

Parametric Depth Based Feature Representation Learning for Object Detection and Segmentation in Bird's-Eye View

Jiayu Yang^{1,3*}, Enze Xie², Miaomiao Liu¹, Jose M. Alvarez³

¹Australian National University, ²The University of Hong Kong, ³NVIDIA

{jiayu.yang, miaomiao.liu}@anu.edu.au, xieenze@connect.hku.hk, josea@nvidia.com

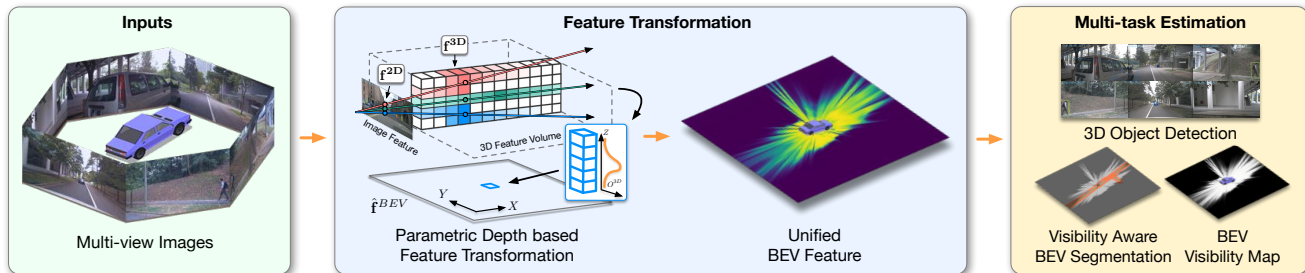


Figure 1: Given multi-view images and camera parameters, our framework utilize parametric depth to transform image feature into BEV space for jointly estimating 3D object detection, BEV segmentation and a BEV visibility map. Code available at: <https://github.com/NVlabs/ParametricBEV>

Abstract

Recent vision-only perception models for autonomous driving achieved promising results by encoding multi-view image features into Bird's-Eye-View (BEV) space. A critical step and the main bottleneck of these methods is transforming image features into the BEV coordinate frame. This paper focuses on leveraging geometry information, such as depth, to model such feature transformation. Existing works rely on non-parametric depth distribution modeling leading to significant memory consumption, or ignore the geometry information to address this problem. In contrast, we propose to use parametric depth distribution modeling for feature transformation. We first lift the 2D image features to the 3D space defined for the ego vehicle via a predicted parametric depth distribution for each pixel in each view. Then, we aggregate the 3D feature volume based on the 3D space occupancy derived from depth to the BEV frame. Finally, we use the transformed features for downstream tasks such as object detection and semantic segmentation. Existing semantic segmentation methods do also suffer from a hallucination problem as they do not take visibility information into account. This hallucination can be particularly problematic for subsequent modules such as control and planning. To mitigate the issue, our method provides depth uncertainty and reliable visibility-aware estimations. We further leverage our parametric depth modeling to present a novel visibility-aware evaluation metric that, when taken

into account, can mitigate the hallucination problem. Extensive experiments on object detection and semantic segmentation on the nuScenes datasets demonstrate that our method outperforms existing methods on both tasks.

1. Introduction

In autonomous driving, multiple input sensors are often available, each of which has its coordinate frame, such as the coordinate image frame used by RGB cameras or the egocentric coordinate frame used by the Lidar scanner. Downstream tasks, such as motion planning, usually require inputs in a unified egocentric coordinate system, like the widely used Bird's Eye View (BEV) space. Thus, transforming features from multiple sensors into the BEV space has become a critical step for autonomous driving. Here, we focus on this transformation for the vision-only setup where we take as input multi-view RGB images captured in a single time stamp by cameras mounted on the ego vehicle and output estimation results, such as object detection and segmentation, in a unified BEV space, see Fig. 1. In general, accurate depth information is crucial to achieve effective transformations.

Early methods[16, 22] forgo explicit depth estimation and learn implicit feature transformations using neural networks, which suffers from the generalization problem since the neural network does not have an explicit prior of the underlying geometric relations. More recent methods [18, 33]

*The work is done during an internship at NVIDIA

adopt explicit but simplified depth representations for the transformation, which either requires large memory consumption, limiting the resolution [18]; or over-simplifies the representation leading to noise in the BEV space[33]. Moreover, these simplified depth representation do not have the ability to efficiently provide visibility information. As downstream tasks such as semantic segmentation are trained using aerial map ground truth, the lack of visibility estimation usually results in hallucination effects where the network segments areas that are not visible to the sensor [18, 33], see Figure 2. As a consequence, those estimations can mislead downstream planning tasks as it is extremely dangerous to drive towards hallucinated road but actually non-driveable, especially in high speed.

To address these limitations, we propose to adopt explicit parametric depth representation and geometric derivations as guidance to build a novel feature transformation pipeline. We estimate a parametric depth distribution and use it to derive both a depth likelihood map and an occupancy distribution to guide the transformation from image features into the BEV space. Our approach consists of two sequential modules: a geometry-aware feature lifting module and an occupancy-aware feature aggregation module. Moreover, our parametric depth-based representation enables us to efficiently derive a visibility map in BEV space, which provides valuable information to decouple visible and occluded areas in the estimations and thus, mitigate the hallucination problem. We also derive ground-truth visibility in BEV space, which enables us to design a novel evaluation metric for BEV segmentation that takes visibility into account and reveals insight of selected recent methods [18, 33] in terms of estimation on visible region and hallucination on occluded region.

Our contributions can be summarized as follows:

- We propose a geometry-aware feature transformation based on parametric depth distribution modeling to map multi-view image features into the BEV space. Our depth distribution modeling enables the estimation of visibility maps to decouple visible and occluded areas for downstream tasks.
- The proposed feature transformation framework consists of a novel feature lifting module that leverages the computed depth likelihood to lift 2D image features to the 3D space; and a feature aggregation module to project feature to the BEV frame through the derived 3D occupancy.
- We further propose a novel visibility-aware evaluation metric for segmentation in BEV space that reveals the insight of estimation on visible space and hallucination on occluded space.

