eTLSD: An Efficient Topological Lane Segment Detector with GNN and Distance Awareness

Author Names Omitted for Anonymous Review. Paper-ID 780

Abstract—**High-Definition Maps (HD Maps) are essential for both Advanced Driver-Assistance Systems (ADAS) and autonomous vehicles. Offline HD map construction is costly and difficult to maintain due to the dynamic nature of real-world driving environments. Consequently, online HD map generation using onboard sensors has become a focus of the autonomous driving community. However, current deep learning-based methods often rely on computationally intensive feature extractors, hindering their practical deployment. In this paper, we propose eTLSD, an efficient end-to-end neural network that generates rich HD maps containing both topological and geometric road information. To achieve this efficiency and improved performance, we introduce four key innovations: (1) an iterative refinement scheme within the decoder; (2) a group-wise one-to-many assignment strategy for faster convergence of the map learning network; (3) a novel module incorporating lane segment coordinates using a graph neural network; and (4) a distance-aware topological reasoning post-processing method to enhance the quality of connectivity outputs. Evaluations on the OpenLane-V2 dataset demonstrate that eTLSD achieves an average OLUS performance improvement of 6.7% (6.4% on mAP and 14.5% on TOP$_{lsls}$ metric) compared to existing methods while utilizing a significantly lighter ResNet-18 backbone instead of the commonly employed ResNet-50. This enhanced performance with a smaller model facilitates easier deployment.**

I. Introduction

High-definition (HD) maps provide rich semantic information about driving scenes due to their powerful modeling capabilities. However, constructing high-quality HD maps remains challenging.

A robust HD map system must accurately represent map elements and promptly reflect real-world changes. However, due to the dynamic nature of real-world environments, manually updating these maps is labor-intensive, making them costly to create and maintain. Therefore, a system capable of constructing HD maps online using onboard sensor data is highly desirable for long-term sustainability.

A variety of sensor configurations are employed when constructing maps from onboard sensors. Autonomous vehicles developed by companies like Waymo and Zoox utilize a combination of LiDAR, radar, and cameras to acquire precise 3D information. While effective, these multi-sensor approaches, particularly those incorporating LiDAR, are not readily scalable due to cost considerations. Furthermore, integrating data from diverse sensors presents significant data fusion challenges due to differing data formats. Among commonly available sensors, cameras offer a cost-effective solution. The rich semantic information captured by cameras makes them well-suited for visual perception tasks. Coupled with advancements in camera-based perception within the research community, online HD map construction using only camera data presents a compelling and sustainable approach.

In the autonomous driving community, many perception works [13],[14],[12],[11],[23],[18] transform the perspective view of multi-camera images into a unified Bird-Eye-View (BEV) representation for downstream tasks such as occupancy prediction, 3D object detection and HD map construction. This provides a versatile 3D representation of the driving scenes thanks to its broad field of view that clearly shows the position of the ego vehicle relative to surrounding driving objects. Perception plays an important role in the driving stacks since it is the first stage and directly influences motion planning. BEV representation is widely adopted because it can seamlessly integrate perception and planning into a unified end-to-end autonomous driving framework.

Current approaches to HD-map construction typically decompose the task into two sub-problems: lane detection and topology reasoning [9, 10, 22]. Many aim to address these jointly within a unified learning framework by transforming surrounding-view images into a Bird's-Eye View (BEV) representation. BEV offers a natural and versatile representation of autonomous driving due to its comprehensive field of view. The existing literature on online map learning can be broadly classified into two main streams: map element detection [8, 14, 17] and centerline perception [7, 9]. While map element detection focuses primarily on perceiving road geometry, it often neglects the connectivity between map elements. Conversely, centerline perception emphasizes detecting road centerlines but largely disregards geometric details.

Therefore, a framework capable of learning both road geometry and topology is needed. To address this, lane segment perception was proposed by Li et al. [10] as a means of combining geometric and topological road information using deformable attention [24]. However, the stacked deformable attention operations in this approach lead to substantial random memory access, creating a bottleneck for edge computing devices. In this paper, we contribute a lightweight lane segment perception model designed for efficient edge deployment, achieved through the following improvements:

- We propose an iterative refinement scheme to better guide the transformer decoder in learning lane relationships from the extracted features.
- We introduce a group-wise one-to-many assignment strategy to improve the convergence of the lane segment perception model during training.
- We design a novel module that leverages a graph neural network to incorporate lane segment coordinates into the

learning of lane connectivity.

- We acclimate a distance-aware post-processing method to enhance topological reasoning.

II. Related works

As previously mentioned, the online HD map construction community primarily utilizes two main frameworks. Map element detection is crucial for scene understanding in autonomous driving. The conventional approach within this community involves transforming camera-derived features into a unified Bird's-Eye View (BEV) representation prior to downstream processing.

