elements. In the adjacency matrices, an entry (i, j) is positive (i.e., 1) if and only if the elements at i and j are connected.

Perception Metrics. The DET score is the typical mean average precision (mAP) for measuring instance-level perception performance. Based on the Fréchet distances (Eiter & Mannila, 1994), the DET_l score is

Table 1: **Comparison with state-of-the-art methods** on the OpenLane-V2 benchmark. TopoNet outperforms all previous works by a wide margin, especially in directed centerline perception and topology reasoning. *: Topology reasoning evaluation is based on matching results on Chamfer distance. The highest score is bolded, 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 (Can et al., 2021)	12.7	0.5	43.0	<u>15.1</u>	25.4
	VectorMapNet (Liu et al., 2023a)	11.1	0.4	41.7	5.9	20.8
	MapTR (Liao et al., 2023a)	8.3	0.2	<u>43.5</u>	5.9	20.0
	MapTR* (Liao et al., 2023a)	<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 (Can et al., 2021)	8.2	0.0	43.9	<u>9.4</u>	21.2
	VectorMapNet (Liu et al., 2023a)	3.5	0.0	49.1	1.4	16.3
	MapTR (Liao et al., 2023a)	8.3	0.1	<u>54.0</u>	3.7	21.1
	MapTR* (Liao et al., 2023a)	<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**

averaged over match thresholds of $\mathbb{T} = \{1.0, 2.0, 3.0\}$:

$$\text{DET}_l = \frac{1}{|\mathbb{T}|} \sum_{t \in \mathbb{T}} AP_t. \tag{10}$$

Note that the defined BEV range is relatively large compared to other lane detection datasets, so 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. The DET_t uses IoU as the similarity measure and is averaged over different attributes $\mathbb{A}$ of traffic elements:

$$\text{DET}_t = \frac{1}{|\mathbb{A}|} \sum_{a \in \mathbb{A}} AP_a. \tag{11}$$

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 projection on the vertices 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 vertice mAP between (V, E) and $(\hat{V}', \hat{E}')$ over all vertices:

$$\text{TOP} = \frac{1}{|V|} \sum_{v \in V} \frac{\sum_{\hat{n}' \in \hat{N}'(v)} P(\hat{n}') \mathbb{1}(\hat{n}' \in N(v))}{|N(v)|}, \tag{12}$$

where $N(v)$ denotes the ordered list of neighbors of vertex v ranked by confidence 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 primary task of the dataset is scene structure perception and reasoning, which requires the model to recognize the dynamic drivable states of lanes in the surrounding environment. The OpenLane-V2 Score (OLS) summarizes metrics covering different aspects of the primary task:

$$\text{OLS} = \frac{1}{4}\left[\text{DET}_l + \text{DET}_t + f(\text{TOP}_{ll}) + f(\text{TOP}_{lt})\right], \tag{13}$$

where f is the square root function.

4.3 Main Results

In Table 1, we compare the proposed TopoNet to several state-of-the-art methods, whose implementation details are described in Appendix A. TopoNet outperforms all previous algorithms by a large margin. As

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 (Can et al., 2021)	✓	14.2	0.6	13.8	**12.8**
VectorMapNet (Liu et al., 2023a)	✗	12.7	-	10.3	1.0
MapTR (Liao et al., 2023a)	✗	10.0	-	21.7	11.5
TopoNet (Ours)	✓	**27.7**	**4.6**	**27.4**	10.1

the SOTA map learning method MapTR ignores the direction of centerlines with the permutation-equivalent modeling (Liao et al., 2023a), we additionally evaluate MapTR based on Chamfer distance matching. However, its performance on DET_l, as well as topology metrics, significantly degenerates. The performance of centerline queries without directional information indicates that understanding the complex scenario and perceiving presented instances are two totally different stories. 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 is noticeable 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 the performance on LC-TE topology reasoning, the superiority of TopoNet can be attributed to the effectiveness of proposed SGNN module, in which different entities are modeled differently, as well as its overall superior centerline and traffic element detection performance.

Comparison on Centerline Perception. To have a fair comparison, we use a unified backbone architecture and PV-to-BEV transformation module for various SOTA methods on centerline perception task. We keep the topology supervision for STSU, as it is originally designed for detecting centerlines and their topology relationship. Since VectorMapNet and MapTR are for the task of 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 design choice.

To better align with previous works (Liu et al., 2023a; Liao et al., 2023a), 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 found that the original design of online mapping approaches struggle with managing lane topology and traffic elements. As shown in Table 1 and Table 2, when the affect 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 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 aligned input size 512× 676, our method has comparable online efficiency but higher performance.

