MAPLIGHTNING: ONLINE VECTORIZED HD MAP CONSTRUCTION WITH 1D MAP TOKENS

Anonymous authors
Paper under double-blind review

ABSTRACT

Online vectorized HD map construction is essential for scaling safe autonomous driving and requires accurate, real-time inference. Prior methods typically rely on dense bird's-eye-view (BEV) grids as the intermediate representation. We propose *MapLightning*, which replaces the dense BEV grid with a compact set of 1D learnable map tokens. To construct map tokens from image features, we choose self-attention over vanilla cross-attention because it enables joint interactions and contextual aggregation among image and map tokens. Our transformer-based mapper concatenates map and image tokens, applies full self-attention, discards the image tokens, and retains the updated map tokens for decoding. This design offers three advantages. First, our representation is efficient, using fewer tokens, consuming less memory, and running faster. Second, the lightweight design allows the map decoder to use full rather than deformable cross-attention for better global context. Third, unlike BEV-based methods, our network does not use camera projection parameters, making it robust to camera-extrinsic perturbations. MapLightning uses up to 16.7× fewer intermediate tokens than dense BEV-based methods and achieves state-of-the-art accuracy and efficiency on nuScenes and Argoverse 2. Its lightweight variant surpasses MapTRv2 by +10.1 mAP on nuScenes and +16.2 mAP on Argoverse 2, while delivering 1.73× faster inference (40+ FPS) with 53% less memory. We further show improvements on uncertainty-aware map construction and downstream trajectory prediction. Code and models will be released.

1 INTRODUCTION

High-definition (HD) maps provide centimeter-level vectorized representations of static road elements such as lane dividers, pedestrian crossings, and road boundaries. These elements encode road topology and driving constraints essential for autonomous navigation. Traditionally, HD maps are constructed offline using SLAM- or SfM-based pipelines (Zhang et al., 2014; Shan & Englot, 2018; Zhanabatyrova et al., 2024), requiring expensive annotation and maintenance.

Recently, online vectorized HD map construction (Liao et al., 2022; Yuan et al., 2024; Chen et al., 2024; Liao et al., 2025; Dong et al., 2025; Monninger et al., 2026) has gained popularity by building local maps at runtime from onboard sensors, reducing reliance on costly offline map construction and maintenance. Online vectorized HD map construction methods typically follow a three-stage pipeline: an image encoder extracts features from surround-view images, a bird's-eye-view (BEV) projection module such as LSS (Philion & Fidler, 2020) or BEVFormer (Li et al., 2024b) projects them into a dense BEV grid, and a DETR-style (Carion et al., 2020) decoder with deformable attention (Zhu et al., 2020) predicts map elements as ordered point sets.

The BEV-based design is effective for dense spatial perception tasks like semantic segmentation (Philion & Fidler, 2020; Zhou & Krähenbühl, 2022) and 3D occupancy prediction (Yu et al., 2023; Hou et al., 2024), where dense spatial prediction naturally aligns with the BEV grid. However, it is less suited for vectorized map prediction, where the output is a sparse set of vectorized map elements that occupy only **4.27%** of the rasterized perception range on nuScenes (Caesar et al., 2020) and **2.65%** on Argoverse 2 (Wilson et al., 2023).

In this paper, we propose to use 1D learnable tokens, which we call *map tokens*, as the intermediate scene representation. This design offers three key advantages. **(i)** Our representation uses fewer intermediate tokens, consumes less memory, and runs faster. **(ii)** The lightweight design allows the

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

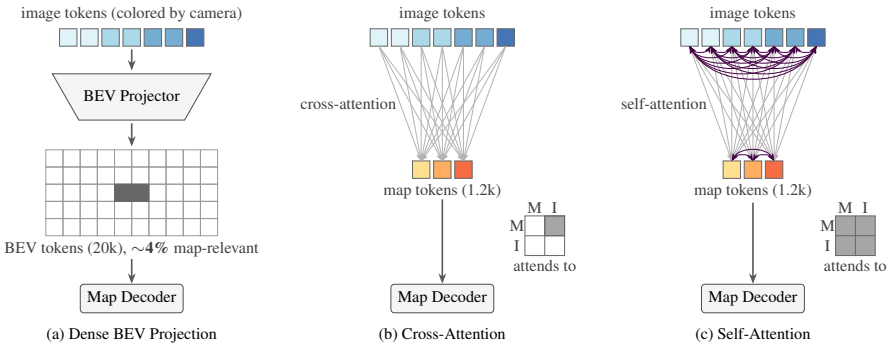


Figure 1: **Aggregating multi-view image tokens into a scene representation.** **(a)** BEV-based methods (Liao et al., 2022; 2025) geometrically project image features onto a dense grid of fixed BEV cells, few of which contain map elements. **(b)** Vanilla cross-attention compresses image tokens into compact map tokens, but image tokens serve only as fixed keys and values, with no image-to-image or map-to-map interaction. **(c)** Self-attention (ours, Section 3.3) attends over the concatenated map and image tokens, refining all tokens jointly. Gray and violet edges denote image–map and intra-group attention, respectively. In the masks, rows are queries, columns are keys/values, shaded cells mark allowed attention, and M/I denote map/image tokens. See Table 1 for comparison.

map decoder to use full attention instead of deformable attention, giving each query access to the entire representation for better global context and long-range dependencies. **(iii)** Unlike BEV-based methods, ours does not use camera projection parameters, making it robust to extrinsic perturbations.

Constructing map tokens requires compressing a large number of multi-view image tokens into a much smaller set, and the choice of compression mechanism is critical. Specifically, we concatenate map and image tokens and apply attention over the joint sequence. As shown in Figure 1, we adopt self-attention instead of vanilla cross-attention, since vanilla cross-attention only lets map tokens read from image tokens, whereas self-attention also allows explicit image-to-image and map-to-map token interactions for richer contextual aggregation. The image-to-image interactions are particularly useful for refining visual context: as illustrated in Figure 3, self-attention progressively concentrates on map-relevant structures across layers. After the final mapper layer, the image tokens are discarded, leaving compact map tokens that retain the aggregated map-relevant information. These map tokens serve as the intermediate representation for the decoder. In the decoder, cross-attention is used because the map queries act as output slots that read from the fixed map-token representation, similar to decoding objects (Carion et al., 2020) or categories (Strudel et al., 2021).

We name our framework *MapLightning* for its fast inference, token-efficient design, and high accuracy. By replacing the dense BEV grid with a compact set of tokens (up to **16.7**× fewer), MapLightning achieves state-of-the-art accuracy and efficiency on nuScenes and Argoverse 2. With a lightweight MobileNetV3-Large backbone, it surpasses MapTRv2 (Liao et al., 2025) by **+10.1** and **+16.2** mAP on nuScenes and Argoverse 2, running **1.73**× faster (**40+ FPS**) with **53%** less memory.

In summary, the main contributions of this paper are:

- We propose MapLightning, the first online vectorized HD map construction framework that replaces dense BEV projection with attention-based mapping into compact 1D map tokens, yielding higher efficiency, better global context, and robustness to extrinsic perturbations.
- We compare cross-attention and self-attention for constructing map tokens from image features, and show that self-attention is more effective for online vectorized HD map construction.
- MapLightning outperforms prior methods on online vectorized HD map construction, uncertainty-aware map construction, downstream trajectory prediction, and 3D object detection.

2 Related Work

2.1 Online Vectorized HD Map Construction

Most online vectorized HD map construction methods follow a MapTR-style (Liao et al., 2022) framework with dense BEV representations and a DETR-style decoder (Liu et al., 2023b; Liao

et al., 2022; 2025). Follow-up works refine the decoder, supervision, and temporal modeling within this paradigm (Yuan et al., 2024; Chen et al., 2024; Dong et al., 2025; Monninger et al., 2026).

Our approach differs from prior online vectorized HD map construction methods in two key aspects: **(i)** Instead of following the conventional dense BEV paradigm, we compress image tokens into compact 1D map tokens using attention, achieving higher efficiency and accuracy. **(ii)** Unlike many previous methods that focus on map construction alone, we show that our representation benefits downstream trajectory prediction and generalizes to 3D object detection.

2.2 Beyond BEV-Based Representation

BEV representations have been widely adopted for dense spatial perception. For example, Lift-Splat-Shoot (Philion & Fidler, 2020) and Cross-View Transformers (Zhou & Krähenbühl, 2022) construct BEV features for semantic segmentation, whereas FlashOcc (Yu et al., 2023) and FastOcc (Hou et al., 2024) extend this representation to 3D occupancy prediction.

Works on sparse prediction have explored removing BEV as an intermediate representation by aggregating image features into task queries. In 3D object detection, SparseBEV (Liu et al., 2023a) achieves performance comparable to dense-BEV methods like BEVFormer v2 (Yang et al., 2023). In online vectorized HD map construction, SparseMeXt (Jiang et al., 2025) follows a similar direction but underperforms dense-BEV methods like MapTRv2 (Liao et al., 2025) in comparable settings.

Unlike prior methods that either retain BEV-based feature projection or drop the intermediate representation, we use 1D learnable map tokens as a compact and performant intermediate representation.

2.3 Compressing Tokens via Attention Mechanisms

Learned query tokens are a common pattern for pooling a large set of input tokens via cross-attention. Queries can decode specific outputs, such as object predictions (Carion et al., 2020) or slots that explain an image (Locatello et al., 2020; Vani et al., 2024). Queries can also compress tokens into a smaller bottleneck for further processing, as in Perceiver's latents (Jaegle et al., 2021; 2022), Flamingo's resampler (Alayrac et al., 2022), and Q-Former (Li et al., 2023a).

Learned tokens can also aggregate information through self-attention by being concatenated with input tokens and jointly updated within the same sequence, which most closely resembles our design. Such tokens have been explored in general representation learning and sequence modeling, including the `[CLS]` token in BERT (Devlin et al., 2019), memory tokens (Burtsev et al., 2020), the `[CLS]` token in ViT (Dosovitskiy et al., 2021), gist tokens (Mu et al., 2023), summary vectors (Chevalier et al., 2023), register tokens in Registers (Darcet et al., 2024), and planning tokens in Flex (Yang et al., 2025a). They have also been used in dense or generative tasks, such as latent tokens in LVSM (Jin et al., 2024) and light tokens in RelitLRM (Zhang et al., 2025).

Unlike previous works targeting dense or general-purpose perception, we apply attention-based token compression to the sparse task of online vectorized HD map construction with a tailored mapper.

