Highlights

TopoBDA: Towards Bezier Deformable Attention for Road Topology Understanding

Muhammet Esat Kalfaoglu, Halil Ibrahim Ozturk, Ozsel Kilinc, Alptekin Temizel

- Novel MPDA integration into Bezier regression improves road topology understanding.
- Novel Bezier Deformable Attention enhances road topology understanding.
- Instance mask formulation boosts road topology understanding performance.
- Multi-modal fusion achieves state-of-the-art road topology understanding.
- One-to-many set prediction loss analyzed for road topology understanding for the first time.

TopoBDA: Towards Bezier Deformable Attention for Road Topology Understanding

Muhammet Esat Kalfaoglu[a,b,*], Halil Ibrahim Ozturk[b], Ozsel Kilinc[b], Alptekin Temizel[a]

[a] *Graduate School of Informatics, Middle East Technical University, Ankara, Turkey*
[b] *Togg/Trutek AI Team, Ankara, Turkey*

Abstract

Understanding road topology is crucial for autonomous driving. This paper introduces TopoBDA (Topology with Bezier Deformable Attention), a novel approach that enhances road topology comprehension by leveraging Bezier Deformable Attention (BDA). TopoBDA processes multi-camera 360-degree imagery to generate Bird's Eye View (BEV) features, which are refined through a transformer decoder employing BDA. BDA utilizes Bezier control points to drive the deformable attention mechanism, improving the detection and representation of elongated and thin polyline structures, such as lane centerlines. Additionally, TopoBDA integrates two auxiliary components: an instance mask formulation loss and a one-to-many set prediction loss strategy, to further refine centerline detection and enhance road topology understanding. Experimental evaluations on the OpenLane-V2 dataset demonstrate that TopoBDA outperforms existing methods, achieving state-of-the-art results in centerline detection and topology reasoning. TopoBDA also achieves the best results on the OpenLane-V1 dataset in 3D lane detection. Further experiments on integrating multi-modal data—such as LiDAR, radar, and SDMap—show that multimodal inputs can further enhance performance in road topology understanding.

Keywords:
Road Topology Understanding, Centerline Detection, Autonomous Driving,

*Corresponding author
Email addresses: esat.kalfaoglu@metu.edu.tr (Muhammet Esat Kalfaoglu), ibrahim.ozturk@togg.com.tr (Halil Ibrahim Ozturk), ozsel.kilinc@togg.com.tr (Ozsel Kilinc), atemizel@metu.edu.tr (Alptekin Temizel)

1. Introduction

In autonomous driving, scenes primarily consist of two types of entities: dynamic and stationary. Dynamic entities encompass objects capable of movement and interaction, such as vehicles (cars, bicycles, motorcycles, trucks), and pedestrians. Stationary entities, on the other hand, are immobile objects and abstract structures such as lane markings, crosswalks, road signs, traffic lights, barriers, and road surfaces, that regulate traffic and ensure the safe and orderly movement of dynamic entities. Furthermore, abstract stationary entities -such as lanes and centerlines- are defined in relation to other stationary objects or specific rules of the driving environment.

For a fully autonomous driving system, it is essential not only to detect stationary entities but also to understand their interrelationships. The challenge of identifying and understanding these connections between stationary objects is referred to as the *road topology* problem. For instance, multiple lanes may converge into one or a few, or a single lane may split into several. This complexity increases at intersections, where numerous lanes interact. Additionally, some traffic lights control only specific lanes. Accurately localizing, categorizing, and understanding the relationships between stationary entities are essential for downstream tasks such as planning and control in autonomous driving systems.

An alternative solution to this problem is the utilization of High-Definition Maps (HDMaps), which provide pre-computed maps of stationary entities. However, HDMaps are expensive to produce, cover only limited geographic areas, and cannot reflect recent changes on the road, requiring continuous updates. Furthermore, errors introduced by the Global Navigation Satellite System (GNSS) receiver on the vehicle can lead to localization inaccuracies. These inaccuracies may cause discrepancies between the actual position of the vehicle and the HDMap data, potentially resulting in drifts in the map-based guidance system. In response to these challenges, automatic HDMap construction has gained significant attention in recent years for two primary reasons [1, 2, 3, 4]. First, it can reduce reliance on HDMaps for autonomous driving. Second, it lowers the cost of creating and maintaining HDMaps, thereby minimizing the need for human effort.

In the context of polyline structures, such as centerlines and lane dividers, the application of standard cross attention [4, 5], deformable cross attention

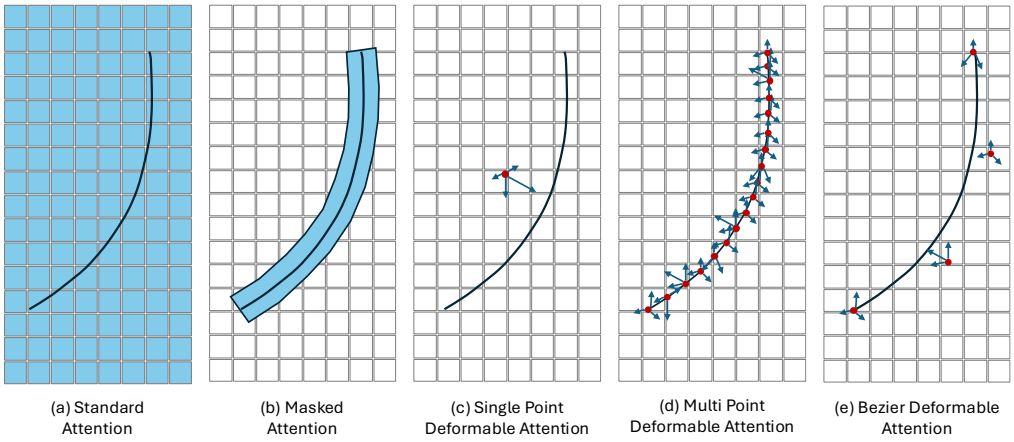


Figure 1: Comparison of various cross-attention mechanisms within the decoder architecture for polyline structures.

[6, 7, 3, 8, 9, 2, 10, 11, 12, 13, 14, 15], and masked cross attention [16, 17, 11, 18] is common. Masked attention [19], as illustrated in Figure 1b, requires tuning hyperparameters for polyline width and uses a thresholding mechanism to differentiate between foreground and background. Moreover, to prevent the failure of the attention mechanism, a foreground check is performed for every query, increasing complexity and reducing deployment efficiency. In contrast, deformable attention [20] can be implemented in two primary ways. The first, Single-Point Deformable Attention (SPDA), shown in Figure 1c, limits attention for each polyline instance to a single point, typically a learnable query embedding or the center point of the polyline's bounding box [2, 6, 7]. However, this method restricts attention to a local scope, making it unsuitable for elongated, thin polyline structures.

The second method for implementing deformable attention, Multi-Point Deformable Attention (MPDA), as shown in Figure 1d, involves distributing attention around every predicted dense lane point [3, 21, 8, 14, 22]. From the MPDA perspective, there are two primary approaches: point query-based methods and instance query-based methods. Point query-based methods are inherently complex because each query represents a single point, and increasing the number of points per instance proportionally increases the complexity [3, 23, 8, 17, 24]. Conversely, instance query-based methods are more efficient [9, 21]. However, a gap in the literature exists, as instance query methods utilizing Bezier control points [7, 6] have not applied MPDA,

relying instead on SPDA.

This study focuses on enhancing centerline detection performance within the broader context of road topology understanding. First, the integration of MPDA into Bezier keypoint-dependent transformer decoder structures is introduced. This adaptation is considered crucial because, unlike other dense polyline prediction methods, the Bezier representation [9] [21], leverages its inherently compact nature to improve the computational efficiency of polyline prediction. By incorporating MPDA into the Bezier keypoint-dependent transformer decoder, the method can more effectively handle elongated and thin polyline structures.

To further optimize the balance between performance and computational complexity, Bezier Deformable Attention (BDA) is introduced. As illustrated in Figure 1e, this method generates deformable attention around predicted Bezier points for centerline prediction. BDA significantly outperforms SPDA while maintaining a negligible performance difference. The improvement, as well as the reason for not observing an increase in computational complexity, is attributed to the use of control points across different attention heads, rather than relying on a single point to drive all attention heads. Additionally, SPDA requires learning an additional regression target for the center of the bounding box of the centerline. Furthermore, BDA achieves slightly better performance than the MPDA adaptation to the Bezier concept, with slightly less computational overhead. Unlike MPDA, BDA eliminates the need to convert Bezier keypoints into multiple polyline points within each transformer decoder layer, thereby reducing computational complexity.

In addition, consistent with the findings of the TopoMaskV2 study [18], the instance-mask formulation is employed to enhance the overall performance of road topology. However, unlike the direct approach used in TopoMaskV2, TopoBDA adopts an indirect application to reduce post-processing requirements. First, it has been demonstrated that incorporating the instance-mask formulation as an auxiliary loss significantly benefits the Bezier head. Second, during the Hungarian matching, using a Mask-L1 mix matcher [25], instead of a pure L1 matcher, has proven to be superior. These proposed mechanisms underscore the effectiveness of the indirect instance-mask formulation in improving road topology performance.

Sensor fusion has shown significant benefits in various domains, including 3D lane detection [26, 27] and HDMap element prediction [28, 29, 3, 2, 1]. Despite these advancements, there remains a notable gap in the literature regarding its application to road topology understanding. Previous studies

have primarily focused on the performance gains of SDMap [30, 31], without exploring the potential of sensor fusion in this context. Our research is the first to investigate and comprehensively evaluate the effects of sensor fusion, utilizing both lidar and radar data for road topology understanding. Additionally, we analyze the benefits of integrating SDMap with lidar and camera sensors, highlighting the novel combination of lidar and SDMap, which has not been explored in the literature.

Auxiliary one-to-many set prediction loss strategy, adapted from hybrid matching techniques [32], is implemented for HDMap element prediction [23] and shown to improve convergence and performance without increasing the inference complexity. While, also employed in road topology understanding problem [7, 18], its quantitative benefits in this context have not been fully explored. Therefore, this research provides a comprehensive analysis of its impact on the proposed TopoBDA architecture.

With the inclusion of BDA, instance mask formulation, and one-to-many set prediction loss, TopoBDA achieves state-of-the-art results in the camera-only benchmark for both OpenLane-V1 and OpenLane-V2 datasets. Furthermore, when utilizing multi-modal data, TopoBDA attains state-of-the-art results in OpenLane-V2. The contribution of this study is detailed in Supplementary Section S.8. To summarize, the key innovations and contributions of this work are as follows:

- **Multi-Point Deformable Attention (MPDA)**: The performance of centerline detection and road topology understanding is enhanced by the novel adaptation of MPDA to Bezier keypoint-dependent transformer decoders.

- **Bezier Deformable Attention (BDA)**: A novel attention mechanism utilizing Bezier control points is introduced. It has been demonstrated that BDA significantly improves the performance of centerline detection and road topology understanding while incurring negligible computational complexity overhead.

- **Instance Mask Formulation**: An instance mask formulation is incorporated as an auxiliary loss alongside the Mask-L1 mix matcher, improving the overall performance in road topology understanding.

- **Multi-Modal Fusion**: Lidar and radar data are utilized for the first time specifically for road topology understanding. Additionally, the

fusion of lidar with SDMap is analyzed, demonstrating the benefits of integrating multi-modal data. By fusing camera, lidar, and SDMap data, state-of-the-art results are achieved.

- **Auxiliary One-to-Many Set Prediction Loss**: The auxiliary one-to-many set prediction loss strategy from the existing literature is adapted for the road topology understanding and experimentally evaluated for the first time.

2. Related Work

This section summarizes the literature on three key aspects of autonomous driving and advanced driver-assistance systems (ADAS): *Lane Divider Detection*, *HDMap Element Prediction*, and the *Road Topology Problem and Centerline Concept*. Lane Divider Detection focuses on accurately identifying lane boundaries to ensure safe and reliable vehicle navigation. HDMap Element Prediction, which also encompasses lane divider detection, involves forecasting the presence and attributes of high-definition map elements. These studies often utilize multi-camera setups that provide 360-degree coverage, which is essential for precise localization and path planning. Additionally, the Road Topology Problem and Centerline Concept, both integral to HDMap Element Prediction, address the challenges of understanding and representing the road network's structure. Together, these components form the backbone of contemporary research aimed at enhancing the safety and efficiency of autonomous vehicles.

2.1. Lane Divider Detection

Lane divider detection methods can be categorized into two primary subcategories: perspective view methods and 3D lane divider methods.

2.1.1. Perspective View Methods

Perspective view (PV) methods focus on detecting lane dividers from the PV and projecting them onto the ground using a homography matrix under the flat surface assumption. Despite the presence of various lane divider instances, semantic approaches are practical and effective, assuming a constant number of lane divider instances (e.g., 1^{st} left, 2^{nd} left, 1^{st} right, 2^{nd} right lane dividers). SCNN [33] adheres to this methodology and introduces a module that sequentially processes the rows and columns of the feature

map. UFLD [34] transitions the formulation from pixel-based to grid-based, proposing row-based and column-based anchor formulations to enhance inference speed. LaneATT [35], inspired by object detection studies, devises an anchor concept specifically for lanes. The emergence of the CurveLanes dataset [36] has led to the adoption of instance-segmentation-based methods for the PV domain. CondLaneNet [37] introduces lane-specific methodologies such as offset prediction and row-wise formulation on top of the instance mask formulation in the PV domain. PolyLaneNet [38] and BezierLaneNet [39] employ polynomial and Bezier curve representations, respectively, to reduce post-processing efforts and improve curve learning.

2.1.2. 3D Lane Divider Methods

Recent studies have shifted towards directly predicting the 3D locations of lane dividers with the introduction of 3D lane divider datasets such as [40], OpenLane [41], and Apollo 3D Synthetic Lane [42]. This approach addresses the limitations of PV methods, specifically the flat world assumption due to the absence of depth information. Persformer [41] utilizes a deformable attention-based decoder with Inverse Perspective Mapping (IPM) and formulates a 3D anchor concept. CurveFormer [10] employs a sparse query design with a deformable attention mechanism and predicts polynomial parameters in the BEV domain. BEV-LaneDet [43] uses a keypoint concept with predicted offsets and groups the keypoints of the same lane instance with an embedding concept. PETRV2 [5] extends its sparse query design for lane detection. M2-3DLaneNet explores the benefits of integrating lidar and camera sensors. Another study [44] follows an instance-offset formulation in the BEV domain and aggregates offsets with a voting mechanism. LATR [8] introduces an end-to-end 3D lane detection framework that directly detects 3D lanes from front-view images using lane-aware queries and dynamic 3D ground positional embedding, significantly improving accuracy and efficiency over previous methods. GLane3D [45] utilizes a graph-based approach with keypoints and directed connections, achieving enhanced cross-dataset generalization performance.

2.2. HDMap Element Prediction

The prediction of High-Definition Map (HDMap) elements, such as lane dividers, road dividers, and pedestrian crossings, is essential for autonomous driving. Various methods have been developed to enhance the accuracy and efficiency of HDMap element prediction. HDMapNet [1] transforms Bird's

Eye View (BEV) semantic segmentation into BEV instance segmentation through extensive post-processing, leveraging predicted instance embeddings and directional information. VectorMapNet [2] introduces an end-to-end vectorized HD map learning pipeline that predicts sparse polylines from sensor observations, explicitly modeling spatial relationships between map elements, thus eliminating the need for dense rasterized segmentation and heuristic post-processing. Contrary to the autoregressive structure of VectorMapNet, MapTR [3] directly predicts points on polylines or polygons using a permutation-invariant Hungarian matcher, combining point query and instance query concepts, achieving real-time inference speeds and robust performance in complex driving scenes. InstaGraM [46] redefines polyline detection as a graph problem, where keypoints are vertices and their connections are edges, leveraging graph neural networks to enhance accuracy and robustness. MGMap [28] uses instance masks to generate map element queries and refine features with mask outputs, significantly improving performance over baseline methods. MapVR [29] rasterizes MapTR outputs and applies instance segmentation loss to address keypoint-based method limitations, enhancing performance without extra computational cost during inference. ADMap [47] employs instance interactive attention and vector direction difference loss to reduce point sequence jitter, enhancing map accuracy and stability. BeMapNet [16] models map elements as multiple piecewise curves using Bezier curves, eliminating the need for post-processing and achieving superior performance on existing benchmarks. StreamMapNet [21] explores the temporal aspects of HDMap element prediction, using temporal information as propagated instance queries and warped BEV features in a recurrent manner. MapTracker [22] formulates mapping as a tracking task, maintaining multiple memory latents to ensure consistent reconstructions over time, significantly outperforming existing methods on consistency-aware metrics. Recent advancements include PriorMapNet [48], which enhances online vectorized HD map construction by incorporating priors, and HIMap [11], which integrates point-level and element-level information to improve prediction accuracy. Additionally, the Multi-Session High-Definition Map-Monitoring System [49] employs machine learning algorithms to track and update map elements across multiple sessions, ensuring HD maps remain accurate and up-to-date. The Ultra-fast Semantic Map Perception [50] leverages both camera and LiDAR data to achieve real-time performance, featuring an orthogonal projection subspace for fast semantic segmentation and a Bayesian framework for enhanced global semantic fusion.

2.3. Road Topology Problem and Centerline Concept

Road topology refers to the interrelationships among lanes, as well as their connections to traffic lights and signs. However, using lane dividers for this problem is inefficient, as each lane requires two separate lane dividers. Consequently, the concept of centerlines has emerged as a more efficient and natural representation of lanes.

