StreamMapNet: Streaming Mapping Network for Vectorized Online HD Map Construction

Tianyuan Yuan[1] Yicheng Liu[1] Yue Wang[2] Yilun Wang[1] Hang Zhao[1*]

[1]Tsinghua University [2]University of Southern California

Abstract

High-Definition (HD) maps are essential for the safety of autonomous driving systems. While existing techniques employ camera images and onboard sensors to generate vectorized high-precision maps, they are constrained by their reliance on single-frame input. This approach limits their stability and performance in complex scenarios such as occlusions, largely due to the absence of temporal information. Moreover, their performance diminishes when applied to broader perception ranges. In this paper, we present StreamMapNet, a novel online mapping pipeline adept at long-sequence temporal modeling of videos. StreamMapNet employs multi-point attention and temporal information which empowers the construction of large-range local HD maps with high stability and further addresses the limitations of existing methods. Furthermore, we critically examine widely used online HD Map construction benchmark and datasets, Argoverse2 and nuScenes, revealing significant bias in the existing evaluation protocols. We propose to resplit the benchmarks according to geographical spans, promoting fair and precise evaluations. Experimental results validate that StreamMapNet significantly outperforms existing methods across all settings while maintaining an online inference speed of 14.2 *FPS. Our code is available at* https://github.com/yuantianyuan01/StreamMapNet.

1. Introduction

High-Definition (HD) maps, designed specifically for autonomous driving, are highly accurate maps that provide detailed and vectorized representations of map elements such as pedestrian crossings, lane dividers, and road boundaries. These maps are essential for self-driving vehicles as they contain rich semantic information about roads, enabling effective navigation. Traditionally, HD maps were constructed offline using SLAM-based methods (LOAM [28], LeGO-LOAM [22]), resulting in complex pipelines and high maintenance costs. However, these methods face scalability issues due to their heavy reliance on human labor for map annotation and updates. In recent years, deep-learning-based methods have emerged as a promising alternative, allowing for the online construction of vectorized HD maps around the ego-vehicle using onboard sensors. These online approaches offer cost savings by eliminating the need for mapping fleets and reducing human labor while maintaining the ability to adapt to new environments or potential map changes.

Early approaches to semantic map learning treated the task as a segmentation problem in Bird's-Eye-View (BEV) space (HDMapNet [9], Lift-Splat-Shoot [19], Roddick and Cipolla [21]). However, these methods generated rasterized maps that lacked the notion of instances, rendering them unsuitable for downstream tasks such as motion forecasting or motion planning (VectorNet [4], LaneGCN [11]) which require vectorized maps. More recently, approaches like VectorMapNet [15] and MapTR [12] achieved promising results in constructing end-to-end vectorized local HD maps using transformer [23] decoders inspired by DETR [2]. Nevertheless, these methods face two main challenges: *(1) Small perception range:* These methods are limited in constructing HD maps with a relatively small perception range of $60 \times 30\,m$, which is impractical for autonomous driving scenarios. When attempting to extend the perception range to a larger scale, such as $100 \times 50\,m$, their performance significantly deteriorates. *(2) Not leveraging temporal information:* These approaches only leverage single-frame inputs and fail to exploit temporal information. As a result, these methods are prone to errors caused by challenging environmental conditions such as occlusions, large crossroads, and extreme camera exposures, which are quite frequent in autonomous driving scenarios. Additionally, temporal inconsistency between maps of different timestamps is extremely challenging for the motion planning module as it creates a constantly changing world for autonomous driving system. To address these issues, we present StreamMapNet, an end-to-end online pipeline that

*Corresponding at: hangzhao@mail.tsinghua.edu.cn

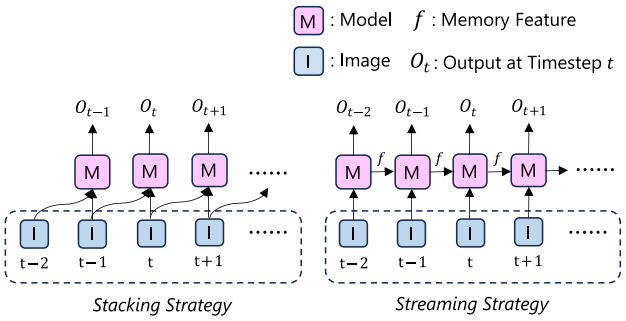

Figure 1. Comparison between *stacking* strategy and *streaming* strategy. *streaming* strategy encodes all historical information into the memory feature to save cost and build long-term association.

utilizes camera videos with a temporal fusion strategy to construct temporal-consistent high-quality vectorized maps covering a wide range.

We frame the map construction task as a detection problem. Our model adopts an encoder-decoder architecture: a general BEV encoder that aggregates features from multiple-view images, and a DETR-like decoder for decoding map element instances. Specifically, we assign one object query to associate with each map element. Unlike typical object detection scenarios where objects exhibit local characteristics, map elements often possess irregular and elongated shapes, necessitating long-range attention modeling that conventional deformable attention [29] fails to capture. To enable a wide perception range, we introduce a novel approach called "Multi-Point Attention" that effectively captures longer attention ranges while maintaining computational efficiency.