A. Map Element Detection

Following the rise of BEV perception, HDMapNet [8] directly constructs HD maps by grouping and vectorizing segmented map elements within a BEV grid, followed by post-processing. Subsequently, VectorMapNet [17] introduced a DETR-based [2] map element detection model to learn vectorized map elements end-to-end, eliminating the need for post-processing. MapTR [14] further proposed a unified permutation-equivalent modeling approach to stabilize the learning process in DETR-like models.

B. Centerline Perception

STSU [1] proposes a DETR-like network for centerline prediction, employing a fully connected network module to determine lane connectivity. Building upon STSU, TopoNet [9] utilizes a graph convolutional network to enhance topology reasoning between centerlines. Conversely, TopoMLP [22] presents a simple yet effective baseline using two transformer decoder branches followed by Multi-Layer Perceptrons (MLPs). Topo2D [7] is another query-based detector that incorporates 2D features as queries to improve lane detection and reasoning in 3D space.

C. Lane Segment Perception

As previously noted, a unified approach encompassing both map element detection and centerline perception can further advance map learning. LaneSegNet [10] pioneered this direction by proposing lane segment perception, a novel lane representation that models both the geometry and connectivity of lanes by predicting left and right boundaries along with the centerline. Topologic [5] developed a new method for calculating lane topology matrices based on geometric distances between lanes and lane query similarity, subsequently fusing these matrices to enhance lane topology computation.

III. Lane segment representation

Given multi-view images of the surrounding driving scene, the objective is to predict lane boundaries and their topological relationships within a BEV representation.

A lane segment instance incorporates both geometric and semantic information about the road. Geometrically, it can be represented as a line segment defined by a vectorized centerline and its corresponding lane boundaries.

$$\mathcal{V} = \{\mathcal{V}_{\text{center}}, \mathcal{V}_{\text{left}}, \mathcal{V}_{\text{right}}\}, \tag{1}$$

where $\mathcal{V}_{\text{center}}, \mathcal{V}_{\text{left}}, \mathcal{V}_{\text{right}}$ are representations of the center, left, and right lane-lines of a lane segment. Each lane line is defined as a sequence of N_p points in 3D space, denoted as $\mathcal{V}_{\text{lane}} = [\mathbf{p}_0, \mathbf{p}_1, ..., \mathbf{p}_{N_p-1}]$, where $\mathbf{p} = (x, y, z) \in \mathbb{R}^3$ represents the 3D vehicle-centric coordinates of a point. $\mathcal{V}_{\text{lane}}$ can be the centerline ($\mathcal{V}_{\text{center}}$), left boundary ($\mathcal{V}_{\text{left}}$), or right boundary ($\mathcal{V}_{\text{right}}$). Semantically, each lane line includes the line type of the left and right lane boundaries (e.g., non-visible, solid, dashed): $\{A_{\text{left}}, A_{\text{right}}\}$. These details provide autonomous vehicles with crucial information regarding deceleration requirements and lane change feasibility. Furthermore, pedestrian crossings, along with lane segments, contribute to the creation of a rich HD map.

Topological information is crucial for path planning. To represent this information, a lane graph, denoted as $\mathcal{G} = (\mathbf{V}, \mathbf{E})$, is constructed for the lane segments. Each lane segment is represented as a node in the graph $\mathbf{V}$, and the edges $\mathbf{E}$ represent the connectivity between lane segments. An adjacency matrix is used to represent this lane graph, where the element $\mathcal{G}_{i,j}$ is set to 1 if the j-th lane segment follows the i-th segment, and 0 otherwise. The output consists of road boundaries or lane segments represented as polylines, as described above, and their topological relationships encoded in the adjacency matrix.

IV. Method

A. Framework of eTLSD

The model consists of three main components: an image encoder, a transformer decoder, and prediction heads for lane segment perception.

1) Image Encoder: The encoder processes multi-view images from vehicle-mounted cameras to generate a unified feature representation. We employ a ResNet-18 backbone [6] to extract multi-scale spatial features, which are then enhanced using a Feature Pyramid Network (FPN) [16]. Following prior work [9, 7, 5, 22, 14], these features are transformed into a BEV representation using the BEVFormer [13] view transformation module, resulting in BEV image features. More specifically, the encoder takes an input image $\mathbf{I}$ and outputs BEV image features $\mathbf{F_{BEV}} \in \mathbb{R}^{\mathbf{H}\times\mathbf{W}\times\mathbf{C}}$, where the BEV grid has dimensions $H \times W$, and each grid cell has C channels.

$$\text{Encoder}(\mathbf{I}) \rightarrow \mathbf{F_{BEV}} \tag{2}$$

2) Transformer Decoder: The transformer decoder uses the attention mechanism across multiple layers to learn features from the BEV representation and extract information about lane segments and their topology. The queries are updated after each layer. The decoder takes as input the BEV features $\mathbf{F_{BEV}}$ and a set of N lane segment queries, denoted as $\mathbf{Q}(= [\mathbf{q_1}, \mathbf{q_2}, \dots \mathbf{q_N}])$.