2.3.1. Centerline Concept

STSU [4] introduced a novel approach for extracting a directed graph of the local road network in bird's-eye-view (BEV) coordinates from a single onboard camera image, significantly improving traffic scene understanding. CenterLineDet [51] uses a transformer network to detect lane centerlines with vehicle-mounted sensors, effectively handling complex graph topologies such as lane intersections. LaneGAP [52] models lane graphs path-wise, preserving lane continuity and improving the accuracy of lane graph construction. MapTRV2 [23] enhances centerline prediction efficiency by treating lane centerlines as paths and incorporating semantic-aware shape modeling. SMERF [30] integrates Standard Definition (SD) maps by tokenizing map elements using a transformer encoder, and employs these tokens in the cross-attention mechanism of a transformer decoder to improve lane detection and topology prediction. In contrast, TopoSD [31] uses both map tokens and feature maps extracted from SD maps to enrich BEV features. SMART [53] uniquely leverages SD and satellite maps to learn robust map priors, enhancing lane topology reasoning for autonomous driving without relying on consistent sensor configurations. LaneSegNet [9] introduces the concept of lane segments, combining geometry and topology information to provide a comprehensive representation of road structures.

2.3.2. Road Topology

TopoNet [6] proposes a graph neural network architecture that models the relationships between centerlines and traffic elements. It incorporates prior relational knowledge to enhance feature interactions. TopoMLP [7] introduces an advanced pipeline for understanding driving topology by incorporating lane coordinates into the topology framework and using an L1 loss function to refine the interaction points. CGNet [54] focuses on preserving the continuity of centerline graphs and improving topology accuracy through modules like Junction Aware Query Enhancement and Bezier Space Connection. Topo2D [24] integrates 2D lane priors to improve 3D lane detection and topology

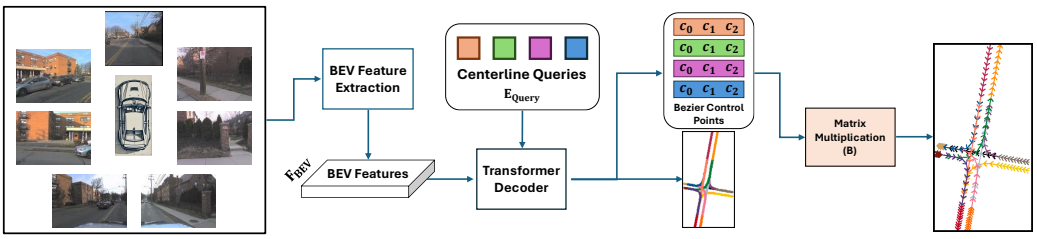


Figure 2: Overview of the TopoBDA architecture. The TopoBDA architecture is based on the instance query concept. The extracted BEV features from the multiple camera images are fed into the transformer decoder. The decoder outputs Bezier control points for each query, which are then converted into centerline instances via matrix multiplication. Additionally, each centerline query predicts instance masks, but only during training.

reasoning. TopoLogic [55] proposes managing lane topology relationships by combining geometric lane distance with similarity-based topology relationships. TopoFormer [56] introduces a lane aggregation layer that leverages geometric distance in driving self-attention, along with a counterfactual intervention layer to improve reasoning by considering alternative scenarios and their causal impacts.

3. Methodology

TopoBDA (Figure 2) starts by extracting Bird's Eye View (BEV) features from multi-camera 360-degree imagery. A transformer decoder then processes these BEV features within its cross-attention mechanism using a sparse query approach. In this approach, each query corresponds to a centerline instance rather than individual points, which improves computational efficiency. Subsequently, the decoder predicts Bezier control points for each instance query which are then converted into dense polyline points via matrix multiplication. The Bezier representation offers a compact and compressed formulation, reducing computational complexity at the regression heads [39, 16, 6]. Additionally, each centerline query predicts an instance mask for auxiliary loss, though these masks are not utilized during inference.

In the TopoBDA architecture, the BEV feature extraction (Figure 2) starts with a set of N images, $\{\mathbf{I}_i\}_{i=1}^{N}$, each having a dimension of $H_I \times W_I \times 3$, with H_I and W_I representing the height and width, respectively. These images are first converted into perspective view features, $\{\mathbf{F}_{PV_i}\}_{i=1}^{N}$, using a feature extraction function f_{PV}, such that $\mathbf{F}_{PV_i} = f_{PV}(\mathbf{I}_i)$ and $\mathbf{F}_{PV_i} \in \mathbb{R}^{H_{PV} \times W_{PV} \times C_{PV}}$. These perspective view features are then aggregated and

projected into a single Bird's Eye View (BEV) feature map, $\mathbf{F}_{BEV}$, using a projection function f_{BEV}. This function can be implemented via Lift-Splat-Shoot (LSS) [57, 58, 59], transformers [60, 61, 62, 63], or other projection techniques [64, 65, 66]. This projection is defined as $\mathbf{F}_{BEV} = f_{BEV}(\{\mathbf{F}_{PV_i}\}_{i=1}^{N})$, where $\mathbf{F}_{BEV} \in \mathbb{R}^{H_{BEV} \times W_{BEV} \times C_{BEV}}$. This methodology effectively captures and transforms spatial features into the BEV space, facilitating further analysis and processing within the TopoBDA framework.

Next, the extracted BEV features $\mathbf{F}_{BEV}$ are fed into the transformer decoder, where they are used as the value input in the cross-attention mechanism. Each query, corresponding to a single centerline instance, interacts with $\mathbf{F}_{BEV}$, and Bezier control points are predicted at each decoder layer. The attention mechanism in TopoBDA is based on Bezier Deformable Attention (BDA) and it is guided by these predicted Bezier control points. Further details of this mechanism are provided in Section 3.1 and the overall structure of the TopoBDA transformer decoder is provided in Section 3.2.

In the transformer decoder of the TopoBDA, an auxiliary instance mask formulation is also employed (Figure 2). In this formulation, each centerline query predicts not only Bezier control points but also a mask probability map for each instance. This indirect utilization enhances the performance of centerline predictions generated from the Bezier control points. Additionally, replacing the pure L1 matcher with a Mask-L1 mix matcher in the Hungarian matcher process further improves accuracy. The details of the instance mask formulation are provided in Section 3.3. This section also introduces the multi-modal data fusion strategy (Section 3.4) and the auxiliary one-to-many set prediction loss strategy (Section 3.5), both of which are integral to the TopoBDA study.

3.1. Towards Bezier Deformable Attention

This section explains the transition from Single-Point Deformable Attention (SPDA) to Multi-Point Deformable Attention (MPDA) for methodologies based on Bezier keypoint regression (or Bezier control point regression). This transition replaces traditional deformable attention heads with dense polyline points obtained via matrix multiplication on Bezier control points (Figure 3 and 4a).

Next, the concept of Bezier Deformable Attention (BDA) is introduced. BDA enhances the deformable attention mechanism by using Bezier control points. In this approach, traditional attention heads are substituted with the control points of a polyline curve, facilitating more flexible and adaptive

attention (Figure 3). In each layer, the predicted control points are utilized to guide the Bezier Deformable Attention process (Figure 4b).

3.1.1. Bezier Curve Representation

A Bezier curve is a parametric curve frequently used in computer graphics. It is defined by a set of control points, and the curve is a linear combination of these points weighted by Bernstein polynomials.

A Bezier curve $\mathbf{S}(t)$ of order N is defined by $N+1$ control points $\mathbf{C} = \{\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_N\}$, as shown in Eq. (1), where $B_{n,N}(t)$ are the Bernstein basis polynomials of degree N which are obtained as in Eq. (2).

$$\mathbf{S}(t) = \sum_{n=0}^{N} B_{n,N}(t)\mathbf{c}_n, \quad 0 \leq t \leq 1. \tag{1}$$

$$B_{n,N}(t) = \binom{N}{n} t^n (1-t)^{N-n}. \tag{2}$$

The control points $\mathbf{c}_n$ determine the shape of the Bezier curve, and the Bernstein polynomials $B_{n,N}(t)$ provide a smooth interpolation between these points.

3.1.2. Deformable Attention Mechanism

The deformable attention mechanism enhances traditional attention by adaptively sampling reference points with learned offsets, enabling greater flexibility. Single Point Deformable Attention (SPDA) mechanism is defined as in Eq. (3).

$$\text{SPDA}(\mathbf{q}, \mathbf{V}, \mathbf{p}) = \sum_{m=1}^{M} \sum_{k=1}^{K} A_{m,k} \mathbf{W}_m \mathbf{V}(\mathbf{p} + \Delta\mathbf{p}_{m,k}), \tag{3}$$

where $\mathbf{q}$ is the query, $\mathbf{V}$ is the value matrix, $A_{m,k}$ are the attention weights, $\mathbf{W}_m$ are learnable weight matrices, $\mathbf{p}$ is the reference point, $\Delta\mathbf{p}_{m,k}$ are the offsets, M is the number of attention heads, K is the number of sampling points per attention head. In this mechanism, the query $\mathbf{q}$ attends to the value matrix $\mathbf{V}$ at positions determined by the reference point $\mathbf{p}$ and the learned offsets $\Delta\mathbf{p}_{m,k}$.

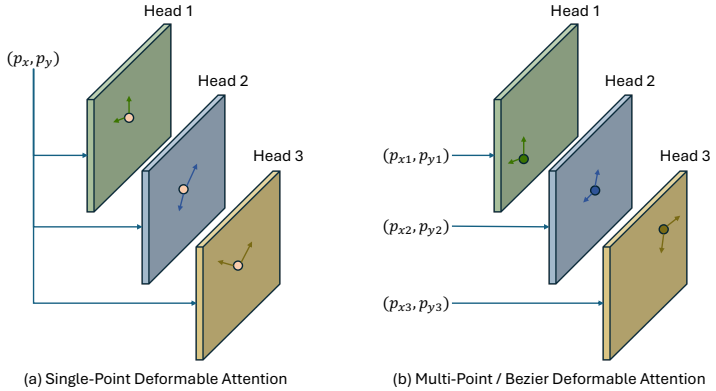


Figure 3: Comparison of Single-Point Deformable Attention (SPDA) with Multi-Point (MPDA) and Bezier (BDA) Deformable Attention. MPDA and BDA share the same underlying mechanism but differ in the selection of multiple reference points (p_x, p_y).

3.1.3. Adaptation of Bezier Regression Methods to Multi-Point Deformable Attention (MPDA)

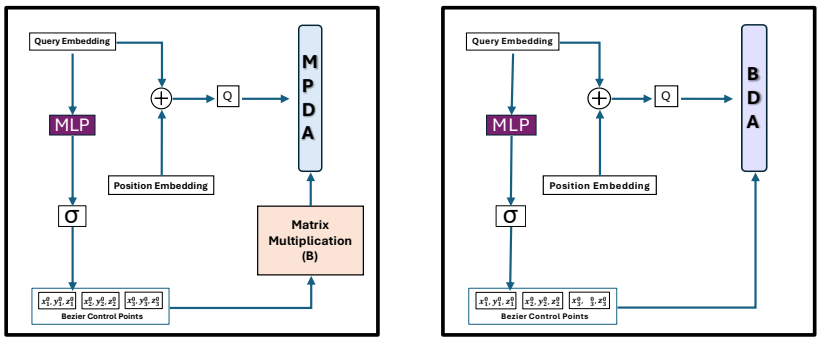
This methodology involves extracting $L + 1$ polyline points $\mathbf{P} = \{\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_L\}$ from the control points $\mathbf{C} = \{\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_N\}$ through Bezier extraction process. These polyline points are then utilized in the deformable attention mechanism. This approach replaces the traditional multi-head attention mechanism, which operates on a single reference point, with a more flexible and adaptive mechanism. Instead of using multiple heads M on a single point $\mathbf{p}$, this method employs a single head for each of the dense polyline points $\mathbf{p}_l$ as shown in Figure 3. By utilizing multiple points distributed along the polyline, this technique captures the thin, elongated characteristics of polylines, thereby enhancing the attention mechanism's ability to model complex patterns and dependencies.

The Multi-Point Deformable Attention (MPDA) mechanism is defined as:

$$\text{MPDA}(\mathbf{q}, \mathbf{V}, \mathbf{P}) = \sum_{l=0}^{L} \sum_{k=1}^{K} A_{l,k} \mathbf{W}_l \mathbf{V}(\mathbf{p}_l + \Delta \mathbf{p}_{l,k}), \tag{4}$$

where $L + 1$ is the number of polyline points, K is the number of sampling points for each polyline point, $\mathbf{p}_l$ represents the polyline points extracted from the Bezier control points and members of $\mathbf{P}$ such that $\mathbf{P} = \{\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_L\}$.

The polyline points $\mathbf{p}_l$ can be extracted from the control points using the



(a) Multi-Point Deformable Attention (MPDA) (b) Bezier Deformable Attention (BDA)

Figure 4: Comparison of Multi-Point Deformable Attention (MPDA) and Bezier Deformable Attention (BDA): MPDA necessitates an additional matrix multiplication block within each transformer decoder. Despite their different input utilizations as reference points, the mechanisms of MPDA and BDA blocks are fundamentally the same.

following formula, which is derived from Eq. (1) and Eq. (2):

$$\mathbf{p}_l = \sum_{n=0}^{N} \binom{N}{n} t_l^n (1-t_l)^{N-n} \mathbf{c}_n, \tag{5}$$

for $l = 0, 1, \ldots, L$, where t_l are uniformly spaced within the interval $[0, 1]$. Eq. (5) demonstrates that adapting MPDA to the Bezier keypoint concept necessitates converting Bezier control points into polyline points. This conversion is implemented through matrix multiplication in each decoder layer, as illustrated in Figure 4a. The detailed process of this implementation is provided in Supplementary Section S.1 of the supplementary materials.

3.1.4. Bezier Deformable Attention

BDA enhances the traditional multi-head attention mechanism by replacing the single-point focus $\mathbf{p}$ with control points $\mathbf{c}_n$ of the Bezier curve. From this perspective, the inherent mechanism of BDA is the same with MPDA as shown in Figure 3. Unlike MPDA, which relies on dense polyline points, BDA directly uses these control points as reference points within the deformable attention mechanism. This approach eliminates the need for converting control points to polyline points in each decoder layer (Figure 4), reducing computational complexity slightly. By predicting Bezier control points and focusing attention around them, BDA improves the learning process, leading to more effective and accurate predictions. From a theoretical computational

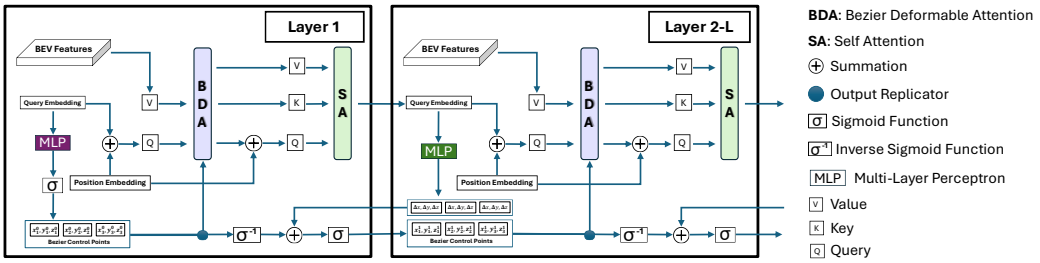


Figure 5: This figure visualizes the layers of TopoBDA, each driven by Bezier Deformable Attention (BDA) using control points predicted through iterative refinement. Note that iterative refinement is not applicable to the first layer, which uses direct prediction.

complexity standpoint, there is no difference between the attention mechanism of SPDA and BDA, as the only modification lies in the reference points that guide attention across different heads. However, SPDA requires learning an additional reference point learning mechanism in polyline detection structures, which increases the complexity slightly.

The Bezier Deformable Attention (BDA) mechanism is defined as:

$$\mathrm{BDA}(\mathbf{q}, \mathbf{V}, \mathbf{C}) = \sum_{n=0}^{N} \sum_{k=1}^{K} A_{n,k} \mathbf{W}_n \mathbf{V}(\mathbf{c}_n + \Delta \mathbf{p}_{n,k}), \tag{6}$$

where $N+1$ is the number of control points, K is the number of sampling points for each control point, $\mathbf{c}_n$ represents the control points and members of $\mathbf{C}$ such that $\mathbf{C} = \{\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_N\}$. In this formulation, each control point $\mathbf{c}_n$ acts as a head in the attention mechanism, allowing the model to attend to different parts of the input sequence based on the shape of the Bezier curve. This approach provides a more flexible and adaptive attention mechanism that can better capture the spatial structure of polylines.

To summarize, the key distinction between SPDA, MPDA, and BDA lies in how reference points are utilized across attention heads. From a computational complexity perspective, these mechanisms are nearly equivalent when the number of attention heads is the same. However, SPDA introduces slight overhead due to its additional reference point learning, while MPDA incurs cost from converting control points into polyline points. BDA avoids this by directly using Bezier control points.