Extensive experiments on the nuScenes dataset on object

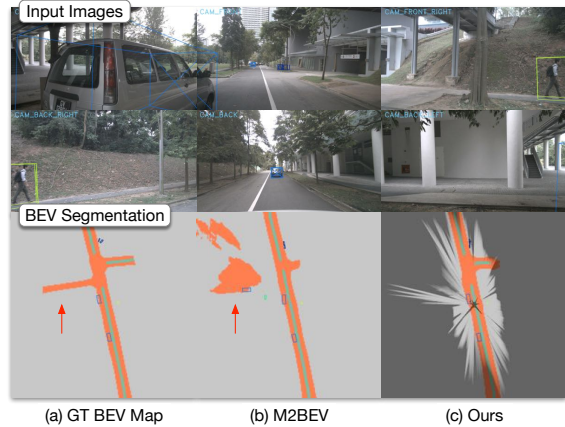


Figure 2: Hallucination in semantic segmentation. Current methods use ground truth obtained from maps (a) and therefore, predicted outputs (b) might represent parts that are not visible to the camera. As information is actually not available, it is not possible to determine if the road areas in the occluded areas is actually free for driving. Our approach enables creating a Visibility map (c) to decouple areas that are totally occluded to the camera from those that are actually visible.

detection and semantic segmentation demonstrate the effectiveness of our method yielding state of the art results for these two tasks with a negligible compute overhead.

2. Related Work

External depth based feature transformations. When given depth input either from Lidar sensor or stereo matching, image feature can easily be transformed into BEV space[9, 25, 24, 34, 35]. PointPillar[9] extract features from a 3D point cloud and aggregate the features into BEV space. PseudoLidar[31, 20] based methods firstly estimate a depth using stereo matching given stereo image pair as input followed by unprojecting the feature based on estimated depth. However, in real-life applications, Lidar sensors or stereo image inputs are not always available, which limits these line of methods.

Feature transformations without reliable depth input. Without reliable depth input, various feature transformation methods have been proposed[4, 8, 19, 21, 26], starting from early methods[16, 22] that learn implicit feature transformations using neural networks. Learned transformation can suffer from the generalization problem, since the neural network does not explicitly account for changes in cameras’ intrinsic and extrinsic parameters. Recent methods [18, 33, 32] adopt various depth representations to explicitly transform features based on multi-view geometry to the BEV space. The key in these methods is the underlying depth representation, which dominates the resolution and accuracy the feature transformation module can achieve. For instance, LSS [18] adopts a non-parametric depth rep-

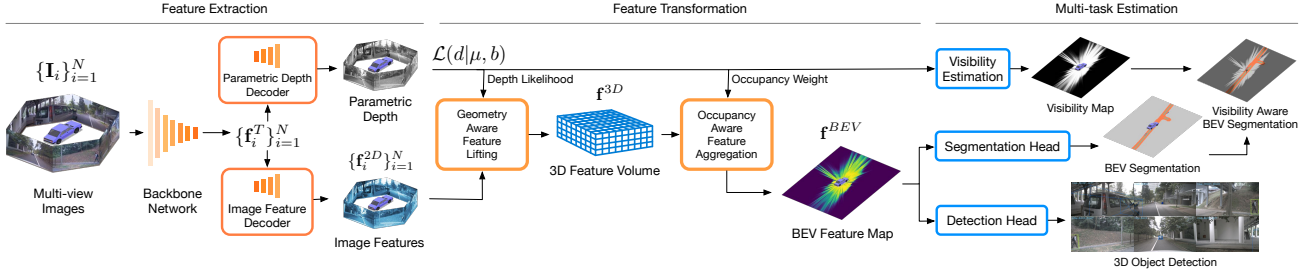


Figure 3: **Proposed framework.** The main novelties are a parametric depth decoder in the feature extraction, a geometry-aware feature lifting module and an occupancy aware feature aggregation in the feature transformation. We also introduce a visibility estimation module as part of the multi-task estimation.

resentation. It represents depth as a discretized probability density function along each visual ray, which can be treated as a categorical distribution of depth. It can further form the depth probability volume in LSS for all pixels in an image. When the sampling rate is sufficient, such non-parametric depth distribution can adequately represent a large variety of depths, including multi-modal depth distributions. In practice, however, to estimate such depth representation, the backbone needs to estimate a probability volume that is cubic with the input image size and increases significantly along the number of input images, which limits the image and depth resolution.

To address this limitation, M²BEV [33] adopts a simplified depth representation assuming the depth of all pixels follows a uniform distribution. Under this assumption, features are directly lifted to every location on the visual ray, resulting identical feature along the entire ray with no difference. Following works [12, 1] followed similar depth representation. Such simplified representation have advantage on efficiency, as the backbone network do not need to estimate any parameter for the depth, but can cause ambiguity and noise in the 3D space.

Unlike the non-parametric depth distribution used in [18] or the uniform depth distribution in M²BEV[33], we adopt a parametric depth distribution to model pixel-wise depth for feature lifting. Parametric depth distribution represents depth as a continuous distribution such as Gaussian or the Laplacian distribution, and its estimated distribution parameters can be used to evaluate depth likelihood or depth probability on any given depth value along each ray. To model the depth for a pixel, it takes only two parameters (μ, σ) for Gaussian and two (μ, b) for Laplacian, so it can be more efficient than non-parametric distribution. Moreover, its continuous nature allows evaluating depth likelihood on any points along the visual ray, which can achieve a higher depth resolution than the discretized non-parametric distribution. We specifically designed our pipeline incorporating parametric depth to improve 2D-BEV feature transformation and also propose the derivation of visibility for subsequent planning tasks and visibility-aware evaluations.

