Stream Query Denoising for Vectorized HD-Map Construction

Shuo Wang[1*†], Fan Jia[2*], Weixin Mao[3†], Yingfei Liu[2], Yucheng Zhao[2], Zehui Chen[1], Tiancai Wang[2], Chi Zhang[4], Xiangyu Zhang[2], and Feng Zhao[1✉]

[1] MoE Key Laboratory of Brain-inspired Intelligent Perception and Cognition, University of Science and Technology of China
[2] MEGVII Technology [3] Waseda University [4] Mach Drive
shuowang2323@mail.ustc.edu.cn

Abstract. This paper introduces the Stream Query Denoising (SQD) strategy, a novel and general approach for high-definition map (HD-map) construction. SQD is designed to improve the modeling capability of map elements by learning temporal consistency. Specifically, SQD involves the process of denoising the queries, which are generated by the noised ground truth of the previous frame. This process aims to reconstruct the ground truth of the current frame during training. Our method can be applied to both static and temporal methods, showing the great effectiveness of SQD strategy. Extensive experiments on nuScenes and Argoverse2 show that our framework achieves superior performance, compared to other existing methods across all settings. Code will be available here.

Keywords: HD-Map · Temporal learning · Query denoising

1 Introduction

The High-Definition Map (HD-map) serves the crucial purpose of furnishing centimeter-level location information for map elements and plays a pivotal role in various applications within autonomous driving, including localization [6, 23, 32, 33, 35, 38] and navigation [1, 2, 11]. Traditionally, the construction of HD-map is conducted offline through SLAM-based methods [30, 40], which is both time-consuming and labor-intensive. Recent research endeavors have shifted towards the construction of local maps within a predetermined range using onboard sensors. Although many existing works frame map construction as a semantic segmentation task [17, 24, 27, 29, 41], rasterized representations in such approaches exhibit redundant information, lack structural relationships between map elements, and often require extensive post-processing efforts [17]. In response to these limitations, MapTR [19] adopts an end-to-end approach to construct vectorized maps, akin to the DETR paradigm [4, 5, 21, 42].

† Work done during an internship at MEGVII Technology.
* Equal contribution. ✉ Corresponding author.

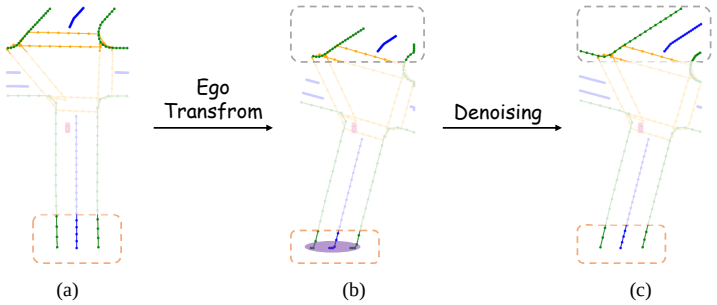


Fig. 1: (a) shows the ground truth of previous frame $t-1$. (b) is the warp result of (a) to current frame t, using ego motion. (c) is the ground truth of current frame.

Nevertheless, the aforementioned methods overlook the incorporation of temporal information. The efficacy of propagating sparse queries (hidden states) from the previous frame to the current frame has been demonstrated in temporal multi-view 3D object detection [20,31]. While recent approaches grounded in vectorized representations share a similar paradigm with object detection, the direct application of the previous temporal methods is not warranted due to inherent modeling variability between curves and bounding boxes. In the context of object detection, determining the speed of the ego and surrounding objects enables the prediction of their positions at the next timestamp. This stands in contrast to scenarios involving lines, where changes over time result in new parts appearing and old parts detaching from the line, presenting a distinctive challenge not encountered in object detection.

Suppose the model's predictions at the preceding moment precisely match the ground truth, and this valuable information is propagated to the current moment for improved initialization. Owing to ego-motion, the predictions must undergo transformation based on the matrix representing the transition between two frames. Fig. 1 (a) and (b) illustrate the curves before and after transformation, respectively, and the purple segment in (b) shows that a number of different points are transformed to almost the same position due to alterations in the perceptual range between the two frames. As shown in Fig. 1 (b) and (c), points within the orange dotted box must assimilate different biases despite originating from nearly the same boundary. At the same time, the gray dotted box signifies the newly added part of the curve, necessitating all points along the entire line to acquire distinct offsets to accommodate the curve's growth. Therefore, the learning of temporal information can be viewed as a denoising process. Explicitly teaching a network to grasp such intricate and diverse changes poses a challenge, and the temporal learning process runs counter to conventional training.

The alignment of curves between frames is the main difficulty in temporal learning of HD-Map. To address the above challenges, we propose a novel approach called Stream Query Denoising (SQD) for HD-map construction. As illustrated in Fig. 2 (a), our SQD strategy, which served as an auxiliary supervision, is designed to learn the temporal consistency among map elements during

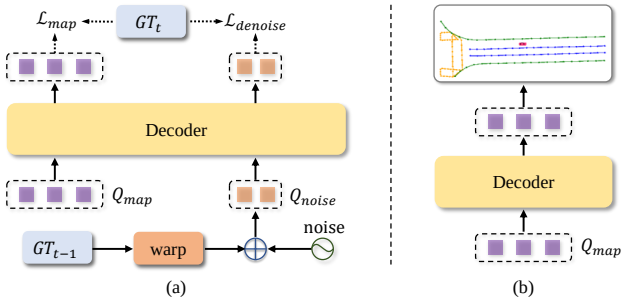


Fig. 2: (a) The diagram of stream query denoising during the training process. (b) The inference pipeline excludes the SQD without introducing extra computation costs.

training. SQD involves the process of denoising the queries Q_{noise}, which are generated by the noised ground-truth of the previous frame G_{t-1}. This process aims to reconstruct the ground truth of current frame G_t. During inference, the SQD process can be removed from the basis framework, without introducing any extra computation cost (see Fig. 2 (b)). The SQD strategy consists of two main components. First, we incorporate adaptive temporal matching to establish an explicit one-to-one correspondence between historical ground truths and current ones. Second, we introduce dynamic curve noising to dynamically decay the noise of the curves, considering the inherent stream noise.