$$\text{Decoder}(\mathbf{Q}, \mathbf{F_{BEV}}) \rightarrow \tilde{\mathbf{Q}} \tag{3}$$

The decoder employs Multi-Head Self-Attention [20], followed by a specialized deformable cross-attention mechanism designed to capture the elongated shapes and spatial relationships of lanes. Unlike conventional deformable attention, this

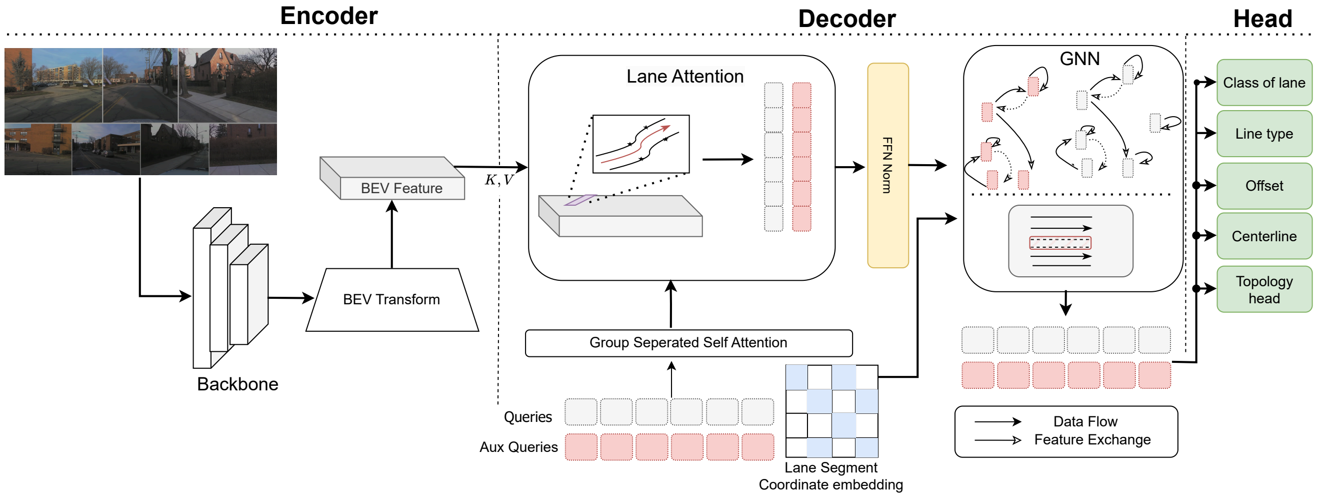

Fig. 1. Overview of the eTLSD framework. We adopt an encoder-decoder architecture inspired by DETR [2]. The encoder extracts features from multi-view images. The decoder comprises a lane attention module and a graph neural network for efficient lane segment feature learning. Finally, prediction heads consisting of Multi-Layer Perceptrons predict lane segment attributes.

lane attention module strategically distributes reference points along lane boundaries. Each reference point is associated with multiple sampling locations, enabling fine-grained attention to local details. This multi-head attention mechanism ensures effective utilization of both local geometric features and global contextual cues.

Specifically, the lane attention module from [10] takes the i-th lane segment query $\mathbf{q_i}$, $\mathbf{F_{BEV}}$, and a set of reference points $\mathbf{p_i}$ as input. The estimated query at index i, $\tilde{\mathbf{q}}_\mathbf{i}$ is based on the formulation from DeformableDETR [24]:

$$\sum_{m=1}^{K} \mathbf{W}_m \sum_{k=1}^{K} [a_{i,m,k} \cdot \mathbf{W}'_m \cdot \mathbf{F_{BEV}}(\mathbf{p_{i,m}} + \Delta \mathbf{p_{i,m,k}})], \tag{4}$$

where m be the attention head index, k be the sampling location index. The term $\Delta p_{i,m,k}$ denotes the sampling offset, and $a_{i,m,k}$ represents the attention weight of sampling point k in attention head m. These values, $a_{i,m,k}$ and $\Delta p_{i,m,k}$, are obtained via a linear projection of the query $\mathbf{q_i}$, with the constraint that $\sum_{k=1}^{K} a_{i,m,k} = 1$. As $\mathbf{p_{i,m}} + \Delta \mathbf{p_{i,m,k}}$ is typically fractional, bilinear interpolation, as described in [4], is used to compute $\mathbf{F_{BEV}}(\mathbf{p_{i,m}} + \Delta \mathbf{p_{i,m,k}})$. The weights $\mathbf{W}'_m \in \mathbb{R}^{C_v \times C}$ and $\mathbf{W}_m \in \mathbb{R}^{C \times C_v}$ are learnable parameters that project the value features into split channels (where $C_v = \frac{C}{M}$) and merge the multi-branch features, preserving information from each attention head.