3 Method

Section 3.1 reviews the standard BEV-based approach to online vectorized HD map construction. We then describe our modifications: flat image-token embeddings (Section 3.2), a self-attention mapper that produces compact map tokens (Section 3.3), and the map decoder (Section 3.4). Figure 2 shows the architecture of MapLightning.

3.1 Background

In this subsection, we formulate the task and review the general setup of BEV-based methods.

Given surround-view images $\mathcal{I} = \{I_{t,c}\}_{t=1,c=1}^{T,C}$ from C cameras over T timesteps ($T = 1$ for single-frame methods such as MapTRv2 (Liao et al., 2025)), an image encoder f_{img}, e.g., ResNet (He et al., 2016) followed by a projection to d channels, extracts per-view features:

$$\mathbf{F}_{t,c} = f_{\text{img}}(I_{t,c}) \in \mathbb{R}^{H \times W \times d}. \tag{1}$$

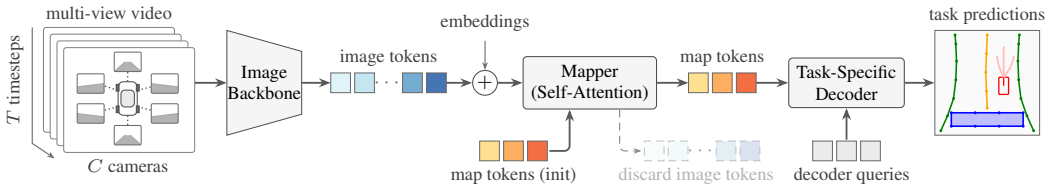


Figure 2: **Overview of *MapLightning*.** A shared image backbone encodes each of the C camera views over T timesteps into image tokens, which are flattened and augmented with learned positional, camera, and time embeddings. Instead of projecting these tokens into a dense BEV grid, we concatenate them with K learnable map tokens and jointly refine both with a self-attention mapper (Figure 1 (c)), without camera parameters or depth. The image tokens are then discarded, and the compact map tokens serve as the scene representation, which task-specific decoders attend to with full cross-attention. Beyond online vectorized HD map construction, this representation supports 3D object detection and downstream trajectory prediction from the predicted maps.

where H and W are the spatial dimensions of the feature map.

Existing BEV-based methods (Liao et al., 2022; 2025) then aggregate these features into a BEV grid using a projection module such as LSS (Philion & Fidler, 2020) or BEVFormer (Li et al., 2024b):

$$\mathbf{B} = f_{\text{BEV}}\left(\{\mathbf{F}_{t,c}\}_{t=1,c=1}^{T,C}\right) \in \mathbb{R}^{N_{\text{BEV}} \times d}, \tag{2}$$

where $N_{\text{BEV}} = H_{\text{BEV}} W_{\text{BEV}}$ and each token corresponds to a fixed BEV location.

A DETR-style (Carion et al., 2020) decoder then predicts M map elements, each with P ordered points, using queries $\mathbf{Q} \in \mathbb{R}^{M \times P \times d}$. In BEV-based methods, the queries retrieve information from $\mathbf{B}$ through deformable cross-attention:

$$\mathbf{Q}^{(l)} = \mathbf{Q}^{(l-1)} + \text{DeformAttn}\left(\mathbf{Q}^{(l-1)}, \mathbf{B}, \mathbf{R}^{(l-1)}\right), \tag{3}$$

where $\mathbf{R}^{(l-1)}$ denotes the BEV sampling reference points.

Each point query is regressed to normalized coordinates $(\hat{x}, \hat{y}) \in [0,1]^2$, which are then converted to metric positions (x, y) within the ego-centric perception range. Crucially, since map points are decoded from queries rather than tied to BEV cells, the dense 2D BEV grid is a design choice for the intermediate scene representation rather than an inherent requirement. This observation motivates our approach: we replace the BEV grid with a compact set of learned map tokens.

3.2 Constructing Flat Image-Token Embeddings

We first describe how we represent the image features without a BEV grid. Instead of lifting $\{\mathbf{F}_{t,c}\}$ into a dense BEV feature $\mathbf{B}$, we directly flatten them into $N = TCHW$ image tokens $X \in \mathbb{R}^{N \times d}$. Flattening, however, discards the spatial, camera, and temporal structure of the features, so we reintroduce it through learned embeddings. We encode within-image position with learnable row and column tables $E^{row} \in \mathbb{R}^{H \times d/2}$ and $E^{col} \in \mathbb{R}^{W \times d/2}$:

$$PE^{pos}_{h,w} = [E^{row}_h; E^{col}_w]. \tag{4}$$

We further add learnable camera and time embeddings $E^{cam} \in \mathbb{R}^{C \times d}$ and $E^{time} \in \mathbb{R}^{T \times d}$, which distinguish tokens from different views and timesteps. Each token $\mathbf{x}_{t,c,h,w}$ is thus embedded as

$$\widetilde{\mathbf{x}}_{t,c,h,w} = \mathbf{x}_{t,c,h,w} + E^{cam}_c + E^{time}_t + PE^{pos}_{h,w}. \tag{5}$$

The embedded tokens form $\mathbf{X}^{(0)} \in \mathbb{R}^{N \times d}$, which serves as the input to the mapper.

Unlike BEV-based methods (Liao et al., 2022; 2025), our mapper uses no camera intrinsics, extrinsics, depth, or explicit ego-pose alignment across timesteps. Instead, camera identity, time, and within-image position are represented solely by the learned embeddings, and the mapping from image space to the map is learned entirely from data.

3.3 Self-Attention Mapper

Our mapper replaces f_{BEV}. Following prior work using learnable latent tokens to summarize dense inputs (Jin et al., 2024; Zhang et al., 2025), we concatenate $\mathbf{X}^{(0)}$ with K learnable map tokens $\mathbf{U}^{(0)} \in \mathbb{R}^{K\times d}$, where (0) denotes the mapper input:

$$\mathbf{Z}^{(0)} = \text{Concat}\left(\mathbf{U}^{(0)}, \mathbf{X}^{(0)}\right) \in \mathbb{R}^{(K+N)\times d}. \tag{6}$$

Map tokens provide a compact scene representation. We use $T = 4$ (current and three past timesteps) and $K = \gamma CT$, with $\gamma = 50$ by default, so $K = 1{,}200$ for nuScenes and $1{,}400$ for Argoverse 2.

Rather than updating map tokens via one-way cross-attention (Jaegle et al., 2021), we jointly refine image and map tokens with self-attention, so image tokens also exchange information across views and timesteps instead of serving as fixed keys and values. Our mapper is an L-layer ($L = 4$) Pre-LN Transformer (Xiong et al., 2020):

$$\widetilde{\mathbf{Z}}^{(l)} = \mathbf{Z}^{(l-1)} + \text{MSA}^{(l)}\big(\text{LN}_1^{(l)}(\mathbf{Z}^{(l-1)})\big), \tag{7}$$

$$\mathbf{Z}^{(l)} = \widetilde{\mathbf{Z}}^{(l)} + \text{FFN}^{(l)}\big(\text{LN}_2^{(l)}(\widetilde{\mathbf{Z}}^{(l)})\big), \qquad l = 1, \ldots, L, \tag{8}$$

where MSA, LN, and FFN denote multi-head self-attention, LayerNorm (Ba et al., 2016), and the feed-forward network, respectively, with

$$\text{FFN}(\mathbf{z}) = \mathbf{W}_2\,\text{GELU}\,(\mathbf{W}_1\mathbf{z} + \mathbf{b}_1) + \mathbf{b}_2. \tag{9}$$

After the final layer, we split the output back into map and image tokens:

$$\mathbf{Z}^{(L)} = \text{Concat}\left(\mathbf{U}^{(L)}, \mathbf{X}^{(L)}\right). \tag{10}$$

Since the map tokens serve as the compact scene representation and retaining image tokens ($N \gg K$) would be costly for decoding, we discard $\mathbf{X}^{(L)}$ and retain only the normalized map tokens:

$$\mathbf{U} = \text{LN}_{\text{out}}\big(\mathbf{U}^{(L)}\big) \in \mathbb{R}^{K\times d}, \tag{11}$$

where LN_{out} is applied independently over each token's d feature dimensions.

3.4 Map Decoder

We retain the decoder architecture of MapTRv2 (Liao et al., 2025), including its $L_d = 6$ layers. We change what the decoder attends to: the compact map tokens $\mathbf{U} \in \mathbb{R}^{K\times d}$ instead of the dense BEV features $\mathbf{B} \in \mathbb{R}^{N_{\text{BEV}}\times d}$. This change affects the decoder in two ways. First, since $\mathbf{U}$ has no fixed 2D coordinates, deformable attention, which samples features around BEV reference points, no longer applies. Second, since $K \ll N_{\text{BEV}}$ (1,200 map tokens versus 20,000 BEV tokens on nuScenes, **16.7**$\times$ fewer), full cross-attention becomes feasible, letting each query attend to all map tokens.

Following MapTRv2 (Liao et al., 2025), each decoder layer first applies decoupled self-attention to the queries along the instance and point dimensions, yielding $\overline{\mathbf{Q}}^{(l)}$. We then replace deformable cross-attention with full cross-attention to the map tokens:

$$\widetilde{\mathbf{Q}}^{(l)} = \overline{\mathbf{Q}}^{(l)} + \text{MHA}\big(\overline{\mathbf{Q}}^{(l)}, \mathbf{U}, \mathbf{U}\big), \qquad l = 1, \ldots, L_d, \tag{12}$$

where MHA denotes multi-head attention with $\overline{\mathbf{Q}}^{(l)}$ as queries and $\mathbf{U}$ as keys and values.

Prediction head. We retain MapTRv2's iterative coordinate refinement. Initial reference points are predicted from the decoder queries via a linear projection and sigmoid. At each layer, the regression branch predicts offsets that refine the current reference points for the next layer, while the classification branch predicts the class of each map element. Unlike MapTRv2, these reference points serve only for coordinate regression, not feature sampling. The final normalized coordinates are converted to metric positions as described in Section 3.1, and predictions from all L_d layers are supervised.

Efficiency. Our mapper and decoder use standard attention and MLP operations, so all attention layers run with FlashAttention (Dao et al., 2022), whose memory scales linearly with sequence length for both self- and cross-attention. Compute is quadratic in $N + K$, but confined to the $L = 4$ mapper layers, where runtime is largely bound by memory access rather than FLOPs. Following prior work (Chen et al., 2024), we cache past-frame features, so only the current frame is encoded. Our model runs at 18.9 FPS on nuScenes (Table 2), faster than MapTRv2's dense BEV projection (14.1 FPS) and comparable to cross-attention (18.8 FPS). Breakdowns are in Appendix B.4.

