M-BEV: Masked BEV Perception for Robust Autonomous Driving

Siran Chen[1,2], **Yue Ma** [4], **Yu Qiao**[1, 3], **Yali Wang**[1, 3, *]

[1] Shenzhen Institute of Advanced Technology, Chinese Academy of Science, Shenzhen, China
[2] School of Artificial Intelligence, University of Chinese Academy of Science, Beijing, China
[3] Shanghai Artificial Intelligence Laboratory, Shanghai, China
[4] Tsinghua Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
Chensiran17@mails.ucas.ac.cn, y-ma21@mails.Tsinghua.edu.cn,
{yu.qiao, yl.wang}@siat.ac.cn

Abstract

3D perception is a critical problem in autonomous driving. Recently, the Bird's-Eye-View (BEV) approach has attracted extensive attention, due to low-cost deployment and desirable vision detection capacity. However, the existing models ignore a realistic scenario during the driving procedure, i.e., one or more view cameras may be failed, which largely deteriorates the performance. To tackle this problem, we propose a generic Masked BEV (M-BEV) perception framework, which can effectively improve robustness to this challenging scenario, by random masking and reconstructing camera views in the end-to-end training. More specifically, we develop a novel Masked View Reconstruction (MVR) module for M-BEV. It mimics various missing cases by randomly masking features of different camera views, then leverages the original features of these views as self-supervision, and reconstructs the masked ones with the distinct spatio-temporal context across views. Via such a plug-and-play MVR, our M-BEV is capable of learning the missing views from the resting ones, and thus well generalized for robust view recovery and accurate perception in the testing. We perform extensive experiments on the popular NuScenes benchmark, where our framework can significantly boost 3D perception performance of the state-of-the-art models on various missing view cases, e.g., for the absence of back view, our M-BEV promotes the PETRv2 model with 10.3% mAP gain.

Introduction

3D perception of surrounding scenes is the key for autonomous driving. Compared to LiDAR-based methods (Mohapatra et al. 2021; Barrera et al. 2020; Ma et al. 2022a; Zhou et al. 2020), camera-based approaches has attracted increasing attention(Li et al. 2022d; Philion and Fidler 2020; Roddick, Kendall, and Cipolla 2018; Wang et al. 2019) since they are easy and cheap for deployment. In particular, the Bird's-Eye-View (BEV) based methods have been highlighted by learning the holistic representation from multi-camera images (Roddick, Kendall, and Cipolla 2018; Wang et al. 2019; Philion and Fidler 2020; Li et al. 2022d,c, 2023; Wang et al. 2022b; Liu et al. 2022a; Chen et al. 2023). Basically, these approaches integrate 2D image information from six distinct views to encode a unified 3D representation of visible scenes, and then decode it to accurately capture the size and location of objects in the surrounding. However, these approaches work on the ideal case in which six cameras always work well, while one or more cameras may be failed or broken down during the realistic driving procedure. In such an emergency, the existing BEV-based approaches would be largely deteriorated, due to the lack of the corresponding visual clues from the missing views. For example, the NDS and mAP of PETRv2 (Liu et al. 2023) have a decrease of 12.4% and 18.0% respectively when the back camera view is missing, which severely affects the safety and reliability of autonomous driving system.

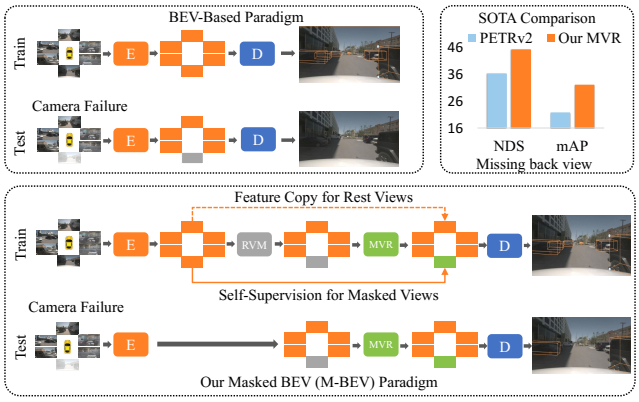


Figure 1: Motivation. In cases where cameras crash, the existing BEV-based approaches would be largely deteriorated. We design a self-supervised Masked View Reconstruction (MVR) module which can significantly boost the state-of-the-art models for various missing camera cases. E: Encoder, D: Decoder, RVM: Random View Masking.

To alleviate this problem, we propose a concise Masked BEV (M-BEV) perception framework, which can effectively boost model's robustness to missing camera views, by randomly masking and recovering view features in the end-to-end training procedure. Specifically, we design a self-supervised Masked View Reconstruction (MVR) module in our M-BEV, we randomly mask the features of different camera views in the training epochs, then leverage the features of the rest views as spatio-temporal context, and re-

*Corresponding author

cover the features of the masked views as their original features. Via such learning, MVR masters the capacity of reconstructing the missing views from the rest views, and thus can effectively tackle the camera failure cases in the testing.

