PseudoMapTrainer: Learning Online Mapping without HD Maps

Christian Löwens[1,3] Thorben Funke[1] Jingchao Xie[1,4] Alexandru Paul Condurache[2,3]
[1]Bosch Research [2]Automated Driving, Bosch [3]University of Lübeck [4]Technical University of Munich
{christian.loewens, thorben.funke}@bosch.com

Abstract

Online mapping models show remarkable results in predicting vectorized maps from multi-view camera images only. However, all existing approaches still rely on ground-truth high-definition maps during training, which are expensive to obtain and often not geographically diverse enough for reliable generalization. In this work, we propose PseudoMapTrainer, a novel approach to online mapping that uses pseudo-labels generated from unlabeled sensor data. We derive those pseudo-labels by reconstructing the road surface from multi-camera imagery using Gaussian splatting and semantics of a pre-trained 2D segmentation network. In addition, we introduce a mask-aware assignment algorithm and loss function to handle partially masked pseudo-labels, allowing for the first time the training of online mapping models without any ground-truth maps. Furthermore, our pseudo-labels can be effectively used to pretrain an online model in a semi-supervised manner to leverage large-scale unlabeled crowdsourced data. The code is available at github.com/boschresearch/PseudoMapTrainer.

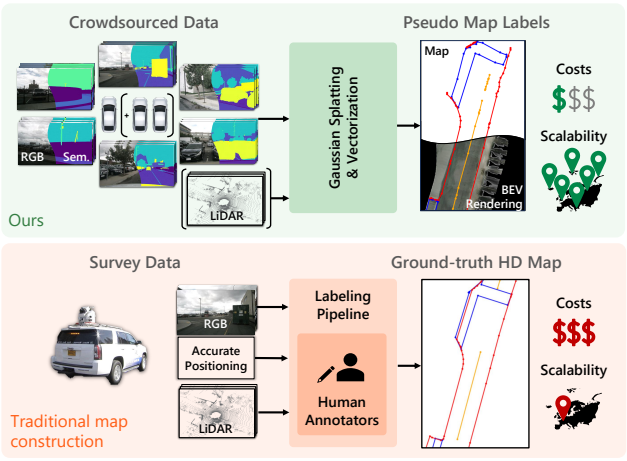

Figure 1. **Motivation for pseudo-labels.** Compared to conventional HD mapping, our method produces maps without human annotations via Gaussian splatting of surrounding camera and segmentation images, greatly reducing costs and enhancing scalability. We then use those labels to train an online mapping model. [·] denotes optional components. Survey vehicle from [34].

1. Introduction

High-definition (HD) maps play a crucial role in autonomous driving, offering precise representations of road geometries, traffic signs, and other essential infrastructure [1, 25]. Traditionally, these maps are constructed from survey vehicles equipped with high-precision sensors and curated by human annotators. While accurate, this process is expensive and faces challenges in maintaining up-to-date information due to the dynamic nature of real-world environments [1]. To mitigate these limitations, the research community has increasingly focused on online mapping methods, which learn to generate maps in real time using only data from vehicle-mounted sensors [16, 17, 19, 26]. A substantial challenge in this domain is the reliance on extensive ground-truth map labels for supervised learning, which are labor-intensive to produce (see Fig. 1) and often not geographically diverse enough for reliable generalization [21].

Given that it is much easier to collect large amounts of unlabeled crowdsourced data from vehicles already on the road, the question arises whether this data can also be leveraged to train online mapping models. Therefore, in this work, we propose PseudoMapTrainer, a novel approach that eliminates the need for ground-truth HD maps even during training. Specifically, we generate pseudo-labels based on road surface reconstruction using Gaussian splatting [14]. We model the road surface as a mesh of Gaussian surfels [10] and optimize it with temporal multi-view camera images. Since surfels can encode not only geometric and color properties but also semantics, we can directly render semantic bird's-eye view (BEV) maps and subsequently derive vectorized pseudo map labels (see Fig. 2).

As our pseudo-labels are derived from observations collected from a single or few vehicle passes, they are inherently incomplete. Thus, a key challenge when training with pseudo-labels is handling occlusions and missing data. To address this, we introduce a mask-aware assignment strat-

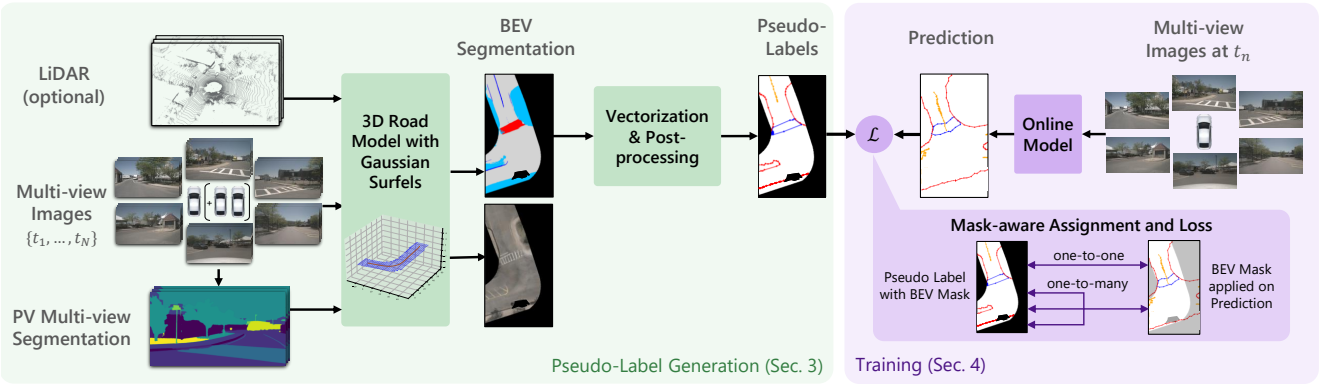

Figure 2. **Label generation and training pipeline of PseudoMapTrainer.** We utilize sensor data from single or multiple trips and infer the corresponding 2D perspective view (PV) segmentations by a pre-trained network to build a coherent meshgrid of Gaussian surfels. Then, we render a BEV segmentation and extract the vectors as our pseudo-labels. Since these labels do not cover the full BEV range (see black regions), we train an online mapping model with a mask-aware one-to-many assignment algorithm and loss function.

egy and loss function enabling robust learning from incomplete labels. Furthermore, we explore the utility of pseudo-labels for semi-supervised learning. By pre-training our model on large-scale pseudo-labeled data and fine-tuning it on a limited set of ground-truth labels, we demonstrate significant performance improvements.

