

# MV-Map: Offboard HD-Map Generation with Multi-view Consistency

Ziyang Xie<sup>\*1</sup>  
Fudan University<sup>1</sup>

ziyangxie19@fudan.edu.cn, ziqip2@illinois.edu, yxw@illinois.edu

Ziqi Pang<sup>\*2</sup>  
University of Illinois Urbana-Champaign<sup>2</sup>

## Abstract

While bird’s-eye-view (BEV) perception models can be useful for building high-definition maps (HD-Maps) with less human labor, their results are often unreliable and demonstrate noticeable inconsistencies in the predicted HD-Maps from different viewpoints. This is because BEV perception is typically set up in an “onboard” manner, which restricts the computation and consequently prevents algorithms from reasoning multiple views simultaneously. This paper overcomes these limitations and advocates a more practical “offboard” HD-Map generation setup that removes the computation constraints, based on the fact that HD-Maps are commonly reusable infrastructures built offline in data centers. To this end, we propose a novel offboard pipeline called MV-Map that capitalizes multi-view consistency and can handle an arbitrary number of frames with the key design of a “region-centric” framework. In MV-Map, the target HD-Maps are created by aggregating all the frames of onboard predictions, weighted by the confidence scores assigned by an “uncertainty network.” To further enhance multi-view consistency, we augment the uncertainty network with the global 3D structure optimized by a voxelized neural radiance field (Voxel-NeRF). Extensive experiments on nuScenes show that our MV-Map significantly improves the quality of HD-Maps, further highlighting the importance of offboard methods for HD-Map generation. Our code and model are available at <https://github.com/ZiYang-xie/MV-Map>.

## 1. Introduction

High-definition maps (HD-Maps) play a crucial role in ensuring the safe navigation of autonomous vehicles, by providing essential positional and semantic information about road elements. Ideally, one would expect the process of constructing HD-Maps to be as simple as collecting numerous sensing data while driving and then utilizing an *automatic* algorithm to extract the road elements, as illustrated in Fig. 1. However, the mainstream solutions

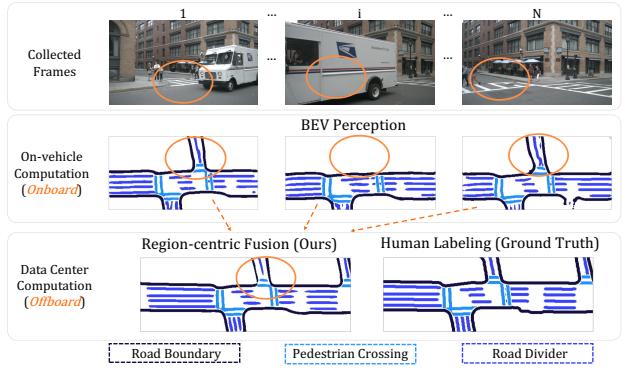


Figure 1: Current *onboard* methods generate unreliable HD-Map predictions that are inconsistent across multiple views due to occlusions or viewpoint changes. By contrast, our *offboard* pipeline constructs a unified and multi-view consistent HD-Map with clearer lanes. Our key design is a *region-centric* framework that aggregates single-frame information for each target HD-Map region.

generally involve human annotators, as seen in widely-used datasets [3, 4, 9, 44]. This design is based on the consideration of the infrastructure role and high re-usability of HD-Maps, which can serve autonomous vehicles for virtually *infinite* times after a *single* construction process. Even so, the expense of manual annotation obstructs the expansion of autonomous driving to new locations, and we aim to develop reliable algorithms that can decrease or replace the need for human labor in HD-Map construction.

Towards this goal, there have been recent attempts that automatically generate HD-Maps using bird’s-eye-view (BEV) perception [12, 16, 18]. However, their results are often unreliable, as illustrated by noticeable inconsistencies in the predicted HD-Maps from different viewpoints (a representative example is in Fig. 1). We argue that *multi-view consistency* is an intrinsic property of HD-Maps, which are rigid and static, and violations of this consistency arise from the fact that existing BEV perception algorithms do not account for all the views explicitly and thus do not align their predictions. This issue further boils down to their *onboard* setting, where the models are only allowed to access com-

<sup>\*</sup>Equal contribution.

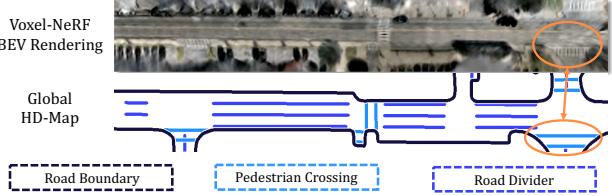


Figure 2: Our Voxel-NeRF reconstructs high-quality 3D structures of the scene. As in the rendering result, the lanes and pedestrian crossings (highlighted) are clear.

puting devices *onboard* in autonomous vehicles and can only handle a single frame or a few neighboring frames.

Given such limitations of the *onboard* setup, we underline a critical yet under-explored *offboard* setup that removes the computation constraints. Our offboard setting aligns well with the *infrastructure* role of HD-Maps: constructing HD-Maps can and should utilize powerful data centers to maximize the fidelity of predictions, thus ensuring the safety and reliability of the virtually infinite usages of HD-Maps. By aggregating information from diverse viewpoints and enhancing consistency, our offboard generation provides a natural improvement. As shown in Fig. 1, having multiple views of a shared region offers richer geometric and semantic cues, as well as improves the completeness of scene understanding, particularly regarding frequent occlusions in urban traffic.

To this end, we propose a framework called *Multiview Map (MV-Map)* that leverages information from every frame’s viewpoint and generates a unified HD-Map consistent with all of them. In contrast to the *frame-centric* design in current onboard methods that merges a fixed number of frames on the input level, we propose a *region-centric* design inspired by “offboard 3D detection” [36] to fully utilize the data from diverse views. Our design connects every HD-Map region with an arbitrary number of input frames covering its area. The pipeline of our framework involves extracting all the HD-Map patches predicted by an off-the-shelf onboard model related to that HD-Map region, and then fusing them into the final result that agrees with all the views, as illustrated in the arrows in Fig. 1. To give more weight to reliable frames, such as those where the target region is clearly visible, we introduce an “*uncertainty network*” as a key component, which assigns confidence scores to onboard results and performs a weighted average of HD-Map patches guided by the confidence.

We further enhance the consensus among all the frames by augmenting the uncertainty network with cross-view consistency information. Our key insight is to learn a coherent 3D structure from diverse views and provide it as an auxiliary input to the uncertainty network. For this purpose, we exploit neural radiance fields (NeRFs) [28], a state-of-the-art approach that represents 3D structures of

scenes. As shown in Fig. 2, our NeRF model synthesizes a high-quality scene structure. Compared with alternative 3D reconstruction strategies like structure from motion (*e.g.*, COLMAP [39]), NeRF is more preferred from a practical perspective, because its runtime grows linearly with the frame number, whereas COLMAP increases quadratically. Moreover, NeRF is *fully self-supervised* and does not require additional annotations, unlike multi-view stereo methods like MVSNet [48]. To further improve the scalability of NeRF, we leverage voxelized variants of NeRF to promote efficiency and propose loss functions that implicitly guide the concentration of NeRF on the near-ground geometry related to HD-Map generation. Additionally, we highlight NeRF’s flexibility and scalability to an arbitrary number of views, making it critical in offboard HD-Map generation.

To summarize, we make the following contributions:

1. We are the *first* to study the problem of learning to generate HD-Maps *offboard*, and we are also the first *vision-oriented* offboard study to our best knowledge.
2. We propose an effective *region-centric* framework MV-Map that can generate a multi-view consistent HD-Map from an arbitrarily large number of frames.
3. We introduce and extend Voxel-NeRF to encode the 3D structure from all frames for HD-Map generation tasks, further guiding the fusion for multi-view consistency.

Large-scale experiments on nuScenes [3] show that MV-Map significantly improves HD-Map quality. Notably, MV-Map can effectively utilize an increasing number of input frames, making it attractive for real-world applications.

## 2. Related Work

**Offboard 3D perception.** The need for large-volume training data encourages developing offboard algorithms. Existing studies mainly focus on predicting the 3D bounding boxes [30, 34, 36, 47]. The most representative “offboard 3D detection” [36] extracts multi-frame point clouds in object tracks and refines the 3D bounding boxes with the “4D” data. Its success heavily relies on the absolute 3D positions of point clouds, where simply overlaying LiDAR points can construct denser surfaces of objects. However, in the HD-Map generation that relies on images, it is not straightforward to accumulate imagery data in the 3D space. To overcome this limitation, we propose region-centric fusion to aggregate multi-frame information and utilize multi-view reconstruction, *e.g.* NeRF, to encode global geometry. Our study is also the *first vision-oriented offboard* pipeline.

**BEV segmentation and HD-Map construction.** Onboard HD-Map construction is closely related to BEV segmentation, as in HDMAPNet [16]. The major challenge in BEV segmentation is to map the image features to the 3D world. The conventional approach leverages inverse perspective warping [1, 2, 7, 33, 37]. BEV perception methods either apply attention to capture the transformation [18, 23],

incorporate depth information [14, 17, 32, 35], or directly query the features from voxels [12]. To better support downstream applications, some recent methods [19, 22] have developed special decoders to generate vectorized HD-Maps. Unlike these *onboard* methods, our proposition is a general *offboard* pipeline that utilizes any off-the-shelf segmentation models as an internal component and refines its results with multi-view consistent fusion.