In summary, the primary contributions of this paper are outlined as follows:

- We propose stream query denoising (SQD) to learn the temporal consistency of map elements for HD-map construction.
- The proposed SQD is a general strategy, especially bringing great performance improvements for temporal approaches.
- StreamMapNet with SQD showcases notable superiority over state-of-the-art methods on existing benchmarks.

2 Related Works

2.1 Online Static Vectorized HD-Map Construction

There has been a surge of interest in leveraging onboard sensors for the construction of vectorized local HD-maps. HDMapNet [17] employs a semantic map prediction approach, followed by the aggregation of pixel-wise segmentation results through post-processing. In an effort to mitigate redundant information and alleviate the need for time-consuming post-processing, VectorMapNet [22] introduces a refinement step for map elements using an auto-regressive transformer. MapTR [19] adopts hierarchical queries and a fixed number of points to represent the map, while BeMapNet [28] utilizes piecewise Bezier curves to model map elements. Additionally, PivotNet [8] presents a map construction method based on pivot-based representations.

2.2 Temporal Camera-based Perception

The significance of temporal information is paramount, especially in intricate scenarios involving long distances, occlusion. Notably, the utilization of temporal information has been a focal point in the domain of camera-based 3D object detection. Approaches such as BEVDet4D [13] and BEVFormer v2 [36] adopt the strategy of stacking features from multiple historical frames and processing them in a single forward pass. However, this method incurs substantial computational costs and imposes limitations on the number of historical frames that can be effectively utilized. In contrast, VideoBEV [10] incorporates a recurrent long-term fusion module to sequentially fuse BEV features in a video stream. StreamPETR [31] and Sparse4D v2 [20] introduce the streaming queries strategy to propagate temporal information. Notably, StreamMapNet [37] extends this core idea to the construction of HD-Map by applying streaming queries and streaming BEV features.

2.3 Query Denoising

DN-DETR [15] pioneers the utilization of query denoising to address the instability inherent in bipartite graph matching. DINO [39] extends this concept by introducing a definition of negative samples based on DN-DETR. Furthering this approach, MaskDINO [16] performs object detection and segmentation tasks concurrently. DN-MOT [9] tailors a denoising strategy to mitigate the impact of occlusion in multiple object tracking. In the context of this paper, we introduce the concept of stream query denoising for HD-map construction. To the best of our knowledge, it is the first work that explores the effectiveness of query denoising in temporal consistency learning for HD-map construction.

3 Preliminary

Since the proposed stream query denoising is a general technique for both static and temporal methods, their main difference lies in the query interaction in the transformer decoder. In this section, We mainly review the static query updating and the stream query propagation in static and temporal methods, respectively.

3.1 Static Query Updating

One typical static method for HD map construction is MapTR [19]. It adopts the structure of a transformer decoder in DETR [4, 42]. Specifically, each map element is represented by a query, which encodes both semantic and geometric information. The queries Q achieve the local perception by the (deformable) cross-attention with BEV features. The deformable cross-attention can be expressed as follows:

$$\boldsymbol{O},\ \boldsymbol{W} = \text{Offset\_Embed}(Q),\ \text{Weight\_Embed}(Q), \tag{1}$$

$$Q^{'} = \boldsymbol{W} \cdot \text{DA}(Q, P + \boldsymbol{O}, \mathcal{F}_{BEV}). \tag{2}$$

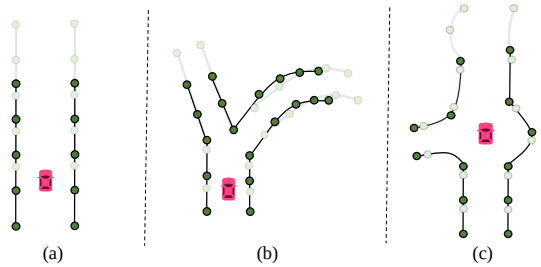


Fig. 3: The definition of stream noise under different cases. The dark green curves represent the wrapped ground-truth of the previous frame $t-1$. The light green curves is the ground-truth of current frame t.

Here $\mathrm{DA}(Q, X, \mathcal{F})$ denotes the deformable attention [42] operation that uses Q as query element to collect features at location X on feature $\mathcal{F}$. P is the reference point. $\boldsymbol{O}$ and $\boldsymbol{W}$ denote the sampling offsets and sampling weights for queries. $Q^{'}$ represents the updated queries.

3.2 Stream Query Propagation

Due to the static nature of map elements, there is a high probability that the instance at the current moment will continue to appear at the next moment, which means that the queries at present can provide a better reference position for the next moment than global initialization. As the only temporal approach, StreamMapNet [37] adopts the query propagation and performs temporal fusion. Concretely, for query propagation, queries Q_{t-1} with the highest k scores from timestep $t-1$ are selected and the corresponding predicted reference points P_{t-1} can be obtained. Considering the movement of the ego, it utilizes the transformation matrix T between the coordinate systems of two frames before propagation. The transformation process can be expressed as:

$$Q'_t = \phi_t(\mathrm{Concat}(Q_{t-1}, \mathrm{Flatten}(T))) + Q_{t-1}, \tag{3}$$

$$P'_t = T \cdot P_{t-1}. \tag{4}$$

To this end, the stream queries Q'_t and the reference point P'_t can be propagated to the next frame. Moreover, StreamMapNet employs a Gated Recurrent Unit [7] (GRU) to fuse these temporal BEV features.

4 Methodology

In this section, we first introduce the definition of stream noise. Then we introduce our approach: stream query denoising, which mainly includes the adaptive temporal matching and dynamic curve noising.

