End-to-End Vectorized HD-map Construction with Piecewise Bézier Curve

Limeng Qiao Wenjie Ding Xi Qiu* Chi Zhang
MEGVII Technology
{qiaolimeng, dingwenjie, qiuxi, zhangchi}@megvii.com

Abstract

*Vectorized high-definition map (HD-map) construction, which focuses on the perception of centimeter-level environmental information, has attracted significant research interest in the autonomous driving community. Most existing approaches first obtain rasterized map with the segmentation-based pipeline and then conduct heavy post-processing for downstream-friendly vectorization. In this paper, by delving into parameterization-based methods, we pioneer a concise and elegant scheme that adopts unified piecewise Bézier curve. In order to vectorize changeful map elements end-to-end, we elaborate a simple yet effective architecture, named Piecewise Bézier HD-map Network (**BeMapNet**), which is formulated as a direct set prediction paradigm and postprocessing-free. Concretely, we first introduce a novel IPM-PE Align module to inject* 3D *geometry prior into BEV features through common position encoding in Transformer. Then a well-designed Piecewise Bézier Head is proposed to output the details of each map element, including the coordinate of control points and the segment number of curves. In addition, based on the progressively restoration of Bézier curve, we also present an efficient Point-Curve-Region Loss for supervising more robust and precise HD-map modeling. Extensive comparisons show that our method is remarkably superior to other existing SOTAs by* 18.0 *mAP at least* [1].

1. Introduction

As one of the most fundamental components in the auto-driving system, high-definition map contains centimeter details of traffic elements, vectorized topology and navigation information, which instruct ego-vehicle to accurately locate itself on the road and understand what is coming up ahead. At present, conventional *SLAM-based* solutions [45, 46, 60] have been widely adopted in practice. Yet, due to dilemmas of high annotation costs and untimely updates, the offline approach is gradually being replaced by the learning-based online *HD-map* construction with onboard sensors.

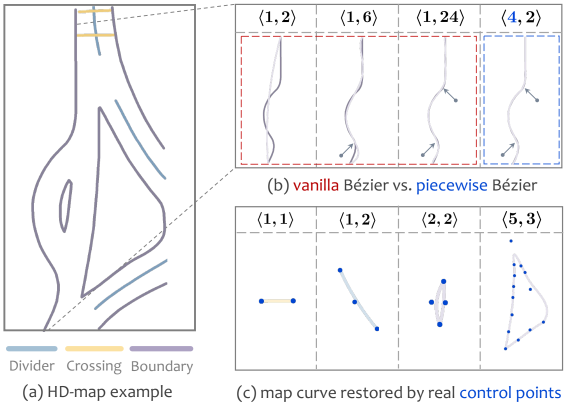

Figure 1. Illustration of our motivation for piecewise Bézier curve, termed as $\langle \boldsymbol{k}, \boldsymbol{n} \rangle$, where $\boldsymbol{k}$ is the piece number and $\boldsymbol{n}$ is the degree. ***Fig.(a)*** is a real *HD-map* case from *NuScenes*. ***Fig.(b)*** compares the difference between vanilla and piecewise Bézier curve through the same map element, where the light purple is the restored curve with Bézier process. The last is more efficient than previous ones with reducing the number of control points by $\mathbf{64}\%$ in this case. ***Fig.(c)*** illustrates that piecewise Bézier curve can model arbitrary-shaped curves. Note the blue circles denote actual control points.

The deep-based paradigm of online *HD-map* building is gradually developing, but it still faces two main challenges: ***1)*** *modeling instance-level vectorized HD-map end-to-end.* Most existing works construct *HD-map* by rasterizing *BEV* (bird-eye-view) maps into semantic pixels with segmentation [24, 42], which not only lacks the modeling of instance-level details, but also requires heavy post-processing to obtain vectorized information. As a sub-task, lane detection makes a relatively better advance for this issue, that is, in addition to segmentation-based methods [39, 41, 62], there are also point-based [25, 47] and curve-based [12, 28] schemes. However, compared to the simple lane scenario, *HD-map* contains more shape-changeful elements, so such methods cannot be directly adopted into the *HD-map* construction. ***2)*** *performing* 2*D-*3*D perspective transformation efficiently.* Obtaining 3*D-BEV* perception from multi-view 2*D* images is an essential step for building *HD-map*, which is mainly divided into three ideas, *i.e*. geometric priors [44], learnable parameters [40, 42], and a combination of the two [17, 43]. Note the assumptions of geometry-based methods often do

*Corresponding author.

[1] https://github.com/er-muyue/BeMapNet

not conform to the actual situation, leading to such schemes are less adaptable, while learning-based methods require a large amount of labeled data to generalize across various scenarios. Combining the above two branches not only has multi-scenario scalability, but also reduces the demand for annotated data, has attracted increasing research interest.

To the best of our knowledge, the curve parameterization construction of *HD-map* in the *BEV* space is vacancy and no one has explored it. Based on the widely used Bézier curve, which is mathematically defined by a set of control points, we pioneer to devise a concise and elegant *HD-map* scheme that adopts ***piecewise*** Bézier curve, where each map curve is divided into multiple $\boldsymbol{k}$ segments and each segment is then represented by a vanilla Bézier curve with degree $\boldsymbol{n}$, hence denoted as $\langle \boldsymbol{k}, \boldsymbol{n} \rangle$. Despite $\langle \mathbf{1}, \boldsymbol{n} \rangle$ is enough to express any map element with infinite $\boldsymbol{n}$ in theory, more complex curve tends to require higher degree, meaning that there are more control points need to be modeled, which is shown in Fig.1. The proposed piecewise strategy allows us to parameterize a curve more compactly with fewer control points and higher capacity, which is extremely scalable and robust in practice.

Inspired by the above motivations, we propose an end-to-end vectorized *HD-map* construction architecture, named as *Piecewise* **Bé**zier *HD-**map*** **Net**work (***BeMapNet***). The overall framework is illustrated in detail in Fig.2, which streamlines the architecture into four primary modules for gradually-enriched information, *i.e.* feature extractor shared among multi-view images, semantic *BEV* decoder for 2*D*-3*D* perspective elevation, instance Bézier decoder for curve-level descriptors, and piecewise Bézier head for point-level parameterization. To be concrete, we first introduce a novel *IPM-PE Align module* into Transformer-based decoders, which injects *IPM* (inverse perspective mapping) geometric priors into *BEV* features via *PE* (position encoding) and hardly adds any parameters except a *FC* layer. Secondly, we further design a *Piecewise Bézier Head* for dynamic curve modeling with adopting two branches as classification and regression, where the former classifies the number of piece to determine the curve length and the latter regresses the coordinate of control points to determine the curve shape. Lastly, we present an *Point-Curve-Region Loss* for robust curve modeling by supervising restoration information as a progressive manner. Since it is modeled as a sparse set prediction task and optimized with a bipartite matching loss, our method is postprocessing-free and high-performance. The main contributions of our approach are three-folds:

- We pioneer the ***BeMapNet*** for concise and elegant modeling of *HD-map* with unified piecewise Bézier curve.
- We elaborate the overall end-to-end architecture with innovatively introducing *IPM-PE Align Module*, *Piecewise Bézier Output Head* and well-designed *PCR-Loss*.
- ***BeMapNet*** is remarkably superior to *SOTAs* on existing benchmarks, revealing the effectiveness of our approach.