3.2. Bezier Deformable Attention Based Transformer Decoder

The overview of the general pipeline of each layer in the transformer decoder of TopoBDA is shown in Figure 5. The step-wise operation detail of the TopoBDA decoder is also shown in Supplementary Algorithm 1 in Section S.7. The iterative refinement-based Bezier control points prediction process involves several steps:

1. **Control Points Generation (First Layer):** The control points of each centerline instance are obtained in a normalized format using a Multi-Layer Perceptron (MLP) and a sigmoid function:

$$\mathbf{C}_{norm}^{(1)} = \sigma(\text{MLP}_{B}^{(1)}(\mathbf{E}_{query}^{(1)})), \tag{7}$$

   where $\mathbf{C}_{norm}^{(1)} = \{\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_N\}$, and each $\mathbf{c}_i$ consists of normalized (scaled to [0,1]) x, y, and z coordinates. $\mathbf{E}_{query}^{(1)}$ is the query embedding at the first layer.
2. **Bezier Deformable Attention (BDA):** The predicted control points are fed into the BDA layer to guide attention. BDA defines the query as the sum of positional embedding and query embedding, while the BEV features serve as the value:

$$\begin{aligned} \mathbf{Q}_{BDA}^{(l)} &= \mathbf{E}_{query}^{(l)} + \mathbf{P}^{(l)}, \\ \mathbf{A}_{BDA}^{(l)} &= \text{BDA}(\mathbf{Q}_{BDA}^{(l)}, \mathbf{F}_{BEV}, \mathbf{C}_{norm}^{(l)}), \end{aligned} \tag{8}$$

   where $\mathbf{Q}_{BDA}^{(l)}$ is the combined query embedding and positional embedding at layer l, and $\mathbf{A}_{BDA}^{(l)}$ is the output of the BDA layer.
3. **Self-Attention:** Self-attention follows the output of BDA and utilizes the same positional embedding:

$$\mathbf{A}_{SA}^{(l)} = \text{SelfAttention}(\mathbf{A}_{BDA}^{(l)}, \mathbf{P}^{(l)}), \tag{9}$$

   where $\mathbf{A}_{SA}^{(l)}$ is the output of the self-attention layer at layer l. The output of the self-attention layer, $\mathbf{A}_{SA}^{(l)}$, serves as the query embedding $\mathbf{E}_{query}^{(l+1)}$ for the next layer.
4. **Iterative Process in Subsequent Layers:** In subsequent layers, the process repeats, but with MLP layers predicting Bezier control points

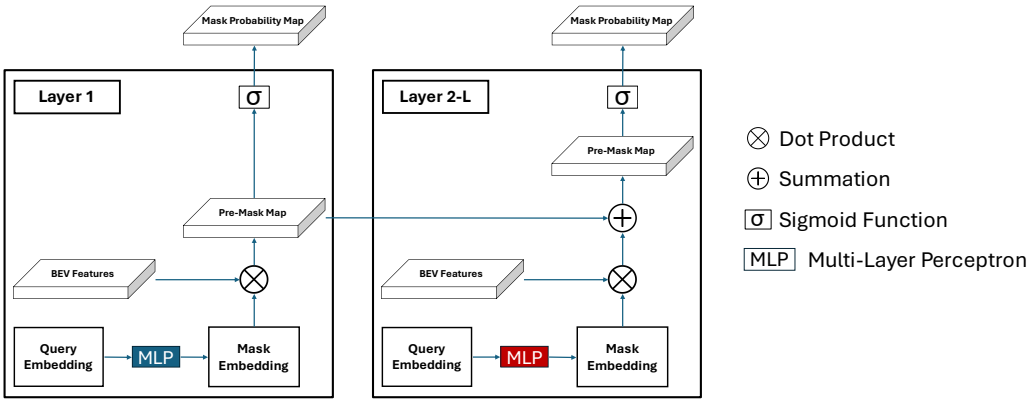


Figure 6: The implementation of instance mask formulation in the TopoBDA decoder. Query embeddings are converted to mask embeddings by MLP layers. The dot product between BEV features and mask embeddings generates a pre-mask map, which is iteratively summed across layers to produce the final pre-mask map

differences. These differences are summed in the inverse sigmoid domain before being transformed back:

$$\begin{aligned} \Delta \mathbf{C}^{(l)} &= \mathrm{MLP}_B^{(l)}(\mathbf{E}_{query}^{(l)}), \\ \mathbf{C}_{inv\_sigmoid}^{(l)} &= \sigma^{-1}(\mathbf{C}_{norm}^{(l-1)}) + \Delta \mathbf{C}^{(l)}, \\ \mathbf{C}_{norm}^{(l)} &= \sigma(\mathbf{C}_{inv\_sigmoid}^{(l)}), \end{aligned} \tag{10}$$

where $\Delta \mathbf{C}^{(l)}$ represents the predicted Bezier control points differences at layer l, $\mathbf{C}_{inv\_sigmoid}^{(l)}$ is the sum in the inverse sigmoid domain, and $\mathbf{C}_{norm}^{(l)}$ are the updated control points.

3.3. The Indirect Benefits of Instance Mask Formulation

The TopoBDA architecture is further enhanced by integrating instance mask formulation in two key areas. First, instance mask loss is used as an auxiliary loss. In each transformer decoder layer, instance masks are predicted for each centerline instance. Second, the Mask-L1 mix matcher is introduced in the Hungarian matcher, combining mask and L1 losses to improve the accuracy of predictions by leveraging both mask and L1 bipartite-matching mechanisms.

3.3.1. Instance Mask Formulation as an Auxiliary Loss

To achieve this, centerlines are converted into masks using a parameter W in the BEV domain. Utilizing the generated ground truth instance masks, each centerline query in the transformer decoder predicts both Bezier control points and a mask probability map for each centerline instance (Figure 6). Supplementary Algorithm 1 in Section S.7 demonstrates the algorithmic details about the mask probability map generation within the TopoBDA transformer decoder.

The mask probability map generation in the decoder follows several steps:

1. **Mask Embedding Generation:** The mask embeddings are generated from the query embeddings using an MLP:

$$\mathbf{E}_{mask}^{(l)} = \mathrm{MLP}_{M}^{(l)}(\mathbf{E}_{query}^{(l)}), \tag{11}$$

   where $\mathbf{E}_{mask}^{(l)}$ is the mask embedding at layer l and $\mathbf{E}_{query}^{(l)}$ is the query embedding at layer l.
2. **Pre-mask Map Generation:** A dot product is applied between the BEV features ($\mathbf{F}_{BEV}$) and the mask embedding to generate the pre-mask map:

$$\mathbf{P}_{mask}^{(l)} = \mathbf{F}_{BEV} \cdot \mathbf{E}_{mask}^{(l)}, \tag{12}$$

   where $\mathbf{P}_{mask}^{(l)}$ is the pre-mask map at layer l.
3. **Summation of Pre-mask Maps:** In consecutive layers, pre-mask maps are summed to refine the mask embeddings at each layer:

$$\mathbf{P}_{mask}^{(l)} = \mathbf{P}_{mask}^{(l-1)} + \mathbf{P}_{mask}^{(l)}, \tag{13}$$

   where $\mathbf{P}_{mask}^{(l)}$ is the pre-mask map at layer l, $\mathbf{P}_{mask}^{(l-1)}$ is the pre-mask map from the previous layer and $\mathbf{P}_{mask}^{(0)}$ is initialized to zero.
4. **Mask Probability Map:** The sigmoid function is used to obtain the mask probability maps from the pre-mask maps:

$$\mathbf{M}_{prob}^{(l)} = \sigma(\mathbf{P}_{mask}^{(l)}), \tag{14}$$

   where $\mathbf{M}_{prob}^{(l)}$ is the mask probability map at layer l and σ is the sigmoid function.

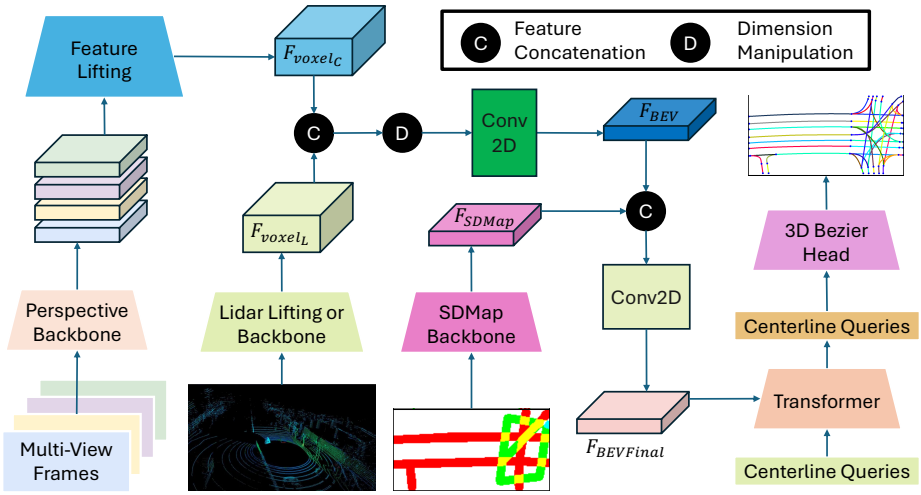


Figure 7: Sensor fusion pipeline in the TopoBDA architecture.

3.3.2. Mask-L1 Mix Matcher

The instance mask concept is employed not only in the main loss but also during the bipartite matching (Hungarian matcher). The loss function of the bipartite matching is defined as follows:

$$\mathcal{L}_{\text{l}} = \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}}. \tag{15}$$

When $\lambda_{\text{reg}} = 0$, it operates as a mask matcher, whereas when $\lambda_{\text{mask}} = 0$, it operates as an L1 matcher. When both λ_{reg} and λ_{mask} are non-zero, it is referred to as a Mask-L1 mix matcher. This hybrid matching approach, inspired by the Mask DINO framework [25], is adapted to enhance road topology understanding. Experimental results show that utilizing the instance mask formulation in both the main loss and bipartite matching (Hungarian matcher) improves performance. Furthermore, the Mask-L1 mix matcher accelerates the convergence of topology performance.

3.4. Fusion Methodology

Sensor fusion for road topology understanding has not been extensively explored in the literature. The sensor and SDMap fusion pipeline used in TopoBDA is illustrated in Figure 7. In TopoBDA, camera and lidar features are first concatenated in the voxel space to preserve spatial granularity. This approach differs from BEVFusion studies [67, 68, 69], which typically perform

fusion directly in the BEV space. The resulting multi-modal voxel features are merged by flattening the height and channel dimensions, followed by a 2D convolution to obtain BEV features. Although radar features are also fused in the voxel space, they are omitted from Figure 7 for clarity. SDMap fusion is performed separately in the BEV domain, where SDMap features are concatenated with the BEV features derived from other sensors. The mathematical formulation of the fusion pipeline, including voxelization, multi-modal concatenation, and SDMap integration, is detailed in Supplementary Section S.2 of the supplementary materials.

3.5. Auxiliary One-to-Many Set Prediction Loss Strategy

To improve training efficiency, an auxiliary one-to-many set prediction loss strategy [32, 23] is employed. This study, to the best of our knowledge, is the first to perform ablation experiments on this strategy for road topology understanding. Results show that this approach significantly enhances the architecture's ability to understand road topology.

The auxiliary one-to-many set prediction loss approach involves a smaller decoder and a larger decoder with shared weights. The smaller decoder uses the ground truth set directly, while the larger decoder employs a repeated ground truth set to increase the number of positive samples. During training, both decoders are utilized, while only the smaller decoder is employed during inference. The mathematical background of this loss strategy, including decoder sharing with masking, is provided in Supplementary Section S.3 of the supplementary materials.

4. Experimental Evaluation

This section provides a comprehensive analysis of the methodologies employed in this study. It commences with a detailed description of the datasets and metrics (Section 4.1). Subsequently, Section 4.2 presents an extensive experimental evaluation, encompassing an analysis of instance mask formulation, comparisons of various attention mechanisms, an examination of auxiliary one-to-many set prediction loss, an efficiency analysis of the proposed Bezier deformable attention, and a study on multi-modal fusion involving camera, radar, lidar, and SDMap. The results are juxtaposed with state-of-the-art methods on both OpenLane-V1 and OpenLane-V2.

Supplementary material provides additional details on the various loss functions (Section S.4), the implementation specifics of the training setup,

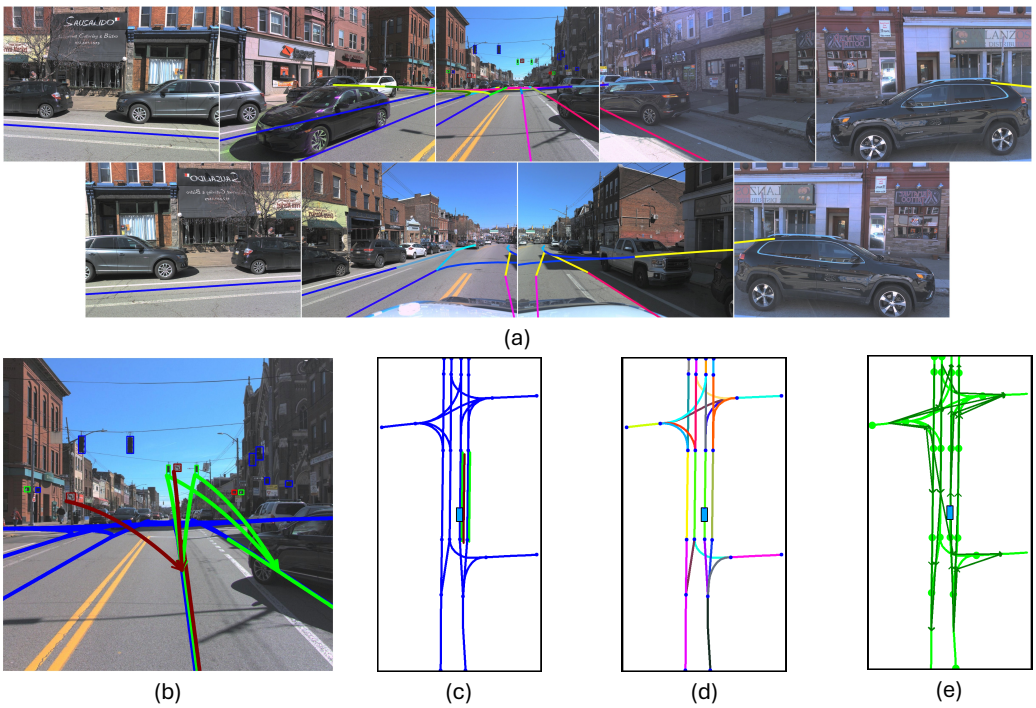


Figure 8: Perspective-view (PV) and bird's-eye-view (BEV) samples from the OpenLane-V2 dataset. (a) and (d) show centerline instances in PV and BEV domains, respectively, with each color representing a distinct instance. (b) and (c) illustrate centerlines with colors indicating topological relationships between centerlines and traffic elements in PV and BEV. (e) visualizes the topological relationships among different centerlines, where directed arrows indicate connectivity between centerlines.

dataset preprocessing and the proposed architecture (Section S.5), and further experiments (Section S.6) including view transformation, backbone variations, number of epochs, number of encoder and decoder layers, and efficient multi-scale implementation ablations.

4.1. Dataset and Metrics

4.1.1. Datasets

The OpenLane-V2 dataset [70] is a centerline detection and road topology understanding dataset and has two subsets: Subset-A and Subset-B. Subset-A is derived from the Argoverse 2 (AV2) [71] dataset, containing samples from six cities (e.g., Miami, Pittsburgh, Austin), collected using a seven-camera setup with the front camera positioned vertically. This subset contains 22,477

training samples, 4,806 validation samples, and 4,816 test samples, all with a resolution of 2048×1550 pixels. Multi-modal fusion studies are conducted with lidar data and SDMap, and no radar data is available in Subset-A.

Subset-B is based on the NuScenes [72] dataset and contains data collected from Boston and Singapore, using a six-camera setup. It includes 27,968 training, 6,019 validation, and 6,008 test samples, with an image resolution of 1600×900 pixels. Subset-B has a higher proportion of night and rainy scenes compared to Subset-A, but lacks ground height information. Sensor fusion studies are implemented with both radar and lidar data, but no available SDMap information in this subset. The details of the image pre-processing pipeline for both subsets are provided in Supplementary Section S.5.4.

Figure 8 presents PV and BEV images from a random sample in Subset-A of the OpenLane-V2 dataset. Due to the vertical positioning of the camera, the front view in the PV domain is cropped. As shown in subfigures (a) and (d), centerline instances are visualized in PV and BEV domains, respectively, with each color representing a distinct instance. Subfigures (b) and (c) highlight the topological relationships between centerlines and traffic elements using color-coded representations in both PV and BEV views. Subfigure (e) illustrates the topological relationships among different centerlines, where directed arrows indicate connectivity between them. More ground truth examples are shown in Supplementary Figure S1 in Section S.9.

The OpenLane-V1 dataset [41] is a comprehensive benchmark for 3D lane detection, derived from the Waymo Open dataset. It consists of 1000 segments with 200K frames, captured under diverse conditions at 1920×1280 resolution. This dataset provides diverse and challenging scenarios for evaluating lane detection algorithms, including a wide range of weather conditions, lighting variations, and road types.

4.1.2. Metrics

For the evaluation of centerlines and traffic elements in both Subset-A and Subset-B of OpenLane-V2, the considered area extends +50 to -50 meters forward and +25 to -25 meters sideways. Both centerline detection and traffic element recognition are evaluated using the Mean Average Precision (mAP) metric. True positive samples are identified using different distance measures depending on the task. For centerline detection, the Fréchet distance and Chamfer distance are used. The Fréchet distance accounts for both distance and directionality between predicted and ground truth centerlines, whereas the Chamfer distance only considers distance, disregarding directionality. For

traffic element recognition, the Intersection over Union (IoU) metric is used with a threshold of 0.75. The Fréchet distance thresholds are 1, 2, and 3 meters, while the Chamfer distance thresholds are 0.5, 1, and 1.5 meters. The mean AP of different thresholds is denoted as DET_l for Fréchet-based mAP and DET_{l_ch} for Chamfer-based mAP, and DET_t for IoU-based mAP.

In addition to these metrics, a specialized mAP metric is proposed to evaluate topology reasoning in the graph domain. This metric assesses the connectivity and relationships between centerlines and traffic elements. For an edge (connectivity) to be a true positive, both vertices must be correctly detected according to the Fréchet distance for centerlines and the IoU criteria for traffic elements. The topology scores are defined as TOP_{ll} for centerline topology and TOP_{lt} for centerline-traffic element topology.

$$\text{OLS} = \frac{1}{4}\bigg[\text{DET}_l + \text{DET}_t + f(\text{TOP}_{ll}) + f(\text{TOP}_{lt})\bigg]. \tag{16}$$

The evaluation framework uses the OpenLane-V2 Score (OLS) as the overall metric, which is calculated as the average of several task-specific metrics: Fréchet-based mAP for centerline prediction (DET_l), IoU-based mAP for traffic element prediction (DET_t), and the topological relationships metrics between centerlines and traffic elements (TOP_{ll} and TOP_{lt}). The evaluation metric pipeline has been updated from version 1.0 to 1.1 to address previously identified issues, as outlined in the TopoMLP study [7].

The potential shortcomings of the 1.1 baseline are discussed in the TopoMaskV2 study [18], and a modified baseline is proposed, which is referred to as version 1.1m in this study. While the metric itself remains unchanged, the topology scores are re-mapped such that $P(x) + 1 \times [P(x) > 0.05]$, where $P(x)$ represents the topology score. To ensure fair comparison with existing literature, state-of-the-art comparisons are conducted using V1.1, while all our ablations and hyperparameter evaluations are performed using the V1.1m baseline.

$$\text{OLS}_l = \frac{1}{3}\bigg[\text{DET}_l + \text{DET}_{l\_ch} + f(\text{TOP}_{ll})\bigg]. \tag{17}$$

To perform ablations focused exclusively on centerline prediction and centerline topology, we use the OLS_l metric Eq. (17), which excludes components associated with traffic elements. Since the architectural framework of TopoBDA is specifically designed to improve centerline prediction, it is

essential to develop metrics dedicated to centerline prediction and centerline topology prediction accuracy.