We adopt a *streaming* strategy for our temporal fusion approach. Instead of *stacking* multiple frames together, our method process each frame individually while propagating a hidden state across time to preserve temporal information, as demonstrated in Figure 1. This *streaming* strategy offers two key advantages over the *stacking* strategy: (1) it facilitates longer temporal associations as the propagated hidden states encode all historical information, and (2) it minimizes memory and latency costs compared to the *stacking* strategy, which consumes memory and computational resources linearly with the number of stacked frames. Recent works in 3D object detection share the same spirit (VideoBEV [5], StreamPETR [24], Sparse4D v2 [14]). In our framework, the propagated hidden states encompass BEV features and refined object queries. We design a dedicated temporal fusion module for each of these components.

Lastly, we critically examine the current evaluation setup in the nuScenes dataset [1], a common benchmark for recent online vectorized map construction methods such as VectorMapNet [15], MapTR [12], and BeMapNet [20]. Our investigation reveals substantial fairness issues in this setup due to a problematic training-validation split. Specifically, we find that more than 84% of validation locations are also present in the training split, which could lead to overfitting. We identified a similar issue with the Argoverse2 dataset [25]. In response, we propose using new, non-overlapping training-validation splits for both datasets, aiming to build a fairer benchmark for future research in this area. To ensure an equitable comparison, we perform extensive experiments on both the original and new splits for each dataset. The resulting quantitative results confirm the superior performance of our method across all experimental settings, surpassing all existing state-of-the-art approaches.

To summarize, our contributions are as follows:

- We introduce a novel approach called "Multi-Point Attention" to extend the perception range of local vectorized HD maps to 100×50 meters, demonstrating improved practicality without experiencing a significant performance drop.
- We design a model that effectively leverages temporal information using *streaming* strategy in our proposed temporal fusion module to improve the temporal consistency and quality of vectorized local HD map.
- We identify and address significant fairness concerns within the current evaluation setting by establishing a fairer benchmark. In both original and new settings, our method consistently outperforms existing state-of-the-art approaches.

2. Related Works

2.1. Online Vectorized Local HD Map Construction

In recent times, there has been a significant focus on utilizing onboard sensors in autonomous driving vehicles for the construction of vectorized local HD maps. HDMapNet [9] initially generates Bird's-Eye-View (BEV) semantic segmentations, followed by a heuristic and time-consuming pose-processing step to generate vectorized map instances. They also propose using mean Average Precision (mAP) as an evaluation metric. VectorMapNet [15] introduces the first end-to-end model that utilizes transformers. It employs a DETR [2] decoder to detect map elements and subsequently refines them with an auto-regressive transformer, enabling the construction of fine-grained shapes. However, the auto-regressive model necessitates a long training schedule and leads to reduced inference speed. MapTR [12] adopts a one-stage transformer approach to decode map elements using hierarchical queries. Nevertheless, its performance suffers when extending to a wider perception range due to the complex associations among numerous queries. BeMapNet [20] utilizes B'ezier curves along with handcrafted rules to model map elements.

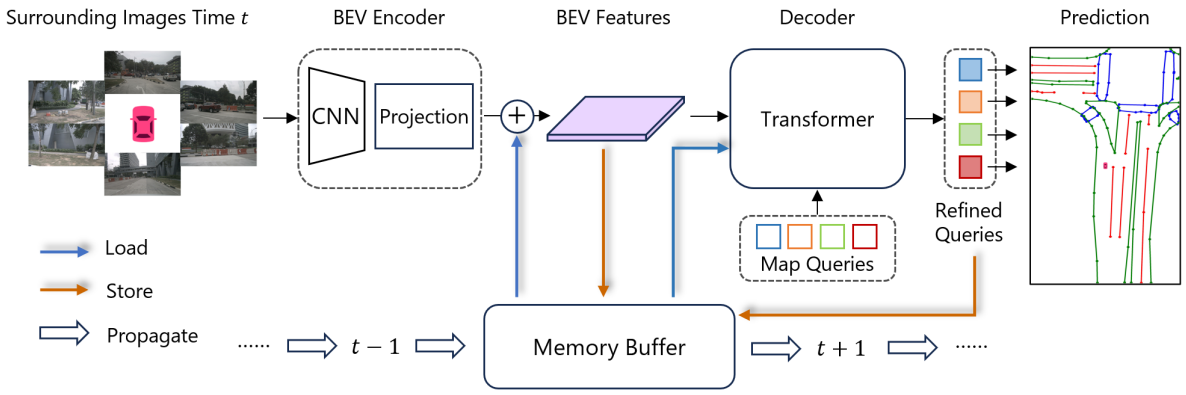

Figure 2. Pipeline of proposed model. Our model architecture comprises three main components: a general image backbone equipped with a BEV encoder for BEV feature extraction, a transformer decoder utilizing Multi-Point Attention for generating predictions, and a memory buffer to store propagated memory features.

2.2. Bird's-Eye-View Perception

BEV perception techniques have been extensively studied in the domains of 3D object detection and BEV segmentation tasks. Lift-Splat-Shoot [19] proposes using per-pixel predicted depth to lift image features to 3D space. BEVFormer [10] employs deformable attention operations to aggregate image features using learnable BEV queries. SimpleBEV [6] utilizes a variant of Inverse Perspective Mapping [17] (IPM) to sample features from 2D images to predefined BEV anchor points.

