ScalableMap: Scalable Map Learning for Online Long-Range Vectorized HD Map Construction

Anonymous Author(s)
Affiliation
Address
email

Abstract: We propose a novel end-to-end pipeline for online long-range vectorized high-definition (HD) map construction using on-board camera sensors. The vectorized representation of HD maps, employing polylines and polygons to represent map elements, is widely used by downstream tasks. However, previous schemes designed with reference to dynamic object detection overlook the structural constraints within linear map elements, resulting in performance degradation in long-range scenarios. In this paper, we exploit the properties of map elements to improve the performance of map construction. We extract more accurate bird's eye view (BEV) features guided by their linear structure, and then propose a hierarchical sparse map representation to further leverage the scalability of vectorized map elements, and design a progressive decoding mechanism and a supervision strategy based on this representation. Our approach, ScalableMap, demonstrates superior performance on the nuScenes dataset, especially in long-range scenarios, surpassing previous state-of-the-art model by 6.5 mAP while achieving 18.3 FPS.

Keywords: Map Construction, Multi-view Perception, Long-range Perception

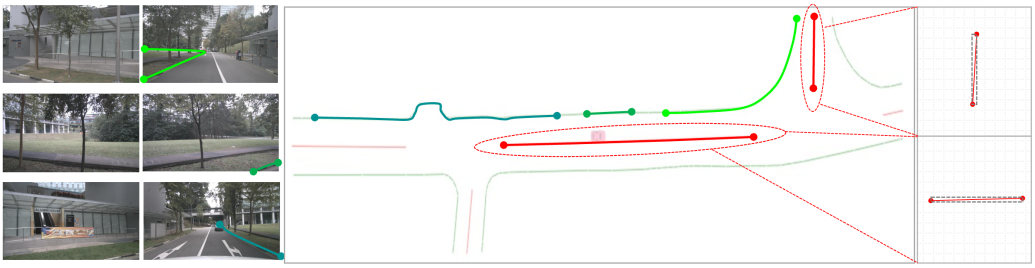
1 Introduction

To ensure the safety of autonomous vehicles on the road, downstream tasks such as trajectory prediction and motion planning typically rely on high-definition (HD) maps as prior information [1, 2], which provide centimeter-level location information for map elements. However, the production of such HD maps is generally carried out offline, involving complex processes known for their high labor and economic costs, making it difficult to construct maps that cover a wide area [3, 4].

Recent researches aim to construct local maps in real-time using on-board sensors. More studies show the superiority of schemes based on bird's-eye view (BEV) representation for unifying data from various sensors. Early attempts [5, 6, 7, 8, 9] regard map construction as a semantic segmentation task, using convolutional neural networks (CNN) to obtain occupancy grid map. However, these schemes can only generate rasterized maps, which lack instance and structure information about map elements and are therefore difficult to apply directly to downstream tasks [10].

HDMapNet [11] uses time-consuming heuristic post-processing algorithms to generate vectorized maps. More recent approaches [12, 13] focus on constructing end-to-end networks similar to dynamic object detection schemes, treating map elements as ordered sets of vertices. However, as shown in Figure 1, the properties of map elements are different from those of dynamic objects. Map elements are typically linear and often parallel to axes [14], which makes it difficult to define bounding boxes. Moreover, in dense vehicle scenarios, the limited visibility of vertices with map elements in the image space hinders accurate map shape inference solely based on heatmaps. While a recent approach [15] proposes hierarchical query embeddings to better describe the arbitrary shape of an element by modeling each vertex as a query, it requires dense points to ensure the shape of elements and need to predict a large number of vertices simultaneously without structural guidance. This

Submitted to the 7th Conference on Robot Learning (CoRL 2023). Do not distribute.



(a) Surround views. (b) Ground truth of BEV. (c) Bounding box.

Figure 1: An example from nuScenes illustrating the differences in properties between map elements and dynamic objects. The road boundary shown in green is a linear element that spans a long range and may appear in multiple views, leading to incomplete and obscured features. The lane dividers shown in red are nearly parallel to the coordinate axes, limiting the bounding box-based solution.

poses challenges to the convergence speed and performance, particularly in long-range scenarios. Therefore, there is still a need for an approach that can effectively capture the structural constraints within map elements to achieve high accuracy in long-range HD map construction tasks.