4 EXPERIMENTS

In this section, we first introduce the dataset splits in Section 4.1, followed by the evaluation metrics and baseline methods in Section 4.2. We then evaluate on online vectorized HD map construction in Section 4.3, followed by uncertainty-aware map construction and its downstream trajectory prediction performance in Section 4.4. The appendix provides additional qualitative comparisons (Appendix A.1), map construction results under different data splits (Appendix B.1), real-time variants (Appendix B.2), 3D object detection (Appendix B.3), ablation studies (Appendix C), data split visualizations (Appendix D), robustness to camera extrinsic perturbations (Appendix E), and implementation details (Appendix F). **Video results are provided in the supplementary material.**

4.1 DATASETS AND SPLITS

Datasets. We conduct experiments on nuScenes (Caesar et al., 2020) and Argoverse 2 (Wilson et al., 2023), using images from 6 and 7 cameras, respectively, together with the corresponding ground truth vectorized map elements. We use the same key frames as MapTRv2 (Liao et al., 2025). The map classes are road boundaries, lane dividers, and pedestrian crossings.

Dataset Splits. To rigorously evaluate generalization and avoid geographic data leakage, we adopt the **Near-Extrapolation split (Lilja et al., 2024) with a 60m×30m perception range as our default benchmark** for both nuScenes and Argoverse 2, following recent protocols (Löwens et al., 2025; Lyu et al., 2025). The original splits of nuScenes and Argoverse 2 suffer from severe train–val overlap (79.4% and 45.0%, respectively). While the StreamMapNet split (Yuan et al., 2024) reduces nuScenes overlap to 2.1%, the Near-Extrapolation split minimizes it to 0.9% (within 5 m) for nuScenes (see Appendix D for visualizations) and 0% for Argoverse 2. For completeness, we also report improvements on the original and StreamMapNet splits in Appendix B.1. For 3D object detection, we follow standard practice and use the original nuScenes split, since geographic overlap is less problematic for dynamic objects than for static HD map elements that persist across traversals.

Table 1: **Comparison of aggregation mechanisms under a common decoder framework in the single-frame setting ($T = 1$).** Self-attention achieves the best accuracy with **66.7**× fewer intermediate tokens than dense BEV projection (Liao et al., 2025), and matches cross-attention in token count and FPS, demonstrating effective token compression and interaction. Token counts (0.3k) are for nuScenes; Argoverse 2 uses 0.35k tokens for 7 cameras (same per-camera budget).

Aggregation	#Tokens (k)↓	FPS↑	nuSc. mAP↑	Argo2 mAP↑
ResNet-50 backbone				
BEV Projection	20.0	14.1	26.7	53.1
Ours (cross-attn)	0.3	19.1	25.5	52.6
Ours (self-attn)	**0.3**	**19.2**	**30.1**	**58.3**
MobileNetV3-Large backbone				
BEV Projection	20.0	23.3	20.5	43.7
Ours (cross-attn)	0.3	41.2	21.1	50.0
Ours (self-attn)	**0.3**	**41.6**	**27.6**	**56.5**
ResNet-18 backbone				
BEV Projection	20.0	24.2	23.0	46.7
Ours (cross-attn)	0.3	44.2	21.2	48.1
Ours (self-attn)	**0.3**	**44.5**	**26.9**	**55.1**

4.2 METRICS AND BASELINES

Metrics. We report mean Average Precision (mAP) based on Chamfer distance, with thresholds of $\{0.5, 1.0, 1.5\}$ m. Following prior work (Liao et al., 2022; 2025), runtime **for all methods** is reported in frames per second (FPS) on an NVIDIA GeForce RTX 3090 for a fair comparison. Best results overall are **bolded**, and the best results among prior methods are underlined.

Baselines. For online vectorized HD map construction, we benchmark against methods with public codebases, including MapTR (Liao et al., 2022), StreamMapNet (Yuan et al., 2024), MapTracker (Chen et al., 2024), MapTRv2 (Liao et al., 2025), DAMap (Dong et al., 2025), and AugMapNet (Monninger et al., 2026), and retrain each on the Near-Extrapolation split using its official code. VectorMapNet (Liu et al., 2023b) results are taken from (Liu et al., 2023b; Lilja et al., 2024) because no official implementation is available. SparseMeXt (Jiang et al., 2025), the closest prior work to ours in removing dense BEV, is excluded from our main comparisons because its code is unavailable and it reports no results on the Near-Extrapolation or StreamMapNet splits; we list its original-split result (100 epochs) in Appendix B.1 for reference. For 3D object detection, we report baseline results from their respective papers. We use **ResNet-50 (He et al., 2016) as the default image backbone** for both map construction and object detection.

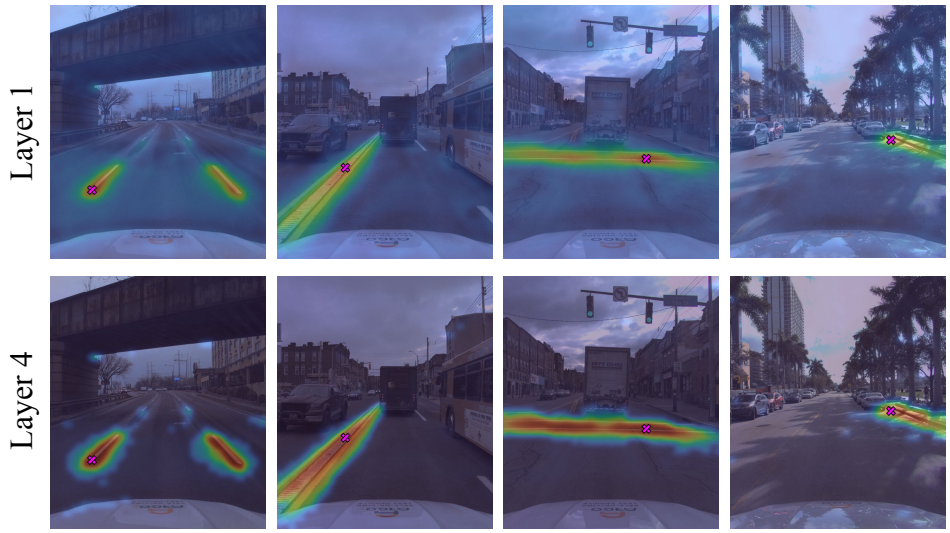


Figure 3: **Image-to-image attention in the self-attention mapper.** We select one image token as the query (magenta cross) and visualize its attention scores over the other image tokens, which serve as keys (red: high, blue: low), at the first and last ($L = 4$) mapper layers. From left to right: (1) a query on a lane divider attends along that divider and to the parallel divider across the lane; (2) a query on a lane divider attends along the divider's full extent toward the horizon; (3) a query on a pedestrian crossing attends across the crossing's full width; and (4) a query on a road boundary attends along the curb. From Layer 1 to Layer 4, responses on these structures become stronger and more complete, while diffuse background responses (e.g., the sky in column 4) fade, illustrating progressive refinement of the image tokens. Such image-to-image interaction is absent in cross-attention, where image tokens serve only as fixed keys and values. Best viewed zoomed in.

Table 2: **Quantitative comparisons of online vectorized HD map construction.** Ours with self-attention outperforms both the cross-attention variant and previous state-of-the-art methods in accuracy on nuScenes and Argoverse 2, while achieving the highest inference speed. FPS is for nuScenes.

	nuScenes				Argoverse 2				
Method	AP_{div}	AP_{ped}	AP_{bound}	mAP↑	AP_{div}	AP_{ped}	AP_{bound}	mAP↑	FPS↑
MapTR (2022)	16.0	14.4	26.7	19.0	50.0	46.6	47.5	48.0	<u>15.1</u>
VectorMapNet (2023b)	13.5	13.7	14.9	14.0	39.8	26.8	31.5	32.7	2.2
StreamMapNet (2024)	23.0	25.8	29.5	26.1	52.7	49.4	50.0	50.7	12.5
MapTracker (2024)	20.0	<u>31.3</u>	31.1	<u>27.5</u>	57.8	51.4	48.8	52.7	13.0
MapTRv2 (2025)	20.9	26.5	32.6	26.7	<u>58.4</u>	49.7	<u>51.3</u>	<u>53.1</u>	14.1
DAMap (2025)	23.5	17.1	<u>35.9</u>	25.5	55.3	44.7	49.2	49.7	12.1
AugMapNet (2026)	<u>24.7</u>	28.0	29.9	<u>27.5</u>	51.9	<u>54.0</u>	50.4	52.1	11.2
Ours (cross-attention)	16.8	32.4	30.0	26.4	61.9	54.0	53.5	56.5	18.8
Ours (self-attention)	**28.3**	**41.0**	**36.7**	**35.3**	**68.2**	**63.9**	**57.9**	**63.3**	**18.9**

4.3 Online Vectorized HD Map Construction

Table 1 compares the three aggregation mechanisms in Figure 1 under a common DETR-style decoder framework in the single-frame setting ($T = 1$). With ResNet-50, our self-attention surpasses dense BEV projection by **+3.4** mAP on nuScenes and **+5.2** mAP on Argoverse 2, and cross-attention by **+4.6** mAP on nuScenes and **+5.7** mAP on Argoverse 2. Cross-attention even falls below the dense BEV baseline on both datasets (**-1.2** and **-0.5** mAP), indicating that the compression mechanism matters. All other experiments use our default setting of $T = 4$, unless otherwise specified.

Table 2 shows that ours with self-attention outperforms prior methods across every category on both datasets. Ours with self-attention surpasses AugMapNet and MapTracker by **+7.8** mAP on nuScenes and MapTRv2 by **+10.2** mAP on Argoverse 2. Additionally, our self-attention variant consistently outperforms the cross-attention variant on both datasets. Ours also achieves **18.9 FPS** on nuScenes. Qualitative comparisons are shown in Figure 4, with additional examples in Appendix A.1.