2.3. Temporal Modeling for Camera-Based 3D Object Detection

Inferring 3D space directly from a single-frame camera image is inherently challenging. Recent advancements in camera-based 3D object detection have explored leveraging temporal information to enhance perception outcomes. Some approaches (BEVDet4D [8], BEVFormer v2 [27]) employ a *stacking* strategy, where multiple historical frames or features are stacked and processed together in a single forward pass. However, this strategy incurs significant computational and memory costs that scale linearly with the number of *stacked* frames, thereby reducing training and inference speed while consuming substantial GPU memory. Consequently, the number of *stacked* frames is often limited, resulting in only short-term temporal fusion. In contrast, recent methods including VideoBEV [5], StreamPETR [24] and Sparse4D v2 [14] introduce a *streaming* fusion strategy. This strategy treats image sequences as streaming data and processes each frame individually, utilizing memory features propagated from the previous frame. Compared to the *stacking* strategy, the *streaming* strategy enables longer temporal associations while saving GPU memory and reducing latency. VideoBEV [5] propagates BEV features as memory features. Sparse4D v2 [14], as our concurrent work, propagates object queries as memory features.

3. StreamMapNet Model

3.1. Overall Architecture

Our model processes sequences of synchronized multi-view images, collected by autonomous vehicles, to create local HD maps. These maps are represented as a set of vectorized instances, each instance consisting of a class label and a polyline parameterized by a sequence of points $\boldsymbol{P} = \{(x_i, y_i)\}_{i=1}^{N_p}$.

As demonstrate in Figure 2, our model architecture comprises three main components: a general image backbone equipped with a Bird's Eye View (BEV) encoder for BEV feature extraction, a transformer decoder utilizing Multi-Point Attention for generating predictions, and a memory buffer to store propagated memory features.

3.2. BEV Feature Encoder

A shared CNN image backbone is first employed to extract 2D features from multi-view images. Subsequently, these features are aggregated and processed by a Feature Pyramid Network [13] (FPN). Finally, a BEV feature extractor is applied to lift 2D features to BEV space to obtain the BEV feature $\mathcal{F}_{\text{BEV}} \in \mathbb{R}^{C \times H \times W}$.

3.3. Decoder Transformer

While the DETR [2] Transformer decoder has demonstrated potency in 3D object detection models operating on 2D BEV features (BEVFormer [10]), its application to HD map construction is not straightforward due to fundamental differences between the tasks. Our approach comprises two key elements in the design of our decoder.

Query Design. Our approach assigns one query to each

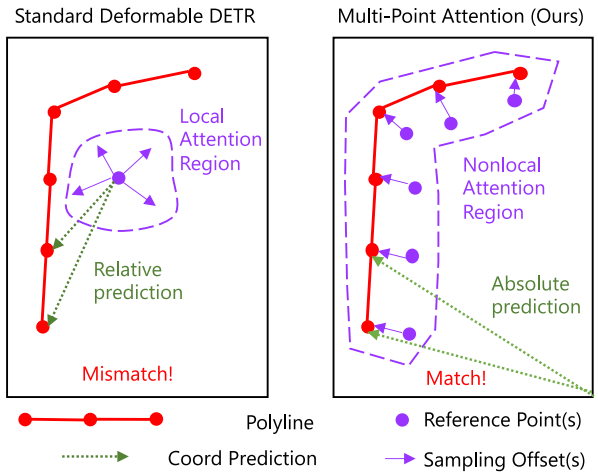

Figure 3. This illustration contrasts conventional deformable DETR with **Multi-Point Attention**. The former one restrict attention to a localized area, which mismatches the elongate shapes of map elements. Our solution builds a more flexible, non-local attention region.

map instance, which could be a complete pedestrian crossing or a continuous road boundary, resulting in total N_q queries. Conceptually, each query encodes both semantic and geometric information of a map element instance, thus enhancing global scene understanding during self-attention operations. When doing bipartite matching in training, each query matches a ground-truth instance or a background class. We leave the matching cost part to section 3.5. During the decoding stage, each query generates a class score and N_p point coordinates through Multi-Layer Perceptrons (MLPs).

Multi-Point Attention. To fit with our query design, we replace the conventional deformable DETR [29] design with our proposed **Multi-Point Attention** in cross attention operation, as demonstrated in Figure 3. In 3D object detection, an object is typically *local*, occupying a small area close to its center in BEV space. The conventional deformable DETR assigns a reference point to each query as an anchor for collecting features from the constructed BEV features. At every transformer layer, this reference point is adjusted towards the object's center by predicting a residual offset relative to its previous location. For the sake of brevity, we present the formulation of the i-th layer below, omitting the self-attention and feed-forward network components:

$$\boldsymbol{O}_i = \text{Offset\_Embed}(Q_{i-1}) \tag{1}$$

$$\boldsymbol{W}_i = \text{Weight\_Embed}(Q_{i-1}) \tag{2}$$

$$Q_i = \sum_{j=1}^{N_{\text{off}}} \boldsymbol{W}_i^j \cdot \text{DA}(Q_{i-1}, R_i + \boldsymbol{O}_i^j, \mathcal{F}_{\text{BEV}}) \tag{3}$$

$$R_{i+1} = \text{sigmoid}(\text{sigmoid}^{-1}(R_i) + \text{Reg}_i(Q_i)) \tag{4}$$

Here, $\text{DA}(Q, x, \mathcal{F})$ denotes the deformable attention operation that uses Q as a query to collect features at location x on $\mathcal{F}$. $\boldsymbol{O}_i$ represents the sampling offsets, N_{off} the number of sampling offsets for each query, $\boldsymbol{W}_i$ the sampling weights, R_i the reference points, and Reg_i the object center regression branch. The subscript i indicates the i-th layer and supercript j indicates the j-th element.