3) Prediction Heads: The prediction heads generate lane segment predictions from the refined lane segment queries output by the decoder. Using a series of Multi-Layer Perceptrons (MLPs):

$$\text{Predictor}(\tilde{\mathbf{Q}}) \rightarrow \mathbf{Y}, \tag{5}$$

where $\mathbf{Y}$ is the final lane segment predictions. The representation of predicted lane segments comprises several key components. First, the *lane coordinates*, $\mathcal{V}_{\text{center}} \in \mathbb{R}^{N_p \times 3}$, define the 3D coordinates of the centerline points, represented as $[(x_0, y_0, z_0)]$. Second, the *lane offsets*, $\mathcal{V}_{\text{offset}} \in \mathbb{R}^{N_p \times 3}$, describe the left and right lane boundaries relative to the centerline, with elements $[(\Delta x_0, \Delta y_0, \Delta z_0)]$. Third, the *lane semantics*, denoted by $\mathcal{C}$ (e.g., lane segment, pedestrian crossing), provide classifications of lane types, along with the line types of the left and right lane boundaries, $\mathcal{A}_{\text{left}}$ and $\mathcal{A}_{\text{right}}$, respectively (e.g., non-visible, solid, or dashed). Finally, the *connectivity matrix*, represented by a topological graph $\mathcal{G}$, captures the relationships between lane segments in the form of a weighted adjacency matrix.

4) Optimization: Following the encoder-decoder paradigm of DETR [2], the Hungarian algorithm is used to compute the optimal assignment between model predictions and ground truth. Training losses are then calculated based on this assignment, as follows:

$$\begin{aligned} \mathcal{L} = &\lambda_{\text{vec}} \mathcal{L}_{\text{vec}} + \lambda_{\text{seg}} \mathcal{L}_{\text{seg}} + \lambda_{\text{cls}} \mathcal{L}_{\text{cls}} \\ &+ \lambda_{\text{type}} \mathcal{L}_{\text{type}} + \lambda_{\text{top}} \mathcal{L}_{\text{top}} , \end{aligned} \tag{6}$$

where the losses functions are:

- The geometric loss $\mathcal{L}_{\text{vec}}$ supervises the accuracy of predicted lane geometries by comparing predicted lane segments to ground truth using the Manhattan distance.
- The classification loss $\mathcal{L}_{\text{cls}}$ addresses class imbalance in predicting lane types and attributes using the focal loss from [19].
- The segmentation loss $\mathcal{L}_{\text{seg}}$ supervises predicted masks using a combination of Dice loss and cross-entropy loss to improve learning from the BEV features $\mathbf{F_{BEV}}$.
- The type loss $\mathcal{L}_{\text{type}}$ supervises lane line type prediction using cross-entropy loss, and the topology loss $\mathcal{L}_{\text{top}}$ uses focal loss to supervise the relationships between lane segments.

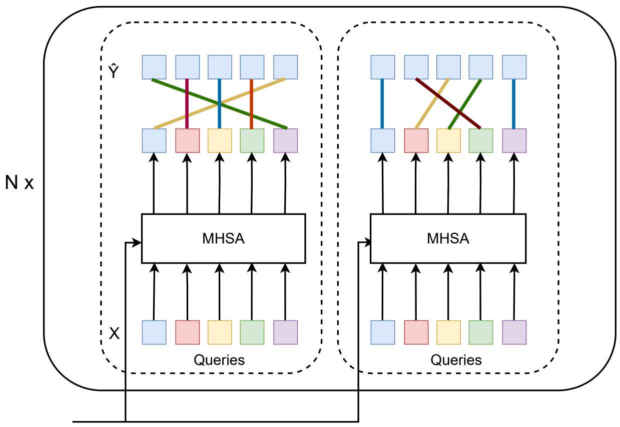

Fig. 2. The group-wise one-to-many assignment training strategy.

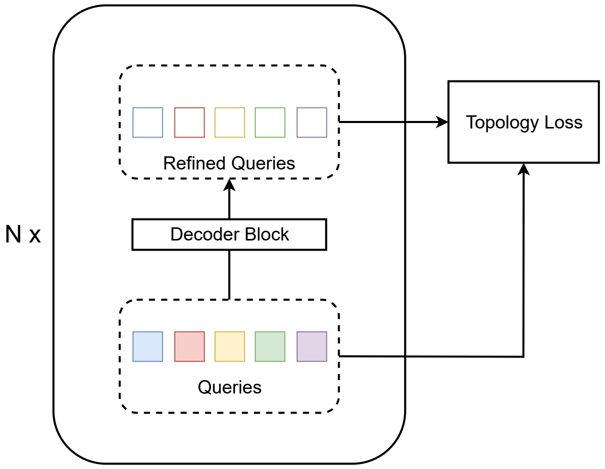

Fig. 3. Illustration of the iterative refinement scheme for topology prediction.