**Neural radiance fields.** NeRF [28] has shown outstanding capability in 3D reconstruction. Recent work [26, 38, 42, 45] has extended NeRF into large unbounded scenes, such as city-scale NeRF with ego-centric camera settings [38, 42, 45] and improvement from depth-supervised methods [6, 31, 43, 46]. With NeRF’s ability to optimize 3D structures from numerous views, it becomes an ideal method to enforce multi-view consistency for offboard perception. However, as we are the first to adapt NeRFs for HD-Map generation, some important modifications are made. First, we adopt voxel-based NeRF [11, 21, 29, 40, 41] to accelerate the NeRF training by voxelizing the space and encoding the parameters for each position in the voxels. This allows us to reconstruct a huge scene from nuScenes within minutes. In addition, we propose a “total-variance loss” to enhance NeRF’s concentration on the near-ground geometry, which also reflects the shift of concentration from pixel quality to downstream HD-Map generation.

### 3. Offboard HD-Map Generation

Given a sequence of sensor data, the goal of HD-Map generation is to predict the positions and semantics of road elements in the BEV space, including road dividers, road boundaries, and pedestrian crossings.

**Problem statement.** We consider the input of HD-Map generation as  $\mathcal{D} = \{(I_i, P_i)\}_{i=1}^N$ , where  $I_i$  denotes the  $i$ -th sensor frame,  $P_i$  is the set of associated sensor poses, and  $N$  is the total number of frames in the database representing diverse views of a scene captured by a moving ego vehicle. The output is denoted as  $\mathcal{M} = \{M_i\}_{i=1}^N$ , where  $M_i$  is the HD-Map for the region nearby the ego vehicle on frame  $i$ . Following HDMAPNet [16], we define  $M_i$  as a local semantic map on BEV. Note that the aforementioned formulation is agnostic to sensor types. In the main paper, we mainly focus on *vision-oriented* HD-Map generation, and we extend it to leveraging additional LiDAR data in Sec. C (Appendix). Specifically, every frame  $I_i$  contains  $K = 6$  RGB images  $\{I_{i,j}\}_{j=1}^K$  on nuScenes [3], and  $P_i = \{P_{i,j}\}_{j=1}^K$  comprises of the intrinsic and extrinsic matrices of corresponding cameras.

**Offboard vs. onboard settings.** Compared with the conventional onboard setup, our offboard setup offers greater flexibility in terms of speed and computation resources. On-

board HD-Map generation algorithms are often constrained by efficiency requirements and cannot use all the  $N$  frames *in a single run*. By contrast, offboard algorithms are allowed to have access to all the  $N$  frames, and can then leverage the offline setting and abundant computation resources to generate HD-Maps of higher quality.

**From frame-centric to region-centric designs.** There are different strategies to utilize the temporal data from  $N$  frames, similar to offboard 3D detection [36]. A direct solution is *frame-centric* [36], in which we naïvely increase the number of frames for existing *onboard* HD-Map construction methods, typically BEV segmentation models, and extend them to long sequences. While previous work [16, 18] has illustrated the benefit of longer temporal horizons, a multi-frame BEV segmentation model can only handle a fixed number of input frames, and increasing the frame number requires a linear growth in GPU capacity. Therefore, simply scaling up the input frames of existing onboard models is not an effective way of exploiting the offboard data, which often have varying and large frame numbers.

To overcome the limitations of the frame-centric design, we propose a novel *region-centric* design that adaptively aggregates information from an arbitrary number of available frames for each HD-Map region. Our design is inspired by the *object-centric* notion in 3D detection [36], but extends to the task of HD-Map construction. Doing so enables the consensus across frames captured from different viewpoints.

## 4. Method: Multi-view Map

**Overview.** Fig. 3 illustrates the overall framework of our Multi-view Map (MV-Map). An onboard HD-Map model processes every frame  $(I_i, P_i)$  and generates its corresponding BEV feature map  $F_i$  and HD-Map semantics  $S_i$  (Sec. 4.1). Then, an uncertainty network assesses the reliability of the single-frame information  $F_i$  for every region on the HD-Map (Sec. 4.2). Meanwhile, a voxelized NeRF  $f_{\text{NeRF}}$  optimizes a global 3D structure from all  $N$  frames and provides multi-view consistency information to the uncertainty network (Sec. 4.3). The final prediction for every region on the HD-Map is produced by a weighted average of the single-frame semantics  $S_i$ , which enables handling an arbitrary number of frames.

### 4.1. Onboard Model

The onboard model is the entry point of our pipeline. Most existing HD-Map generation methods follow an encoder-decoder design. The encoder generates a BEV feature map  $F_i$  from the input  $(I_i, P_i)$  as  $\text{Encoder}(I_i, P_i) \rightarrow F_i$ , and the decoder converts the feature map  $F_i$  into a semantic map  $S_i$  as  $\text{Decoder}(F_i) \rightarrow S_i$ .

Since our pipeline only requires the BEV feature map  $F_i$  to activate the subsequent modules, MV-Map is agnostic to specific encoder-decoder designs. Without the loss

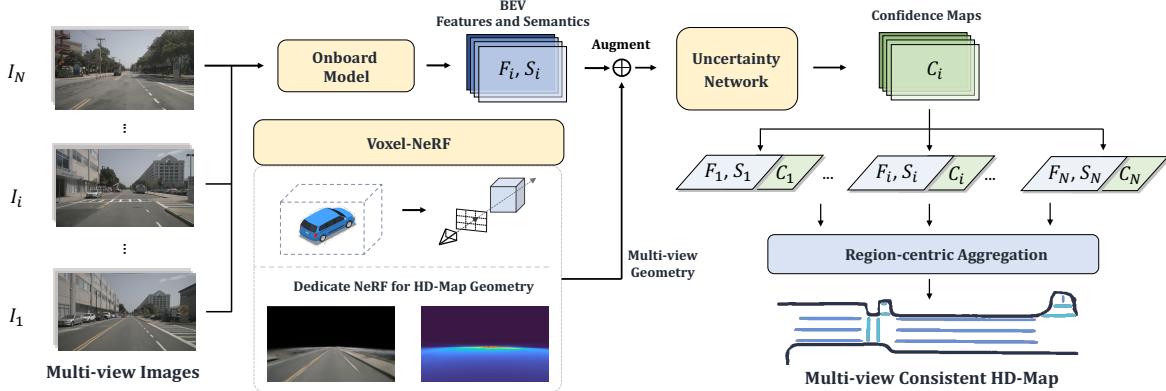


Figure 3: Offboard pipeline of MV-Map. Given an arbitrary number of input frames, MV-Map first leverages an off-the-shelf *onboard model* to generate BEV features and semantic maps for each frame. Then an *uncertainty network* predicts their corresponding confidence maps and guides the region-centric aggregation of a unified HD-Map. Our pipeline further develops a Voxel-NeRF tailored to HD-Map-related 3D structures to augment our pipeline with multi-view geometry.

of generality, here we mainly adopt the encoder in Simple-BEV [12]<sup>1</sup> and use a lightweight convolutional decoder.

**Encoder.** For each frame, a convolutional backbone first converts its  $K$  images  $\{I_{i,j}\}_{j=1}^K$  into 2D image feature maps  $\{F_{i,j}^{2D}\}_{j=1}^K$ . The features are then *lifted* into the 3D world, through a set of voxels that are pre-defined by the encoder with shape  $X \times Y \times Z$  centered around the ego vehicle: the 2D features are bi-linearly sampled for every voxel based on their projected locations on the images, leading to a voxelized 3D feature map  $F_i^{3D}$ . Finally, reducing the Z-axis of  $F_i^{3D}$  produces a BEV feature map  $F_i$  with shape  $X \times Y \times C$ , where  $C$  is the feature dimension.

**Decoder.** Our decoder is a fully-convolutional segmentation head that predicts the logits of semantics from every BEV grid in  $F_i$ . It generates the surrounding HD-Map as the semantic segmentation result  $S_i$  with shape  $X \times Y$ .

**Region-centric extension.** Our *region-centric* design considers each BEV grid as an HD-Map region. If a grid is covered in  $N'$  frames, it receives  $N'$  features and predictions from different viewpoints. MV-Map then fuses the  $N'$  view-specific information to create a multi-view consistent feature for this region, detailed as below.

## 4.2. Global Aggregation via Uncertainty Network

**Region-centric uncertainty-aware fusion.** Our region-centric offboard pipeline learns to aggregate the  $N$  frames of independent HD-Map predictions  $\{S_i\}_{i=1}^N$  into a multi-view consistent prediction for each region. Our key design is to introduce an *uncertainty network*. For the HD-Map predictions from all viewpoints, the uncertainty network assigns a confidence score to each BEV grid, resulting in  $N \times X \times Y$  scores that reflect the pairwise reliability of a viewpoint contributing to an HD-Map region. Specifically,

the uncertainty network takes the BEV features  $\{F_i\}_{i=1}^N$  as input and generates the confidence maps  $\{C_i\}_{i=1}^N$ , with  $C_i$  of shape  $X \times Y$ . In Sec. 4.3, we will describe how we further incorporate global geometry encoded by Voxel-NeRF into the uncertainty network.

We then aggregate the per-frame semantics and confidences into a final HD-Map. Suppose an arbitrary target position  $(x^w, y^w)$  is specified in the world coordinate system, we transform it to the local coordinate system of every frame with poses  $\{P_i\}_{i=1}^N$  and sample the semantic maps and confidence scores at corresponding locations. Finally, the prediction for  $(x^w, y^w)$  is obtained by weighted average of per-frame semantics according to their confidences.