However, a map element may display a highly irregular and elongated shape, making it *nonlocal* in BEV space. Therefore, our method use the N_p predicted points, rather than the object center, from the previous layer as the reference points in the current layer. While facilitating long-range attention in BEV space, this approach maintains low complexity: $O(N_p)$, compared to $O(HW)$ for global attention. We employ a shared MLP for all layers as the regression branch to predict the absolute coordinates rather than a residual offset. The i-th layer can then be formulated as follows:

$$\boldsymbol{O}_i = \text{Offset\_Embed}(Q_{i-1}) \tag{5}$$

$$\boldsymbol{W}_i = \text{Weight\_Embed}(Q_{i-1}) \tag{6}$$

$$Q_i = \sum_{j=1}^{N_p} \sum_{k=1}^{N_{\text{off}}} \boldsymbol{W}_i^{(j-1)\cdot N_{\text{off}}+k} \cdot \text{DA}(Q_{i-1}, \boldsymbol{P}_i^j + \boldsymbol{O}_i^{(j-1)\cdot N_{\text{off}}+k}, \mathcal{F}_{\text{BEV}}) \tag{7}$$

$$\boldsymbol{P}_{i+1} = \text{sigmoid}(\text{Reg}(Q_i)) \tag{8}$$

Please note that in this context, $\boldsymbol{P}_i^j$ represent the coordinates of the j-th points on the predicted polyline at i-th layer.

3.4. Temporal Fusion

This section describes two temporal fusion modules that integrate temporal information from memory features into the current frame: **Query Propagation** and **BEV Fusion**.

Query Propagation. In map construction scenarios, all map instances are static, suggesting that instances in the current frame are likely to persist in subsequent frames. This motivates us to propagate queries with the highest k confidence scores to the next frame, providing a valuable positional prior and retaining temporal features across all historical frames. Given that we employ ego-coordinates, these propagated queries must be transformed before utilization. We employ a MLP with a residual connection to facilitate this transformation in the latent space.

$$Q_t = \phi_{\text{t}}\left(\text{Concat}(Q_{t-1}, \text{flatten}(\boldsymbol{T}))\right) + Q_{t-1} \tag{9}$$

Here, $\boldsymbol{T}$ denotes a standard 4×4 transformation matrix between the coordinate systems of two frames. We also convert the predicted N_p-point polyline to the new coordinate system to serve as the initial reference points for the propagated queries.

$$\boldsymbol{P}_t = \boldsymbol{T} \cdot \text{homogeneous}(\boldsymbol{P}_{t-1})_{:,0:2} \tag{10}$$

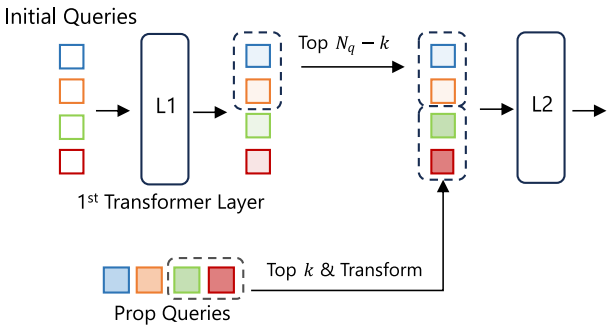

Figure 4. The top k refined queries are propagated from the previous frame. Post transformation, these queries are integrated with the top $N_q - k$ queries from the first Transformer layer, assembling a refreshed set of N_q queries.

Figure 4 illustrates the incorporation of propagated queries into the decoder. We use initial object queries in the first decoder. Post the initial decoder layer, we select the top $N_q - k$ queries based on confidence scores as potential foreground queries, and concatenate them with propagated queries. This approach aligns with the principles of Sparse4D v2 [14]. We add an auxiliary transformation loss to assist transformation learning:

$$\hat{\boldsymbol{P}} = \mathrm{Reg}(Q_t) \tag{11}$$

$$\mathcal{L}_{\text{trans}} = \sum_{j=1}^{N_p} \mathcal{L}_{\text{SmoothL1}}(\hat{\boldsymbol{P}}^j, \boldsymbol{P}_t^j) \tag{12}$$

BEV Fusion. While **Query Propagation** operates temporal association on sparse queries, the dense BEV features can also benefit from incorporating historical features. We recurrently propagate BEV features and warp them based on the ego vehicle's pose, as illustrated in Figure 5. Drawing inspiration from the Neural Map Prior [26], we employ a Gated Recurrent Unit [3] (GRU) to fuse these BEV features. To ensure training stability, we introduce a layer normalization operation in the final step.

$$\tilde{\mathcal{F}}_{\text{BEV}}^{t-1} = \mathrm{Warp}\left(\mathcal{F}_{\text{BEV}}^{t-1}, \boldsymbol{T}\right) \tag{13}$$

$$\mathcal{F}_{\text{BEV}}^{t} = \mathrm{LayerNorm}\left(\mathrm{GRU}\left(\tilde{\mathcal{F}}_{\text{BEV}}^{t-1}, \mathcal{F}_{\text{BEV}}^{t}\right)\right) \tag{14}$$