Aggregating 3D feature into BEV space. Given the lifted

feature in 3D space, most existing works including LSS [18] and M²BEV [33] use the feature concatenation method introduced by Pointpillars[9] for transforming 3D features into BEV space. The 3D feature volume is split along horizontal dimensions and interpreted as pillars of features. Then, a feature vector is created by concatenating features along the vertical dimension for each pillar. All the concatenated features form a 2D feature map, which is converted into BEV feature map by few convolution layers. This design allows each voxel along the Z-axis to have equal contribution to the final BEV feature. However, this method can be affected by noisy features on empty spaces. We thus propose to compress the features based on a calculated space occupancy probability from the parametric depth distribution. Our proposed method can largely reduce the influence of those empty voxels to the aggregated features.

Joint Detection and Segmentation in BEV space. M²BEV recently proposed a unified detection and segmentation framework in BEV space, which we leverage to evaluate the effectiveness of our method. Specifically, the image features are transformed into a unified BEV feature, which is used by two parallel heads, a detection head and a segmentation head, to achieve multi-task estimation. M²BEV leverage a detection head design from Lidar-based detection methods [9] and modify it to better suit camera-based methods. Their segmentation head is inspired by the design from [18]. However, in contrast to prior work, we leverage the proposed explicit feature transformations based on parametric depth to address its weaknesses.

Temporal extension. Few concurrent methods [12, 13, 36, 7, 28, 17, 6] proposed to utilize temporal information to further boost segmentation and detection performance in BEV space and achieved promising results. Most of these methods, including BEVFormer[12], BEVerse[36], BEVDet4D[7] are based on the feature transformation module in LSS[18]. [11, 10] adopt depth supervision and temporal stereo matching to improve depth quality and further propose a more efficient implementation of LSS’s Lift-splat step. [13, 12, 1] query 2D features from projected location of 3D voxels, which does not explicitly use depth and is similar to the uniform depth assumption in M²BEV[33].

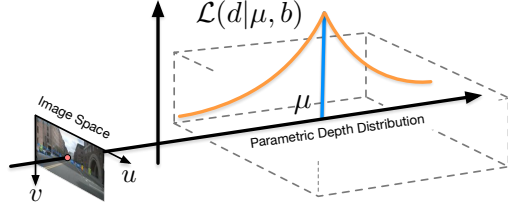


Figure 4: **Parametric depth distribution modeling.** We model the depth as a Laplacian distribution.

Our contributions focusing on depth representation, feature transformation and visibility estimation is orthogonal to the temporal extension of these methods and our method can potentially be applied to these methods to further boost their performance and enable the efficient visibility inference.

3. Method

Let us now introduce our framework to jointly perform segmentation and object detection. Shown in Fig. 3, our framework comprised of three fundamental components: feature extraction, feature transformation, and multi-task estimation. The framework’s key contributions include a parametric depth decoder integrated into the feature extraction, a geometry-aware feature lifting module, and an occupancy-aware feature aggregation module. Furthermore, we introduce a visibility estimation module as a constituent of the multi-task estimation that provide crucial visibility information for down-streaming planning tasks.

3.1. Problem Statement

Let $\{\mathbf{I}_i\}_{i=1}^N$, $\mathbf{I}_i \in \mathbb{R}^{H \times W \times 3}$, be a set of RGB images taken at the same time slot, H and W define the image dimension, and $\{\mathbf{K}_i, \mathbf{R}_i, \mathbf{T}_i\}_{i=1}^N$ represent the intrinsic and extrinsic parameters for their corresponding camera poses, respectively. We focus on lifting the image features $\mathbf{f}_i^{2D} \in \mathbb{R}^{H \times W \times CH}$ to the 3D space as $\mathbf{f}_i^{3D} \in \mathbb{R}^{X' \times Y' \times Z' \times CH}$ and then aggregate them to the BEV space as $\mathbf{f}^{BEV} \in \mathbb{R}^{X \times Y \times CH_B}$ for 3D object detection and segmentation.

3.2. Parametric Depth Distribution Modelling

Let us first introduce our parametric depth distribution modelling. Given an image \mathbf{I}_i , we extract its latent features \mathbf{f}_i^T using a backbone network followed by a image feature decoder network to extract 2D image features, \mathbf{f}_i^{2D} , see fig. 3. Then, following depth estimation methods [27, 3], we adopt a Laplacian distribution to model depth in real-world scenarios where the depth distribution for each pixel is given by,

$$\mathcal{L}(d|\mu, b) = \frac{1}{2b} \exp\left(-\frac{|d - \mu|}{b}\right), \quad (1)$$

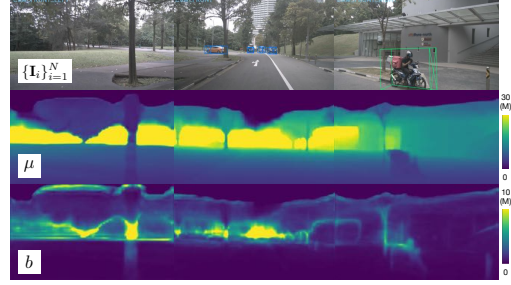


Figure 5: **Example of estimated parametric depth-distribution.** From top to bottom: input image, the estimated depth (μ) and the diversity parameter (b) interpreted as the uncertainty of the estimation.

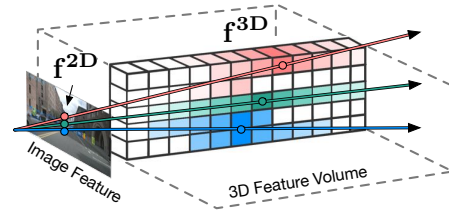


Figure 6: Geometry-aware feature lifting

where μ provides an estimation of the depth, and b is the diversity parameter of the distribution, see Fig. 4. The goal in this module is to estimate (μ, b) .