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

Method	mIoU↑
HDMapNet (Li et al., 2022a)	18.3
STSU (Can et al., 2021)	24.6
VectorMapNet (Liu et al., 2023a)	18.9
MapTR (Liao et al., 2023a)	32.1
LaneGAP (Liao et al., 2024)	35.0
TopoNet (Ours)	**35.1**

Comparison on BEV Segmentation. DET_l is a rigorous and effective metric for evaluating the validity of each point on a single centerline, requiring a consistent instance representation of lanes. In contrast, the Intersection over Union (IoU) metric changes continuously, and insensitive to minor prediction variations. However, it facilitates the instant assessment of the overall geometric accuracy across different methods with varying formulation, such as HDMapNet (Li et al., 2022a) and LaneGAP (Liao et al., 2024). Except for HDMapNet, the vectorized centerline prediction of each method are rendered to BEV with a fixed line width

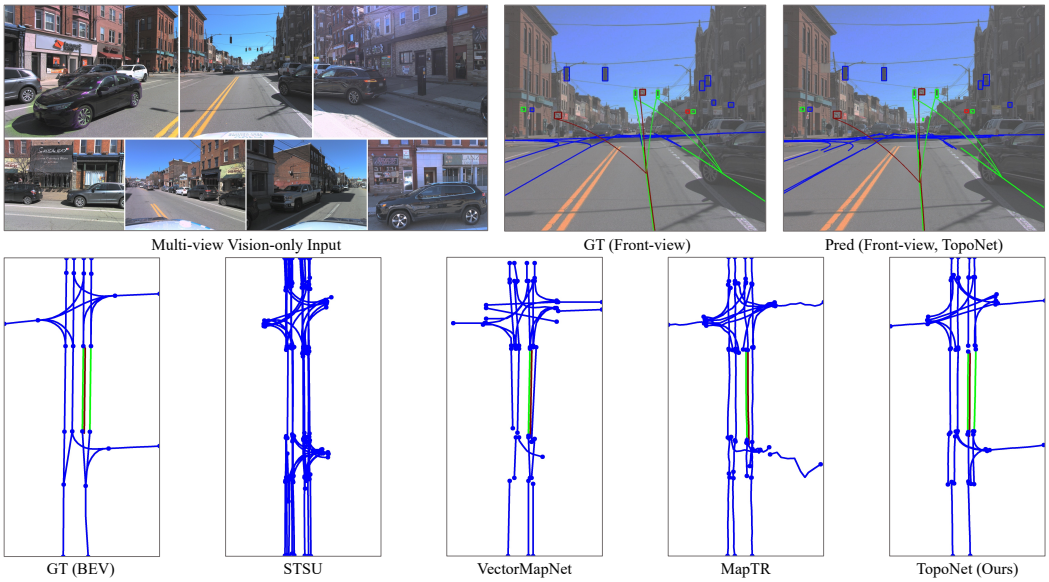

Figure 4: **Qualitative results** of TopoNet and other algorithms on *subset_A* of the OpenLane-V2 dataset. While driving in complex scenarios, TopoNet achieves superior lane graph prediction performance compared to other SOTA methods. It also successfully builds all connections between traffic elements and lanes (top right, and correspondingly colored lines in BEV). Colors denote categories of traffic elements.

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**

Table 5: **Ablation on feature propagation** in the SGNN. "LL only" denotes aggregation of 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**

of $0.75m$ 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 a concurrent work LaneGAP (Liao et al., 2024), which utilizes a path-wise modeling to represent 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 comparable 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. The concatenation and down-sampling operations, as well as the traffic element embedding, are also removed. As illustrated in Table 4, the proposed SKG outperforms models in other settings, demonstrating its effectiveness for topology understanding. Compared to the SG version, the scene knowledge graph provides an additional improvement of 0.8% for centerline perception, owning 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.

Table 6: **Ablation on the number of GNN layers** in the scene knowledge graph. Model performance drops as the number of SGNN layers increases.

# GNN	DET_l↑	TOP_{ll}↑	DET_t↑	TOP_{lt}↑	OLS↑
1	**28.5**	**4.1**	**48.1**	**20.8**	**35.6**
2	27.9	4.0	47.5	20.9	35.3
3	20.4	0.5	46.1	15.7	28.3

Table 7: **Ablation on edge weight** in the scene knowledge graph. The magnitude of edge weight has an impact on model performance.