3.5. Matching Cost and Training Loss

Our model adopts an end-to-end training approach. We employ standard bipartite matching to pair predicted map instances with their ground-truth counterparts, denoted as $(c_i, \boldsymbol{P}_i)i = 1^{N\text{gt}}$. Predicted instances are represented as $(\hat{c}_i, \hat{\boldsymbol{P}}i)i = 1^{N_q}$. In this context, we slightly modify the notation $\hat{\boldsymbol{P}}_i$ to indicate the predicted polyline of the i-th query, diverging from equation 8. The polyline-wise matching cost

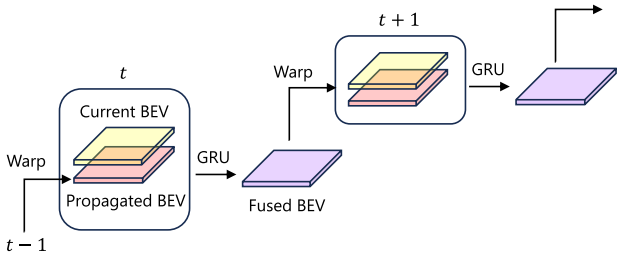

Figure 5. The propagated BEV feature, serving as a recurrent memory, is warped and updated for each frame.

is defined as follows:

$$\mathcal{L}_{\text{line}}(\hat{\boldsymbol{P}}, \boldsymbol{P}) = \min_{\gamma \in \Gamma} \frac{1}{N_p} \sum_{j=1}^{N_p} \mathcal{L}_{\text{SmoothL1}}(\hat{p}_j, p_{\gamma(j)}) \tag{15}$$

In this case, $\hat{p}_j$ signifies the j-th point of $\hat{\boldsymbol{P}}$. The permutation group introduced by MapTR [12] is denoted by Γ. For the classification matching cost, we utilize Focal Loss. The final matching cost is then expressed as:

$$\begin{aligned} &\mathcal{L}_{\text{match}}\left((\hat{c}_i, \hat{\boldsymbol{P}}_i), (c_i, \boldsymbol{P}_i)\right) = \\ &\lambda_1 \mathcal{L}_{\text{line}}(\hat{\boldsymbol{P}}, \boldsymbol{P}) + \lambda_2\, \mathcal{L}_{\text{Focal}}(\hat{c}_i, c_i) \end{aligned} \tag{16}$$

Despite the auxiliary loss presented in equation 12, the training loss mirrors the structure of the matching cost and is defined as:

$$\mathcal{L}_{\text{train}} = \lambda_1 \mathcal{L}_{\text{line}} + \lambda_2\, \mathcal{L}_{\text{Focal}} + \lambda_3 \mathcal{L}_{\text{trans}} \tag{17}$$

4. Experiments

4.1. Rethinking on Datasets

NuScenes [1], a popular benchmark in autonomous driving research, offers around 1000 scenes with six synchronized cameras and precise ego-vehicle poses. Most online HD map construction models test their performance on this dataset, following the official 700/150/150 split for training/validation/testing scenes. However, this split, intended for object detection tasks, falls short of map construction.

We found an overlap of over 84% of locations between the training and validation sets. This overlap might not be problematic for object detection, given the significant variance in objects across different traversals. Yet, the map remains essentially unchanged, which means a model could simply memorize location-HD map pairs from the training set and perform exceptionally well on the validation set. However, such a model would completely fail to generalize to new scenes. This clearly contradicts the essence of online map construction: the goal is to develop models that can generalize to unseen environments and adapt to potential map changes, rather than simply memorizing the training set.

Range	Method	Backbone	Image Size	Epoch	AP_{ped}	AP_{div}	AP_{bound}	mAP	FPS
$60 \times 30\,m$	VectorMapNet	R50	384×384	120	35.6	34.9	37.8	36.1	5.5
	MapTR	R50	608×608	30	48.1	50.4	55.0	51.1	**18.0**
	StreamMapNet (Ours)	R50	608×608	30	**56.9**	**55.9**	**61.4**	**58.1**	14.2
$100 \times 50\,m$	VectorMapNet	R50	384×384	120	32.4	20.6	24.3	25.7	5.5
	MapTR	R50	608×608	30	46.3	36.3	38.0	40.2	**18.0**
	StreamMapNet (Ours)	R50	608×608	30	**60.5**	**44.4**	**48.6**	**51.2**	14.2

Table 1. Performance comparison of various methods on the new Argoverse2 split at both $30m$ and $50m$ perception ranges. StreamMapNet notably outperforms other methods in all categories, exhibiting robustness to long perception ranges due to its integration of temporal association and long-range attention mechanism.

Similar issues were detected with Argoverse2 [25], another dataset with 1000 scenes from six cities, where we identified a 54% overlap between validation and training locations. To address this, we use new training/validation splits for both datasets that minimize overlap and ensure balanced location, object, and weather conditions distribution. We employ Roddick and Cipolla's [21] splits for NuScenes, and introduce a new split for Argoverse2. Both splits result in a 700/150 division for training/validation scenes. It will be released along with the code. It's worth noting that when we discuss results on a specific split, we are referring to models trained on that split's training set and evaluated on its validation set. While we primarily compare performance on these new splits, we also present results on the original splits for thorough and equitable comparison with existing methods.