**KL-divergence loss for enhanced uncertainty learning.** In addition to generating confidence scores, we add a multi-layer perceptron (MLP) head to the uncertainty network to infer the KL-divergence between the predicted and ground truth semantics. During training, by encouraging the inferred divergence  $KL^U$  to be close to the true divergence  $KL^G$  between semantics  $S_i$  and  $S_i$ 's ground truth, formally,

$$\mathcal{L}_{KL} = \frac{1}{XY} \sum_{x=1}^X \sum_{y=1}^Y \|KL^G[x, y] - KL^U[x, y]\|_2^2, \quad (1)$$

we *explicitly* differentiate reliable predictions from unreliable ones.

We train the uncertainty network with both a cross-entropy loss between the fusion result and the ground truth semantics at each location  $(x^w, y^w)$  [16] and our auxiliary loss in Eqn. 1. Given that the weighted average operation is differentiable, the gradients can be back-propagated to the confidence scores for updating the uncertainty network.

## 4.3. Voxel-NeRF for Multi-view Consistency

MV-Map further leverages a Voxelized NeRF to effectively construct a *unified* 3D structure of the scene from the

<sup>1</sup>Results based on additional models are in Table D (Appendix).

$N$  frames, which is incorporated with the uncertainty network to improve the multi-view consistency of HD-Maps.

**Voxel-NeRF for traffic scenes.** NeRF [28] represents a 3D scene as a continuous function  $f_{\text{NeRF}} : (\mathbf{x}, \theta) \rightarrow (\mathbf{c}, \sigma)$ , which maps every point  $x$  in the 3D space to its color  $c$  and density  $\sigma$ , relative to the viewing direction  $\theta$ . By explicitly encoding camera projection in the neural rendering process, the learned NeRF model  $f_{\text{NeRF}}$  encodes the 3D geometry of the corresponding scene from input images. Despite the success of the vanilla NeRF, applying it to autonomous driving datasets poses significant challenges, because of the unbounded nature of the scenes and the huge quantities of data involved (*e.g.*, 850 scenes on nuScenes [3]). Therefore, we introduce a voxelized NeRF based on DVGO [41] for better training speed and scalability. Our Voxel-NeRF captures multi-view consistent geometry for outdoor scenes, instead of small objects as in conventional NeRFs. To achieve this, we initialize voxel grids with shape  $X_s \times Y_s \times Z_s$  to cover the entire scene, which is larger than  $X \times Y \times Z$  used in onboard models for single-frame areas. For each camera ray, the neural rendering operation in Voxel-NeRF *concurrently* queries every voxel intersected by the ray. This concurrent querying of voxels significantly accelerates the training of our Voxel-NeRF, *from hours to minutes* for any scene in nuScenes, enabling a reasonable computation budget for MV-Map. More details can be found in Sec. 5.

**Augmenting uncertainty network.** Conceptually, the predicted semantics at a position is more reliable when it resides on the object surfaces. Once NeRF produces a multi-view consistent structure, we can compute the distance between each voxel center and its closest surface. Such a clue can be exploited to evaluate the reliability of semantic maps. To this end, for an arbitrary  $(x, y)$  on BEV, we first recover all the voxel center locations at the BEV coordinate  $(x, y)$  as  $L_{\text{Voxel}} = \{(x, y, z_i)\}_{i=1}^Z$  and then compute their corresponding pixel locations on the images  $L_{\text{Image}} = \{(x_i^p, y_i^p)\}_{i=1}^Z$ . By volume rendering along the camera rays crossing these pixels<sup>2</sup>,  $f_{\text{NeRF}}$  reconstructs the 3D positions of these pixels, denoted as  $L_{\text{NeRF}} = \{(x_i^R, y_i^R, z_i^R)\}_{i=1}^Z$ , which are generally the intersections between their camera rays and surfaces. By calculating  $\Delta L_{\text{NV}} = \{(x_i^R - x, y_i^R - y, z_i^R - z_i)\}_{i=1}^Z$ , we assess the consistency between voxel centers and the global 3D structure. Finally, we employ an MLP upon  $\Delta L_{\text{NV}}$  and concatenate its output with the BEV feature, which is then used as the augmented input to the uncertainty network.

**Dedicating NeRF for HD-Maps with total-variance loss.** Note that *our objective is to facilitate HD-Map generation, rather than optimizing rendering quality*. With the majority of HD-Map elements situated on the ground, we modify the NeRF to focus less on the quality of pixels in the air. To this end, we introduce a simple yet effective “total-variance

loss” that guides the optimization of near-ground geometry *implicitly*. This total-variance loss  $\mathcal{L}_{\text{TV}}$  is obtained by accumulating the total-variance  $\text{TV}(\cdot)$  at each BEV position:

$$\mathcal{L}_{\text{TV}} = -\frac{1}{X_s Y_s} \sum_{x=1}^{X_s} \sum_{y=1}^{Y_s} \text{TV}(x, y). \quad (2)$$

Here the total-variance  $\text{TV}(\cdot)$  is defined as the L2-norm of the differences of occupancies along the Z-axis, given by

$$\text{TV}(x, y) = \|O[x, y, 2:Z_s] - O[x, y, 1:Z_s - 1]\|_2, \quad (3)$$

where  $O[x, y, z]$  represents the density of voxel  $(x, y, z)$  predicted by NeRF and  $\|\cdot\|_2$  denotes the L2-norm.

We emphasize the “negative” sign in Eqn. 2 that indicates “maximizing” the variance, because an accurate ground plane has a *peak* distribution of voxel occupancy on the Z-axis instead of a *uniform* one. TV-loss enables Voxel-NeRF to assign larger densities to the ground plane than transient objects, leading to high-quality 3D structures as in Fig. 2.

#### 4.4. Training and Inference

The procedure of our offboard pipeline follows three steps: (1) we adopt an existing onboard model, (2) train Voxel-NeRF on sequences, and (3) train and infer the uncertainty network. We describe these steps in order and leave detailed configurations in Sec. G (Appendix).

**Onboard model.** As MV-Map is agnostic to the choice of onboard models (Sec. 4.1), here we adopt an *off-the-shelf* BEV segmentation model and freeze its parameters during both training and inference stages of the offboard pipeline.

**Voxel-NeRF.** We train the Voxel-NeRF for all sequences in our training and validation datasets, using both conventional photometric loss and our total-variance loss (Sec. 4.3). Note that our NeRF training is entirely *self-supervised and does not require any annotations*.

**Uncertainty network.** The *region-centric* design enables the uncertainty network to handle varying frame numbers of offboard data. In practice, however, the GPU capacities and batching during training limit the network to a fixed and restricted frame number. To overcome this issue, we adopt the solution from video-based tasks (*e.g.*, 2D multi-object tracking [27, 50]), where models are trained on *short video clips* but are inferred iteratively on *unbounded sequences*.

Similarly, given the input  $N$  frames of a scene, the uncertainty network is trained with samples containing  $M$  ( $M < N$ ) adjacent frames to fit into limited GPU memory. The loss is a weighted sum of our KL-divergence loss and a BEV segmentation loss (Sec. 4.2). During inference, we apply the uncertainty network to all the  $N$  frames independently and use region-centric aggregation to fuse single-frame semantics into a unified HD-Map.

<sup>2</sup>Ray casting is described in Sec. B.2 (Appendix).

Table 1: Comparison with state-of-the-art HD-Map generation methods on nuScenes [3]. “\*” means the results reported in HDMapNet [16]. “Average Fusion” is an offboard baseline explained in Sec. 5.2. The quantitative results indicate that our MV-Map has significant benefits to HD-Maps generation and outperforms baseline offboard approaches.

Setup	Method	Divider	mIoU (Short-range / Long-range)		
			Ped Crossing	Boundary	All
Onboard	IPM(B)*	25.5 / -	12.1 / -	27.1 / -	21.6 / -
	IPM(B+C)*	38.6 / -	19.3 / -	39.3 / -	32.4 / -
	VPN*	36.5 / -	15.8 / -	35.6 / -	29.3 / -
	Lift-Splat-Shoot[35]*	38.3 / -	14.9 / -	39.3 / -	30.8 / -
	HDMapNet [16] (Surr)	40.6 / 33.9	18.7 / 19.4	39.5 / 34.9	32.9 / 29.4
	Onboard Model (Ours)	46.4 / 39.3	29.7 / 26.4	48.1 / 39.1	41.4 / 35.0
Offboard	Average Fusion	48.86 / 42.83	31.55 / 24.75	51.98 / 43.91	44.13 / 37.16
	MV-Map (Ours)	<b>50.87 / 48.15</b>	<b>34.52 / 33.34</b>	<b>55.64 / 50.28</b>	<b>47.01 / 43.92</b>

## 5. Experiments

### 5.1. Dataset and Implementation Details

**Dataset.** We conduct experiments on the large-scale autonomous driving dataset: nuScenes [3]. It contains 850 videos with 28,130 and 6,019 frames for training and validation, respectively. On each timestamp, six surrounding cameras collect high-resolution images as input.

**Evaluation metrics.** Following prior work [16], we compute the intersection-over-union (IoU) for HD-Map categories: divider, pedestrian crossing, and road boundaries. To highlight the challenge of predicting a scene-scale HD-Map, our evaluation adopts both a *short-range* setting [16] covering  $60\text{m} \times 30\text{m}$  and a new *long-range* setting covering  $100\text{m} \times 100\text{m}$ , which aligns with the common perception range in self-driving [8, 10]. Without further mentioning, we conduct our ablation studies under the more challenging long-range setting.