We design the parametric depth decoder network Φ_θ to map the latent feature to the parameter space of the depth distribution: $\Phi_\theta : \mathbb{R}^{H \times W \times CH_T} \rightarrow \mathbb{R}^{H \times W \times 2}$, where CH_T is the latent feature dimension. Note that when the ground-truth depth for each pixel is known, the depth distribution becomes a delta function, where the depth probability $p(d_{gt})$ on ground-truth depth d_{gt} is one and zero anywhere else. However, in practice, the depth is unknown for each pixel. Given our modelled depth distribution, we can calculate the depth likelihood analytically based on our parametric modelling. Fig. 5 shows an example of depth distribution where μ gives an estimate of the depth and b could be interpreted as the uncertainty of each estimation. Larger values of b correspond to areas where the estimation is more uncertain.

3.3. Geometry-aware Feature Lifting

Fig. 6 depicts our geometry-aware feature lifting module to transform the 2D image features $\mathbf{f}_i^{2D} \in \mathbb{R}^{H \times W \times CH}$ from the camera coordinate system into 3D space defined for the ego vehicle coordinate system, generating the 3D feature volume $\mathbf{f}_i^{3D} \in \mathbb{R}^{X' \times Y' \times Z' \times CH_i}$.

Ideally, the 2D image feature for each pixel is back-projected along the visual ray to the 3D location defined by its ground truth depth value $\mathbf{f}^{3D}(\mathbf{P}_{gt}) = \mathbf{f}^{2D}(\mathbf{p})$, where $\mathbf{P}_{gt} = d_{gt} \mathbf{K}_i^{-1} \tilde{\mathbf{p}}$, $\tilde{\mathbf{p}}$ is the homogeneous coordinate for \mathbf{p} . Without knowing the true depth value for each pixel, we

discretize the 3D space into voxels and thus aggregate the feature for each voxel by forward projecting it to multi-view images.

Precisely, let $\mathbf{P}_j = (x_j, y_j, z_j)^T$ define the 3D coordinate of centre for voxel j . Given the camera poses for multiple views, we project it to image \mathbf{I}_i as $d_j^i \tilde{\mathbf{p}}_j^i = \mathbf{K}_i(\mathbf{R}_i \tilde{\mathbf{P}}_j + \mathbf{T}_i)$ where $\tilde{\mathbf{p}}_j^i$ denotes the homogenous coordinate of \mathbf{p}_j^i in image \mathbf{I}_i . Meanwhile, we can obtain the depth value of \mathbf{P}_j in view i as d_j^i . Based on our parametric depth modelling, we obtain the likelihood of d_j^i being on the object surface as

$$\alpha_{d_j^i} = \mathcal{L}(d_j^i | \mu_{\mathbf{p}_j^i}^i, b_{\mathbf{p}_j^i}^i) = \frac{1}{2b_{\mathbf{p}_j^i}^i} \exp\left(-\frac{|d_j^i - \mu_{\mathbf{p}_j^i}^i|}{b_{\mathbf{p}_j^i}^i}\right). \quad (2)$$

We similarly project the voxel to all views and aggregate the feature for the j -th voxel as

$$\mathbf{f}_j^{3D} = \sum_{i=1}^N \alpha_{d_j^i} \mathbf{f}_i^{2D}(\mathbf{p}_j^i), \quad (3)$$

where \mathbf{f}_i^{2D} is the extracted image feature. We adopts bilinear interpolation to obtain $\mathbf{f}_i^{2D}(\mathbf{p}_j^i)$ when \mathbf{p}_j^i is a non-grid coordinate. All lifted 3D features form the 3D feature volume $\mathbf{f}^{3D} \in \mathbb{R}^{X' \times Y' \times Z' \times CH}$, which is then aggregated by our occupancy aware feature aggregation module into 2D BEV feature, introduced in the following section.

3.4. Occupancy-aware Feature Aggregation

Our occupancy-aware feature aggregation module aggregates the 3D feature volume $\mathbf{f}^{3D} \in \mathbb{R}^{X' \times Y' \times Z' \times CH}$ from ego vehicle 3D coordinate frame into BEV space, forming BEV feature map $\mathbf{f}^{BEV} \in \mathbb{R}^{X \times Y \times CH_B}$.

As shown in Fig. 7, the 2D BEV coordinate system is aligned with the XY plane of the ego vehicle coordinate system where the shared origin is defined on the center of the ego vehicle. Note that the BEV coordinate system only has 2 dimensions, forgoing the Z dimension. The goal of the feature aggregation is to transform the 3D feature volume in ego vehicle coordinate into a 2D feature map in the BEV space, which can be treated as aggregating the 3D feature volume along its Z axis. To this end, we first rearrange the previously computed depth likelihood for all voxels by Eq. 2 into a depth likelihood volume $P^{3D} \in \mathbb{R}^{X' \times Y' \times Z'}$, which shares the same volumetric coordinate as that of 3D feature volume \mathbf{f}^{3D} . For each column along the Z -axis in the depth likelihood volume, the likelihood of each voxel of different height reflects its spatial occupancy. Thus, we normalize the depth likelihood along Z axis into a spatial occupancy distribution, forming a spatial occupancy volume $O^{3D} \in \mathbb{R}^{X' \times Y' \times Z'}$ defined as

$$O^{3D}(x, y, z) = \frac{P^{3D}(x, y, z) + b_o}{\sum_{z_i=0}^{Z'-1} P^{3D}(x, y, z_i) + b_o}, \quad (4)$$

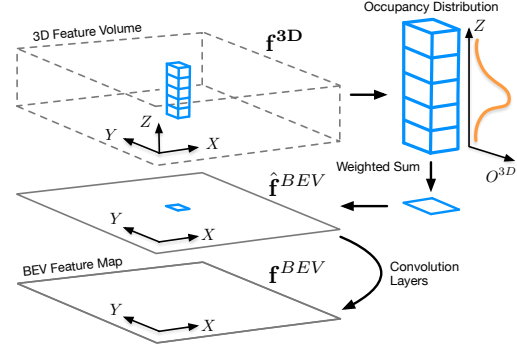


Figure 7: Occupancy aware feature aggregation

where the b_o is a bias term to encourage an equal contribution of feature on completely occluded region.