4.1 Definition of Stream Noise

As shown in Fig. 3, the dark green curves indicate the ground truth of the previous frame warped to the current frame through the ego-motion, and the

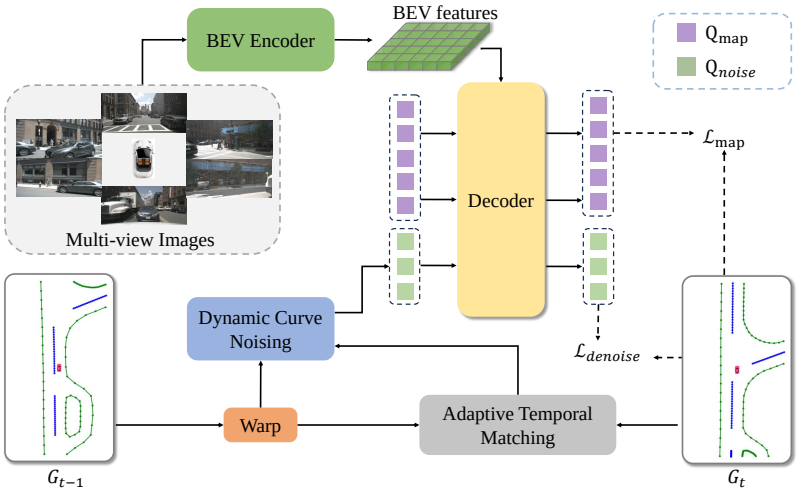


Fig. 4: The overall framework of stream query denoising. G_{t-1} is the ground-truth of previous frame $t-1$. Q_{map} and Q_{noise} are the map queries and noised queries, respectively. Distance and index are the matched results between G_{t-1} and G_t.

light green curves represent the ground truth of the current frame. It can be seen that the position offsets within adjacent frames are different for curves of different shapes and positions. In addition, points on a curve have different deviations, compared to the corresponding ground truth, due to their different positions. Therefore, the position offsets from the temporal perspective can be regarded as the stream noise.

4.2 Stream Query Denoising

As mentioned above, there exists the stream noise in the temporal propagation of map elements. Performing the denoising process on stream noise can help the model learn the temporal consistency among map elements. To facilitate the learning of temporal consistency, we propose Stream Query Denoising (SQD). The overall architecture of our method is shown in Fig. 4. The learning of SQD is an auxiliary supervision task, built on the HD-map construction framework (e.g., MapTR). It first warp the ground-truth of the previous frame G_{t-1} to the current frame by ego motion. Then the warped result and the ground-truth of the current frame G_t are input to the adaptive temporal matching module, producing the matched results (distance and index) between G_{t-1} and G_t. The matched distance as well as the warped result are further injected to the dynamic curve noising module to generate the noised queries Q_{noise}. The map queries Q_{map} and Q_{noise} are concatenated together and input to the transformer decoder for interaction with the BEV features. The updated noised queries are used to reconstruct the G_t. Next, we will describe each part in detail.

Adaptive Temporal Matching In the process of stream query denoising, we have no access to an explicit one-to-one correspondence between previous ground truth and current ground truth. In order to circumvent the instability associated with bipartite graph matching, we propose Adaptive Temporal Matching (ATM).

Let $\{Y_1^{t-1}, Y_2^{t-1}, \cdots, Y_m^{t-1}\}$ and $\{Y_1^t, Y_2^t, \cdots, Y_n^t\}$ represent the ground-truths of previous frame and the current frame, respectively. Here, Y is composed of a fixed number of points, and m and n denote the number of curves in the respective ground truth. Due to the movement of the ego, the transformation matrix T between the coordinate systems of the two frames is employed to convert $\{Y_1^{t-1}, Y_2^{t-1}, \cdots, Y_m^{t-1}\}$ to $\{\hat{Y}_1^{t-1}, \hat{Y}_2^{t-1}, \cdots, \hat{Y}_m^{t-1}\}$. For each curve in the current frame Y_i^t, we compute the Chamfer distance (CD) between it and each instance in the previous frame. The minimum distance and its corresponding location are preserved:

$$\mathrm{CD}_{Dir}(S_1, S_2) = \frac{1}{S_1} \sum_{p \in S_1} \min_{q \in S_2} ||p - q||_2, \tag{5}$$

$$\mathrm{CD}_{Bi}(S_1, S_2) = \mathrm{CD}_{Dir}(S_1, S_2) + \mathrm{CD}_{Dir}(S_2, S_1), \tag{6}$$

$$D, idx = \min_{j \in [1,m]} (\mathrm{CD}_{Bi}(Y_i^t, \hat{Y}_j^{t-1})), \tag{7}$$

where CD_{Dir} is the directional Chamfer distance and CD_{Bi} is the bi-directional Chamfer distance; S_1 and S_2 are the two sets of points on the curves; idx denotes the index closest to the curve itself and D is the minimum distance value.

Due to the inherent differences in the properties of each curve, different curves tolerate different noises for matching. Therefore, we propose to set a unique matching threshold for each instance. Suppose that a curve is composed of the point set $\{(p_1^x, p_1^y), (p_2^x, p_2^y), \cdots, (p_c^x, p_c^y)\}$, we define the scale of the curve as:

$$Scale = \sum_{i \in \{x,y\}} (\max(\{p_1^i, p_2^i, \cdots, p_c^i\}) - \min(\{p_1^i, p_2^i, \cdots, p_c^i\})), \tag{8}$$

where c is the number of the point set. As the scale of the curve gets larger, the curve becomes more tolerant of matching errors. Then the matching threshold δ is calculated as:

$$\delta = \alpha \cdot Scale, \tag{9}$$

where α is the degree of the tolerance.

Only when the minimum distance D in Eq. (7) is less than the matching threshold δ of the corresponding instance in the current frame, we will assign the instance to the current curve. Consequently, we establish the correspondence between the temporal ground truths and the current ones through ATM. The specific framework of ATM is shown in Fig. 5a.