2. Related Work

2.1. Vectorized *HD-map* Construction

HD-map is precise at centimeter-level and consists of vectorized details not normally present on standard map [27], which highly aggravates the difficulty of obtaining accurate annotations in practice. Most previous methods focus on SLAM [11, 18, 22, 36, 46, 58] with exploring LiDAR points of the environment, involving large-scale data acquisition and labor-intensive annotations. Recently, more and more researchers formulate the process of building *HD-map* as a segmentation task [6, 13, 20, 32] from various sensors, such as cameras [15, 26, 34, 35, 40, 43, 59, 63] and LiDAR [1, 57]. HDMapNet [24] extracts features from the observations of multi-modalities and groups semantic rasterized maps with heuristic post-processing for vectorized results. Without any further vectorization, [42] only obtains semantic map representations from a Transformer-based camera-to-BEV module. Different from the above segmentation-based frameworks, our proposed ***BeMapNet*** adopts the parameterization-based paradigm and constructs instance-level vectorized *HD-map* end-to-end with a multi-view and camera-only manner, which is a more flexible and scalable solution for downstream tasks.

2.2. Structure Modeling of Geometric Data

HD-map contains various kinds of map elements, including $\textit{lane-divider}$, $\textit{ped-crossing}$, $\textit{road-boundary}$, *etc*., which are typically regarded as geometric data, *e.g.* points, polygons and curves. To the best of knowledge, there are two mainstream exploration directions for deep geometric modeling, one is point-based, including uniform-points and keypoints estimation, and the other is curve-based, including polynomial curve and Bézier curve. Taking the most related lane detection as an example, LineCNN [25] presents a novel line proposal unit to detect lanes as a set of points, and later LaneATT [47] represents a lane by equally-spaced 2*D*-points that achieves good performance. GANet [54] focuses on keypoint estimation and association with adaptively lane feature aggregator. HDMapGen [37] proposes a hierarchical point generative model for producing high-quality HD lane map. As for cruve-based, [51] and [48] directly predict the polynomial coefficients with a differentiable least-squares fitting module and simple *FC* respectively. Recently, based on the DETR [4], LSTR [28] presents a query-based detector to decode poly-parameters of a lane end-to-end. Apart from this, some studies also adopt Bézier curve for geometry, such as text boundary [29], center-line segment [3] and lane curve [12]. Unlike these approaches of addressing simple-shape elements, our method innovatively employs piecewise Bézier curves without any complicated geometric assumptions to parameterize arbitrary-shape geometry in *HD-map* scenarios.

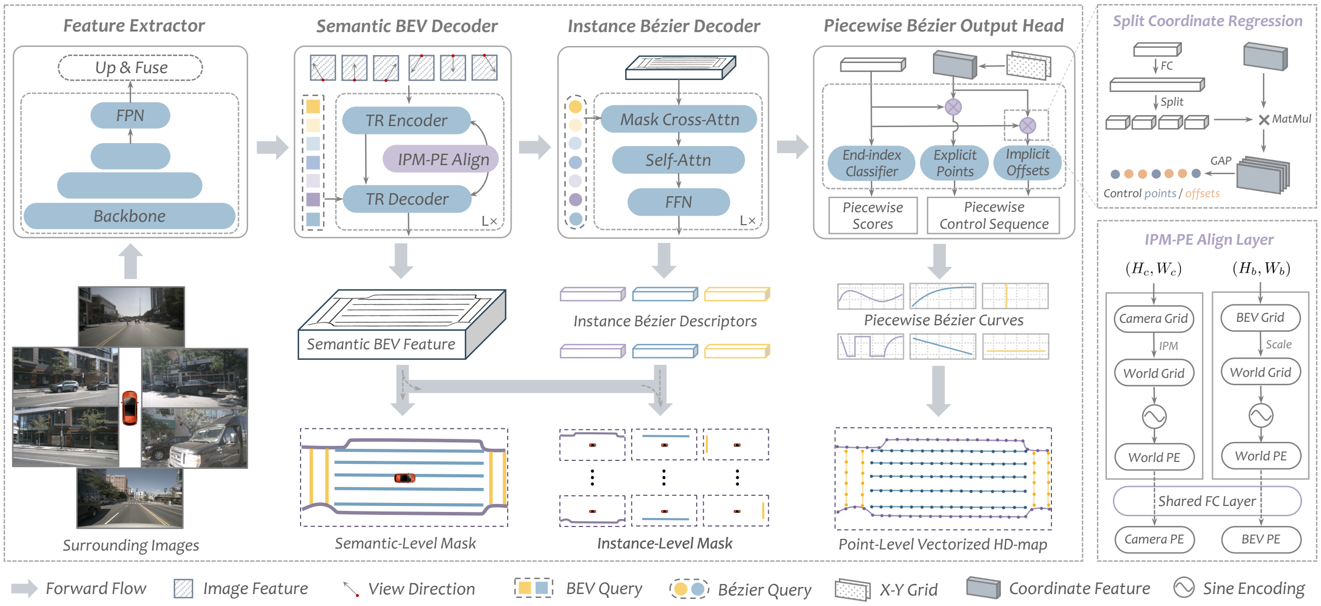

Figure 2. The architecture of our proposed ***BeMapNet***, containing four primary components for extracting progressively richer information, *i.e. image-level* multi-scale features, *semantic-level BEV* feature, *instance-level* curve descriptors, and *point-level* Bézier control sequence. Right-top: The blue and orange circles represent explicit control points and implicit control offsets respectively. Note that the term *GAP* is *global average pooling* and *MatMul* is the *matrix multiplication*. Right-bottom: $(H_c, W_c)/(H_b, W_b)$ is the shape of image/*BEV* feature.

2.3. Multi-view Camera-to-BEV Transformation

In most 3*D* research studies, obtaining high-quality *BEV* representations from multi-view camera features is the top priority of various domains, such as 3*D* object detection [17,30,43,55,56], motion prediction [10,16,19,23,53] and map construction [24,26,40,42]. IPM [44] is the most basic and straightforward method with a homography transformation, which is precisely calculated by camera parameters. Without adopting the 3*D* geometry prior, some methods [3, 40, 42] directly leverage learnable parameters to complete the perspective transformation. VPN [40] and Neat [9] both utilize a *FC* layer to transform the image features into the *BEV* space, and [42] further designs a multi-camera deformable attention unit based on transformer. However, in order to obtain more explicit *BEV* representations, more and more researches argue that the geometry priors has great advantages for model convergence and performance. Based on VPN, [24] fuse the multi-camera *BEV* spaces with the camera poses. LSS [43] and BEVDet [17] build the connection between camera-view and *BEV* based on the depth distribution estimation. DETR3D [56] manipulates predictions directly in 3*D* space with linking 3*D* positions to 2*D* spaces and [5] generates *BEV* features by regarding camera parameters as a reference. PETR [30] encodes the 3*D* information as position embedding and conducts query decoding on position-aware features. In addition to introducing the 3*D* geometry prior, we further performs position embedding alignment for transforming the camera features and *BEV* features into the world coordinate system.

