Mapping like a Skeptic: Probabilistic BEV Projection for Online HD Mapping

Fatih Erdoğan[1]
https://fatih-erdogan.github.io/
Merve Rabia Barın[1, 2]
https://mrabiabrn.github.io/
Fatma Güney[1, 2]
https://mysite.ku.edu.tr/fguney/

[1] Department of Computer Engineering,
Koç University,
Istanbul, Turkey
[2] KUIS AI Center

Abstract

Constructing high-definition (HD) maps from sensory input requires accurately mapping the road elements in image space to the Bird's Eye View (BEV) space. The precision of this mapping directly impacts the quality of the final vectorized HD map. Existing HD mapping approaches outsource the projection to standard mapping techniques, such as attention-based ones. However, these methods struggle with accuracy due to generalization problems, often hallucinating non-existent road elements. Our key idea is to start with a geometric mapping based on camera parameters and adapt it to the scene to extract relevant map information from camera images. To implement this, we propose a novel probabilistic projection mechanism with confidence scores to (i) refine the mapping to better align with the scene and (ii) filter out irrelevant elements that should not influence HD map generation. In addition, we improve temporal processing by using confidence scores to selectively accumulate reliable information over time. Experiments on new splits of the nuScenes and Argoverse2 datasets demonstrate improved performance over state-of-the-art approaches, indicating better generalization. The improvements are particularly pronounced on nuScenes and in the challenging long perception range. Our code and model checkpoints are available at https://github.com/Fatih-Erdogan/mapping-like-skeptic.

1 Introduction

HD maps are crucial in self-driving to provide rich semantic information about roads. Due to the high cost of maintaining offline maps, online solutions that use onboard sensors are more desirable. In online HD map estimation, ensuring the accuracy of predicted road elements is critical to safe navigation. Recent approaches estimate vectorized HD maps by first transforming sensory data into BEV space. A key challenge in this process is to extract relevant information from camera images and map it to the BEV space. In this work, we address this challenge by designing a probabilistic formulation that guides geometric mapping to capture map-specific elements from camera images for accurate HD mapping.

The first approaches to online HD map estimation focused on switching from rasterized to vectorized [14] and improving vectorized prediction with architectural modifications and additional losses [16, 17, 19]. Recent approaches focus on incorporating temporal information [4, 21, 24, 29], where the selection of relevant information becomes more crucial due to the increased information with the time axis. However, none of these approaches focus on improving the accuracy of the transformation and rely on standard methods, such as depth-based [22] or attention-based [5, 15], to project image features to the BEV space. We first show that replacing the attention-based projection in the SOTA method [4] with a simple projection [9], i.e., sampling the corresponding image features based on camera parameters, performs surprisingly well, even slightly outperforming the original approach.

We refer to this mapping based on camera parameters as static mapping and highlight its shortcomings for HD maps, such as inaccuracies when the slope changes or when the road is occluded by scene elements like vehicles or pedestrians. Our key idea is that adapting the mapping to the scene can enhance the transformation by more precisely extracting relevant information. However, fully relying on data is also not ideal, as evidenced by the increased false positives in attention-based mapping approaches. Our novelty lies in a probabilistic projection approach for HD mapping. We learn probabilistic distributions and associated confidences over sampling locations to (i) adjust mapping offsets and (ii) filter out irrelevant elements that should not influence the HD map. We also improve temporal processing by using predicted confidences to selectively accumulate reliable information over time.

We evaluated our approach on the challenging new splits of nuScenes and Argoverse2, ensuring separate training and test regions for generalization. Our results show that probabilistic projection significantly outperforms the state-of-the-art across nearly all metrics on both datasets. In particular, we observe improvements in challenging long-range perception on both datasets. While temporal consistency remains similar to the state-of-the-art on Argoverse2, it is significantly improved on nuScenes. Our visual analysis reveals fewer false positives, especially in pedestrian crossings, compared to attention-based baselines. Our contributions can be summarized as follows:

- A novel probabilistic projection method to improve mapping from camera images to BEV by better capturing road structures.
- Improved temporal processing with confidence-based selection, effectively accumulating historical information.
- Advancing state-of-the-art in both short and long perception ranges on nuScenes and Argoverse2.