Dynamic Curve Noising When transforming the ground truth of previous frame into current frame based on ego motion, the stream noise for different instances has no significant change. The small variance may easily lead the model

to overfitting. Hence, we propose dynamic curve noising, which adds dynamic random noise to enhance the diversity of curves. It is worth noting that perturbing all ground truths equally, without considering the instance's inherent stream noise, is sub-optimal. To address it, we propose to dynamically decay the noise of the curve. Specifically, after obtaining the matching results, the corresponding minimum Chamfer distance D can also be derived from Eq. (7). Subsequently, we define the decay rate of noise for each curve as follows:

$$R_{decay} = 1 - \frac{D}{\gamma \cdot Scale}, \tag{10}$$

where γ is the predefined decay scale and $Scale$ is acquired by Eq. (8).

Concretely, suppose that a curve is composed of the point set $\{(p_1^x, p_1^y), (p_2^x, p_2^y), \cdots, (p_c^x, p_c^y)\}$, and n is the number of the point set. Then we define the approximate pseudo scale and pseudo center of the curve as:

$$L = \max(\{p_1^x, p_2^x, \cdots, p_c^x\}) - \min(\{p_1^x, p_2^x, \cdots, p_c^x\}), \tag{11}$$

$$W = \max(\{p_1^y, p_2^y, \cdots, p_c^y\}) - \min(\{p_1^y, p_2^y, \cdots, p_c^y\}), \tag{12}$$

$$C_x = \min(\{p_1^x, p_2^x, \cdots, p_c^x\}) + L/2, \tag{13}$$

$$C_y = \min(\{p_1^y, p_2^y, \cdots, p_c^y\}) + W/2. \tag{14}$$

To illustrate, considering the original random noise scale as $\{\Delta x, \Delta y, \Delta l, \Delta w\}$, then the final dynamic noise is represented as $\{\Delta x^{'} = R_{decay} \cdot \Delta x, \Delta y^{'} = R_{decay} \cdot \Delta y, \Delta l^{'} = R_{decay} \cdot \Delta l, \Delta w^{'} = R_{decay} \cdot \Delta w\}$, which means that the larger the deviation between the ground truth of the past frame and the ground truth of the current frame, the greater the decay of the original random noise. The final noising point set $\{(\hat{p}_1^x, \hat{p}_1^y), (\hat{p}_2^x, \hat{p}_2^y), \cdots, (\hat{p}_c^x, \hat{p}_c^y)\}$ can be calculated as:

$$\hat{p}_i^x = \frac{L + \Delta l^{'}}{L}(p_i^x - C_x) + \Delta x^{'}, \tag{15}$$

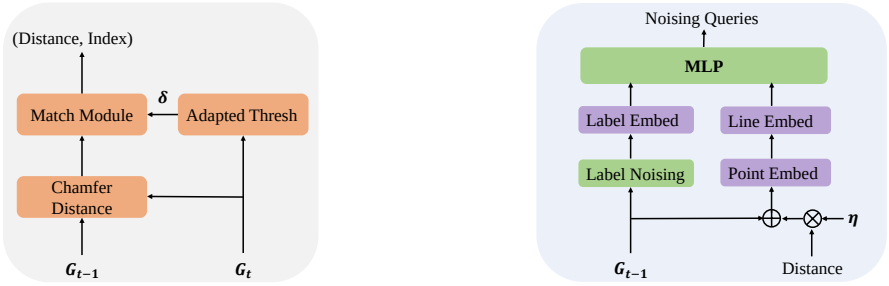
$$\hat{p}_i^y = \frac{W + \Delta w^{'}}{W}(p_i^y - C_y) + \Delta y^{'}. \tag{16}$$

Given a noising curve, its category and point set are denoted as cls and $\{(\hat{p}_1^x, \hat{p}_1^y), (\hat{p}_2^x, \hat{p}_2^y), \cdots, (\hat{p}_c^x, \hat{p}_c^y)\}$, where c is the number of points forming the curve. A learnable embedding is set for each category and then we can acquire the label embedding $C_q \in \mathbb{R}^{\frac{D}{2}}$, where D is the dimension of decoder embedding. As is shown in Fig. 5b, for the i-th point $(\hat{p}_i^x, \hat{p}_i^y)$, the point embedding can be generated by

$$P_i = \text{MLP}(\text{Concat}(\text{PE}(\hat{p}_i^x), \text{PE}(\hat{p}_i^y))). \tag{17}$$

where the positional encoding function PE maps a float to a vector with $\frac{D}{4}$ dimensions as: PE: $\mathbb{R} \to \mathbb{R}^{\frac{D}{4}}$, and the function MLP projects a $\frac{D}{2}$ dimensional vector into $\frac{D}{2}$ dimensions: MLP: $\mathbb{R}^{\frac{D}{2}} \to \mathbb{R}^{\frac{D}{2}}$. Then the line embedding of the curve can be obtained as

$$Pos_q = \text{MLP}(\text{Concat}(P_1, P_2, \cdots, P_c)), \tag{18}$$



(a) Adaptive Temporal Matching **(b)** Dynamic Curve Noising

Fig. 5: Illustration of Adaptive Temporal Matching and Dynamic Curve Noising

where MLP fuses the information from all the points into the position information of the curve.

Since we use deformable attention to interact the queries with the reference points for information, we acquire the noised query Q_{noise} by fusing the content information with the position information via MLP: $\mathbb{R}^D \rightarrow \mathbb{R}^D$ as

$$Q_{noise} = \text{MLP}(\text{Concat}(C_q, Pos_q)). \tag{19}$$

Objective Function Our model adopts an end-to-end training approach. For the predictions of map queries, we employ the same map loss function as baseline methods, mainly including the classification loss $\mathcal{L}_{Focal}$, the line loss $\mathcal{L}_{line}$.

$$\mathcal{L}_{map} = \lambda_1 \mathcal{L}_{Focal} + \lambda_2 \mathcal{L}_{line} \tag{20}$$

where λ_1 and λ_2 are the default hyper-parameters. Note that for the temporal methods, like StreamMapNet [37], the translation loss $\mathcal{L}_{trans}$ is also included.