In this paper, we aim to exploit the structural properties of vectorized map elements to address the challenges of accurately detecting map elements at longer ranges. First, we extract position-aware BEV features and instance-aware BEV features via two branches respectively and fuse them under the guidance of linear structure to get hybrid BEV features. Next, we propose a hierarchical sparse map representation (HSMR) to abstract map elements in a sparse but accurate manner. Integrating this representation with cascaded decoding layers proposed by DETR [16], we design a progressive decoder to enhance the constraints of structured information by exploiting the scalability of vectorized map elements and a progressive supervision strategy to improve the accuracy of inference. Our scheme, ScalableMap, dynamically increases the sampling density of map to get inference results at various scales, allowing us to obtain more accurate map information faster.

Contributions. Our contributions are summarized as follows: (i) We propose ScalableMap, a first end-to-end long-range vectorized map construction pipeline. We exploit the structural properties of map elements to extract more accurate BEV features, propose a HSMR based on the scalable vectorized elements, and design a progressive decoder and supervision strategy accordingly. All of these result in superior long-range map perception. (ii) We evaluate the performance of ScalableMap on the nuScenes dataset [17] through extensive experiments. Our proposed approach achieves state-of-the-art results in long-range HD map learning, surpassing existing multimodal methods by 6.5 mAP while achieving 18.3 FPS.

2 Related work

Lane Detection The lane detection task has been a popular research topic for many years. Early approaches to these tasks [18, 5] usually rely on segmentation schemes that require complex post-processing to obtain the final result. In order to obtain structured information, some schemes [19, 20] aim to find a unified representation of curves, while others [21, 22, 23, 24] utilize anchor-based schemes to abstract map elements with open shapes. Compared with the above solutions, our thinking is closer to HRAN [4], which directly outputs structured polylines. However, it relies on a recurrent network that is known to be computationally inefficient. Our ScalableMap is capable of handling real map elements with complex geometric structures, while the previously mentioned methods can only handle a single type or regular shape.

Boundary Extraction Boundary Extraction aims to predict polygon boundaries for object instances on images. Polygon-RNN [25, 26] adopts recurrent structure to trace each boundary sequentially, which is not suitable for scenarios with real-time requirements. Some works [22, 27, 28]

achieve good results in boundary extraction, but they are generally designed for polygons in image space and are not suitable for map construction tasks. The closest to our proposed scheme is BoundaryFormer [29], which uses queries to predict vertices of polygons to obtain vectorized polygon boundaries. However, the differentiable loss it defines for closed-shape elements in image space is not suitable for map element that is dominated by open-shape linear elements, as they have less concentrated features compared to dynamic objects.

Vectorized HD Map Construction Recent work tries to infer vectorized HD maps directly from on-board sensor data. HDMapNet [11] generates vectorized maps using a time-consuming heuristic post-processing method, while VectorMapNet [12] proposes a two-stage framework with an end-to-end pipeline using a slow auto-regressive decoder to recurrently predict vertices. InstaGraM [13] proposes a graph modeling approach based on vertex and edge heatmaps to reason about instance-vertex relations, which may be difficult to infer some vertices of a map element appeared in multiple views. Given the challenge of dealing with arbitrary shapes and varying numbers of vertices in elements, MapTR [15] tackles this by employing a fixed number of interpolations to obtain a uniform representation. But MapTR's hierarchical query design primarily focuses on the structural association of elements during the initialization phase, resulting in slow convergence and deteriorating performance as the perception range increases. Only SuperFusion [30] is a relevant work for long-range vectorized HD map construction, which also uses post-processing to obtain vectorized results. Our model is the first end-to-end scheme that utilizes the structural properties of map elements throughout the entire process to construct long-range vectorized maps.

3 Methodology

3.1 Overview

Given a set of surround-view images $\{I_1, ..., I_k\}$ captured from k on-board cameras, the goal of ScalableMap is to predict M local map elements $\{L^{(j)}; j = 1, ..., M\}$ within a certain range in real-time, including lane dividers, road boundaries, and pedestrian crossings. Each map element is represented by a sparse set of ordered vertices, which can be described as $L^{(j)} = \{(x_0, y_0, z_0), ..., (x_{m_j}, y_{m_j}, z_{m_j})\}$, where m_j is the number of vertices of element $L^{(j)}$ and (x, y, z) is the coordinates of each vertex in a unified vehicle coordinate system.

The architecture of ScalableMap is illustrated in Figure 2. We build the model with three components to construct long-range vectorized HD maps: (1) *structure-guided hybrid BEV feature extractor*: transforming camera sensor data into BEV features with structure-guided fusion (Section 3.2) ; (2) *progressive decoder*: layer by layer map element decoding based on the proposed HSMR (Section 3.3) ; (3) *progressive supervision*: bipartite matching and training for HSMR (Section 3.4).