Weight	DET_l↑	TOP_{ll}↑	DET_t↑	TOP_{lt}↑	OLS↑
0.5	28.4	4.0	47.7	**20.8**	35.4
0.6	**28.5**	**4.1**	**48.1**	**20.8**	**35.6**
0.7	27.3	**4.1**	47.7	20.7	35.1

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 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.

Effect on the Number of GNN Layers. Though GNN is beneficial for propagating features in the knowledge graph, raising the number of GNN layers leads to degenerated performance. As shown in Table 6, SGNN with a single GNN layer achieves the best performance. The reason is that a GNN layer increases the similarity of adjacent vertices. With multiple GNN layers, features of all vertices become less discriminative.

Effect on Edge Weight. Edge weight in the scene knowledge graph represents how much information is propagated through the SGNN layers. In Table 7, 0.6 corresponds to the most appropriate ratio.

4.5 Qualitative Analysis

We provide a qualitative comparison on *validation* set in Fig. 4. We present the raw output of each method, abstaining from the post-processing technique suggested in STSU (Can et al., 2021), to avoid the potential introduction of 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. More visualizations are provided in Appendix C.

5 Conclusion and Future Work

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

Limitations and Future Work. Due to the query-based design for feature interactions, TopoNet performs well in achieving most positive predictions, while post-processes such as merging or pruning are still needed to produce clean output as in lane topology works (Can et al., 2021; Büchner et al., 2023). How to incorporate the merging ability with auto-regressive or other association mechanisms deserves future exploration. Meanwhile, it will be interesting to see if more categories of traffic elements, and correspondingly more sophisticated knowledge graphs will make any advances.

References

Wele Gedara Chaminda Bandara, Jeya Maria Jose Valanarasu, and Vishal M Patel. SPIN Road Mapper: Extracting roads from aerial images via spatial and interaction space graph reasoning for autonomous driving. In *ICRA*, 2022. 2

Mayank Bansal, Alex Krizhevsky, and Abhijit Ogale. ChauffeurNet: Learning to drive by imitating the best and synthesizing the worst. *arXiv preprint arXiv:1812.03079*, 2018. 1

Martin Büchner, Jannik Zürn, Ion-George Todoran, Abhinav Valada, and Wolfram Burgard. Learning and aggregating lane graphs for urban automated driving. In *CVPR*, 2023. 2, 12

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020. 1, 8

Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Structured bird's-eye-view traffic scene understanding from onboard images. In *ICCV*, 2021. 1, 2, 7, 9, 10, 12, 17, 18

Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Topology preserving local road network estimation from single onboard camera image. In *CVPR*, 2022a. 1, 2

Yigit Baran Can, Alexander Liniger, Ozan Unal, Danda Paudel, and Luc Van Gool. Understanding bird's-eye view of road semantics using an onboard camera. *RA-L*, 2022b. 3

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 2, 4, 7

Yuning Chai, Benjamin Sapp, Mayank Bansal, and Dragomir Anguelov. MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction. In *CoRL*, 2020. 1

Xiaojun Chang, Pengzhen Ren, Pengfei Xu, Zhihui Li, Xiaojiang Chen, and Alex Hauptmann. A comprehensive survey of scene graphs: Generation and application. *PAMI*, 2021. 3

Li Chen, Chonghao Sima, Yang Li, Zehan Zheng, Jiajie Xu, Xiangwei Geng, Hongyang Li, Conghui He, Jianping Shi, Yu Qiao, and Junchi Yan. PersFormer: 3d lane detection via perspective transformer and the openlane benchmark. In *ECCV*, 2022. 1

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 7

Thomas Eiter and Heikki Mannila. Computing discrete fréchet distance. 1994. 8

Jianwu Fang, Chen Zhu, Pu Zhang, Hongkai Yu, and Jianru Xue. Heterogeneous trajectory forecasting via risk and scene graph learning. *TITS*, 2023. 3

Yulu Gao, Chonghao Sima, Shaoshuai Shi, Shangzhe Di, Si Liu, and Hongyang Li. Sparse dense fusion for 3d object detection. In *IROS*, 2023. 3

Noa Garnett, Rafi Cohen, Tomer Pe'er, Roee Lahav, and Dan Levi. 3D-LaneNet: End-to-end 3d multiple lane detection. In *ICCV*, 2019. 1

Yuliang Guo, Guang Chen, Peitao Zhao, Weide Zhang, Jinghao Miao, Jingao Wang, and Tae Eun Choe. Gen-LaneNet: A generalized and scalable approach for 3d lane detection. In *ECCV*, 2020. 1

Zhiwei Guo and Heng Wang. A deep graph neural network-based mechanism for social recommendations. *IEEE TII*, 2020. 3