B. Group Query

As previously mentioned, training DETR-based frameworks [2] involves computing the optimal assignment between model predictions and ground truth prior to loss calculation. The original DETR employs a one-to-one assignment using the Hungarian algorithm, which can lead to slow convergence and prolonged training times. Several works have addressed this issue by incorporating many-to-one assignment strategies into DETR training. For example, MapTRv2 [15] introduces hierarchical bipartite matching to enhance DETR training for map element detection, along with a new loss function designed to facilitate this matching scheme.

Chen et al. [3] proposed a simple yet effective method for improving DETR training. They introduced a grouping scheme for object queries, termed group-wise one-to-many assignment. This approach performs one-to-one matching within each query group, allowing a single ground-truth object to be assigned to multiple predictions. Consequently, the prediction closest to the ground-truth object receives a high matching score, while redundant predictions within the same group receive lower scores.

We employ a group-wise one-to-many assignment strategy as follows: N lane segment queries form the primary group, denoted as $\mathbf{Q}$. Subsequently, $K-1$ additional groups, each containing N queries, are introduced, resulting in K groups in total, $\mathbf{Q}_1$ to $\mathbf{Q}_K$. Correspondingly, we have K groups of lane segment predictions, denoted as $\mathbf{Y}_1$ through to $\mathbf{Y_K}$. One-to-one assignment is then performed within each group to determine the optimal matching between each group of predicted lane segments and the ground truth $(\mathbf{Y_k}, \hat{\mathbf{Y}})$. Parallel Multi-Head Self-Attention is applied to each query group, and the resulting outputs are concatenated and fed to the lane attention module. With this group-wise assignment strategy, the training procedure proceeds as follows:

$$\begin{aligned}
&\text{Decoder}(\mathbf{F_{BEV}},\mathbf{Q}_1)\rightarrow\tilde{\mathbf{Q}}_1, &&\text{Predictor}(\tilde{\mathbf{Q}}_1)\rightarrow\mathbf{Y}_1,\\
&\text{Decoder}(\mathbf{F_{BEV}},\mathbf{Q}_2)\rightarrow\tilde{\mathbf{Q}}_2, &&\text{Predictor}(\tilde{\mathbf{Q}}_2)\rightarrow\mathbf{Y}_2,\\
&\quad\vdots\\
&\text{Decoder}(\mathbf{F_{BEV}},\mathbf{Q}_K)\rightarrow\tilde{\mathbf{Q}}_K, &&\text{Predictor}(\tilde{\mathbf{Q}}_K)\rightarrow\mathbf{Y}_K
\end{aligned} \tag{7}$$

During inference, the decoder operates similarly to the training phase, with the key difference being that only one group of queries is used. The total loss during training is the sum of the individual losses from each of the K decoders, expressed as follows:

$$\frac{1}{K}\sum_{k=1}^{K}\sum_{n=1}^{N}\mathcal{L}(\mathbf{Y}_{\sigma_k(n)},\tilde{\mathbf{Y}}_{kn}), \tag{8}$$

where $\mathcal{L}$ is the loss function from (6) and $\sigma_k(.)$ is the optimal permutation of N indices for the k-th decoder as in [3].

C. Topology Iterative Refinement

Previous map learning approaches typically do not employ iterative refinement for topology prediction; instead, they apply loss functions only to the output of the final decoder layer. Inspired by the bounding box refinement strategy in the DETR framework [2], we apply this effective refinement scheme to topology prediction. In eTLSD, after each transformer decoder layer, the lane segment queries are iteratively refined by applying the loss function.

D. Graph Neural Network

TopoNet [9] first proposed a Scene Knowledge Graph by constructing a learnable weight matrix for the lane graph, denoted 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

$$\begin{aligned}
K_{ll}^{i}&=\texttt{stack}\left(A_{ll}^{i-1},\texttt{transpose}\left(A_{ll}^{i-1}\right),I\right),\\
Q_{l_{(x)}}^{i'}&=\sum_{\forall y\in N(x)}\sum_{\forall c_l\in C_l}\beta_{ll}\cdot K_{ll_{(c_l,x,y)}}^{i}\mathbf{W}_{l_{l(c_l)}}^{i}Q_{l_{(y)}}^{i}.
\end{aligned} \tag{9}$$

where $N(x)$ outputs the indices of all neighbors of the vertex with index x; A_{ll}^{i-1} is the adjacency matrix. The topology head reasons pairwise relationships on the given embeddings Q_a' and Q_b' to predict TOP_lsls. TopoNet [9] uses two MLP layers to reduce the feature dimension for 2 instance queries Q_a and Q_b with 256 feature channels. The number of output channels is 128:

$$Q_a' = \text{MLP}_a(Q_a), \quad Q_b' = \text{MLP}_b(Q_b), \tag{10}$$

In our work, to enhance the topology features for each candidate, we concatenate the instance queries Q_a, Q_b with their respective lane segment coordinate embeddings, l_a, l_b, before feeding them into the topology head. The embedding network consists of a two-layer MLP with ReLU activations that encodes the predicted lane segment coordinates (an 11×3 point set per centerline) into 256 feature channels.