3.2 BEV Feature Extractor

The ill-posed nature of 2D-3D transformation is exacerbated by the elongated and linear characteristics of map elements, leading to feature misalignment and discontinuity. To obtain hybrid BEV features, we utilize one branch for extracting position-aware BEV features and another branch for extracting instance-aware BEV features. These branches are then fused together, guided by the structural properties of map elements.

Perspective View Converter. We start by extracting image features through ResNet. Method proposed by BEVFormer [31] is adopted to obtain position-aware BEV features F^p_{bev}, which utilizes deformable attention [32] to enable spatial interaction between BEV queries and corresponding image features based on a predefined 3D grid and calibration parameters. Additionally, we use several multi-layer perception (MLP) layers to obtain instance-aware BEV features F^i_{bev} since they are effective at preserving continuous features in image space [33]. K image features are individually converted to their respective top-views using k MLPs. To further improve feature continuity across views, we use a linear layer to transform top-view features into a unified BEV feature.

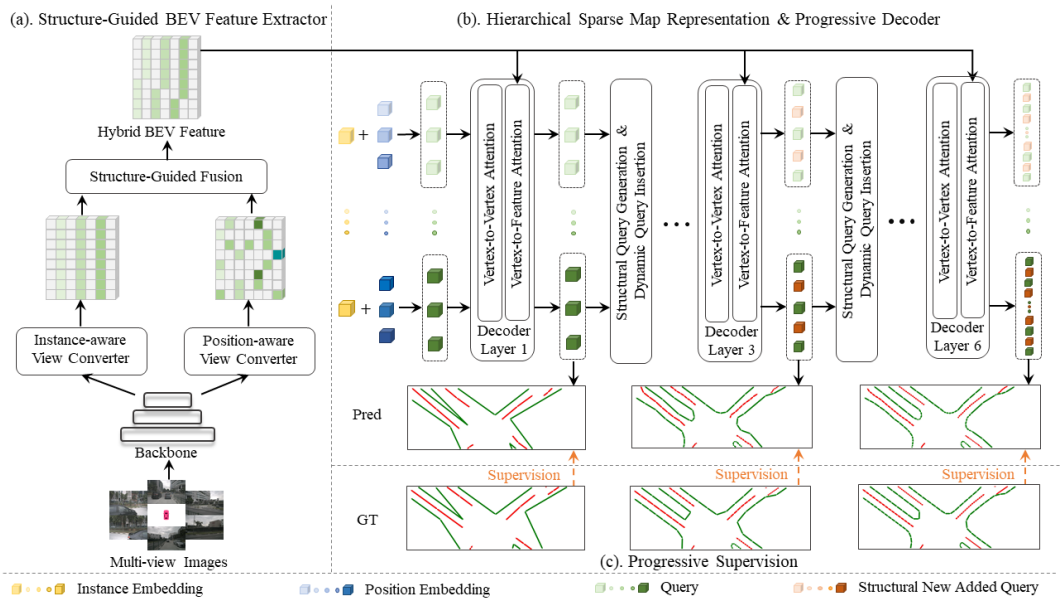


Figure 2: Overview of ScalableMap. (a) Structure-guided hybrid BEV feature extractor. (b) Hierarchical sparse map representation & Progressive decoder. (c) Progressive supervision.

Structure-Guided Feature Fusion. To obtain a stronger feature for more accurate map construction, we employ a mutual correction approach, leveraging the information from two features: F^p_{bev}, which contains reasonably accurate position information for some vertices of map elements, and F^i_{bev}, which provides relatively comprehensive shape information for map elements. We directly add these two features to get the modified $F^{i'}_{bev}$, and add a segmentation head to $F^{i'}_{bev}$ to guide it to focus on the drivable area to enhance the performance of this parameter-free conversion. Following that, we concatenate F^p_{bev} with the corrected $F^{i'}_{bev}$ and utilize a convolutional layer to fuse their information. The latter process corrects the misaligned features in F^p_{bev}, resulting in a hybrid BEV feature that exhibits enhanced information richness and accuracy.

3.3 Progressive Decoder

The varied shapes of vectorized map elements present challenges for conventional abstraction schemes like bounding box-based and anchor-based approaches. To address this, we introduce a HSMR as the core idea of our approach. HSMR provides a sparse and unified representation that accurately describes the actual shape of elements while supporting fast inference. Building upon this, we design a progressive decoder inspired by the DETR paradigm. Moreover, we incorporate a module that generates structural queries first and then dynamically inserts queries, acting as a vital bridge to connect maps of different densities.