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, 2017. 3

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

Songtao He and Hari Balakrishnan. Lane-level street map extraction from aerial imagery. In *WACV*, 2022. 2

Songtao He, Favyen Bastani, Satvat Jagwani, Mohammad Alizadeh, Hari Balakrishnan, Sanjay Chawla, Mohamed M Elshrif, Samuel Madden, and Mohammad Amin Sadeghi. Sat2Graph: Road graph extraction through graph-tensor encoding. In *ECCV*, 2020. 2

Namdar Homayounfar, Wei-Chiu Ma, Justin Liang, Xinyu Wu, Jack Fan, and Raquel Urtasun. DAGMapper: Learning to map by discovering lane topology. In *ICCV*, 2019. 2

Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. ST-P3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *ECCV*, 2022. 3

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, Lewei Lu, Xiaosong Jia, Qiang Liu, Jifeng Dai, Yu Qiao, and Hongyang Li. Planning-oriented autonomous driving. In *CVPR*, 2023. 3

Xiaosong Jia, Li Chen, Penghao Wu, Jia Zeng, Junchi Yan, Hongyang Li, and Yu Qiao. Towards capturing the temporal dynamics for trajectory prediction: a coarse-to-fine approach. In *CoRL*, 2022. 3

Xiaosong Jia, Penghao Wu, Li Chen, Hongyang Li, Yu Liu, and Junchi Yan. HDGT: Heterogeneous driving graph transformer for multi-agent trajectory prediction via scene encoding. *PAMI*, 2023. 3

Yanqin Jiang, Li Zhang, Zhenwei Miao, Xiatian Zhu, Jin Gao, Weiming Hu, and Yu-Gang Jiang. PolarFormer: Multi-camera 3d object detection with polar transformers. In *AAAI*, 2023. 3

M. Esat Kalfaoglu, Halil Ibrahim Ozturk, Ozsel Kilinc, and Alptekin Temizel. TopoMask: Instance-Mask-Based Formulation for the Road Topology Problem via Transformer-Based Architecture. https://opendrivelab.com/e2ead/AD23Challenge/Track_1_PlatypusWhisperers.pdf, 2023. 18

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017. 3, 6, 7

Tristan Langenberg, Timo Lüddecke, and Florentin Wörgötter. Deep metadata fusion for traffic light to lane assignment. *RA-L*, 2019. 2, 3

Chengxi Li, Yue Meng, Stanley H Chan, and Yi-Ting Chen. Learning 3d-aware egocentric spatial-temporal interaction via graph convolutional networks. In *ICRA*, 2020. 3

Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. HDMapNet: An online hd map construction and evaluation framework. In *ICRA*, 2022a. 1, 3, 10

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. BEVFormer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *ECCV*, 2022b. 3, 7

Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, 2020. 3

Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. MapTR: Structured modeling and learning for online vectorized HD map construction. In *ICLR*, 2023a. 1, 3, 5, 9, 10, 17

Bencheng Liao, Shaoyu Chen, Yunchi Zhang, Bo Jiang, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. MapTRv2: An end-to-end framework for online vectorized HD map construction. *arXiv preprint arXiv:2308.05736*, 2023b. 1, 3

Bencheng Liao, Shaoyu Chen, Bo Jiang, Tianheng Cheng, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Lane Graph as Path: Continuity-preserving path-wise modeling for online lane graph construction. In *ECCV*, 2024. 2, 10, 11

Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, 2017a. 7

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017b. 7

Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. VectorMapNet: End-to-end vectorized hd map learning. In *ICML*, 2023a. 1, 3, 5, 9, 10, 17

Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. BEV-Fusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *ICRA*, 2023b. 3

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2018. 8

Mingjie Lu, Yuanxian Huang, Ji Liu, Jinzhang Peng, Lu Tian, and Ashish Sirasao. Separated RoadTopoFormer. https://opendrivelab.com/e2ead/AD23Challenge/Track_1_Victory.pdf, 2023. 18

Arnav Vaibhav Malawade, Shih-Yuan Yu, Brandon Hsu, Harsimrat Kaeley, Anurag Karra, and Mohammad Abdullah Al Faruque. roadscene2vec: A tool for extracting and embedding road scene-graphs. *Knowledge-Based Systems*, 2022a. 3

Arnav Vaibhav Malawade, Shih-Yuan Yu, Brandon Hsu, Deepan Muthirayan, Pramod P Khargonekar, and Mohammad Abdullah Al Faruque. Spatiotemporal scene-graph embedding for autonomous vehicle collision prediction. *IoT-J*, 2022b. 3