2 Related Work

Online Vectorized HD Mapping: Recent HD map methods build maps online from multiple camera views, using vector representations that preserve instance details. HDMapNet [14] is the first method to frame the vectorized HD mapping task as a rasterized BEV segmentation problem. However, it requires post-processing to convert the resulting semantic, instance, and directional BEV maps into a fully vectorized map representation. VectorMapNet [19] introduces an end-to-end framework that directly predicts polylines with a DETR-like detection head [2], but it generates polyline vertices in an autoregressive manner, which may cause inefficiency. MapTR [16] has a similar but more efficient transformer-based architecture, using hierarchical modeling for map elements and matching queries level by level to predict all vertices of an element at once. MapTRv2 [17] improves MapTR with

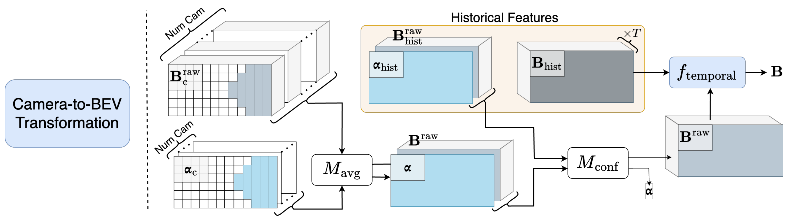
Camera-to-BEV Transformation
Num Cam
$\mathbf{B}^{\text{raw}}_c$
Num Cam
α_c
M_{avg}
$\mathbf{B}^{\text{raw}}$
α
Historical Features
$\mathbf{B}^{\text{raw}}_{\text{hist}}$
α_{hist}
$\mathbf{B}_{\text{hist}}$
$\times T$
M_{conf}
α
f_{temporal}
$\mathbf{B}$
$\mathbf{B}^{\text{raw}}$

Figure 1: **Overall Framework.** We start by projecting camera features onto the BEV grid using probabilistic projection (detailed in Fig. 2), resulting in per-camera raw BEV features and the corresponding confidence scores $\{\mathbf{B}^{\text{raw}}_c, \alpha_c \mid c \in \mathcal{C}\}$ for each camera. Then, we merge per-camera features and confidences (M_{avg}), resulting in $\mathbf{B}^{\text{raw}}$. We then incorporate temporal information by taking a weighted sum (M_{conf}) with historical raw BEV features $\mathbf{B}^{\text{raw}}_{\text{hist}}$ according to the confidence scores. Finally, we process the raw BEV features and the historical BEV features $\mathbf{B}_{\text{hist}}$ with f_{temporal} to obtain the final BEV features, $\mathbf{B}$.

auxiliary segmentation losses and a memory-efficient transformer decoder using decoupled self-attention. Some works focus on refining the interactions in the decoder mechanism and/or adding additional task losses [11, 18, 20, 28]. Different from these approaches, Mask2Map [6] replaces the detection-based pipeline with a segmentation mask-based approach. Others concentrate on improving vectorized map modeling [7, 23, 30, 32].

Incorporating Temporal Information: StreamMapNet [29] uses a memory module with a streaming strategy [8, 25] to incorporate temporal information, processing each frame separately while propagating a hidden state to maintain long-term temporal connections. SQD-MapNet [26] inserts denoised ground truth via queries to improve the temporal consistency of map elements. PrevPredMap [21] selects the top predicted queries based on their confidence to form new queries with category and location information. MapTracker [4] uses two strategies for its BEV and vector memory modules. For its BEV memory module, it selects past information based on distance rather than merging all historical data, reducing information loss. For its vector module, it uses query propagation from tracking to link elements from previous frames with those in the current frame. MemFusionMap [24] utilizes a temporal overlap heatmap to combine the current and past features, highlighting areas that should favor current data over memory while also encoding vehicle trajectory for better temporal reasoning. Lastly, MapUnveiler [13] utilizes clip-level information from adjacent frames by learning tokens that help with occluded elements, ensuring long-term consistency.

Recent work focuses on architectural changes [11, 16, 17, 18, 19, 20, 28] or temporal module [4, 13, 21, 24, 26, 29] while relying on off-the-shelf methods [5, 12, 15, 22] for projecting image features into BEV. In this work, we focus on improving the projection with a probabilistic framework to capture map elements more accurately. For the vector module, we build on MapTracker's design [4], which is shown to achieve the best performance.

Bird's Eye View Segmentation: Techniques used in BEV perception are essential for map learning since they transform 2D camera features into the BEV space. This transformation can be divided into three approaches. *Depth-based* [12, 22] models learn an explicit depth distribution at each pixel. *Attention-based* methods [5, 31] learn geometry implicitly through cross-attention between sensor features and predefined 3D grid coordinates. *Sampling-based* [3, 9] methods map voxel grids to camera image pixels and bilinearly sample the intersecting image features.