4.2. Implementation Details

Our model trains on 8 GTX3090 GPUs with a batch size of 32, using an AdamW optimizer [16] with a learning rate of 5×10^{-4}. We adopt ResNet50 [7] as backbones and use BEVFormer [10] with a single encoder layer for BEV feature extraction, consistent with MapTR [12]. The model trains for 24 epochs on the NuScenes dataset and 30 epochs on Argoverse2. We set $N_q = 100$, $N_{\text{off}} = 1$, $N_p = 20$, $k = 33$, $\lambda_1 = 50.0$, $\lambda_2 = 5.0$, $\lambda_1 = 5.0$ as the hyperparameters for all settings and perception ranges without further tuning.

Streaming Training. We adopt the *streaming* training strategy for temporal fusion, as illustrated in Figure 1. Gradients on memory features do not propagate back to previous frames. For each training sequence, we randomly divide it into 2 splits at the start of each training epoch to foster more diverse data sequences. During inference, we use the entire sequences. To stabilize streaming training, we train the initial 4 epochs with single-frame input, inspired by SOLOFusion [18].

4.3. Metrics

In line with existing works, we consider three types of map elements: pedestrian crossings, lane dividers, and road boundaries. We enlarge the perception range to cover an area of $50\,m$ front and back, and $25\,m$ left and right, aligning with the scope of 3D object detection tasks. Concurrently, we also present results for a smaller range ($30\,m$ front and back, $15\,m$ left and right), as used by prior works. We adopt Average Precision (AP) as the evaluation metric proposed in [9] and [15]. AP calculations are conducted under distinct thresholds: $\{1.0\,m, 1.5\,m, 2.0\,m\}$ for $50\,m$ range, and $\{0.5\,m, 1.0\,m, 1.5\,m\}$ for the $30\,m$ range.

4.4. Comparison with Baselines

We implemented VectorMapNet [15] and MapTR [12] using their official codebases, altering only the perception range and training-validation split. VectorMapNet's input image size was adjusted to suit the memory of an RTX3090 GPU. For MapTR, BEVFormer [10] was used as the BEV feature extractor to ensure a fair comparison with our model. As BeMapNet's [20] code is not publicly available, we could only compare with their reported results on the original NuScenes split with a $30\,m$ perception range.

4.4.1 Performance on Argoverse2 Dataset

Argoverse2 dataset originally provide 10 Hz camera frame rate. To align with the NuScenes setup, we set the camera frame rate to 2 HZ.

New Split. Table 1 showcases the performance comparison on the new Argoverse2 split. We report results for both $30\,m$ and $50\,m$ perception ranges. At both both perception ranges, StreamMapNet demonstrates superior performance over other methods across all categories while maintaining a online inference speed, showing the effectiveness of our approach. Existing methods experience a significant drop in mAP when the perception range is increased.In contrast, our method is more robust due to the incorporation of temporal associations and long-range attention mechanism.

Original Split. Table 3 presents performance results on the original training/validation split at the $50\,m$ range for a comprehensive comparison. StreamMapNet consistently surpasses other methods on the original split by a significant margin of at least 10.2 mAP. A significant performance

Range	Method	Backbone	Image Size	Epoch	AP_{ped}	AP_{div}	AP_{bound}	mAP	FPS
$60 \times 30\,m$	VectorMapNet	R50	256×480	120	15.8	17.0	21.2	18.0	3.8
	MapTR	R50	480×800	24	6.4	20.7	35.5	20.9	**16.0**
	StreamMapNet (Ours)	R50	480×800	24	**29.6**	**30.1**	**41.9**	**33.9**	13.2
$100 \times 50\,m$	VectorMapNet	R50	256×480	120	12.0	8.1	6.3	8.8	3.8
	MapTR	R50	480×800	24	8.3	16.0	20.0	14.8	**16.0**
	StreamMapNet (Ours)	R50	480×800	24	**24.8**	**19.6**	**24.7**	**23.0**	13.2

Table 2. Performance comparison with baseline methods on the new NuScenes split at both $30\,m$ and $50\,m$ perception ranges. SteamMapNet outperforms existing methods. While StreamMapNet exhibits superior performance compared to existing methods, all approaches experience a performance reduction relative to the results obtained using the Argoverse2 new split.

Method	Image Size	Epoch	mAP
VectorMapNet	384×384	120	30.2
MapTR	608×608	30	47.5
StreamMapNet (Ours)	608×608	30	**57.7**

Table 3. Performance comparison on the original Argoverse2 training/validation split at a $50\,m$ perception range. Our method consistently outperforms other methods.

gap can be found when comparing results between the new split and the original split (Table 1), indicating the overfitting problem of the original split cannot be ignored.

4.4.2 Performance on NuScenes Dataset

New Split. Table 2 compares performance on the new NuScenes split at both $30\,m$ and $50\,m$ perception ranges. Our method shows a considerable improvement, surpassing existing methods by 13.0 mAP at the $30\,m$ range and 8.2 mAP at the $50\,m$ range.