Additionally, for the prediction results of noised queries, the correspondences between predictions and ground truth are obtained by ATM module. Then we use the same type of classification loss and line loss to construct $\mathcal{L}_{denoise}$.

$$\mathcal{L}_{denoise} = \lambda_1 \mathcal{L}^{DN}_{Focal} + \lambda_2 \mathcal{L}^{DN}_{line}, \tag{21}$$

where $\mathcal{L}^{DN}_{Focal}$ and $\mathcal{L}^{DN}_{line}$ are classification loss and the line loss of the denoising predictions. Finally, the overall loss is defined as:

$$\mathcal{L}_{train} = \mathcal{L}_{map} + \mathcal{L}_{denoise}. \tag{22}$$

5 Experiments

5.1 Experimental Settings

Datasets. We evaluate the SQD on two competitive and large-scale datase-ts, *i.e.*, nuScenes [3] and Argoverse2 [34]. The nuScenes dataset is annotated with

Table 1: The effectiveness of SQD on different methods.

Method	Backbone	Image Size	Epoch	AP_{ped}	AP_{div}	AP_{bound}	mAP
MapTR [19]	R50	480 × 800	24	46.3	51.5	53.1	50.3
+ SQD	R50	480 × 800	24	**47.2**	**53.9**	**55.6**	**52.2**
BeMapNet [28]	R50	512 × 896	24	57.7	62.3	59.4	59.8
+ SQD	R50	512 × 896	24	**59.1**	**64.0**	**62.5**	**61.9**
StreamMapNet [37]	R50	480 × 800	24	60.4	61.9	58.9	60.4
+ SQD	R50	480 × 800	24	**63.0**	**65.5**	**63.3**	**63.9**

2 Hz and each sample comprises 6 synchronized cameras. The Argoverse2 is annotated with 10 Hz. Each frame contains 7 ring cameras and 2 stereo cameras. We adopt images from the ring cameras only and unify the frame rate of the dataset to 2 Hz following the implementation in [8, 37].

Evaluation Metrics. For the sake of fair comparison, we focus on three static map categories, namely *lane-divider, ped-crossing*, and *road-boundary*. We evaluate the models on both small perceptual range ($30m$ front and back, $15m$ left and right) and larger perceptual range ($50m$ front and back, $25m$ left and right). The distinct thresholds to calculate the AP is set to $\{0.5m, 1.0m, 1.5m\}$ for the $30m$ range, and $\{1.0m, 1.5m, 2.0m\}$ for the $50m$ range.

Implementation Details. We adopt ResNet-50 [12] as backbones and use BEVFormer [18] with a single encoder layer for BEV feature extraction. The sizes of the BEV feature map are 100×50 m for the small perceptual range and 200×100 m for the larger range. The loss weight λ_1 and λ_2 are set to 4.0 and 50.0, respectively. During the single-frame training phase, we adopt the normal query denoising instead of stream query denoising. All models are trained for 24 epochs on the nuScenes dataset and 30 epochs on the Argoverse 2 dataset. We adopt AdamW optimizer [25] with a learning rate of 5×10^{-4}. All experiments are conducted on 8 NVIDIA Telsa V100 GPUs with a batch size of 32. More implementation details can be found in the *supplementary material*.

5.2 Effectiveness of Stream Query Denoising

Since our SQD is a general strategy for both static and temporal methods, we conduct the experiments on three typical frameworks to verify the effectiveness of SQD. As shown in Tab. 1, we first implement the SQD on some static methods. Specifically, SQD can bring gains of 1.9 mAP and 2.1 mAP to MapTR [19] and BeMapNet [28], respectively. When the SQD strategy is applied to the temporal method StreamMapNet [37], it can bring a overall gain of 3.5 mAP. It shows that it brings more gains compared to static methods. The denoising process of SQD mainly focus on the learning of temporal consistency of map elements, which greatly benefits to temporal approaches.

Table 2: Comparision with SOTAs on nuScenes [3] at 30 m range. The † indicates that we have added tricks to align with experimental setups of StreamMapNet [37]. The - means that no corresponding results were reported in the original paper. * indicates the backbone is pre-trained on DD3D [26].

Method	Backbone	Image Size	Epoch	AP_{ped}	AP_{div}	AP_{bound}	mAP
VectorMapNet [22]	R50	256 × 480	110	36.1	47.3	39.3	40.9
MapTR [19]	R50	480 × 800	24	46.3	51.5	53.1	50.3
BeMapNet [28]	R50	512 × 896	30	57.7	62.3	59.4	59.8
PivotNet [8]	R50	-	24	56.2	56.5	60.1	57.6
StreamMapNet [37]	R50	480 × 800	24	60.4	61.9	58.9	60.4
SQD-MapNet (Ours)	R50	480 × 800	24	**63.0**	**65.5**	**63.3**	**63.9**
StreamMapNet† [37]	R50	480 × 800	24	-	-	-	62.9
SQD-MapNet† (Ours)	R50	480 × 800	24	**63.6**	**66.6**	**64.8**	**65.0**
SQD-MapNet (Ours)	R50	480 × 800	110	69.2	68.3	67.0	68.2
SQD-MapNet (Ours)	V2-99*	480 × 800	24	65.4	68.7	69.8	68.0
SQD-MapNet (Ours)	V2-99*	900 × 1600	24	66.0	68.5	71.0	68.5
SQD-MapNet (Ours)	V2-99*	900 × 1600	110	**74.2**	**72.3**	**75.6**	**74.0**

Table 3: Comparision with SOTAs on nuScenes [3] at 50 m range. We reproduce the results of MapTR [19] and StreamMapNet [37] with their public codes. * indicates the backbone is pre-trained on DD3D [26].