Note that, there are two critical differences between our MVR and MAE (He et al. 2022). *First*, the masking goal and design are different. MAE aims at learning scalable image representation in general. Hence, it masks several *patches* of each image and reconstructs them from the rest patches. Alternatively, our MVR aims at tackling the missing camera cases in BEV-based driving. Hence, it masks several *images* of six camera views and reconstructs them from the rest images. More importantly, besides using all the rest images for reconstruction, we also propose to exploit the distinct contexts in the surrounding images for reconstruction, based on spatio-temporal overlaps across BEV cameras. *Second*, the training and testing manners are different. In the training stage, MAE uses the *same* proportion for masking patches of each input image in all the epochs, while our MVR uses *different* proportion for masking images of six views in the different epochs, in order to contain various missing camera views which can possibly happen. In the testing stage, MAE mainly uses *encoder* as general feature extractor and ignores the decoder. Alternatively, our MVR uses *decoder* to reconstruct the missing camera views and leverages the recovered views for 3D perception in the BEV-based driving.

Finally, we implement our general M-BEV framework on two state-of-the-art BEV-based models, i.e., PETRv2 (Liu et al. 2023), and BEVStereo (Li et al. 2023) for 3D object detection in autonomous driving, since they maintain a preferable accuracy-efficiency balance. To verify the effectiveness, we perform extensive experiments on various missing camera cases, on the popular NuScenes(Caesar et al. 2020) dataset. The results show that, M-BEV framework significantly boosts the performance of the SOTA models for various missing camera emergencies. For example, for the absence of back view, M-BEV helps to boost PETRv2 baseline with 10.3% mAP improvement, while only takes extra 0.6ms for once response.

Related Work

Multi-View 3D Object Detection. 3D object detection is one of the key technologies for autonomous driving, which takes LiDAR (Zhou and Tuzel 2018; Ma et al. 2022c; Li et al. 2021; Lang et al. 2019), camera (Philion and Fidler 2020; Huang et al. 2021; Liu et al. 2022a; Wang et al. 2021), or multi-modal input data (Liu et al. 2022b; Xu et al. 2021; Ma et al. 2022b; Yin, Zhou, and Krähenbühl 2021) to predict the location, size, velocity, and category of the targets in real 3D space. Cameras-based methods (Li et al. 2022d; Liu et al. 2023; Ma et al. 2023b; Li et al. 2022c, 2023) stand out, due to the low cost and easy access to visual data from six camera views. The key problem of these works is the conversion between 2D and 3D space, and BEV representation works as a suitable bond. OFT(Roddick, Kendall, and Cipolla 2018) first makes a direct transformation from 2D features to 3D BEV features for monocular 3D object detection. The following works expand this style, by learnable 3D object queries (Wang et al. 2022b), 3D position-aware embedding (Liu et al. 2022a, 2023), temporal information integration (Li et al. 2022d; Ma et al. 2023a; Huang et al. 2021; Huang and Huang 2022; Liu et al. 2023), etc. Additionally, depth supervision is another main direction for 3D performance enhancement (Li et al. 2022c; Chen et al. 2022; Huang et al. 2022; Li et al. 2023). However, all of these existing BEV-based methods rely on the high-quality camera inputs in the ideal case. When the camera views fails in practice, their performance declines severely. RoboBEV (Zhu et al. 2023) establishes a comprehensive driving benchmark under various natural and adversarial corruptions. MetaBEV(Ge et al. 2023) solves the problem by cross-modal data fusion with both Camera and LiDAR. As far as we know, our M-BEV is the first camera-only solution for such view failure with great robustness.

Masked Visual Modeling. Masked modeling pipeline was first used in NLP (Radford et al. 2019). The masking operation is treated as a noise type and processed by traditional denoising encoders(Vincent et al. 2008). Then ViT(Dosovitskiy et al. 2020) uses masked token prediction and paves the way for self-supervised pre-training. More recently, MAE(He et al. 2022) is introduced as an asymmetric transformer-based encoder-decoder architecture, which is achieved by reconstructing the pixels of the masked image. The pre-trained autoencoder could be applied for various downstream tasks with minor fine-tuning. Similar thoughts are raised by BEiT(Bao et al. 2021), BEVT(Wang et al. 2022a), and VIMPAC(Tan et al. 2021) while the reconstruction is based on token-level. MaskFeat(Wei et al. 2022) chooses to reconstruct HOG(Dalal and Triggs 2005) features of the masked token as self-supervised pre-training. Moreover, UM-MAE(Li et al. 2022b) and SemMAE(Li et al. 2022a) design distinctive masking strategies, VideoMAE(Tong et al. 2022; Wang et al. 2023) series simply expand masking in the temporal dimension of videos and achieve impressive performance. Compared to MAE-style design, our MVR has two critical differences which has been carefully discussed in the introduction.