Hierarchical Sparse Map Representation. Polyline representations of map elements are typically obtained by sampling points where the curvature exceeds a threshold, thus resulting in varying numbers of vertices for each element. We define the number of vertices forming each element as the map density to ensure a consistent representation. Based on this density, we employ uniform point sampling for elements with an excessive number of vertices, while for elements with fewer vertices than the desired density, we perform point subsampling based on distances between the original vertices. This approach allows us to obtain representations of the same element at arbitrary densities. By combining the iterative optimization idea of DETR paradigm with the dynamically adjustable density of the vectorized map, we hierarchically utilize a low-density map as an abstract representation of the high-density map. The low-density map captures map element shapes adequately while being sufficiently sparse. We show the performance of this representation is Section 4.

Decoder Layers. We define query $q_{n,m}$ responsible for the m-th vertex of the n-th element. Leveraging the hierarchical sparse representation of map elements, a small number of queries are initially generated to capture the approximate shape of each map element. Each query is formed by adding an instance embedding q_n^{ins} and a position embedding $q_{n,m}^{pos}$. Our progressive map element decoder is composed of multiple decoder layers, each containing two types of attention mechanisms. These attention mechanisms facilitate information exchange among vertices, and enable interaction between each vertex and its corresponding BEV feature. The exchange between vertices is implemented using multi-head self-attention [34], while the other is implemented using deformable attention [32].

Structural Query Generation and Dynamic Query Insertion. To connect layers that handle different densities, we exploit the positional constraints among adjacent vertices within the same element to augment the map density. We introduce new queries by taking the mean value of two adjacent queries that share an edge, and dynamically insert new queries between these two queries. Rather than employing traditional methods that initialize a large number of queries simultaneously and update them iteratively, we adopt a strategy of initializing each element with only a limited number of queries and gradually increasing the map density layer by layer. This enables the module to focus on the original sparse instance features and leverage the structural characteristics of vectorized map elements, ensuring robust long-range perceptual capabilities.

3.4 Progressive Supervision

During training, ScalableMap infers N map elements $\{\hat{L}^i\}_{i=1}^N$ each layer, and N is set to be larger than the typical number of elements in a scene. Assume there are M targets $\{L^i\}_{i=1}^M$, which is padded with $\emptyset$ to form a set of size N. Following [16, 35], bipartite matching is employed to search for a permutation $\sigma \in \Sigma$ with lowest cost. Σ includes the equivalent permutation for each element, as multiple vertex orders can represent the actual shape of an element in map construction task [15]:

$$\sigma^* = argmin_{\sigma \in \Sigma} \sum_{i=1}^{N} [-\mathbb{1}_{\{c_i \neq \emptyset\}} \hat{p}_{\sigma(i)}(c_i) + \mathbb{1}_{\{c_i \neq \emptyset\}} \mathcal{L}_{match}(\hat{L}_{\sigma(i)}, L_i)] \tag{1}$$

where $\hat{p}_{\sigma(i)}(c_i)$ is the probability of class c_i for the prediction with index $\sigma(i)$ and $\mathcal{L}_{match}$ is a pair-wise polyline matching cost between prediction $\hat{L}_{\sigma(i)}$ and ground truth L_i. We use Hungarian algorithm [36] to find the optimal assignment σ^*. We employ focal loss to supervise the element category and drivable area, and additional loss terms are incorporated in the following loss function:

$$\mathcal{L}_{polyline} = \lambda_v \mathcal{L}_{vertex} + \mathcal{L}_{edge} \tag{2}$$

Vertex Loss. Considering HSMR involves subsampling process, we differentiate the supervision between original vertices and newly added vertices. For each of predicted original vertex $\hat{v}_{\sigma*(i),j}$ assigned to vertex $v_{i,j}$, we employ L1 distance supervision to ensure prediction accuracy. With N_v^l standing for the number of original vertices in layer l, vertex loss of each element is formulated as:

$$\mathcal{L}_{vertex} = \sum_{i=1}^{N} \mathbb{1}_{\{c_i \neq \emptyset\}} \sum_{j=0}^{N_v^l - 1} \|\hat{v}_{\sigma^*(i),j} - v_{i,j}\|_1 \tag{3}$$