**Implementation details.** We follow the training and inference settings in Sec. 4.4 and discuss the details in Sec. G (Appendix). We emphasize that MV-Map is scalable to large volumes of offboard data. Within 15 minutes on a single A40 GPU, our Voxel-NeRF can optimize the 3D structure from each nuScenes sequence, which typically has over 1k images covering regions with an average length of  $\sim 300\text{m}$ , less than 1 second per frame. In comparison, COLMAP [39] may take several hours or even days. Moreover, our uncertainty network is trained on the samples with  $M = 5$  adjacent frames to fit into our GPU memory, but it can jointly handle all the frames ( $\sim 40$ ) in a nuScenes sequence during the inference stage, as explained in Sec. 4.4.

### 5.2. Comparison with State-of-the-Art Methods

As our work represents the *first* study on offboard HD-Map generation, there are no readily available competing methods. Additionally, our MV-Map can utilize any off-the-shelf onboard model as its internal component. To ensure a meaningful and fair comparison, we organize the experimental results and analysis in Table 1 as follows.

First, our onboard model adopts the simple-yet-effective

Table 2: The components in MV-Map effectively improve HD-Map generation step by step. We analyze the uncertainty network (UN) and KL divergence loss ( $\mathcal{L}_{\text{KL}}$ ) discussed in Sec. 4.2, and Voxel-NeRF (NeRF) and total-variance loss ( $\mathcal{L}_{\text{TV}}$ ) discussed in Sec. 4.3.

ID	Offboard Components	Divider	mIoU							
			UN	$\mathcal{L}_{\text{KL}}$	NeRF	$\mathcal{L}_{\text{TV}}$	Crossing	Boundary	All	
1	Onboard Model	39.30	26.44	39.10	34.95					
2	Average fusion	42.83	24.75	43.91	37.16					
3	✓						46.90	30.30	49.07	42.09
4	✓	✓					47.38	31.11	49.53	42.67
5	✓	✓	✓				47.64	32.36	49.67	43.22
6	✓		✓	✓			48.01	32.65	50.12	43.59
7	✓	✓	✓	✓	✓		<b>48.15</b>	<b>33.34</b>	<b>50.28</b>	<b>43.92</b>

design from SimpleBEV [12]. As shown in the “onboard” lines, our onboard model already *consistently* outperforms previous baselines in both short-range and long-range settings. Second, our MV-Map brings a significant improvement of  $\sim 7\%$  mIoU compared with our already effective onboard model. Notably, our offboard method is better than HDMapNet [16] by around 50% with over 15% IoU increase on all the categories. Finally, we develop an offboard baseline algorithm called “Average Fusion.” It does not consider the quality of different viewpoints and performs region-centric aggregation by equally averaging the single-frame semantic maps. Compared with “Average fusion,” our MV-Map still improves the HD-Map quality by a large margin of over  $\sim 7\%$  mIoU under the long-range setting.

### 5.3. Ablation Studies

**MV-Map Components.** We quantify the improvement from each offboard module in Table 2.

(1) **Region-centric fusion baseline.** Beginning from the onboard model (line 1), we first apply average fusion (line 2) to it (discussed in Sec. 5.2) as a baseline. The improvement indicates that our region-centric design indeed helps by fusing numerous frames into a unified HD-Map. (2) **Uncertainty network.** Replacing the average fusion (line 2) with the uncertainty network (line 3) enables larger contributions from more reliable frames and the  $\sim 5\%$  increase in mIoU proves that assessing quality is critical for better HD-Map quality. (3) **KL**

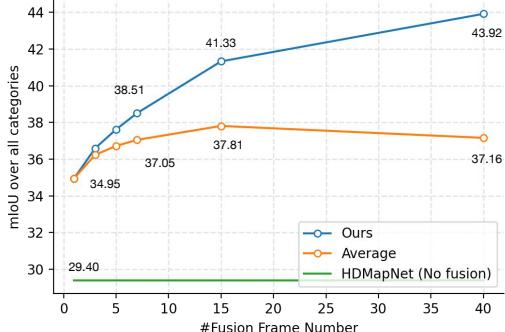


Figure 4: Our MV-Map can significantly benefit from more input frames, which is attractive for offboard applications. Notably, the performance of the “average fusion” baseline saturates and even decreases with more input frames.

**divergence loss.** The  $\sim 0.5\%$  mIoU on using KL divergence loss or not (line 3 and line 4) supports the value of explicitly supervising the uncertainty network. (4) **Voxel-NeRF.** Adding NeRF to a full-fledged uncertainty network further improves the mIoU (line 4 and line 5). In the category-level analysis, we highlight that NeRF is critical for the fusion *especially on the challenging structures with smaller regions*, e.g., pedestrian crossings. This evidence proves the importance of global geometry in multi-view consistency. (5) **Total-variance loss.** Utilizing it further boosts the performance in all the scenarios, validating our effort to dedicate NeRFs for the downstream HD-Map generation.

**Scaling to more frames.** We demonstrate that our fusion strategy can handle and significantly benefit from a larger number of frames, which is critical for offboard HD-Map generation. We evaluate our offboard framework under varied input frames in Fig. 4. MV-Map can utilize all the keyframes (40 frames) in nuScenes and this number is only bounded by the sequence length. As clearly shown in the blue curve of Fig. 4, MV-Map benefits from more frames, indicating its *scalability* for offboard scenarios, especially compared with the average fusion baseline, whose performance drops after using more than 15 frames. This indicates that our region-centric fusion strategy is able to *reason the complementary regions among the frames*, instead of blindly averaging them all.

**Qualitative comparison.** We visualize the generated HD-Maps in Fig. 5. As clearly shown, MV-Map corrects the artifacts from onboard models and achieves better completeness and details. In addition, the HD-Maps generated offboard have high fidelity compared with the ground truth, especially in the center regions covered by more frames.

**Analyzing KL-divergence and confidence scores.** We empirically analyze the output of the uncertainty network in Fig. 6. As for the confidence scores, we observe that they indeed decrease the contributions of unreliable regions, such as the part with occlusion in Fig. 6a, highlighted with solid

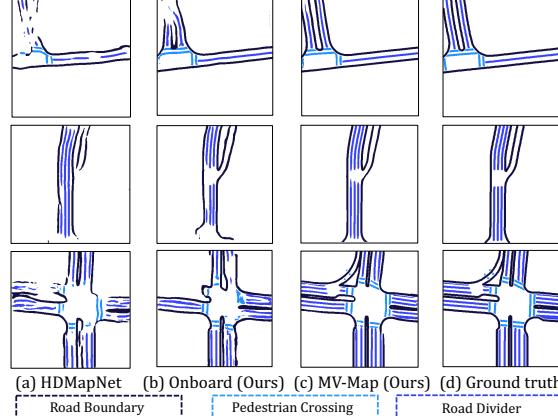


Figure 5: Qualitative comparison in the long-range settings. HD-Map generated offboard has significantly better quality by fixing the artifacts of the onboard model.

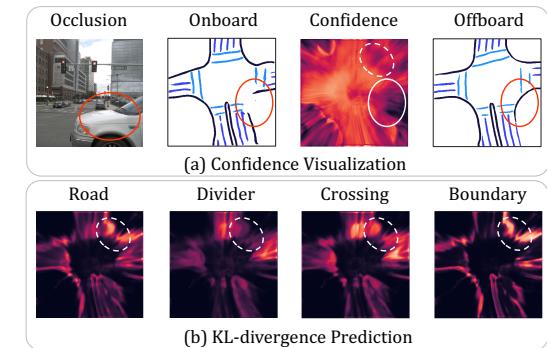


Figure 6: The confidence scores and KL divergence predicted by the uncertainty network *capture the challenging regions* (solid circle) and *correspond to each other* (dashed circle). (a) HD-Map prediction and corresponding confidence from our uncertainty network. Darker colors indicate smaller confidence values. (b) Predicted KL divergence between the prediction and ground truth label. Darker colors indicate smaller KL divergence values.

circles. Additionally, we transfer the KL-divergence prediction head to the validation set and find the predictions reasonably reflect the differences between the predictions and ground truth, as in Fig. 6b. We further notice that the regions with higher KL-divergence values (Fig. 6a) also have lower confidences (Fig. 6b), highlighted with dashed circles.