3. Method

3.1. Problem Formulation

Preliminary on Piecewise Bézier Curve. A Bézier curve is a parametric curve which is formulated by a set of ordered control points c_0 through c_n as,

$$p(t) = \sum_{i=0}^{n} b_{i,n}(t)c_i, \; t \in [0, 1] \tag{1}$$

where n is the degree of the curve and $b_{i,n}(t)$ is known as Bernstein basis polynomial of degree n,

$$b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}, \; i = 0, \ldots, n \tag{2}$$

According to Eq.1, we know the first and last control points are endpoints of the curve, *i.e.* $p(0) = c_0, \; p(1) = c_n$. Then we define a piecewise Bézier curve $\langle \boldsymbol{k}, \boldsymbol{n} \rangle$ consists of k segments, each of which is an n-*order* Bézier and two consecutive segments satisfy the positional continuity condition as $p^j(1) = p^{j+1}(0)$, where j is the segment *id* varying from 0 to k-2. Hereafter, given a piecewise Bézier curve as shown in Fig.3, we denote its ordered control points as,

$$\mathbb{C} = \{c_i^j \in \mathbb{R}^2 | i \in [0, n], j \in [0, k-1], c_n^j = c_0^{j+1}\} \tag{3}$$

Moreover, due to the first and last control points of each segment are always on the curve, we naturally term these points that retain a distinct significance as *explicit control points* $\mathbb{C}^E$ and the remaining as *implicit control points* $\mathbb{C}^{\mathcal{I}}$, where $\mathbb{C} = \mathbb{C}^E \cup \mathbb{C}^{\mathcal{I}}$, $|\mathbb{C}^E| = k + 1$ and $|\mathbb{C}^{\mathcal{I}}| = nk - k$.

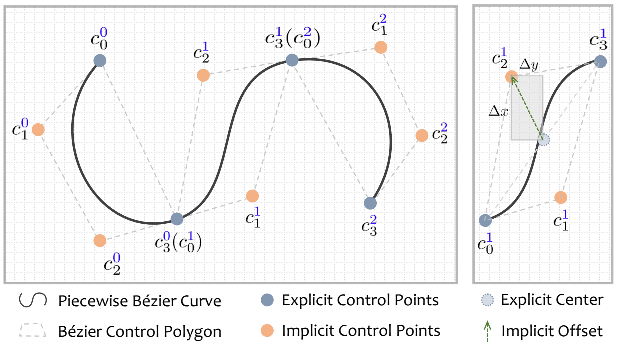

Figure 3. Illustration of a *3-pieces 3-order* piecewise bézier curve, denoted as $\langle \mathbf{3}, \mathbf{3} \rangle$ for the left part. For each segment, we model the implicit control point coordinates by an offset $(\Delta x, \Delta y)$ from the center of explicit control points as shown on the right part.

Vectorized *HD-map* Modeling. By sequentially sampling points along the elements in *HD-map*, each object can be structured as an open-shape curve as a vectorized manner. To uniformly parameterize these various shape-changeful map elements, including points, polygons and curves, we employ the piecewise Bézier curve proposed in Eq.3 and formalize a vectorized *HD-map* as a piecewise Bézier curve set $\mathcal{M} = \{\mathbb{C}_i | i = 1, \ldots, |\mathcal{M}|\}$, where $|\mathcal{M}|$ is the number of map instances. Our objective is to learn a model that extracts compact information from the camera sensors and predicts, for each map element, its corresponding Bézier control sequence, which uniquely determine the shape and position of the map curve. In addition, for the purpose of reducing the modeling complexity of implicit control points $\mathbb{C}^{\mathcal{I}}$ for each segment, we subtly generate implicit coordinates through the center of explicit control points and their relative offsets as shown in the right of Fig.3.

Piecewise Bézier Curve Modeling Principle. Considering the large-variety and changeful-shape of map elements, we present a ***consistent-degree but dynamic-piece*** principle for modeling piecewise curve more precisely and efficiently as, (1) ***consistent-degree on semantic-level***. Due to a Bézier curve of degree k can be easily converted into degree $k + 1$ with the same shape, *i.e*. *degree elevation* in mathematics, the same curve may correspond to multiple Bézier solutions with only different degree. This kind of obfuscation is devastating for the network modeling. And we argue that keeping the *consistent-degree* within the same semantic category is the most efficient way to solve this dilemma.

(2) ***dynamic-piece on instance-level***. Considering that the complexity of different map curve is always inconsistent, the modeling method with a fixed and large number of segments will not only cause excessive redundancy in the expression of each curve, but also lead to confusion in the definition of sub-segment, which places a certain burden on the model optimization. The *dynamic-piece* rule demands the model to choose the number of segments dynamically and parameterize a Bézier curve as compactly as possible.

3.2. Piecewise Bézier HD-map Network

3.2.1 Overall Architecture

The overall model architecture is illustrated in detail in Fig. 2 , which streamlines the framework into four parts, namely feature extractor, semantic *BEV* decoder, instance Bézier decoder and piecewise Bézier output head respectively.

Feature Extractor. Given surrounding images from multi-camera as inputs, a shared CNN backbone is first employed to obtain each image feature, and then these multi-scale features from different stages are fed into FPN [50] to integrate rich environmental information. Last we upsample pyramid features to the same size and stack them together as outputs.

Semantic *BEV* Decoder. We leverage a standard encoder-decoder paradigm based on Transformer to elevate camera-view features into canonical *BEV* spaces. By treating the perspective transformation as a direct set prediction task, the *BEV* decoder takes camera features with shape $H_c \times W_c$ and $H_q \times W_q$ learnable *BEV* queries as inputs, and produces *BEV* features $F_b \in \mathbb{R}^{C \times H_b \times W_b}$ by modeling all pairwise interactions among elements with self- and cross-attention. Different from the region-agnostic query in DETR, we correspond each query with a $\gamma^h \times \gamma^w$ region on *BEV* features one-to-one, *i.e*. $H_b \times W_b = \gamma^h H_q \times \gamma^w W_q$. The F_b is then fed into a 1×1 *conv* and *upsample* block to obtain the final semantic mask $\mathbb{M}_s \in \mathbb{R}^{U \times H_s \times W_s}$, where U is the number of classes and $H_s \times W_s$ is the map shape. In addition, we also propose a novel ***IPM-PE Align Layer*** to refactor the features in different coordinate systems through the most common position encoding layer in Transformer.

Instance Bézier Decoder. So as to perform more accurate parametric modeling of each curve, we further equip an instance Bézier decoder based on masked cross-attention [8]. To be concrete, given semantic features F_b and learnable instance Bézier queries $Q \in \mathbb{R}^{V \times C}$, where V is the max number of instance, this module aggregates information and decodes Bézier descriptors $\vec{z} \in \mathbb{R}^{V \times C}$, which contain key info of geometric and positional relationships between different map elements. With extra performing matrix multiplication between $\vec{z}$ and F_b, we obtain instance map mask $\mathbb{M}_z \in \mathbb{R}^{V \times H_s \times W_s}$, which is used as the foreground mask for next decoder layer and also to perform segmentation supervision for more spatial context information.