Method	Backbone	Image Size	Epoch	AP_{ped}	AP_{div}	AP_{bound}	mAP
MapTR [19]	R50	480 × 800	24	45.5	47.1	43.9	45.5
StreamMapNet [37]	R50	480 × 800	24	62.9	63.1	55.8	60.6
SQD-MapNet (Ours)	R50	480 × 800	24	**67.0**	**65.5**	**59.5**	**64.0**
SQD-MapNet (Ours)	V2-99*	900 × 1600	110	**75.5**	**74.9**	**75.2**	**75.2**

5.3 Comparisons with State-of-the-arts

Since StreamMapNet is a temporal method and its performance is state-of-the-art, we equip it with our proposed SQD strategy and rename the overall framework as SQD-MapNet for subsequent performance comparison.

Performance on nuScenes. We first compare the proposed SQD-MapNet with previous competitive vision-based counterparts on the nuScenes validation set for both 30m and 50m perception ranges. As shown in Tab. 2, SQD-MapNet outperforms existing approaches under the 30 m range setting by a significant margin. Specifically, SQD-MapNet achieves 63.9 mAP within only 24 epochs under the short-range evaluation settings, surpassing the previous state-of-the-art method, StreamMapNet, by more than 3.0 mAP. Notably, armed with the strong V2-99 [14] backbone pretrained on DD3D [26], our SQD-MapNet achieves 68.0 and 74.0 mAP at 24 epochs and 110 epochs respectively, setting new state-of-the-art for the competitive nuScenes benchmark. At the same time, the results under the 50 m range setting are shown in Tab. 3. According to Tab. 3, SQD-

Table 4: Performance comparison of various methods on Argoverse 2 [34] at 30 m range. - means that no corresponding results were reported in the original paper Since PivotNet [8] trains the model on the full training set, we reimplement it by training with the same number of iterations for a fair comparison.

Method	Backbone	Image Size	Epoch	AP_{ped}	AP_{div}	AP_{bound}	mAP
HDMapNet [17]	Effi-B0	-	-	13.1	5.7	37.6	18.8
VectorMapNet [22]	R50	-	-	38.3	36.1	39.2	37.9
PivotNet [8]	R50	-	6	31.3	47.5	43.4	40.7
StreamMapNet [37]	R50	608 × 608	30	62.0	59.5	63.0	61.5
SQD-MapNet (Ours)	R50	608 × 608	30	**64.9**	**60.2**	**64.9**	**63.3**

Table 5: Performance comparison of various methods on Argoverse 2 [34] at 50 m range. The results of VectorMapNet [22] and MapTR [19] are directly borrowed from StreamMapNet [37].

Method	Backbone	Image Size	Epoch	mAP
VectorMapNet [22]	R50	384 × 384	120	30.2
MapTR [19]	R50	608 × 608	30	47.5
StreamMapNet [37]	R50	608 × 608	30	57.7
SQD-MapNet (Ours)	R50	608 × 608	30	**59.3**

MapNet achieves 64.0 mAP, outperforming other methods. We also provide the results on the new split of the dataset adopted in StreamMapNet [37] in the *supplementary material*.

Performance on Argoverse2. In addition to the widely-adopted nuScenes dataset, we also gauge SQD-MapNet on the large-scale dataset Argoverse2 [34] to further validate the effectiveness of our approach. To keep consistent with the evaluation protocol in nuScenes dataset, we report the results for both 60 ×30 m and 100 × 50 m ranges. As shown in Tab. 4 and Tab. 5, SQD-MapNet consistently outperforms StreamMapNet by about 2.0 mAP under the 30 m range and the 50 m range, validating the generalization and superiority of our approach. For the new split of the dataset adopted by StreamMapNet [37], we additionally show the results in the *supplementary material*.

5.4 Ablation Study

In this section, we provide extensive ablation studies to explore the effectiveness of the main components in SQD-MapNet, providing a deeper understanding of our approach. If not specified, all experiments are conducted on nuScenes dataset at a perceptual range of 60 × 30 m.

Main Ablations To understand how each component contributes to the final performance, we subsequently add the proposed modules to our baseline and report the performance in Tab. 6. First, to demonstrate the effectiveness of the SQD strategy, we remove the SQD strategy from SQD-MapNet, which

Table 6: Ablations on each component in SQD-MapNet. ATM denotes Adaptive Temporal Matching and DCN denotes Dynamic Curve Noising.

Method	AP_{ped}	AP_{div}	AP_{bound}	mAP
w/o SQD	57.4	61.8	58.3	59.2
w/o ATM	61.4	63.1	**64.3**	62.9
w/o DCN	60.7	62.9	61.0	61.5
SQD-MapNet	**63.0**	**65.5**	63.3	**63.9**

Table 7: (a) Ablations on matching scale in Adaptive Temporal Matching. (b) Effectiveness of different decay rates of noise in Dynamic Curve Noising.

Scale	ATM	mAP
Fixed δ	✗	62.9
$\alpha = 0.05$	✓	63.0
$\alpha = 0.1$	✓	**63.5**
$\alpha = 0.2$	✓	62.9
$\alpha = 0.3$	✓	63.0

Decay Rate	AP_{ped}	AP_{div}	AP_{bound}	mAP
$\gamma = 0.0$	61.8	64.1	62.8	62.9
$\gamma = 0.2$	**63.0**	65.5	63.3	**63.9**
$\gamma = 0.3$	59.0	64.1	**63.8**	62.3
$\gamma = 0.5$	59.7	**65.8**	63.7	63.0
$\gamma = 0.7$	61.7	64.0	61.3	62.4

is denoted as w/o SQD. Next, to verify the effectiveness of Adaptive Temporal Matching (ATM), we replace the ATM with a fixed threshold matching, which is denoted as w/o ATM. Finally, to verify the effectiveness of Dynamic Curve Noising (DCN), we remove DCN from SQD-MapNet, and this is denoted as w/o DCN. As shown in Tab. 6, the SQD strategy can yield a performance enhancement of 4.7 mAP. More specifically, ATM obtains a gain of 1.0 mAP and DCN improves the performance by a large margin (2.4 mAP).