Edge Loss. We use edge loss to supervise edge shape, which includes distance from newly added vertices $\{\hat{v}_{\sigma(i),j,k}\}_{k=1}^{N_v^j - 1}$ corresponding to the current edge $e_{i,j}$, edge slope, and angle formed by adjacent edges. The distance component is supervised with L1 loss, while the slope and angle components are supervised with cosine similarity. The edge loss of each element is formulated as:

$$\mathcal{L}_e = \sum_{i=1}^{N} \mathbb{1}_{\{c_i \neq \emptyset\}} \{ \sum_{j=0}^{N_v^l - 1} [\lambda_p \sum_{k=0}^{N_{v,j} - 1} d(\hat{v}_{\sigma^*(i),j,k}, e_{i,j}) + \lambda_s c(\hat{e}_{\sigma^*(i),j}, e_{i,j})] + \lambda_a \sum_{j=0}^{N_v - 2} c(\hat{a}_{\sigma^*(i),j}, a_{i,j}) \} \tag{4}$$

where $\hat{e}_{\sigma^*(i),j}$ is the edge formed by two adjacent vertices, $\hat{a}_{\sigma^*(i),j}$ is the angle formed by two adjacent edges, $d(\hat{v}, e)$ denotes the distance from vertex v to edge e, and $c(\hat{a}, a)$ denotes the cosine similarity between two edges. $N_{v,j}$ is the number of added vertices corresponding to edge $e_{\sigma^*(i),j}$.

Table 1: Results on nuScenes validation dataset. C denotes camera only and C+L denotes camera-LiDAR fusion. Range represents the perceived range along the Y-axis. FPS of ScalableMap is measured on a single RTX 3090 GPU, with batch size 1 and GPU warm-up. The metrics of MapTR* are obtained by retraining the model while modifying only the perception range, following the official code and ensuring consistency with the claimed specifications. Metric values marked with † represents the AP value under a threshold of 1.0. Since SuperFusion only provides this metric, we conduct the same benchmark test for a fair comparison.

Method	Modality	Range	AP_{ped}	$AP_{divider}$	$AP_{boundary}$	mAP	FPS
HDMapNet	C	[-30.0, 30.0]	14.4	21.7	33.0	23.0	0.8
HDMapNet	C+L	[-30.0, 30.0]	16.3	29.6	46.7	31.0	0.5
VectorMapNet	C	[-30.0, 30.0]	36.1	47.3	39.3	40.9	2.9
VectorMapNet	C+L	[-30.0, 30.0]	37.6	50.5	47.5	45.2	-
InstaGraM	C	[-30.0, 30.0]	47.2	33.8	44.0	41.7	17.6
MapTR	C	[-30.0, 30.0]	56.2	59.8	60.1	58.7	11.2
ScalableMap	C	[-30.0, 30.0]	**57.3**	**60.9**	**63.8**	**60.6**	**18.3**
MapTR*	C	[-60.0, 60.0]	35.6	46.0	35.7	39.1	11.2
ScalableMap	C	[-60.0, 60.0]	**44.8**	**49.0**	**43.1**	**45.6**	**18.3**
SuperFusion	C+L	[0.0, 60.0]	22.3†	30.3†	**53.4**†	35.3†	-
ScalableMap	C	[0.0, 60.0]	**51.0**†	**55.1**†	48.4†	**51.5**†	**18.3**

4 Experiments

4.1 Experimental Settings

Dataset and Metrics. We evaluate ScalableMap on the nuScenes dataset, which consists of 1000 scenes. Each scene has a duration of approximately 20 seconds. The dataset provides a 360-degree field of view around an ego-car, captured by six cameras. Following previous works [12**?** , 15], we use the average precision(AP) metric to evaluate the performance, and chamfer distance to determine which positive matches the ground truth. We calculate the AP for the three categories, and for each category, AP is computed under several thresholds $\{0.5, 1.0, 1.5\}$.

Implementation Details. We train ScalableMap for 110 epochs on RTX 3090 GPUs with batch size 32. The perception range for regular range test is $[-15.0m, 15.0m]$ along the X-axis and $[-30.0m, 30.0m]$ along the Y-axis, and the perception range for long-range test is expanded to $[-60.0m, 60.0m]$ along the Y-axis. To unify the representation of vertices, we use the Ramer–Douglas–Peucker algorithm [37] to simplify the original polyline with a threshold of $0.05m$ before subsampling. For training, we set the loss scales λ_{cls} as 2.0, λ_v, λ_p as 5.0 and λ_s, λ_a as $5e^{-3}$ respectively. The progressive decoder is composed of six decoder layers.

4.2 Results.

Comparison with Baselines. We evaluate the performance of ScalableMap by comparing it with that of state-of-the-art methods on nuScenes validation test. As shown in Table 1, under camera modality, ScalableMap performs slightly better than MapTR, achieving 1.9 higher mAP and faster inference speed within the conventional perception range of $[-30.0m, 30.0m]$ along the Y-axis. When the same models are directly applied to $[-60.0m, 60.0m]$ scenario, ScalableMap achieves 45.6 mAP and 18.3 FPS, while MapTR's corresponding values are 39.1 and 11.2. It is noted that SuperFusion is the only method which publishes experiment results in this range. However, it is a fusion model of lidar and single-view camera. The mAP achieved by our approach is higher than that of SuperFusion by 16.2 under the same benchmark, demonstrating the superior performance even in a multi-view camera modality with near real-time inference speed. The results demonstrate that our scheme effectively meets the real-time requirements of online map construction tasks, delivering superior accuracy in both conventional perception range tests and long-range tests.