Piecewise Bézier Output Head. Following the principle elaborated in Sec.3.1, we closely design a piecewise Bézier output head with two core modules, *i.e*. ***Split Coordinate Regression*** and ***Dynamic End-index Classification***, which are utilized to output the coordinates of Bézier control point sequence and the number of segments respectively. With easily recovering piecewise Bézier curves and further introducing the bipartite matching setting, the proposed output head constructs vectorized local *HD-map* very efficiently.

3.2.2 IPM-PE Align Layer

As a basic module of traditional Transformer architecture, positional encoding utilizes the sequence order by injecting information about the position of tokens [52], where *sin-cos* and *learned-based* functions are the most common practices. Yet, for the purpose of perspective transformation between $2D$ camera-views and $3D$ *BEV*, we argue it is not enough to only encode the *position correspondence* within single view, but it is also necessary to maintain the *position consistency* relationships between two perspectives. Thence we put forward a novel *IPM-PE Align Layer* to encode the $2D$-$3D$ geometry priors into features from different coordinate systems. To be concrete, given a point $p_c^{f_i} = (u, v, 1)^\top$ on the i-th camera-view feature and its corresponding world point $p_c^w = (x, y, z)^\top$, the following mathematical equation is satisfied on the assumption of pinhole camera model,

$$d \cdot A^{-1} \cdot p_c^{f_i} = K^i \cdot T^i \cdot p_c^w \tag{4}$$

where d is the depth, A is the transformation matrix between image-grid and feature-grid, K^i and T^i are i-th intrinsic and extrinsic matrices. Based on the assumption of that the ground surface is flat and at a fixed height in *IPM*, the world position (x, y) and depth d can be easily calculated through Eq.4. Note $d < 0$ denotes the position is not valid. As for the other branch of *BEV* perspective, given a point p_b^f from the *BEV* feature F_b and its world point p_b^w, there is usually only a scale relationship $\boldsymbol{\kappa} = (\kappa_x, \kappa_y)$ between them,

$$p_b^f = \boldsymbol{\kappa} \cdot p_b^w \tag{5}$$

Then we leverage the standard *sin-cos function* to convert all these world coordinates P_c^w and P_b^w as position embedding f_c^{pe} and f_b^{pe} respectively. Since the assumption of flat-ground and known-height usually does not hold in practice, f_c^{pe} and f_b^{pe} are not exactly aligned in the world coordinate system, we further adopt a shared *FC* layer on f_c^{pe} and f_b^{pe} to perform embedding alignment and then obtain more unified position encoding, which is shown in Fig.2 in detail.

3.2.3 Piecewise Bézier Output Head

Split Coordinate Regression Head. Following the illustration in the upper right corner of the Fig.2, we elaborate the forward flow of proposed regression head as four steps, ***1)*** with i-th incoming Bézier descriptor $\vec{z}_i \in \mathbb{R}^C$, we first convert its channel from C to $u \cdot v$ with adopting a standard *FC* layer and then split it into u parts to get a collection of coordinate descriptors $x_i^j \in \mathbb{R}^v$, where $j \in [1, u]$, u is the number of coordinate and v is the channel of descriptor.
2) given the desired output shape (H_s, W_s), a hard-coded candidate coordinates grid $G \in \mathbb{R}^{2 \times H_s \times W_s}$ is generated with X-Y two channels and next a 1×1 *conv* layer is utilized to convert G to its coordinate feature $F_G \in \mathbb{R}^{v \times H_s \times W_s}$.
3) through conducting the matrix multiplication between the coordinate descriptor x_i^j and feature F_G, the coordinate activation map $h_i^j \in \mathbb{R}^{H_s \times W_s}$ is obtained dynamically.
4) after using a global average pooling on the spatial of h_i^j, the activation map is regressed to the final coordinate value.

Dynamic End-index Classification Head. Based on the prior of that the end point of a piecewise Bézier curve must be explicit, the proposed module model the dynamic length of segments prediction task as a N-classification problem, where N is the maximum pieces number for a certain map class. Specifically, with using a common *FC* and *softmax* block, each Bézier descriptor is naturally transformed into a N-dimensional probability vector, where each position denotes the score of the current index as a termination point. This novel indeterminate length modeling greatly increases the adaptability and scalability of our proposed framework.

3.3. End-to-End Training

According to the matrix form of Eq.1, that is, $\boldsymbol{P} = \boldsymbol{BC}$, where $B \in \mathbb{R}^{m \times n}$ is the Bernstein matrix, $C \in \mathbb{R}^{n \times 2}$ is control points, $P \in \mathbb{R}^{m \times 2}$ is vectorized points on the curve, m is the number of points and n is the degree. Obviously, given m, n, we can easily implement following procedures:
1) vectorization. Given the C, its corresponding vectorized curve can be restored efficiently with matrix multiplication.
2) construction. Given the orderly sampling points P, we readily construct C with solving the least-squares problem, *i.e.* $C = B^+ P$ where $^+$ is the pseudo-inverse of a matrix.

Piecewise Bézier Ground Truth. The common *HD-map* annotation protocol always represents a curve with a set of vectorized points. We propose to first select some annotated keypoints and then divide the curve into k segments, where each is compactly modeled by an n-order Bézier curve. Following the procedure of ***construction***, we present the ground truth generation algorithm in the Algorithm 1.

Algorithm 1: Piecewise Bézier Curve ***GenGT***.

```
input  : Annotated points $P$, Parameters $n$, $m$, $\epsilon$
output: Piecewise Bézier curves $\mathbb{C}$
$B$ = GetBernsteinCoefficient ($n, m$);
$B^+$ = MatrixPseudoInverse ($B$);
$l \leftarrow |P|, s \leftarrow 0, e \leftarrow l - 1$;
while $s < e$ do
    $P^\dagger$ = CurveInterpolate ($P[s, e]$);
    $\mathbb{C}^\dagger = B^+ \times P^\dagger$;
    $P^\ddagger = B \times \mathbb{C}^\dagger$;
    $D' =$ ChamferDistance ($P^\dagger$, $P^\ddagger$);
    if $D' < \epsilon$ then
        $\mathbb{C} \leftarrow [\mathbb{C}, \mathbb{C}^\dagger], s \leftarrow e, e \leftarrow l - 1$;
    else $e \leftarrow e - 1$ ;
```

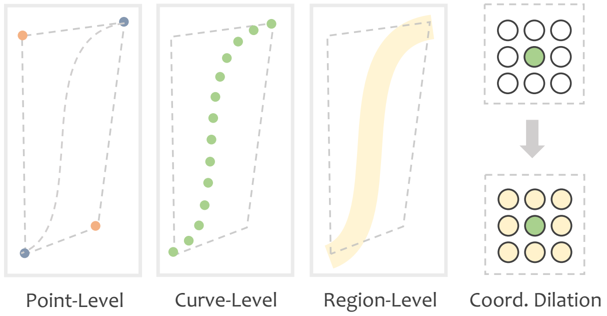

Figure 4. The first three diagrams are illustrations of *Point-Curve-Region Recovery Loss* for Bézier curve modeling. On the far right is a simple schematic diagram of coordinate dilation with $\omega = 1$.