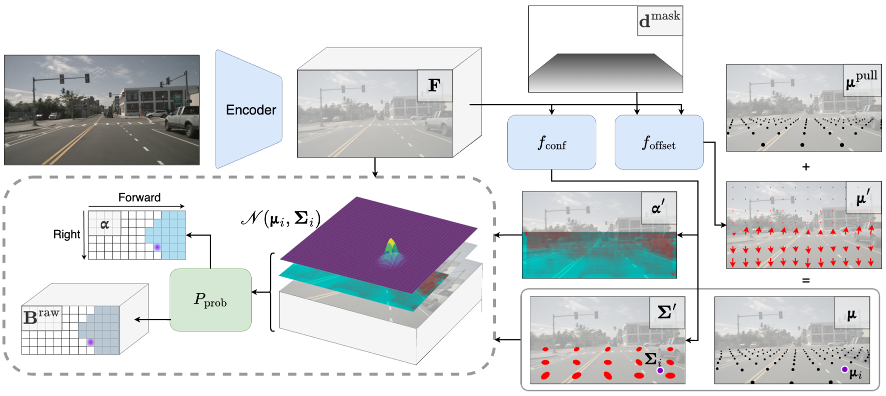

Figure 2: **Camera-to-BEV Transformation using Probabilistic Projection.** We illustrate the camera-to-BEV transformation for a single camera by dropping the subscript c in the notation. The camera image is first processed by an encoder to extract features $\mathbf{F}$. These features are then processed by two networks, f_{offset} and f_{conf}, to predict the probabilistic projection parameters. Given a distance mask $\mathbf{d}^{\text{mask}}$ as an additional input, f_{offset} predicts the offset values μ', which adjust the static BEV-to-image mapping μ^{pull} to form a more accurate mapping μ. Meanwhile, f_{conf} estimates the covariance Σ' and the confidence scores α'. **Probabilistic Projection (dashed box):** For a grid cell i, the corresponding values μ_i and Σ_i from μ and Σ' parameterize the Gaussian distribution $\mathcal{N}(\mu_i, \Sigma_i)$. Based on this distribution, we sample K locations in the image and construct the feature of the grid cell as a weighted sum of feature vectors projected from these sampled locations. The weight, representing each pixel's contribution, is determined by the likelihood of the mapping and the pixel's confidence. Note that we use $'$ with a variable to denote the variable in image space, except for μ^{pull} and μ, which are shown in image space for illustration purposes but are formally defined in BEV space. For more details, see the text.

Both depth-based [22] and attention-based [6] models have been utilized by the vectorized HD mapping methods [16, 17]. However, depth-based methods struggle with the complexity of accurately predicting depth for every pixel, while attention-based methods do not benefit from explicit geometric modeling. Both approaches are prone to projection errors. BEVFormer [15] uses both attention and sampling mechanisms with deformable attention to aggregate image features into 3D reference grids, a method preferred by the latest mapping methods [4, 29]. However, this method suffers from false positives by placing road elements where they occur most frequently in the training data distribution. With the proposed probabilistic projection for HD mapping, our method better preserves the geometry of road structures.

3 Methodology

Given images from multiple cameras surrounding a vehicle, online mapping approaches aim to extract an HD map of the environment by segmenting lane boundaries, dividers, and pedestrian crossings in BEV representation. Recent work [4] first constructs a BEV feature space using a BEV encoder. Based on the extracted BEV features, a vectorized map representation is decoded. In this work, we focus on improving the BEV feature space in the

Algorithm 1 Map Prediction with Probabilistic Projection

```
1:  Input: The set of cameras $\mathcal{C}$, combined historical raw BEV features $\mathbf{B}^{\text{raw}}_{\text{hist}}$, their confidence scores $\alpha_{\text{hist}}$, selected historical BEV features $\mathbf{B}_{\text{hist}}$, and for each camera, encoded features, distance mask, and pull indices, $\{\mathbf{F}_c, \mathbf{d}^{\text{mask}}_c, \mu^{\text{pull}}_c | c \in \mathcal{C}\}$
2:  Output: Predicted BEV features, $\mathbf{B}$
3:  for each camera $c \in \mathcal{C}$ do
4:      Offset Prediction
        $\mu'_c \leftarrow f_{\text{offset}}(\mathbf{F}_c, \mathbf{d}^{\text{mask}}_c)$
5:      Covariance and Confidence Prediction
        $(\Sigma'_c, \alpha'_c) \leftarrow f_{\text{conf}}(\mathbf{F}_c)$
6:      Offset Sampling
        $\mu^{\text{off}}_c \leftarrow P(\mu'_c, \mu^{\text{pull}}_c)$
7:      Updating Pull Index
        $\mu_c \leftarrow \mu^{\text{pull}}_c + \mu^{\text{off}}_c$
8:      Covariance Sampling
        $\Sigma_c \leftarrow P(\Sigma'_c, \mu_c)$
9:      Probabilistic Projection
        $(\mathbf{B}^{\text{raw}}_c, \alpha_c) \leftarrow P_{\text{prob}}(\mathbf{F}_c, \mu_c, \Sigma_c, \alpha'_c)$
10: end for
11: Merging Camera Features
    $\mathbf{B}^{\text{raw}}, \alpha \leftarrow M_{\text{avg}}(\{\mathbf{B}^{\text{raw}}_c \mid c \in \mathcal{C}\}), M_{\text{avg}}(\{\alpha_c \mid c \in \mathcal{C}\})$
12: Merging with Historical Raw Features
    $(\mathbf{B}^{\text{raw}}, \alpha) \leftarrow M_{\text{conf}}(\mathbf{B}^{\text{raw}}, \alpha, \mathbf{B}^{\text{raw}}_{\text{hist}}, \alpha_{\text{hist}})$
13: Parametric Temporal Fusion
    $\mathbf{B} \leftarrow f_{\text{temporal}}(\mathbf{B}^{\text{raw}}, \mathbf{B}_{\text{hist}})$
14: Save $\mathbf{B}$, $\mathbf{B}^{\text{raw}}$, $\alpha$
15: Return $\mathbf{B}$
```