To the best of our knowledge, we propose the first approach to vectorized online mapping without any ground-truth HD map. Our contributions are:

- A pseudo-label generation pipeline based on a road surface reconstruction using Gaussian surfels.
- A mask-aware assignment and loss function that robustly trains online models despite partial BEV observations.
- Improvement of online models with limited access to ground-truth labels via semi-supervised learning.

2. Related Work

Road surface reconstruction. 3D reconstruction methods based on structure-from-motion and multi-view stereo have shown strong performance in well-textured environments [41, 42]. However, these techniques often struggle with flat, low-texture road surfaces. Drawing inspiration from neural radiance fields (NeRF) [31], recent approaches use an explicit mesh to model the road geometry, while elevation [30] and color [45] are represented implicitly. Since these methods exhibit a high computational demand, 3D Gaussian splatting [14] has emerged as an efficient alternative representing scenes using Gaussian spheres. To reconstruct surfaces, these spheres can be reduced to flat surfels [8, 13]. Accordingly, RoGs [10] models the road by a meshgrid of surfels, which we adopt to generate pseudo-labels.

Online mapping. Traditional HD maps constructed from survey vehicles [1] are both expensive and prone to rapid obsolescence in dynamic environments. This has motivated the development of online mapping methods that generate maps in real time using vehicle-mounted sensors. Early approaches produce rasterized BEV segmentations [18, 35, 56] that lack the instance information required for tasks like motion planning [11], while others predict lane instances [28, 36, 37] but do not include other map classes.

To overcome these limitations, vectorized map construction methods have emerged, starting with HDMapNet [16]. VectorMapNet [26] reformulates online mapping as a detection task and thus adopts a one-to-one assignment between prediction and ground-truth elements as proposed by Detection Transformer (DETR) [3]. MapTR [19] further refined the assignment, and subsequent improvements were achieved through adaptations of the DETR-based queries, as seen in MapTRv2 [20] and others [7, 57]. Additional gains have been achieved by incorporating temporal context [4, 50]. MapVR [51] introduces a differentiable rasterization loss as an auxiliary task, which we adapt for training with pseudo-labels. While some work explores semi-supervised BEV segmentation [22, 58], no semi- or unsupervised method has been proposed for vectorized online mapping. Our work aims to fill this gap.

Offline mapping and annotation pipelines. Complementary to online mapping, offline mapping models learn to predict vectorized maps from temporal multi-view sensor data captured during single or multiple trips. After training the offline model with ground-truth maps, it can be deployed in data centers and automatically label large-scale crowdsourced data with further refinements by human annotators. The final maps serve as a more scalable alternative to traditional HD maps [46, 47]. MV-Map [48] learns offline mapping from single-trip data and ground-truth maps by incorporating a NeRF-based approach to enforce multi-view consistency. Building on the road reconstruction of RoMe [30], CAMA [5, 53] generates novel BEV images

that are processed by a BEV-compatible version of MapTR [19] to predict a map. After refinements by human experts, this map is fed back into the model for additional training. A second branch of offline approaches stores historical maps onboard and uses them as priors during online mapping [43, 49, 55]. Although we also propose an offline framework, PseudoMapTrainer does not require ground-truth maps and primarily trains a model to run online.

3. Pseudo-Labels with Gaussian Splatting

The pseudo-labels are generated in four stages, which are shown in Fig. 2. First, 2D semantic segmentation is performed. Second, a 3D meshgrid of Gaussian surfels is initialized to model the color, semantics, and geometry of the road surface. Third, an optimization procedure refines the surfel parameters by aligning the rendered outputs with both the raw camera images and the segmentation. Fourth, post-processing techniques derive the vectorized map elements.

3.1. Task Formulation

Let the full set of images over a sequence of timestamps $t_1, \ldots, t_N$ be $\mathcal{I} = \{I_c(t) \mid t \in \{t_n\}_{n=1}^{N}, c \in \mathcal{C}\}$, where $\mathcal{C}$ denotes the set of available camera views. A pre-trained 2D semantic segmentation network f_{seg} predicts a segmentation for each input image $I \in \mathcal{I}$. Furthermore, the relative ego pose $e(t)$ and all sensor poses at timestamp t are known.

Our goal is to generate a unified 3D road surface representation $\mathcal{M}_\theta$ that explains the camera and segmentation observations across all views and timestamps. Once the optimized parameters θ^* are obtained, a pseudo-label $G(t)$ is generated by a bird's-eye view rendering rend_{BEV} and a subsequent postprocessing post:

$$G(t) = \text{post}\big(\text{rend}_{\text{BEV}}\big(e(t), \mathcal{M}_{\theta^*}\big)\big), \tag{1}$$

where $G(t)$ is the set of the final map elements for timestamp t represented as polyline or polygon vectors.

3.2. Semantics

Accurate semantic segmentation is crucial for generating high-quality pseudo-labels. To achieve this, we utilize a state-of-the-art segmentation model, Mask2Former [6], trained on the Mapillary Vistas V2 dataset [33]. We use its rich class taxonomy to remove segments such as arrows or text on the road surface. Once trained, our segmentation network f_{seg} is deployed to infer semantic segmentations $\mathcal{I}_{\text{seg}} = \{f_{\text{seg}}(I)\}_{I \in \mathcal{I}}$. By shifting the labeling effort from costly HD map annotation to image segmentation, a well-established and scalable task, this approach significantly enhances the adaptability to diverse environments and is robust to different sensor arrangements that limit the generalizability of current online mapping models [52].