Point-Curve-Region Recovery Loss. Given the Bézier control points, a novel ***PCR-Loss*** is proposed to supervise the learning procedure from three progressive levels, *i.e.* point, curve and region, which is shown in Fig.4 in detail.

1) *point-level supervision.* Assume the prediction of control sequence from the output head as $\widehat{\mathbb{C}}$ and its corresponding ground truth is $\mathbb{C}$, we firstly leverage the $L1$-loss between these coordinates sequentially as,

$$\mathcal{L}_{point} = \frac{1}{|\widehat{\mathbb{C}}|} \sum_{i=1}^{|\widehat{\mathbb{C}}|} \|\mathbb{C}_i - \widehat{\mathbb{C}}_i\|_1, \tag{6}$$

2) *curve-level supervision.* Due to the parametric property of Bézier curve that a slight deviation of control points $\widehat{\mathbb{C}}$ might cause a great change of the curve, we further restore the control sequence $\widehat{\mathbb{C}}$ (or $\mathbb{C}$) to curve sequence $\widehat{\mathcal{P}}$ (or $\mathcal{P}$) with the ***vectorization*** process and adopt the $L1$-loss as,

$$\mathcal{L}_{curve} = \frac{1}{|\widehat{\mathcal{P}}|} \sum_{i=1}^{|\widehat{\mathcal{P}}|} \|\mathcal{P}_i - \widehat{\mathcal{P}}_i\|_1 \tag{7}$$

3) *region-level supervision.* For digging out more intuitive supervision information, we further recover curve $\widehat{\mathcal{P}}$ from the form of discrete coordinates into region mask. First of all, we introduce a coordinate dilation operator, which takes the current coordinate p_{ij} as the center and then generates $(2\omega+1)^2$ surrounding coordinates $p_{\alpha\beta}$ with the giving dilation width ω, where $\alpha \in [i-\omega, i+\omega]$ and $\beta \in [j-\omega, j+\omega]$. Secondly, we conduct the coordinate dilation on each point from curve $\widehat{\mathcal{P}}$ and then obtain dilated curve points sequence $\widehat{\mathcal{P}}^\sharp$, which can be regarded as the foreground coordinates of the predicted mask $\widehat{M}$, *i.e.* $\mathcal{S}(\widehat{M}, \widehat{\mathcal{P}}^\sharp) = \mathbf{1}$, where $\mathcal{S}$ is the grid sampling operation to compute the output values using $\widehat{M}$ and point coordinates $\widehat{\mathcal{P}}^\sharp$ from grid. Finally, after preforming the grid sampling $\mathcal{S}$ on ground truth map mask M with the same dilated prediction curve, we then bridge a segmentation-base region supervision between M and $\widehat{\mathcal{P}}^\sharp$, which can be formulated mathematically as,

$$\mathcal{L}_{region} = \mathcal{L}_{dice}(\mathcal{S}(M, \widehat{\mathcal{P}}^\sharp),\ \mathcal{S}(\widehat{M}, \widehat{\mathcal{P}}^\sharp)) \tag{8}$$

where $\mathcal{L}_{dice}$ is the common dice loss function in [38].

4) *overall* ***PCR-Loss***. In order to exert the above three-level supervision at the same time, we put forward that the overall *PCR-Loss* is a weighted sum of all three losses as,

$$\mathcal{L}_{PCR} = \lambda_p \mathcal{L}_{point} + \lambda_c \mathcal{L}_{curve} + \lambda_r \mathcal{L}_{region} \tag{9}$$

Multi-task Auxiliary Loss. The paradigm of multi-task learning can reduce the risk of overfitting by leveraging the domain-specific information included in the training signals of related tasks [61]. Thence, in addition to supervising the curve-level Bézier modeling procedure of the final head, we also perform auxiliary segmentation-based supervision on the intermediate modules, *i.e.* semantic-level *BEV* decoder and instance-level Bézier decoder. Given the output masks $\widehat{\mathbb{M}}_s$ and $\widehat{\mathbb{M}}_z$ in the Sec.3.2, we formulate the auxiliary loss with the combination of two tasks supervision as,

$$\mathcal{L}_{AUX} = \lambda_s \mathcal{L}(\mathbb{M}_s, \widehat{\mathbb{M}}_s) + \lambda_z \mathcal{L}(\mathbb{M}_z, \widehat{\mathbb{M}}_z) \tag{10}$$

where $\mathbb{M}_\star$ denotes the ground truth and $\lambda_\star$ is the weighted factor. Note that $\mathcal{L}$ is a compound loss with common cross entropy loss and dice loss, namely $\mathcal{L} = \mathcal{L}_{ce} + \mathcal{L}_{dice}$.

4. Experiments

4.1. Experimental Settings

Existing Benchmarks. To evaluate the proposed approach, we conduct experiments on the popular *NuScenes* dataset [2], which consists of $28,130/6,019$ samples and $700/150$ driving scenes for the training/validation set respectively. Each scene contains roughly 40 samples and each sample includes 6 surrounding images, covering 360° FOV of the ego-vehicle. For the sake of fair comparison, we follow the previous work [24] and focus on three static map categories, *i.e.* $lane\text{-}divider$, $ped\text{-}crossing$ and $road\text{-}boundary$. Taking the ego-vehicle as the center, we set the perception range to $[30, 30, 15, 15]m$, which corresponds to the distances of front, rear, left and right respectively, and fix the resolution of ego-to-pixel as $0.15\ m/pixel$. In addition, for exploring the performance of our method under different lighting and weather conditions, we further divide *NuScenes* into five kinds of scene, namely *day*, *night*, *sunny*, *cloudy* and *rainy*. Refer to the supplementary material for more details.

Evaluation Metrics. We utilize the exact same evaluation protocol as [24] of average precision (AP) to access the map construction quality over the instance-level. To be concrete, given a pair of instances from ground-truth and predictions respectively, this protocol computes the Chamfer Distance between them and considers the prediction as true-positive only if the distance is less than a specified threshold, which is set to $[0.2, 0.5, 1.0]m$ in our experiment. Note the overall AP metric is obtained by averaging across three thresholds. Moreover, as for a more informative comparison, the results under different lighting/weather conditions and a much simpler threshold setup of $[0.5, 1.0, 1.5]m$ are further provided.