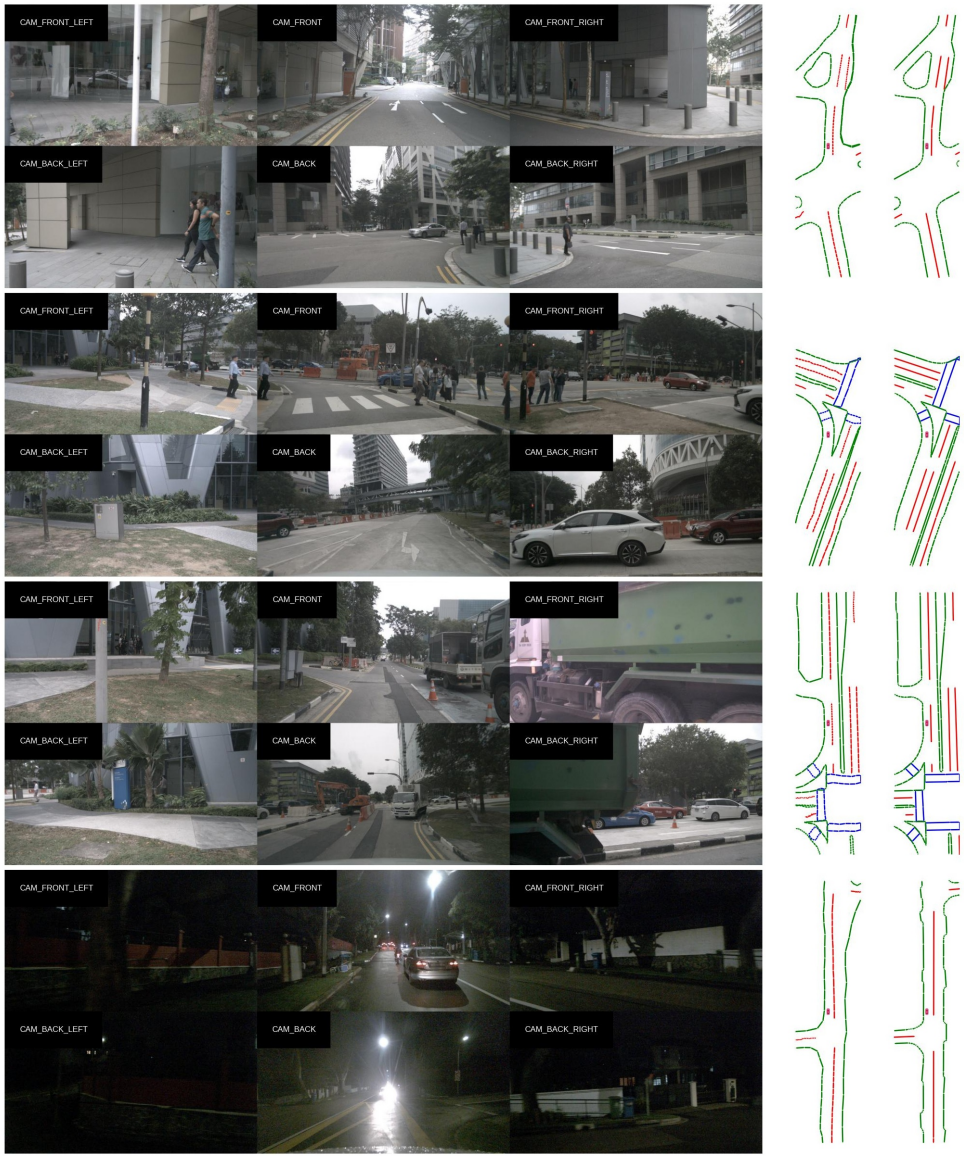
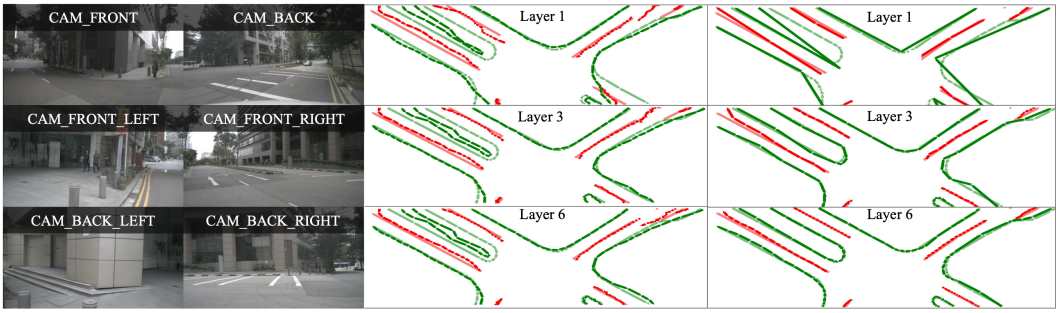