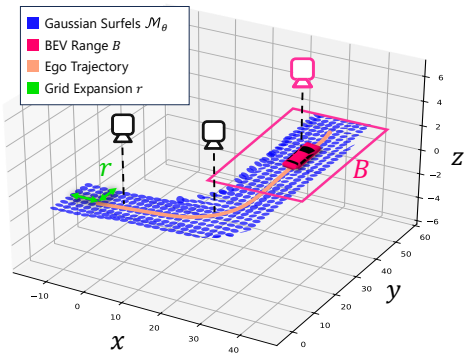

Figure 3. **Surface model and BEV rendering.** We initialize a 3D meshgrid $\mathcal{M}_\theta$ of flat Gaussian surfels along the vehicle's trajectory. After optimization, we render the orthographic BEV images for every vehicle pose $e(t)$.

3.3. Surface Model and Optimization

In 3D Gaussian splatting [14], each Gaussian sphere can be modeled with a 3×3 rotation matrix $\mathbf{R}_{\text{GS}}$ and a 3×3 scaling matrix $\mathbf{S}_{\text{GS}}$. As recent work in surface reconstruction [13] shows, reducing Gaussian spheres to flat surfels by restricting the scaling matrix to $\mathbf{S}_{\text{GS}} = \text{Diag}([s_x, s_y, 0])$ is more effective for modeling surfaces.

Therefore, we model the road as a meshgrid of Gaussian surfels $\mathcal{M}_\theta$, as illustrated in Fig. 3. Following RoGs [10], we initialize the meshgrid along the vehicle poses $e(t)$ of the sequence with an offset r in both the x and y direction. Each Gaussian surfel is parameterized by its 3D center coordinate, 3D orientation, 2D scale, opacity, color, and semantic class probability.

After initialization, the parameters θ of the Gaussian surfels are optimized by minimizing the discrepancy between rendered outputs, comprising both RGB and semantic channels, and the corresponding camera images $\mathcal{I}$ and PV segmentation labels $\mathcal{I}_{\text{seg}}$. To ensure that only road-related information contributes to the optimization, we mask out pixels belonging to non-road classes, such as vehicles, buildings, and pedestrians, resulting in unobserved areas. Since road elements remain static over time, our approach does not require additional compensation for dynamic objects. Optionally, we can use LiDAR data to further improve the z-accuracy of each surfel.

3.4. Postprocessing and Vectorization

To ensure comparability with existing work in online mapping, we focus on the three most commonly used map classes: lane dividers, road boundaries, and pedestrian crossings, representing them as 2D polylines or polygons. Nonetheless, the approach is generalizable to any map class that can be reliably detected by a 2D segmentation network. Moreover, since our underlying road model is inher-

ently 3D, our method could be extended to generate pseudo-labels for 3D online mapping.

BEV rendering. Once the surface optimization is finished, we render a semantic BEV segmentation map. To do this, for each timestamp t, we place a virtual orthographic BEV camera at the xy-position of the vehicle pose $e(t)$, oriented to match its heading, as illustrated in Fig. 3. The camera is positioned to look directly downward along the negative z-axis, capturing the BEV range B on the xy-plane.

Postprocessing. Our postprocessing refines the initial BEV segmentation to produce vectorized map elements. Thereby, we remove small enclosed segments, followed by morphological filtering to remove spurious artifacts and to smooth segmentation boundaries. Fragmented lane markings are connected by dilation and then skeletonized. Road boundaries are extracted by the segment border between the road class and the adjacent outside classes, such as curbs, terrain, and driveways. All rasterized elements are then vectorized with an iterative procedure based on the Ramer-Douglas-Peucker algorithm [9, 38]. We outline all postprocessing details and ablate the main parameters in Supp. A.

3.5. Multi-trip Optimization

As shown in Fig. 2, the final pseudo map labels often exhibit large masked areas due to partial visits and occlusions. While our mask-aware assignment and loss function, introduced in Sec. 4, help to mitigate this issue, the potential of this method is limited as the assignment becomes more arbitrary with a higher mask ratio. Since our pseudo-label generation operates in an offline setting, we are not constrained to a single driving sequence. Instead, we can aggregate observations from multiple trips, potentially crowdsourced from fleet data. This approach not only enhances BEV coverage but also improves label consistency and quality, for instance, by supplementing nighttime sequences with data captured under daylight conditions.

Given multiple driving sequences with known relative poses in a common coordinate system, we initialize our meshgrid along the combined trajectories. The optimization of Gaussian surfels then proceeds in the same manner as for single trips but leverages a significantly larger set of observations, including more camera images, inferred PV segmentations, and LiDAR scans.

4. Training with Pseudo-Labels

In vectorized online mapping, the objective is to predict a set of map elements Q, represented as polygons or polylines, within a BEV range B using multi-view camera images $\{I_c(t)\}_{c\in\mathcal{C}}$ at timestamp t. In contrast, offline mapping uses the sensor measurements of an entire sequence.

For supervised approaches, the training loss is typically based on an optimal one-to-one assignment between a predicted and a ground-truth map element. The primary challenge in training an online mapping model with our pseudo-labels is the handling of incomplete and fragmented map elements since the generated labels suffer from occlusion by non-road objects and viewpoint limitations, as shown in Fig. 4. To address these challenges, we propose a mask-aware one-to-many assignment strategy and a corresponding loss function that accounts for the inherent uncertainty in the data and integrates it into the training process.

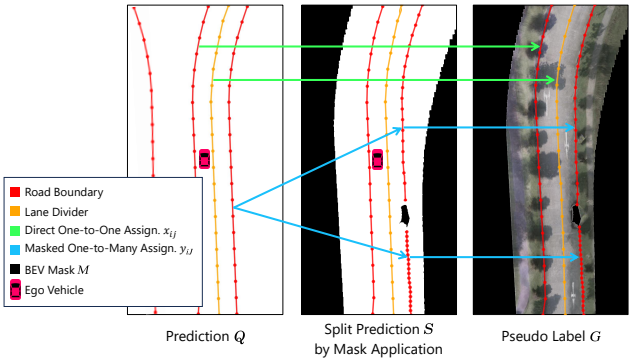

Figure 4. **One-to-many assignment.** In this example, one road boundary of the pseudo-label G is partially interrupted by a parked car, which forms part of the BEV mask M. Thus, we propose a one-to-many assignment algorithm to match one predicted map element to many pseudo-label elements.