Method	Backbone	Epoch	AP$_{divider}$	AP$_{cross}$	AP$_{boundary}$	mAP	mAP$_{day}$	mAP$_{night}$	mAP$_{sunny}$	mAP$_{cloudy}$	mAP$_{rainy}$
IPM(B) [24]	Eff-B0	30	10.7	4.7	11.7	9.0	-	-	-	-	-
IPM(CB) [24]	Eff-B0	30	24.0	7.3	27.8	19.7	-	-	-	-	-
LSS [43]	Eff-B0	30	22.9	5.1	24.2	17.4	11.5$^\flat$	13.3$^\flat$	12.0$^\flat$	12.8$^\flat$	9.5$^\flat$
VPN [40]	Eff-B0	30	22.1	5.2	25.3	17.5	17.9$^\flat$	17.5$^\flat$	18.2$^\flat$	18.5$^\flat$	15.9$^\flat$
HDMapNet [24]	Eff-B0	30	28.3	7.1	32.6	22.7	21.4$^\flat$	17.4$^\flat$	21.5$^\flat$	23.4$^\flat$	18.9$^\flat$
BeMapNet	Eff-B0	30	46.7	37.4	38.0	40.7	41.3	28.6	41.4	44.9	36.2
BeMapNet	Res-50	30	46.9	39.0	37.8	41.3	41.9	30.4	42.6	43.6	37.4
BeMapNet	Swin-T	30	**49.1**	**42.2**	**39.9**	**43.7**	**44.3**	**34.4**	**44.8**	**47.9**	**37.8**
BeMapNet	Res-50	110	52.7	44.5	44.2	47.1	47.7	39.2	48.7	50.3	41.4
BeMapNet	Swin-T	110	54.2	46.5	46.5	49.1	49.8	37.3	50.4	53.5	42.8
BeMapNet	Swin-B	110	55.3	47.0	49.4	50.5	51.2	38.0	52.2	54.8	43.6
BeMapNet	Res-50	30	62.3	57.7	59.4	59.8	60.5	46.9	62.5	61.9	53.0
BeMapNet	Swin-T	30	64.4	61.3	61.6	62.5	63.1	53.2	64.7	66.0	54.2
BeMapNet	Swin-B	110	69.0	64.4	69.7	67.7	68.3	54.6	70.3	71.6	58.0

Table 1. Comparisons with *SOTAs* on *NuScenes* under thresholds of $[0.2, 0.5, 1.0]m$ and $[0.5, 1.0, 1.5]m$, where the results of latter easier evaluation protocol is marked by *Green* shade. The *Blue* shade contains the results that used in all ablation studies for a fair comparison. Note $^\flat$ and - indicate the results are re-implemented by us with public code and not available respectively.

Implementation Details. We employ EfficientNet-B0 [49], ResNet-50 [14] and SwinTR [31] as backbones, which are all initialized by ImageNet [21] pretraining. The following semantic *BEV* decoder stacks 2 transformer encoder layers and 4 decoder layers with 64×32 *BEV* queries, and the instance Bézier decoder stacks 6 mask-transformer decoder layers with 60 queries, where $20, 25, 15$ for $lane\text{-}divider$, $ped\text{-}crossing$ and $road\text{-}boundary$ respectively. The shape of input image is resized to 896×512 and the mini-batch size is set to 1 per GPU. We train our model with 8 GPUs for $30/110$ epochs and adopt multi-step schedule with milestone $[0.7, 0.9]$ and $\gamma = \frac{1}{3}$. The AdamW [33] optimizer is employed with a weight decay of $1e^{-4}$ and a learning rate of $2e^{-4}$, which is multiplied by 0.1 for backbone. As for hyper-parameters of loss weight, we set $\lambda_s, \lambda_z, \lambda_p, \lambda_c, \lambda_r$ to $1, 5, 5, 10, 1$ respectively and the dilated width ω in $\mathcal{L}_{region}$ to 5. The deployments of piecewise Bézier curve $\langle \boldsymbol{k}, \boldsymbol{n} \rangle$ for $lane\text{-}divider$, $ped\text{-}crossing$ and $road\text{-}boundary$ are set to $\langle \mathbf{3}, \mathbf{2} \rangle, \langle \mathbf{1}, \mathbf{1} \rangle, \langle \mathbf{7}, \mathbf{3} \rangle$. Note m in Algorithm 1 is set to 100.

4.2. Comparisons with State-of-the-art Methods

We present the overall evaluation results on *NuScenes* in Table 1, which shows that our ***BeMapNet*** is significantly superior to the existing *SOTA* approaches by a large margin (up to $\mathbf{18.0}$) under the same setting of EfficientNet-B0 and 30 epochs, indicating the effectiveness of our approach. In addition, after replacing the backbone with more common ResNet-50 and more popular SwinTiny, our model achieves a further improvement of 0.6 and 3.0 AP respectively. Next, considering the slow convergence of the Transformer-based model, we increase the training schedule to 110 epochs and gain at least another 5.4 improvements. Even in somewhat unconventional scenarios, such as night and rainy, our proposed approach still shows a great advantage, *i.e.* 13.2 and 16.9 AP are obtained respectively. Furthermore, another interesting observation is that regardless of which method is used, we find that the performance of sunny scene is always slightly lower than cloudy. We believe that strong illuminations and ground shades on sunny day might have a certain impact on the perception of map objects. As a supplement, the last three rows of Table 1 show the performance of our approach on a much simpler evaluation protocol.

Position Encoding	AP$_{divider}$	AP$_{cross}$	AP$_{boundary}$	mAP
Sine PE	42.8	34.1	34.7	37.2
Learned PE	40.2	32.2	33.8	35.4
IPM-PE	46.4	39.2	37.8	41.1
+ Exclusive *FC*	38.6	33.3	33.9	35.3
+ **Shared** *FC*	**49.1**	**42.2**	**39.9**	**43.7**

Table 2. Comparisons on different methods of position encoding.

4.3. Ablation Study

The Different Way of Position Encoding. Table 2 shows the performance of different *PE* in semantic *BEV* decoder for the map construction. Compared with the *sine/learned-based* method, our proposed *IPM-PE* module improves AP by $3.9/5.7$ respectively, which proves the multi-perspective *PE* guided by the intrinsic and extrinsic parameters is very effective. Moreover, through the results in the 3-rd and 5-th row, we find that *IPM-PE* with shared *FC* layer alignment gains an further improvement of 2.6 points. Yet, in order to rule out the possibility of that the attached *FC* layer brings an increment, we then feed the position embeddings from *IPM-PE* into two **exclusive** *FC*s and the 4-th row results exclude the influence of *FC*. These comparisons indicate the shared align layer is the exactly essential part to mitigate the performance decline brought by the unreasonable assumption of *IPM*, which validates our conjecture in Sec.3.2.2.