Figure 3: Visualization of qualitative results of ScalableMap in challenging scenes from nuScenes validation dataset. The left column is the surround views, the middle column is the inference results of the ScalableMap, the right column is corresponding ground truth. Green lines indicate boundaries, red lines indicate lane dividers, and blue lines indicate pedestrian crossings.

Qualitative Results Visualization. The visualization of qualitative results of ScalableMap on nuScenes validation dataset in long-range test is shown in Figure 3. Our model still performs well even in curved roads, intersections, congested roads, and night scenes. We further visualize three of six decoder layers of MapTR and ScalableMap in Figure 4. Our strategy demonstrates a faster ability to focus on the instance features, while the progressive iteration yields more precise shapes of elements.

4.3 Ablation Studies

We conduct ablation experiments on nuScenes validation set to verify the effectiveness of the components of the proposed method and different design. Settings of all experiments are kept the same as mentioned before.



Surround views MapTR* ScalableMap

Figure 4: Visualization of prediction from three decoder layers of MapTR* and ScalableMap. The perception range along the Y-axis is $[-60.0m, 60.0m]$. The light-colored lines on the image represent ground truth, while the dark-colored lines represent the inference results.

Table 2: Ablation study of three components proposed in this paper.

HSMR	SQG&DQI	SGFF	mAP
✓			39.7
✓	✓		42.6
✓	✓	✓	**45.6**

Table 3: Ablation study of the number of vertices in each decoder layer.

Number of vertices	mAP	FPS
2/3/5/9/9/9	43.6	**19.2**
3/5/9/17/17/17	**45.6**	18.3
4/7/13/25/25/25	44.2	17.6

Ablation of Proposed Components. In Table 2, we show the results of experiments about the effect of our proposed components. SGFF stands for structure-guided feature fusion. HSMR stands for hierarchical sparse map representation. SQG&DQI stands for structural query generation and dynamic query insertion. Results show that HSMR can achieve good results on long-range perception in a relatively sparse manner. SQG&DQI can further strengthen the structural information within the map elements, and SGFF module can also greatly improve the performance.

Ablation of Number of Vertices. Ablations of the effect of number of vertices forming each element on long-range perception in each decoder layer are presented in Table 3. The experimental results show that, based on our proposed HSMR, the model performance is quite stable with the number of vertices. We trade-off accuracy and speed to select the appropriate parameters.

5 Discussion

We propose ScalableMap, an innovative pipeline for constructing long-range vectorized HD maps. We exploit the inherent structure of map elements to extract accurate BEV features, propose the concept of HSMR based on the scalable vectorized maps, and design progressive decoder and supervision strategy accordingly to ensure fast convergence. Through these designs, our method effectively captures information over long distances. Experiment results on nuScenes dataset demonstrate compelling performance, particularly in long-range scenarios, thus affirming its real-time applicability and effectiveness in real-world environments.

Limitations. Our method relies solely on real-time camera sensor data, thus its performance depends on the visibility of the scenarios, which may be limited in situations like traffic congestion or extreme weather conditions. Additionally, accurate camera calibration parameters are assumed, which can pose a constraint in practical deployment. Future research can focus on reducing the reliance on calibration parameters by developing calibration-free approaches or incorporating online calibration methods. Exploring the integration of positional constraints among map elements or leveraging global coarse maps as prior knowledge may further enhance the robustness and accuracy.

References

[1] J. Gu, C. Sun, and H. Zhao. Densetnt: End-to-end trajectory prediction from dense goal sets. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15303–15312, 2021.

[2] W. Zeng, W. Luo, S. Suo, A. Sadat, B. Yang, S. Casas, and R. Urtasun. End-to-end interpretable neural motion planner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8660–8669, 2019.

[3] N. Homayounfar, W.-C. Ma, J. Liang, X. Wu, J. Fan, and R. Urtasun. Dagmapper: Learning to map by discovering lane topology. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2911–2920, 2019.

[4] N. Homayounfar, W.-C. Ma, S. K. Lakshmikanth, and R. Urtasun. Hierarchical recurrent attention networks for structured online maps. *Computer Vision and Pattern Recognition*, 2018.