Mobileye. Mobileye under the hood. https://www.mobileye.com/ces-2022/, 2022. 3

Abduallah Mohamed, Kun Qian, Mohamed Elhoseiny, and Christian Claudel. Social-STGCNN: A social spatio-temporal graph convolutional neural network for human trajectory prediction. In *CVPR*, 2020. 3

Sravan Mylavarapu, Mahtab Sandhu, Priyesh Vijayan, K Madhava Krishna, Balaraman Ravindran, and Anoop Namboodiri. Towards accurate vehicle behaviour classification with multi-relational graph convolutional networks. In *IV*, 2020a. 3

Sravan Mylavarapu, Mahtab Sandhu, Priyesh Vijayan, K Madhava Krishna, Balaraman Ravindran, and Anoop Namboodiri. Understanding dynamic scenes using graph convolution networks. In *IROS*, 2020b. 3

Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *RA-L*, 2020. 1

Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *AAAI*, 2018. 1

Jonah Philion and Sanja Fidler. Lift, Splat, Shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020. 3

P Pradhyumna and GP Shreya. Graph neural network (gnn) in image and video understanding using deep learning for computer vision applications. In *ICESC*, 2021. 3

Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *CVPR*, 2020. 1

Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. The graph neural network model. *IEEE TNNLS*, 2008. 3

Juyeb Shin, Francois Rameau, Hyeonjun Jeong, and Dongsuk Kum. InstaGraM: Instance-level graph modeling for vectorized hd map learning. *arXiv preprint arXiv:2301.04470*, 2023. 3

Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Keep your eyes on the lane: Real-time attention-guided lane detection. In *CVPR*, 2021. 1

Tesla. Tesla AI Day. https://www.youtube.com/watch?v=ODSJsviD_SU, 2022. 2

Yafu Tian, Alexander Carballo, Ruifeng Li, and Kazuya Takeda. Road Scene Graph: A semantic graph-based scene representation dataset for intelligent vehicles. *arXiv preprint arXiv:2011.13588*, 2020. 3

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018. 3

Huijie Wang, Tianyu Li, Yang Li, Li Chen, Chonghao Sima, Zhenbo Liu, Bangjun Wang, Peijin Jia, Yuting Wang, Shengyin Jiang, Feng Wen, Hang Xu, Ping Luo, Junchi Yan, Wei Zhang, and Hongyang Li. OpenLane-V2: A topology reasoning benchmark for unified 3d hd mapping. In *NeurIPS Datasets and Benchmarks*, 2023. 2, 8

Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M Kitani. GNN3DMOT: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In *CVPR*, 2020. 3

Xinshuo Weng, Ye Yuan, and Kris Kitani. PTP: Parallelized tracking and prediction with graph neural networks and diversity sampling. *RA-L*, 2021. 3

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *NeurIPS Datasets and Benchmarks*, 2021. 1, 8

Dongming Wu, Fan Jia, Jiahao Chang, Zhuoling Li, Jianjian Sun, Chunrui Han, Shuailin Li, Yingfei Liu, Zheng Ge, and Tiancai Wang. The 1st-place Solution for CVPR 2023 OpenLane Topology in Autonomous Driving Challenge. https://opendrivelab.com/e2ead/AD23Challenge/Track_1_MFV.pdf, 2023. 18

Junfeng Wu, Yi Jiang, Song Bai, Wenqing Zhang, and Xiang Bai. SeqFormer: Sequential transformer for video instance segmentation. In *ECCV*, 2022. 4

Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M Alvarez. M^2BEV: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. *arXiv preprint arXiv:2204.05088*, 2022. 3

Zhenhua Xu, Yuxuan Liu, Yuxiang Sun, Ming Liu, and Lujia Wang. CenterLineDet: Road lane centerline graph detection with vehicle-mounted sensors by transformer for high-definition map creation. In *ICRA*, 2023. 1, 2

Shih-Yuan Yu, Arnav Vaibhav Malawade, Deepan Muthirayan, Pramod P Khargonekar, and Mohammad Abdullah Al Faruque. Scene-graph augmented data-driven risk assessment of autonomous vehicle decisions. *TITS*, 2021. 3

Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *CVPR*, 2022. 3

Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable DETR: Deformable transformers for end-to-end object detection. In *ICLR*, 2020. 2, 4, 7

Maximilian Zipfl and J Marius Zöllner. Towards traffic scene description: The semantic scene graph. In *ITSC*, 2022. 3