first stage by introducing a novel probabilistic projection mechanism.

Our overall framework is illustrated in Fig. 1, with the corresponding steps detailed in Algorithm 1. Additionally, the probabilistic projection is illustrated in Fig. 2.

We first process camera images with a ResNet [10] encoder, followed by a Feature Pyramid Network, resulting in features $\mathbf{F}_c \in \mathbb{R}^{C \times H \times W}$ for the camera c where $H \times W$ denotes the reduced spatial resolution of the image and C, the hidden dimension size. Assuming known intrinsics and extrinsics, we can define the mapping $\mu^{\text{pull}}_c \in \mathbb{R}^{h \times w \times 2}$ to pull features from the image plane of the camera c to a BEV grid of size $h \times w$.

3.1 Probabilistic Projection

The static mapping μ^{pull}_c based on camera parameters fails to capture scene structures, e.g., when the slope changes (see Supplementary for a visualization). In this work, we propose to improve this mapping by predicting per-pixel offset distributions $\mathcal{N}(\mu'_c, \Sigma'_c)$ and an associated confidence map α'_c to direct the offsets towards more relevant locations on the image:

$$\mu'_c = f_{\text{offset}}(\mathbf{F}_c, \mathbf{d}^{\text{mask}}_c) \tag{1}$$
$$\Sigma'_c, \alpha'_c = f_{\text{conf}}(\mathbf{F}_c) \tag{2}$$

where f_{offset} is a 3-layer CNN processing the camera features $\mathbf{F}_c$ together with a distance mask $\mathbf{d}^{\text{mask}}_c \in \mathbb{R}^{H \times W}$ for positional information. Since the correction we want to apply to a point depends on how far the point is from the camera due to perspective projection, we encode the distance mask $\mathbf{d}^{\text{mask}}_c$ with a 3-layer CNN and provide it as input to the offset predictor f_{offset}. We use another 3-layer CNN, f_{conf}, to predict covariance and confidences; differently, we do not provide the distance mask as input.

Updated Projection Parameters: Given the projection parameters, i.e. offsets μ'_c and the covariance Σ'_c, predicted from the camera features, we project them to the BEV grid. The projection operation, denoted by $P(\cdot,\cdot)$, uses the second argument to sample bilinearly from the first argument:

$$\begin{aligned} \mu_c^{\text{off}} &= P(\mu_c', \mu_c^{\text{pull}}) & (3)\\ \mu_c &= \mu_c^{\text{pull}} + \mu_c^{\text{off}} & (4)\\ \Sigma_c &= P(\Sigma_c', \mu_c) & (5) \end{aligned}$$

We first project offsets μ_c' using the static mapping μ_c^{pull} in (3). We then residually add the projected offsets μ_c^{off} to the static mapping μ_c^{pull} to obtain the updated mappings μ_c in (4). Finally, we project the covariance Σ_c' by using the updated mappings μ_c in (5).

On a practical note, the projection is implemented using `grid_sample`, followed by a `permute` operation to arrange dimensions. To further clarify the notation and dimensionality, we use μ_c', Σ_c' to represent the probabilistic projection parameters with respect to the perspective camera at the spatial resolution $H \times W$. We denote the projected version respectively as μ_c, Σ_c, in the spatial resolution of the BEV grid, i.e. $h \times w$. The offset parameters μ_c have 2 channels and the covariance Σ_c has 3 channels. Note that μ_c and Σ_c are empty in the locations where the corresponding BEV grid is outside the camera's field of view.

Probabilistic Feature Projection (P_{prob}): The updated projection parameters define a probabilistic mapping at each grid cell, pointing to the most likely locations in the respective camera to pull features from. Based on this distribution, we sample K pixels from camera features $\mathbf{F}_c$ for a grid cell and then represent the grid cell as a weighted sum of feature vectors of the sampled pixels. We define the weight by combining the Gaussian likelihoods and the confidence scores of the sampled locations.