**Using geometric information from data-driven priors.** Our Voxel-NeRF offers geometric information in a *fully self-supervised* manner. Meanwhile, our MV-Map framework is general and can leverage alternative approaches for providing geometric information, such as *learning data-driven priors* from large-scale datasets. We investigate this type of approach here and consider representative monocular depth estimators that are learned off-the-shelf in a *supervised* manner. Specifically, we replace the rendering process of Voxel-NeRF with the results from *NewCRFs* [49] (de-

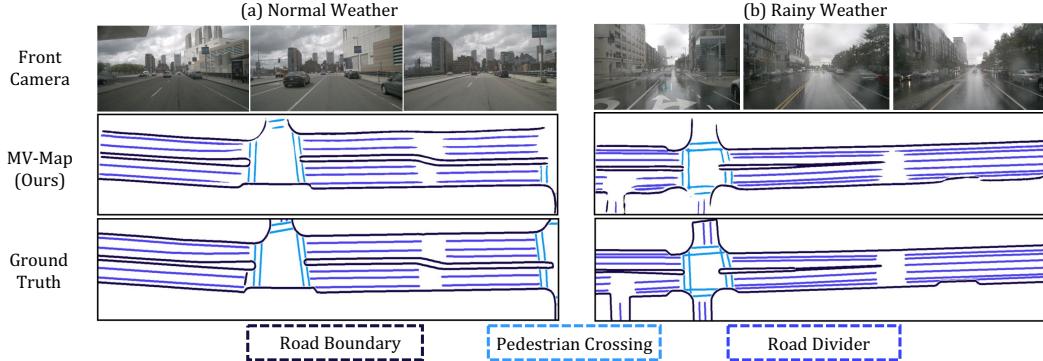


Figure 7: Visualization of a **unified, scene-scale HD-Map** through our MV-Map. It can fuse numerous frames and generate global-scale HD-Maps with high quality. It is also *robust* under different weather conditions and complex road topology.

Table 3: MV-Map is a general framework: In addition to NeRF, MV-Map can benefit from data-driven priors of geometric information; here we use monocular depth estimation [49] as an example. “UN-Only:” using the uncertainty network without augmentation of 3D structural information. Then we separately incorporate mono-depth or NeRF to it.

Methods	mIoU			
	Divider	Ped Crossing	Boundary	All
Average Fusion	42.83	24.75	43.91	37.16
MV-Map (UN-Only)	47.38	31.11	49.53	42.67
MV-Map (Mono-Depth)	48.04	32.96	50.08	43.69
MV-Map (NeRF)	<b>48.15</b>	<b>33.34</b>	<b>50.28</b>	<b>43.92</b>

Table 4: Comparison between fusing BEV feature maps  $F_i$  and semantic maps. We choose to fuse semantic maps in Sec. 4.2 because of its better performance.

Fusion	mIoU			
	Divider	Ped Crossing	Boundary	All
Onboard	39.30	26.44	39.10	34.95
BEV feature	45.88	33.06	46.38	41.77
Semantic	<b>48.15</b>	<b>33.34</b>	<b>50.28</b>	<b>43.92</b>

tails in Sec. G, Appendix). As in Table 3, monocular depth can improve the uncertainty fusion as well (line 2 and line 3). We further notice that NeRF performs slightly better because it *encodes multiple views consistently in a shared 3D structure*, while monocular depth considers each view independently and suffers from scale variation across frames. Encouraged by the benefits of these two distinct types of geometric information, future work is to combine NeRF with learnable priors into our framework.

**Fusing semantics versus BEV features.** Our region-centric framework performs weighted averages over the semantic maps  $S_i$  instead of the BEV features  $F_i$ . In Table 4, we justify our design choices, where fusing BEV features is worse than fusing semantic maps. The main reason is the domain shift between training and inference when we have numerous input frames of offboard data. Furthermore, fusing BEV features is also less practical for requiring significantly more disk space to store high-dimensional features.

#### 5.4. Globally Consistent HD-Map Generation

Our offboard MV-Map can handle numerous frames. Its application is to expand the range of HD-Map generation from a local region around the ego-vehicle to a global region covering all the input frames, which saves the labor in stitching multiple local predictions in the real world. Our global maps in Fig. 7 demonstrate high fidelity for complex topology in two challenging scenes. While some regions do not match the ground truth, we argue that these regions fall outside the collected frames and perception ranges, which are beyond the scope of offboard algorithms. Thus, MV-Map can construct high-quality HD-Maps.

## 6. Conclusion

Regarding the infrastructure role of HD-Maps, we propose a novel *offboard* HD-Map generation setup to address the unreliability of *onboard* BEV perception. By removing the computation constraints, the models are allowed to reason all the frames altogether and construct multi-view consistent HD-Maps. Concretely, we propose an offboard HD-Map generation framework called MV-Map. To address numerous frames, MV-Map designs region-centric aggregation to unify the HD-Maps from all the frames. The key design is an uncertainty network that weighs the contribution of different frames and utilizes a Voxel-NeRF to provide multi-view consistent 3D structural information. Experiments imply that MV-Map is scalable to large volumes of offboard data and significantly improves the HD-Map quality. We hope that our framework can become an effective augmentor for onboard algorithms and also inspire future research on offboard problems.

**Limitations and future work.** Although our Voxel-NeRF improves the offboard pipeline in a scalable way, several challenges still present, including moving objects in traffic scenes and exploiting data-driven priors for better geometric information. In addition, we seek to link our work with auto-labeling and compare it with human annotation quality, so as to explore more potential applications such as autonomous vehicle navigation and urban planning.

## References

- [1] Mohamed Aly. Real time detection of lane markers in urban streets. In *IEEE Intelligent Vehicles Symposium*, 2008. 2
- [2] Massimo Bertozzi and Alberto Broggi. Real-time lane and obstacle detection on the GOLD system. In *IEEE Intelligent Vehicles Symposium*, 1996. 2
- [3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giacarlo Baldan, and Oscar Beijbom. nuScenes: A multi-modal dataset for autonomous driving. In *CVPR*, 2020. 1, 2, 3, 5, 6, 12
- [4] Ming-Fang Chang, John W Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3D tracking and forecasting with rich maps. In *CVPR*, 2019. 1
- [5] 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. 14
- [6] Kangle Deng, Andrew Liu, Jun-Yan Zhu, and Deva Ramanan. Depth-supervised NeRF: Fewer views and faster training for free. In *CVPR*, 2022. 3, 12
- [7] Liuyuan Deng, Ming Yang, Hao Li, Tianyi Li, Bing Hu, and Chunxiang Wang. Restricted deformable convolution-based road scene semantic segmentation using surround view cameras. In *ITSC*, 2020. 2
- [8] Hao Dong, Xianjing Zhang, Xuan Jiang, Jun Zhang, Jintao Xu, Rui Ai, Weihao Gu, Huimin Lu, Juho Kannala, and Xie Yuanli Chen. SuperFusion: Multilevel LiDAR-camera fusion for long-range HD map generation and prediction. *arXiv preprint arXiv:2211.15656*, 2022. 6
- [9] Scott Ettinger, Shuyang Cheng, Benjamin Caine, Chenxi Liu, Hang Zhao, Sabeek Pradhan, Yuning Chai, Ben Sapp, Charles R. Qi, Yin Zhou, Zoey Yang, Aur’elien Chouard, Pei Sun, Jiquan Ngiam, Vijay Vasudevan, Alexander McCauley, Jonathon Shlens, and Dragomir Anguelov. Large scale interactive motion forecasting for autonomous driving: The Waymo open motion dataset. In *ICCV*, 2021. 1
- [10] Lue Fan, Yuxue Yang, Feng Wang, Naiyan Wang, and Zhaoxiang Zhang. Super sparse 3D object detection. *arXiv preprint arXiv:2301.02562*, 2023. 6
- [11] Sara Fridovich-Keil, Alex Yu, Matthew Tancik, Qinrong Chen, Benjamin Recht, and Angjoo Kanazawa. Plenoxels: Radiance fields without neural networks. In *CVPR*, 2022. 3
- [12] Adam W Harley, Zhaoyuan Fang, Jie Li, Rares Ambrus, and Katerina Fragkiadaki. Simple-BEV: What really matters for multi-sensor BEV perception? *arXiv preprint arXiv:2206.07959*, 2022. 1, 3, 4, 6, 13, 14
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 14
- [14] Anthony Hu, Zak Murez, Nikhil Mohan, Sofía Dudas, Jeffrey Hawke, Vijay Badrinarayanan, Roberto Cipolla, and Alex Kendall. FIERY: Future instance prediction in Bird’s-Eye View from surround monocular cameras. *arXiv preprint arXiv:2104.10490*, 2021. 3
- [15] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. PointPillars: Fast encoders for object detection from point clouds. In *CVPR*, 2019. 12
- [16] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. HDMapNet: An online hd map construction and evaluation framework. In *ICRA*, 2022. 1, 2, 3, 4, 6, 11, 13, 14
- [17] 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*, 2023. 3
- [18] Zhiqi Li, Wenhui 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. 1, 2, 3
- [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. 3
- [20] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *arXiv preprint arXiv:1708.02002*, 2017. 14
- [21] Lingjie Liu, Jatao Gu, Kyaw Zaw Lin, Tat-Seng Chua, and Christian Theobalt. Neural sparse voxel fields. In *NeurIPS*, 2020. 3
- [22] Yicheng Liu, Yuan Yuantian, Yue Wang, Yilun Wang, and Hang Zhao. VectorMapNet: End-to-end vectorized HD map learning. *arXiv preprint arXiv:2206.08920*, 2022. 3
- [23] Zhijian Liu, Haotian Tang, Alexander Amini, Xingyu Yang, Huizi Mao, Daniela Rus, and Song Han. BEVFusion: Multi-task multi-sensor fusion with unified Bird’s-Eye View representation. In *ICRA*, 2023. 2
- [24] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 14, 15
- [25] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *ICLR*, 2019. 14
- [26] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. NeRF in the wild: Neural radiance fields for unconstrained photo collections. In *CVPR*, 2021. 3, 11
- [27] Tim Meinhardt, Alexander Kirillov, Laura Leal-Taixe, and Christoph Feichtenhofer. TrackFormer: Multi-object tracking with transformers. In *CVPR*, 2022. 5
- [28] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 2, 3, 5, 11
- [29] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM Trans. Graph.*, 2022. 3
- [30] Mahyar Najibi, Jingwei Ji, Yin Zhou, Charles R Qi, Xinchen Yan, Scott Ettinger, and Dragomir Anguelov. Motion inspired unsupervised perception and prediction in autonomous driving. In *ECCV*, 2022. 2
- [31] Thomas Neff, Pascal Stadlbauer, Mathias Parger, Andreas Kurz, Joerg H. Mueller, Chakravarty R. Alla Chaitanya, An-