Jannik Zürn, Johan Vertens, and Wolfram Burgard. Lane graph estimation for scene understanding in urban driving. *RA-L*, 2021. 2

Appendix

A Re-implementation of SOTA Methods

Since there are no prior methods for the task of driving scene understanding, we adapt three state-of-the-art algorithms which are initially designed for lane graph estimation or map learning: STSU (Can et al., 2021), VectorMapNet (Liu et al., 2023a), and MapTR (Liao et al., 2023a). To ensure a fair comparison, we employed the same input resolution, the same ResNet-50 image backbone, and the same FPN neck for extracting features from surrounding images. Additionally, we incorporated a Deformable DETR head specifically for traffic elements, aligning all settings with TopoNet. As for topology reasoning, we treat it differently based on their own modeling concepts of instance query. The topology heads for each methods are the same MLPs as in TopoNet. All the methods are trained for 24 epochs to ensure a fair comparison.

STSU. The original model predicts centerlines and their relationships under the monocular setting. It employs a BEV positional embedding and a DETR head to predict three Bezier control points for each centerline, and uses object queries in the decoder to predict the connectivity of centerlines. To adapt to the multi-view inputs, we re-implement STSU by computing and concatenate the BEV embedding of images from different views. The concatenated embedding is then fed into the DETR encoder. We retain the original DETR decoder to predict the Bezier control points, which are interpolated into 11 equidistant points as outputs. The lane-lane relationship prediction head of STSU is preserved as well.

VectorMapNet utilizes a DETR-like decoder to estimate key points and an auto-regressive module to generate detailed graphical information for a map elements instance, such as lanelines and pedestrian crossings. We supervise VectorMapNet's decoder with centerline labels to adapt with OpenLane-V2 task. The perception range is defined as $\pm 30m \times \pm 15m$ in the original setting, and we expand it to $\pm 50m \times \pm 25m$. The centerline outputs of VectorMapNet are interpolated to 11 equidistant points during the prediction process. For topology prediction, we use the key point object queries in the VectorMapNet decoder as instance queries of centerlines. We implement the modification on the given codebase of VectorMapNet while retaining other settings. However, due to their lack of support for 3D centerlines, we only predict 2D centerlines in the BEV space and ignore the height dimension during training and evaluation.

MapTR. MapTR directly predicts polylines with a fixed number of points using a DETR-like decoder. It utilizes a hierarchical query, representing each line instance with multiple point queries and one instance query. For topology prediction, we use the average of the hierarchical queries of an instance in the MapTR decoder as the instance query of a centerline. The traffic element head and the topology head are with the same setting as in TopoNet. We align the original backbone setting with TopoNet. The perception region is also expanded to $\pm 50m \times \pm 25m$. The implementation is also done on the open-source codebase of MapTR with other settings retained. Due to the lack of support for 3D centerlines, we only predict 2D centerlines in BEV and ignore the height dimension during training and evaluation.

B More Experiments

Table 8: **Ablation on traffic element embedding.** TE embedding is necessary to deal with inconsistency in the feature space of different queries.

Method	$\text{DET}_l\uparrow$	$\text{TOP}_{ll}\uparrow$	$\text{DET}_t\uparrow$	$\text{TOP}_{lt}\uparrow$	OLS↑
w/o embedding	28.4	**4.1**	46.9	20.5	35.2
TopoNet	**28.5**	**4.1**	**48.1**	**20.8**	**35.6**

Effect on Traffic Element Embedding. In the "w/o embedding" setting, we remove the traffic element embedding network and use Q_t as the input of SGNN directly. As shown in Table 8, removing the embedding results in a 1.2% performance drop in traffic element recognition. The reason is that TE queries contain a large amount of spatial information in the PV space due to the 2D detection supervision signals,

Table 9: **Comparison with the awarded methods in the CVPR 2023 Autonomous Driving Challenge**. The upper part is the Leaderboard on the OpenLane-V2 *test* split. The down part is the performance on the *val* split with ResNet-50 backbones. The listed teams utilized non-shared backbones for the lane and traffic element branches. "# Params." refers to the total number of backbone parameters. "*": using post-processing on the topology prediction. TopoNet surpasses third-place method on the overall performance, with only 25M backbone parameters and 24 epoch training.