Method

Overall Architecture

The overall architecture of M-BEV is shown in Fig. 2, it consists of 3D object detector, Random View Masking (RVM), and Masked View Reconstruction (MVR) modules. Basically, the 3D object detector contains a visual encoder, translator, and decoder. First, the encoder transforms the input images of multiple camera views into their corresponding 2D visual features. Second, the translator transforms these 2D features into 3D-relevant features, by 3D position embedding (Liu et al. 2023, 2022a), depth estimation (Li et al. 2022c, 2023; Chen et al. 2022), attention mechanism (Li et al. 2022d; Yang et al. 2022; Ma et al. 2022d; Wang et al. 2022b), etc. Finally, the decoder transforms 3D features for final object detection. Note that, when one or more cameras fail, there are no corresponding features from encoder. To address this, we incorporate the RVM and MVR modules after the encoder to recover the missing features. In this case, we denote the combination of translator and decoder as an

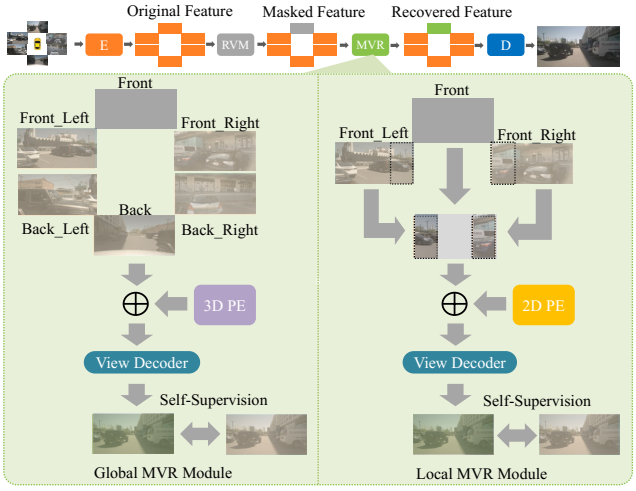


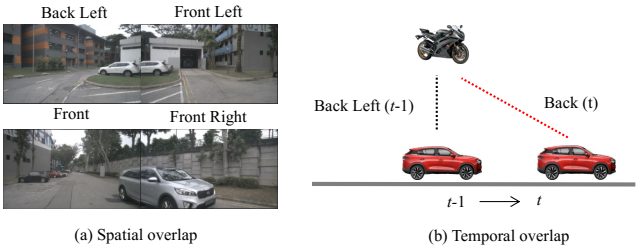
Figure 2: Overall Architecture of Our Proposed M-BEV Framework. M-BEV consists of 3D Object Detector with Visual Encoder (E) and Decoder (D), Random View Masking (RVM) and Masked View Reconstruction (MVR) modules. We propose two MVR modules, namely Global MVR and Local MVR, to recover the features of masked views.

integrated decoder in this paper without loss of generality.

To mimic the emergency situation of camera failure in the testing phrase, we propose to perform view masking and recovering in the training phrase. Specifically, RVM is used to randomly select the camera views for masking, and MVR is used to recover the masked views by exploiting spatio-temporal context from the rest views. Note that, recovering is performed on the feature-level of the views from encoder, instead of raw images like in MAE (He et al. 2022). The main reason is that our goal is to reconstruct the missing views for 3D object detection, if we reconstruct the raw images, these predicted images have to be fed into the visual encoder again to obtain features for detection, this makes the whole paradigm tedious with unnecessary processing and computation. Alternatively, if we reconstruct the features of the masked images, we could directly use the visual encoder(e.g., VoVNet, ViT, ResNet, etc) in the BEV-based models for both detection and reconstruction. These recovered features can be straightforwardly used for subsequent decoding without any difficulty. In this case, we can effectively leverage the self-supervised advantages of mask modeling, by only adding MVR with a lightweight feature decoder.

Masked View Reconstruction Module

Typically, BEV-based approaches (Li et al. 2022d; Huang and Huang 2022; Liu et al. 2023; Jiang et al. 2022; Li et al. 2022c; Lin et al. 2022) take the images of six camera views at previous step $t-1$ and current step t as input. After encoding these images as visual features, we use RVM module to randomly mask the features of several camera views. Note that in real situation, for the missing views, all previous frames of this view are lost, so we mask both frames



(a) Spatial overlap (b) Temporal overlap

Figure 3: Illustration of spatial and temporal overlap.

from $t-1$ and t. The next question is how to recover the features of these masked views. In our task, the entire image is masked, we can not explore the relations within an image like the MAE style, but the relations between the six images instead. As shown in Fig. 3, six camera views share the overlapped regions, which offer the distinct spatio-temporal clues, i.e., the same object might appear in different views at different time steps. Based on this observation, we propose two types of MVR module to recover the features of masked views, by using the features of rest views.