4.1. Prediction Masking

Let $Q = \{q_i\}_{i=1}^{|Q|}$ denotes the set of predictions of the online model and $G = \{g_j\}_{j=1}^{|G|}$ the set of pseudo-label elements represented as 2D vectors with $q_i, g_j \in \mathbb{R}^{L\times 2}$. For our mask-aware assignment and loss, we first need to apply the binary BEV mask M to the prediction, as in Fig. 4. Thus, we split q_i into a set of subsegments S^i such that all points and their edges lie within the unmasked region:

$$S^i = \{\text{resample}(s, L) \mid s \in \text{split}(q_i, M), |s| \geq L_\text{m}\} \quad (2)$$

$\text{resample}(s, L)$ standardizes a valid subsegment s to a fixed length L with subsegments shorter than L_m are discarded. In practice, we choose $L_\text{m} = 4$ since a lower value would lead to improper polygon resamples.

4.2. Mask-aware Assignment

To train an online model that predicts map elements across the full BEV range B, we introduce a hybrid assignment algorithm that combines two distinct strategies. First, we perform a standard one-to-one assignment, where each predicted element q_i is matched to a single pseudo-label element g_j based on the assignment cost $c_{\text{o2o}}(q_i, g_j)$. In this step, the BEV mask M is not considered. Second, to handle incomplete pseudo-labels, we introduce a mask-aware one-to-many assignment. In this case, a predicted element

q_i, specifically its unmasked subsegments S^i, is matched to a subset of pseudo-label elements $J \subseteq G_{\text{ind}}$ using the cost $c_{\text{o2m}}(S^i, J)$. Here, $Q_{\text{ind}} = \{i\}_{i=1}^{|Q|}$ and $G_{\text{ind}} = \{j\}_{j=1}^{|G|}$ denote the sets of the corresponding indices. The final assignment is obtained by solving a linear program.

Assignment costs. For the one-to-one assignment costs c_{o2o}, any common cost function can be chosen. We adopt the one by MapVR [51] to be consistent with our loss selection as described in Sec. 4.3. It adds a rendering cost to the class and position costs proposed by MapTR [19].

For the one-to-many assignment costs c_{o2m}, we collect the set $\mathcal{J}$ of all possible subsets of G_{ind}, where all corresponding elements of a subset $J \in \mathcal{J}$ belong to the same class. Now, the one-to-many cost is defined as:

$$c_{\text{o2m}}(S^i, J) = \begin{cases} c_{\text{hungarian}}(S^i, J), & \text{if } |S^i| = |J| \\ \infty, & \text{otherwise} \end{cases} \tag{3}$$

with $c_{\text{hungarian}}$ as the optimal cost based on the local matching $\pi \in \Pi_{\text{local}}$ and the one-to-one cost function $c_{\text{o2o}}(q_i, g_j)$:

$$c_{\text{hungarian}}(S^i, J) = \min_{\pi \in \Pi_{\text{local}}} \sum_{j \in J} c_{\text{o2o}}(s^i_{\pi(j)}, g_j) \tag{4}$$

Here, the optimal local matching π^*_{local} between multiple pseudo-label elements and the prediction subsegments is found with the Hungarian algorithm [15].

Optimal assignment. We solve the final global assignment with binary integer linear programming. It yields an optimal matching with $x^*_{ij}, y^*_{iJ} \in \{0, 1\}$ denoting the direct one-to-one assignment for q_i and g_j and the one-to-many assignment for q_i and the corresponding elements in J, respectively. We summarize the assignment as

$$\pi^*_{\text{global}}(i) = \begin{cases} \{j\}, & \text{if } \exists j \text{ s.t. } x^*_{ij} = 1 \\ J, & \text{if } \exists J \text{ s.t. } y^*_{iJ} = 1 \\ \emptyset, & \text{otherwise} \end{cases}. \tag{5}$$

For the full problem formulation, we refer the reader to Supp. B. In case that no prediction is split into more than one subsegment, i.e. $|S^i| \leq 1 \; \forall q_i$, the problem can also be solved optimally with the Hungarian algorithm by padding G with $\emptyset$ elements such that $|G| = |Q|$. This makes the assignment faster during training.

Note that our one-to-many assignment differs from the one used in MapTRv2 [20], where a single ground-truth element is assigned to multiple predicted auxiliary elements. In contrast, we assign one predicted element to zero, one, or several pseudo-ground-truth elements. Both approaches are complementary, but we exclude the one from MapTRv2 as it dramatically increases the combinatorial complexity of the assignment. While this can be handled by the Hungarian algorithm in MapTRv2, it would increase training times by an order of magnitude when used with our linear program solver, making it impractical.

4.3. Mask-aware Loss

We build upon the losses proposed by MapTRv2 [20] and MapVR [51], extending them to handle partially masked predictions. Predicted elements that are completely masked and also not matched one-to-one are excluded from the loss. The remaining elements form the subset $Q'_{\text{ind}} \subseteq Q_{\text{ind}}$ used for calculating the final loss.

Classification loss. Given the optimal assignment $\pi^*_{\text{global}}(i)$, the classification loss is defined using the Focal loss [23] with the predicted class probability $\hat{p}_i$ and the class label cls of the assigned pseudo-label element:

$$\mathcal{L}_{\text{cls}} = \sum_{i \in Q'_{\text{ind}}} \mathcal{L}_{\text{Focal}} \left(\hat{p}_i, \text{cls}(\pi^*_{\text{global}}(i))\right) \tag{6}$$

Point-wise loss. For a one-to-many matching, a point-wise L1 loss as in MapTR [19] is not straightforward since a single predicted map element may correspond to multiple pseudo-label elements. Thus, we compute the loss exclusively for direct one-to-one assignments where $x^*_{ij} = 1$:

$$\mathcal{L}_{\text{pt}} = \sum_{i \in Q'_{\text{ind}}} \sum_{j \in G_{\text{ind}}} x^*_{ij} \sum_{l=1}^{L} \|q_{i,l} - g_{j,\gamma_j(l)}\|_1. \tag{7}$$

$\gamma_j(l)$ denotes the optimal point-wise assignment for each predicted point to its corresponding pseudo-label point.