Data	Team & Method	Backbone	# Params.	Epoch	$\text{DET}_l\uparrow$	$\text{TOP}_{ll}\uparrow$	$\text{DET}_t\uparrow$	$\text{TOP}_{lt}\uparrow$	OLS↑
test	MFV (Wu et al., 2023) (1st)	ViT-L + CSPNet (YOLOv8x)	375M	48 + 20	35.8	22.5*	79.7	33.5*	55.2
	Victory (Lu et al., 2023) (2nd)	Swin-S + Swin-S	100M	Unknown	21.8	13.2*	72.5	22.6	44.6
	PlatypusWhispers (Kalfaoglu et al., 2023) (3rd)	RegNetY-800mf + ConvNext-B	95M	40 + 30	22.1	6.0	70.6	15.7	39.2
	TopoNet (Ours)	**ResNet-50 (shared)**	**25M**	**24**	25.8	10.1*	59.5	23.7*	41.4
val	MFV (Wu et al., 2023) (1st)	ResNet-50 (LC only)	25M	20	18.2	-	-	-	-
	PlatypusWhispers (Kalfaoglu et al., 2023) (3rd)	ResNet-50 + ResNet-50	50M	24 + 24	22.1	5.8	58.2	15.5	36.0
	TopoNet (Ours)	**ResNet-50 (shared)**	**25M**	**24**	**28.5**	**4.1**	48.1	**20.8**	35.6

resulting in significant inconsistencies in the feature spaces. In all, the experiments demonstrate that TE embedding effectively filters out irrelevant spatial information and extracts high-level semantic knowledge to help centerline detection and lane topology reasoning.

Comparison on the OpenLane-V2 leaderboard methods. We compare TopoNet with the awarded methods in the CVPR 2023 Autonomous Driving Challenge in Table 9. The leading methods of the competition employed various tricks to maximize the performance, such as stronger and non-shared backbones, longer training epochs, training on the validation set, extensive hyper-parameter tuning, and complex data augmentation and post-processing strategies. Because most methods in the competition employ SOTA 2D detection approaches and non-shared backbone, we primarily compare the effectiveness of TopoNet in the context of lane graph perception. After utilizing the post-processing technique of MFV (Wu et al., 2023) on lane-lane topology prediction, TopoNet achieves a DET_l score of 25.8 and a TOP_{ll} of 10.1 on the OpenLane-V2 *test* set, achieving superior centerline detection performance compared to the second-place method. TopoNet employs a shared ResNet-50 backbone, being up to 15× smaller in backbone parameter size than the awarded methods, demonstrating great training efficiency.

We further provide the comparison on the *validation* split, where these methods report performance with a ResNet-50 backbone and without most tricks. MFV, the first-place team in the competition, achieves a DET_l score of 18.2, and the third-place team PlatypusWhisperers (Kalfaoglu et al., 2023) gets a DET_l score of 22.1. With less data augmentation and hyper-parameter tuning, TopoNet achieves a much higher DET_l score of 28.5, surpassing all methods above. These fair comparisons on the validation set well demonstrate the effectiveness of TopoNet's pipeline.

C More Visualization

We provide additional qualitative comparisons on *subset_B* of OpenLane-V2 in Fig. 5. We present the raw output of each method, abstaining from the post-processing technique suggested in STSU (Can et al., 2021), to avoid the potential introduction of 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.

Fig. 6 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. A large-scale dataset and learning techniques, such as active learning, would solve such failure cases in a real-world deployment.