[5] J. Philion and 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*, pages 194–210. Springer, 2020.

[6] T. Roddick and R. Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11138–11147, 2020.

[7] L. Reiher, B. Lampe, and L. 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 *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE, 2020.

[8] H. X. W. S. B. Z. J. M. Runsheng Xu, Zhengzhong Tu. Cobevt: Cooperative bird's eye view semantic segmentation with sparse transformers. In *Conference on Robot Learning (CoRL)*, 2022.

[9] B. Yang, M. Liang, and R. Urtasun. Hdnet: Exploiting hd maps for 3d object detection. In *Conference on Robot Learning*, pages 146–155. PMLR, 2018.

[10] J. Gao, C. Sun, H. Zhao, Y. Shen, D. Anguelov, C. Li, and C. Schmid. Vectornet: Encoding hd maps and agent dynamics from vectorized representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11525–11533, 2020.

[11] Q. Li, Y. Wang, Y. Wang, and H. Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *2022 International Conference on Robotics and Automation (ICRA)*, 2022.

[12] Y. Liu, Y. Wang, Y. Wang, and H. Zhao. Vectormapnet: End-to-end vectorized hd map learning. *arXiv preprint arXiv:2206.08920*, 2022.

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

[14] Y. Xu, W. Xu, D. Cheung, and Z. Tu. Line segment detection using transformers without edges. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4257–4266, 2021.

[15] B. Liao, S. Chen, X. Wang, T. Cheng, Q. Zhang, W. Liu, and C. Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. In *International Conference on Learning Representations*, 2023.

[16] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part I 16*, pages 213–229. Springer, 2020.

[17] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[18] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool. Towards end-to-end lane detection: an instance segmentation approach. In *2018 IEEE intelligent vehicles symposium (IV)*, pages 286–291. IEEE, 2018.

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

[20] W. Van Gansbeke, B. De Brabandere, D. Neven, M. Proesmans, and L. Van Gool. End-to-end lane detection through differentiable least-squares fitting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[21] L. Tabelini, R. Berriel, T. M. Paixao, C. Badue, A. F. De Souza, and 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*, pages 294–302, 2021.

[22] S. Zhang, S. Zhang, X. Pan, Z. Wang, and C. Lu. Curve-gcn: Enhancing curve-based road object detection with graph convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1529–1538, 2020.

[23] N. Garnett, R. Cohen, T. Pe'er, R. Lahav, and D. Levi. 3d-lanenet: end-to-end 3d multiple lane detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2921–2930, 2019.

[24] L. Chen, C. Sima, Y. Li, Z. Zheng, J. Xu, X. Geng, H. Li, C. He, J. Shi, Y. Qiao, et al. Persformer: 3d lane detection via perspective transformer and the openlane benchmark. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*, pages 550–567. Springer, 2022.

[25] L. Castrejon, K. Kundu, R. Urtasun, and S. Fidler. Annotating object instances with a polygon-rnn. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5230–5238, 2017.

[26] D. Acuna, H. Ling, A. Kar, and S. Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 859–868, 2018.

[27] J. Liang, N. Homayounfar, W.-C. Ma, Y. Xiong, R. Hu, and R. Urtasun. Polytransform: Deep polygon transformer for instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9131–9140, 2020.

[28] S. Zorzi, S. Bazrafkan, S. Habenschuss, and F. Fraundorfer. Polyworld: Polygonal building extraction with graph neural networks in satellite images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1848–1857, 2022.

[29] J. Lazarow, W. Xu, and Z. Tu. Instance segmentation with mask-supervised polygonal boundary transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4382–4391, 2022.

[30] H. Dong, X. Zhang, X. Jiang, J. Zhang, J. Xu, R. Ai, W. Gu, H. Lu, J. Kannala, and X. Chen. Superfusion: Multilevel lidar-camera fusion for long-range hd map generation and prediction. *arXiv preprint arXiv:2211.15656*, 2022.

[31] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Y. Qiao, and J. Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 1–18. Springer, 2022.

[32] X. Zhu, W. Su, L. Lu, B. Li, X. Wang, and J. Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[33] J. Zou, J. Xiao, Z. Zhu, J. Huang, G. Huang, D. Du, and X. Wang. Hft: Lifting perspective representations via hybrid feature transformation. *arXiv preprint arXiv:2204.05068*, 2022.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[35] Y. Wang, V. Guizilini, T. Zhang, Y. Wang, H. Zhao, , and J. M. Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *The Conference on Robot Learning (CoRL)*, 2021.

[36] H. W. Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

[37] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2):112–122, 1973.