Global MVR Module. In this design, we use all the rest views as context to reconstruct the masked ones. First, we concatenate the features of all the views. For the missing views, we pad them with the shared and learned masked tokens which are randomly initialized, i.e., $\mathbf{V}_{mask}$. For the rest views, we use their corresponding 2D features from encoder, i.e., $\mathbf{F}_{rest}$. Second, we need to distinguish the location of these views to encode 3D relationships among them. Hence, we apply 3D position embedding (Liu et al. 2022a) to encode 3D coordinates of different views, i.e., $\mathbf{P}_{3D}$. Finally, we add all the features with the corresponding position embedding, and feed them into a feature decoder for reconstructing the masked features. The decoder is composed of transformer blocks which are the same as MAE (He et al. 2022), except that the last layer outputs the feature tokens, instead of the raw pixels of the masked views.

$$\mathbf{U}_{mask} = Decoder(\mathcal{C}[\mathbf{V}_{mask},\ \mathbf{F}_{rest}] + \mathbf{P}_{3D}), \qquad (1)$$

$\mathbf{U}_{mask}$ is the reconstructed features of the masked views, $\mathcal{C}$ means we concat the tokens.

Local MVR Module. In fact, a camera view is strongly correlated to its adjacent views, instead of all the views. As shown in Fig. 2, the left part of Front view is relevant to the right part of Front_Left view, while the right part of Front view is relevant to the left part of Front_Right view. Hence, using all the views of Global MVR may introduce the noisy reconstruction, with irrelevant surrounding scenes. Based on this analysis, we propose to design a local MVR module, which uses the relevant context from the neighboring views to recover the missing views. Specifically, for the feature of a masked view, we divide it into three parts, i.e., the left, middle, and right parts, with a dividing portion ratio. We investigate different portion ratios in our experiments. (1) *Left Part*: As shown in Fig. 2, for the left part of this masked view, we refer to the adjacent camera on its left view. In particular, we crop the right part of left-view features $\mathbf{F}_{left}(right)$ at $t-1$ and t with the same ratio. Then, we use the mean of them

as the left part of the masked view for both time stamps. (2) *Right Part*: Similarly, for the right part of this masked view, we refer to the adjacent camera on its right view. In particular, we crop the left part of right-view features $\mathbf{F}_{right}(left)$ at $t-1$ and t with the same ratio. Then, we use the mean of them as the right part of the masked view for both time stamps. (3) *Middle Part*: For the middle part of this masked view, there is no clue. Hence, we use the masked tokens just like Global MVR, i.e., $\mathbf{V}_{mask}(mid)$. As a result, the masked view becomes the concatenation of these features,

$$\mathbf{V}_{mask} = \mathcal{C}[\mathbf{F}_{left}(right),\ \mathbf{V}_{mask}(mid),\ \mathbf{F}_{right}(left)]. \quad (2)$$

Multiple views can be randomly masked. If the adjacent views of a masked view are also missing, there are no left and/or right clues for this view. In this case, we use the masked features $\mathbf{V}_{mask}(left)$ or/and $\mathbf{V}_{mask}(right)$ in the corresponding left or/and right parts. Finally, since $\mathbf{V}_{mask}$ is the 2D feature for a camera view, we use 2D positional embedding $\mathbf{P}_{2D}$ by the sine-cosine version (He et al. 2022), and feed the sum of $\mathbf{V}_{mask}$ and $\mathbf{P}_{2D}$ into the decoder,

$$\mathbf{U}_{mask} = Decoder(\mathbf{V}_{mask} + \mathbf{P}_{2D}), \quad (3)$$

where $\mathbf{U}_{mask}$ is the reconstructed features of the masked views. Additionally, the decoder for Local MVR has the same transformer structure as the one for Global MVR. But it only needs to process the tokens of one image size in Local MVR, instead of all the tokens in the six images of Global MVR. Hence, the computation cost of Local MVR is less expensive than Global MVR.