Formally, let $\mathbf{B}_c^{\text{raw}} \in \mathbb{R}^{C\times h\times w}$ be the result of the probabilistic projection of the image features for the camera c. For a grid cell i on $\mathbf{B}_c^{\text{raw}}$, let $\mathbf{b}_i \in \mathbb{R}^{C\times 1}$ denote its feature vector pulled from camera c according to the probabilistic mapping proposed. Dropping the dependency on camera index c for clarity, we first sample K locations based on the distribution $\mathcal{N}(\mu_i, \Sigma_i)$ of the grid cell i. We then construct $\mathbf{b}_i$ as a weighted sum of feature vectors projected from the sampled locations. The weight w_k denotes the contribution of pixel k. We define it based on the predicted confidence α_k' for the pixel k (2) and the likelihood of the mapping.

$$\mu_{i,k} \sim \mathcal{N}(\mu_i, \Sigma_i) \qquad \mathbf{b}_i = \sum_{k=1}^{K} w_k\, P(\mathbf{F}, \mu_{i,k}) \qquad w_k = \alpha_k' \cdot \mathcal{N}(\mu_{i,k}; \mu_i, \Sigma_i) \tag{6}$$

We normalize the likelihood of each sample k, $\mathcal{N}(\mu_{i,k}; \mu_i, \Sigma_i)$, by the sum of K Gaussian likelihoods. The confidence score for each grid location i is updated as the sum of weights over sampled locations $\alpha_i = \sum_{k=1}^{K} w_k$.

Merging Camera Features: We merge the resulting BEV grid $\mathbf{B}_c^{\text{raw}}$ and α_c from each camera into $\mathbf{B}^{\text{raw}}$ and α by a simple averaging (M_{avg}):

$$\mathbf{B}^{\text{raw}} = \sum_c \mathbf{B}_c^{\text{raw}} \oslash \max(1, \mathbf{V}) \qquad \alpha = \sum_c \alpha_c \oslash \max(1, \mathbf{V}) \tag{7}$$

where $\oslash$ denotes the elementwise Hadamard division and $\mathbf{V} \in \{0, \ldots, C\}^{h\times w}$ counts the number of cameras contributing to each grid cell.

3.2 Temporal Information

Previous works [4, 21] have shown the importance of utilizing temporal information for HD map prediction. For example, MapTracker [4] warps the previous BEV estimation to use as initialization at the current time step and proposes a memory mechanism. We also utilize temporal information in two ways.

Historical Raw Features: First, we keep track of the most confident raw features and their associated confidences for each grid cell over a time horizon in $\mathbf{B}^{\text{raw}}_{\text{hist}}$ and α_{hist}. We update $\mathbf{B}^{\text{raw}}$ according to historical information by first warping $\mathbf{B}^{\text{raw}}_{\text{hist}}$ to align with the current time step. The update is performed simply by adding the two, each weighted by their corresponding confidences (M_{conf}). We update the confidence matrix α in a similar way (see Supplementary for details).

$$\begin{aligned} \mathbf{B}^{\text{raw}} &= \alpha \odot \mathbf{B}^{\text{raw}} + \alpha_{\text{hist}} \odot \mathbf{B}^{\text{raw}}_{\text{hist}} & \qquad \alpha &= \alpha \odot \alpha + \alpha_{\text{hist}} \odot \alpha_{\text{hist}} \\ \mathbf{B}^{\text{raw}} &= \mathbf{B}^{\text{raw}} \oslash (\alpha + \alpha_{\text{hist}}) & \qquad \alpha &= \alpha \oslash (\alpha + \alpha_{\text{hist}}) \end{aligned} \tag{8}$$

We normalize both by dividing elementwise by the sum of confidences. We save them in the history buffer to be warped and used in future time steps.

Parametric Temporal Fusion (f_{temporal}): So far, the merged BEV grid is simply the result of projecting image features to the BEV grid, i.e. *raw*, without any learning on top of it. Following the memory mechanism proposed in MapTracker [4], we store the last 20 estimated BEV representations in memory and select a subset of size $T = 4$ based on vehicle positions, denoted by $\mathbf{B}_{\text{hist}} \in \mathbb{R}^{T \times C \times h \times w}$. We concatenate $\mathbf{B}^{\text{raw}}$ and $\mathbf{B}_{\text{hist}}$ and feed it into f_{temporal} to obtain the final BEV feature map, $\mathbf{B}$.

Table 1: **Quantitative Results.** Comparison with the state-of-the-art models on nuScenes [1] and Argoverse2 [27] in two ranges, using the new split. Our method outperforms existing methods on both datasets in almost all metrics, with significant improvements on nuScenes.

