Addressing Diverging Training Costs using Local Restoration for Precise Bird's Eye View Map Construction

Minsu Kim[1†], Giseop Kim[2], Sunwook Choi[2*]

***Abstract*— Recent advancements in Bird's Eye View (BEV) fusion for map construction have demonstrated remarkable mapping of urban environments. However, their deep and bulky architecture incurs substantial amounts of backpropagation memory and computing latency. Consequently, the problem poses an unavoidable bottleneck in constructing high-resolution (HR) BEV maps, as their large-sized features cause significant increases in costs including GPU memory consumption and computing latency, named *diverging training costs* issue. Affected by the problem, most existing methods adopt low-resolution (LR) BEV and struggle to estimate the precise locations of urban scene components like road lanes, and sidewalks. As the imprecision leads to risky self-driving, the diverging training costs issue has to be resolved. In this paper, we address the issue with our novel Trumpet Neural Network (TNN) mechanism. The framework utilizes LR BEV space and outputs an up-sampled semantic BEV map to create a memory-efficient pipeline. To this end, we introduce *Local Restoration* of BEV representation. Specifically, the up-sampled BEV representation has severely aliased, blocky signals, and thick semantic labels. Our proposed *Local Restoration* restores the signals and *thins* (or narrows down) the width of the labels. Our extensive experiments show that the TNN mechanism provides a plug-and-play memory-efficient pipeline, thereby enabling the effective estimation of real-sized (or *precise*) semantic labels for BEV map construction. Our codes will be publicly released.**

I. INTRODUCTION

Bird's Eye View (BEV) provides an egocentric agent with a comprehensive view of large scenes, facilitating the understanding of a global scene. Owing to its capabilities, it is widely used in constructing urban maps for autonomous driving. Recent suggestions of BEV representation to evaluate latent BEV spaces often employ pillar representation with infinite-height voxels. This approach enables the sensors to share their estimated 3D geometries and promotes collaborative sensing. For example, cameras effectively complement the sparse data distribution of radar and LiDAR with their dense sensing, while LiDAR robustly and broadly collects surround geometry, and radar can estimate accurate poses of objects. Thus, their collaborative fusion in a high-dimensional BEV space enhances a robotic system's scene understanding, as evidenced in Kim *et al.*'s study [10].

However, existing BEV fusions face a significant hurdle in inferring high-resolution (HR) BEV representations (See Fig. 1 for various BEV resolutions). The heavy architecture of the frameworks, which includes multiple encoders, decoders, feature pyramid networks [14], and a feature lifting [21] module, cause substantial memory consumption and computational bottlenecks. Consequently, when a BEV is configured with a higher resolution or heavy mechanism like attention [30], both backpropagation memory and computing latency increase significantly. Because the issue, which we term the *diverging training costs*, limits the learning of HR BEVs with high costs, recent self-driving agents rely on low-resolution (LR) BEVs exposing them to risky behaviors such as crossing stop lines or road lanes. For example, in Fig. 1, despite depicting the same section of the road, the low-resolution maps indicate that the vehicle remains within its lane, while the high-resolution maps definitively show that the vehicle crosses the lane boundary. Since this misunderstanding endangers safe motion planning, we have to address this problem. To this end, it is necessary to explore efficient pipelines for representing a HR BEV, thereby it becomes generally available in self-driving fields.

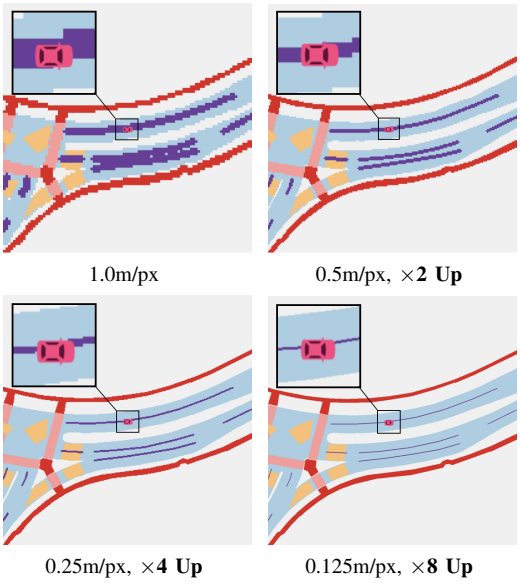


Fig. 1: **Local Restoration for Precise Map Construction.** "m/px" means meter per pixel. Precise map construction is crucial for safe autonomous driving. However, its training is significantly limited by huge memory due to deep and bulky BEV architecture. Our plug-and-play mechanism resolves the issue, allowing for efficient and precise map construction.

In this paper, we introduce a plug-and-play, memory-efficient approach, Trumpet Neural Network (TNN). Our method evaluates camera and LiDAR BEV features in a

[1]Minsu Kim is with Korea Institute of Science and Technology minsukim@kist.re.kr
[2]Giseop Kim and Sunwook Choi are with vision group of NAVER LABS {giseop.kim, sunwook.choi}@naverlabs.com
*Corresponding author.
† Work done during an internship at NAVER LABS.

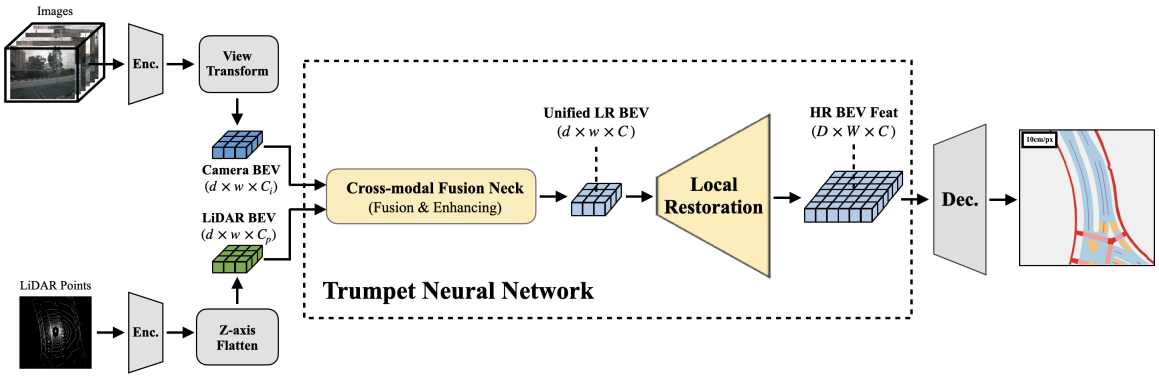


Fig. 2: **Overview of Our Proposed Methods.** Our trumpet neural network take in LiDAR ($\mathbf{z}_p^{LR}$) and Camera ($\mathbf{z}_i^{LR}$) BEV features. After the cross-modal fusion neck ($\mathbf{B}_\psi$) fuse and enhance them, our Local Restoration (S) up-samples and restores a unified HR BEV feature. Then decoding CNNs ($\mathbf{D}_\phi$) estimate an HR semantic map.