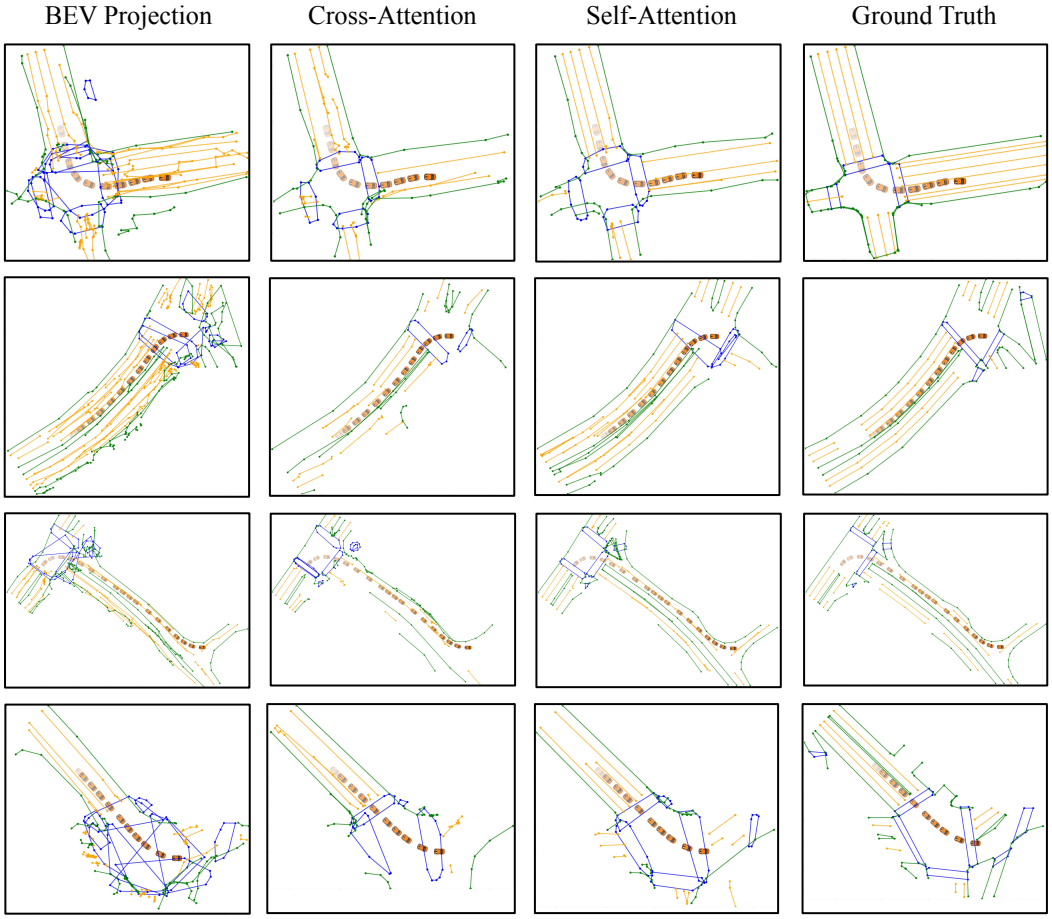


Figure 4: **Qualitative comparisons of online vectorized HD map construction** for Dense BEV Projection (Liao et al., 2025), ours (cross-attention), and ours (self-attention). Self-attention produces higher-quality and more consistent vectorized map reconstructions, with fewer spurious elements, more complete map geometries, and better-preserved intersection geometry. The first three rows are from nuScenes, while the last row is from Argoverse 2. Orange, blue, and green denote lane dividers, pedestrian crossings, and road boundaries, respectively. Best viewed zoomed in.

4.4 Uncertainty-Aware Map Construction and Trajectory Prediction

Table 3 shows that ours with self-attention surpasses MapTRv2 on uncertainty-aware map construction by **+14.4** mAP, and the cross-attention variant by **+9.0** mAP. Self-attention also improves downstream trajectory prediction across all three metrics (minADE, minFDE, and MR). Qualitative comparisons of uncertainty-aware map predictions are provided in Figure 5.

While MapUncertainty (Gu et al., 2024) reports negligible impact on the original split, adding the uncertainty head reduces mAP for all methods on the Near-Extrapolation split (Lilja et al., 2024), possibly because NLL down-weights gradients for poorly predicted, high-variance samples (Seitzer et al., 2022). Comparing Tables 2 and 3, our method degrades the least: **-3.0** and **-2.9** mAP for our cross- and self-attention variants, versus **-6.8**, **-15.2**, and **-8.7** mAP for MapTR, StreamMapNet, and MapTRv2. Although the uncertainty head slightly reduces our map construction accuracy, it enables uncertainty-aware downstream trajectory prediction (Zhou et al., 2022).

5 Conclusion

We proposed MapLightning, which replaces the dense 2D BEV grid with compact 1D learnable map tokens as the intermediate scene representation. Our mapper concatenates map and image tokens,

Table 3: **Quantitative comparisons of uncertainty-aware online vectorized HD map construction and downstream trajectory prediction on nuScenes.** Following MapUncertainty (Gu et al., 2024), we train each map construction model with an added uncertainty head, precompute its uncertainty-aware maps, and train a separate HiVT (Zhou et al., 2022) on them with identical settings for trajectory prediction. We include MapTR, StreamMapNet, and MapTRv2 because MapUncertainty provides uncertainty-head implementations for them, and we add the same uncertainty head to ours as for MapTRv2. Ours with self-attention outperforms the cross-attention variant and previous methods on map construction and downstream trajectory prediction.

	Map Construction				Trajectory Prediction		
Method	AP_{div}	AP_{ped}	AP_{bound}	mAP↑	minADE↓	minFDE↓	MR↓
MapTR (2022)	12.9	3.9	19.9	12.2	<u>0.432</u>	<u>0.952</u>	0.1291
StreamMapNet (2024)	9.5	<u>11.5</u>	11.8	10.9	0.440	0.981	0.1412
MapTRv2 (2025)	<u>16.2</u>	9.7	<u>28.3</u>	<u>18.0</u>	0.435	0.958	<u>0.1274</u>
Ours (cross-attention)	17.4	26.3	26.4	23.4	0.438	0.979	0.1385
Ours (self-attention)	**25.1**	**39.4**	**32.7**	**32.4**	**0.406**	**0.914**	**0.1167**

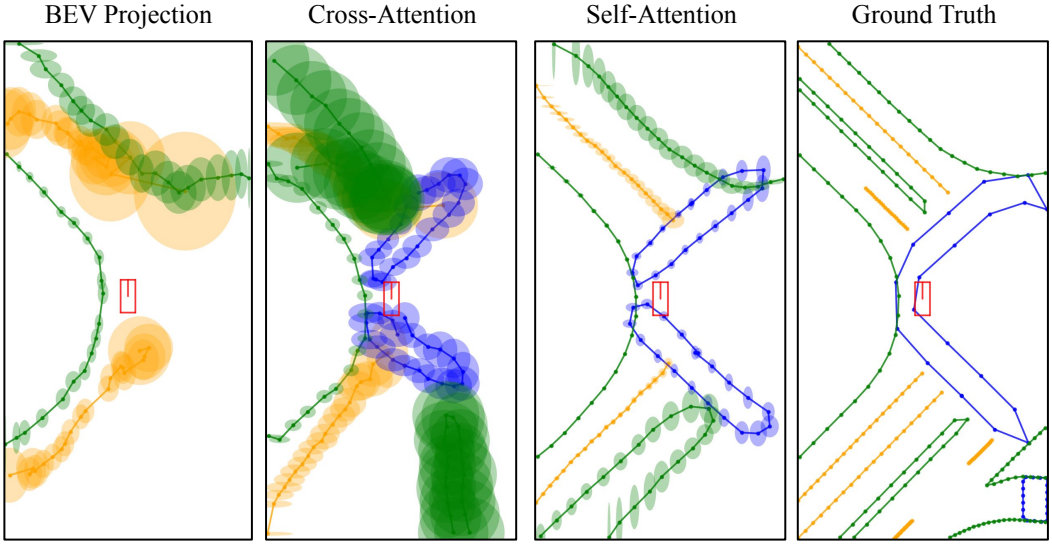


Figure 5: **Qualitative comparison of uncertainty-aware online vectorized HD map construction** for Dense BEV Projection (Liao et al., 2025), ours (cross-attention), and ours (self-attention) on nuScenes. Uncertainty is visualized at each predicted vector point as a semi-transparent ellipse whose size reflects the predicted Laplace variance. Self-attention produces more localized predictions with substantially less diffuse uncertainty, providing cleaner uncertainty-aware maps for downstream trajectory prediction. Orange, blue, and green denote lane dividers, pedestrian crossings, and road boundaries, respectively, while the red box marks the ego vehicle.

applies self-attention, discards the image tokens, and retains the updated map tokens for decoding. This representation uses far fewer tokens, consumes less memory, runs faster, enables full cross-attention for global context, and avoids dependence on camera projection parameters. MapLightning achieves state-of-the-art accuracy and efficiency on nuScenes and Argoverse 2, and generalizes to uncertainty-aware map construction, downstream trajectory prediction, and 3D object detection.

Limitations. **(i)** Replacing dense BEV features with compact 1D tokens may require additional adaptation in end-to-end frameworks, whose task decoders rely explicitly on the BEV grid structure (e.g., MapFormer and OccFormer in UniAD (Hu et al., 2023)). **(ii)** Our fixed token budget may require tuning for other tasks or datasets, and dynamic token selection (Rao et al., 2021; Fayyaz et al., 2022; Yin et al., 2022) is a promising direction for adapting the budget to input complexity.