Training and Testing

Training. After obtaining the reconstructed features of the masked views $\mathbf{U}_{mask}$ from MVR, we compute the L2 loss between the original features $\mathbf{F}_{mask}$ and the reconstructed ones for self-supervised pretraining of our M-BEV, we freeze the encoder during pretraining.

$$\mathcal{L}_{mvr} = \|\mathbf{F}_{mask} - \mathbf{U}_{mask}\|_2. \quad (4)$$

However, only feature supervision is not enough for perception, we further fine-tune our M-BEV with 3D detection supervision end to end,

$$\mathcal{L}_{total} = \mathcal{L}_{det} + \alpha\mathcal{L}_{mvr}, \quad (5)$$

where $\mathcal{L}_{det}$ is the 3D detection loss which consists of focal loss (Lin et al. 2017) for object classification and L1 loss for 3D bounding box regression. Since our goal is to boost detection in case of emergency, the detection loss should be the major loss. In this case, we set a weight coefficient $\alpha = 0.05$ for the reconstruction loss in the fine-tuning. Moreover, we randomly select different number of camera views in different training epochs, e.g., one view can be randomly picked for masking in the previous epoch, while four views can be randomly picked for masking in the current epoch. This is different from the training style of MAE (He et al. 2022), where the same masking proportion is adopted in all the epochs. The main reason is that our goal is to tackle all the missing view cases which can possibly happen. Hence, the training procedure should cover various missing views by random masking in different epochs. This design has not been attempted in previous research on mask modeling.

Testing. In the testing process, we preserve the view decoder in MVR to predict the features of missing views. This design can effectively tackle the camera failure emergency in the driving procedure. Additionally, for the regular case without missing views, our M-BEV still works well, where we can simply ignore the trained MVR and feed the features of encoder into 3D detection decoder. In fact, our trained M-BEV is better for the regular case, compared to the baseline model (without view masking and recovering). This is mainly because that training with MVR can generalize our M-BEV model for various hard masking cases. In our experiment, we have validated this conclusion (Table. 5).

Experiments

Datasets and Metrics

We conduct our experiments on the popular NuScenes dataset (Caesar et al. 2020). NuScenes is a large-scale benchmark for autonomous driving, where the data is collected from 1000 real driving scenes with around 20 seconds duration. The scenes are divided: 700 of them for training, and 150 each for validation and testing. We report the officially used metrics of 3D object detection in BEV-based research(Caesar et al. 2020; Lang et al. 2019; Wang et al. 2022b) , i.e., mean Average Precision(mAP) and five True Positive metrics, including mean Average Translation Error (mATE), mean Average Scale Error (mASE), mean Average Orientation Error(mAOE), mean Average Velocity Error(mAVE), mean Average Attribute Error(mAAE), where the lower value is better. Besides, the NuScenes Detection Score (NDS) comprehensively reflects these metrics, and it is the most concerned metric for performance evaluation.

Implementation Details

Driving scenes in the real world are often complex, we choose a challenging setting to mimic real situation, that is, we randomly discard images of the corresponding views using our RVM module, all previous frames for the view are also missing, so we can evaluate how other views help for the reconstruction of the missing ones. For the baseline models, we follow the official implementation on open-sourced code bases. For PETRv2(Liu et al. 2023), we use images of 320×800 resolution as input, and the visual backbone is pretrained VoVNet-99(Lee and Park 2020). For BEVStereo(Li et al. 2023), the model is obtained by the official code on GitHub with ResNet-50(He et al. 2016) as visual encoder and uses 256x704 input resolution. The models are trained with official settings and get comparable performance with the official report. And for inference, we do the evaluation on all possible situations. The MVR module is fine-tuned for 48 epochs, the learning rate is set to 2.0×10^{-4}. The transformer layer of decoder is four, and the hidden dimension is 512. We use 8 A5000 GPUs for all experiments. No test-time augmentation methods are used during inference.

SOTA Comparison

As mentioned before, we apply M-BEV paradigm on two recent state-of-the-art approaches, PETRv2(Liu et al. 2023) and BEVStereo(Li et al. 2023), where we insert our local

W/O Missing	Method	NDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
Standard	PETRv2 (Liu et al. 2023)	0.4853	0.3977	0.7531	0.2693	0.4978	0.4310	0.1840
Missing	**Method**	**NDS ↑**	**mAP ↑**	**mATE ↓**	**mASE ↓**	**mAOE ↓**	**mAVE ↓**	**mAAE ↓**
Front	PETRv2 (Liu et al. 2023)	0.4238	0.3022	0.7757	0.2740	0.5311	0.5043	0.1883
	Our M-BEV (PETRv2)	**0.4504**	**0.3263**	0.7234	0.2736	0.4996	0.4449	0.1862
Front_Right	PETRv2 (Liu et al. 2023)	0.4363	0.3294	0.8444	0.2706	0.5333	0.4551	0.1808
	Our M-BEV (PETRv2)	**0.4712**	**0.3666**	0.7346	0.2704	0.5098	0.4214	0.1850
Front_Left	PETRv2 (Liu et al. 2023)	0.4405	0.3355	0.8195	0.2733	0.5216	0.4664	0.1912
	Our M-BEV (PETRv2)	**0.4678**	**0.3628**	0.7308	0.2740	0.5113	0.4298	0.1905
Back	PETRv2 (Liu et al. 2023)	0.3616	0.2179	1.0176	0.2977	0.5618	0.4477	0.1726
	Our M-BEV (PETRv2)	**0.4516**	**0.3206**	0.7283	0.2688	0.4908	0.4237	0.1754
Back_Left	PETRv2 (Liu et al. 2023)	0.4568	0.3513	0.7910	0.2700	0.4903	0.4476	0.1895
	Our M-BEV (PETRv2)	**0.4753**	**0.3694**	0.7277	0.2694	0.4770	0.4291	0.1909
Back_Right	PETRv2 (Liu et al. 2023)	0.4556	0.3544	0.7892	0.2700	0.5157	0.4508	0.1902
	Our M-BEV (PETRv2)	**0.4756**	**0.3730**	0.7294	0.2711	0.4990	0.4211	0.1879