sparsely voxelized 3D space. Then, after feature enhancement, TNN up-samples and restores local details of BEV as in Fig. 3. This pipeline aims to construct HR BEV space, akin to the amplified buzzing sound of a trumpet. Our approach alleviates diverging training costs while achieving performance gains, thereby establishing a new baseline of BEV representation. With the improvements, the proposed mechanism provides an effective model training, and the achievement facilitates the learning of HR map construction. In our experiments, we focus on examining the compatibility of TNN with BEV representation using straightforward vanilla frameworks. Our contributions are summarized as:

- Our proposed mechanism addresses the issue of diverging training costs, thereby facilitating efficient training of models for representing high-resolution BEVs, which are crucial for safe self-driving.
- TNN is the first framework to utilize BEV restoration, which up-samples small-sized BEV features, thins the semantic labels, and restores their aliasing and blocky artifacts. Owing to this novel approach, TNN constructs maps in LR BEV and decodes them to precise HR BEV, effectively addressing the diverging training costs.
- We extensively investigate the compatibility of the TNN mechanism with LiDAR, camera, and their fusion.

II. RELATED WORKS

The learning methodology for the fusion of LiDAR and camera has shown promising demonstrations in map construction [24], [8], [9], [25], [36], [13], [34], [12], [20], [1], [33], [18], [31], [35], [7], [32], [33], [10], [26], [11], [37], [4]. Borse *et al.* suggested X-Align [1] and demonstrates that domain alignment enhances the robustness of BEV features against noisy and blurred effects. Kim *et al.* [10] introduced a methodology for broad BEV perception. Wang *et al.* [32] suggested an effective transformer architecture for multi-modal fusion. Kim *et al.* suggested CRN [11], a camera-radar fusion utilizing prospectively projected camera images for enriching semantic camera BEV features. Chang *et al.* proposed a BEVMap [4], which utilizes a pre-built map to address the inaccurate placement of camera BEV features in 3D space. HDMapNet [12] suggests a general framework for HD map construction with BEV segmentation. LiDAR2Map [33] proposed a distillation strategy to construct HD maps with LiDAR-only modality. Shin *et al.* suggested InstaGraM [29], a real-time HD map constructor. Qiao *et al.* proposed BeMapNet [22], a map constructor with bezier curves, and Machmap [23], an end-to-end solution for BEV map construction.

Typically, the configurations of BEV scope and resolution they employ are categorized according to their task-specific aims. For example, to construct an HD map, a model should densely perceive the urban environment. Thus, most models adopt (-30m, 30m) X-axis and (-15m, 15m) Y-axis scope in nuScenes LiDAR coordinate system [3] with 0.15m/px. In contrast, BEV segmentation utilizes (-50m, 50m) X and Y-axis scope with 0.5m/px resolution allowing for sparse, thereby utilizing relatively small-sized BEV features. Thus it enables a broad perception range.

Considering the diverging training costs of HR BEV fusion systems, there is no alternative but to use the task-specific BEV scopes as the dense and broad urban mapping requires large-sized tensor templates of BEV features. For example, if a map constructor uses a broad and dense (-50m, 50m, 0.1m/px) scope, it would encounter severe computational overheads and memory consumption due to the size of BEV features ($1000 \times 1000 \times C$). Therefore, utilizing the HR variables with limited hardware resources necessitates narrowing perception ranges to mitigate diverging training costs. To address the problem, we suggest Trumpet Neural Network, which processes coarsely voxelized BEVs and outputs an up-sampled HR BEV representation.

III. PROBLEM FORMULATION

This section outlines the formulation of TNN ($\mathbf{G}_\theta$), where θ denotes parameter. First, we revisit the conventional representation of a fused BEV feature map. Then, we describe the TNN mechanism and clarify the challenges of its implementation. Our formulations are under the assumption that a camera ($\mathbf{z}_i \in \mathbb{R}^{C_i}$), a LiDAR ($\mathbf{z}_p \in \mathbb{R}^{C_p}$), and a fused ($\mathbf{z} \in \mathbb{R}^{C_f}$) BEV feature are prepared from K images of multiple cameras ($\mathbf{I}$) and N points of LiDAR ($\mathbf{P}$) as in Liu