AI Disclosure. We used generative AI to help polish writing, search for and identify related work, assist with research ideation, and edit source code. We have reviewed all AI-assisted work.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *NeurIPS*, 2022. 3

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

Mikhail S Burtsev, Yuri Kuratov, Anton Peganov, and Grigory V Sapunov. Memory transformer. *arXiv preprint arXiv:2006.11527*, 2020. 3

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 *CVPR*, 2020. 1, 6

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, 2020. 1, 2, 3, 4

Jiacheng Chen, Yuefan Wu, Jiaqi Tan, Hang Ma, and Yasutaka Furukawa. Maptracker: Tracking with strided memory fusion for consistent vector hd mapping. In *ECCV*, 2024. 1, 3, 5, 6, 7, 15, 18

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. Adapting language models to compress contexts. In *EMNLP*, 2023. 3

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *NeurIPS*, 2022. 5

Timothée Darcet, Maxime Oquab, Julien Mairal, and Piotr Bojanowski. Vision transformers need registers. In *ICLR*, 2024. 3

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 22

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019. 3

Jinpeng Dong, Chen Li, Yutong Lin, Jingwen Fu, Sanping Zhou, and Nanning Zheng. Damap: Distance-aware mapnet for high quality hd map construction. In *ICCV*, 2025. 1, 3, 6, 7, 15, 18

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 3

Huijie Fan, Pengrui Huang, Qiang Wang, Baojie Fan, Jiahua Dong, and Liangqiong Qu. Stur3d: Spatio-temporal unified representation learning for 3d object detection. In *CVPR*, 2026. 17, 22

Mohsen Fayyaz, Soroush Abbasi Koohpayegani, Farnoush Rezaei Jafari, Sunando Sengupta, Hamid Reza Vaezi Joze, Eric Sommerlade, Hamed Pirsiavash, and Jürgen Gall. Adaptive token sampling for efficient vision transformers. In *ECCV*, 2022. 9

Xunjiang Gu, Guanyu Song, Igor Gilitschenski, Marco Pavone, and Boris Ivanovic. Producing and leveraging online map uncertainty in trajectory prediction. In *CVPR*, 2024. 8, 9

Chunrui Han, Jinrong Yang, Jianjian Sun, Zheng Ge, Runpei Dong, Hongyu Zhou, Weixin Mao, Yuang Peng, and Xiangyu Zhang. Exploring recurrent long-term temporal fusion for multi-view 3d perception. *RA-L*, 2024. 17, 22

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3, 6

Jiawei Hou, Xiaoyan Li, Wenhao Guan, Gang Zhang, Di Feng, Yuheng Du, Xiangyang Xue, and Jian Pu. Fastocc: Accelerating 3d occupancy prediction by fusing the 2d bird's-eye view and perspective view. In *ICRA*, 2024. 1, 3

Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *ICCV*, 2019. 16

Yihan Hu, Jiazhi Yang, Li Chen, Keyu Li, Chonghao Sima, Xizhou Zhu, Siqi Chai, Senyao Du, Tianwei Lin, Wenhai Wang, et al. Planning-oriented autonomous driving. In *CVPR*, 2023. 9

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *ICML*, 2021. 3, 5

Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, et al. Perceiver io: A general architecture for structured inputs & outputs. In *ICLR*, 2022. 3

Anqing Jiang, Jinhao Chai, Yu Gao, Yiru Wang, Yuwen Heng, Zhigang Sun, Hao Sun, Zezhong Zhao, Li Sun, Jian Zhou, et al. Sparsemext: unlocking the potential of sparse representations for hd map construction. In *IROS*, 2025. 3, 6, 15

Haian Jin, Hanwen Jiang, Hao Tan, Kai Zhang, Sai Bi, Tianyuan Zhang, Fujun Luan, Noah Snavely, and Zexiang Xu. Lvsm: A large view synthesis model with minimal 3d inductive bias. *arXiv preprint arXiv:2410.17242*, 2024. 3, 5

Nayeon Kim, Hongje Seong, Daehyun Ji, and Sujin Jang. Unveiling the hidden: Online vectorized hd map construction with clip-level token interaction and propagation. *NeurIPS*, 2024. 15

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *ICML*, 2023a. 3

Yinhao Li, Zheng Ge, Guanyi Yu, Jinrong Yang, Zengran Wang, Yukang Shi, Jianjian Sun, and Zeming Li. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. In *AAAI*, 2023b. 17, 22

Zhenxin Li, Shiyi Lan, Jose M Alvarez, and Zuxuan Wu. Bevnext: Reviving dense bev frameworks for 3d object detection. In *CVPR*, 2024a. 17, 22

Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: learning bird's-eye-view representation from lidar-camera via spatiotemporal transformers. *TPAMI*, 2024b. 1, 4

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. *arXiv preprint arXiv:2208.14437*, 2022. 1, 2, 4, 6, 7, 9, 15, 18

Bencheng Liao, Shaoyu Chen, Yunchi Zhang, Bo Jiang, Qian Zhang, Wenyu Liu, Chang Huang, and Xinggang Wang. Maptrv2: An end-to-end framework for online vectorized hd map construction. *IJCV*, 2025. 1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 15, 16, 18, 21, 22

Adam Lilja, Junsheng Fu, Erik Stenborg, and Lars Hammarstrand. Localization is all you evaluate: Data leakage in online mapping datasets and how to fix it. In *CVPR*, 2024. 6, 8, 20

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 22

Xuewu Lin, Tianwei Lin, Zixiang Pei, Lichao Huang, and Zhizhong Su. Sparse4d v2: Recurrent temporal fusion with sparse model. *arXiv preprint arXiv:2305.14018*, 2023. 17, 22

Haisong Liu, Yao Teng, Tao Lu, Haiguang Wang, and Limin Wang. Sparsebev: High-performance sparse 3d object detection from multi-camera videos. In *ICCV*, 2023a. 3, 17, 22

Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *ICML*, 2023b. 2, 6, 7, 15, 18

Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf. Object-centric learning with slot attention. *NeurIPS*, 2020. 3

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 22

Christian Löwens, Thorben Funke, Jingchao Xie, and Alexandru Paul Condurache. Pseudomap-trainer: Learning online mapping without hd maps. In *ICCV*, 2025. 6

Hongyu Lyu, Thomas Monninger, Julie Stephany Berrio Perez, Mao Shan, Zhenxing Ming, and Stewart Worrall. Maprf: Weakly supervised online hd map construction via nerf-guided self-training. *arXiv preprint arXiv:2511.19527*, 2025. 6

Thomas Monninger, Md Zafar Anwar, Stanislaw Antol, Steffen Staab, and Sihao Ding. Augmapnet: Improving spatial latent structure via bev grid augmentation for enhanced vectorized online hd map construction. In *WACV*, 2026. 1, 3, 6, 7, 15, 18

Jesse Mu, Xiang Li, and Noah Goodman. Learning to compress prompts with gist tokens. *NeurIPS*, 2023. 3

Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*, 2020. 1, 3, 4

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *NeurIPS*, 2021. 9

Maximilian Seitzer, Arash Tavakoli, Dimitrije Antic, and Georg Martius. On the pitfalls of heteroscedastic uncertainty estimation with probabilistic neural networks. *arXiv preprint arXiv:2203.09168*, 2022. 8

Tixiao Shan and Brendan Englot. Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In *IROS*, 2018. 1

Jingyu Song, Xudong Chen, Liupei Lu, Jie Li, and Katherine A Skinner. Memfusionmap: Working memory fusion for online vectorized hd map construction. In *WACV*, 2025. 19

Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *ICCV*, 2021. 2

Haisheng Su, Junjie Zhang, Feixiang Song, Sanping Zhou, Wei Wu, Junchi Yan, and Nanning Zheng. Freqpde: Rethinking positional depth embedding for multi-view 3d object detection transformers. In *ICCV*, 2025. 17, 22

Ankit Vani, Bac Nguyen, Samuel Lavoie, Ranjay Krishna, and Aaron Courville. Sparo: Selective attention for robust and compositional transformer encodings for vision. In *ECCV*, 2024. 3

Shihao Wang, Yingfei Liu, Tiancai Wang, Ying Li, and Xiangyu Zhang. Exploring object-centric temporal modeling for efficient multi-view 3d object detection. In *ICCV*, 2023. 17, 22

Yuanlong Wang, Hengtao Jiang, Guanying Chen, Tong Zhang, Jiaqing Zhou, Zezheng Qing, Chunyan Wang, and Wanzhong Zhao. Efficient and robust multi-camera 3d object detection in bird-eye-view. *IVC*, 2025. 21

Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemodel Pontes, et al. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023. 1, 6

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *ICML*, 2020. 5

Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *CVPR*, 2023. 3, 17, 22

Jiawei Yang, Ziyu Chen, Yurong You, Yan Wang, Yiming Li, Yuxiao Chen, Boyi Li, Boris Ivanovic, Marco Pavone, and Yue Wang. Towards efficient and effective multi-camera encoding for end-to-end driving. *arXiv preprint arXiv:2512.10947*, 2025a. 3

Ruizi Yang, Xiaolu Liu, Junbo Chen, and Jianke Zhu. Mambamap: Online vectorized hd map construction using state space model. In *IROS*, 2025b. 19

Hongxu Yin, Arash Vahdat, Jose M Alvarez, Arun Mallya, Jan Kautz, and Pavlo Molchanov. A-vit: Adaptive tokens for efficient vision transformer. In *CVPR*, 2022. 9

Zichen Yu, Changyong Shu, Jiajun Deng, Kangjie Lu, Zongdai Liu, Jiangyong Yu, Dawei Yang, Hui Li, and Yan Chen. Flashocc: Fast and memory-efficient occupancy prediction via channel-to-height plugin. *arXiv preprint arXiv:2311.12058*, 2023. 1, 3

Tianyuan Yuan, Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. Streammapnet: Streaming mapping network for vectorized online hd map construction. In *WACV*, 2024. 1, 3, 6, 7, 9, 15, 18

Aziza Zhanabatyrova, Clayton Souza Leite, and Yu Xiao. Structure from motion-based mapping for autonomous driving: practice and experience. *TIOT*, 2024. 1

Ji Zhang, Sanjiv Singh, et al. Loam: Lidar odometry and mapping in real-time. In *RSS*, 2014. 1

Tianyuan Zhang, Zhengfei Kuang, Haian Jin, Zexiang Xu, Sai Bi, Hao Tan, He Zhang, Yiwei Hu, Milos Hasan, William Freeman, et al. Relitlrm: Generative relightable radiance for large reconstruction models. In *ICLR*, 2025. 3, 5

Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *CVPR*, 2022. 1, 3

Daquan Zhou, Bingyi Kang, Xiaojie Jin, Linjie Yang, Xiaochen Lian, Zihang Jiang, Qibin Hou, and Jiashi Feng. Deepvit: Towards deeper vision transformer. *arXiv preprint arXiv:2103.11886*, 2021. 19

Zikang Zhou, Luyao Ye, Jianping Wang, Kui Wu, and Kejie Lu. Hivt: Hierarchical vector transformer for multi-agent motion prediction. In *CVPR*, 2022. 8, 9