- ton S. Kaplanyan, and Markus Steinberger. DONeRF: Towards real-time rendering of compact neural radiance fields using depth oracle networks. 2021. 3, 11
- [32] Mong H. Ng, Kaahan Radia, Jianfei Chen, Dequan Wang, Ionel Gog, and Joseph E Gonzalez. BEV-Seg: Bird’s Eye View semantic segmentation using geometry and semantic point cloud. *arXiv preprint arXiv:2006.11436*, 2020. 3
- [33] Bowen Pan, Jiankai Sun, Ho Yin Tiga Leung, Alex Andonian, and Bolei Zhou. Cross-view semantic segmentation for sensing surroundings. *IEEE Robotics and Automation Letters*, 2020. 2
- [34] Ziqi Pang, Zhichao Li, and Naiyan Wang. Model-free vehicle tracking and state estimation in point cloud sequences. In *IROS*, 2021. 2
- [35] Jonah Philion and Sanja Fidler. Lift, Splat, Shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3D. In *ECCV*, 2020. 3, 6
- [36] Charles R Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3D object detection from point cloud sequences. In *CVPR*, 2021. 2, 3, 12
- [37] Lennart Reiher, Bastian Lampe, and Lutz Eckstein. A Sim2Real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in Bird’s Eye View. In *ITSC*, 2020. 2
- [38] Konstantinos Rematas, Andrew Liu, Pratul P. Srinivasan, Jonathan T. Barron, Andrea Tagliasacchi, Tom Funkhouser, and Vittorio Ferrari. Urban radiance fields. In *CVPR*, 2022. 3
- [39] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 2, 6
- [40] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Direct voxel grid optimization: Super-fast convergence for radiance fields reconstruction. In *CVPR*, 2022. 3, 14
- [41] Cheng Sun, Min Sun, and Hwann-Tzong Chen. Improved direct voxel grid optimization for radiance fields reconstruction. *arXiv preprint arXiv:2206.05085*, 2022. 3, 5
- [42] Matthew Tancik, Vincent Casser, Xinchen Yan, Sabeek Pradhan, Ben Mildenhall, Pratul Srinivasan, Jonathan T. Barron, and Henrik Kretzschmar. Block-NeRF: Scalable large scene neural view synthesis. 2022. 3, 11
- [43] Yi Wei, Shaohui Liu, Yongming Rao, Wang Zhao, Jiwen Lu, and Jie Zhou. NerfingMVS: Guided optimization of neural radiance fields for indoor multi-view stereo. In *ICCV*, 2021. 3
- [44] Benjamin Wilson, William Qi, Tanmay Agarwal, John Lambert, Jagjeet Singh, Siddhesh Khandelwal, Bowen Pan, Ratnesh Kumar, Andrew Hartnett, Jhony Kaesemeyer Pontes, Deva Ramanan, Peter Carr, and James Hays. Argoverse 2: Next generation datasets for self-driving perception and forecasting. *arXiv preprint arXiv:2301.00493*, 2023. 1
- [45] Ziyang Xie, Junge Zhang, Wenye Li, Feihu Zhang, and Li Zhang. S-NeRF: Neural radiance fields for street views. In *ICLR*, 2023. 3
- [46] Qiangeng Xu, Zexiang Xu, Julien Philip, Sai Bi, Zhixin Shu, Kalyan Sunkavalli, and Ulrich Neumann. Point-NeRF: Point-based neural radiance fields. In *CVPR*, 2022. 3
- [47] Bin Yang, Min Bai, Ming Liang, Wenyuan Zeng, and Raquel Urtasun. Auto4D: Learning to label 4D objects from sequential point clouds. *arXiv preprint arXiv:2101.06586*, 2021. 2
- [48] Yao Yao, Zixin Luo, Shiwei Li, Tian Fang, and Long Quan. MVSNet: Depth inference for unstructured multi-view stereo. In *ECCV*, 2018. 2
- [49] Weihao Yuan, Xiaodong Gu, Zuozhuo Dai, Siyu Zhu, and Ping Tan. NeWCRFs: Neural window fully-connected CRFs for monocular depth estimation. In *CVPR*, 2022. 7, 8, 15
- [50] Fangao Zeng, Bin Dong, Yuang Zhang, Tiancai Wang, Xiangyu Zhang, and Yichen Wei. MOTR: End-to-end multiple-object tracking with transformer. In *ECCV*, 2022. 5

## Appendix

Our appendix contains the following contents:

- (A) **Demo video.** We provide a demo for offboard HD-Map generation in Sec. A.
- (B) **Voxel-NeRF details.** We explain the formulation of training and using Voxel-NeRFs in Sec. B.
- (C) **Generalizability of MV-Map with LiDAR.** In addition to the vision-oriented experimentation in the main paper, we show the generalizability of MV-Map and incorporate it with the LiDAR modality in Sec. C.
- (D) **Applications of Auto-labeling.** We validate the effectiveness of MV-Map for auto-labeling, by using it to generate pseudo HD-Map labels in Sec. D.
- (E) **Additional quantitative results.** We supplement ablation studies, especially using additional onboard models, in Sec. E.
- (F) **Additional qualitative results.** The generated HD-Maps together with the reconstructed 3D structure via our Voxel-NeRF are visualized in Sec. F.
- (G) **Implementation details.** We describe additional implementation details for reproducing our results in Sec. G.

### A. Demo Video

We provide a demo video at <https://youtu.be/SN14oTyMFrk> that showcases how our MV-Map produces high-quality HD-Maps by fusing frames from diverse viewpoints. Notably, the video highlights the effectiveness of MV-Map in *iteratively refining* complex road topologies and long road elements while dealing with frequent occlusions in urban traffic.

### B. Voxel-NeRF Details

In this section, we introduce the details of optimizing our Voxel-NeRF and augmenting our MV-Map with the encoded 3D structure (Sec. 4.3 of the main paper).

#### B.1. NeRF optimization

We supervise our Voxel-NeRF in a way that is identical to standard NeRF models [26, 28, 31, 42], by using a photometric loss between the rendered pixel color and the ground-truth color.

We first describe how NeRF infers the color of every pixel in this process. NeRF renders the color of an arbitrary pixel by accumulating the density and color information along the camera ray. Specifically, we denote the camera ray for the pixel as  $\mathbf{r}$ , which is unique for each pixel. By denoting the camera origin as  $\mathbf{o}$  and the direction of  $\mathbf{r}$  as  $\mathbf{d}$ , every 3D coordinate along the ray can be written as  $\{\mathbf{o} + t\mathbf{d} | t \in \mathcal{R}^+\}$ . The RGB color of the pixel comes from

the integral along the ray  $\mathbf{r}$ :

$$\hat{\mathbf{C}}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)\mathbf{c}(t)dt, \quad (\text{D})$$

where  $t$  ranges from the near and far planes  $t_n$  and  $t_f$ ,  $T(t) = \exp(-\int_{t_n}^t \sigma(\mathbf{o} + s\mathbf{d})ds)$  models the accumulated transmittance along the ray from  $t_n$  to  $t$ , and  $\sigma$  and  $\mathbf{c}$  denote the density and color encoded in NeRF, respectively.

The photometric loss is a reconstruction loss between the RGB colors predicted by NeRF and from the ground-truth images:

$$\mathcal{L}_{\text{color}} = \mathbb{E}_{\mathbf{r}} \|\hat{\mathbf{C}}(\mathbf{r}) - \mathbf{C}(\mathbf{r})\|_2^2, \quad (\text{E})$$

where  $\mathbf{C}(\mathbf{r})$  is the ground-truth RGB values extracted from the images.

As discussed in Sec. 4.3 of the main paper, we further add a total-variance loss  $\mathcal{L}_{\text{TV}}$  to guide the optimization of near-ground geometry. The final loss term is:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{color}} + \lambda_2 \mathcal{L}_{\text{TV}}, \quad (\text{F})$$

where  $\lambda_1$  and  $\lambda_2$  are trade-off hyper-parameters.

#### B.2. NeRF ray casting

In Sec. 4.3 of the main paper, we show how we incorporate the multi-view geometry in Voxel-NeRF with our uncertainty network. The *key operator* is to reconstruct the position of the nearest surface for each voxel by ray-casting through the corresponding image pixel. We achieve this by rendering the *termination depth* through volume rendering.

Specifically, for every camera ray represented in the form of  $\{\mathbf{o} + t\mathbf{d} | t \in \mathcal{R}^+\}$  (explained in Sec. B.1), the termination depth of the ray  $\hat{D}(\mathbf{r})$  is:

$$\hat{D}(\mathbf{r}) = \int_{t_n}^{t_f} T(t)\sigma(t)dt. \quad (\text{G})$$

Similar to Eqn. D,  $t$  ranges from the near and far planes  $t_n$  and  $t_f$ ,  $T(t) = \exp(-\int_{t_n}^t \sigma(\mathbf{o} + s\mathbf{d})ds)$  models the accumulated transmittance along the ray from  $t_n$  to  $t$ , and  $\sigma$  denotes the density in NeRF.