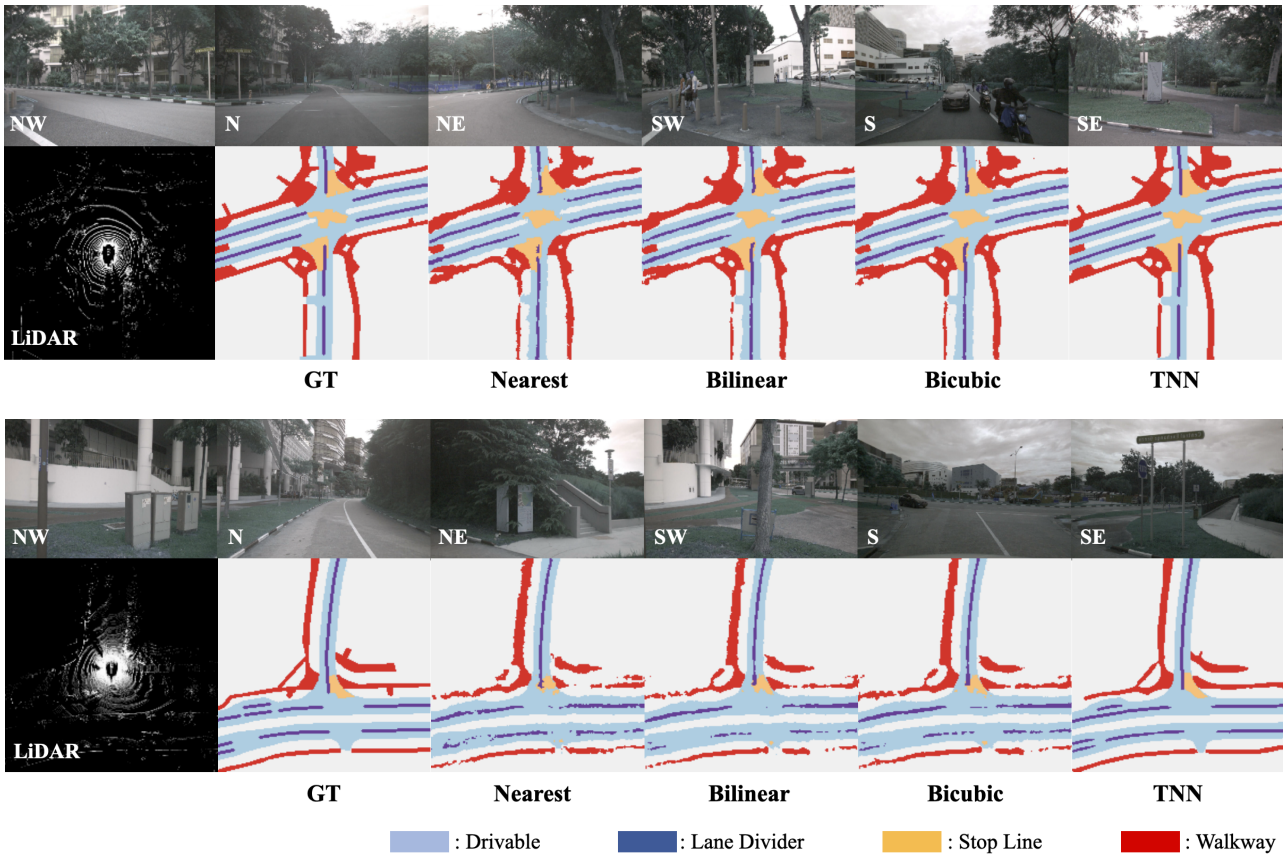


Fig. 3: **Comparison of TNN Restoration to conventional methods.** We use a (-50m, 50m) BEV scope with 2.0m/px LR BEV and 0.5m/px HR BEV resolutions. The hand-crafted **Nearest**, **Bilinear**, and **Bicubic** methods suffer from the incomplete restoration of aliasing and blocky artifacts, compared to the **TNN**, which utilizes learnable restoration.

TABLE I: **Comparisons of up-sampling methods and computational efficiency using $\times 4$ models.** The Learnable Deconvolution and Pixel Shuffle show effective restoration. Deconvolution is an over-determined form of Pixel Shuffle that consumes large computational costs without improving performance, which is why we utilize Pixel Shuffle.

Method	Latency (ms)	GPU Mem. (GB)	mIoU
Nearest	**151**	**3.6**	62.9
Bilinear	155	**3.6**	63.5
Bicubic	158	**3.6**	63.6
Deconvolution	167	3.9	**70.9**
Pixel Shuffle	164	3.8	**70.9**

TABLE II: **Plug-and-play performance gains of TNN.** We plug $\times 4$ TNN into the camera, LiDAR, and LiDAR-camera fusion modalities, employing three distinct backbones to investigate TNN's compatibility.

Method	Modality	$\times 4$ TNN	mIoU
Swin-T [17] + LSS [21]	Camera		57.1
		✓	**61.4**
Sparse Conv. [16]	LiDAR		48.6
		✓	**68.7**
BEVFusion [18]	Lidar-camera Fusion		62.7
		✓	**70.9**

et al's method [18]. We respectively notate $D \times W$ and $d \times w$ as HR and LR sizes.

A. Convention of BEV Fusion

The existing approaches [18], [1], [20], [10] extract an optimally fused BEV feature $\mathbf{z}^* \in \mathbb{R}^{D \times W \times C_f}$ as:

$$\mathbf{z}^* = \arg\min_{\hat{\mathbf{z}}} \; CE(M, \hat{M}), \tag{1}$$

$$\text{where} \;\; \hat{M} = \mathbf{D}_\phi(\hat{\mathbf{z}}), \;\; \hat{\mathbf{z}} = \mathbf{B}_\psi(f_\nu(\mathbf{z}_p, \, \mathbf{z}_i)), \tag{2}$$

'CE' means the standard cross-entropy with focal loss [15] between ground-truth ($M \in \mathbb{R}^{D \times W}$) and predicted ($\hat{M} \in \mathbb{R}^{D \times W}$) maps. The subscript of $\mathbf{z}_{(\cdot)}$ means the data modality, where i and p respectively correspond to the images and points from the camera and LiDAR, respectively. $\mathbf{D}_\phi$ is a decoding head for BEV segmentation with learnable parameter ϕ, $f_\nu(\cdot,\cdot) : (\mathbb{R}^{C_p}, \mathbb{R}^{C_i}) \mapsto \mathbb{R}^{C_f}$ denotes neural networks with parameter ν for BEV fusion such as CNN, Transformer, $\mathbf{B}_\psi(\cdot) : \mathbb{R}^{C_f} \mapsto \mathbb{R}^{C}$ denotes cross-modal fusion neck enhancing BEV features.