Our feature aggregation along the Z -axis could minimize the influence of features from empty voxels to the final feature in the BEV frame. Given the spatial occupancy volume O^{3D} , we compute the final 2D BEV feature as a weighted sum of 3D features

$$\hat{\mathbf{f}}^{BEV}(x, y) = \sum_{z_i=0}^{Z'-1} (O^{3D}(x, y, z_i) \times \mathbf{f}^{3D}(x, y, z_i)), \quad (5)$$

where we use the normalized spatial occupancy distribution as the 3D feature weight.

We further transform $\hat{\mathbf{f}}^{BEV}$ via a few layers of convolution to obtain the final feature for BEV space \mathbf{f}^{BEV} which is then applied to detection and segmentation tasks.

3.5. Object Detection and Segmentation

Given the BEV feature map, we use two heads for detection and segmentation. Specifically, we adopt the detection head and segmentation head from M²BEV [33] without modification for fair comparison. The detection head consists of three convolution layers and outputs dense 3D anchors in BEV space along with category, box size, and direction of each object. The segmentation head consists of five convolution layers and outputs 2 classes predictions, *road* and *lane*, as originally defined by LSS[18].

3.6. Training Strategy

We adopt supervised training strategy. We supervise the parametric depth estimation by maximizing its depth likelihood on ground-truth depth observations. Specifically, we minimize the negative log-likelihood loss \mathcal{L}_D using sparse ground-truth depth d_{gt} generated from sparse lidar measurements. Here \mathcal{L} represent Laplacian distribution and P_{gt}^i represent set of pixels where ground-truth lidar measurements is valid for image i .

$$\mathcal{L}_D(\theta) = \sum_{i=1}^N \sum_{p \in P^i} -\log(\mathcal{L}(d_{gt,i}^p | \mu_i^p(\theta), b_i^p(\theta))) \quad (6)$$

where \mathcal{P}^i defines the set of pixel coordinates with valid ground truth depth map for view i .

For detection head, we use the 3D detection loss used in PointPillars[9] as follows, where \mathcal{L}_{loc} is the total localization loss, \mathcal{L}_{cls} is the object classification loss, \mathcal{L}_{dir} is the direction classification loss, N_{pos} refer to the number of positive samples and $\beta_{cls}, \beta_{loc}, \beta_{dir}$ are set to 1.0, 0.8, 0.8 accordingly.

$$\mathcal{L}_{det} = \frac{1}{N_{pos}}(\beta_{cls}\mathcal{L}_{cls} + \beta_{loc}\mathcal{L}_{loc} + \beta_{dir}\mathcal{L}_{dir}) \quad (7)$$

Please refer to [9] for more details.

For segmentation head, we use both Dice loss \mathcal{L}_{dice} and binary cross entropy loss \mathcal{L}_{bce} as segmentation loss \mathcal{L}_{seg} and use equal weight $\beta_{dice} = \beta_{bce} = 1$.

$$\mathcal{L}_{seg} = \beta_{dice}\mathcal{L}_{dice} + \beta_{bce}\mathcal{L}_{bce} \quad (8)$$

For the visibility map and additional outputs, since they are geometrically derived from the estimated parametric depth representation without any learned parameters, it's not necessary to apply supervision on them.

4. Visibility

4.1. Visibility Map

The segmentation in BEV space mainly focuses on segmenting lane regions. However, those regions are not always visible in the camera views due to the occlusion of vertical scene structures such as building (see Fig.2). We thus propose to use our parametric depth modeling to infer a visibility map which decouples visible and occluded areas and, will contribute to mitigate the hallucination effect.

We define a visibility map $V^{BEV} \in \mathbb{R}^{X \times Y}$ to describe the visibility range of ego vehicle's multi-view cameras. Starting from the likelihood of the Laplacian distribution in Eq. 2, the occlusion probability $B(d)$ of a voxel in 3D space that has a back-projected depth d in camera view is

$$B(d) = \int_0^d \mathcal{L}(x|\mu, b)dx. \quad (9)$$

We derive this occlusion probability as follows. Firstly we find the indefinite integral of Eq. 2 as

$$F(x) = \int_{-\infty}^x \mathcal{L}(x|\mu, b)dx = \begin{cases} \frac{1}{2} \exp(\frac{x-\mu}{b}) & \text{if } x < \mu \\ 1 - \frac{1}{2} \exp(-\frac{x-\mu}{b}) & \text{if } x \geq \mu. \end{cases} \quad (10)$$

Then we calculate the definite integral between $[0, d]$ as the occlusion probability $B(d)$, which is defined as $B(d) = F(d) - F(0) = F(d) - \frac{1}{2} \exp(-\frac{\mu}{b})$.

In practice, this is computed very efficiently, without the need to perform the discrete integration of the depth likelihood over the range $[0, d]$. Based on the relationship between visibility and occlusion, we convert the occlusion

probability B to visibility probability V by

$$V(d) = 1 - B(d) = 1 + \frac{1}{2} \exp(-\frac{\mu}{b}) - F(d). \quad (11)$$

To finally compute the visibility in BEV space, we take the maximum visibility probability along the Z axis to form the visibility map V^{BEV} .

$$\tilde{V}^{BEV}(x, y) = \max_{z \in \mathcal{Z}'} V(x, y, z) \quad (12)$$

where $\mathcal{Z}' = \{0, 1, 2 \dots Z' - 1\}$. The V^{BEV} is obtained via interpolation from \tilde{V}^{BEV} .

4.2. Visibility-aware Evaluation

For semantic segmentation where the ground-truth is usually generated using aerial images, it is not possible evaluate predictions in visible and occluded areas by using the standard evaluation metrics. Therefore, in this section, we follow a similar process as the one to generate the visibility map to derive a visibility-aware evaluation method for segmentation in BEV space. In this case, however, we project the lidar 3D points (ground-truth) into multi-view image space and use a depth completion network to obtain multi-view dense depth maps. This depth map is then used as the expected depth value to build a parametric depth representation $F(\theta_{gt})$. We then evaluate the ground-truth depth likelihood on each voxel in 3D space using Eq. 2, forming the ground-truth depth likelihood volume L_{gt} . Finally, we derive the ground-truth visibility map in BEV space V using Eq. 11 and Eq. 12.