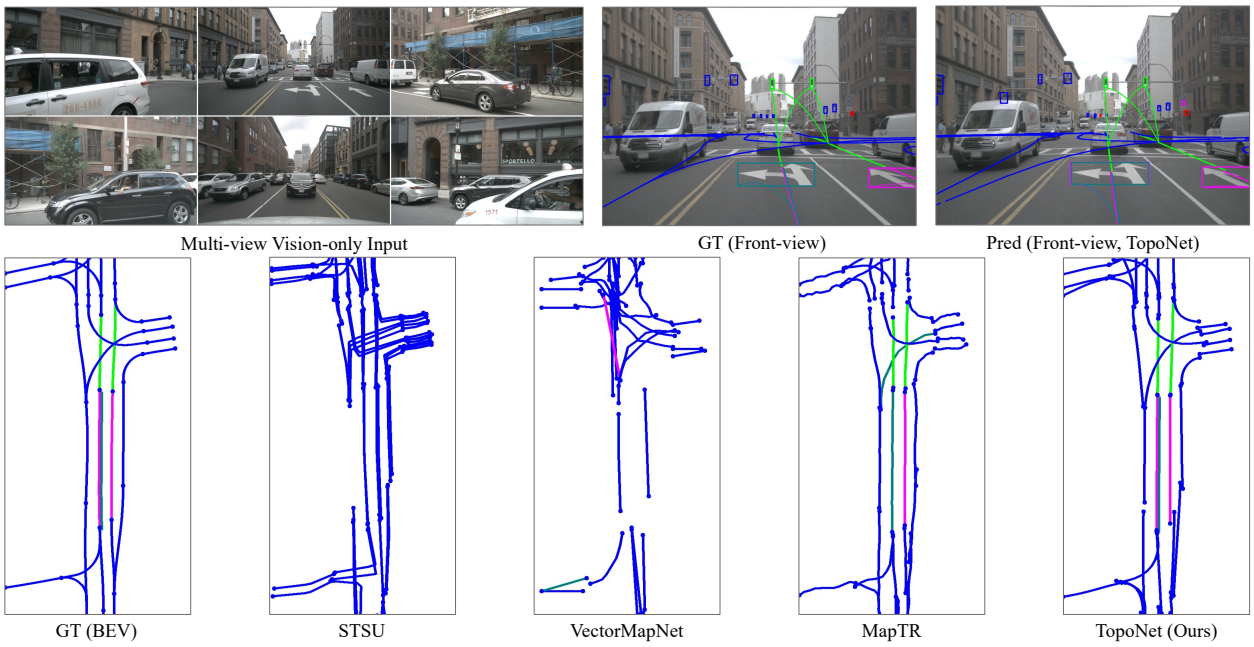

Figure 5: **Qualitative results** of TopoNet and other algorithms on *subset_B* of the OpenLane-V2 dataset. Colors denote categories of traffic elements.

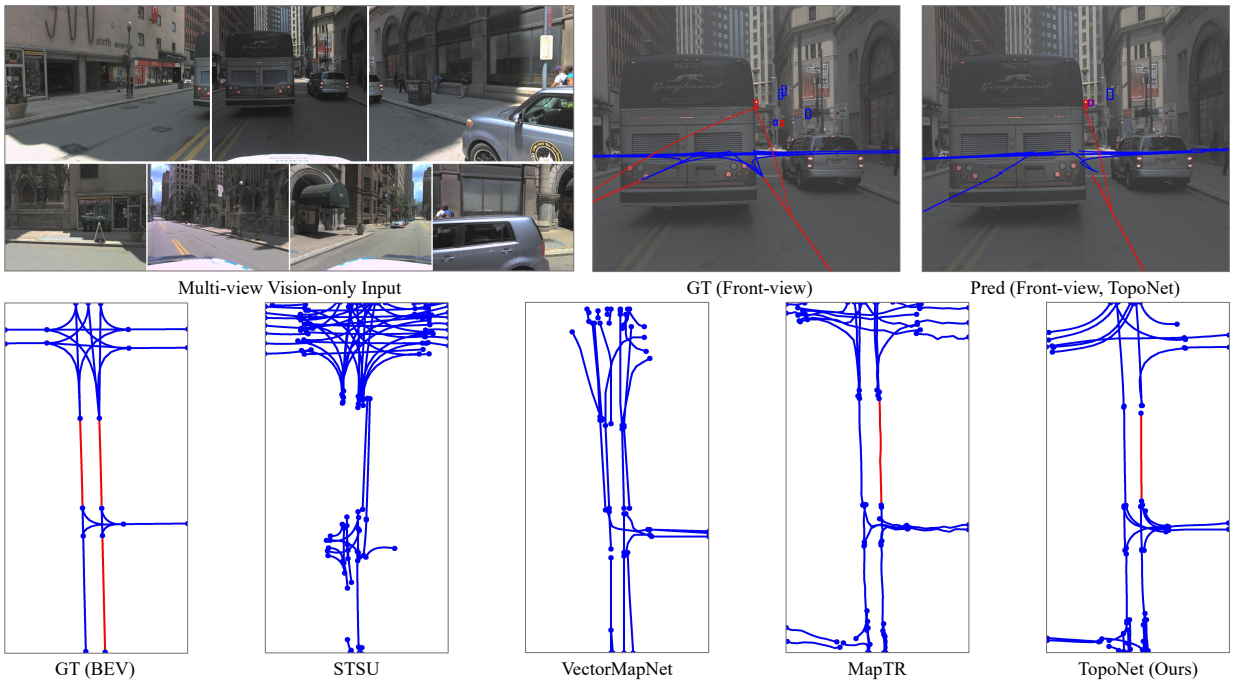

Figure 6: **Failure case under large-area occlusion.** TopoNet fails to predict centerlines and the lane graph in the intersection with a large bus colluding in front. Note that the relationship between the left lane and the red light is an incorrect annotation where our algorithm reasons about the direction of the left lane and avoids the false positive prediction.