For 3D lane detection evaluation in the OpenLane-V1 dataset, the F1 metric is preferred, as it calculates the harmonic mean of precision and recall. A detection is considered a true positive if at least 75% of the compared points are within the predefined threshold of 1.5 meters or 0.5 meters from the ground truth lane dividers. The detection range for the metric is set from 3 to 103 meters in the forward direction and -10 to 10 meters in the lateral direction. A threshold of 40 meters is used to distinguish between close and far ranges, with lateral and height translation errors measured separately for these regions.

4.2. Experimental Evaluation

In this section, the experimental results are presented. The training setup, including optimizer configurations, learning rate schedules, batch size, architectural specifications, and hyperparameter settings, is described in detail in Section S.5 of the supplementary material. This section also includes the backbone configurations, transformer parameters, loss coefficients, and implementation details necessary for reproducibility.

The experiments are structured as follows. First, the impact of the instance-mask formulation is analyzed. This is followed by a comparative evaluation of various attention mechanisms and a conceptual analysis of the auxiliary one-to-many set prediction loss. Subsequently, the performance of sensor fusion and SDMap integration, as well as the efficiency of different attention mechanisms within decoder implementations, is assessed. Finally, the results are compared against state-of-the-art benchmarks on the OpenLane-V2 and OpenLane-V1 datasets.

The supplementary material section provides further insights into our experiments (Section S.6), including a comparative analysis of view transformation methods, a comparison of standard and efficient multi-scale implementations, a further efficiency analysis of attention types, the number of encoder and decoder layers, an impact of number of control points, an evaluation of the influence of different backbones, and impact of epochs and multi-modality on performance.

4.2.1. Analysis of Instance Mask Formulation

Table 1 presents the impact of indirect instance mask formulation on TopoBDA with the V1.1m metric baseline. These experiments utilize SwinB as

Table 1: Impact of indirect instance mask formulation on TopoBDA in Subset-A of OpenLane-V2 Metric with V1.1m Baseline. IMAL refers to the Instance Mask Auxiliary Loss and ML1M refers to the Mask-L1 Mix Matcher.

IMAL	**ML1M**	$\mathbf{DET_l}$	$\mathbf{DET_{l\_ch}}$	$\mathbf{TOP_{ll}}$	$\mathbf{OLS_l}$
		37.0	39.8	29.0	43.6
✓		<u>40.7</u>	<u>42.1</u>	<u>32.4</u>	<u>46.6</u>
✓	✓	**40.8**	**45.8**	**32.9**	**48.0**

the backbone of the architecture. The impact of other backbone architectures is shown in Supplementary Table S7 in Section S.6.6. The results show that incorporating the instance mask auxiliary loss (IMAL) significantly improves performance across various metrics, with a 3.7 points increase in DET_l score, and a 3.3 point increase in OLS_l (See Section 4.1 for the details of metrics and Eq. (17) for OLS_l).

Further enhancement with the mask-L1 mix matcher (ML1M) provides additional gains, with a 3.7-point increase in $\text{DET}_{\text{l\_ch}}$ and a 1.4-point increase in OLS_l. The improvements underscore the effectiveness of the instance mask auxiliary loss and the mask-L1 mix matcher, and validate their adoption for use in TopoBDA. For detailed information about the implementation of instance mask formulation, please refer to Section 3.3 and Supplementary Section S.7.

4.2.2. Comparison of Different Attention Mechanisms

In Table 2a, a comparative analysis of various attention mechanisms with the V1.1m metric baseline is presented, utilizing SwinB as the 2D backbone. The results show that deformable attention-based decoders significantly outperform both Standard Attention (SA) and Masked Attention (MA). The number of control points of all experiments is set to 4 to align with the literature [7] and for its practicality (See Supplementary Table S6 in Section S.6.5).

Specifically, Single-Point Deformable Attention (SPDA) outperforms MA by 1.5 points in OLS_l, but it performs the worst among the deformable attention baselines. MPDA4 outperforms SPDA by 3.3 points in OLS_l, highlighting the importance of applying the attention mechanism to different regions around the polyline instead of a single fixed center point. Notably, there is a negligible performance difference between 4-point (MPDA4) and 16-point (MPDA16) multi-point deformable attentions, with only a 0.1 point

Table 2: The left table presents a comparison of different attention mechanisms, Standard Attention (SA), Masked Attention (MA), Single Point Deformable Attention (SPDA), 4-point Multi-Point Deformable Attention (MPDA4), 16-point Deformable Attention (MPDA16), and Bezier Deformable Attention (BDA). The right table illustrates the impact of ground truth and query repetition (R) in the auxiliary one-to-many set prediction loss strategy, where R=0 indicates the absence of the auxiliary one-to-many set prediction loss. Both table results are on Subset-A of the Openlane-V2 dataset using the V1.1m Baseline.

(a) Attention Mechanisms

Attn.	**DET_l**	**DET_{l_ch}**	**TOP_{ll}**	**OLS_l**
SA	34.5	38.4	25.1	41.0
MA	35.8	40.2	26.9	42.6
SPDA	38.3	39.8	29.5	44.1
MPDA4	40.2	45.0	32.6	47.4
MPDA16	40.3	45.1	32.7	47.5
BDA	**40.8**	**45.8**	**32.9**	**48.0**

(b) Auxiliary One-to-many Set Prediction Loss

R	**DET_l**	**DET_{l_ch}**	**TOP_{ll}**	**OLS_l**
0	39.4	44.0	31.4	46.5
1	39.0	45.0	32.0	46.9
2	40.1	45.0	32.1	47.2
3	40.7	45.5	32.6	47.8
4	40.8	45.8	32.9	48.0
5	**41.0**	**45.9**	**33.1**	**48.1**

improvement in favor of MPDA16 in OLS_l. The 4-Point Bezier Deformable Attention (BDA), which employs 4 control points, surpasses both MPDA4 and MPDA16 by 0.7 points in DET_{l_ch} and 0.5 points in OLS_l. Despite the limited performance improvement, BDA achieves these results with reduced computational complexity, as it does not require converting Bezier control points to polyline points in each transformer decoder layer using matrix multiplication, unlike MPDA. A comparison of the runtimes of the attention mechanisms is provided in Table 4 of Section 4.2.5, a more detailed computational complexity analysis is presented in Table S3 of Supplementary Section S.6.3, and theoretical details of SPDA, MPDA, and BDA are explained in Section 3.1.

4.2.3. Auxiliary One-to-Many Set Prediction Loss: Conceptual Analysis

This study evaluates the impact of varying the number of repetitions (R) of ground truths and queries on the auxiliary one-to-many set prediction loss across the key metrics: DET_l, DET_{l_ch}, TOP_{ll}, and OLS_l. The results with the V1.1m metric baseline, summarized in Table 2b, are based on experiments conducted using the SwinB backbone. The findings indicate that increasing the number of repetitions (R) enhances performance across most metrics. Consequently, with this strategy, DET_l, DET_{l_ch}, TOP_{ll}, and OLS_l improve by up to 1.6, 1.9, 1.7, and 1.6 points, respectively. A detailed explanation of the auxiliary one-to-many set prediction loss strategy is provided in Section 3.5.

Table 3: Sensor Fusion and SDMap Integration Ablations in Subset-A and Subset-B of OpenLane-V2 with V1.1m Baseline. The configuration with the lidar encoder (SECOND) is marked with a dagger (†) [73].

Subset	Configuration	$\mathbf{DET_l}$	$\mathbf{DET_{l\text{-}ch}}$	$\mathbf{TOP_{ll}}$	$\mathbf{OLS_l}$
A	Camera	38.9	39.2	29.4	44.1
	Camera + SDMap	42.7	48.0	35.7	50.1
	Camera + Lidar	46.5	49.0	36.7	52.0
	Camera + Lidar (†)	47.3	51.2	37.3	53.2
	Camera + Lidar (†) + SDMap	**52.0**	**52.8**	**40.0**	**56.0**
B	Camera	45.1	45.1	35.6	49.9
	Camera + Radar	49.4	52.8	38.6	54.8
	Camera + Lidar	57.5	59.4	46.0	61.6
	Camera + Lidar (†)	**57.7**	**60.0**	**46.7**	**62.0**

4.2.4. Performance Analysis of Sensor Fusion and SDMap Integration

Table 3 summarizes the experimental results for multi-modal fusion ablations in the OpenLane-V2 V1.1m metric baseline, with ResNet50 as the 2D backbone. Performance is evaluated across the DET_l, $\text{DET}_\text{l-ch}$, TOP_ll, and OLS_l metrics.

The camera-only configuration serves as the baseline. In Subset-A, adding lidar significantly improves performance, with OLS_l increasing from 44.1 to 52.0. Using the lidar encoder (SECOND) [73] further boosts OLS_l to 53.2. Similarly, in Subset-B, integrating lidar with the camera results in substantial performance gains, with OLS_l increasing from 49.9 to 61.6 and the use of the lidar encoder further increasing to 62.0.

Integration of SDMap information with the camera in Subset-A also brings notable improvements, increasing OLS_l from 44.1 to 50.1. Using camera, lidar, and SDMap altogether yields the highest performance gains, achieving an OLS_l of 56.0. This demonstrates the complementary nature of lidar sensors and SDMap information, providing richer contextual data and enhancing the model's performance. Specifically, adding lidar to the camera + SDMap configuration improves OLS_l from 50.1 to 56.0, while adding SDMap to the camera + lidar configuration improves OLS_l from 53.2 to 56.0.

In Subset-B, integrating radar with the camera shows notable improvements, increasing OLS_l from 49.9 to 54.8. However, the combination of radar and lidar does not yield further improvements beyond the camera-lidar configuration, suggesting that lidar already encapsulates the benefits provided

Table 4: Decoder runtimes (in ms) for different attention types in Torch and ONNX. ONNX* indicates inference without auxiliary mask heads.

Runtime Type	**BDA**	**SPDA**	**MPDA4**	**MPDA16**	**SA**	**MA**
Torch	<u>18.47</u>	18.55	20.31	22.73	**15.96**	24.76
ONNX	**8.07**	8.19	<u>8.11</u>	8.20	11.03	33.41
ONNX*	**4.66**	4.70	<u>4.67</u>	4.75	9.50	23.09

by radar.

Figure 9 illustrates the advantages of integrating lidar and SDMap with camera sensors in the BEV domain. The corresponding camera views for these examples are shown in Supplementary Figure S1 of Section S.9. In Figure 9, inaccurate centerline detections are highlighted with circular regions in the BEV images of specific modalities (camera, lidar, and SDMap) or their combinations. 'GT' denotes ground truth, while 'C', 'SD', and 'L' represent the camera, SDMap, and lidar, respectively. In the comparison figures, green centerlines denote the ground truth, while red centerlines represent the predictions, overlaid for easier comparison. Lidar proves advantageous for detecting and localizing long-distance centerlines, while SDMap improves centerline localization and orientation, and is beneficial for occluded and unseen areas. When used together, SDMap and lidar minimize errors, offering the best performance.

Overall, these results underscore the significant benefits of multi-modal fusion, particularly the integration of lidar and SDMap, in enhancing road topology understanding. The benefits of multi-modality in higher epoch regimes are shown in Supplementary Table S8 (Section S.6.7).

4.2.5. Efficiency Analysis of Attention Mechanisms in Decoder Implementations

The time complexity analysis of the decoder for different attention mechanisms is presented in Table 4. This analysis was conducted using the NVIDIA RTX A6000 GPU within the NVIDIA NGC PyTorch 24.10 container. To ensure a fair comparison, the number of cross attention heads for all attention mechanisms is set to 4—matching the 4 control points of BDA—except for MPDA16, which uses 16 heads. For this analysis, an efficient multi-scale implementation [19], which processes different feature scales successively in each decoder layer in a round-robin fashion, is employed. A computational and performance comparison between this efficient strategy and the

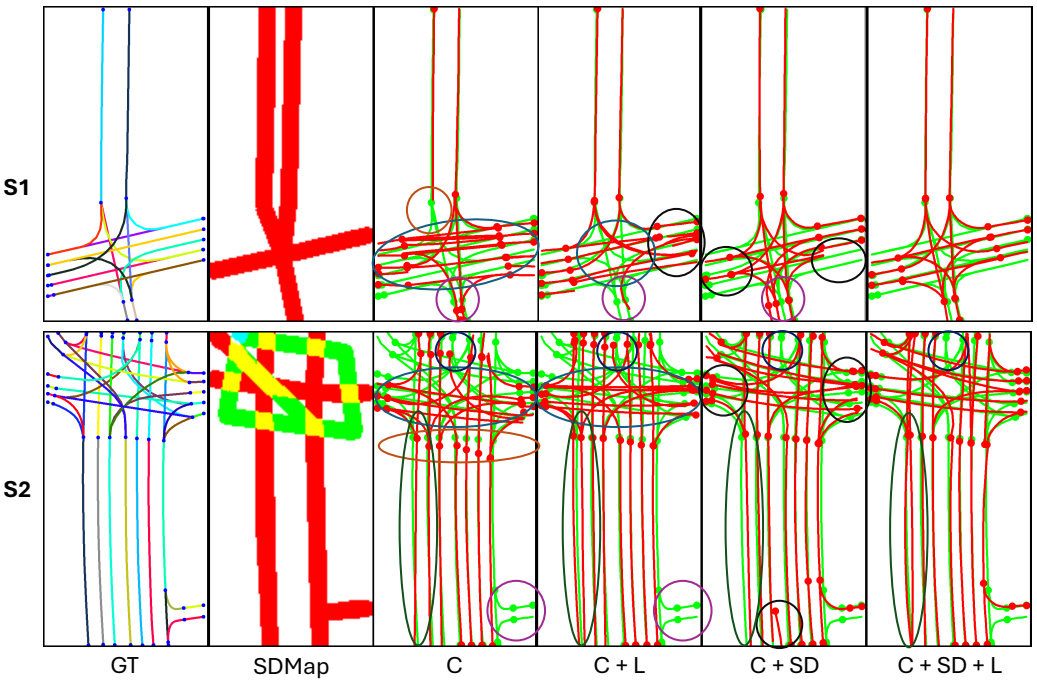


Figure 9: Visual demonstration in the BEV domain showing the impact of lidar and SDMap additions in Subset-A of the OpenLane-V2 dataset. C, SD, and L represent the camera, SDMap, and lidar, respectively. Green polylines indicate the ground truth, and red polylines represent predictions. The circular regions highlight the inaccurate regions compared to other reference BEV images.

standard multi-scale implementation is provided in Supplementary Table S2 (Section S.6.2).

In this table, the time complexity of Bezier Deformable Attention (BDA), Single Point Deformable Attention (SPDA), Multi Point Deformable Attention (MPDA), Self Attention (SA), and Masked Attention (MA) is compared in both Torch and ONNX runtimes. ONNX* indicates the ONNX runtime of the model without the auxiliary instance mask head, which will be the ideal case in deployment. Runtimes indicate the duration of the complete 10-layer decoder with the specified attention type.

In the Torch runtime, SA has the shortest computation duration, likely due to optimization for dynamic graphs. BDA and SPDA have similar durations, with BDA outperforming MPDA4 and MPDA16 by approximately 1.8 ms and 4.3ms, respectively. In the ONNX runtime, BDA has the shortest duration. While the runtime gap between BDA and MPDA narrows in the ONNX

runtime compared to the Torch runtime, BDA consistently surpasses both MPDA4 and MPDA16. This reduction in runtime is likely attributable to more efficient matrix multiplication operations enabled by the static graph optimization in ONNX. MA exhibits the highest computational complexity due to the resize operations of generated masks and foreground checks for proper attention mechanisms.

The comparable runtime between SPDA and BDA is theoretically expected when the number of heads is equal, as the substitution of reference points across attention heads does not inherently alter the computational complexity of the attention mechanism (See Figure 3). However, the requirement of SPDA for additional reference point prediction makes it slightly more computationally expensive. The benefits of BDA over MPDA arise from its removal of matrix multiplication, which can be especially observed in the Torch runtime (See Figure 4). Although BDA does not have the shortest computation duration in the Torch runtime, it is the best in the ONNX runtime, and it consistently outperforms other attention mechanisms in terms of main evaluation metrics (Table 2a).

Supplementary Table S3 in Section S.6.3 further demonstrates the number of flops, memory utilization, and number of parameters of onnx models without auxiliary heads, and it can be seen that the onnx runtimes and number of flops are consistent with each other. This supplementary section also examines the runtimes of varying numbers of encoder and decoder layers within the TopoBDA architecture (Supplementary Table S4 and S5).

4.2.6. Comparison with the State of the Art Results in OpenLane-V2

This section presents a comparative analysis of the TopoBDA architecture against other state-of-the-art methods for road topology understanding. Table 5 illustrates this comparison for Subset-A and Subset-B of the OpenLane-V2 dataset, evaluated using the OpenLane-V2 V1.1 metric baseline. To ensure fair comparison with existing literature, input images are downsampled to $0.5\times$ their original resolution prior to processing. The complete details of the image pre-processing pipeline are provided in Supplementary Section S.5.4.

TopoBDA achieves state-of-the-art results across all metrics in both subsets. It excels with both camera-only data and when combining camera and SDMap information, highlighting its robustness and effectiveness. Notably, TopoBDA is the best-performing model in both camera-only and camera + SDMap configurations, solidifying its position as the leading solution for road topology understanding.

Table 5: Comparative Evaluation of TopoBDA Architecture with Other Methods in Subset-A and Subset-B of OpenLane-V2 with V1.1 Metric Baseline. All methods utilize the ResNet50 backbone with 24 epochs. For the sensors, C denotes the camera, L denotes the lidar, and SD denotes the SDMap.