In this case, V reflects the maximum visibility of the multi-view cameras in BEV space. Thus, it can be used as a mask to explicitly evaluate results in BEV space subject to visibility. Specifically, we use a threshold τ_{vis} to split the predicted segmentation s_{pred} and ground-truth segmentation label s_{gt} into visible region $\{s_{pred}^{vis}, s_{gt}^{vis}\}$ and occluded region $\{s_{pred}^{occ}, s_{gt}^{occ}\}$. We can then compute the IoU for the visible (IoU_{vis}) and occluded (IoU_{occ}) regions separately as $s^{vis} = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} s(x, y) \times \mathbb{1}(V(x, y) \geq \tau_{vis})$, $s^{occ} = \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} s(x, y) \times \mathbb{1}(V(x, y) < \tau_{occ})$, $IoU_{vis} = \frac{s_{pred}^{vis} \cap s_{gt}^{vis}}{s_{pred}^{vis} \cup s_{gt}^{vis}}$, $IoU_{occ} = \frac{s_{pred}^{occ} \cap s_{gt}^{occ}}{s_{pred}^{occ} \cup s_{gt}^{occ}}$ where $\mathcal{X} = \{0, 1, \dots, X - 1\}$, $\mathcal{Y} = \{0, 1, \dots, Y - 1\}$, and $\mathbb{1}(\cdot)$ is the indicator function. We also report the occlusion rate on nuScenes as the percentage of visible or occluded segmentation labels over total number of segmentation labels.

5. Experiments

In this section, we first detail our experimental settings, then we demonstrate the effectiveness of our approach on the nuScenes dataset, and, finally, we provide ablation studies on the main components of our method.

Camera-based Methods	mAP \uparrow	NDS \uparrow	mATE \downarrow	mASE \downarrow	mAOE \downarrow	mAVE \downarrow	mAAE \downarrow
CenterNet[5]	0.306	0.328	0.716	0.264	0.609	1.426	0.658
FCOS3D[30]	0.343	0.415	0.725	0.263	0.422	1.292	0.153
DETR3D[32]	0.349	0.434	0.716	0.268	0.379	0.842	0.200
PGD[29]	0.369	0.428	0.683	0.26	0.439	1.268	0.185
M ² BEV[33]	0.417	0.470	0.647	0.275	0.377	0.834	0.245
BEVFormer[12] (single-frame version)	0.417	0.448	0.647	0.275	0.377	0.834	0.245
BEVFusion[14] (camera-only version)	0.417	0.321	0.647	0.275	0.377	0.834	0.245
Ours	0.436	0.496	0.637	0.264	0.367	0.810	0.194

Table 1: **Detection results on the nuScenes val. set.** Compared to other camera-based methods, our approach outperforms existing methods for all the metrics except for mASE and mAAE where the performance is slightly lower than PGD [29].

Methods	Modality	mAP \uparrow	NDS \uparrow
PointPillars[9]	Lidar	0.305	0.453
CenterFusion[15]	Camera + Lidar	0.326	0.449
CenterPoint v2[35]	Camera + Lidar + Radar	0.671	0.714
CenterNet[5]	Camera	0.338	0.400
FCOS3D[30]	Camera	0.358	0.428
DETR3D[32]	Camera	0.349	0.434
PGD[29]	Camera	0.386	0.448
M ² BEV[33]	Camera	0.429	0.474
Ours	Camera	0.468	0.495

Table 2: **Detection results on nuScenes test set.** Our method outperforms existing camera based methods for both mAP and NDS.

Camera-based Methods	Road \uparrow	Lane \uparrow
CNN[18]	68.9	16.5
Frozen Encoder[18]	61.6	16.9
PON[22]	60.4	-
OFT[23]	71.6	18.0
LSS[18]	72.9	19.9
M ² BEV[33]	77.2	40.5
Ours* (w/o depth sup.)	77.9	40.8
Ours	78.7	41.0

Table 3: **Segmentation results on the nuScenes validation set.** We report our results compared to other camera-based methods. Our approach outperforms all existing approaches in the literature including M²BEV, demonstrating the benefit of our feature transformation module.

Methods	Vis. (66.9%)		Occ. (33.1%)		All region	
	Road \uparrow	Lane \uparrow	Road \uparrow	Lane \uparrow	Road \uparrow	Lane \uparrow
LSS[18]	79.4	23.1	47.1	6.5	72.9	19.9
M ² BEV[33]	82.9	39.8	48.9	12.8	73.2	36.1
Ours	84.8	41.9	48.9	12.4	73.8	36.5

Table 4: **Segmentation results on the nuScenes validation set under visibility constraints.** We decouple the evaluation of the segmentation results on NuScenes validation set into visible areas (66.9%) and occluded areas (33.1%) based on the visibility map. Our approach performs on par on hallucinated areas and, importantly, for visible areas yields significant improvements compared to existing methods.

5.1. Implementation Details

Dataset. We conduct our experiments on the nuScenes dataset [2]. The nuScenes dataset provides video sequences along with multiple sensor outputs including Lidar, Radar,

Methods	mAP \uparrow	NDS \uparrow	Road \uparrow	Lane \uparrow
M ² BEV[33]	0.408	0.454	75.9	38.0
Ours	0.424	0.467	76.5	39.9

Table 5: **Joint detection and segmentation results on the nuScenes validation set.** We compare our results to the baseline method M²BEV. Our method outperforms the baseline for all the metrics.

Model	mAP \uparrow	mIoU \uparrow	FPS	Memory
M ² BEV[33]	0.408	56.9	1.2	7718
Ours	0.424	58.2	1.3	8902

Table 6: **Performance analysis.** We report frames per second (FPS) and memory requirements for our model and M²BEV when running on a Nvidia Titan V100 GPU.