B. Our Strategy

A semantic BEV map for driving perception is composed of high-level global features. Thus, learning restoration of the aliased and blocky BEV can fulfill the requirements for large-scale perception in urban environments thanks to its patterned and simple-to-estimate shapes [19]. To learn local restoration, our proposed mechanism leverages Pixel Shuffle

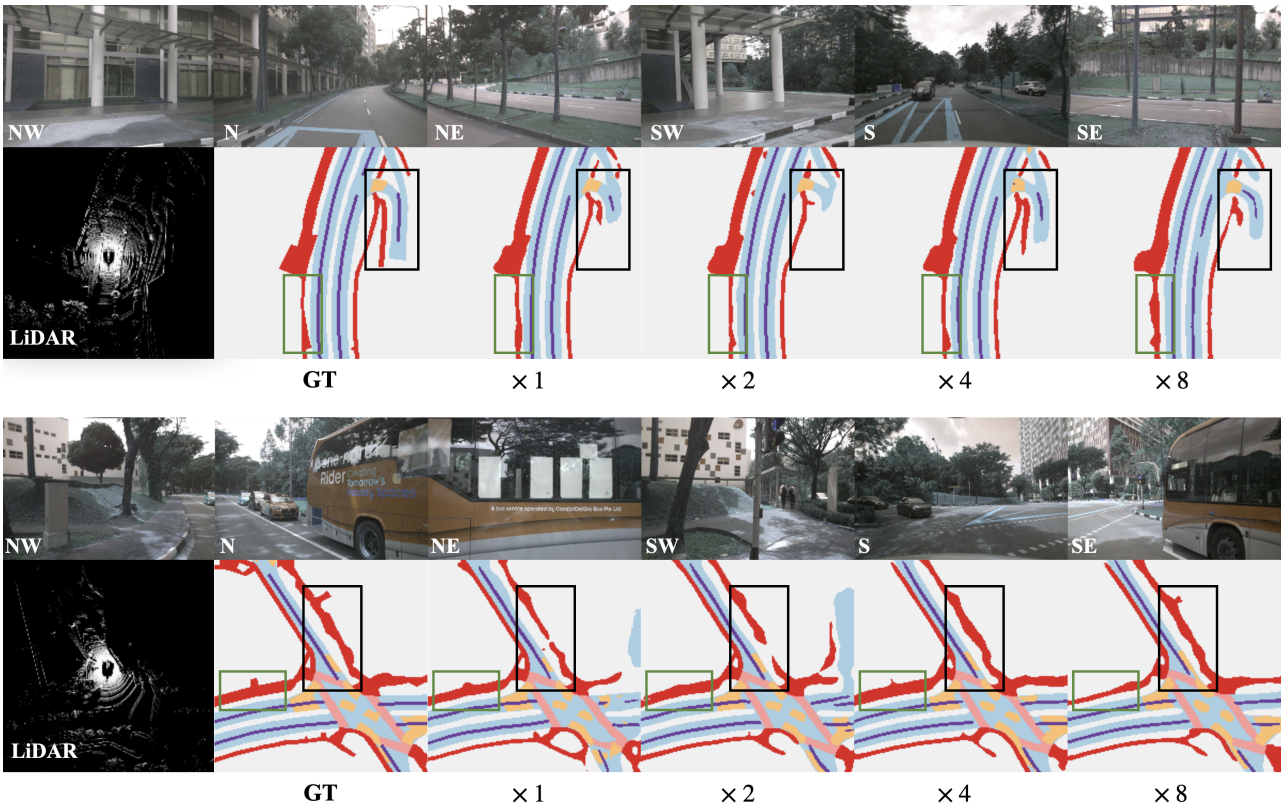


Fig. 4: **Restoration Comparisons on varying up-sampling factors.** We use a (-50m, 50m) BEV scope with 0.5m/px, 1.0m/px, 2.0m/px, 4.0m/px LR BEV resolutions for $\times 1$, $\times 2$, $\times 4$, and $\times 8$ TNN, respectively. Their HR BEV space utilizes 0.5m/px HR resolution. The $\times 1$ and $\times 2$ TNNs fail to learn high-level scene features (Black ROIs). In contrast, $\times 8$ TNN suffers from inaccurate prediction (Green ROIs) caused by severe loss of prior, compared to the $\times 4$ TNN.

[27] as:

$$\hat{\mathbf{z}} = S(\mathbf{B}_\psi(f_\nu(\mathbf{z}_p^{LR},\ \mathbf{z}_i^{LR})); \varphi), \quad S = \mathcal{PS} \circ f_\varphi, \tag{3}$$

$S(\cdot;\varphi) : \mathbb{R}^{d\times w} \mapsto \mathbb{R}^{D\times W}$ is a sequential operation of neural networks (f_φ) followed by Pixel Shuffle ($\mathcal{PS}$), $\mathbf{z}^{LR} \in \mathbb{R}^{C_f}$ is an LR BEV feature. With our HR BEV evaluation in Eq. (3), f_φ learns the restored HR BEV representation, and the composed neural nets learn compressed semantic labels as in the zoom-in regions of Fig. 7. Note that the BEV of urban environments has categorized landscapes as Lynch *et al.*'s analysis [19]. LR BEV's representation power for the approximation is enough. Our suggested mechanism answers semantic labels in LR BEV space and up-samples them to HR BEV space by restoring aliased and blocky features.

IV. METHOD

The overall pipeline is illustrated in Fig. 2. Our framework takes in K images and N LiDAR points to estimate a semantic BEV map. It adheres to the conventions established by BEVFusion [18] for extracting camera and LiDAR BEV features. Subsequently, the TNN unifies the two BEV features into a unified BEV feature map and enhances it with a fusion neck. Afterward, it deploys local restoration to construct HR BEV and decode them into a semantic map.

A. Evaluation of Low-resolution BEV

In our equations, the range of BEV grid $\mathbf{R} = (lb,\ ub,\ r)$ represents 'lb' as the lower and 'ub' as the upper bound distances from an egocentric vehicle with 'r' meters per pixel (m/px) resolution. Each trained model assumes that (lb, ub, r) is fixed for all the BEV representation axes.

Camera Branch. When given backbone nets ($\mathbf{E}_{\omega,i}$) and a down-sizing scale factor (s), the BEV features ($\mathbf{z}_i \in \mathbb{R}^{H/s\times W/s\times C_i}$) are evaluated as:

$$\mathbf{z}_i = \mathbf{T}(\mathbf{E}_{\omega,i}(\mathbf{I}), \mathbf{R}; s), \tag{4}$$

where $\mathbf{T}$ denote view transform [18]. The learnable parameters are predominantly located in backbone networks ($\mathbf{E}_{\omega,i}$), and as described in Eq. (4), the encoding of image features is not hindered by a shortage of local details. Thus, the extraction of camera BEV features prevents the degradation of representational capacity for generalizable perception.