Rendering loss. MapVR introduces a differentiable rendering loss, where each map element is first rasterized, and then the Dice loss [32] is computed between prediction and ground-truth rasterizations.

We find this loss particularly well suited for adaptation to our one-to-many assignment as we can render all pseudo-label elements $\{g_j\}_{j \in \pi^*_{\text{global}}(i)}$ assigned to a single prediction q_i into a unified raster. The Dice loss is then computed between this aggregated rasterization and the rasterized prediction of q_i. Additionally, we apply the BEV mask M to exclude unobserved regions, ensuring that the loss is computed only over unmasked grid cells. This rendering loss, $\mathcal{L}_{\text{rend}}$, serves as an effective alternative to the point-wise loss for one-to-many assignments.

Direction loss. We adopt the self-supervised direction loss $\mathcal{L}_{\text{dir}}$ from MapVR, which regularizes the model and prevents overfitting to imperfect pseudo-labels.

Segmentation loss. Following MapTRv2, we adopt the binary segmentation loss in BEV and perspective view (PV) for auxiliary supervision of the BEV and PV features. Analogous to the rendering loss, we mitigate the impact of unobserved regions by applying the mask M to the BEV predictions before calculating the final loss $\mathcal{L}_{\text{BEV}}$.

For the PV segmentation loss $\mathcal{L}_{\text{PV}}$, MapTRv2 projects and rasterizes the ground-truth map to supervise the PV features. Since some datasets like nuScenes [2] contain only

(a) Projected GT in MapTRv2 [20]

(b) Our PV segmentation

Figure 5. **PV segmentation.** The projected ground-truth (GT) map in MapTRv2 suffers from misalignment, while our PV segmentation aligns well as it is derived directly from the image. This provides more precise supervision for the PV features.

2D maps without elevation, the projected map will be misaligned with the actual image, as demonstrated in Fig. 5a.

Given that we have direct access to high-quality PV image segmentation produced by our pre-trained segmentation network f_{seg}, we can leverage this data to produce more accurate labels. In comparison to our map pseudo-labels, the PV segmentation has not undergone the aforementioned postprocessing steps, where some of the information naturally gets lost. The PV segmentation contains particularly valuable information and is also unrestrictedly available compared to the masked BEV. Thus, we extract the map segments of our original segmentation images $\mathcal{I}_{\text{seg}}$ and downsample them to the dimension of the PV feature map using max-pooling. We notice that our segmentations are more aligned with the actual image than the projected 2D ground-truth map utilized in MapTRv2, as shown in Fig. 5.

Depth and final loss. We adopt the depth loss $\mathcal{L}_{\text{depth}}$ from MapTRv2 [20], leveraging LiDAR data for additional depth supervision during training. However, the model itself remains camera-only. The final loss $\mathcal{L}$ is a weighted sum of the losses $\mathcal{L}_{\text{cls}}$, $\mathcal{L}_{\text{pt}}$, $\mathcal{L}_{\text{rend}}$, $\mathcal{L}_{\text{dir}}$, $\mathcal{L}_{\text{BEV}}$, $\mathcal{L}_{\text{PV}}$, and $\mathcal{L}_{\text{depth}}$.

5. Experiments

5.1. Experimental Setup

Dataset. We conduct our experiments on nuScenes [2], a large-scale dataset that provides multi-view images, LiDAR, and HD map annotations. As shown by multiple studies [21, 50], the original training and validation set contain highly overlapping locations, such that results reported on this split demonstrate memorization rather than generalization capabilities. Thus, we train and evaluate all models on the geographically disjoint data split proposed by Lilja et al. [21]. Furthermore, we select the three main map classes, such as lane dividers, road boundaries, and pedestrian crossings, for evaluation. The pseudo-labels are generated from single and multiple trips, in both cases using LiDAR data to supervise the elevation of the road surface.

Metrics. We adopt the average precision (AP) based on the Chamfer distance as introduced by HDMapNet [16] with common thresholds of $\{0.5\,\text{m}, 1.0\,\text{m}, 1.5\,\text{m}\}$ and report the average for all map classes. In addition, we report the inference speed in frames per second (FPS).

Implementation details. All evaluated online models utilize a camera-only sensor setup with a ResNet-50 [12] image backbone. All details regarding the pseudo-label generation can be found in Supp. E.

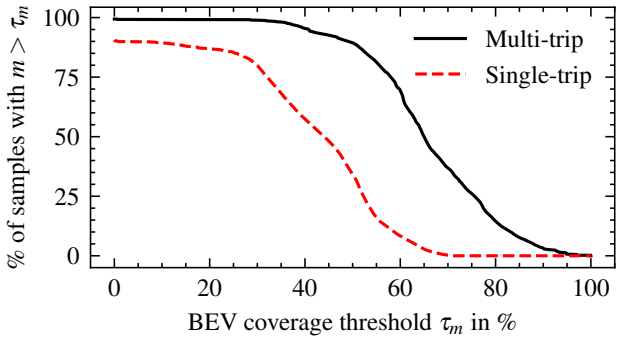

Figure 6. **BEV coverage evaluation.** Comparison of the BEV coverage m between pseudo-labels generated from a single trip and multiple trips on the training set.

Table 1. **Offline performance.** Evaluation of the pseudo-labels on the validation set based on the complete BEV range or on the observed area only.

Pseudo-label collection	Evaluation area	AP ped.	AP div.	AP bdry.	AP *mean*
Single-trip	Full BEV range	9.5	1.9	3.2	4.9
	Observed area only	23.6	6.0	26.8	18.8
Multi-trip	Full BEV range	18.3	1.9	10.7	10.3
	Observed area only	25.8	2.3	26.2	18.1

5.2. How accurate are the pseudo-labels?

We evaluate the quality of our pseudo-labels by their BEV coverage and accuracy compared to ground-truth HD maps.