Table 1: Performance comparison on PETRv2 (Liu et al. 2023) when losing each of six camera views. The effect of our M-BEV is impressive, e.g., for the absence of back view, our M-BEV achieves 10.3% mAP improvement.

W/O Missing	Method	NDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
Standard	BEVStereo (Li et al. 2023)	0.4432	0.3439	0.6583	0.2823	0.5860	0.5287	0.2327
Missing	**Method**	**NDS ↑**	**mAP ↑**	**mATE ↓**	**mASE ↓**	**mAOE ↓**	**mAVE ↓**	**mAAE ↓**
Front	BEVStereo (Li et al. 2023)	0.3901	0.2462	0.6799	0.2867	0.6192	0.5159	0.2283
	Our M-BEV (BEVStereo)	0.**4027**	**0.2667**	0.6744	0.2866	0.6223	0.5103	0.2130
Front_Right	BEVStereo (Li et al. 2023)	0.4042	0.2832	0.6770	0.2841	0.6027	0.5811	0.2298
	Our M-BEV (BEVStereo)	**0.4148**	**0.3004**	0.6733	0.2845	0.6122	0.5791	0.2153
Front_Left	BEVStereo (Li et al. 2023)	0.4039	0.2795	0.6720	0.2820	0.5960	0.5645	0.2429
	Our M-BEV (BEVStereo)	**0.4126**	**0.2967**	0.6725	0.2851	0.6134	0.5690	0.2269
Back	BEVStereo (Li et al. 2023)	0.3894	0.2373	0.6761	0.2840	0.6071	0.5126	0.2126
	Our M-BEV (BEVStereo)	**0.3949**	**0.2580**	0.6703	0.2880	0.6196	0.5540	0.2087
Back_Left	BEVStereo (Li et al. 2023)	0.4132	0.2891	0.6753	0.2828	0.5687	0.5457	0.2408
	Our M-BEV (BEVStereo)	**0.4149**	**0.2989**	0.6762	0.2859	0.5966	0.5709	0.2259
Back_Right	BEVStereo (Li et al. 2023)	0.4081	0.2898	0.6813	0.2817	0.5945	0.5739	0.2367
	Our M-BEV (BEVStereo)	**0.4157**	**0.3027**	0.6719	0.2843	0.6052	0.5738	0.2209

Table 2: Performance comparison on BEVStereo (Li et al. 2023) when losing each of six camera views. Overall, our M-BEV gives a comprehensive improvement for the baseline.

MVR after the visual encoder in these models. We compare our M-BEV paradigm with the original BEV paradigm on these models. Moreover, to explicitly evaluate the effectiveness, we investigate the result for missing each of the six camera views. As shown in Table 1 and Table 2, our M-BEV comprehensively improves the performance, compared to the original model. For example, M-BEV is remarkable for the back-view camera failure, with a 10.3% mAP growth on the PETRv2 baseline. All these prove the effectiveness of our design.

Ablation Study

Global MVR *v.s.* Local MVR. We first ablate the effect of Global MVR and Local MVR, where we use the PETRv2 baseline due to its good performance. Both two MVR variants are trained with the same schedule when each camera view loses, and all other hyper-parameters keep the same. The results are shown in Table. 3, Local MVR method outperforms Global MVR in all metrics. This superiority may be attributed to the fact that, Local MVR exploits the distinct spatio-temporal context across adjacent cameras, instead of using all the views which may contain noise.

Number of Missing Camera Views. It is natural to evaluate the robustness if more than one camera is lost in real-world scenarios. Note that, there are several possible missing view choices for each setting, e.g., there are 6/15/20/15/6 choices for missing 1/2/3/4/5 views. Hence, we compute the NDS and mAP metrics for each choice and average them as the final NDS and mAP metrics. As mentioned in the training section, we train a single model to handle all these situations to show our robustness. As shown in Fig. 4, our method outperforms the baseline PETRv2 for all the missing cases, especially when the number of missing views increases. It clearly shows the robustness of our M-BEV. Moreover, Local MVR is consistently better than Global MVR, based on the exploration of distinct contexts across adjacent views.