Range (m)	Dataset	Method	*AP_p*	*AP_d*	*AP_b*	*mAP*	*C-mAP*
60×30	nuScenes [1]	StreamMapNet [29]	31.6	28.1	40.7	33.5	22.2
		MapTracker [4]	45.9	30.0	45.1	40.3	32.5
		Ours	49.8 (+8.5%)	36.2 (+20.7%)	50.1 (+11.1%)	45.4 (+12.7%)	37.2 (+14.5%)
	Argoverse2 [27]	StreamMapNet [29]	61.8	68.2	63.2	64.4	54.4
		MapTracker [4]	70.0	75.1	68.9	71.3	63.2
		Ours	72.9 (+4.1%)	76.9 (+2.4%)	67.5 (-2.0%)	72.4 (+1.5%)	62.7 (-0.8%)
100×50	nuScenes [1]	StreamMapNet [29]	25.1	18.9	25.0	23.0	14.6
		MapTracker [4]	45.9	24.3	38.4	36.2	27.5
		Ours	52.5 (+14.4%)	33.1 (+36.2%)	42.9 (+11.7%)	42.8 (+18.2%)	33.5 (+21.8%)
	Argoverse2 [27]	StreamMapNet [29]	60.1	56.1	47.5	54.6	41.3
		MapTracker [4]	71.2	64.6	58.5	64.8	55.7
		Ours	74.9 (+5.2%)	67.4 (+4.3%)	58.9 (+0.7%)	67.1 (+3.5%)	56.1 (+0.7%)

4 Experiments

Experimental Setup: We evaluate our model on the nuScenes [1] and Argoverse2 [27] datasets, using the revised non-overlapping geographical splits proposed in StreamMap-

Net [29] to better assess generalization, and adopt MapTracker's [4] temporally-aligned ground truth. Performance is measured under short (60m×30m) and long (100m×50m) range settings using Average Precision (AP) at varying distance thresholds, and reported per class as AP_p, AP_d, and AP_b, with both mean AP (*mAP*) and consistency-aware *C-mAP*. We follow the MapTracker [4] training pipeline for both datasets. Full details are provided in the Supplementary.

4.1 Quantitative Comparison

We compare our approach with state-of-the-art online mapping methods StreamMapNet [29] and MapTracker [4] in Table 1. We directly obtained the baseline results from MapTracker and verified its performance using the official codebase, while the StreamMapNet results are taken as reported. Improvements over the previous best results are highlighted in green as percentages, while performance losses are shown in red.

We first note the significant improvement in generalization capability with our method, as the new split ensures no overlapping regions between the training and test sets. The improvements are consistent across almost all categories on both datasets, except for the boundary class on Argoverse2. The results are particularly impressive on nuScenes, with a 12.7% overall improvement in mAP in the short range and 18.2% in the long range, consistently across all categories, especially the divider. The largest improvements on both datasets are achieved in the long range, with +18.2% on nuScenes and +3.5% on Argoverse2. This is particularly noteworthy given the difficulty of long perception ranges, such as distant pedestrian crossings or long dividers. Furthermore, our method improves the temporal consistency on nuScenes in both ranges, demonstrating the effectiveness of temporal processing in our approach.

nuScenes vs. Argoverse2: Our improvements are more pronounced on nuScenes than on Argoverse2. In general, all methods perform better on Argoverse2 than on nuScenes, probably due to its better diversity, covering six cities compared to two on nuScenes. Furthermore, the longer viewing range of the cameras in Argoverse2 reduces the gap between the short- and long-range perception results compared to nuScenes. On Argoverse2, our method achieves notable improvements in pedestrian crossings and lane dividers across both ranges. However, while boundary detection improves slightly in the long range, it performs worse than MapTracker in the short range. Similarly, *C-mAP* decreases slightly in the short range but shows a slight improvement in the long range.

4.2 Qualitative Comparison

In Fig. 3, we present qualitative comparison of our method with MapTracker [4]. While MapTracker often introduces false positives, particularly by placing pedestrian crossings near road junctions (middle), where they frequently appear in the training data, our geometry-based mapping avoids such mode-fitting behavior. Additionally, as shown in Fig. 3 (right and left), our approach more faithfully captures the actual structure of road elements, producing more accurate representations of lane dividers and boundaries. Please see Supplementary for more qualitative analysis.

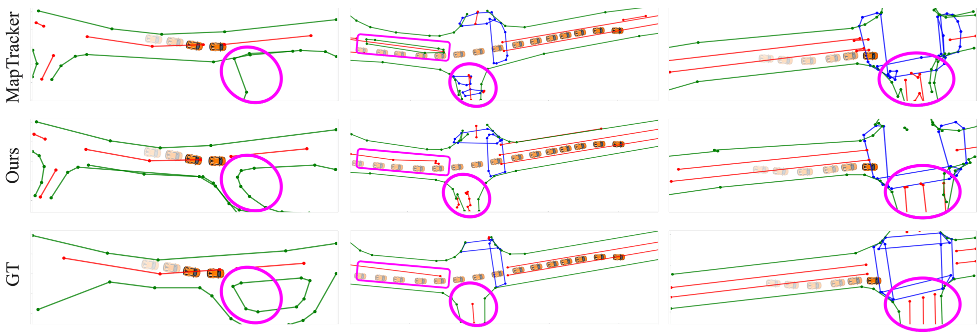

Figure 3: **Qualitative Comparison.** We qualitatively compare our method to MapTracker [4]. Pedestrian crossings, lane dividers, and road boundaries are shown in blue, red, and green, respectively. Our improvements are highlighted with a purple circle for road boundaries (left), pedestrian crossings (middle), and lane dividers (right) on nuScenes.