BEV coverage. Since the pseudo-labels do not cover the entire BEV range B, we evaluate the coverage ratio, denoted as m, which measures the proportion of the unmasked BEV range. We plot the percentage of training samples that exceed a given threshold τ_m in Fig. 6. For single-trip data, we achieve an average coverage ratio of 40.0 %. Extending to multi-trip data increases this ratio to 65.5 %, with almost all samples exceeding 30 % coverage. These results highlight the potential of aggregating crowdsourced data, where multiple vehicles contribute partial observations to construct a more complete map.

Comparison with ground truth. To assess the quality of our pseudo-labels, we compare them against the ground-truth map of the validation set in Tab. 1. We conduct evaluations under two conditions: (1) comparing pseudo-labels against all ground-truth elements, including those in un-

(a) GT (b) Single-trip (c) Multi-trip (d) GT (e) Single-trip (f) Multi-trip (g) GT (h) Single-trip (i) Multi-trip

Figure 7. **Qualitative pseudo-label evaluation.** Comparison of the generated pseudo-labels from a single trip and multiple trips with the ground truth (GT) for three diverse scenes with (g)-(i) showing a low-light scenario. We plot the lane dividers (orange), road boundaries (red), and pedestrian crossings (blue) and use the colored BEV rendering as background for a visual evaluation. In all three cases, we identify inconsistencies for the ground-truth lane dividers, where the pseudo-labels sometimes provide more plausible results. Additional qualitative results are provided in Supp. D.

observed regions, and (2) restricting the evaluation to only the observed regions by applying the BEV mask M to the ground truth, similar to the prediction masking in Sec. 4.1. The latter provides a more precise assessment of the accuracy in the areas covered by the pseudo-labels. In addition, we provide qualitative results in Fig. 7 as well as in Supp. D.

The results in Tab. 1 highlight substantial differences across map classes. Road boundaries, typically located farther from the vehicle's trajectory, are often underrepresented in the pseudo-labels, resulting in lower performance when evaluated across the full BEV range. The lane divider class exhibits particularly poor performance, which can partially be attributed to inconsistencies in the ground-truth annotations, as shown in Fig. 7a, 7d and 7g and noted in previous work [5, 53]. Also, lane markings, being narrow structures, are more prone to being overridden by adjacent road class segments during optimization, causing them to disappear in the final BEV segmentation. This issue arises when camera poses are suboptimal, which is especially common in multi-trip data, see Fig. 7c. This also explains the lower performance for lane dividers coming from multiple trips. Despite these limitations, the pseudo-labels for pedestrian crossings and road boundaries within observed areas achieve comparable performance to the best-performing online models in Tab. 2 trained on ground truth.

5.3. Can online models train on pseudo-labels?

Main results. We compare our approach against common supervised baseline methods in Tab. 2. For MapTRv2 [20], we conduct additional experiments by training the model naively on pseudo-labels from single and multiple trips. To ensure meaningful training, we filter out samples with a BEV coverage below $\tau_m = 50\,\%$. When training MapTRv2 with PseudoMapTrainer, we lower the coverage threshold to $\tau_m = 30\,\%$ for single-trip training as it can handle unobserved areas due to its mask-aware approach.

As expected, MapTRv2 trained on pseudo-labels exhibits a significant performance drop, particularly when using single-trip data. However, incorporating multiple trips improves performance, benefiting from more consistent pseudo-labels and increased BEV coverage. Further gains are achieved when training MapTRv2 with PseudoMapTrainer, which outperforms VectorMapNet [26] on pedestrian crossings and boundary elements - despite never being trained on ground truth. Nonetheless, compared to the best-performing supervised methods, our approach still has a notable performance gap, especially for the lane divider class. This shortfall is attributed to the differences between ground truth and pseudo-labels, as discussed in Sec. 5.2.

Ablation study. We conduct an ablation study on the key components of PseudoMapTrainer, training on pseudo-labels from single trips. The results are summarized in Tab. 3. A naive training of MapTRv2 struggles with pseudo-labels when the BEV coverage falls below 50 %. Introducing the rendering and directional losses proposed by MapVR [51] along with the PV segmentation labels derived directly from our segmentation network, leads to slight performance improvements. A significant gain is observed when incorporating the mask-aware assignment and loss, which enables the model to effectively leverage low-coverage samples. This is particularly evident for boundary elements, which are mostly located in unobserved regions.

5.4. How does it benefit semi-supervised learning?

PseudoMapTrainer can be used to pre-train a model that is later fine-tuned with ground-truth maps in a semi-supervised manner. To evaluate its effectiveness, we pre-train MapTRv2 with PseudoMapTrainer using multi-trip pseudo-labels from the full training set and then fine-tune it on a fraction of the available ground-truth labels. The performance is compared to a purely supervised MapTRv2 baseline in Fig. 8. Our results show that PseudoMap-

Table 2. **Online mapping performance.** Comparison of our method and baselines on the validation set trained on ground truth or pseudo-labels. We highlight the best results per type of training label. † means the results reported by [21]. The FPS results are taken from MapTRv2 [20]. ∗ indicates methods that have access to previous frames for prediction.

Training Labels		Method	Epochs	FPS	AP			
					ped.	div.	bdry.	*mean*
Ground Truth		VectorMapNet [26]†	110	2.2	13.7	13.5	14.9	14.0
		MapTR [19]†	24	15.1	14.4	16.0	26.7	19.0
		MapVR [51]	24	15.1	17.0	16.3	27.6	20.3
		StreamMapNet [50]†∗	24	–	**25.8**	**23.0**	29.5	**26.1**
		MapTRv2 [20]	24	14.1	23.8	19.5	**32.7**	25.4
Pseudo-Labels	Single-trip	MapTRv2 [20]	24	14.1	9.9	3.2	7.0	6.7
		MapTRv2 [20] + PseudoMapTrainer	24	14.1	**12.3**	**3.8**	**8.3**	**8.2**
	Multi-trip	MapTRv2 [20]	24	14.1	14.4	2.6	14.5	10.5
		MapTRv2 [20] + PseudoMapTrainer	24	14.1	**18.1**	**4.1**	**17.4**	**13.2**

Table 3. **Ablation study.** Performance comparison of key components of PseudoMapTrainer, trained on single-trip pseudo-labels.