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. 1

A Additional Qualitative Results

A.1 Online Vectorized HD Map Construction

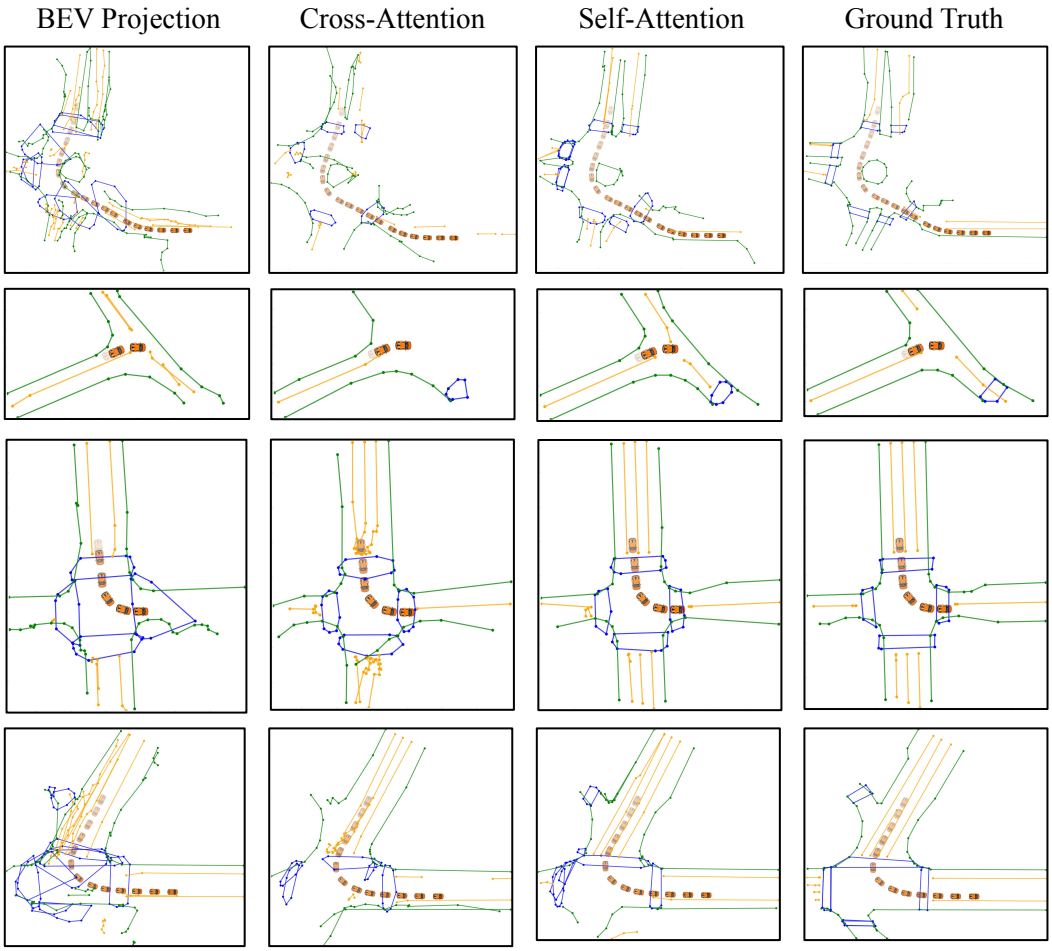


Figure 6: **Qualitative comparisons of online vectorized HD map construction** for Dense BEV Projection (Liao et al., 2025), ours (cross-attention), and ours (self-attention). Self-attention produces higher-quality and more consistent vectorized map reconstructions, with fewer spurious elements, more complete map geometries, and better-preserved intersection geometry. The top two examples are from nuScenes, while the bottom two are from Argoverse 2. Orange, blue, and green denote lane dividers, pedestrian crossings, and road boundaries, respectively. Best viewed zoomed in.

B Additional Quantitative Results

B.1 Map Construction under Different Splits

Table 4 shows that, under the StreamMapNet split, ours surpasses MapTracker by +5.5 mAP on nuScenes and +2.8 mAP on Argoverse 2 at 60m×30m, and by +5.9 and +2.4 mAP, respectively, at 100m×50m.

Table 4: **Quantitative comparisons of online vectorized HD map construction on the StreamMapNet split (Yuan et al., 2024).** VectorMapNet, MapTracker, and AugMapNet results are taken from their original papers, and MapTR and StreamMapNet results are from Yuan et al. (2024). AugMapNet reports no Argoverse 2 results, and MapTRv2 and DAMap report none on this split.

Method	nuScenes				Argoverse 2			
	AP_{div}	AP_{ped}	AP_{bound}	mAP↑	AP_{div}	AP_{ped}	AP_{bound}	mAP↑
60m × 30m perception range								
MapTR (Liao et al., 2022)	20.7	6.4	35.5	20.9	50.4	48.1	55.0	51.1
VectorMapNet (Liu et al., 2023b)	17.0	15.8	21.2	18.0	34.9	35.6	37.8	36.1
StreamMapNet (Yuan et al., 2024)	30.1	29.6	41.9	33.9	55.9	56.9	61.4	58.1
MapTracker (Chen et al., 2024)	30.0	45.9	45.1	40.3	75.1	70.0	68.9	71.3
AugMapNet (Monninger et al., 2026)	30.3	39.4	45.3	38.3	—	—	—	—
Ours (self-attention)	**36.1**	**49.6**	**51.7**	**45.8**	**79.2**	**72.8**	**70.2**	**74.1**
100m × 50m perception range								
MapTR (Liao et al., 2022)	16.0	8.3	20.0	14.8	36.3	46.3	38.0	40.2
VectorMapNet (Liu et al., 2023b)	8.1	12.0	6.3	8.8	20.6	32.4	24.3	25.7
StreamMapNet (Yuan et al., 2024)	19.6	24.8	24.7	23.0	44.4	60.5	48.6	51.2
MapTracker (Chen et al., 2024)	24.3	45.9	38.4	36.2	64.6	71.2	58.5	64.8
AugMapNet (Monninger et al., 2026)	22.8	35.5	28.4	28.9	—	—	—	—
Ours (self-attention)	**33.9**	**51.9**	**40.4**	**42.1**	**67.1**	**73.9**	**60.7**	**67.2**

Table 5 shows that, under the original split, ours surpasses MapTracker by +1.6 mAP on nuScenes and +1.1 mAP on Argoverse 2. As discussed in Section 4.1 in the main paper, substantial train–val geographic overlap allows models to encounter the same static map regions during training and evaluation. As a result, performance reflects memorization in addition to geographic generalization, which helps explain the smaller gains.

Table 5: **Quantitative comparisons of online vectorized HD map construction on the original split.** MapTR's Argoverse 2 result is taken from MapUnveiler (Kim et al., 2024). SparseMeXt uses a different 100-epoch training schedule and reports only nuScenes results; the remaining results are taken from each method's original paper.

Method	nuScenes				Argoverse 2			
	AP_{div}	AP_{ped}	AP_{bound}	mAP↑	AP_{div}	AP_{ped}	AP_{bound}	mAP↑
MapTR (Liao et al., 2022)	51.5	46.3	53.1	50.3	58.1	54.7	56.7	56.5
VectorMapNet (Liu et al., 2023b)	47.3	36.1	39.3	40.9	36.1	38.3	39.2	37.9
StreamMapNet (Yuan et al., 2024)	66.3	61.9	62.1	63.4	74.2	70.5	66.1	70.3
MapTracker (Chen et al., 2024)	74.1	**80.0**	74.1	76.1	80.0	77.0	73.7	76.9
MapTRv2 (Liao et al., 2025)	62.4	59.8	62.4	61.5	72.1	62.9	67.1	67.4
DAMap (Dong et al., 2025)	64.7	58.5	65.1	62.8	73.2	64.7	70.2	69.4
SparseMeXt (Jiang et al., 2025)	67.0	62.6	66.1	65.2	–	–	–	–
AugMapNet (Monninger et al., 2026)	65.4	61.9	63.6	63.6	57.4	57.4	61.6	58.8
Ours (self-attention)	**77.2**	76.5	**79.5**	**77.7**	**80.2**	**78.5**	**75.3**	**78.0**

B.2 Real-Time Map Construction and Trajectory Prediction

Table 6 shows that, with a MobileNetV3-Large (Howard et al., 2019) backbone, ours with self-attention surpasses MapTRv2 (Liao et al., 2025) by **+10.1** mAP on nuScenes and **+16.2** mAP on Argoverse 2, and the cross-attention variant by **+6.5** and **+5.5** mAP, respectively. With a ResNet-18 backbone, ours surpasses MapTRv2 by **+7.3** mAP on nuScenes and **+11.5** mAP on Argoverse 2, and the cross-attention variant by **+6.3** and **+7.1** mAP, respectively. Ours also achieves the highest inference speed with both backbones, reaching **40.2 FPS** with MobileNetV3-Large and **43.0 FPS** with ResNet-18.

Table 6: **Quantitative comparisons of online vectorized HD map construction using real-time image backbones.** Ours with self-attention surpasses both the cross-attention variant and MapTRv2 on nuScenes and Argoverse 2, while also achieving the highest inference speed (FPS) on nuScenes.

Method	nuScenes				Argoverse 2				
	AP_{div}	AP_{ped}	AP_{bound}	mAP↑	AP_{div}	AP_{ped}	AP_{bound}	mAP↑	FPS↑
MobileNetV3-Large backbone									
MapTRv2 (Liao et al., 2025)	19.2	13.0	29.2	20.5	50.7	37.2	43.2	43.7	23.3
Ours (cross-attention)	14.6	30.1	27.5	24.1	57.5	54.2	51.4	54.4	39.8
Ours (self-attention)	**25.4**	**34.9**	**31.6**	**30.6**	**65.3**	**61.1**	**53.2**	**59.9**	**40.2**
ResNet-18 backbone									
MapTRv2 (Liao et al., 2025)	20.3	15.7	33.0	23.0	53.4	41.7	45.1	46.7	24.2
Ours (cross-attention)	15.9	28.1	28.1	24.0	56.4	48.6	48.3	51.1	42.5
Ours (self-attention)	**23.0**	**32.5**	**35.3**	**30.3**	**64.5**	**57.2**	**52.8**	**58.2**	**43.0**

Table 7 shows that, for uncertainty-aware map construction, ours with self-attention surpasses MapTRv2 by **+10.7** mAP with a MobileNetV3-Large backbone and **+10.6** mAP with a ResNet-18 backbone, and the cross-attention variant by **+6.3** and **+6.1** mAP, respectively. For downstream trajectory prediction, ours with self-attention achieves the best minADE, minFDE, and MR across both backbones.

Table 7: **Quantitative comparisons of uncertainty-aware online vectorized HD map construction and downstream trajectory prediction on nuScenes using real-time image backbones.** Ours with self-attention surpasses both the cross-attention variant and MapTRv2 on both map construction and downstream trajectory prediction across both image backbones.