4.3 Ablation Study

In Table 2, we evaluate the effect of the proposed components on nuScenes in the short range. We begin with the vanilla version in A, which replaces the deformable attention-based projection [15] in MapTracker [4] with a static pull mechanism using bilinear sampling [9]. Note that the vanilla version still has the vector module and the memory mechanism from MapTracker. This version already slightly outperforms MapTracker (40.3 vs. 41.0), showing the potential of a geometry-based projection [9].

Probabilistic Projection ($B_{1,2}$): Learning to adjust the static mapping with per-pixel offset distributions improves the performance from 41.0 to 43.5 (B_1). These results support our motivation to update projection parameters based on scene structures. Grid points adapt to road slope changes, improving coverage of pedestrian crossings and road boundaries (see Supplementary for a visualization).

Results do not change significantly with the inclusion of confidences (B_2), likely due to the uncertainty already being captured with Σ. Without $\mathbf{B}^{\text{raw}}_{\text{hist}}$ in experiments $B_{1,2}$, the confidences do not play any other role in the pipeline.

Table 2: **Ablation Study.** We evaluate the effect of the proposed components: probabilistic projection ($\mathcal{N}(\mu,\Sigma)$), confidence scores (α), and historical raw features ($\mathbf{B}^{\text{raw}}_{\text{hist}}$) on nuScenes in the short range.

ID	$\mathcal{N}(\mu,\Sigma)$	α	$\mathbf{B}^{\text{raw}}_{\text{hist}}$	AP_p	AP_d	AP_b	mAP	$C\text{-}mAP$
A	✗	✗	✗	43.9	31.6	47.4	41.0	33.4
B_1	✓	✗	✗	48.8	35.2	46.6	43.5	34.9
B_2		✓		48.9	34.2	48.0	43.6	35.4
C_1	✗	✗	✓	44.2	32.6	46.0	41.0	33.2
C_2		✓		46.6	35.6	48.1	43.4	35.2
D	✓	✗	✓	47.4	34.2	48.6	43.4	35.3
E		✓		49.8	36.2	50.1	45.4	37.2

Historical Raw Features ($C_{1,2}$): Merging historical raw features without using confidence scores (C_1) does not lead to any improvement. However, when historical information is selected based on confidence scores (C_2), *mAP* increases by +2.4, highlighting the importance of selective merging. Additionally, in C_2, *C-mAP* also improves, indicating better temporal consistency with raw historical features, weighted by confidence. Note that these additional gains come from temporal processing on top of the memory mechanism of MapTracker, which selects frames based on the positions of the vehicle. With confidence scores, our raw historical features selectively accumulate temporal information, keeping only the most reliable information, which leads to improved performance.

Confidence Map (D, E): We use confidence scores in two ways: for weighting features in weighted projection (6), and for selecting and merging historical raw BEV features into the current (8). Removing confidence scores has the least impact on *mAP* (D), likely because uncertainty in weighted projection can still be captured by Σ as mentioned earlier. However, confidence-based selection still plays a crucial role in merging historical raw features, as the best performance is achieved with the complete model (E) when confidences are included. The confidence score learns to assign high values to road structures while filtering out pedestrians, sidewalks, walls, and buildings (see Supplementary for a visualization).

Efficiency Comparison: Our model, with 63.8M parameters, is similar in size to MapTracker (65.8M) and achieves a comparable inference speed of 15.9 FPS vs. 16.3 FPS for MapTracker, both measured on a single NVIDIA A100.

5 Conclusion

We propose a probabilistic projection mechanism to enhance the accuracy of mapping from image to BEV space in online HD map estimation. Our probabilistic formulation, combined with confidence scores, significantly improves performance on the new splits of nuScenes and Argoverse2, particularly in the long range. Qualitative analysis shows that our probabilistic mapping effectively reduces false positives while adapting to scene variations. Additionally, our method improves temporal consistency on nuScenes through confidence-based accumulation of temporal information, emphasizing the importance of adapting temporal selection to the scene.

Limitations and Future Work: While replacing the transformer block with an offset-based feature sampling module reduced false positives and hallucinations, we observed that the model does not fully utilize neighboring context when target regions are occluded (e.g., partially visible lane lines). Hybrid or context-aware mechanisms could be explored to address this limitation. Additionally, our findings indicate that there is room for improvement in leveraging temporal information more effectively. We leave these challenges as promising directions for future work.