$$\begin{aligned} Q_a' &= \texttt{concat}(\text{MLP}_{qa}(Q_a), \text{MLP}_{la}(l_a)), \\ Q_b' &= \texttt{concat}(\text{MLP}_{qb}(Q_b), \text{MLP}_{lb}(l_b)) \end{aligned} \tag{11}$$

The concatenated feature is then passed through another MLP with a sigmoid activation to predict their relationship. Based on the matching results from the perception heads, the ground truth for each pair of embeddings is assigned. For each pair of queries $q_a' \in Q_a'$ and $q_b' \in Q_b'$, the output is the confidence of the relationship, with independent MLPs for different types of relationships:

$$\text{confidence} = \texttt{sigmoid}(\text{MLP}_{\text{top}}(\texttt{concat}(q_a', q_b')), \tag{12}$$

where $\texttt{concat}, \texttt{sigmoid}$ are the concatenate and sigmoid function.

E. Distance-aware post-processing

The topological relationship between centerlines depends not only on semantic information but also on their geometric locations. If the endpoints of two centerlines are in close proximity, they are likely topologically related. At the i-th lane decoder layer, we predict N_l lane lines and then compute the geometric distances between the endpoint of one lane centerline and the starting points of the others.

$$\begin{aligned} l_0, \ldots, l_{N_l-1} &= \text{LaneHead}\left(Q_l^i\right) \\ d_{ij} &= \left| l_i^{\text{end}} - l_j^{\text{start}} \right| \\ D &= \{d_{ij} \mid i, j = 0 \ldots N_l - 1\} \end{aligned} \tag{13}$$

where $D \in \mathcal{R}^{N_l \times N_l}$ is the lane geometric distance matrix that contains L1 distance d_{ij} between l_i^{end} - the last point of lane line l_i, and l_j^{start} - the first point of lane line l_j,

To map distance to topology matrix, we utilize Topologic[5] mapping function $f : \mathcal{R} \rightarrow [0, 1]$ as follows:

$$f = e^{-\frac{x^\alpha}{\lambda \cdot \sigma}} \tag{14}$$

where $x = d_{ij}$, σ is the standard deviation of the geometric distance matrix D. In previous work[5], α, λ are learanable parameters, however, since we employ mapping function in the post-processing phase, we choose fixed values for α, λ derived from [5]. With the help of such mapping, we can get a lane topology as follows:

$$G_{dis} = \{f(d_{ij}) \mid i, j = 0 \ldots N_l - 1\} \tag{15}$$

The final connectivity matrix outputs combine both distance awareness and the high-space features:

$$G = \lambda_{dis} \cdot G_{dis} + \lambda_{hf} \cdot G_{hf}, \tag{16}$$

where G_{hf} is the output introduced in section IV-D, and $\lambda_{dis}, \lambda_{hf}$ are set to 1 to balance the contribution of both information.

V. Experiments

A. Dataset and metrics

1) Dataset: We evaluate our method on a widely used dataset for this task: the OpenLane-V2 Subset A dataset [21]. This dataset contains 1,000 15-second scenes, with 27,000 frames for training and 4,800 frames for validation.

2) Metrics: There are five metrics to evaluate the mapping tasks:

- **AP**$_{ls}$: shows the average precision of lane segment computed under `Chamfer` and `Frechet` distance threshold of $\{0.5, 1.0, 1.5\}$ meters. Concretely, to measure the accuracy between the predicted segment $\tilde{S} = \{\tilde{\mathcal{V}}_{left}, \tilde{\mathcal{V}}_{center}, \tilde{\mathcal{V}}_{right}\}$ and ground truth $S = \{\mathcal{V}_{left}, \mathcal{V}_{center}, \mathcal{V}_{right}\}$, we employ the following equation:

$$\begin{aligned} D(\tilde{S}, S) = &\frac{1}{2}\Big[D_{Frechet}(\tilde{\mathcal{V}}_{center}, \mathcal{V}_{center}) \\ &+ D_{Chamfer}(\tilde{\mathcal{V}}_{left}, \mathcal{V}_{left}) \\ &+ D_{Chamfer}(\tilde{\mathcal{V}}_{right}, \mathcal{V}_{right})\Big] \end{aligned} \tag{17}$$