MVR Designs. To verify the contributions of the strategies in our proposed MVR module, we conduct ablation experiments on several settings which will be explained in detail below. All ablations are conducted with the Local MVR method which performs better. In Table4, we ablate the MVR module employed in the model, fine-tuning process, the position embedding(PE), and the masking ratio in sequence to validate how they contribute to the final results,

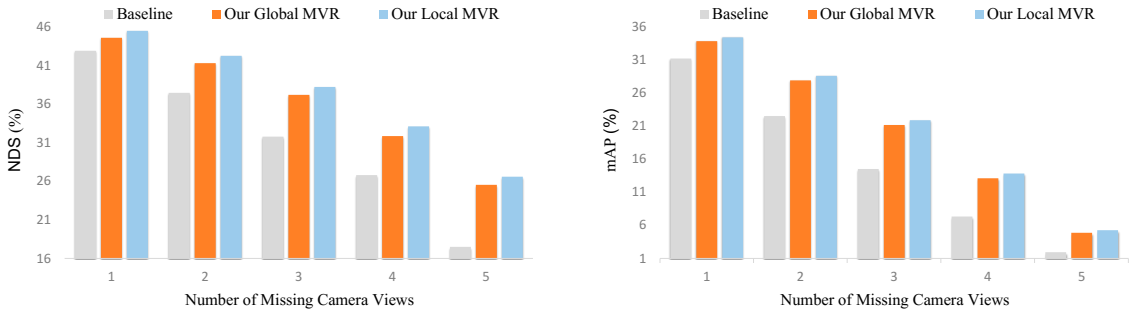


Figure 4: Performance of baseline, our Global MVR and Local MVR settings for different number of missing cameras.

Missing	Our M-BEV	NDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
Front	Global MVR	0.4361	0.3160	0.7524	0.2725	0.5489	0.4584	0.1873
	Local MVR	**0.4504**	**0.3263**	0.7234	0.2736	0.4996	0.4449	0.1862
Front_Right	Global MVR	0.4588	0.3592	0.7622	0.2684	0.5550	0.4371	0.1852
	Local MVR	**0.4712**	**0.3666**	0.7346	0.2704	0.5098	0.4214	0.1850
Front_Left	Global MVR	0.4574	0.3570	0.7635	0.2727	0.5410	0.4410	0.1928
	Local MVR	**0.4678**	**0.3628**	0.7308	0.2740	0.5113	0.4298	0.1905
Back	Global MVR	0.4390	0.3146	0.7642	0.2679	0.5340	0.4400	0.1773
	Local MVR	**0.4516**	**0.3206**	0.7283	0.2688	0.4908	0.4237	0.1754
Back_Left	Global MVR	0.4632	0.3633	0.7645	0.2697	0.5214	0.4350	0.1944
	Local MVR	**0.4753**	**0.3694**	0.7277	0.2694	0.4770	0.4291	0.1909
Back_Right	Global MVR	0.4629	0.3671	0.7638	0.2694	0.5504	0.4333	0.1895
	Local MVR	**0.4756**	**0.3730**	0.7294	0.2711	0.4990	0.4211	0.1879

Table 3: The performance after reconstruction of two MVR module variants. Local MVR is a better choice by exploiting the distinct context from adjacent camera views.

the results show the average values under one random camera failure. Note that the best performance is obtained with all the strategies used and 76% masking ratio, put in bold in Table4. (1)First, we ablate the MVR module which is the core of our M-BEV approach. To verify its effectiveness, we remove our proposed MVR module and perform the same fine-tuning using RVM module. The final results show a drop in NDS and mAP by 1.14% and 1.28% respectively compared to the best model with MVR, indicating that our reconstructed features could offer extra information beyond the original model. (2)Then we explore the influence of fine-tuning process which aims to alleviate the domain gap between reconstruction task and detection task. The model without fine-tuning shows a drop in NDS and mAP by 2.61% and 3.69%, which strongly confirms the significance of fine-tuning with our RVM module. (3)Next, as the composition of input for MVR decoder, PE is added to the 2D feature tokens for better localization. The ablation shows the influence of PE which indeed makes a progress. The model with PE have an improvement of 0.28% NDS and 0.22% mAP on average compared to model without it. (4)Finally, we ablate the masking ratio. The masking ratio depends on the dividing portion we use from the neighboring overlaps, the left and right views could both offer prompts for the missing camera, so we only need to mask the middle part. A proper ratio close to the real situation is also crucial for good reconstruction. We have evaluated different masking ratios from 60% to 80%, the optimal masking ratio is 76% for both NDS and mAP.