### C. Generalizability of MV-Map with LiDAR

To demonstrate the generalizability of our framework, we analyze incorporating the LiDAR sensor into MV-Map. In the main paper, we focused on cameras, because they contribute primarily to HD-Map generation as shown in HDMAPNet [16]. On the other hand, LiDARs are also widely used for their accurate distance sensing and their ability to enhance localization, which motivates our investigation here. We first describe the design of utilizing the extra LiDAR modality and then analyze the results. The details for training and inference are explained in Sec. G.4.

Table A: Performance of incorporating the *LiDAR modality* in MV-Map, evaluated under the long-range setting on the validation set. As a *general* framework, MV-Map can exploit multi-modality as input and improve the performance consistently and significantly.

Methods	mIoU (Long-range)			
	Divider	Ped Crossing	Boundary	All
Onboard	41.63	27.13	41.65	36.80
MV-Map	<b>50.72</b>	<b>32.99</b>	<b>54.63</b>	<b>46.11</b>

**Onboard model.** We modify the original image-based onboard model by adding a branch of LiDAR encoder with PointPillar [15] to generate the BEV feature maps from the point clouds. The BEV feature maps generated by the LiDAR encoder are later stacked with the image-based BEV features to form the final BEV features.

**Uncertainty network.** The architecture of the uncertainty network remains unchanged when integrating the LiDAR sensor, as our offboard fusion pipeline is *agnostic* to the upstream BEV perception modules.

**Voxel-NeRF.** In addition to optimizing the Voxel-NeRF with the photometric loss and total-variance loss as in Sec. 4.3 (main paper), we further leverage the point clouds to improve NeRF. Specifically, we follow DS-NeRF [6] and apply an extra depth loss term:

$$\mathcal{L}_{\text{depth}} = \mathbb{E}_{\mathbf{r}} \|\hat{D}(\mathbf{r}) - D(\mathbf{r})\|_2^2, \quad (\text{H})$$

where  $\hat{D}(\mathbf{r})$  is the rendered termination depth in Eqn. G, and  $D(\mathbf{r})$  is the depth of LiDAR points. Note that point clouds are sparser than image pixels, so we project LiDAR points onto the images and only apply the above loss term to the pixels that correspond to LiDAR points for supervision.

Our final training loss for Voxel-NeRF combines the photometric, total-variance, and depth losses when the LiDAR modality is available:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{color}} + \lambda_2 \mathcal{L}_{\text{TV}} + \lambda_3 \mathcal{L}_{\text{depth}}, \quad (\text{I})$$

where  $\lambda_1$ ,  $\lambda_2$ , and  $\lambda_3$  are trade-off hyper-parameters.

**Results.** Table A summarizes the result of MV-Map with LiDAR, which again significantly outperforms the onboard model. In addition, due to leveraging the additional modality, MV-Map with both camera and LiDAR achieves larger improvement, compared with the unimodal model result with only camera shown in Table 1. This result serves as further evidence that our framework is capable of adapting to multi-modality and achieving improved performance.

## D. Applications of Auto-labeling

The experimental results in the main paper show that MV-Map generates high-quality HD-Map labels. This in-

Table B: Comparison between onboard models trained with either ground-truth labels (GT) or pseudo-labels generated by our MV-Map (PL). The model trained with our pseudo-labels achieves comparable performance. This validates the high quality of HD-Maps generated by MV-Map and further supports its effectiveness for auto-labeling.

Label	mIoU (Validation set, Long-range)			
	Divider	Ped Crossing	Boundary	All
PL (Ours)	<b>38.99</b>	25.15	<b>38.68</b>	<b>34.27</b>
GT	38.89	<b>25.40</b>	38.16	34.15

dicates that our method is an effective *auto-labeling* strategy, which can potentially serve as a substitute for human labeling and thus support downstream applications. To further assess the quality of these labels, which we refer to as “*pseudo-labels*,” we conduct an experiment by training a new onboard model with pseudo-labels and comparing its efficacy with that trained with ground-truth labels.

To this end, we follow the *semi-supervised learning* experimental setup introduced in offboard 3D detection [36]. We use 50 out of 700 sequences on the training set of nuScenes [3] to train our uncertainty network and deploy it to infer the HD-Map labels for the remaining 650 sequences on the training set. We then train an onboard model *from scratch* on these 650 sequences with either the ground-truth labels or pseudo-labels from MV-Map.

The result in Table B shows that the model trained with pseudo-labels achieves comparable performance to that trained with ground-truth labels. This suggests that our auto-labeling approach is effective for supporting semi-supervised training. It is worth noting that our pseudo-labeling performs slightly better, likely because it helps to reduce over-fitting as evidenced by that on the *training* set using ground-truth labels results in 2.5% higher mIoU over pseudo-labels. Based on the high quality of the generated HD-Maps, our auto-labeling pipeline has the potential to be useful for other BEV perception tasks that involve traffic elements, such as BEV segmentation and lane detection. We leave such investigation as interesting future work.

## E. Additional Quantitative Results

### E.1. Impact of Training-time Frame Number

As described in Sec. 4.4, we train the uncertainty network on clips with a fixed number of frames due to GPU capacities but later apply it to *sequences with unbounded lengths*. We analyze how the training-time frame number impacts the performance of the uncertainty network. In Table C, we demonstrate that *increasing the number of frames is beneficial to the fusion performance*, as the uncertainty network has access to more diverse viewpoints for fusion during the training time. Moreover, increasing from 3 to 5

Table C: Performance of MV-Map with varied training-time frame numbers. Given that the performance saturates at 5 frames, we adopt 5 frames for training to best trade-off accuracy and computational cost. Note that during inference, we apply MV-Map to sequences with unbounded lengths.

#Frames	mIoU (Long-range)			
	Divider	Ped Crossing	Boundary	All
3	47.64	32.36	49.67	43.22
5	48.15	33.34	<b>50.28</b>	43.92
7	<b>48.23</b>	<b>34.31</b>	50.11	<b>44.22</b>

Table D: MV-Map generalizes to other onboard models. Using HDMapNet [16] with both camera and LiDAR modalities as our onboard model, MV-Map is consistently effective and significantly improves the HD-Map quality. We evaluate under the same short-range setting as HDMapNet.

MV-Map	mIoU (Short-range)			
	Divider	Ped Crossing	Boundary	All
✗	46.20	24.38	56.99	42.52
✓	<b>49.82</b>	<b>29.83</b>	<b>58.54</b>	<b>46.06</b>

frames has a significant gain in performance, while increasing from 5 to 7 frames only has marginal improvement. Our experiments in the main paper use 5-frames for training to balance the performance and computation cost.

## E.2. Generalizing to Additional Onboard Models

We demonstrate that the region-centric fusion approach in our MV-Map is *generalizable to other onboard BEV perception models*. To this end, we adopt HDMapNet [16], which is another widely-used onboard model and different from the one (SimpleBEV [12]) described in Sec. 4.1 (main paper). Following our experiment in Sec. C, we incorporate the HDMapNet encoder with the LiDAR point clouds. As shown in Table D, our MV-Map significantly improves upon the HDMapNet result, thus supporting the generalizability of MV-Map.

## F. Additional Qualitative Results

### F.1. HD-Map Visualization

We provide more qualitative results on HD-Map generation in Fig. A, in addition to our visualizations in Fig. 5 and Fig. 7 of the main paper. Compared with onboard approaches, our offboard MV-Map significantly improves the quality of HD-Maps for complex structures.

### F.2. Voxel-NeRF Visualization

We provide more visualization results of our Voxel-NeRF to indicate its capability of encoding multi-view consistency. In Fig. B, we show the reconstructed 3D structure

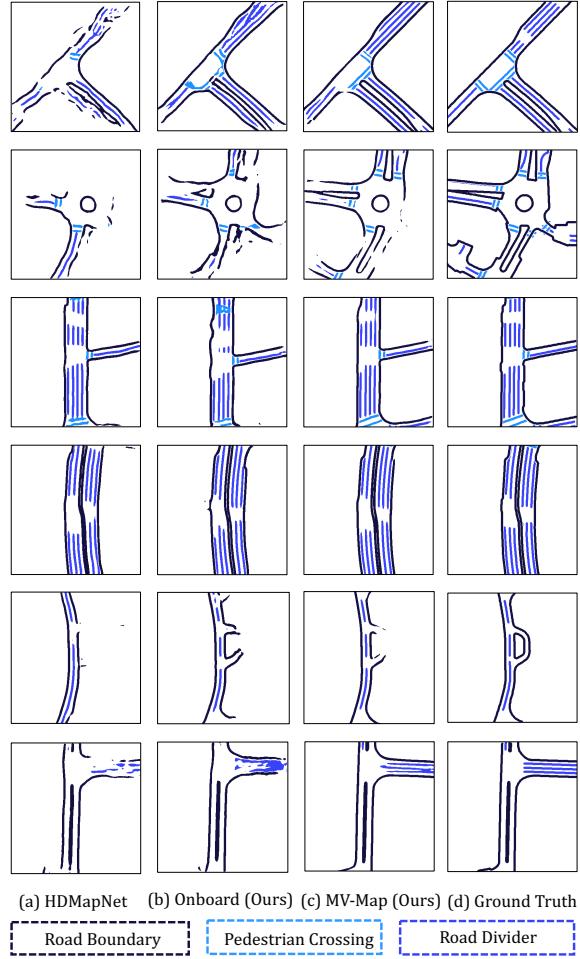


Figure A: Qualitative results for HD-Map generation. We compare the generated results from HDMapNet [16], our onboard model, and the offboard fused results from MV-Map. Compared with other approaches, our MV-Map achieves better fidelity for complex road topology.

of the scenes by our Voxel-NeRF model, through converting the diffuse color and opacity of every voxel to a colored point cloud. Qualitative results demonstrate that our Voxel-NeRF successfully optimizes a high-resolution scene representation with multi-view consistency.