Training Configuration	AP			
	ped.	div.	bdry.	*mean*
Baseline (MapTRv2, $\tau_m = 50\,\%$)	9.9	3.2	7.0	6.7
+ lower τ_m to $30\,\%$	8.4	2.6	5.0	5.3
+ rendering & direction losses	10.9	3.5	4.0	6.1
+ PV segmentation loss w/o projection	10.9	3.6	4.7	6.4
+ mask-aware assignment & loss	**12.3**	**3.8**	**8.3**	**8.2**

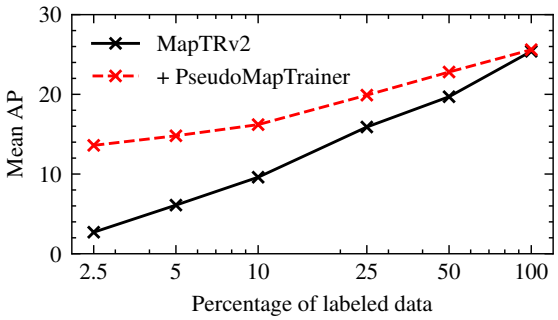

Figure 8. **Semi-supervised training.** Performance comparison between supervised MapTRv2 training and the same model pre-trained on pseudo-labels.

Trainer significantly improves performance, particularly in low-label regimes. This highlights its potential for enhancing online mapping in large-scale scenarios where abundant unlabeled data, such as crowdsourced data, is available, but ground-truth annotations are limited.

5.5. Limitations

Like most camera-based methods, our offline mapping approach is sensitive to challenging lighting conditions, such as nighttime (see Fig. 7h). However, incorporating data from multiple trips under different conditions helps mitigate these limitations, as demonstrated in Fig. 7i.

Merging the data from multiple trips requires highly precise relative positioning between sequences, which we presuppose in this study. In practice, achieving such precision can be challenging, particularly for vehicles relying solely on cameras and consumer-grade positioning systems. However, using additional LiDAR or radar sensors, previous work [24, 29] showed that the relative vehicle poses can be accurately recovered based on unsupervised learned registration methods. Additionally, care must be taken to ensure that merged sequences correspond to timestamps without significant road changes, such as construction. Thus, we propose both a single-trip and a multi-trip approach for pseudo-label generation, providing flexibility depending on sensor availability and environmental stability.

6. Conclusion

We demonstrate the effectiveness of training online mapping models without relying on ground-truth HD maps. Our pseudo-labels also enable efficient pre-training in semi-supervised scenarios with significant performance improvements. This highlights the value of leveraging large-scale crowdsourced data for scalable online mapping.

However, we still see potential for future work. In particular, the lane dividers need to be better preserved through targeted adaptations of the Gaussian optimization process. In addition, incorporating inexpensive SD maps and satellite images could further improve the pseudo-label quality. Another promising direction are pseudo-labels for centerlines, as discussed in Supp. C. For the online model training, self-supervised pre-training presents an opportunity to improve robustness against noisy pseudo-labels.

References

[1] Kaleab Taye Asrat and Hyung-Ju Cho. A comprehensive survey on high-definition map generation and maintenance. *ISPRS*, 13(7):232, 2024. 1, 2

[2] 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*, pages 11621–11631. IEEE, 2020. 5, 6

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020. 2

[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*, pages 90–107. Springer, 2024. 2

[5] Shiyuan Chen, Jiaxin Zhang, Ruohong Mei, Yingfeng Cai, Haoran Yin, Tao Chen, Wei Sui, and Cong Yang. Camav2: A vision-centric approach for static map element annotation. *arXiv preprint 2407.21331*, 2024. 2, 7

[6] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *CVPR*, pages 1290–1299. IEEE, 2022. 3, 2

[7] Sehwan Choi, Jungho Kim, Hongjae Shin, and Jun Won Choi. Mask2map: Vectorized hd map construction using bird's eye view segmentation masks. In *ECCV*, pages 19–36. Springer, 2024. 2

[8] Pinxuan Dai, Jiamin Xu, Wenxiang Xie, Xinguo Liu, Huamin Wang, and Weiwei Xu. High-quality surface reconstruction using gaussian surfels. In *SIGGRAPH*. ACM, 2024. 2

[9] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica*, 10(2):112–122, 1973. 4, 1

[10] Zhiheng Feng, Wenhua Wu, and Hesheng Wang. Rogs: Large scale road surface reconstruction based on 2d gaussian splatting. *arXiv preprint 2405.14342*, 2024. 1, 2, 3

[11] Steffen Hagedorn, Marcel Hallgarten, Martin Stoll, and Alexandru Paul Condurache. The integration of prediction and planning in deep learning automated driving systems: A review. *IEEE T-IV*, 2024. 2

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE, 2016. 6

[13] Binbin Huang, Zehao Yu, Anpei Chen, Andreas Geiger, and Shenghua Gao. 2d gaussian splatting for geometrically accurate radiance fields. In *SIGGRAPH*, pages 1–11. ACM, 2024. 2, 3

[14] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM TOG*, 42(4):1–14, 2023. 1, 2, 3

[15] Harold W Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955. 5

[16] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *ICRA*, pages 4628–4634. IEEE, 2022. 1, 2, 6

[17] Tianyu Li, Peijin Jia, Bangjun Wang, Li Chen, Kun Jiang, Junchi Yan, and Hongyang Li. Lanesegnet: Map learning with lane segment perception for autonomous driving. In *ICLR*, 2024. 1

[18] 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*, pages 1–18. Springer, 2022. 2

[19] 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. 1, 2, 3, 5, 8

[20] 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. 2, 5, 6, 7, 8

[21] 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*, pages 22150–22159. IEEE, 2024. 1, 6, 8

[22] Adam Lilja, Erik Wallin, Junsheng Fu, and Lars Hammarstrand. Exploring semi-supervised learning for online mapping. In *CVPRW*, pages 2477–2487, 2025. 2

[23] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *ICCV*, pages 2980–2988. IEEE, 2017. 5

[24] Quan Liu, Hongzi Zhu, Zhenxi Wang, Yunsong Zhou, Shan Chang, and Minyi Guo. Extend your own correspondences: Unsupervised distant point cloud registration by progressive distance extension. In *CVPR*, pages 20816–20826. IEEE, 2024. 8