Subset	Method	Sensor	$\mathbf{DET_l}$	$\mathbf{DET_t}$	$\mathbf{TOP_{ll}}$	$\mathbf{TOP_{lt}}$	OLS
A	STSU [4]	C	12.7	43.0	2.9	19.8	29.3
	VectorMapNet [2]	C	11.1	41.7	2.7	9.2	24.9
	MapTR [3]	C	8.3	43.5	2.3	8.3	24.2
	TopoNet [6]	C	28.6	48.6	10.9	23.9	39.8
	TopoMLP [7]	C	28.5	49.5	21.7	26.9	44.1
	Topo2D [24]	C	29.1	50.6	22.3	26.2	44.4
	TopoLogic [55]	C	29.9	47.2	23.9	25.4	44.1
	RoadPainter [74]	C	30.7	47.7	22.8	27.2	44.6
	TopoFormer [56]	C	34.7	48.2	24.1	29.5	46.3
	TopoMaskV2 [18]	C	34.5	53.8	24.5	35.6	49.4
	TopoBDA (Ours)	C	38.9	**54.3**	27.6	37.3	51.7
	TopoBDA (Ours)	C + L	**47.3**	54.0	35.5	**41.9**	**56.4**
	SMERF [30]	C + SD	33.4	48.6	15.4	25.4	42.9
	TopoLogic [55]	C + SD	34.4	48.3	28.9	28.7	47.5
	RoadPainter [74]	C + SD	36.9	47.1	29.6	29.5	48.2
	TopoBDA (Ours)	C + SD	42.7	**52.4**	34.3	41.7	54.6
	TopoBDA (Ours)	C + L + SD	**52.0**	**52.4**	**38.5**	**45.3**	**58.4**
B	TopoNet [6]	C	24.3	55.0	6.7	16.7	36.0
	TopoMLP [7]	C	25.2	**63.1**	20.7	20.3	44.7
	TopoLogic [55]	C	25.9	54.7	21.6	17.9	42.3
	TopoFormer [56]	C	34.8	58.9	23.2	23.3	47.5
	TopoMaskV2 [18]	C	41.6	61.1	28.7	26.1	51.8
	TopoBDA (Ours)	C	45.1	61.4	34.0	27.6	54.3
	TopoBDA (Ours)	C + L	**57.7**	62.9	**45.0**	**35.2**	**61.7**

The integration of lidar sensors significantly enhances performance, underscoring the importance of sensor fusion in improving road topology understanding. While incorporating SDMap information also boosts TopoBDA's performance, leading to substantial improvements, the enhancements from lidar are more pronounced. This demonstrates the value of both lidar and map information, with lidar providing a greater impact on performance.

The highest performance gains are achieved by combining camera, lidar, and SDMap data. This synergy of multiple inputs results in top scores across all metrics, showcasing the comprehensive understanding of road topology that TopoBDA offers. Lidar sensors and SDMap provide complementary information, further enhancing the model's performance.

Table 6: Comparative Evaluation of TopoBDA architecture with other methods on the OpenLane-V1 Dataset. The table compares the performance comparisoın of different methods based on various metrics.

Dist.	Methods	Backbone	F1-Score ↑	X-error near (m) ↓	X-error far (m) ↓	Z-error near (m) ↓	Z-error far (m) ↓
1.5m	PersFormer [41]	ResNet-50	52.7	0.307	0.319	0.083	0.117
	Anchor3DLane [75]	ResNet-50	57.5	0.229	**0.243**	0.079	0.106
	GroupLane	ResNet-50	60.2	0.371	0.476	0.220	0.357
	LaneCPP [76]	EffNet-B7	60.3	0.264	0.310	0.077	0.117
	LATR [8]	ResNet-50	61.9	**0.219**	0.259	0.075	0.104
	PVALane [77]	ResNet-50	62.7	0.232	0.259	0.092	0.118
	TopoBDA (Ours)	ResNet-50	**63.9**	0.224	**0.243**	**0.069**	**0.101**
0.5m	PersFormer [41]	ResNet-50	43.2	0.229	0.245	0.078	0.106
	DV-3DLane (Camera) [26]	ResNet-34	52.9	0.173	0.212	0.069	0.098
	LATR [8]	ResNet-50	54.0	0.171	0.201	0.072	0.099
	TopoBDA (Ours)	ResNet-50	**57.9**	**0.157**	**0.179**	**0.067**	**0.087**

4.2.7. Comparison with the State of the Art Results in OpenLane-V1

The proposed TopoBDA architecture outperforms other methods on the OpenLane-V1 dataset, achieving state-of-the-art results across multiple metrics (Table 6). For this analysis, we utilized the ResNet-50 backbone and processed images at 0.5× of the original resolution.

TopoBDA attains the highest F1-Score for both distance categories (1.5m and 0.5m), with scores of 63.9 and 57.9, respectively. Additionally, it records the lowest X-error and Z-error values, indicating exceptional accuracy in both lateral positioning and depth estimation.

Although LATR achieves the lowest X-error (near) for the 1.5m distance, the difference is minimal, with TopoBDA trailing by only 5 millimeters. Furthermore, TopoBDA surpasses LATR in the X-error (near) metric for the 0.5m distance, outperforming by 14 millimeters. Notably, despite the hyperparameters being optimized for centerline detection in the OpenLane-V2 dataset, TopoBDA also achieves the best results for lane divider detection in the OpenLane-V1 dataset. Further optimization of score thresholds and hyperparameters for this task is expected to improve TopoBDA's performance on the OpenLane-V1 dataset.

5. Conclusion

Experimental evaluations demonstrate that **TopoBDA achieves state-of-the-art performance** across both subsets of the OpenLane-V2 dataset, using the version 1.1 metric baseline. Specifically, TopoBDA surpasses existing

methods with a **$\mathbf{DET}_l$ score of 38.9** and an **OLS score of 51.7** in Subset-A, and a **$\mathbf{DET}_l$ score of 45.1** and an **OLS score of 54.3** in Subset-B. The integration of multi-modal data significantly boosts performance: fusing camera and LiDAR data increases the OLS score in Subset-A from **51.7 to 56.4**, and in Subset-B from **54.3 to 61.7**. Further incorporating SDMap alongside camera and LiDAR sensors raises the OLS score in Subset-A to **58.4**. These results underscore the effectiveness of TopoBDA in road topology comprehension and highlight the substantial benefits of multi-modal fusion.

Additionally, TopoBDA achieves superior results on the OpenLane-V1 benchmark for 3D lane detection, with F1-scores of **63.9** at a 1.5m distance and **57.9** at a 0.5m distance.

This work contributes toward closing existing gaps in HDMap element prediction, offering a unified framework for road topology understanding and 3D lane detection in autonomous driving. By leveraging Bezier deformable attention, instance mask formulation, multi-modal fusion, and an auxiliary one-to-many set prediction loss strategy, TopoBDA delivers high accuracy in centerline detection and topological reasoning. The approach not only improves computational efficiency but also sets a new benchmark in the field, highlighting its potential for practical applications in autonomous driving systems. The contributions of the TopoBDA study are also demonstrated in Supplementary Table S9 in Section S.8.

Despite its strengths, TopoBDA has certain limitations. While it performs well in detecting structured elements such as centerlines and lane dividers, it may face challenges in drivable area prediction. These regions often exhibit complex geometries with protrusions and indentations, which are difficult to represent using a fixed number of Bezier control points. Accurately modeling such shapes may require a larger number of control points, increasing the computational burden of the attention mechanism.

Moreover, the architecture imposes constraints on feature dimensionality: the number of channels in query features must be divisible by both the number of self-attention heads and the number of Bezier control points, which correspond to the number of heads in the cross-attention (Bezier Deformable Attention). Consequently, adjusting the number of control points may necessitate changes in feature dimensions, potentially complicating design flexibility. A detailed analysis of these constraints is provided in Supplementary Section S.6.5.

Acknowledgements

We acknowledge the use of the TRUBA high-performance computing infrastructure provided by TÜBİTAK ULAKBİM for the computations in this study. We also extend our gratitude to the Barcelona Supercomputing Center (BSC-CNS) and the EuroHPC Joint Undertaking for granting access to the MareNostrum 5 supercomputer, which provided additional computational resources.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used Microsoft Copilot to improve the language of the paper and to format the tables. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

References

[1] Q. Li, Y. Wang, Y. Wang, H. Zhao, Hdmapnet: An online hd map construction and evaluation framework, in: 2022 International Conference on Robotics and Automation (ICRA), IEEE, 2022, pp. 4628–4634.

[2] Y. Liu, T. Yuan, Y. Wang, Y. Wang, H. Zhao, Vectormapnet: End-to-end vectorized hd map learning, in: International Conference on Machine Learning, PMLR, 2023, pp. 22352–22369.

[3] B. Liao, S. Chen, X. Wang, T. Cheng, Q. Zhang, W. Liu, C. Huang, MapTR: Structured modeling and learning for online vectorized HD map construction, in: The Eleventh International Conference on Learning Representations, 2023.
URL https://openreview.net/forum?id=k7p_YAO7yE

[4] Y. B. Can, A. Liniger, D. P. Paudel, L. Van Gool, Structured bird's-eye-view traffic scene understanding from onboard images, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 15661–15670.

[5] Y. Liu, J. Yan, F. Jia, S. Li, A. Gao, T. Wang, X. Zhang, Petrv2: A unified framework for 3d perception from multi-camera images, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 3262–3272.

[6] T. Li, L. Chen, H. Wang, Y. Li, J. Yang, X. Geng, S. Jiang, Y. Wang, H. Xu, C. Xu, et al., Graph-based topology reasoning for driving scenes, arXiv preprint arXiv:2304.05277 (2023).

[7] D. Wu, J. Chang, F. Jia, Y. Liu, T. Wang, J. Shen, Topomlp: An simple yet strong pipeline for driving topology reasoning, ICLR (2024).

[8] Y. Luo, C. Zheng, X. Yan, T. Kun, C. Zheng, S. Cui, Z. Li, Latr: 3d lane detection from monocular images with transformer, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 7941–7952.

[9] T. Li, P. Jia, B. Wang, L. Chen, K. Jiang, J. Yan, H. Li, Lanesegnet: Map learning with lane segment perception for autonomous driving, in: ICLR, 2024.

[10] Y. Bai, Z. Chen, Z. Fu, L. Peng, P. Liang, E. Cheng, Curveformer: 3d lane detection by curve propagation with curve queries and attention, in: 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 7062–7068.

[11] Y. Zhou, H. Zhang, J. Yu, Y. Yang, S. Jung, S.-I. Park, B. Yoo, Himap: Hybrid representation learning for end-to-end vectorized hd map construction, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 15396–15406.

[12] S. Choi, J. Kim, H. Shin, J. W. Choi, Mask2map: Vectorized hd map construction using bird's eye view segmentation masks, arXiv preprint arXiv:2407.13517 (2024).

[13] Z. Liu, X. Zhang, G. Liu, J. Zhao, N. Xu, Leveraging enhanced queries of point sets for vectorized map construction, arXiv preprint arXiv:2402.17430 (2024).

[14] Z. Xu, K.-Y. K. Wong, H. Zhao, Insmapper: Exploring inner-instance information for vectorized hd mapping (2024). arXiv:2308.08543. URL https://arxiv.org/abs/2308.08543

[15] J. Yu, Z. Zhang, S. Xia, J. Sang, Scalablemap: Scalable map learning for online long-range vectorized hd map construction, arXiv preprint arXiv:2310.13378 (2023).

[16] L. Qiao, W. Ding, X. Qiu, C. Zhang, End-to-end vectorized hd-map construction with piecewise bezier curve, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 13218–13228.

[17] W. Ding, L. Qiao, X. Qiu, C. Zhang, Pivotnet: Vectorized pivot learning for end-to-end hd map construction, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 3672–3682.

[18] M. E. Kalfaoglu, H. I. Ozturk, O. Kilinc, A. Temizel, Topomaskv2: Enhanced instance-mask-based formulation for the road topology problem, arXiv preprint arXiv:2409.11325 (2024).

[19] B. Cheng, I. Misra, A. G. Schwing, A. Kirillov, R. Girdhar, Masked-attention mask transformer for universal image segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 1290–1299.

[20] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, J. Dai, Deformable detr: Deformable transformers for end-to-end object detection, in: International Conference on Learning Representations, 2020.

[21] T. Yuan, Y. Liu, Y. Wang, Y. Wang, H. Zhao, Streammapnet: Streaming mapping network for vectorized online hd map construction, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 7356–7365.

[22] J. Chen, Y. Wu, J. Tan, H. Ma, Y. Furukawa, Maptracker: Tracking with strided memory fusion for consistent vector hd mapping, in: European Conference on Computer Vision, Springer, 2025, pp. 90–107.

[23] B. Liao, S. Chen, Y. Zhang, B. Jiang, Q. Zhang, W. Liu, C. Huang, X. Wang, Maptrv2: An end-to-end framework for online vectorized hd map construction, International Journal of Computer Vision (2024) 1–23.

[24] H. Li, Z. Huang, Z. Wang, W. Rong, N. Wang, S. Liu, Enhancing 3d lane detection and topology reasoning with 2d lane priors, arXiv preprint arXiv:2406.03105 (2024).

[25] F. Li, H. Zhang, H. Xu, S. Liu, L. Zhang, L. M. Ni, H.-Y. Shum, Mask dino: Towards a unified transformer-based framework for object detection and segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 3041–3050.

[26] Y. Luo, S. Cui, Z. Li, Dv-3dlane: End-to-end multi-modal 3d lane detection with dual-view representation, arXiv preprint arXiv:2406.16072 (2024).

[27] Y. Luo, X. Yan, C. Zheng, C. Zheng, S. Mei, T. Kun, S. Cui, Z. Li, M^2-3dlanenet: Exploring multi-modal 3d lane detection, arXiv preprint arXiv:2209.05996 (2022).

[28] X. Liu, S. Wang, W. Li, R. Yang, J. Chen, J. Zhu, Mgmap: Mask-guided learning for online vectorized hd map construction, in: Proceedings of the

IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 14812–14821.

[29] G. Zhang, J. Lin, S. Wu, Z. Luo, Y. Xue, S. Lu, Z. Wang, et al., Online map vectorization for autonomous driving: A rasterization perspective, Advances in Neural Information Processing Systems 36 (2024).

[30] K. Z. Luo, X. Weng, Y. Wang, S. Wu, J. Li, K. Q. Weinberger, Y. Wang, M. Pavone, Augmenting lane perception and topology understanding with standard definition navigation maps, arXiv preprint arXiv:2311.04079 (2023).

[31] S. Yang, M. Jiang, Z. Fan, X. Xie, X. Tan, Y. Li, E. Ding, L. Wang, J. Wang, Toposd: Topology-enhanced lane segment perception with sdmap prior, arXiv preprint arXiv:2411.14751 (2024).

[32] D. Jia, Y. Yuan, H. He, X. Wu, H. Yu, W. Lin, L. Sun, C. Zhang, H. Hu, Detrs with hybrid matching, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 19702–19712.

[33] X. Pan, J. Shi, P. Luo, X. Wang, X. Tang, Spatial as deep: Spatial cnn for traffic scene understanding, in: Proceedings of the AAAI conference on artificial intelligence, Vol. 32, 2018.

[34] Z. Qin, P. Zhang, X. Li, Ultra fast deep lane detection with hybrid anchor driven ordinal classification, IEEE transactions on pattern analysis and machine intelligence 46 (5) (2022) 2555–2568.

[35] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, T. Oliveira-Santos, Keep your eyes on the lane: Real-time attention-guided lane detection, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 294–302.

[36] H. Xu, S. Wang, X. Cai, W. Zhang, X. Liang, Z. Li, Curvelane-nas: Unifying lane-sensitive architecture search and adaptive point blending, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16, Springer, 2020, pp. 689–704.

[37] L. Liu, X. Chen, S. Zhu, P. Tan, Condlanenet: a top-to-down lane detection framework based on conditional convolution, in: Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 3773–3782.

[38] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, T. Oliveira-Santos, Polylanenet: Lane estimation via deep polynomial regression, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE, 2021, pp. 6150–6156.

[39] Z. Feng, S. Guo, X. Tan, K. Xu, M. Wang, L. Ma, Rethinking efficient lane detection via curve modeling, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 17062–17070.

[40] F. Yan, M. Nie, X. Cai, J. Han, H. Xu, Z. Yang, C. Ye, Y. Fu, M. B. Mi, L. Zhang, Once-3dlanes: Building monocular 3d lane detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 17143–17152.

[41] L. Chen, C. Sima, Y. Li, Z. Zheng, J. Xu, X. Geng, H. Li, C. He, J. Shi, Y. Qiao, J. Yan, Persformer: 3d lane detection via perspective transformer and the openlane benchmark, in: European Conference on Computer Vision (ECCV), 2022.

[42] Y. Guo, G. Chen, P. Zhao, W. Zhang, J. Miao, J. Wang, T. E. Choe, Gen-lanenet: A generalized and scalable approach for 3d lane detection, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXI 16, Springer, 2020, pp. 666–681.

[43] R. Wang, J. Qin, K. Li, Y. Li, D. Cao, J. Xu, Bev-lanedet: An efficient 3d lane detection based on virtual camera via key-points, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 1002–1011.

[44] Z. Chen, K. Smith-Miles, B. Du, G. Qian, M. Gong, An efficient transformer for simultaneous learning of bev and lane representations in 3d lane detection, arXiv preprint arXiv:2306.04927 (2023).

[45] H. İbrahim Öztürk, M. E. Kalfaoglu, O. Kilinc, Glane3d: Detecting lanes with graph of 3d keypoints, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2025.

[46] J. Shin, F. Rameau, H. Jeong, D. Kum, Instagram: Instance-level graph modeling for vectorized hd map learning, arXiv preprint arXiv:2301.04470 (2023).

[47] H. Hu, F. Wang, Y. Wang, L. Hu, J. Xu, Z. Zhang, Admap: Anti-disturbance framework for reconstructing online vectorized hd map, arXiv preprint arXiv:2401.13172 (2024).