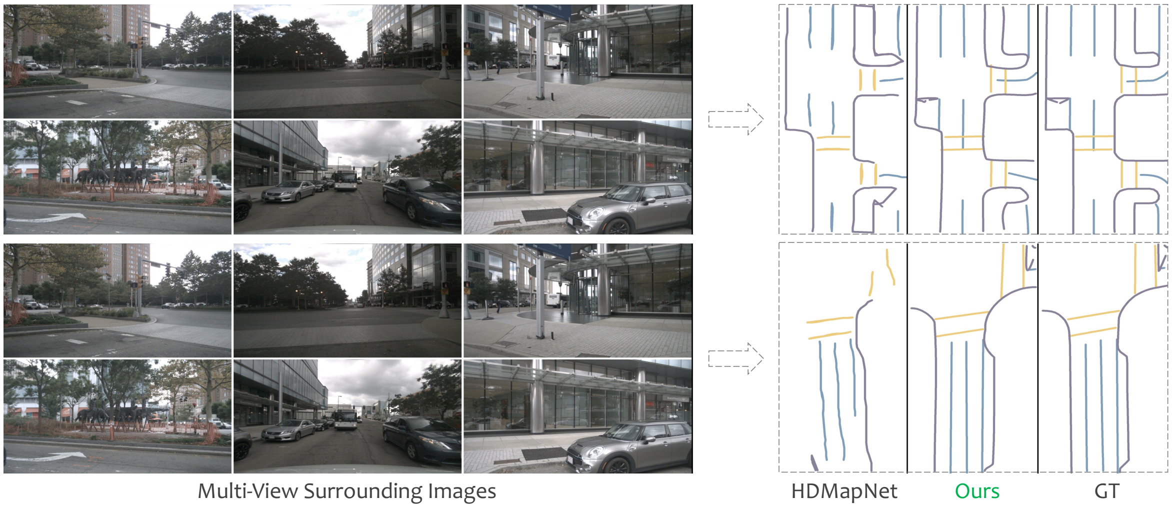

Figure 5. The qualitative comparison results of *HDMapNet* [24], ***BeMapNet*** and *GroundTruth* [2] under different scenarios.

Effectiveness of *PCR-Loss*. The *PCR-Loss* mainly consists of three parts, *i.e*. point, curve and region. Table 3 shows detailed ablation results of the role of each part. Specifically, with adopting only one supervision in rows $1 \sim 3$, the performance trend is roughly: *curve* > *point* > *region*. Among them, it is worth noting that the only *region-based* supervision shows very poor performance, which is understandable because the generation of region mask relies on a collection of relatively accurate points. This conjecture is also verified in rows $4 \sim 5$ of the Table 3. Furthermore, when integrating multiple loss supervision, the final performance is consistently improved compared to the single one, and three parts used together achieves the optimal AP results of 43.7.

Point	Curve	Region	$\text{AP}_{divider}$	AP_{ped}	$\text{AP}_{boundary}$	mAP
✓	✗	✗	39.1	33.3	23.7	32.0
✗	✓	✗	47.2	35.1	35.9	39.4
✗	✗	✓	3.2	2.6	0.5	2.1
✓	✗	✓	40.7	39.4	16.7	32.9
✗	✓	✓	48.2	38.5	39.7	42.1
✓	✓	✗	46.5	38.5	36.2	40.4
✓	✓	✓	**49.1**	**42.2**	**39.9**	**43.7**

Table 3. The effectiveness of different modules in *PCR-Loss*.

Effectiveness of Multi-task Auxiliary Loss. Without any auxiliary loss, the average AP of our proposed ***BeMapNet*** is only 20.0. Yet, after adopting another semantic/instance supervision, the result is improved to 34.5/38.4 respectively and goes further to 43.7 with employing both, which shows the auxiliary tasks for producing high-quality semantic *BEV* features and instance descriptors are very important.

Semantic	Instance	$\text{AP}_{divider}$	AP_{ped}	$\text{AP}_{boundary}$	mAP
✗	✗	21.5	17.8	20.7	20.0
✓	✗	39.3	31.5	32.8	34.5
✗	✓	44.3	34.6	36.5	38.4
✓	✓	**49.1**	**42.2**	**39.9**	**43.7**

Table 4. The effectiveness of different modules in *AUX-Loss*.

Comparison with Polyline Vectorization. An amusing fact is that a Bézier curve with $n = 1$ is simply a straight line between two control points. By setting the degree of all elements to 1, our framework naturally degenerates into the *polyline-based* method. Note we further set $\boldsymbol{k}$ to $9, 1, 29$ for $lane\text{-}divider$, $ped\text{-}crossing$ and $road\text{-}boundary$ respectively. The comparisons in Table 5 shows the Bézier-based vectorization approach achieves better 4.5 AP. Interestingly, as the complexity of map element shape increases (usually $ped\text{-}crossing < lane\text{-}divider < road\text{-}boundary$), we find that the corresponding *AP* improvement increases as well.

Vectorization	$\text{AP}_{divider}$	AP_{ped}	$\text{AP}_{boundary}$	mAP
Polyline	45.0	39.1	33.4	39.2
Bézier	**49.1** (+4.1)	**42.2** (+3.1)	**39.9** (+6.5)	**43.7**(+4.5)

Table 5. Comparison with the different type of vectorization.

Qualitative Analysis. We show the qualitative comparisons with *SOTAs* in Fig. 5. Besides avoiding complex vectorized post-processing, the proposed *BeMapNet* constructs various and changeful *HD*-elements more compactly and robust. Note that more extensive visualizations under different conditions are further provided in our supplementary material.

5. Conclusion

Vectorized *HD-map* online construction focuses on the perception of centimeter-level environmental information. Starting from the conventional parameterization-based methods, this paper presents an end-to-end postprocessing-free architecture, namely ***BeMapNet***, with leveraging unified piecewise Bézier curve for various and changeful map elements. By introducing three well-designed modules, *i.e. IPM-PE Align Module*, *Piecewise Bézier Head* and *Point-Curve-Region Loss*, the overall framework is concise and elegant, which reaches state-of-the-art performance and provides a new perspective for future *HD-map* research.

References

[1] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pages 586–606. Spie, 1992. 2

[2] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. 6, 8

[3] Yigit Baran Can, Alexander Liniger, Danda Pani Paudel, and Luc Van Gool. Structured bird's-eye-view traffic scene understanding from onboard images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 15661–15670, 2021. 2, 3

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 2

[5] Li Chen, Chonghao Sima, Yang Li, Zehan Zheng, Jiajie Xu, Xiangwei Geng, Hongyang Li, Conghui He, Jianping Shi, Yu Qiao, et al. Persformer: 3d lane detection via perspective transformer and the openlane benchmark. *arXiv preprint arXiv:2203.11089*, 2022. 3

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2017. 2

[7] Yinpeng Chen, Xiyang Dai, Mengchen Liu, Dongdong Chen, Lu Yuan, and Zicheng Liu. Dynamic convolution: Attention over convolution kernels. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11030–11039, 2020.