LiDAR Branch. After N LiDAR points ($\mathbf{P} = \{p_1, p_2, ..., p_N\}$) are voxelized, their features within a common X and Y BEV range ($\mathbf{R}$), are extracted. When given backbone nets ($\mathbf{E}_{\omega,p}$), the evaluation of LiDAR BEV features ($\mathbf{z}_p \in \mathbb{R}^{H/s\times W/s\times C_p}$) is as:

$$\mathbf{z}_p = \text{Flatten} \circ \mathbf{E}_{\omega,p} \circ \text{Voxelize}(\mathbf{P}, \mathbf{R}; s), \tag{5}$$

where 'Flatten' means the Z-axis flattening of voxelized features [18]. The LR BEV offers memory-efficient computation and global features. Furthermore, LR BEV space is effectively utilized to represent the vast scope of the standardized urban scenes, as its global representation excels in approximating them.

TABLE III: **Analysis of TNN's effectiveness tendency with increasing up-scaling factors.** Compact, smaller LR BEV space ($\times 4, \times 8$) effectively approximate urban scenes. However, excessively small-sized BEV spaces often exhibit severely aliased and blocky features as indicated by the mIoU gap between $\times 4$ and $\times 8$ scales.

Scale Factor	mIoU	Latency (ms)	GPU Mem. (GB)
Baseline	62.7	176	**3.1**
$\times 1$	66.3	218	3.8
$\times 2$	69.6	187	4.2
$\times 4$	**70.9**	164	3.8
$\times 8$	70.7	**153**	3.7

TABLE IV: **Comparison of LR to HR BEV segmentations. "mIoU (r)"** denotes the evaluated **mIoU** of "r" BEV resolution. "TNN" method is pre-trained on 2.0m/px, and fine-tuned on 0.5m/px BEV segmentation.

Method	Trained BEV Resolution	mIoU (2.0m/px)	mIoU (0.5m/px)
LR BEV	2.0m/px	70.8	-
HR BEV	0.5m/px	-	62.7
$\times 4$ TNN	2.0m/px$\rightarrow$ 0.5m/px	70.8	**70.9**

B. Trumpet Neural Network

TNN simultaneously carries out up-sampling and the estimation of local details in the BEV representation ($\hat{\mathbf{z}}^{HR}$) with our proposed operator ($S(\cdot;\varphi,s)$) as:

$$\hat{\mathbf{z}} = S(\hat{\mathbf{z}}^{LR};\varphi,s). \tag{6}$$

C. Manipulation of BEV Scope and Resolution

Spatial Pooling. We use convolutional layers with more than 2-stride and spatial pooling like average or max pooling to enlarge the size of the receptive field and the coverage of BEV ranges. Given that $\mathbf{U} = [u, v]$ represents a pixel coordinate, χ contains the relative coordinates $\mathbf{p} = [p_i, p_j]$ of pixels in a pooling kernel, and $\mathbf{z}'$ is the pooled feature map, the BEV space of a fiber $\mathbf{z}'(\mathbf{U})$ with $\mathbf{R}_{(\cdot)}$ scope and $\mathbf{r}_{(\cdot)}$ resolution are defined as:

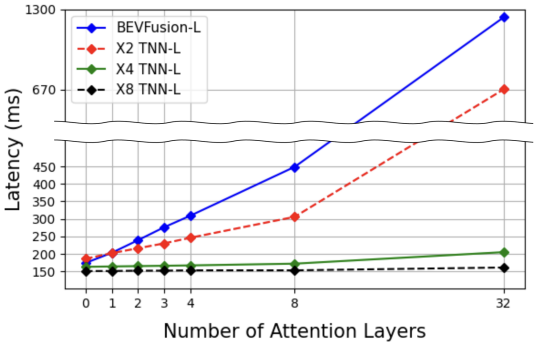
$$\mathbf{R}(\mathbf{U};\ \chi) = \mathbf{U}_i \times \mathbf{U}_j,\ \ \mathbf{r}(\mathbf{U};\ \chi) = \max\ \mathbf{p} - \min\ \mathbf{p}, \tag{7}$$

$$\text{where}\ \ \mathbf{U}_i = (u + \min\ p_i, u + \max\ p_i), \tag{8}$$

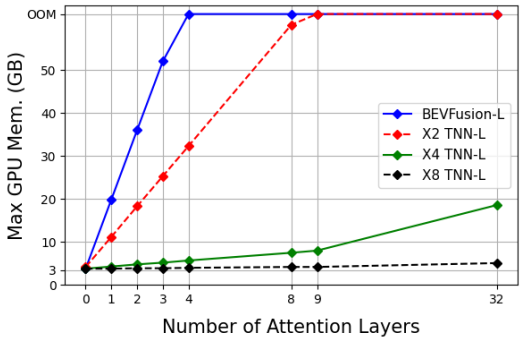
$$\mathbf{U}_j = (v + \min\ p_j,\ v + \max\ p_j). \tag{9}$$

As in Eq. (7), processing BEV features with max-pooling manipulates the pixel-wise coverage and size of the receptive field. Therefore, the BEV feature's resolution is down-sampled while becoming a global representation.

Local Restoration. The operation consists of pixel shuffle ($\mathcal{PS}$) and CNNs (f_φ). In analytical point of view, $f_\varphi(\cdot) : \mathbb{R}^{h\times w\times C} \mapsto \mathbb{R}^{h\times w\times s^2 C}$ projects latent geometry features with s-scaled scope in their channels, $\mathcal{PS}(\cdot) : \mathbb{R}^{sh\times sw\times C}$ reshapes them to an HR BEV. Thus, local restoration preserves BEV scope ($\mathbf{R}$) but manipulates resolution (r). It estimates interpolated BEV features from LR BEVs restoring local details, due to the optimization policy that imposes the model to align with the HR semantic BEV maps' ground truth.



(a) **Inference latency.**



(b) **Training GPU memory.**

Fig. 5: **Comparison on increasing costs to BEVFusion.** We incrementally add an MSA layer to the fusion neck and compare their costs. X and Y axes denote costs and the number of MSA layers, respectively.