[48] R. Wang, X. Lu, X. Liu, X. Zou, T. Cao, Y. Li, Priormapnet: Enhancing online vectorized hd map construction with priors, arXiv preprint arXiv:2408.08802 (2024).

[49] B. Wijaya, M. Yang, T. Wen, K. Jiang, Y. Wang, Z. Fu, X. Tang, D. O. Sigomo, J. Miao, D. Yang, Multi-session high-definition map-monitoring system for map update, ISPRS International Journal of Geo-Information 13 (1) (2023) 6.

[50] Z. Xu, S. Li, L. Peng, B. Jiang, R. Huang, Y. Chen, Ultra-fast semantic map perception model for autonomous driving, Neurocomputing 599 (2024) 128162.

[51] Z. Xu, Y. Liu, Y. Sun, M. Liu, L. Wang, Centerlinedet: Road lane centerline graph detection with vehicle-mounted sensors by transformer for high-definition map creation, arXiv preprint arXiv:2209.07734 (2022).

[52] B. Liao, S. Chen, B. Jiang, T. Cheng, Q. Zhang, W. Liu, C. Huang, X. Wang, Lane graph as path: Continuity-preserving path-wise modeling for online lane graph construction, arXiv preprint arXiv:2303.08815 (2023).

[53] J. Ye, D. Paz, H. Zhang, Y. Guo, X. Huang, H. I. Christensen, Y. Wang, L. Ren, Smart: Advancing scalable map priors for driving topology reasoning, arXiv preprint arXiv:2502.04329 (2025).

[54] Y. Han, K. Yu, Z. Li, Continuity preserving online centerline graph learning, arXiv preprint arXiv:2407.11337 (2024).

[55] Y. Fu, W. Liao, X. Liu, Y. Ma, F. Dai, Y. Zhang, et al., Topologic: An interpretable pipeline for lane topology reasoning on driving scenes, arXiv preprint arXiv:2405.14747 (2024).

[56] C. Lv, M. Qi, L. Liu, H. Ma, T2sg: Traffic topology scene graph for topology reasoning in autonomous driving, arXiv preprint arXiv:2411.18894 (2024).

[57] J. Philion, S. Fidler, Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16, Springer, 2020, pp. 194–210.

[58] J. Huang, G. Huang, Z. Zhu, Y. Yun, D. Du, Bevdet: High-performance multi-camera 3d object detection in bird-eye-view, arXiv preprint arXiv:2112.11790 (2021).

[59] Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, Z. Li, Bevdepth: Acquisition of reliable depth for multi-view 3d object detection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 37, 2023, pp. 1477–1485.

[60] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, J. Dai, Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers, in: European conference on computer vision, Springer, 2022, pp. 1–18.

[61] S. Chen, T. Cheng, X. Wang, W. Meng, Q. Zhang, W. Liu, Efficient and robust 2d-to-bev representation learning via geometry-guided kernel transformer, arXiv preprint arXiv:2206.04584 (2022).

[62] B. Zhou, P. Krähenbühl, Cross-view transformers for real-time map-view semantic segmentation, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 13760–13769.

[63] S. Wang, Y. Liu, T. Wang, Y. Li, X. Zhang, Exploring object-centric temporal modeling for efficient multi-view 3d object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 3621–3631.

[64] Y. Li, B. Huang, Z. Chen, Y. Cui, F. Liang, M. Shen, F. Liu, E. Xie, L. Sheng, W. Ouyang, et al., Fast-bev: A fast and strong bird's-eye view perception baseline, IEEE Transactions on Pattern Analysis and Machine Intelligence (2024).

[65] E. Xie, Z. Yu, D. Zhou, J. Philion, A. Anandkumar, S. Fidler, P. Luo, J. M. Alvarez, M^2bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation, arXiv preprint arXiv:2204.05088 (2022).

[66] A. W. Harley, Z. Fang, J. Li, R. Ambrus, K. Fragkiadaki, Simple-bev: What really matters for multi-sensor bev perception?, in: 2023 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2023, pp. 2759–2765.

[67] T. Liang, H. Xie, K. Yu, Z. Xia, Z. Lin, Y. Wang, T. Tang, B. Wang, Z. Tang, Bevfusion: A simple and robust lidar-camera fusion framework, Advances in Neural Information Processing Systems 35 (2022) 10421–10434.

[68] Z. Liu, H. Tang, A. Amini, X. Yang, H. Mao, D. L. Rus, S. Han, Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation, in: 2023 IEEE international conference on robotics and automation (ICRA), IEEE, 2023, pp. 2774–2781.

[69] Y. Tang, H. He, Y. Wang, Z. Mao, H. Wang, Multi-modality 3d object detection in autonomous driving: A review, Neurocomputing 553 (2023) 126587.

[70] H. Wang, T. Li, Y. Li, L. Chen, C. Sima, Z. Liu, B. Wang, P. Jia, Y. Wang, S. Jiang, et al., Openlane-v2: A topology reasoning benchmark for unified 3d hd mapping, Advances in Neural Information Processing Systems 36 (2024).

[71] B. Wilson, W. Qi, T. Agarwal, J. Lambert, J. Singh, S. Khandelwal, B. Pan, R. Kumar, A. Hartnett, J. K. Pontes, et al., Argoverse 2: Next generation datasets for self-driving perception and forecasting, arXiv preprint arXiv:2301.00493 (2023).

[72] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, O. Beijbom, nuscenes: A multimodal dataset

for autonomous driving, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 11621–11631.

[73] Y. Yan, Y. Mao, B. Li, Second: Sparsely embedded convolutional detection, Sensors 18 (10) (2018) 3337.

[74] Z. Ma, S. Liang, Y. Wen, W. Lu, G. Wan, Roadpainter: Points are ideal navigators for topology transformer, arXiv preprint arXiv:2407.15349 (2024).

[75] S. Huang, Z. Shen, Z. Huang, Z.-h. Ding, J. Dai, J. Han, N. Wang, S. Liu, Anchor3dlane: Learning to regress 3d anchors for monocular 3d lane detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 17451–17460.

[76] M. Pittner, J. Janai, A. P. Condurache, Lanecpp: Continuous 3d lane detection using physical priors, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024, pp. 10639–10648.

[77] Z. Zheng, X. Zhang, Y. Mou, X. Gao, C. Li, G. Huang, C.-M. Pun, X. Yuan, Pvalane: Prior-guided 3d lane detection with view-agnostic feature alignment, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 38, 2024, pp. 7597–7604.

[78] A. Kirillov, Y. Wu, K. He, R. Girshick, Pointrend: Image segmentation as rendering, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 9799–9808.

[79] S. Liu, F. Li, H. Zhang, X. Yang, X. Qi, H. Su, J. Zhu, L. Zhang, DAB-DETR: Dynamic anchor boxes are better queries for DETR, in: International Conference on Learning Representations, 2022.
URL `https://openreview.net/forum?id=oMI9PjOb9Jl`

[80] F. Li, H. Zhang, S. Liu, J. Guo, L. M. Ni, L. Zhang, Dn-detr: Accelerate detr training by introducing query denoising, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2022, pp. 13619–13627.

[81] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, H.-Y. Shum, Dino: Detr with improved denoising anchor boxes for end-to-end object detection (2022). arXiv:2203.03605.

[82] T. maintainers, contributors, Torchvision: Pytorch's computer vision library, https://github.com/pytorch/vision, gitHub repository (2016).

[83] R. Wightman, Pytorch image models, https://github.com/rwightman/pytorch-image-models (2019). doi:10.5281/zenodo.4414861.

[84] J. Huang, G. Huang, Bevpoolv2: A cutting-edge implementation of bevdet toward deployment (2022). arXiv:2211.17111.

[85] ThanatosShinji, onnx-tool: A parser, editor and profiler tool for onnx models, https://github.com/ThanatosShinji/onnx-tool, accessed: 2025-08-24 (2025).

S. Supplementary Materials

In this supplementary materials section, we delve into the intricate details underpinning the TopoBDA study. Each section complements the main manuscript by providing deeper insights into the methodology, implementation, and evaluation.

- **Section S.1:** Presents the matrix formulation for extracting polyline points from Bezier control points using Bernstein basis functions.
- **Section S.2:** Details the mathematical background of sensor fusion and SDMap integration, including voxelization, multi-modal concatenation, and BEV-level fusion strategies.
- **Section S.3:** Explores the auxiliary one-to-many set prediction loss strategy, including its mathematical formulation and decoder sharing mechanism.
- **Section S.4:** Describes the loss functions used in TopoBDA, including regression, mask, dice, and classification losses, as well as the overall training objective.
- **Section S.5:** Provides comprehensive implementation details, covering architectural configurations, optimization parameters, backbone choices, and dataset preprocessing.
- **Section S.6:** Presents extended experimental analyses, including ablations on view transformation methods, multi-scale implementations, attention mechanisms, encoder/decoder depth, control point variation, backbone types, and training epochs.
- **Section S.7:** Introduces a step-wise algorithmic breakdown of the TopoBDA decoder, highlighting iterative refinement of Bezier control points and mask predictions across layers.
- **Section S.8:** Offers a comparative novelty analysis across road topology and HDMap element prediction methods, positioning TopoBDA in the broader research landscape.
- **Section S.9:** Showcases visual results in both perspective and BEV domains, illustrating the model's performance in centerline detection and topological reasoning.

Together, these supplementary sections provide a thorough and transparent account of the TopoBDA framework, supporting reproducibility and facilitating deeper understanding of the proposed contributions.

S.1. Matrix Formulation for Extracting Polyline Points Using Bernstein Basis

The mathematical formulation for converting Bezier control points to polyline points is provided in Eq. (5). This formulation underpins the extraction of $L+1$ polyline points $\mathbf{P} = \{\mathbf{p}_0, \mathbf{p}_1, \ldots, \mathbf{p}_L\}$ from the $N+1$ control points $\mathbf{C} = \{\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_N\}$. In practice, Eq. (5) is realized through matrix multiplication as shown in Eq. S1.

$$\mathbf{P} = \mathbf{B}\mathbf{C}, \tag{S1}$$

where the Bernstein basis matrix $\mathbf{B}$ is defined as:

$$\mathbf{B} = \begin{bmatrix} B_{0,0} & B_{0,1} & \cdots & B_{0,N} \\ B_{1,0} & B_{1,1} & \cdots & B_{1,N} \\ \vdots & \vdots & \ddots & \vdots \\ B_{L,0} & B_{L,1} & \cdots & B_{L,N} \end{bmatrix}. \tag{S2}$$

In Eq. (S2), each element $B_{l,n}$ represents the discretized version of the continuous Bernstein polynomials $B_{n,N}(t)$, shown in Eq. (2), and the mathematical formulation of $B_{l,n}$ is given in Eq. S3.

$$B_{l,n} = \binom{N}{n} t_l^n (1-t_l)^{N-n}, \tag{S3}$$

for $l = 0, 1, \ldots, L$ and $n = 0, 1, \ldots, N$, where t_l are uniformly spaced within the interval $[0, 1]$.

S.2. Mathematical Background of Sensor Fusion and SDMap

The fusion methodology is detailed in the following equation flow:

1. **Image Acquisition:** A set of N images is captured, denoted as $\{\mathbf{I}_i\}_{i=1}^N$, where each image $\mathbf{I}_i$ has dimensions $H_I \times W_I \times 3$.
2. **Feature Extraction:** Each image $\mathbf{I}_i$ is converted into perspective view features using a feature extraction function $\mathbf{f}_{PV}$:

$$\{\mathbf{F}_{PV_i}\}_{i=1}^N = \mathbf{f}_{PV}(\{\mathbf{I}_i\}_{i=1}^N), \tag{S4}$$

where $\mathbf{F}_{PV_i} \in \mathbb{R}^{H_{PV} \times W_{PV} \times C_{PV}}$.

3. **Voxelization:** The N perspective view features are converted into a single voxel space using a voxelization function $\mathbf{f}_{voxel}$, which can be any voxel creation algorithm. In this case, a multi-height bin implementation of the Lift Splat algorithm [18] is used:

$$\mathbf{F}_{\text{voxelCam}} = \mathbf{f}_{voxel}(\{\mathbf{F}_{PV_i}\}_{i=1}^{N}), \tag{S5}$$

where $\mathbf{F}_{\text{voxelCam}} \in \mathbb{R}^{H_{\text{bev}} \times W_{\text{bev}} \times Z \times C_{\text{camera}}}$.

4. **Radar and Lidar Point Clouds:** The radar and lidar point clouds are integrated into the voxel space. Each point is characterized by its (x, y, z) coordinates and associated feature vectors $\mathbf{F}_{\text{radar}}$ and $\mathbf{F}_{\text{lidar}}$, with dimensions C_{radar} and C_{lidar}, respectively:

$$\begin{aligned} \text{Radar Point Cloud: } & \{\mathbf{p}_{\text{radar}}^{i}\}_{i=1}^{N_{\text{radar}}}, \\ & \mathbf{p}_{\text{radar}}^{i} = (x, y, z, \mathbf{F}_{\text{radar}}), \\ & \mathbf{F}_{\text{radar}} \in \mathbb{R}^{C_{\text{radar}}}, \end{aligned} \tag{S6}$$

$$\begin{aligned} \text{Lidar Point Cloud: } & \{\mathbf{p}_{\text{lidar}}^{i}\}_{i=1}^{N_{\text{lidar}}}, \\ & \mathbf{p}_{\text{lidar}}^{i} = (x, y, z, \mathbf{F}_{\text{lidar}}), \\ & \mathbf{F}_{\text{lidar}} \in \mathbb{R}^{C_{\text{lidar}}}. \end{aligned} \tag{S7}$$

Each point is assigned to the nearest voxel:

$$\begin{aligned} \mathbf{F}_{\text{voxelRadar}} &\in \mathbb{R}^{H_{\text{bev}} \times W_{\text{bev}} \times Z \times C_{\text{radar}}}, \\ \mathbf{F}_{\text{voxelLidar}} &\in \mathbb{R}^{H_{\text{bev}} \times W_{\text{bev}} \times Z \times C_{\text{lidar}}}. \end{aligned} \tag{S8}$$

As an alternative to directly utilizing raw lidar features in the fusion process, another option is provided to enhance the lidar data integration. High-resolution voxels are initially created and subsequently processed using a lidar encoder $\mathbf{f}_{\text{lidar}}$, such as SECOND [73], which reduces them to the desired voxel dimensions suitable for concatenation:

$$\mathbf{F}_{\text{voxelLidar}} = \mathbf{f}_{\text{lidar}}(\{\mathbf{p}_{\text{lidar}}^{i}\}_{i=1}^{N_{\text{lidar}}}). \tag{S9}$$

5. **Obtaining BEV Features:** The voxel features from the camera, radar, and lidar are concatenated:

$$\mathbf{F}_{\text{sensors}} = \text{concat}(\mathbf{F}_{\text{voxelCam}}, \mathbf{F}_{\text{voxelRadar}}, \mathbf{F}_{\text{voxelLidar}}), \tag{S10}$$

and the concatenated dimension is:

$$\mathbf{F}_{\text{sensors}} \in \mathbb{R}^{H_{\text{bev}} \times W_{\text{bev}} \times Z \times (C_{\text{camera}} + C_{\text{radar}} + C_{\text{lidar}})}. \tag{S11}$$

The channel and z dimensions are combined and a 2D convolution is applied to convert the concatenated features to BEV features:

$$\mathbf{F}_{\text{bev}} = \mathbf{f}_{\text{conv2}}(\mathbf{F}_{\text{sensors}}). \tag{S12}$$

6. **Fusion of SDMap:** If SDMap is enabled, it is created by rasterizing the map information, denoted as $\mathbf{M}$, into a tensor of dimensions $H_{\text{bev}} \times W_{\text{bev}} \times 3$. The map information $\mathbf{M}$ includes crosswalks, pedestrian crossings, and drivable area points, with each type of information occupying one channel of the tensor. This tensor is further processed through three convolutional layers, batch normalization, and ReLU operations, denoted as $\mathbf{f}_{\text{SDMap}}$. The processed SDMap tensor is then concatenated with the resulting BEV features from the sensors:

$$\mathbf{F}_{\text{SDMap}} = \mathbf{f}_{\text{SDMap}}(\mathbf{r}_{\text{map}}(\mathbf{M})), \tag{S13}$$

where $\mathbf{r}_{\text{map}}(\mathbf{M})$ represents the rasterization of the map information $\mathbf{M}$. The concatenated dimension is:

$$\mathbf{F}_{\text{sensors+SDMap}} = \text{concat}(\mathbf{F}_{\text{bev}}, \mathbf{F}_{\text{SDMap}}). \tag{S14}$$

Finally, a 2D convolution, denoted as $\mathbf{f}_{\text{conv2}}$, is applied to the concatenated features to yield the desired BEV features:

$$\mathbf{F}_{\text{finalBEV}} = \mathbf{f}_{\text{conv2}}(\mathbf{F}_{\text{sensor+SDMap}}). \tag{S15}$$

Both $\mathbf{F}_{\text{bev}}$ and $\mathbf{F}_{\text{finalBEV}}$ are in $\mathbb{R}^{H_{\text{bev}} \times W_{\text{bev}} \times C_{\text{BEV}}}$.

S.3. Mathematical Background of Auxiliary One-to-Many Set Prediction Loss

The following equations detail the mathematical formulation of this approach:

1. **Ground Truth Set:** Define the ground truth set $\mathbf{G}$:

$$\mathbf{G} = \{g_0, g_1, \ldots, g_{n-1}\}. \tag{S16}$$

2. **Repeated Ground Truth Set:** Define the repeated ground truth set $\mathbf{S}$, where R denotes the number of repetitions for ground truth centerlines:

$$\mathbf{S} = \{s_0, s_1, \ldots, s_{nR-1}\} \quad \text{where} \quad s_i = g_{(i \mod n)}. \tag{S17}$$

3. **Query Sets:** Define the query sets $\mathbf{Q}$ and $\mathbf{RQ}$:

$$\begin{aligned} \mathbf{Q} &= \{q_0, q_1, \ldots, q_{m-1}\}, \\ \mathbf{RQ} &= \{rq_0, rq_1, \ldots, rq_{mR-1}\}. \end{aligned} \tag{S18}$$

   During inference, only the query set $\mathbf{Q}$ is utilized. The query set $\mathbf{RQ}$ is used only during training to improve the model's performance.