- **AP**$_{ped}$: illustrates the average precision of pedestrian crossing to evaluate map element construction quality. Similar to lane segment, this score is computed by Chamfer distance.
- **mAP**: is the mean AP computed as the average of AP$_{ls}$ and AP$_{ped}$.
- **TOP**$_{lsls}$: measures the performance of topology reasoning. Specifically, the ground truth connectivity graph $G = (V, E)$ and the corresponding prediction $\tilde{G} = (\tilde{V}, \tilde{E})$, two vertices are considered connected if the predicted confidence score of the edge between them exceeds 0.5. The **TOP**$_{lsls}$ for a given vertex is determined by ranking all predicted edges associated with it and computing the cumulative mean of the precision values. Mathematically, the **TOP**$_{lsls}$ is expressed as:

$$\textbf{TOP}_{lsls} = \frac{1}{|V|} \sum_{v \in V} \frac{\sum_{\hat{n} \in N(v)} P(\hat{n}) \cdot \mathbb{1}(\hat{n} \in N(v))}{|N(v)|} \tag{18}$$

- **OLUS**: The overall performance for HDMap construction, equal $\frac{1}{2}\left(\textbf{mAP} + f(\text{TOP}_{lsls})\right)$, where f is the square root function to emphasize the importance of topology detection accuracy.

B. Results

We first conduct a comprehensive comparison between the baseline methods (TopoNet, MapTR, MapTRv2, LaneSegNet, Topologic) and our approach (eTLSD) trained on the training dataset of OpenLane-V2 Subset A [21]. Table I reports the performance on the accuracy of map element detection and the topology prediction between lanes. In evaluation configuration, our eTLSD with a much smaller backbone (ResNet-18) still achieves state-of-the-art detection results (mAP, AP$_{ls}$, AP$_{ped}$) while has a comparable overall OLUS score compared to Topologic. In detail, eTLSD outperforms other ResNet-50 methods up to 4.5% on mAP, 2.1% on lane detection AP$_{ls}$, and 6.6% on pedestrian crossing detection AP$_{ped}$. Under a fair comparison in terms of backbone, eTLSD consistently surpasses LaneSegNet for the road geometry detection and topology prediction tasks. Notably, eTLSD beats LaneSegNet with ResNet-50 by a large margin.

Method	**Arch**	**mAP**	**AP$_{ls}$**	**AP$_{ped}$**	**TOP$_{lsls}$**	**OLUS**
TopoNet [9]	R50	23.0	23.9	22.0	-	-
MapTR [14]	R50	27.0	25.9	28.1	-	-
MapTRv2 [15]	R50	28.5	26.6	30.4	-	-
Topologic [5]	R50	33.2	33.0	33.4	**30.8**	**44.3**
LaneSegNet [10]	R50	32.6	32.3	32.9	25.4	41.5
LaneSegNet [10]	R18	30.4	30.4	30.4	24.9	40.1
eTLSD (ours)	R18	**34.7**	**33.7**	**35.6**	29.1	**44.3**

TABLE I
COMPARISON OF ETLSD WITH STATE-OF-THE-ART METHODS. WE ACHIEVE SUPERIOR PERFORMANCE ON MOST METRICS USING A SIGNIFICANTLY SMALLER BACKBONE (RESNET-18).

C. Ablation Study

In table II, we provide ablations to analyze merits of our proposed modules. First, polishing topology queries iteratively across decoder layers increases the lane prediction score by 2.3% and achieve the same topology performance as LaneSegNet with ResNet-50. In the second contribution, compared to the one-to-one assignment applied in DETR that demote the duplicate predictions, the use of one-to-many assignment in group DETR significantly improves the overall performance, especially enhances the pedestrian crossing prediction by 17.7%. Our experiment shows that Lane Coordinate Embedding enhance lane detection, however, it maintain topology score while dropping AP$_{ped}$. The reason for that is topology loss prioritizes lane over pedestrian predictions. The replacement of reasoning and lane decoders from MLP in baseline to SGNN illustrates a slight improvement overally, which is expected since SGNN is designed to enhance lane-lane feature interaction by propagating information among lane queries across decoder layers. Finally, we adopt Distance-aware post-processing to utilize geometric clues for relationship prediction between lanes, which brings about a 3.5% gain of TOP$_{lsls}$ score.

Table III compares the efficiency of eTLSD versus the baseline [10] in terms of FPS and number of parameters. We benchmarked the performance on a single Nvidia RTX4090 GPU. Our method has 19.7% fewer parameters and 32.1% larger FPS.