[25] Rong Liu, Jinling Wang, and Bingqi Zhang. High definition map for automated driving: Overview and analysis. *Journal of Navigation*, 73(2):324–341, 2020. 1

[26] Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Vectormapnet: End-to-end vectorized hd map learning. In *ICML*, pages 22352–22369. PMLR, 2023. 1, 2, 7, 8

[27] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022. IEEE, 2021. 2

[28] Yueru Luo, Chaoda Zheng, Xu Yan, Tang Kun, Chao Zheng, Shuguang Cui, and Zhen Li. Latr: 3d lane detection from monocular images with transformer. In *ICCV*, pages 7941–7952. IEEE, 2023. 2

[29] Christian Löwens, Thorben Funke, André Wagner, and Alexandru Paul Condurache. Unsupervised point cloud registration with self-distillation. In *BMVC*. BMVA, 2024. 8

[30] Ruohong Mei, Wei Sui, Jiaxin Zhang, Xue Qin, Gang Wang, Tao Peng, Tao Chen, and Cong Yang. Rome: Towards large

scale road surface reconstruction via mesh representation. *IEEE T-IV*, 2024. 2

[31] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*. Springer, 2020. 2

[32] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *3DV*, pages 565–571. IEEE, 2016. 5

[33] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *ICCV*, pages 4990–4999. IEEE, 2017. 3

[34] Oregon Department of Transportation. 3d mobile mapping unit. flickr.com/photos/28364885@N02/24784201084, 2016. CC BY 2.0. 1

[35] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*. Springer, 2020. 2

[36] Maximilian Pittner, Alexandru Condurache, and Joel Janai. 3d-splinenet: 3d traffic line detection using parametric spline representations. In *WACV*, pages 602–611. IEEE, 2023. 2

[37] Maximilian Pittner, Joel Janai, and Alexandru P Condurache. Lanecpp: Continuous 3d lane detection using physical priors. In *CVPR*, pages 10639–10648. IEEE, 2024. 2

[38] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256, 1972. 4

[39] Azriel Rosenfeld and John L Pfaltz. Sequential operations in digital picture processing. *Journal of the ACM*, 13(4):471–494, 1966. 1

[40] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115:211–252, 2015. 2

[41] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*. IEEE, 2016. 2

[42] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*. Springer, 2016. 2

[43] Anqi Shi, Yuze Cai, Xiangyu Chen, Jian Pu, Zeyu Fu, and Hong Lu. Globalmapnet: An online framework for vectorized global hd map construction. *arXiv preprint 2409.10063*, 2024. 3

[44] Satoshi Suzuki et al. Topological structural analysis of digitized binary images by border following. *Computer vision, graphics, and image processing*, 30(1):32–46, 1985. 1

[45] Wenhua Wu, Qi Wang, Guangming Wang, Junping Wang, Tiankun Zhao, Yang Liu, Dongchao Gao, Zhe Liu, and Hesheng Wang. Emie-map: Large-scale road surface reconstruction based on explicit mesh and implicit encoding. In *ECCV*, pages 370–386. Springer, 2024. 2

[46] Deguo Xia, Weiming Zhang, Xiyan Liu, Wei Zhang, Chenting Gong, Jizhou Huang, Mengmeng Yang, and Diange Yang. Dumapnet: An end-to-end vectorization system for city-scale lane-level map generation. In *SIGKDD*, pages 6015–6024, 2024. 2

[47] Deguo Xia, Weiming Zhang, Xiyan Liu, Wei Zhang, Chenting Gong, Xiao Tan, Jizhou Huang, Mengmeng Yang, and Diange Yang. Ldmapnet-u: An end-to-end system for city-scale lane-level map updating. *arXiv preprint 2501.02763*, 2025. 2

[48] Ziyang Xie, Ziqi Pang, and Yu-Xiong Wang. Mv-map: Offboard hd-map generation with multi-view consistency. In *ICCV*, pages 8658–8668. IEEE, 2023. 2

[49] Xuan Xiong, Yicheng Liu, Tianyuan Yuan, Yue Wang, Yilun Wang, and Hang Zhao. Neural map prior for autonomous driving. In *CVPR*, pages 17535–17544. IEEE, 2023. 3

[50] Tianyuan Yuan, Yicheng Liu, Yue Wang, Yilun Wang, and Hang Zhao. Streammapnet: Streaming mapping network for vectorized online hd map construction. In *WACV*, pages 7356–7365. IEEE, 2024. 2, 6, 8

[51] Gongjie Zhang, Jiahao Lin, Shuang Wu, Zhipeng Luo, Yang Xue, Shijian Lu, Zuoguan Wang, et al. Online map vectorization for autonomous driving: A rasterization perspective. In *NeurIPS*, pages 31865–31877. Curran Associates Inc., 2023. 2, 5, 7, 8

[52] Hengyuan Zhang, David Paz, Yuliang Guo, Xinyu Huang, Henrik I Christensen, and Liu Ren. Mapgs: Generalizable pretraining and data augmentation for online mapping via novel view synthesis. *arXiv preprint 2501.06660*, 2025. 3

[53] Jiaxin Zhang, Shiyuan Chen, Haoran Yin, Ruohong Mei, Xuan Liu, Cong Yang, Qian Zhang, and Wei Sui. A vision-centric approach for static map element annotation. In *ICRA*, pages 15861–15867. IEEE, 2024. 2, 7

[54] Tongjie Y Zhang and Ching Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, 1984. 1

[55] Xiaoyu Zhang, Guangwei Liu, Zihao Liu, Ningyi Xu, Yunhui Liu, and Ji Zhao. Enhancing vectorized map perception with historical rasterized maps. In *ECCV*, pages 422–439. Springer, 2024. 3

[56] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *CVPR*, pages 13760–13769. IEEE, 2022. 2

[57] 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*, pages 15396–15406. IEEE, 2024. 2

[58] Junyu Zhu, Lina Liu, Yu Tang, Feng Wen, Wanlong Li, and Yong Liu. Semi-supervised learning for visual bird's eye view semantic segmentation. In *ICRA*, pages 9079–9085. IEEE, 2024. 2