Example. Our $\times 4$ TNN employs (-52.0m, 52.0m, 2.0m/px) and (-50m, 50m, 0.5m/px) BEV space for an LR and HR BEV space, for X and Y axes, commonly. 1) A two-level FPN layer with 2-stride CNN and up-sampling layers, 2) a range crop operation, 3) and 4-scale SR are applied on the LR BEV. Thus, the scope manipulation procedure of $\times 4$ TNN is summarized as:

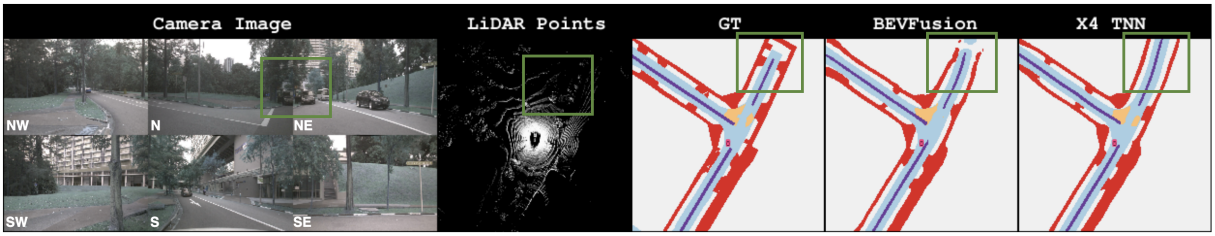
- FPN layer: 2.0m/px $\mapsto$ 0.5m/px $\mapsto$ 2.0m/px
- Range Crop: ($-$52.0m, 52.0m) $\mapsto$ ($-$50m, 50m)
- $\times 4$ up-sampling: 2.0m/px $\mapsto$ 0.5m/px
- **Output**: (-50m, 50m, 0.5m/px)

D. Training Strategy

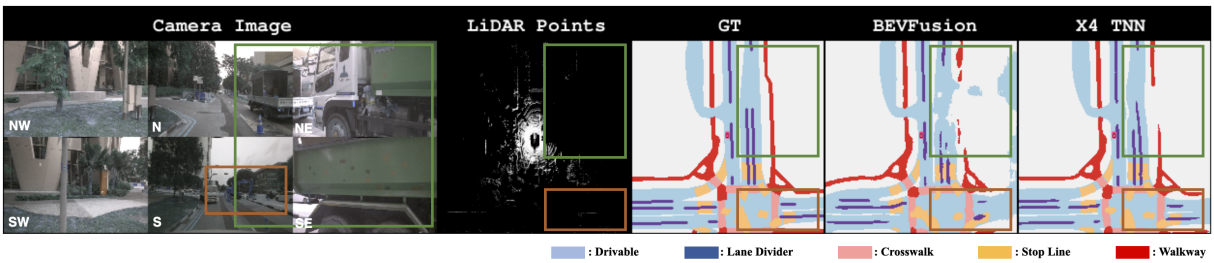
Our optimization aims to implement a modular TNN mechanism. We first train a base model, which utilizes LR BEV. Afterward, we freeze the parameters before local restoration and fine-tune local restoration networks and decoder to use HR BEV space. This strategy enables attaching varying-scaled learnable up-sampling to the latter of the model. Furthermore, the designs address the concern that high-resolution BEV consumes large memory space generating partial backpropagation memory.

E. Implementation Details

Our implementations are built on top of mmdetections [5], [6]. We use the Swin-T (Tiny) [17] model and feature lifting



(a) Green ROIs: Far regions occluded by cars.



(b) Green ROIs: Nearby Regions Occluded by White and Green Trucks, Brown ROIs: Faded Far regions.

Fig. 6: **Overconfident Issues.** Our map construction struggles with the overconfident prediction of uncertain regions.

module [21] to encode the camera BEV. We adopt sparse convolution [16] as a LiDAR backbone. To fuse the LiDAR and the camera BEV features, we use a fully convolutional layer with Feature Pyramid Networks (FPN) [14]. The data augmentations we used are the same methods as the baseline work [18]. Detailed model and training configurations are shown in our publicly opened codes.

V. EXPERIMENTAL RESULTS

In our experiments, we concentrate on exploring the cost-efficiency of the TNN mechanism, its impact on local restoration for perception, and the compatibility of TNN with both LiDAR and camera in Tables I to III. Then we analyze model efficiency with Fig. 5a and Fig. 5b.

Dataset. We use nuScenes [3]. The input images are resized to 256×704 resolution, and LiDAR points are voxelized to (0.125m, 0.125m, 0.2m), (0.125m, 0.125m, 0.2m), (0.25m, 0.25m, 0.2m), and (0.5m, 0.5m, 0.2m) XYZ resolutions for $\times 2$, $\times 4$, and $\times 8$ scale factors, respectively.

Ground truth. The semantic BEV maps of nuScenes are depicted through vectorized nodes (or geometric layers). Because the rasterization of BEV maps is derived from the nodes, any loss of ground truth is limited to negligible errors in node positions, without compromising the quality of the rasterized maps. We leverage the nuScenes map expansion [2] to generate scalable ground truth.

A. Quantitative Results

Setup. In Table I, we compare trainable up-samplings with the hand-crafted approaches: 'Nearest' neighbor, 'Bilinear', and 'Bicubic' interpolations, to validate the importance of the learnable local restoration. In the table, we examine the mean Intersection over Union (mIoU) alongside the inference time latency and the training time maximum memory of a GPU. The reported memory usage and latency are measured based on a single batch, and the memory excludes cache memory. In Table II, we validate the TNN compatibility to camera, LiDAR, and LiDAR-camera fusion modality cases. In both Table I and Table II, we use (-50m, 50m, 2.0m) BEV range and restore the extracted BEV features to (-50m, 50m, 0.5m) scope, applying a $\times 4$ TNN. In Table I, we compare scale factors of $\times 2$, $\times 4$, and $\times 8$ to explore the representation power of LR BEV. In Table IV, we assess the descriptive capabilities of both LR and HR BEVs to analyze the difference of both spaces. As an evaluation metric, we use mIoU protocols of BEVFusion [18]. Gray shading in each table represents the specifications of our base model.