Method	Map Construction				Trajectory Prediction		
	AP_{div}	AP_{ped}	AP_{bound}	mAP↑	minADE↓	minFDE↓	MR↓
MobileNetV3-Large backbone							
MapTRv2 (Liao et al., 2025)	15.2	9.5	25.7	16.8	0.438	0.969	0.1330
Ours (cross-attention)	16.1	24.1	23.5	21.2	0.447	0.987	0.1351
Ours (self-attention)	**24.1**	**30.0**	**28.4**	**27.5**	**0.413**	**0.933**	**0.1271**
ResNet-18 backbone							
MapTRv2 (Liao et al., 2025)	16.0	8.0	27.1	17.0	0.440	0.965	0.1299
Ours (cross-attention)	15.0	26.9	22.7	21.5	0.444	0.987	0.1350
Ours (self-attention)	**20.6**	**31.9**	**30.2**	**27.6**	**0.420**	**0.942**	**0.1274**

B.3 3D Object Detection

Table 8 shows that replacing the BEV projection modules of BEVFormer v2 (Yang et al., 2023) with our self-attention mapper improves **+2.6** NDS and **+3.7** mAP, while increasing inference speed from 8.1 to **26.9 FPS** (3.3× faster). Even compared with the recent state-of-the-art STUR3D (Fan et al., 2026), ours improves **+0.5** NDS and **+1.2** mAP, while achieving lower errors across all five nuScenes metrics: mATE, mASE, mAOE, mAVE, and mAAE. Ours with self-attention also substantially outperforms the cross-attention variant in accuracy.

Table 8: **Quantitative comparisons of 3D object detection on the nuScenes original split with a ResNet-50 backbone.** Ours with self-attention surpasses both the cross-attention variant and state-of-the-art methods in accuracy, and achieves the highest inference speed (FPS) among methods that report it. Following common 3D object detection protocol, NDS and mAP are reported as fractions in $[0, 1]$, whereas improvements in the text are in percentage points. All listed results are without perspective-view pre-training on nuImages. All methods use 256×704 inputs, except BEVFormer v2, which uses 640×1600 inputs and future frames as in its original paper. FPS of BEVFormer v2 and ours are measured on an NVIDIA RTX 3090; other FPS values are as reported in the original papers on RTX 3090. "–" indicates FPS not reported for this setting.

Method	NDS↑	mAP↑	mATE↓	mASE↓	mAOE↓	mAVE↓	mAAE↓	FPS↑
BEVDepth (2023b)	0.475	0.351	0.639	0.267	0.479	0.428	0.198	15.7
Sparse4Dv2 (2023)	0.539	0.439	0.598	0.270	0.475	0.282	0.179	20.3
StreamPETR (2023)	0.540	0.432	0.581	0.272	0.413	0.295	0.195	26.7
SparseBEV (2023a)	0.545	0.432	0.606	0.274	0.387	0.251	0.186	23.5
BEVFormer v2 (2023)	0.529	0.423	0.618	0.273	0.413	0.333	0.181	8.1
VideoBEV (2024)	0.535	0.422	0.564	0.276	0.440	0.286	0.198	–
BEVNeXt (2024a)	0.548	0.437	0.550	0.265	0.427	0.260	0.208	–
FreqPDE (2025)	0.543	0.435	0.577	0.270	0.442	0.257	0.199	–
STUR3D (2026)	0.550	0.448	0.558	0.275	0.431	0.257	0.208	–
Ours (cross-attention)	0.463	0.390	0.710	0.271	0.403	0.355	0.185	26.7
Ours (self-attention)	**0.555**	**0.460**	**0.518**	**0.262**	**0.366**	**0.236**	**0.170**	**26.9**

B.4 Efficiency Comparisons

Table 9: **Efficiency comparison with state-of-the-art methods on nuScenes.** Our MapLightning uses substantially fewer intermediate tokens and achieves higher inference speed than prior methods. While existing approaches use 5k–20k tokens, ours requires only 1.2k tokens. With self-attention, our method achieves the highest FPS within each backbone: 18.9 FPS with ResNet-50, 40.2 FPS with MobileNetV3-Large, and 43.0 FPS with ResNet-18.

Method	Backbone	Projector	Decoder	# Tokens (k)↓	FPS↑
VectorMapNet (Liu et al., 2023b)	ResNet-50	IPM	ART	20.0	2.2
MapTR (Liao et al., 2022)	ResNet-50	GKT	Deformable DETR	20.0	15.1
StreamMapNet (Yuan et al., 2024)	ResNet-50	BEVFormer	Deformable DETR	5.0	12.5
MapTracker (Chen et al., 2024)	ResNet-50	BEVFormer	Deformable DETR	5.0	13.0
MapTRv2 (Liao et al., 2025)	ResNet-50	LSS	Deformable DETR	20.0	14.1
DAMap (Dong et al., 2025)	ResNet-50	LSS	Deformable DETR	20.0	12.1
AugMapNet (Monninger et al., 2026)	ResNet-50	BEVFormer	Deformable DETR	5.0	11.2
Ours (cross-attention)	ResNet-50	Cross-Attention	DETR	**1.2**	18.8
Ours (self-attention)	ResNet-50	Self-Attention	DETR	**1.2**	**18.9**
MapTRv2 (Liao et al., 2025)	MobileNetV3-Large	LSS	Deformable DETR	20.0	23.3
Ours (cross-attention)	MobileNetV3-Large	Cross-Attention	DETR	**1.2**	39.8
Ours (self-attention)	MobileNetV3-Large	Self-Attention	DETR	**1.2**	**40.2**
MapTRv2 (Liao et al., 2025)	ResNet-18	LSS	Deformable DETR	20.0	24.2
Ours (cross-attention)	ResNet-18	Cross-Attention	DETR	**1.2**	42.5
Ours (self-attention)	ResNet-18	Self-Attention	DETR	**1.2**	**43.0**

Table 10: **Component-wise runtime (ms) and allocated memory (MiB) on nuScenes.** We compare MapTRv2 with ours across different backbones, intermediate token counts, and numbers of mapper layers. Our design substantially reduces the runtime and memory of both the mapper and decoder, while leaving the backbone cost unchanged. The cross-attention variant has comparable cost to our self-attention design. MapTRv2 uses LSS rather than transformer layers, so its Mapper-layer count is denoted by "–". Components are rounded individually.

Method	#Tokens (k)	#Mapper Layers	Runtime (ms)↓				Memory (MiB)↓			
			Backbone	Mapper	Decoder	Total	Backbone	Mapper	Decoder	Total
ResNet-50 backbone										
MapTRv2 (Liao et al., 2025)	20.0	–	42.4	11.6	16.9	70.9	590.9	363.2	126.5	1080.6
Ours (cross-attention)	1.2	4	42.4	1.9	8.9	53.2	590.9	22.8	28.7	642.4
Ours (self-attention)	0.9	4	42.4	1.6	8.7	52.7	590.9	32.7	28.6	652.2
Ours (self-attention)	1.5	4	42.4	1.6	8.9	52.9	590.9	33.5	28.7	653.1
Ours (self-attention)	1.2	3	42.4	1.3	8.9	52.5	590.9	31.6	28.7	651.2
Ours (self-attention)	1.2	5	42.4	1.9	8.9	53.2	590.9	34.6	28.7	654.2
Ours (self-attention)	1.2	4	42.4	1.7	8.9	52.9	590.9	33.1	28.7	652.7
MobileNetV3-Large backbone										
MapTRv2 (Liao et al., 2025)	20.0	–	14.4	11.6	16.9	42.9	321.6	363.2	126.5	811.3
Ours (cross-attention)	1.2	4	14.4	1.9	8.9	25.1	321.6	22.8	28.7	373.1
Ours (self-attention)	1.2	4	14.4	1.7	8.9	24.9	321.6	33.1	28.7	383.4
ResNet-18 backbone										
MapTRv2 (Liao et al., 2025)	20.0	–	12.8	11.6	16.9	41.3	354.7	363.2	126.5	844.4
Ours (cross-attention)	1.2	4	12.8	1.9	8.9	23.5	354.7	22.8	28.7	406.2
Ours (self-attention)	1.2	4	12.8	1.7	8.9	23.3	354.7	33.1	28.7	416.5

C Ablation Studies

Ablation on Token Count and Mapper Depth. Table 11 examines the effect of token count and mapper depth on accuracy. Increasing the number of tokens from 0.9k to 1.2k improves accuracy, but further increasing to 1.5k degrades it, possibly due to increasing redundancy among 1D map tokens. Varying the mapper depth shows a similar pattern: a depth of 3 is insufficient, while a depth of 5 slightly degrades accuracy, consistent with prior observations that transformers can exhibit diminishing returns as attention representations become increasingly redundant (Zhou et al., 2021). Our final configuration of 1.2k tokens and a mapper depth of 4 therefore offers the best accuracy among the settings tested, with FPS remaining nearly constant across all configurations.

Ablation on Timesteps. Table 11 also shows that accuracy improves consistently up to 4 timesteps and slightly degrades at 6, making 4 timesteps the best-performing setting. This observation is consistent with MemFusionMap (Song et al., 2025), which conjectures that excessive temporal fusion dilutes the current-frame feature, and MambaMap (Yang et al., 2025b), which conjectures that an excessively long memory window introduces redundancy and noise.

Table 11: **Ablations over token count, mapper depth, and timesteps**. Our final configuration of 1.2k tokens, 4 mapper layers, and 4 timesteps (current frame + 3 past frames) achieves the best accuracy across configurations. FPS varies only from 18.8 to 19.2 across all configurations and is thus omitted.