Matching scale for Adaptive Temporal Matching Tab. 7 ablates the performance of different tolerance degrees of matching on SQD-MapNet. Concretely, α denotes the value of the tolerance degree, which is defined in Eq. (9). Due to the ignorance of the curve's properties, the strategy of a fixed threshold only achieves 62.8 mAP. When the tolerance degree α is small, the result is not optimal, suggesting that strict matching may cause some ground truths to be filtered out during the denoising process. As the value of α increases, which means the tolerance level of transformation bias becomes larger, more positive samples of the previous frame can be matched. Specifically, the performance reaches 63.5 mAP when α equals 0.1. However, when α keeps increasing, the detection accuracy starts to incline, indicating that there are plenty of noisy samples from the previous frames incorrectly matched with ground truths at the current frame.

Decay Rate of Noise After transforming the curves of the previous frame to the current frame, there is natural noise between the previous ground truth and the current ones. On this basis, we add extra noise during the dynamic curve noising. Therefore, deciding the right level of decay rates of the added noise is non-trivial. We experiment SQD-MapNet with different values of γ defined in Tab. 7. From the table, we can conclude that a small ratio of 0.2 can balance both the learning diversity and the negative effect introduced by noise.

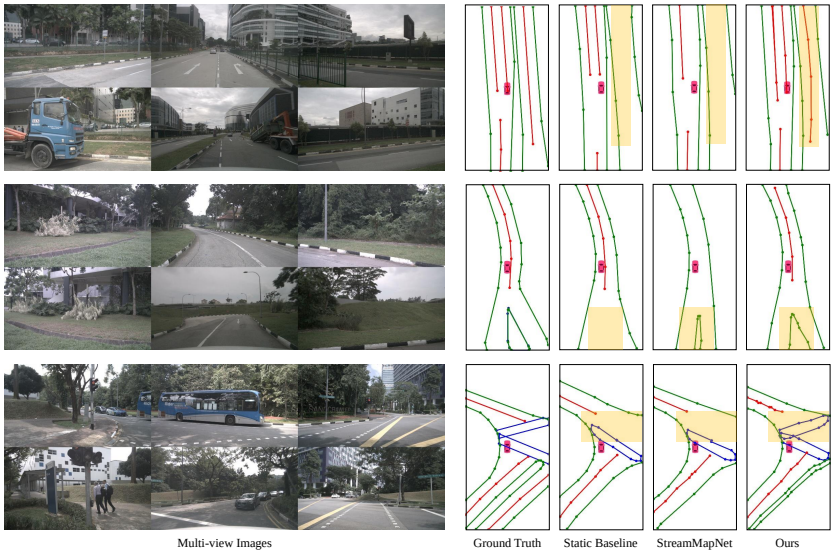


Fig. 6: Comparison with the static baseline and StreamMapNet [37] on qualitative visualization under different scenarios. In HD-map, green lines denote *road boundaries*, red lines indicate *lane-dividers*, and blue lines denote *pedestrian crossings*.

5.5 Qualitative Analysis

We show some qualitative comparisons with the static baseline, StreamMapNet [37], and SQD-MapNet in Fig. 6. The static baseline is reproduced with StreamMapNet without temporal propagation. We can find that the static baseline and StreamMapNet easily fail to recognize some curves in both simple and complex scenes. Compared to other methods, SQD-MapNet achieves the temporal consistency effectively, and accurately recognizes curves.

6 Conclusion

In this paper, we introduce stream query denoising (SQD) strategy to learn the temporal consistency of map elements for HD-map construction. Such strategy is only applied during the training process and can be removed for inference. Extensive experiments on nuScenes and Argoverse2 show that the performance of both static and temporal models are greatly improved with the proposed SQD strategy. The resulting SQD-MapNet framework is remarkably superior to other existing methods across all settings of close range and long range.

Acknowledgements The work was supported by National Science and Technology Major Project of China (2023ZD0121300). Also, this work was supported by the Anhui Provincial Natural Science Foundation under Grant 2108085UD12. We acknowledge the support of GPU cluster built by MCC Lab of Information Science and Technology Institution, USTC.

References

1. Antonello, M., Carraro, M., Pierobon, M., Menegatti, E.: Fast and robust detection of fallen people from a mobile robot. In: 2017 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 4159–4166. IEEE (2017)
2. Bekir, E.: Introduction to modern navigation systems. World scientific (2007)
3. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
4. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
5. Chang, J., Wang, S., Xu, H.M., Chen, Z., Yang, C., Zhao, F.: Detrdistill: A universal knowledge distillation framework for detr-families. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6898–6908 (2023)
6. Chen, Z., Li, Z., Wang, S., Fu, D., Zhao, F.: Learning from noisy data for semi-supervised 3d object detection. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6929–6939 (2023)
7. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
8. Ding, W., Qiao, L., Qiu, X., Zhang, C.: Pivotnet: Vectorized pivot learning for end-to-end hd map construction. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3672–3682 (2023)
9. Fu, T., Wang, X., Yu, H., Niu, K., Li, B., Xue, X.: Denoising-mot: Towards multiple object tracking with severe occlusions. In: Proceedings of the 31st ACM International Conference on Multimedia. pp. 2734–2743 (2023)
10. Han, C., Sun, J., Ge, Z., Yang, J., Dong, R., Zhou, H., Mao, W., Peng, Y., Zhang, X.: Exploring recurrent long-term temporal fusion for multi-view 3d perception. arXiv preprint arXiv:2303.05970 (2023)
11. Hasan, A.M., Samsudin, K., Ramli, A.R., Azmir, R., Ismaeel, S.: A review of navigation systems (integration and algorithms). Australian journal of basic and applied sciences **3**(2), 943–959 (2009)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Huang, J., Huang, G.: Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. arXiv preprint arXiv:2203.17054 (2022)
14. Lee, Y., Hwang, J.w., Lee, S., Bae, Y., Park, J.: An energy and gpu-computation efficient backbone network for real-time object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. pp. 0–0 (2019)
15. Li, F., Zhang, H., Liu, S., Guo, J., Ni, L.M., Zhang, L.: Dn-detr: Accelerate detr training by introducing query denoising. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13619–13627 (2022)
16. Li, F., Zhang, H., Xu, H., Liu, S., Zhang, L., Ni, L.M., Shum, H.Y.: Mask dino: Towards a unified transformer-based framework for object detection and segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3041–3050 (2023)