GPS and IMU, all of which are collected by calibrated and synchronized sensors mounted on an vehicle driving across Boston and Singapore. The dataset consists of 1000 sequences, split into 700 for training and 150 for validation and testing, respectively. Each sample provides six RGB images captured by 6 cameras with divergent viewing directions along with Lidar sparse 3D points, Radar sparse 3D points, GPS pose and IMU readouts. We follow [18, 33] to generate ground-truth segmentation labels from the global map provided by nuScenes dataset.

Evaluation metrics. We report our results using the same metrics as in the nuScenes benchmark. For detection, we report mean Average Precision (mAP) and the nuScenes detection score [2]. For segmentation, we follow LSS [18], and report the mean IoU score (mIoU). In addition, we report results using the proposed visibility-aware evaluation detailed in Sec. 4. Unless specified, we report numbers on the validation set.

Network architecture. We use a unified framework to demonstrate benefits of our depth-based feature transformation module. The network consists of a backbone image encoder and two decoding heads, one for segmentation and one for detection. We use ResNet with deformable convolution as the image encoder. For the decoding heads, we use the same architecture as the one in PointPillars [9].

We set the size of the intermediate 3D volume consisting of $X' \times Y' \times Z' = 400 \times 400 \times 12$ voxels, with a voxel size of $0.25m \times 0.25m \times 0.5m$, respectively. The final BEV space dimension consists of $X \times Y = 200 \times 200$ grids. Each

Lift	Aggregate	mAP	NDS	Road	Lane
PON[22]	PON[22]	-	-	60.4	-
Non-parametric[18]	PP[9]	0.409	0.455	75.9	37.9
Non-parametric[18]	Our Occupancy	0.414	0.459	76.1	38.3
Uniform[33]	PP[9]	0.408	0.454	75.9	38.0
Uniform[33]	Our Occupancy	0.413	0.459	76.0	38.2
Our Parametric depth	PP[9]	0.410	0.457	76.0	38.0
Our Parametric depth	Our Occupancy	0.424	0.467	76.5	39.9

Table 7: **Ablation Study.** Influence of the different components of our feature transformation approach and their comparison to other methods available in the literature.

grid is of size $0.5m \times 0.5m$.

Training and inference. During training, we use 6 RGB images and corresponding camera parameters as input. The training for parametric depth estimation is supervised by the ground-truth sparse Lidar points provided in the dataset. Ground-truth detection and segmentation labels are used to supervise the detection and segmentation heads. We set batch size to 1 per GPU and use 3 nodes with 8 Nvidia V100 GPUs. For inference, our method only requires the 6 input RGB images together with the corresponding camera parameters.

5.2. Results

We now compare our results with M²BEV and other state-of-art methods on the nuScenes dataset. To facilitate the comparison to other approaches, we use ResNeXt-101 as the backbone of our method for detection and segmentation experiments and use ResNet-50 as the backbone for multi-task learning experiments and efficiency analysis.

Detection. We report the results of our method and related state of the art methods in Tab. 1 and Tab. 2, for the validation set and the test set respectively. For the validation set, we only include frame-wise camera-based methods. That is, we exclude those approaches using temporal information. For the test set, we include the latest results including Camera, Lidar, Radar and their combination. As we can see, in both sets, our approach outperforms all existing camera-based methods on both mAP and the NDS score.

Segmentation. We now focus on evaluating our semantic segmentation results. We report our performance compared to state-of-the-art methods on the nuScenes validation set in Tab. 3. We also report a variant of our model trained without depth supervision (Ours*) to fairly compare with LSS [18]. Our method performs significantly better compared to LSS [18] on both road and lane segmentation and slightly better compared to M²BEV [33], the closest method to ours. Our model without depth supervision still outperforms existing methods. Interestingly, if we take the visibility into account, as shown in Tab. 4 and Fig. 2, our method clearly outperforms the baselines on the visible areas while maintain the performance compared to M²BEV on the occluded regions. These results evidence the benefits of our parametric depth approach.

Joint detection and segmentation. Finally, we report results for jointly evaluating both tasks. In this case, we compare our results to the multi-task version of M²BEV. We show results for this experiment in Tab. 5. Our method, once again, outperforms the baseline on both detection and segmentation tasks. These results further evidence the benefits of an improved depth representation in the 2D to 3D feature transformation process.

Efficiency. Our parametric depth estimation requires the estimation of additional parameters compared to simplified depth estimation approaches. As shown in Tab. 6, our model requires slightly larger amount of memory; However, that does not lead to a significant increase in the inference time.

5.3. Ablation Studies

We carry out ablation experiments to study the influence of feature transformations on final detection and segmentation performance and the robustness of our model to calibration error. More ablation experiments can be found in supplementary material. We use ResNet-50 as the backbone for all ablation experiments.

Feature transformations We evaluate the effectiveness of the parametric depth based feature lifting and aggregation module comparing with baseline non-parametric depth based lifting LSS[18], baseline uniform depth based lifting similar to M²BEV and the widely used Pointpillar[9] feature aggregation. Results are in Tab. 7. Our proposed parametric depth based lifting coupled with occupancy based feature aggregation achieved best performance for both detection and segmentation.

Limitations. Like all camera based methods, our method can only provide reliable detection and segmentation results on visible region. On occluded region, although our method can provide hallucination results and visibility information, the results are not reliable for making critical driving decision. Following planning tasks should utilize the visibility and uncertainty information to achieve reliable planning.

6. Conclusion

We propose a parametric depth distribution modeling-based feature transformation that efficiently transforms 2D image features to BEV space. By incorporating visibility inference, our method can provide crucial visibility information to down-streaming planning tasks. Moreover, our approach outperforms existing methods in both detection and segmentation tasks, making it a promising candidate for feature transformation in future works. In our future work, we aim to investigate the integration of temporal information to improve estimation accuracy.