Config	AP_{div}	AP_{ped}	AP_{bound}	mAP↑
Ablation on token count (4 layers, 4 timesteps)				
0.9k tokens	26.5	39.2	34.1	33.3
1.2k tokens	**28.3**	**41.0**	**36.7**	**35.3**
1.5k tokens	27.8	40.7	36.1	34.9
Ablation on mapper depth (1.2k tokens, 4 timesteps)				
3 layers	26.8	39.8	34.8	33.8
4 layers	**28.3**	**41.0**	**36.7**	**35.3**
5 layers	27.5	40.9	35.8	34.7
Ablation on timesteps (4 layers, 0.3k tokens/step)				
1 timestep	23.1	35.1	32.2	30.1
2 timesteps	26.9	38.3	33.5	32.9
4 timesteps	**28.3**	**41.0**	**36.7**	**35.3**
6 timesteps	27.1	39.4	34.6	33.7

D JUSTIFICATION FOR THE NEAR-EXTRAPOLATION SPLIT

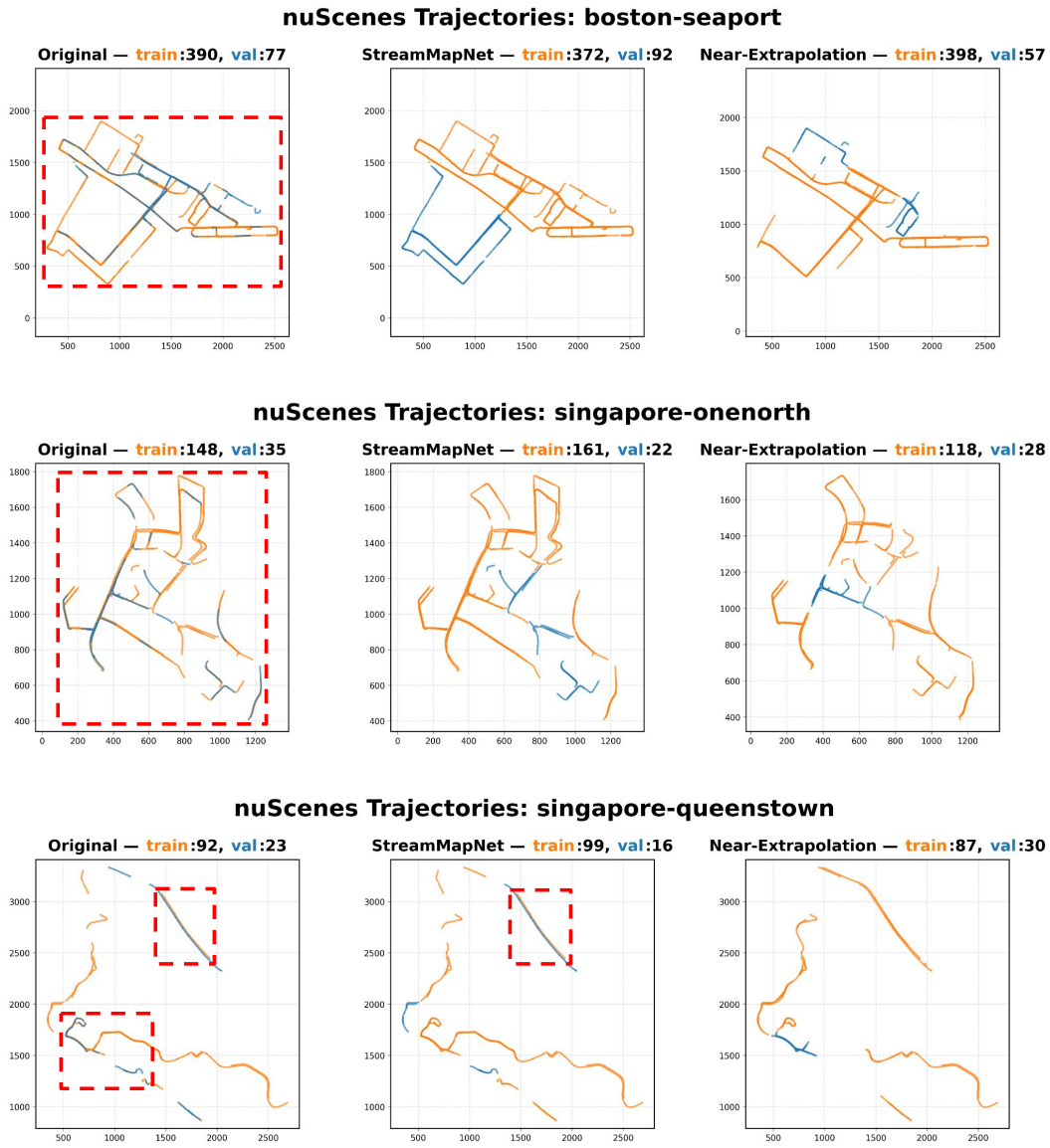


Figure 7: **Visualization of train–val geographic overlap across nuScenes regions.** The Near-Extrapolation split (Lilja et al., 2024), used in our online vectorized HD map construction experiments as the default setting, shows almost no train–val overlap. In contrast, the original split exhibits substantial train–val geographic overlap (79.4%), and the StreamMapNet split retains residual overlap (2.1%), indicating data leakage that undermines their reliability for evaluating online vectorized HD map construction. **Red** dashed boxes highlight regions of train–val geographic overlap within 5 m. Best viewed zoomed in.

E Robustness to Camera Extrinsic Perturbations

In real-world deployment, camera intrinsics are generally stable, while camera extrinsic parameters may be affected by calibration errors (Liao et al., 2025; Wang et al., 2025). Following MapTRv2 (Liao et al., 2025), we evaluate robustness to camera extrinsic perturbations by independently adding zero-mean Gaussian noise with varying standard deviations to the rotation and translation components of the camera extrinsics. Results are reported in Figure 8, where our method shows no degradation, maintaining 35.3 mAP on nuScenes and 63.3 mAP on Argoverse 2 across all perturbation levels, as it does not explicitly rely on camera extrinsics. In contrast, under the largest perturbations, Dense BEV Projection drops from 26.7 mAP to 12.8 under rotation and 7.5 under translation on nuScenes, and from 53.1 mAP to 28.4 under rotation and 23.5 under translation on Argoverse 2.

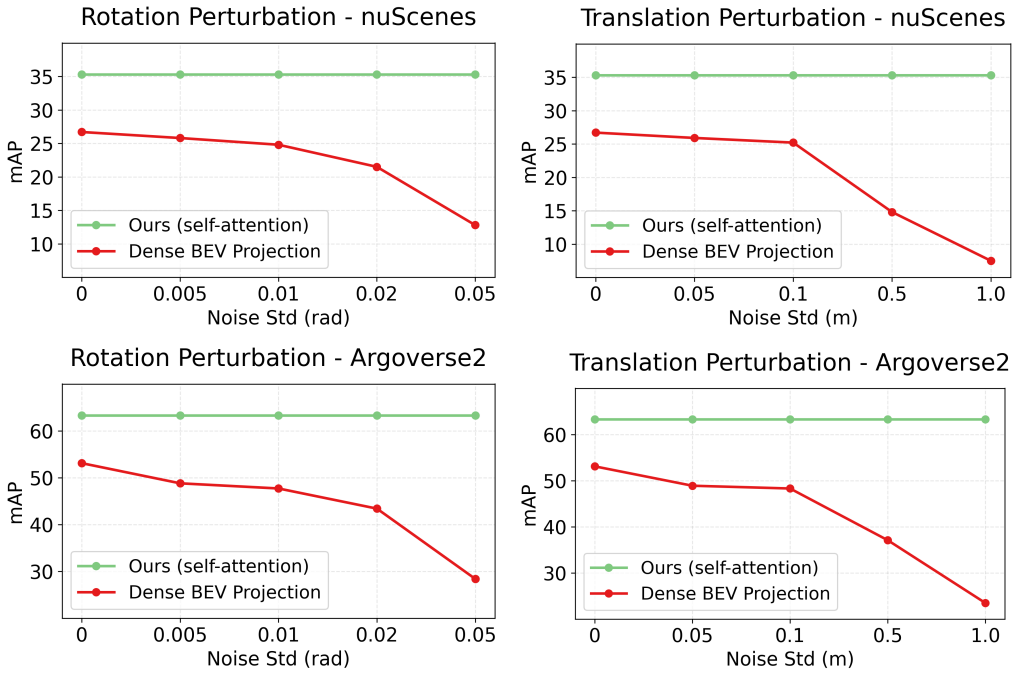


Figure 8: **Robustness to camera extrinsic perturbations.** We independently add zero-mean Gaussian noise to the rotation $(\theta_x, \theta_y, \theta_z)$ and translation $(\Delta x, \Delta y, \Delta z)$ components of all camera extrinsics. Our model does not use camera extrinsics and therefore shows no degradation, whereas Dense BEV Projection (Liao et al., 2025) explicitly relies on them and suffers substantial mAP drops under increasing perturbations.

F Implementation Details

For both online vectorized HD map construction and 3D object detection, we replace the original BEV projection modules with our self-attention mapper and use full instead of deformable cross-attention in the decoder. Unlike the single-frame MapTRv2 (Liao et al., 2025) setting, we use $T = 4$ timesteps and 50 map tokens per camera per timestep, yielding 1.2k map tokens for nuScenes (6 cameras) and 1.4k for Argoverse 2 (7 cameras).

F.1 Online Vectorized HD Map Construction

We mainly follow the MapTRv2 (Liao et al., 2025) framework and training settings, with ImageNet-pretrained (Deng et al., 2009) image backbones and color-jitter augmentation. Input images are resized by a factor of 0.5 on nuScenes (1600×900 to 800×450) and, after padding to 2048×2048, by a factor of 0.3 on Argoverse 2. The decoder uses 50 instance queries, 20 point queries per element, and 6 layers, with classification, point, and direction loss weights of 2, 5, and 0.005. We remove MapTRv2's depth, BEV segmentation, and PV segmentation auxiliary losses, since our mapper involves no depth estimation or BEV grid, and retain its one-to-many matching branch (300 extra instance queries, ground truth repeated 6 times). We train with AdamW (Loshchilov & Hutter, 2017) with a weight decay of 0.01 and a batch size of 4 per GPU on 8 NVIDIA RTX 3090 GPUs for 24 epochs on nuScenes and 6 epochs on Argoverse 2, with an initial learning rate of 6×10^{-4} and cosine decay. Adjacent input frames are 0.5 s apart on nuScenes and 0.4 s apart on Argoverse 2.

F.2 3D Object Detection

We mainly follow the BEVFormer v2 (Yang et al., 2023) framework and training settings on nuScenes, including COCO-pretrained (Lin et al., 2014) backbone initialization and image-level data augmentation, with a ResNet-50 backbone and a 6-layer detection decoder. Unlike BEVFormer v2, which also fuses future frames and uses first-stage perspective proposals as hybrid object queries, we use only past frames (4-frame input, adjacent frames 0.5 s apart) and learned object queries. For easier comparison with prior work (Li et al., 2023b; Lin et al., 2023; Wang et al., 2023; Liu et al., 2023a; Han et al., 2024; Li et al., 2024a; Su et al., 2025; Fan et al., 2026), we use an input resolution of 256×704. We train for 24 epochs on 8 NVIDIA RTX 3090 GPUs with a batch size of 2 per GPU using AdamW (Loshchilov & Hutter, 2017) with a weight decay of 0.01, an initial learning rate of 4×10^{-4}, and an image-backbone learning rate of 2×10^{-4}.