A decline in performance is observed across all methods when compared to results on the Argoverse2 dataset (Table 1). We attribute this to two main reasons: (1) The Argoverse2 dataset offers images from more cameras with higher resolutions (7 cameras with resolution 1550×2048), whereas NuScenes provides images from 6 cameras with a resolution of 900×1600. Cameras in Argoverse2 are positioned at higher viewpoints, thus providing a longer viewing range. (2) The Argoverse2 dataset contains more diverse training data with locations across six different cities, in contrast to NuScenes' two cities. Despite these challenges, NuScenes remains a valuable benchmark for evaluating online map construction tasks.

Original Split. A majority of existing methods primarily evaluate their results on the original NuScenes split at a perception range of $30\,m$. Despite the tendency for overfitting within this setting, we provide a comparison of StreamMapNet's performance against these methods to ensure comprehensive analysis. As seen in Table 4, StreamMapNet outperforms existing methodologies even with fewer or equivalent training epochs. A comparison with the results in Table 2 reveals that transitioning to the new split induces approximately a 50% performance decrease across all methods, which fully substantiates the concerns raised in section 4.1. The original split's validation set seems prone to overfitting, rendering it less reliable when evaluating the generalization capabilities of online map construction models, especially those utilizing a large backbone with a higher capacity for memorization.

Method	Image Size	Epoch	mAP
VectorMapNet	256×480	110	40.9
MapTR	480×800	24	48.7
BeMapNet [20]	512×896	30	59.8
StreamMapNet (Ours)	480×800	24	**62.9**

Table 4. Performance comparison on the original NuScenes split with $30\,m$ range, a widely used benchmark for evaluating online map construction tasks. StreamMapNet outperforms existing methods. However, this validation set in this split is prone to overfitting.

4.5. Ablation Studies

Index	Method	mAP
(a)	Single-frame baseline w. relative predict	33.7
(b)	− Multi-Point attention	-
(c)	+ Direct predict	41.7
(d)	+ Query propagation (w.o. trans. loss)	42.8
(e)	+ Transformation loss	43.7
(f)	+ BEV fusion	46.1
(g)	+ Image size 608×608	51.2
	StreamMapNet	**51.2**

Table 5. Ablation study of each component. Starting from a single-frame baseline model to the full model. Each modification contributes to the performance gain.

We examine the efficacy of each StreamMapNet component through ablation studies, utilizing the new Argoverse2 split at a $50\,m$ perception range. Unless stated otherwise, we adjust image sizes to 384×384. The influence of each

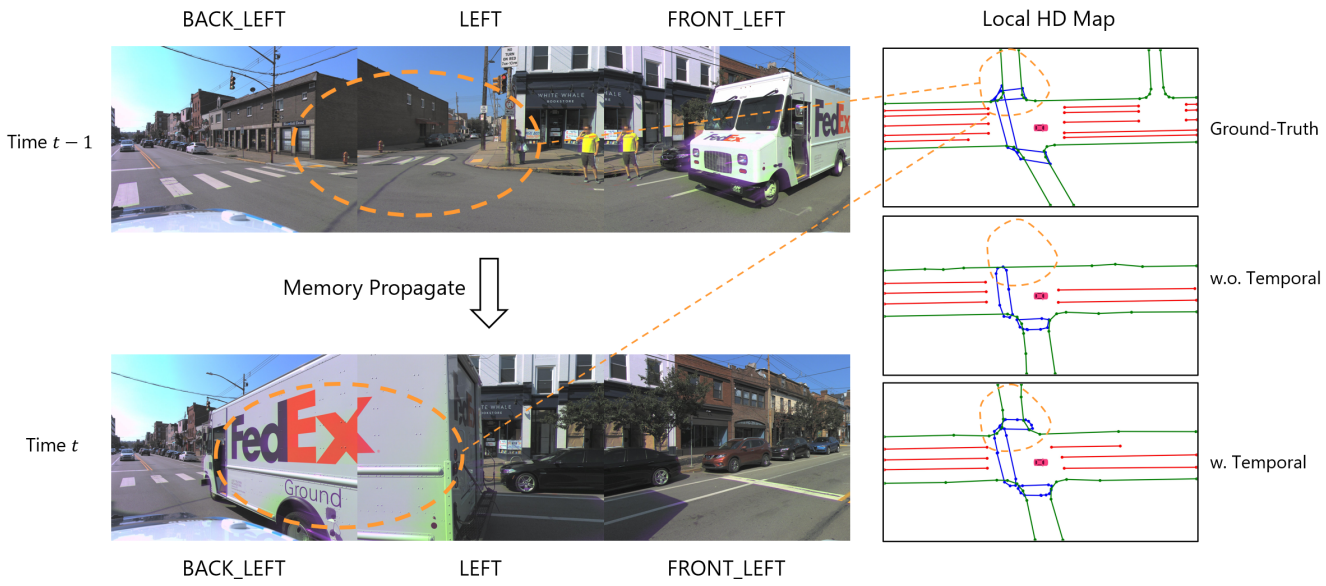

Figure 6. We compare models with and without temporal information in occlusion scenarios. At Time t, the crossroad is occluded by a white truck (highlighted by the orange circle). The model incorporating temporal information successfully constructs the road structure, while the single-frame model falls short. In the HD maps, *green* lines denote road boundaries, *red* lines indicate lane denote, and *blue* lines denote pedestrian crossings.

component is demonstrated in Table 5. Initially, we build a single-frame baseline model employing only Multi-Point attention. The term *relative predict* refers to the operation in Equation 4, a technique commonly found in object detection models. To assess the importance of Multi-Point Attention, we replace it with the conventional deformable design, in which each query is allocated a reference point at its center of gravity (substituting Equation 7 with Equation 3). This replacement hampers model convergence due to the restrictive attention range, validating the necessity of Multi-Point Attention. For integration with Multi-Point Attention, we replace *Predict Refinement* with *Direct Predict*, achieving a robust single-frame model. Progressing from model (d) to (f), we gradually introduce the temporal fusion components, consistently enhancing performance and underscoring the significance of temporal information association in online map construction tasks.