Acknowledgments

Miaomiao Liu was supported by Australian Research Council grants (DE180100628, DP200102274).

References

- [1] Florent Bartoccioni, Éloi Zablocki, Andrei Bursuc, Patrick Pérez, Matthieu Cord, and Karteek Alahari. Lara: Latents and rays for multi-camera bird's-eye-view semantic segmentation. *arXiv preprint arXiv:2206.13294*, 2022. [3](#)
- [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 multi-modal dataset for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11621–11631, 2020. [7](#)
- [3] Chuangrong Chen, Xiaozhi Chen, and Hui Cheng. On the over-smoothing problem of cnn based disparity estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8997–9005, 2019. [4](#)
- [4] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016. [2](#)
- [5] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019. [7](#)
- [6] Shengchao Hu, Li Chen, Penghao Wu, Hongyang Li, Junchi Yan, and Dacheng Tao. St-p3: End-to-end vision-based autonomous driving via spatial-temporal feature learning. In *Computer Vision—ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXVIII*, pages 533–549. Springer, 2022. [3](#)
- [7] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. [3](#)
- [8] Wadim Kehl, Fabian Manhardt, Federico Tombari, Slobodan Ilic, and Nassir Navab. Ssd-6d: Making rgb-based 3d detection and 6d pose estimation great again. In *Proceedings of the IEEE international conference on computer vision*, pages 1521–1529, 2017. [2](#)
- [9] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. [2](#), [3](#), [6](#), [7](#), [8](#)
- [10] Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with dynamic temporal stereo. *arXiv preprint arXiv:2209.10248*, 2022. [3](#)
- [11] 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. *arXiv preprint arXiv:2206.10092*, 2022. [3](#)
- [12] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. [3](#), [7](#)
- [13] Yingfei Liu, Junjie Yan, Fan Jia, Shuailin Li, Qi Gao, Tiancai Wang, Xiangyu Zhang, and Jian Sun. Petrv2: A unified framework for 3d perception from multi-camera images. *arXiv preprint arXiv:2206.01256*, 2022. [3](#)
- [14] Zhijian Liu, Haotian Tang, Alexander Amini, Xinyu Yang, Huizi Mao, Daniela Rus, and Song Han. Bevfusion: Multi-task multi-sensor fusion with unified bird's-eye view representation. *arXiv preprint arXiv:2205.13542*, 2022. [7](#)
- [15] Ramin Nabati and Hairong Qi. Centerfusion: Center-based radar and camera fusion for 3d object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021. [7](#)
- [16] 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*, 5(3):4867–4873, 2020. [1](#), [2](#)
- [17] Jinhyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris Kitani, Masayoshi Tomizuka, and Wei Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. *arXiv preprint arXiv:2210.02443*, 2022. [3](#)
- [18] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *ECCV*. Springer, 2020. [1](#), [2](#), [3](#), [5](#), [7](#), [8](#)
- [19] Patrick Poirson, Phil Ammirato, Cheng-Yang Fu, Wei Liu, Jana Kosecka, and Alexander C Berg. Fast single shot detection and pose estimation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 676–684. IEEE, 2016. [2](#)
- [20] Rui Qian, Divyansh Garg, Yan Wang, Yurong You, Serge Belongie, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. End-to-end pseudo-lidar for image-based 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5881–5890, 2020. [2](#)
- [21] Zengyi Qin, Jinglu Wang, and Yan Lu. Monogmet: A geometric reasoning network for monocular 3d object localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8851–8858, 2019. [2](#)
- [22] Thomas Roddick and Roberto Cipolla. Predicting semantic map representations from images using pyramid occupancy networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020. [1](#), [2](#), [7](#), [8](#)
- [23] Thomas Roddick, Alex Kendall, and Roberto Cipolla. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*, 2018. [7](#)
- [24] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10529–10538, 2020. [2](#)
- [25] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Point-rcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019. [2](#)
- [26] Andrea Simonelli, Samuel Rota Buló, Lorenzo Porzi, Manuel López-Antequera, and Peter Kontschieder. Disentangling monocular 3d object detection. In *Proceedings of*

- the IEEE/CVF International Conference on Computer Vision*, pages 1991–1999, 2019. [2](#)
- [27] Fabio Tosi, Yiyi Liao, Carolin Schmitt, and Andreas Geiger. Smd-nets: Stereo mixture density networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8942–8952, 2021. [4](#)
- [28] Tai Wang, Jiangmiao Pang, and Dahua Lin. Monocular 3d object detection with depth from motion. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, pages 386–403. Springer, 2022. [3](#)
- [29] Tai Wang, ZHU Xinge, Jiangmiao Pang, and Dahua Lin. Probabilistic and geometric depth: Detecting objects in perspective. In *Conference on Robot Learning*. PMLR, 2022. [7](#)
- [30] Tai Wang, Xinge Zhu, Jiangmiao Pang, and Dahua Lin. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 913–922, 2021. [7](#)
- [31] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q Weinberger. Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019. [2](#)
- [32] Yue Wang, Vitor Campagnolo Guizilini, Tianyuan Zhang, Yilun Wang, Hang Zhao, and Justin Solomon. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, pages 180–191. PMLR, 2022. [2](#), [7](#)
- [33] Enze Xie, Zhiding Yu, Daquan Zhou, Jonah Philion, Anima Anandkumar, Sanja Fidler, Ping Luo, and Jose M Alvarez. M²bev: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. *arXiv preprint arXiv:2204.05088*, 2022. [1](#), [2](#), [3](#), [5](#), [7](#), [8](#)
- [34] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10):3337, 2018. [2](#)
- [35] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11784–11793, 2021. [2](#), [7](#)
- [36] Yunpeng Zhang, Zheng Zhu, Wenzhao Zheng, Junjie Huang, Guan Huang, Jie Zhou, and Jiwen Lu. Beverse: Unified perception and prediction in birds-eye-view for vision-centric autonomous driving. *arXiv preprint arXiv:2205.09743*, 2022. [3](#)