[8] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1290–1299, 2022. 4

[9] Kashyap Chitta, Aditya Prakash, and Andreas Geiger. Neat: Neural attention fields for end-to-end autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 15793–15803, 2021. 3

[10] Nemanja Djuric, Vladan Radosavljevic, Henggang Cui, Thi Nguyen, Fang-Chieh Chou, Tsung-Han Lin, Nitin Singh, and Jeff Schneider. Uncertainty-aware short-term motion prediction of traffic actors for autonomous driving. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 2095–2104, 2020. 3

[11] Renaud Dubé, Abel Gawel, Hannes Sommer, Juan Nieto, Roland Siegwart, and Cesar Cadena. An online multi-robot slam system for 3d lidars. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1004–1011. IEEE, 2017. 2

[12] Zhengyang Feng, Shaohua Guo, Xin Tan, Ke Xu, Min Wang, and Lizhuang Ma. Rethinking efficient lane detection via curve modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 17062–17070, 2022. 1, 2

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017. 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 7

[15] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 15273–15282, October 2021. 2

[16] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. Fiery: Future instance prediction in bird's-eye view from surround monocular cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 15273–15282, 2021. 3

[17] Junjie Huang, Guan Huang, Zheng Zhu, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 1, 3

[18] Jialin Jiao. Machine learning assisted high-definition map creation. In *IEEE Annual Computer Software and Applications Conference*, volume 1, pages 367–373. IEEE, 2018. 2

[19] Xiaojie Jin, Huaxin Xiao, Xiaohui Shen, Jimei Yang, Zhe Lin, Yunpeng Chen, Zequn Jie, Jiashi Feng, and Shuicheng Yan. Predicting scene parsing and motion dynamics in the future. *Advances in Neural Information Processing Systems*, 30, 2017. 3

[20] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9404–9413, 2019. 2

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 7

[22] Gim Hee Lee, Friedrich Fraundorfer, and Marc Pollefeys. Robust pose-graph loop-closures with expectation-maximization. In *IEEE International Conference on Intelligent Robots and Systems*, pages 556–563. IEEE, 2013. 2

[23] Namhoon Lee, Wongun Choi, Paul Vernaza, Christopher B Choy, Philip HS Torr, and Manmohan Chandraker. Desire: Distant future prediction in dynamic scenes with interacting agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 336–345, 2017. 3

[24] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *International Conference on Robotics and Automation*, pages 4628–4634. IEEE, 2022. 1, 2, 3, 6, 7, 8

[25] Xiang Li, Jun Li, Xiaolin Hu, and Jian Yang. Line-cnn: End-to-end traffic line detection with line proposal unit. *IEEE Transactions on Intelligent Transportation Systems*, 21(1):248–258, 2019. 1, 2

[26] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 2, 3

[27] Rong Liu, Jinling Wang, and Bingqi Zhang. High definition map for automated driving: Overview and analysis. *The Journal of Navigation*, 73(2):324–341, 2020. 2

[28] Ruijin Liu, Zejian Yuan, Tie Liu, and Zhiliang Xiong. End-to-end lane shape prediction with transformers. In *Proceedings of the IEEE Winter Conference on Applications of Computer Vision*, pages 3694–3702, 2021. 1, 2

[29] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9809–9818, 2020. 2

[30] Yingfei Liu, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petr: Position embedding transformation for multi-view 3d object detection. *European Conference on Computer Vision*, 2022. 3

[31] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10012–10022, 2021. 7

[32] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015. 2

[33] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2018. 7

[34] Gellert Mattyus, Shenlong Wang, Sanja Fidler, and Raquel Urtasun. Enhancing road maps by parsing aerial images around the world. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1689–1697, 2015. 2

[35] Gellért Máttyus, Shenlong Wang, Sanja Fidler, and Raquel Urtasun. Hd maps: Fine-grained road segmentation by parsing ground and aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3611–3619, 2016. 2

[36] Ellon Mendes, Pierrick Koch, and Simon Lacroix. Icp-based pose-graph slam. In *IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 195–200. IEEE, 2016. 2

[37] Lu Mi, Hang Zhao, Charlie Nash, Xiaohan Jin, Jiyang Gao, Chen Sun, Cordelia Schmid, Nir Shavit, Yuning Chai, and Dragomir Anguelov. Hdmapgen: A hierarchical graph generative model of high definition maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4227–4236, 2021. 2

[38] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *International Conference on 3D Vision*, pages 565–571. IEEE, 2016. 6

[39] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In *IEEE Intelligent Vehicles Symposium*, pages 286–291. IEEE, 2018. 1

[40] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *IEEE Robotics and Automation Letters*, 5(3):4867–4873, 2020. 1, 2, 3, 7

[41] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 1

[42] Lang Peng, Zhirong Chen, Zhangjie Fu, Pengpeng Liang, and Erkang Cheng. Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs. *IEEE Winter Conference on Applications of Computer Vision*, 2023. 1, 2, 3

[43] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *European Conference on Computer Vision*, pages 194–210. Springer, 2020. 1, 2, 3, 7

[44] Lennart Reiher, Bastian Lampe, and Lutz Eckstein. A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view. In *IEEE International Conference on Intelligent Transportation Systems*, pages 1–7. IEEE, 2020. 1, 3

[45] Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IEEE International Conference on Intelligent Robots and Systems*, pages 4758–4765. IEEE, 2018. 1

[46] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *IEEE International Conference on Intelligent Robots and Systems*, pages 5135–5142. IEEE, 2020. 1, 2

[47] Lucas Tabelini, Rodrigo Berriel, Thiago M. Paixao, Claudine Badue, Alberto F. De Souza, and Thiago Oliveira-Santos. Keep your eyes on the lane: Real-time attention-guided lane detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 294–302, June 2021. 1, 2

[48] Lucas Tabelini, Rodrigo Berriel, Thiago M Paixao, Claudine Badue, Alberto F De Souza, and Thiago Oliveira-Santos. Polylanenet: Lane estimation via deep polynomial regression. In *International Conference on Pattern Recognition*, pages 6150–6156. IEEE, 2021. 2

[49] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. 7

[50] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10781–10790, 2020. 4

[51] Wouter Van Gansbeke, Bert De Brabandere, Davy Neven, Marc Proesmans, and Luc Van Gool. End-to-end lane detection through differentiable least-squares fitting. In *Pro-*

ceedings of the IEEE International Conference on Computer Vision Workshops, pages 0–0, 2019. 2

[52] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017. 5

[53] Jacob Walker, Abhinav Gupta, and Martial Hebert. Patch to the future: Unsupervised visual prediction. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 3302–3309, 2014. 3

[54] Jinsheng Wang, Yinchao Ma, Shaofei Huang, Tianrui Hui, Fei Wang, Chen Qian, and Tianzhu Zhang. A keypoint-based global association network for lane detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1392–1401, 2022. 2

[55] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 913–922, 2021. 3

[56] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022. 3

[57] Bin Yang, Ming Liang, and Raquel Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *Conference on Robot Learning*, pages 146–155. PMLR, 2018. 2

[58] Sheng Yang, Xiaoling Zhu, Xing Nian, Lu Feng, Xiaozhi Qu, and Teng Ma. A robust pose graph approach for city scale lidar mapping. In *IEEE International Conference on Intelligent Robots and Systems*, pages 1175–1182. IEEE, 2018. 2

[59] Weixiang Yang, Qi Li, Wenxi Liu, Yuanlong Yu, Yuexin Ma, Shengfeng He, and Jia Pan. Projecting your view attentively: Monocular road scene layout estimation via cross-view transformation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 15536–15545, 2021. 2

[60] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, pages 1–9. Berkeley, CA, 2014. 1

[61] Yu Zhang and Qiang Yang. A survey on multi-task learning. *IEEE Transactions on Knowledge and Data Engineering*, 2021. 6

[62] Tu Zheng, Hao Fang, Yi Zhang, Wenjian Tang, Zheng Yang, Haifeng Liu, and Deng Cai. Resa: Recurrent feature-shift aggregator for lane detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 3547–3554, 2021. 1

[63] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 13760–13769, 2022. 2