4. **Set Prediction Losses:** Define the one-to-one set prediction loss between $\mathbf{G}$ and $\mathbf{Q}$ and the one-to-many set prediction loss between $\mathbf{RQ}$ and $\mathbf{S}$:

$$\begin{aligned} \mathcal{L}_{\text{one-to-one}} &= \text{SetPredictionLoss}(\mathbf{G}, \mathbf{Q}), \\ \mathcal{L}_{\text{one-to-many}} &= \text{SetPredictionLoss}(\mathbf{RQ}, \mathbf{S}). \end{aligned} \tag{S19}$$

5. **Concatenation of Q and RQ:** In the realization of this two-decoder concept, the query sets $\mathbf{Q}$ and $\mathbf{RQ}$ are concatenated and a single decoder is utilized.

$$\mathbf{Q}_{\text{concat}} = \mathbf{Q} \cup \mathbf{RQ}. \tag{S20}$$

6. **Self-Attention with Masking:** Within this single decoder, self-attention is applied to the concatenated query set using masking to ensure attention weights are zero between $\mathbf{Q}$ and $\mathbf{RQ}$. In this way, a single decoder behaves like two different decoders.

   - Define the attention logits matrix $\mathbf{A}$ for the concatenated query set $\mathbf{Q}_{\text{concat}}$.
   - Create a mask matrix $\mathbf{M}$ of the same size as $\mathbf{A}$, where:

$$M_{ij} = \begin{cases} -\infty & \text{if } q_i \in \mathbf{Q} \text{ and } q_j \in \mathbf{RQ}, \\ -\infty & \text{if } q_i \in \mathbf{RQ} \text{ and } q_j \in \mathbf{Q}, \\ 0 & \text{otherwise.} \end{cases} \tag{S21}$$

   - Apply the mask to the attention logits before the softmax operation:

$$\mathbf{A}_{\text{masked}} = \mathbf{A} + \mathbf{M}. \tag{S22}$$

- Compute the attention weights using the softmax function:

$$\mathbf{A}_{\text{weights}} = \text{softmax}(\mathbf{A}_{\text{masked}}). \tag{S23}$$

7. **Training Loss:** Define the total training loss as the sum of the one-to-one and one-to-many set prediction losses, weighted by a factor λ:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{one-to-one}} + \lambda \mathcal{L}_{\text{one-to-many}}. \tag{S24}$$

S.4. Loss Function

In the transformer-based architecture, TopoBDA, a comprehensive loss function is defined to optimize the prediction of centerlines. The loss function is composed of three main components: L1 regression loss for control points, mask, and dice loss for instance mask prediction, and softmax-based classification loss for centerline existence.

S.4.1. L1 Regression Loss for Control Points

The normalized control points $\mathbf{C}_{\text{norm}} = \{\mathbf{c}_0, \mathbf{c}_1, \ldots, \mathbf{c}_N\}$ are predicted using an L1 regression loss, which measures the absolute differences between the predicted and ground truth control points for precise localization.

$$\mathcal{L}_{\text{reg}} = \frac{1}{L} \sum_{j=1}^{L} \sum_{i=0}^{N} \|\mathbf{c}_{i,j} - \hat{\mathbf{c}}_{i,j}\|_1, \tag{S25}$$

where L is the number of ground truth centerlines in a batch.

S.4.2. Mask and Dice Loss for Instance Mask Prediction

To compute the total loss for each centerline instance, the predicted mask probability map $\mathbf{M}_{\text{prob}}$ is utilized. For each instance, K points are sampled according to $\mathbf{M}_{\text{prob}}$ [78], forming the set $\mathbf{A}$:

$$\mathbf{A} = \{\mathbf{a}_k\}_{k=1}^{K} \quad \text{where} \quad \mathbf{a}_k \sim \mathbf{M}_{\text{prob}},$$

where $\mathbf{a}_k$ are not integer values, so $\mathbf{M}_{\text{prob}}(\mathbf{a}_k)$ is sampled bilinearly. The total loss for L ground truth instances in each batch is:

$$\begin{aligned} \mathcal{L}_{\text{mask}} = \frac{1}{L} \sum_{i=1}^{L} \Bigg(& \frac{1}{K} \sum_{k=1}^{K} \text{BCE}(\mathbf{M}_{\text{prob}}(\mathbf{a}_k), \mathbf{G}_{\text{map}}(\mathbf{a}_k)) \\ & + \frac{2 \sum_{k=1}^{K} \mathbf{M}_{\text{prob}}(\mathbf{a}_k) \mathbf{G}_{\text{map}}(\mathbf{a}_k)}{\sum_{k=1}^{K} \mathbf{M}_{\text{prob}}(\mathbf{a}_k) + \sum_{k=1}^{K} \mathbf{G}_{\text{map}}(\mathbf{a}_k)} \Bigg) \end{aligned} \tag{S26}$$

where $\mathbf{G}_{\text{map}}(\mathbf{a}_k)$ is the ground truth value at the sampled point $\mathbf{a}_k$. In Eq. (S26), BCE refers to the Binary Cross Entropy loss, and the second part represents the dice loss as in [19].

S.4.3. Softmax-Based Classification Loss for Centerline Detection

A softmax-based classification loss is used to predict the presence of a centerline for each query.

$$\mathcal{L}_{\text{cls}} = -\frac{1}{Q}\sum_{i=1}^{Q}\sum_{j=0}^{1}\alpha_i y_{ij}\log(p_{ij}), \tag{S27}$$

where Q is the number of queries, and α_i is the loss coefficient, set to 0.1 for queries that match with ground truths and 1 for the others.

S.4.4. Centerline Loss

The centerline loss function is a weighted sum of the individual losses, balancing the contributions of each component to optimize the overall performance of the model.

$$\mathcal{L}_{\text{l}} = \lambda_{\text{reg}}\mathcal{L}_{\text{reg}} + \lambda_{\text{mask}}\mathcal{L}_{\text{mask}} + \lambda_{\text{cls}}\mathcal{L}_{\text{cls}}, \tag{S28}$$

where $\lambda_{\text{reg}}, \lambda_{\text{mask}}, \lambda_{\text{cls}}$ are the weights for the respective loss components, determined through cross-validation.

S.4.5. Total Loss

The total loss is the sum of the centerline loss, traffic element loss (as in DAB-DETR [79]), and topology losses, including the topology loss among centerlines and the topology loss between centerlines and traffic elements (both as in TopoNet [6]).

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{l}} + \mathcal{L}_{\text{t}} + \mathcal{L}_{\text{ll}} + \mathcal{L}_{\text{lt}}. \tag{S29}$$

S.5. Implementation Details

S.5.1. TopoBDA Architecture Overview

TopoBDA, built on the TopoMaskV2 [18], features distinct backbones for traffic elements and centerline branches, ensuring no weight sharing. This design allows the traffic element branch to leverage various augmentation strategies. Specifically, a multi-scale augmentation technique is used for training [20]. The traffic element branch employs DAB-DETR [79], a deformable

attention-based detection transformer, to extract 2D traffic element queries. Additionally, the traffic element branch incorporates the denoising training strategy from DN-DETR [80] and a two-stage structure inspired by DINO [81]. ResNet50 is utilized as the backbone of the traffic element branch. For the centerline branch, ResNet50 and SwinB are utilized depending on the experimentation. Additionally, Supplementary Table S7 in Section S.6.6 shows the impact of various backbone architectures on the centerline branch, which are ResNet, ConvNext, and Swin families. In our experiments, ResNets are utilized from the TorchVision library [82], and Swins and ConvNexts are utilized from the Timm library [83].

For the BEV feature extraction, we utilize the multi-height bin implementation of the Lift-Splat algorithm [18, 57] and follow the efficient CUDA implementation of the Voxel Pooling algorithm [84]. For the topology heads, query embeddings of both traffic elements and centerlines ($\mathbf{E}_{query}$) are projected to different spaces with MLP, and the projected features are concatenated. The resulting concatenated features are again utilized in an MLP for the final topology results. For the implementation of MPDA and BDA, we have benefited from the code baseline of the LaneSegNet study [9].

S.5.2. Optimization Parameters

In the experiments, a batch size of 8 and a learning rate of 3×10^{-4} were maintained, with a 0.1 scaling factor applied to both PV and BEV backbones. The AdamW optimizer, incorporating a weight decay of 1×10^{-2}, was employed. A polynomial learning rate decay strategy with a decay factor of 0.9 and a warm-up phase spanning 1000 iterations was adopted. Gradient norm clipping was set to 35. The loss coefficients for centerline predictions were set as follows: $\lambda_{\text{reg}} = 3$, $\lambda_{\text{mask}_{BCE}} = 5$, $\lambda_{\text{mask}_{Dice}} = 5$, and $\lambda_{\text{cls}} = 2$, as detailed in Supplementary Section S.4. The bipartite matcher also utilizes the same parameters except that $\lambda_{\text{reg}} = 5$. Additionally, the auxiliary one-to-many set prediction loss coefficient was set to 1, as described in Section 3.5.

S.5.3. TopoBDA Architecture Hyperparameters

The dimensions H_{bev} and W_{bev} are set to 200 and 104, respectively, defining the size of the BEV features ($\mathbf{F}_{\text{bev}}$). Each grid in $\mathbf{F}_{\text{bev}}$ corresponds to an area of 0.5×0.5 square meters. The height dimension Z is set to 20, spanning from $[-10, 10]$ with 1-meter intervals (See Section 3.4 for more details). During the conversion of the point set into the mask structure for ground truth mask

generation, the centerline instance width is set to 4. In the PV domain, multi-scale feature selection is typically chosen as scales of $\frac{1}{8}$, $\frac{1}{16}$, and $\frac{1}{32}$. However, for BEV features, we set the scales to 1, $\frac{1}{2}$, and $\frac{1}{4}$ due to their inherently lower dimensions compared to PV images. The number of control points is set to 4 (See Section S.6.5), and the number of queries is set to 200. In the transformer encoder, the transformer decoder, and the topology heads, the number of hidden channels is 256. The number of layers in the transformer encoder and the transformer decoder is set to 6 and 10, respectively. The number of attention offsets is set to 32 for each control point of the TopoBDA decoder.

S.5.4. Dataset Preprocessing

We follow standard preprocessing practices aligned with the literature, applying a uniform 0.5× scaling to all camera inputs. In Subset-A, the front camera images have an original resolution of 2048×1550 (height × width), while other cameras are 1550×2048. After scaling, all images are resized to 1024×736 (width × height), with top crops of 178 pixels for the front view and 19 pixels for others. In Subset-B, all six camera views share the same original resolution of 900×1600, and are uniformly resized to 800×448 and cropped by 2 pixels from the top.

Traffic element detection is defined only for the front camera. In Subset-A, training randomly selects a shorter side from $\{480, 512, \ldots, 800\}$, and evaluation uses 800 pixels. In Subset-B, training uses shorter sides from $\{352, 384, \ldots, 544\}$, and evaluation uses 448 pixels. These transformations preserve the 0.5× scale and follow common practices.

We utilize the standard OpenLane-V2 dataloader, including SDMap features directly from the dataset. For lidar integration, frame IDs in OpenLane-V2 are saved as front camera timestamps in both subsets. In Subset-A, these timestamps are matched with Argoverse 2 lidar point clouds using the official `av2` API. In Subset-B, matching is performed using the front camera timestamp retrieved from nuScenes sample metadata.

S.6. Experiments

S.6.1. Comparative Analysis of View Transformation Methods

For this analysis, two primary methods were selected: the Inverse Perspective Mapping (IPM) method [65, 64, 66] and the Lift-Splat method [58, 57, 59], which employs depth estimation to scatter pixels. Subsequently, TopoMaskV2 [18] adapted the Lift-Splat method into a multi-height bin implementation.

Table S1: Performance Comparison of View Transformation Methods from PV to BEV.

Type	**DET_l**	**DET_{l_ch}**	**TOP_{ll}**	**OLS_l**
IPM Single Bin	36.1	39.2	28.9	43.0
LSS Single Bin	40.3	<u>45.1</u>	<u>32.8</u>	47.6
IPM Multi-Height Bin	**41.4**	45.0	<u>32.8</u>	<u>47.9</u>
LSS Multi-Height Bin	<u>41.0</u>	**45.9**	**33.1**	**48.1**

Table S2: Analysis of Efficient and Standard Multi-Scale Implementations for BDA. The table compares different attention mechanisms based on the OLS_l score and processing times in Torch and ONNX runtimes. ONNX* indicates inference without auxiliary mask heads.

Attention	**OLS_l ↑**	**Torch (ms) ↓**	**ONNX (ms) ↓**	**ONNX* (ms) ↓**
SA Efficient MS	41.0	**16.73**	12.89	9.50
BDA Efficient MS	46.9	<u>18.47</u>	**8.07**	**4.66**
BDA Standard MS	**48.0**	21.71	<u>12.26</u>	<u>8.77</u>

The results presented in Table S1 offer a comparative analysis of various view transformation methods. The multi-height bin implementations utilize 20 bins, with lower and upper bounds set to -10 and 10, respectively, as described in [18]. For the single bin implementations, the lower and upper bounds are set to -5 and 3 meters, following the literature. The IPM Multi-Height Bin method achieves the highest DET_l score, while the LSS Multi-Height Bin method excels in DET_{l_ch}, TOP_{ll}, and OLS_l metrics. Notably, the IPM Single Bin method exhibits a significant drop in performance compared to its multi-height bin counterpart, whereas the LSS Single Bin method does not experience a drastic decline.

S.6.2. The comparison of Standard and Efficient Multi-Scale Implementations

Efficient multi-scale implementation processes different feature scales successively in each decoder layer in a round-robin fashion, optimizing computational efficiency. In contrast, standard multi-scale implementation incorporates all feature scales in each decoder layer, providing a comprehensive but potentially less efficient approach. In Table 2a, MA and SA are implemented as efficient multi-scale implementations, while deformable attention structures (SPDA, MPDA4, MPDA16, and BDA) use standard multi-scale implementations that incorporate all feature scales in each decoder layer [20, 79]. Efficient multi-scale implementation can also be adapted for deformable structures.

Table S3: Comparison of Attention Mechanisms in Terms of Computational and Memory Metrics

Configuration	MACs (GFLOPs)	Memory (GB)	Parameters (M)
BDA	**9.9997**	**1.4763**	**14.7806**
SPDA	10.0032	1.4786	14.7816
MPDA4	9.9998	1.4765	**14.7806**
MPDA16	10.3153	1.5226	14.9283
SA	28.3765	1.9950	16.0420
MA	41.5945	5.0734	16.2394

Table S4: Decoder layer analysis: OLS_l scores, runtimes, and computational metrics for different decoder depths. NDL indicates the number of decoder layers.

NDL	OLS_l	Torch (ms)	ONNX (ms)	MACs (GFLOPs)	Memory (GB)	Params (M)
1	40.6	3.93	1.51	2.76	0.78	1.88
4	47.1	9.76	3.93	9.91	1.64	6.21
7	47.6	15.47	6.36	17.06	2.49	10.55
10	48.1	21.71	8.77	37.81	3.93	15.07

Table S2 shows that opting for efficient multi-scale implementation for BDA improves computation duration by approximately 3.2ms to 4.2ms, respectively for Torch and Onnx runtimes, at the cost of 1.1 OLS_l. Additionally, BDA significantly outperforms SA also in efficient multi-scale implementation.

S.6.3. Further Performance and Efficiency Analysis of TopoBDA Architectures

In this section, we analyze the computational and memory efficiency of various attention mechanisms in terms of FLOPs, memory usage, and parameter count. This analysis is implemented via the onnx-tool package [85]. For doing this, a complete 10-layer decoder architecture has been analyzed with the specified attention type. As shown in Table S3, the BDA configuration demonstrates the lowest computational cost across all metrics, making it the most efficient option. While **MPDA4** exhibits nearly identical values to BDA, BDA still outperforms it in terms of OLS_l score by 0.6 points, as reported in Table 2a. Furthermore, according to Table 2a, **BDA** also outperforms SPDA, MPDA16, SA, and MA in terms of both OLS_l score and other road topology metrics, thereby reinforcing its overall effectiveness compared to more computationally intensive alternatives.

In this subsection, we examine the impact of varying the number of encoder and decoder layers on key performance metrics and their runtime metrics. In both analyses, standard multi-scale implementation has been utilized.

Table S5: Encoder layer analysis: OLS_l scores and Torch runtimes for different encoder depths. NEL indicates the number of encoder layers.

NEL	**0**	**1**	**3**	**6**
OLS_l	44.7	45.3	46.3	48.1
Torch (ms)	0.00	8.64	15.78	29.58

Table S4 demonstrates the analysis of the number of layers on the decoder benchmark. As the table shows, increasing the number of decoder layers (NDL) from 1 to 4 significantly improves performance, and further increases result in marginal gains. From a computational and runtime perspective, the decoder layer of 4 might be the optimum for some deployment systems, as reducing the number of decoder layers from 10 to 4 could decrease both runtimes and memory utilization by approximately 55%, with only a minimal reduction of 1 OLS_l score. The number layer analyses on the encoder performance are shown in Table S5. An encoder layer of zero indicates that there is no encoder; therefore, the runtime is 0 ms. According to this table, increasing the number of encoder layers (NEL) consistently enhances performance across all metrics. This indicates that encoder layers generally have a more consistent impact on overall performance compared to decoder layers.

S.6.4. Comparative Analysis with TopoMaskV2 Study

The TopoMaskV2 study [18] suggests that Masked Attention (MA) outperforms Single-Point Deformable Attention (SPDA) based on a limited analysis of different attention mechanisms. In contrast, our experimental results (see Table 2a) indicate the opposite. We speculate that there are two reasons for this discrepancy. First, our study observes that deformable attention performs better with the inclusion of the L1 Matcher (Mask-L1 Mix Matcher), whereas TopoMaskV2 relies solely on a mask matcher for its bipartite matching strategy. Second, TopoMaskV2 combines both the Bezier Head and Mask Head, which may result in the mask head benefiting more from MA.