## G. Implementation Details

We provide the detailed hyper-parameters and procedures to reproduce MV-Map as described in Sec. 4.4 and Sec. 5 (main paper), as well as Sec. C.

### G.1. Onboard Model

As SimpleBEV [12] was not originally proposed for HD-Map construction (despite its strong performance), we inherit their encoder design and train our own onboard models.

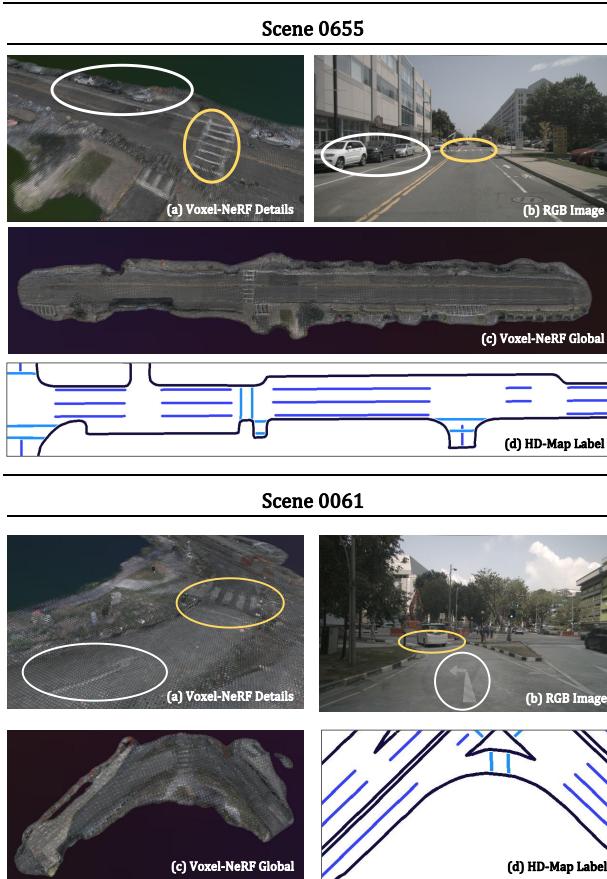


Figure B: Visualization of reconstruction results by Voxel-NeRF on the scene 0655 and 0061 from nuScenes. **(a)** NeRF’s results in the highlighted regions; **(b)** images captured by the ego vehicle; **(c)** NeRF’s results for the whole scene; **(d)** ground-truth HD-Map labels. As highlighted here, our Voxel-NeRF optimizes the 3D structure of the whole scene with high quality and multi-view consistency.

**Backbone.** Same as [12], we adopt ResNet-50 [13] as the backbone to extract the feature maps for 6 surrounding images per frame on nuScenes. The third and final stages of the ResNet output are used. We apply an additional convolution layer to generate the final feature map with a length and width of 1/8 compared to the original size of the images.

**Feature lifting.** For feature lifting, we use a sampling-based BEV encoder to lift the 2D image feature into BEV space. We first construct a local 3D voxel grid shaped  $400 \times 400 \times 6$  around the ego vehicle. The voxel grid size is 0.15m for the short-range ( $60m \times 30m$ ) setting and 0.25m for the long-range ( $100m \times 100m$ ) setting. Then, for each grid point, we acquire its positions on the image plane with intrinsic and extrinsic matrices, bi-linear sample the image features, and fill them back into each voxel grid. Finally, we reduce the voxel features into a BEV feature map with

an additional voxel encoder to make it a 2D BEV feature. During this process, the height range of the sampled voxel grid in our BEV encoder is -4m to 2m relative to the sensor origin. The final output BEV feature map shapes  $128 \times 400 \times 400$ , where 128 is the feature dimension, and 400 is the length and width of the BEV feature map.

**Decoder.** To maintain generality and comparability, we used the same decoder as HDMapNet [16]. The main structure contains three blocks from ResNet18 [13] to generate the final prediction.

**Loss function.** We use Focal loss [20], which is a dynamically scaled cross-entropy loss as our segmentation loss to solve the imbalance distribution between the most common road label and the crossing label that is relatively scarce and hard to learn:

$$\text{FL}(p_t) = -\alpha_t(1 - p_t)^\gamma \log(p_t), \quad (\text{J})$$

where we set the factor  $\alpha_t = 1$  and  $\gamma = 2$ . During training, the scaling factor can reduce the impact of dominant categories (e.g., road segments) and increase the loss assigned to challenging ones (e.g., pedestrian crossing).

**Training.** During training, we initialize the backbone ResNet-50 from the ImageNet1k [5] pretrained checkpoint. It is then trained for 16 epochs with an AdamW [24] optimizer, with an initial learning rate of 1e-3 under a 1-cycle schedule and focal loss [20] as loss function. We train our model with  $4 \times$ A100 GPUs with 2 samples per GPU. The total training process takes around 10 hours.

## G.2. Voxel-NeRF

**Architecture.** Our Voxel-NeRF models are trained with a fixed voxel size of 0.5m. The height range of our voxel is set to -4m to 2m relative to the height value of sensor origins. The near plane and the far plane of our NeRF model are 0.1m and 64m, respectively. The RGB network has a width of 128 and a depth of 3 layers. Within our model, each voxel encodes the feature with dimension 12. The first 3 channels represent the diffuse color, and the rest 9 channels concatenate the viewing directions to decode the final RGB color  $c$  with an RGB MLP.

**Training.** We reconstruct all 850 scenes in nuScenes and train 30,000 iterations with AdamW [25] optimizer and 1e-3 learning rate for each scene. Please note that our training is *without* the coarse-to-fine strategy proposed in [40]. As our scene scale is predefined and does not need the coarse stage to find a tight bounding box for further optimization. When we use the total-variance loss (Sec. 4.3), the balance loss weights  $\lambda_1$  is 1 and  $\lambda_2$  is 1e-5. The training process takes around 15 minutes for each scene on a single A40 GPU.

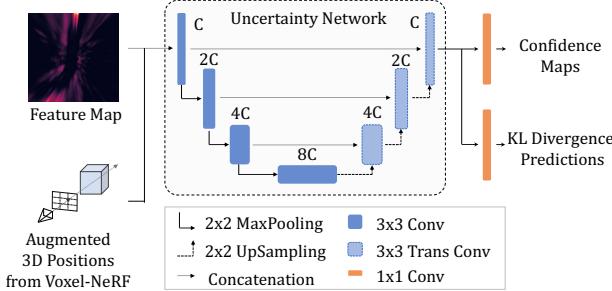


Figure C: Illustration of our **uncertainty network**, which takes as inputs the feature map and the augmented 3D positions from NeRF for each voxel (Sec. 4.3, main paper). It outputs the confidence maps for region-centric fusion and, optionally, the predicted KL-divergence for KL-divergence loss (Sec. 4.2, main paper).

### G.3. Uncertainty Network

**Architecture.** We show our uncertainty network design in Fig. C. It has a U-Net-like architecture which takes in the 128-channel BEV feature map with channels, and a 6-channel augmented input from NeRF (as Sec. 4.3, main paper), and outputs a final 128-channel feature map. Then the per-pixel confidence weight and the KL divergence prediction are output by two independent  $1 \times 1$  convolution layers with kernel size 1.

**Training.** Due to our storage constraints, we train the uncertainty network on a small subset (50 scenes) of the full training set for 5 epochs, but we manage to evaluate the full validation set of nuScenes with 150 sequences, which enables a fair comparison with other methods. As explained in Sec. 4.4 and Sec. 5 (main paper), we train the uncertainty network on short video clips with 5 frames and deploy it to all the frames in a nuScenes sequence during the inference time. A key detail to handle the varying sequence lengths for inference time is to set the *batch size* to 5 during the training time. The final loss is a weighted sum of the segmentation loss and the auxiliary KL divergence loss (Sec. 4.2, main paper), the loss weights are 1 and 0.1, respectively. The network is trained with an AdamW [24] optimizer with a learning rate of 1e-3. The training process takes around 30 minutes on a single A100 GPU.

### G.4. MV-Map with LiDAR

For the implementation of the LiDAR MV-Map, we maintain the hyperparameters of the onboard model and the uncertainty network at the same values as the unimodal model. The onboard model’s training process requires around 12 hours to complete using  $4 \times$ A100 GPUs, while the uncertainty network training process takes an additional hour using a single A100 GPU.

As discussed in Sec. C, when training with LiDAR sig-

nals, our Voxel-NeRF applies an extra depth loss term with  $\lambda_3 = 0.1$ . We keep other hyperparameters the same and train our Voxel-NeRF for 30,000 iterations for each scene, which takes 15 minutes on a single A100 GPU.

### G.5. MV-Map with Monocular Depth

We experiment using monocular depth for uncertainty network in Sec. 5.3 (main paper). Specifically, we replace the termination depth from NeRF (Eqn. G) with the depth generated by *NewCRFs* [49] to estimate the 3D positions of surface points. The other implementation details are untouched.