Acknowledgements: This project is funded by the European Union (ERC, ENSURE, 101116486) with additional compute support from Leonardo Booster (EuroHPC Joint Undertaking, EHPC-AI-2024A01-060). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, 2020.

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

[3] Loick Chambon, Eloi Zablocki, Mickaël Chen, Florent Bartoccioni, Patrick Pérez, and Matthieu Cord. PointBeV: A sparse approach for bev predictions. In *CVPR*, 2024.

[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.

[5] Shaoyu Chen, Tianheng Cheng, Xinggang Wang, Wenming Meng, Qian Zhang, and Wenyu Liu. Efficient and robust 2D-to-BEV representation learning via geometry-guided kernel transformer. *arXiv preprint arXiv:2206.04584*, 2022.

[6] Sehwan Choi, Jungho Kim, Hongjae Shin, and Jun Won Choi. Mask2map: Vectorized hd map construction using bird's eye view segmentation masks. In *ECCV*, 2024.

[7] Wenjie Ding, Limeng Qiao, Xi Qiu, and Chi Zhang. Pivotnet: Vectorized pivot learning for end-to-end hd map construction. In *ICCV*, 2023.

[8] 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. *RAL*, 9(7):6544–6551, 2024.

[9] Adam W Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. Simple-bev: What really matters for multi-sensor bev perception? In *ICRA*, 2023.

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

[11] Haotian Hu, Fanyi Wang, Yaonong Wang, Laifeng Hu, Jingwei Xu, and Zhiwang Zhang. ADMap: Anti-disturbance framework for vectorized hd map construction. In *ECCV*, 2024.

[12] Junjie Huang and Guan Huang. Bevpoolv2: A cutting-edge implementation of bevdet toward deployment. *arXiv preprint arXiv:2211.17111*, 2022.

[13] Nayeon Kim, Hongje Seong, Daehyun Ji, and Sujin Jang. Unveiling the hidden: Online vectorized HD map construction with clip-level token interaction and propagation. In *NeurIPS*, 2025.

[14] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *ICRA*, 2022.

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

[16] Bencheng Liao, Shaoyu Chen, Xinggang Wang, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Chang Huang. MapTR: Structured modeling and learning for online vectorized hd map construction. In *ICLR*, 2023.

[17] 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*, pages 1–23, 2024.

[18] Xiaolu Liu, Song Wang, Wentong Li, Ruizi Yang, Junbo Chen, and Jianke Zhu. Mgmap: Mask-guided learning for online vectorized hd map construction. In *CVPR*, 2024.

[19] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *ICLR*, 2023.

[20] Zihao Liu, Xiaoyu Zhang, Guangwei Liu, Ji Zhao, and Ningyi Xu. Leveraging enhanced queries of point sets for vectorized map construction. In *ECCV*, 2024.

[21] Nan Peng, Xun Zhou, Mingming Wang, Xiaojun Yang, Songming Chen, and Guisong Chen. PrevPredMap: Exploring temporal modeling with previous predictions for online vectorized hd map construction. *arXiv preprint arXiv:2407.17378*, 2024.

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

[23] Limeng Qiao, Wenjie Ding, Xi Qiu, and Chi Zhang. End-to-end vectorized hd-map construction with piecewise bezier curve. In *CVPR*, 2023.

[24] Jingyu Song, Xudong Chen, Liupei Lu, Jie Li, and Katherine A Skinner. MemFusionMap: Working memory fusion for online vectorized hd map construction. *arXiv preprint arXiv:2409.18737*, 2024.

[25] 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.

[26] Shuo Wang, Fan Jia, Weixin Mao, Yingfei Liu, Yucheng Zhao, Zehui Chen, Tiancai Wang, Chi Zhang, Xiangyu Zhang, and Feng Zhao. Stream query denoising for vectorized HD-map construction. In *ECCV*, 2024.

[27] 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.

[28] Zhenhua Xu, Kwan-Yee K. Wong, and Hengshuang Zhao. InsMapper: Exploring inner-instance information for vectorized hd mapping. In *ECCV*, 2024.

[29] Tianyuan Yuan, Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. Streammapnet: Streaming mapping network for vectorized online hd map construction. In *WACV*, 2024.

[30] Zhixin Zhang, Yiyuan Zhang, Xiaohan Ding, Fusheng Jin, and Xiangyu Yue. Online vectorized HD map construction using geometry. In *ECCV*, 2024.

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

[32] Yi Zhou, Hui Zhang, Jiaqian Yu, Yifan Yang, Sangil Jung, Seung-In Park, and ByungIn Yoo. Himap: Hybrid representation learning for end-to-end vectorized hd map construction. In *CVPR*, 2024.