Another aspect to consider is the extent of improvement brought by the multi-height bin implementation. In the TopoMaskV2 study, the improvements in DET_l and DET_{l_ch} are 3.2 and 3.1 points, respectively. In comparison, the improvements of TopoBDA are relatively modest, at 0.7 and 0.8 points, respectively, as shown in Figure S1. This might indicate that TopoBDA's superior performance leaves less room for improvement, whereas TopoMaskV2,

Table S6: Performance impact of varying the number of control points on road topology understanding in Subset-A of OpenLane-V2, evaluated using the V1.1m baseline.

Control Points	$\mathbf{DET_l}$	$\mathbf{DET_{l\_ch}}$	$\mathbf{TOP_{ll}}$	$\mathbf{OLS_l}$
3	40.3	42.4	31.5	46.3
4	41.0	<u>45.9</u>	<u>33.1</u>	48.1
5	<u>41.1</u>	**46.1**	33.0	<u>48.2</u>
6	40.8	44.8	32.9	47.7
8	**41.2**	**46.1**	**33.4**	**48.4**

with its lower baseline, benefits more significantly from the multi-height bin implementation.

S.6.5. Impact of Control Point Variation

Table S6 summarizes how varying the number of control points affects road topology understanding. This variation influences key components of the TopoBDA architecture, including attention head configuration, regression target dimensionality, and matrix multiplication used to convert Bézier control points into lane coordinates, all contributing to computational complexity.

Increasing control points from 3 to 4 yields a notable improvement in OLS_l (+1.8), while the gain from 4 to 5 is marginal (+0.1). Performance slightly drops at 6 points, suggesting diminishing returns. The configuration with 8 control points achieves the highest $\mathrm{OLS_l}$ score (48.4), indicating optimal expressiveness.

Practically, 4 control points align with the default setting of 256 query channels and 8 attention heads. In contrast, 5 control points require the number of query channels to be divisible by 5, resulting in a change to either 240 or 280, which introduces additional design complexity. Although 8 control points yield the best performance, they also increase attention complexity, regression target dimensionality, and matrix operations—factors that may be limiting in resource-constrained ADAS deployments. Nevertheless, in scenarios where maximum accuracy is critical, the trade-off may be justified. In the experiments of this study, the number of control points is set to 4 for its practicality and to align with the literature [7].

S.6.6. Impact of Backbone Variation

A diverse set of backbone architectures was evaluated for road topology understanding on Subset-A of OpenLane-V2 using the V1.1m baseline. ResNet variants were sourced from the `torchvision` library, while Swin and

Table S7: Performance impact of varying backbone types on road topology understanding in Subset-A of OpenLane-V2, evaluated using the V1.1m baseline.

Backbone Type	DET_{l}	$\text{DET}_{\text{l\_ch}}$	TOP_{ll}	OLS_{l}
ResNet18	36.3	38.0	27.7	42.3
ResNet50	38.9	39.2	29.4	44.1
ResNet101	38.4	41.0	29.9	44.7
ConvNeXt-B	39.4	40.8	30.7	45.2
ConvNeXt-L	39.6	42.1	30.2	45.6
ConvNeXt-B (CLIP)	40.2	43.5	31.4	46.6
SwinB	41.0	45.9	33.1	48.1
Swin-L	41.2	46.1	33.4	48.4
ConvNeXt-XXL (CLIP)	**42.3**	**47.5**	**33.6**	**49.3**

Table S8: Impact of number of epochs and multi-modality on TopoBDA in OpenLane-V2 with V1.1m Metric Baseline.

Subset	Epochs	Sensor	DET_{l}	$\text{DET}_{\text{l\_ch}}$	TOP_{ll}	OLS_{l}
A	24	Camera	41.0	45.9	33.1	48.1
	48	Camera	41.6	46.7	33.9	48.8
	48	Camera + Lidar	<u>51.2</u>	<u>56.7</u>	<u>39.8</u>	<u>57.0</u>
	48	Camera + Lidar + SDMap	**51.3**	**56.8**	**41.3**	**57.4**
B	24	Camera	49.9	50.9	40.3	54.8
	48	Camera	<u>53.2</u>	<u>54.9</u>	<u>43.6</u>	<u>58.0</u>
	48	Camera + Lidar	**60.6**	**63.0**	**49.4**	**64.6**

ConvNeXt models were accessed via the `timm` interface. All Swin and ConvNeXt models were pre-trained on ImageNet-22k, except those annotated with `(CLIP)`, which were pre-trained using the CLIP framework.

As shown in Table S7, Swin backbones consistently outperformed ConvNeXt variants, which in turn surpassed ResNet architectures across all evaluated metrics. CLIP pretraining led to a notable improvement in the performance of ConvNeXt-B, increasing its OLS_{l} from 45.2 to 46.6. The highest overall performance was achieved by ConvNeXt-XXL (CLIP), which recorded an OLS_{l} of 49.3, outperforming all other models in the comparison.

S.6.7. Evaluating the Influence of Epochs, and Multi-Modality on Performance Metrics

In this section, we analyze increasing the number of epochs from 24 to 48 and incorporating multi-modality with camera, lidar and SDMap. Table S8 presents the results of these modifications.

For the OLS_l metric, when the number of epochs is increased from 24 to 48, the OLS_l metric improves. In subset A, it increases from 48.1 to 48.8 (an additional increase of 0.7), and in subset B, it increases from 54.8 to 58.0 (an additional increase of 3.2). Additionally, incorporating sensor fusion with lidar results in the highest performance gains. In subset A, the OLS_l metric increases from 48.8 to 57.0 (an additional increase of 8.2), and in subset B, it increases from 58.0 to 64.6 (an additional increase of 6.6).

Moreover, in Subset-A, the integration of SDMap on top of the camera and lidar fusion further improves the OLS_l metric from 57.0 to 57.4, representing an additional increase of 0.4. Although the detection performance does not improve significantly, the topology performance shows a notable increase of 1.5 in TOP_{ll}. While there is a slight increase in OLS_l, it is less pronounced compared to the results in Table 3. One possible reason for this might be that SDMap facilitates faster convergence, and as the epoch regime reaches 48 epochs, the difference starts to diminish.

S.7. Step-wise Operation of the TopoBDA Decoder

To clearly illustrate the internal workings of our proposed decoder, we present a step-wise breakdown of the TopoBDA decoder module in Algorithm 1. This procedure outlines the iterative refinement of query embeddings, Bezier control points, and mask predictions across multiple decoding layers. Each stage incorporates Bezier Deformable Attention (BDA), self-attention, and feedforward updates. The algorithm also highlights how control points are progressively refined in the inverse sigmoid space.

S.8. Comparative Novelty Analysis Across Road Topology and HDMap Element Prediction Methods

To highlight the broader novelty of TopoBDA, we present a comparative analysis not only against road topology models such as TopoMaskV2, but also against HDMap element prediction methods. Table S9 summarizes architectural, functional, and analytical distinctions.

In addition to the table, we itemize the key contributions of TopoBDA below for clarity:

- **First integration of MPDA into Bézier regression:** TopoBDA is the first to adapt Multi-Point Deformable Attention (MPDA) to Bézier keypoint-based transformer decoders, enabling richer spatial reasoning for polyline structures.

- **Bezier Deformable Attention (BDA):** A novel attention mechanism that directly uses Bézier control points as reference targets, eliminating the need for polyline point conversion and reducing computational complexity.
- **Indirect instance mask formulation:** Unlike TopoMaskV2, which mixes direct and indirect usage, TopoBDA isolates and demonstrates the benefits of indirect formulation for centerline prediction.
- **Mask-L1 mix matcher:** A new matcher strategy tailored for deformable attention architectures, improving convergence and accuracy.
- **Sensor fusion for road topology understanding:** TopoBDA is the first to analyze the impact of combining lidar and radar for this task.
- **SDMap-lidar synergy:** Demonstrates the benefits of combining SDMap with lidar, which has not been explored in prior literature.
- **Comprehensive attention type and complexity analysis:** In addition to SA and MA, includes runtime profiling of SPDA, MPDA, and BDA in both Torch and ONNXRuntime environments.
- **Quantitative evaluation of one-to-many matching strategies:** First to analyze the impact of this strategy on convergence and performance in road topology understanding.

S.9. Visual Results

Figure S1 presents the visual PV and BEV results for samples 'S1' and 'S2'. 'GT' and 'Pred' denote the ground truth and prediction, respectively. The BEV images illustrate the performance of centerline detection, centerline-traffic element topology, and centerline-centerline topology. For additional details, refer to Section 4.1 and Figure 8. TopoBDA was trained for 48 epochs using the camera, lidar, and SDMap fusion options for these visuals. Samples S1 and S2 are also depicted in Figure 9, allowing for an analysis of the impact of the SwinB backbone in conjunction with 48 epochs of training. Compared to that figure, the results indicate an improvement in centerline detection performance at the intersection areas for both S1 and S2 samples.

Algorithm 1 Bezier Deformable Attention Decoder with Iterative Refinement

Require: Multi-scale BEV features $\{\mathbf{F}_{BEV}\}$

Ensure: Final predictions: class logits $\mathbf{C}^{(L)}$, mask logits $\mathbf{P}^{(L)}_{mask}$, Bezier control points $\mathbf{C}^{(L)}_{norm}$, query embeddings $\mathbf{E}^{(L)}_{query}$

Feature Preparation:

Project each $\mathbf{F}_{BEV}$ to hidden dimension and add level embeddings

Extract sine positional encodings $\{\mathbf{P}_{BEV}\}$

Set mask features: $\mathbf{F}_{mask} = \mathbf{F}^{high}_{BEV}$ *(highest-resolution scale)*

Initialize learnable query embeddings $\mathbf{E}^{(0)}_{query}$

Initial Prediction:

Predict initial control points and mask logits:

$$\mathbf{C}^{(0)}_{norm} = \sigma(\mathrm{MLP}^{(0)}_{B}(\mathbf{E}^{(0)}_{query})), \quad \mathbf{P}^{(0)}_{mask} = \mathbf{F}_{mask} \cdot \mathrm{MLP}^{(0)}_{M}(\mathbf{E}^{(0)}_{query})$$

for $l = 0$ to $L - 1$ **do**

Reference Point Construction:

Reshape $\mathbf{C}^{(l)}_{norm}$ to $(B, N_q, N_{ctrl}, 3)$ and discard height:

$$\mathbf{R}^{(l)} = \mathbf{C}^{(l)}_{norm}[:, :, :, : 2]$$

Positional Embedding:

Generate sine embeddings from $\mathbf{R}^{(l)}$ and apply MLP:

$$\mathbf{P}^{(l)} = \mathrm{MLP}_{pos}(\mathrm{sine}(\mathbf{R}^{(l)}))$$

Bezier Deformable Attention:

Compute query input: $\mathbf{Q}^{(l)}_{BDA} = \mathbf{E}^{(l)}_{query} + \mathbf{P}^{(l)}$

Apply BDA:

$$\mathbf{A}^{(l)}_{BDA} = \mathrm{BDA}(\mathbf{Q}^{(l)}_{BDA}, \{\mathbf{F}_{BEV}\}, \mathbf{R}^{(l)})$$

Self-Attention and FFN:

Apply multi-head self-attention and feedforward network:

$$\mathbf{E}^{(l+1)}_{query} = \mathrm{FFN}(\mathrm{SelfAttention}(\mathbf{A}^{(l)}_{BDA}, \mathbf{P}^{(l)}))$$

Bezier Control Point Refinement:

Predict delta and update in inverse sigmoid domain:

$$\Delta\mathbf{C}^{(l+1)} = \mathrm{MLP}^{(l+1)}_{B}(\mathbf{E}^{(l+1)}_{query}), \quad \mathbf{C}^{(l+1)}_{norm} = \sigma(\sigma^{-1}(\mathbf{C}^{(l)}_{norm}) + \Delta\mathbf{C}^{(l+1)})$$

Mask Logit Refinement:

Predict mask embedding and update logits:

$$\mathbf{E}^{(l+1)}_{mask} = \mathrm{MLP}^{(l+1)}_{M}(\mathbf{E}^{(l+1)}_{query}), \quad \mathbf{P}^{(l+1)}_{mask} = \mathbf{P}^{(l)}_{mask} + \mathbf{F}_{mask} \cdot \mathbf{E}^{(l+1)}_{mask}$$

Class Prediction:

Predict class logits:

$$\mathbf{C}^{(l+1)} = \mathrm{MLP}^{(l+1)}_{cls}(\mathbf{E}^{(l+1)}_{query})$$

end for

Return $\mathbf{C}^{(L)}$, $\mathbf{P}^{(L)}_{mask}$, $\mathbf{C}^{(L)}_{norm}$, $\mathbf{E}^{(L)}_{query}$

Table S9: Comparative Analysis of TopoBDA, TopoMaskV2, and Other Baselines in Road Topology Understanding

Aspect	**Other Baselines**	**TopoMaskV2**	**TopoBDA (Ours)**	**Novelty Highlight**
Attention Mechanism	SPDA, MPDA, masked attention	Masked attention	Bezier Deformable Attention (BDA)	First use of Bezier curves for flexible multi-point attention in general polyline generation literature
MPDA Integration	MPDA used, but not with Bezier	Not explored	MPDA integrated with Bezier regression	Novel combination for richer spatial reasoning
Instance Mask Formulation	Explored for HDMap elements (e.g., lanes, signs), not for centerlines or road topology	Mixed direct/indirect usage	Isolated indirect formulation for centerlines	First to show indirect formulation boosts Bezier head performance
Matcher Strategy	Mostly L1 matcher	Mask matcher	Mask-L1 mix matcher	Tailored matcher for deformable attention architectures
Sensor Fusion (Road Topology)	Absent	Absent	Fusion of lidar and radar	First sensor fusion analysis for road topology understanding
SDMap Usage (Road Topology)	SDMap-only (no fusion)	Not used	SDMap used with lidar	First to demonstrate SDMap-lidar synergy for topology prediction
Attention Type Comparison & Complexity (Road Topology)	Absent	Limited	Comparative analysis of more attention types with Torch and ONNXRuntime runtime profiling	First to broaden attention type evaluation with computational complexity analysis in this domain
Matching Strategy Evaluation (Road Topology)	Used but not evaluated	Used but not evaluated	Quantitative evaluation of one-to-many matching	First to analyze matching strategy impact in road topology understanding

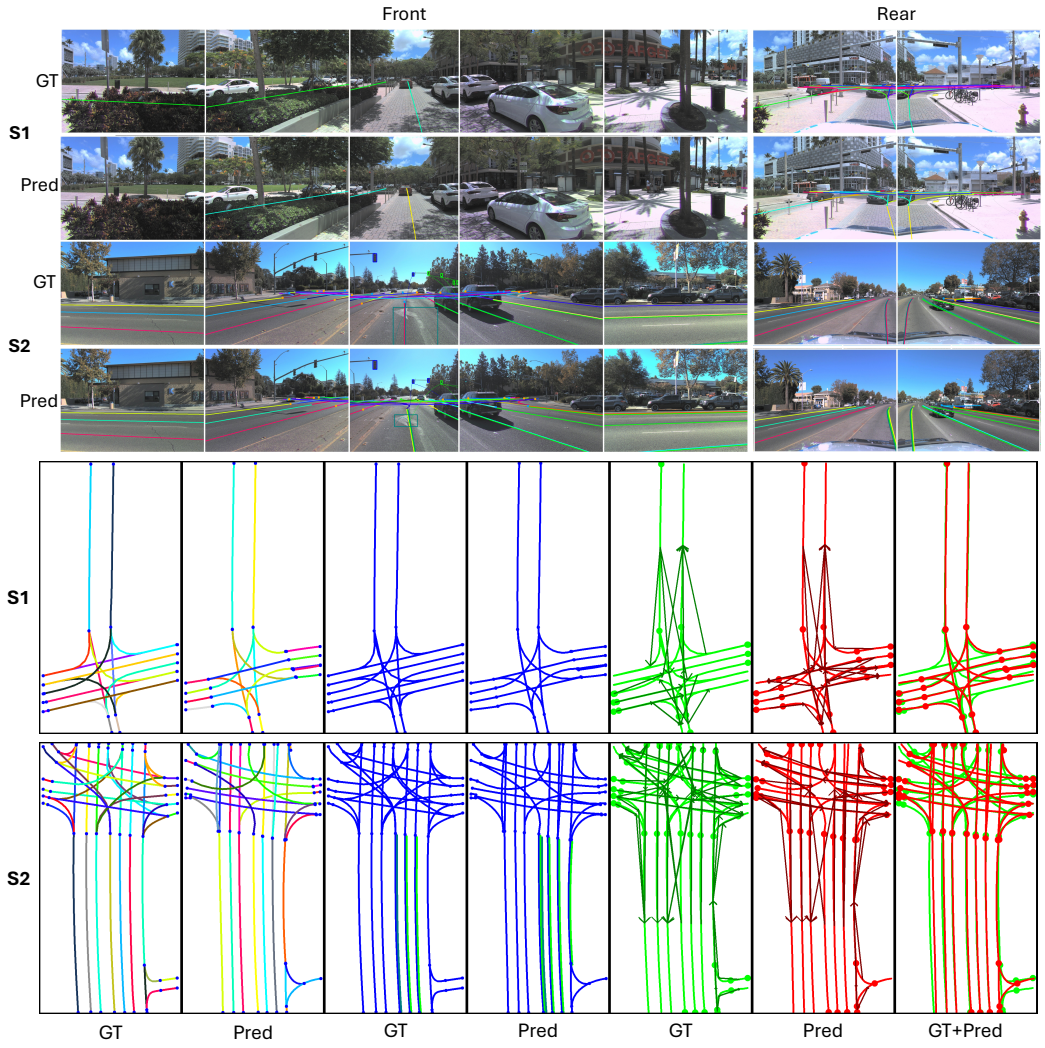


Figure S1: Visual results demonstrating the performance of TopoBDA trained for 48 epochs with camera, lidar, and SDMap fusion. The visuals include perspective images at the top and BEV images at the bottom. 'GT' and 'Pred' denote the ground truth and predictions, respectively.