17. Li, Q., Wang, Y., Wang, Y., Zhao, H.: Hdmapnet: An online hd map construction and evaluation framework. In: 2022 International Conference on Robotics and Automation (ICRA). pp. 4628–4634. IEEE (2022)
18. Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Qiao, Y., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In: European conference on computer vision. pp. 1–18. Springer (2022)
19. Liao, B., Chen, S., Wang, X., Cheng, T., Zhang, Q., Liu, W., Huang, C.: Maptr: Structured modeling and learning for online vectorized hd map construction. arXiv preprint arXiv:2208.14437 (2022)
20. Lin, X., Lin, T., Pei, Z., Huang, L., Su, Z.: Sparse4d v2: Recurrent temporal fusion with sparse model. arXiv preprint arXiv:2305.14018 (2023)
21. Liu, S., Li, F., Zhang, H., Yang, X., Qi, X., Su, H., Zhu, J., Zhang, L.: Dab-detr: Dynamic anchor boxes are better queries for detr. arXiv preprint arXiv:2201.12329 (2022)
22. Liu, Y., Yuan, T., Wang, Y., Wang, Y., Zhao, H.: Vectormapnet: End-to-end vectorized hd map learning. In: International Conference on Machine Learning. pp. 22352–22369. PMLR (2023)
23. Liu, Y., Wang, T., Zhang, X., Sun, J.: Petr: Position embedding transformation for multi-view 3d object detection. In: European Conference on Computer Vision. pp. 531–548. Springer (2022)
24. Liu, Y., Yan, J., Jia, F., Li, S., Gao, A., Wang, T., Zhang, X.: Petrv2: A unified framework for 3d perception from multi-camera images. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3262–3272 (2023)
25. Loshchilov, I., Hutter, F.: Fixing weight decay regularization in adam (2018)
26. Park, D., Ambrus, R., Guizilini, V., Li, J., Gaidon, A.: Is pseudo-lidar needed for monocular 3d object detection? In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3142–3152 (2021)
27. Peng, L., Chen, Z., Fu, Z., Liang, P., Cheng, E.: Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 5935–5943 (2023)
28. Qiao, L., Ding, W., Qiu, X., Zhang, C.: End-to-end vectorized hd-map construction with piecewise bezier curve. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13218–13228 (2023)
29. Qin, Z., Chen, J., Chen, C., Chen, X., Li, X.: Unifusion: Unified multi-view fusion transformer for spatial-temporal representation in bird's-eye-view. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 8690–8699 (2023)
30. Shan, T., Englot, B.: Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 4758–4765. IEEE (2018)
31. Wang, S., Liu, Y., Wang, T., Li, Y., Zhang, X.: Exploring object-centric temporal modeling for efficient multi-view 3d object detection. arXiv preprint arXiv:2303.11926 (2023)
32. Wang, S., Zhao, X., Xu, H.M., Chen, Z., Yu, D., Chang, J., Yang, Z., Zhao, F.: Towards domain generalization for multi-view 3d object detection in bird-eye-view. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13333–13342 (2023)
33. Wang, Y., Mao, Q., Zhu, H., Deng, J., Zhang, Y., Ji, J., Li, H., Zhang, Y.: Multi-modal 3d object detection in autonomous driving: a survey. International Journal of Computer Vision pp. 1–31 (2023)

34. Wilson, B., Qi, W., Agarwal, T., Lambert, J., Singh, J., Khandelwal, S., Pan, B., Kumar, R., Hartnett, A., Pontes, J.K., et al.: Argoverse 2: Next generation datasets for self-driving perception and forecasting. arXiv preprint arXiv:2301.00493 (2023)
35. Wu, D., Jia, F., Chang, J., Li, Z., Sun, J., Han, C., Li, S., Liu, Y., Ge, Z., Wang, T.: The 1st-place solution for cvpr 2023 openlane topology in autonomous driving challenge. arXiv preprint arXiv:2306.09590 (2023)
36. Yang, C., Chen, Y., Tian, H., Tao, C., Zhu, X., Zhang, Z., Huang, G., Li, H., Qiao, Y., Lu, L., et al.: Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17830–17839 (2023)
37. Yuan, T., Liu, Y., Wang, Y., Wang, Y., Zhao, H.: Streammapnet: Streaming mapping network for vectorized online hd map construction. arXiv preprint arXiv:2308.12570 (2023)
38. Yurtsever, E., Lambert, J., Carballo, A., Takeda, K.: A survey of autonomous driving: Common practices and emerging technologies. IEEE access **8**, 58443–58469 (2020)
39. Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L.M., Shum, H.Y.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection. arXiv preprint arXiv:2203.03605 (2022)
40. Zhang, J., Singh, S.: Loam: Lidar odometry and mapping in real-time. In: Robotics: Science and systems. vol. 2, pp. 1–9. Berkeley, CA (2014)
41. Zhang, Y., Zhu, Z., Zheng, W., Huang, J., Huang, G., Zhou, J., Lu, J.: Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. arXiv preprint arXiv:2205.09743 (2022)
42. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)