4.6. Qualitative Analysis

In this section, we present qualitative results from our StreamMapNet, emphasizing the importance of temporal modeling by comparing it with a single-frame model. Figure 6 illustrates a commonplace scenario in autonomous driving where a large truck obstructs part of the camera's field of view. For better visualization, we focus only on images from the left side. The ground-truth map indicates a crossroad to the left of the ego vehicle, an area briefly hidden by the truck in the current frame. Without the benefit of temporal information, the single-frame model, lacking visual information beyond the truck, fails to accurately replicate the crossroad. However, our model effectively uses temporal information from previous frames to correctly reproduce the road structure. This leads to the generation of a stable and reliable HD map, which is vital to ensure the safety of autonomous vehicles.

5. Conclusion & Acknowledgement

In this study, we have proposed an end-to-end model for the online construction of vectorized, local HD maps. By leveraging temporal information, our approach promotes stability in wide-range map perception. Importantly, we scrutinize the prevalent evaluation settings on NuScenes and Argoverse2 datasets and identified improper training/validation division that leads to the overfitting problem. As a remedy, we propose new, non-overlapping splits for both datasets. We hope that these refined splits will foster a more balanced benchmark for future research in this field.

Discussion of Potential Negative Societal Impact. While our model significantly improves upon existing methods, it may still make false predictions in challenging scenarios, which underlines the necessity for comprehensive safety testing before deploying our model in real autonomous driving vehicles. Moreover, HD map data can be sensitive. The collection and use of such data might violate laws and regulations in certain countries or regions. As such, it is imperative to take the necessary precautions before gathering this kind of data and training our model on it, to ensure the privacy and legal rights of individuals are respected.

Acknowledgement This work is supported by Tsinghua University Dushi Program.

References

[1] 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/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2, 5

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers, 2020. 1, 2, 3

[3] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling, 2014. 5

[4] Jiyang Gao, Chen Sun, Hang Zhao, Yi Shen, Dragomir Anguelov, Congcong Li, and Cordelia 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. 1

[5] Chunrui Han, Jianjian Sun, Zheng Ge, Jinrong Yang, Runpei Dong, Hongyu Zhou, Weixin Mao, Yuang Peng, and Xiangyu Zhang. Exploring recurrent long-term temporal fusion for multi-view 3d perception, 2023. 2, 3

[6] Adam W. Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. A simple baseline for bev perception without lidar. In *arXiv:2206.07959*, 2022. 3

[7] 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. 6

[8] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 3

[9] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: A local semantic map learning and evaluation framework. *arXiv preprint arXiv:2107.06307*, 2021. 1, 2, 6

[10] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 3, 6

[11] Ming Liang, Bin Yang, Rui Hu, Yun Chen, Renjie Liao, Song Feng, and Raquel Urtasun. Learning lane graph representations for motion forecasting. In *ECCV*, 2020. 1

[12] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. Maptr: Structured modeling and learning for online vectorized hd map construction. In *International Conference on Learning Representations*, 2023. 1, 2, 5, 6

[13] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection, 2017. 3

[14] Xuewu Lin, Tianwei Lin, Zixiang Pei, Lichao Huang, and Zhizhong Su. Sparse4d v2: Recurrent temporal fusion with sparse model, 2023. 2, 3, 5

[15] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning, 2023. 1, 2, 6

[16] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 6

[17] Hanspeter A Mallot, Heinrich H Bülthoff, JJ Little, and Stefan Bohrer. Inverse perspective mapping simplifies optical flow computation and obstacle detection. *Biological cybernetics*, 64(3):177–185, 1991. 3

[18] Jinhyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris Kitani, Masayoshi Tomizuka, and Wei Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. 2023. 6

[19] 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, 3

[20] Limeng Qiao, Wenjie Ding, Xi Qiu, and Chi Zhang. End-to-end vectorized hd-map construction with piecewise bezier curve, 2023. 2, 6, 7

[21] Thomas Roddick and Roberto 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. 1, 6

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

[23] 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. 1

[24] Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. *arXiv preprint arXiv:2303.11926*, 2023. 2, 3

[25] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks (NeurIPS Datasets and Benchmarks 2021)*, 2021. 2, 6

[26] Xuan Xiong, Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Zhao Hang. Neural map prior for autonomous driving. In *CVPR*, 2023. 5

[27] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, Jie Zhou, and Jifeng Dai. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision, 2022. 3

[28] Ji Zhang and Sanjiv Singh. Loam : Lidar odometry and mapping in real-time. *Robotics: Science and Systems Conference (RSS)*, pages 109–111, 01 2014. 1

[29] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 2, 4