Local Restoration. The comparison in Table I reveals three insights. 1) Learnable up-samplings (Pixel shuffle, deconvolution) show superior BEV representation compared to handcrafted methods (Nearest neighbor, Bilinear, and Bicubic interpolations). This suggests that non-linear aliasing and blocky artifacts in BEV can be effectively restored using neural networks. 2) Deconvolution is an over-determined form of pixel shuffle as analyzed by Shi *et al.* [28]. Both methods yield similar performance although pixel shuffle is more cost-effective. 3) The TNN mechanism's learnable up-sampling approach offers a reasonable trade-off between performance and costs. Compared to non-learnable methods, it achieves 14.2% mIoU gain with an additional 0.4GB GPU memory consumption and a 6ms latency increase in comparison to pixel shuffle over Bicubic interpolation. In Table IV, we analyze the restoration performance. For this purpose, we establish a baseline (or BEVFusion) trained under two different conditions: a 0.5m/px **HR BEV** space and a 2.0m/px **LR BEV** space. Additionally, we include a base model equipped with our proposed method for comparison. **TNN** uses 2.0m/px for LR BEV and up-samples it to 0.5m/px BEV resolution. As shown in the table, BEV map construction in LR space typically offers superior perception for understanding large urban scenes due to the standardized shapes. Our mechanism leverages this property by locally restoring the up-sampled BEV representation.

Modality Compatibility. In Table II, we evaluate the com-

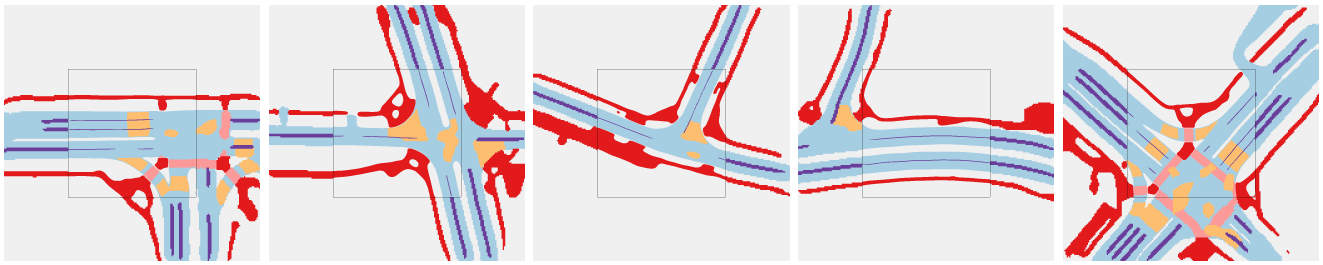


Fig. 7: **HR perception using $\times 8$ TNN.** Maps have 0.1m/px inner and 0.8m/px outer scopes.

patibility of the TNN mechanism across various modalities, including LiDAR, camera, and LiDAR-camera fusion. As indicated in the table, all modalities exhibit performance gains, which can be attributed to the superior representation power of LR BEV for semantic map construction, coupled with local restoration to evaluate HR BEV.

Representation Power. To delve into a trade-off between efficiency and representation power, we investigate the map construction performances of varying-scaled TNN in Table III with our LiDAR-camera fusion baseline, BEVFusion [18]. The exploration assesses mIoU, computation latency, and maximum GPU memory consumption. $\times 2$, $\times 4$, and $\times 8$ scale models are respectively trained under 0.5m/px, 1.0m/px, 2.0m/px, and 4.0m/px BEV resolutions for fair comparison in (50m, 50m, 0.5m/px) BEV space. The results imply there is the most reasonable scale factor to represent BEV providing powerful perception with proper memory consumption as the results of $\times 4$ scale. We see that the memory consumption of $\times 2$ requires the largest memories due to the additional memory of learnable up-sampling.

Efficiency. In Fig. 5b and Fig. 5a, we show the efficiency of our mechanism across increasingly heavier architectural configurations, illustrating how TNNs maintain efficiency even as the system architecture grows. Specifically, we add MSA layers, a well-known operator requiring large memory space, to the fusion neck of BEVFusion and TNN. In Fig. 5a, BEVFusion exhibits increasing latency, and Fig. 5b shows diverging memory. In contrast, our model maintains stable cost consumption even when incorporating 32 MSA layers into its architecture. This property will contribute to applying heavy training tricks like vision foundation models.

B. Qualitative Results

Local Restoration. In Fig. 3, we compare up-sampling methods qualitatively to demonstrate the limited map construction of the hand-crafted approaches (nearest neighbor, bilinear, bicubic interpolations). The images of the first row are camera images, and the first column of the second row shows LiDAR points of BEV. The hand-crafted methods fail to reconstruct blocky LR BEV showing uncompleted construction of deteriorated class labels like walkways, and road lanes. In addition, they fail to restore aliasing as in the edges of "walkway" labels. However, our pixel shuffle-based learnable upsampler resolves the problems showing successful restorations of the aliased and blocky BEV features.

Up-scaling Factor. In Fig. 4, we explore various up-scale factors, qualitatively. TNNs with low up-scale factors ($\times 1, \times 2$) fail to capture high-level scene features but preserve local details as shown in the black and green ROIs, respectively. In contrast, TNNs with high up-scale factors ($\times 4, \times 8$) facilitate approximating the high-level description. However, excessive TNN with up-scale factors like $\times 8$ cause severe signal aliasing and blocky artifacts, thereby imposing a reasonable selection of a specific point that has compactness and tolerant signal quality.

Overconfident Issue. We have introduced an effective strategy for capturing the structured patterns of the urban environments [19] in LR BEV space and up-sampling the features to HR BEV space. However, this method can sometimes yield overconfidence in the predicted semantic labels for occluded regions, leading to errors as shown in Fig. 6. We believe that incorporating temporal information for enriching prior could enhance the delineation of uncertain areas in future in-depth studies.

HR BEV Perception. In Fig. 7, we demonstrate our mechanism's HR perception. The task severely limits BEVFusion to training its model due to the slow training speed and huge backpropagation memory. However, the model with our mechanism is free from the diverging training costs of learning HR BEV representation as our $\times 8$ TNN model, which respectively adopts 0.8m/px and 0.1m/px for LR and HR BEV resolution, has efficient training pipelines. Our proposed mechanism enables efficient training, facilitates HR perception, and enhances map construction performance.