Setting	**mAP**	**AP$_{ls}$**	**AP$_{ped}$**	**TOP$_{lsls}$**
Lansegnet [10] R50 (from paper)	32.6	32.3	32.9	25.4
Lansegnet [10] R18	30.4	30.4	30.4	24.9
+ Topology Iterative Refinement	30.6	31.1	30.1	25.4
+ Group DETR	33.8	31.8	**35.8**	27.3
+ Lane Coordinate Embedding	33.4	33.3	33.4	27.3
+ SGNN	**34.7**	**33.7**	35.6	28.1
+ Distance-aware PP	**34.7**	**33.7**	35.6	**29.1**

TABLE II
ABLATION STUDY OF PROPOSED TECHNIQUES WITH CUMULATIVE ADDITION.

Methods	**FPS**	**#Params**
LaneSegNet [10] R18 (from paper)	17.6	32.4M
LaneSegNet [10] R50 (from paper)	10.6	45.1M
eTLSD	**14.0**	**36.2M**

TABLE III
COMPARISON ON FPS AND PARAMETER COUNT.

VI. LIMITATIONS AND FUTURE WORKS

While this work focuses primarily on lanes and their topological relationships, a natural extension would be to incorporate other factors, such as traffic element detection and relationships between lanes and traffic. Moreover, while the geometric constraints we employ significantly improve lane topology prediction, they could potentially bias the model toward reconstructing wrong lane lines. A promising direction for future work would be to integrate geometric cues into map element prediction tasks. On the other hand, we have yet to take into account the temporal nature of video while designing this framework. Integrating temporal relationship of visual data is a promising future direction thanks to the rich semantic information containing within video.

VII. CONCLUSION

This work introduces eTLSD, a novel approach for generating HD maps. Our method achieves state-of-the-art performance in map element prediction accuracy and model efficiency by leveraging a lightweight backbone combined with a sophisticated training and post-processing strategy. During training, we incorporate iterative topology refinement, group-wise DETR training, and a graph neural network with lane coordinate embeddings. Post-processing employs a distance-aware map function to further enhance topology prediction. Overall, eTLSD significantly improves both map element prediction and lane topology, achieving a 6.7% gain in OLUS score compared to the baseline. Our model is designed for deployment on edge devices in Self-Driving Vehicles (SDVs). However, we still need to improve the efficiency of the model to be deployed and operate in real-time on edge devices.

References

[1] Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Structured bird's-eye-view traffic scene understanding from onboard images. *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, 2020.

[3] Qiang Chen, Xiaokang Chen, Jian Wang, Shan Zhang, Kun Yao, Haocheng Feng, Junyu Han, Errui Ding, Gang Zeng, and Jingdong Wang. Group detr: Fast detr training with group-wise one-to-many assignment. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2023.

[4] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[5] Yanping Fu, Wenbin Liao, Xinyuan Liu, Hang xu, Yike Ma, Feng Dai, and Yucheng Zhang. Topologic: An interpretable pipeline for lane topology reasoning on driving scenes, 2024.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[7] Han Li, Zehao Huang, Zitian Wang, Wenge Rong, Naiyan Wang, and Si Liu. Enhancing 3d lane detection and topology reasoning with 2d lane priors. *arXiv preprint arXiv:2406.03105*, 2024.

[8] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4628–4634. IEEE, 2022.

[9] Tianyu Li, Li Chen, Huijie Wang, Yang Li, Jiazhi Yang, Xiangwei Geng, Shengyin Jiang, Yuting Wang, Hang Xu, Chunjing Xu, Junchi Yan, Ping Luo, and Hongyang Li. Graph-based topology reasoning for driving scenes. *arXiv preprint arXiv:2304.05277*, 2023.

[10] Tianyu Li, Peijin Jia, Bangjun Wang, Li Chen, Kun Jiang, Junchi Yan, and Hongyang Li. Lanesegnet: Map learning with lane segment perception for autonomous driving. In *ICLR*, 2024.

[11] Yangguang Li, Bin Huang, Zeren Chen, Yufeng Cui, Feng Liang, Mingzhu Shen, Fenggang Liu, Enze Xie, Lu Sheng, Wanli Ouyang, and Jing Shao. Fast-bev: A fast and strong bird's-eye view perception baseline. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(12):8665–8679, 2024.

[12] Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2023.

[13] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European conference on computer vision*, 2022.

[14] 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 *International Conference on Learning Representations*, 2023.

[15] 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. *International Journal of Computer Vision*, pages 1–23, 2024.

[16] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[17] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *International Conference on Machine Learning*, pages 22352–22369. PMLR, 2023.

[18] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE international conference on robotics and automation (ICRA)*, pages 2774–2781. IEEE, 2023.

[19] T-YLPG Ross and GKHP Dollár. Focal loss for dense object detection. In *proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[20] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[21] 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*, 2023.

[22] Dongming Wu, Jiahao Chang, Fan Jia, Yingfei Liu, Tiancai Wang, and Jianbing Shen. Topomlp: An simple yet strong pipeline for driving topology reasoning. *ICLR*, 2024.

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

[24] 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*, 2021.