VI. CONCLUSIONS

We suggested the trumpet neural network (TNN), a novel architecture tailored with a local restoration focus, aimed at the precise construction of bird's eye view (BEV) maps. TNN resolved diverging training costs, thereby facilitating the training for the extraction of high-resolution BEV maps. Our experiments in semantic BEV map construction demonstrated that our proposed approach finely, and precisely restores BEV, achieving 8.2% mIoU gain over the existing baseline model, BEVFusion. Moreover, the adaptability and effectiveness of the TNN mechanism are validated across various sensory modalities, including camera, LiDAR, and LiDAR-camera fusion, respectively showing 4.3%, 10.1%, and 8.2% mIoU gains compared to the same models without the TNN mechanism. The experimental results show the solid compatibility and effectiveness of TNN, establishing a new baseline for precise BEV map construction which is essential for safe autonomous driving.

REFERENCES

[1] Shubhankar Borse, Marvin Klingner, Varun Ravi Kumar, Hong Cai, Abdulaziz Almuzairee, Senthil Yogamani, and Fatih Porikli. X-align: Cross-modal cross-view alignment for bird's-eye-view segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3287–3297, 2023.

[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. *arXiv preprint arXiv:1903.11027*, 2019.

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

[4] Mincheol Chang, Seokha Moon, Reza Mahjourian, and Jinkyu Kim. Bevmap: Map-aware bev modeling for 3d perception. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 7419–7428, 2024.

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[6] MMDetection3D Contributors. MMDetection3D: OpenMMLab next-generation platform for general 3D object detection. https://github.com/open-mmlab/mmdetection3d, 2020.

[7] Chongjian Ge, Junsong Chen, Enze Xie, Zhongdao Wang, Lanqing Hong, Huchuan Lu, Zhenguo Li, and Ping Luo. Metabev: Solving sensor failures for bev detection and map segmentation. *arXiv preprint arXiv:2304.09801*, 2023.

[8] Noureldin Hendy, Cooper Sloan, Feng Tian, Pengfei Duan, Nick Charchut, Yuesong Xie, Chuang Wang, and James Philbin. Fishing net: Future inference of semantic heatmaps in grids. *arXiv preprint arXiv:2006.09917*, 2020.

[9] 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. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15273–15282, 2021.

[10] Minsu Kim, Giseop Kim, Kyong Hwan Jin, and Sunwook Choi. Broadbev: Collaborative lidar-camera fusion for broad-sighted bird's eye view map construction. *arXiv preprint arXiv:2309.11119*, 2023.

[11] Youngseok Kim, Juyeb Shin, Sanmin Kim, In-Jae Lee, Jun Won Choi, and Dongsuk Kum. Crn: Camera radar net for accurate, robust, efficient 3d perception. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17615–17626, 2023.

[12] Qi Li, Yue Wang, Yilun Wang, and Hang Zhao. Hdmapnet: An online hd map construction and evaluation framework. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 4628–4634. IEEE, 2022.

[13] 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 *European conference on computer vision*, pages 1–18. Springer, 2022.

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[16] Baoyuan Liu, Min Wang, Hassan Foroosh, Marshall Tappen, and Marianna Pensky. Sparse convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 806–814, 2015.

[17] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.

[18] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela L Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2774–2781. IEEE, 2023.

[19] Kevin Lynch. *The image of the city*. MIT press, 1964.

[20] Yunze Man, Liang-Yan Gui, and Yu-Xiong Wang. Bev-guided multi-modality fusion for driving perception. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21960–21969, 2023.

[21] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020.

[22] Limeng Qiao, Wenjie Ding, Xi Qiu, and Chi Zhang. End-to-end vectorized hd-map construction with piecewise bezier curve. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13218–13228, 2023.

[23] Limeng Qiao, Yongchao Zheng, Peng Zhang, Wenjie Ding, Xi Qiu, Xing Wei, and Chi Zhang. Machmap: End-to-end vectorized solution for compact hd-map construction. *arXiv preprint arXiv:2306.10301*, 2023.

[24] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018.

[25] Avishkar Saha, Oscar Mendez, Chris Russell, and Richard Bowden. Translating images into maps. In *2022 International conference on robotics and automation (ICRA)*, pages 9200–9206. IEEE, 2022.

[26] Sushil Sharma, Arindam Das, Ganesh Sistu, Mark Halton, and Ciarán Eising. Bevseg2tp: Surround view camera bird's-eye-view based joint vehicle segmentation and ego vehicle trajectory prediction. *arXiv preprint arXiv:2312.13081*, 2023.

[27] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[28] Wenzhe Shi, Jose Caballero, Lucas Theis, Ferenc Huszar, Andrew Aitken, Christian Ledig, and Zehan Wang. Is the deconvolution layer the same as a convolutional layer? *arXiv preprint arXiv:1609.07009*, 2016.

[29] Juyeb Shin, Francois Rameau, Hyeonjun Jeong, and Dongsuk Kum. Instagram: Instance-level graph modeling for vectorized hd map learning. *arXiv preprint arXiv:2301.04470*, 2023.

[30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[31] Sourabh Vora, Alex H Lang, Bassam Helou, and Oscar Beijbom. Pointpainting: Sequential fusion for 3d object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4604–4612, 2020.

[32] Haiyang Wang, Hao Tang, Shaoshuai Shi, Aoxue Li, Zhenguo Li, Bernt Schiele, and Liwei Wang. Unitr: A unified and efficient multi-modal transformer for bird's-eye-view representation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6792–6802, 2023.

[33] Song Wang, Wentong Li, Wenyu Liu, Xiaolu Liu, and Jianke Zhu. Lidar2map: In defense of lidar-based semantic map construction using online camera distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5186–5195, 2023.

[34] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17830–17839, 2023.

[35] Tianwei Yin, Xingyi Zhou, and Philipp Krähenbühl. Multimodal virtual point 3d detection. *Advances in Neural Information Processing Systems*, 34:16494–16507, 2021.

[36] Brady Zhou and Philipp Krähenbühl. Cross-view transformers for real-time map-view semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13760–13769, 2022.

[37] Xiyue Zhu, Vlas Zyrianov, Zhijian Liu, and Shenlong Wang. Mapprior: Bird's-eye view map layout estimation with generative models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8228–8239, 2023.