Segmentation. We also evaluate our M-BEV for map segmentation tasks on nuScenes with PETRv2(Liu et al. 2023) baseline, where only the prediction head needs to be changed. When missing one camera view, the IoU scores of Drive, Lane and Vehicle drop form 79.5%, 46.2% and 49.9% to 76.6%, 41.1% and 43.5% respectively on average, while with our MVR, the IoU scores are 78.2%, 45.1% and 45.6%, much better than that of original model.

Generalization vs. Computation. Finally, we evaluate the generalization and computation cost of M-BEV. As shown in the Table. 5, for the regular cases without missing views, we can directly deactivate the MVR module after training, the GFLOPS keep the same as the baseline. However, the model co-trained with MVR performs better than the baseline, showing that, M-BEV paradigm can generalize the learning capacity for 3D object detection, by masking view modeling. For the camera failure case (one view missing setting), local MVR has better performance with little extra computation cost, and it only takes about 6ms for the local MVR to response for once detection, which is negligible. All these prove its potential for practical application.

Visualization

In Fig. 5, we give some examples of the reconstructed features of missing cameras and the detection results of the

MVR	FineTune	PE	MaskRatio	NDS ↑	mAP ↑	mATE ↓	mASE ↓	mAOE ↓	mAVE ↓	mAAE ↓
✗	✗	✗	✗	0.4285	0.3117	0.8279	0.2730	0.5228	0.4638	0.1865
✗	✓	✗	✗	0.4466	0.3371	0.7752	0.2741	0.5256	0.4469	0.1975
✓	✗	✓	76%	0.4319	0.3130	0.8196	0.2721	0.5181	0.4545	0.1852
✓	✓	✗	76%	0.4552	0.3477	0.7556	0.2715	0.5190	0.4376	0.1870
✓	✓	✓	76%	**0.4580**	**0.3499**	0.7544	0.2712	0.5186	0.4368	0.1878
✓	✓	✓	60%	0.4576	0.3495	0.7538	0.2710	0.5171	0.4402	0.1885
✓	✓	✓	64%	0.4566	0.3474	0.7529	0.2720	0.5185	0.4378	0.1894
✓	✓	✓	68%	0.4570	0.3491	0.7596	0.2714	0.5246	0.4302	0.1892
✓	✓	✓	72%	0.4579	0.3485	0.7540	0.2721	0.5126	0.4352	0.1895
✓	✓	✓	80%	0.4560	0.3478	0.7578	0.2716	0.5249	0.4330	0.1916

Table 4: The ablation of MVR module, fine-tuning, position embedding and masking ratio for M-BEV. All models are trained with same schedule and hyper-parameters. We randomly drop one camera and calculate the average metrics for inference.

Figure 5: Visualization for feature maps and detection results. M-BEV reconstructed features could be a rough substitute of the original feature, and could help for the detection of the vehicles on the left and right sides of the missing view.

W/O Missing	NDS↑	mAP↑	GFLOPS↓
PETRv2 Baseline	0.4853	0.3977	1047
Our Global MVR	0.4872	0.4007	1047
Our Local MVR	**0.4898**	**0.4039**	1047
Missing	**NDS↑**	**mAP↑**	**GFLOPS↓**
PETRv2 Baseline	0.4285	0.3117	1047
Our Global MVR	0.4534	0.3467	1088
Our Local MVR	**0.4580**	**0.3499**	1053

Table 5: Generalization vs. Computation. The model trained with our MVR performs better the original baseline under both no-missing and missing settings, while requiring only little extra GFLOPS for computation.

missing view. As shown in Fig. 5, our M-BEV reconstructed features could be a rough substitute of the original features, from which we can see the outline of the road and the major targets.For the detection results, our M-BEV is helpful for detection of the vehicles on left and right sides of the missing view. For example, due to the missing of back view, PETRv2 model can't detect any object in the view, but with our reconstruction, the vehicles near the overlap regions could be detected, while it's still hard to detect the small and far objects in the middle part, which may need further exploration.

Conclusion and Future Work

Recent researches have primarily focused on improving detection performance, while our work focuses on the robustness of these models, which is essential for ensuring driving safety. In this paper, we put forward a novel reconstruction architecture to address the emergence of camera crashes. To compensate for the lost information of missing camera views, we design a distinct MVR module that leverages the related tokens from neighboring cameras. The reconstructed image features are capable of boosting the detection results, compared to the original models. Furthermore, M-BEV has great generalization ability and requires little extra computation. Extensive experiments verify the effectiveness of our M-BEV, which could be widely applied as a plug-and-play module to enhance the robustness of 3D perception models.

Acknowledgements

This work was supported by the National Key R&D Program of China(NO.2022ZD0160505), and the Joint Lab of CAS-HK.

References

Bao, H.; Dong, L.; Piao, S.; and Wei, F. 2021. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*.

Barrera, A.; Guindel, C.; Beltrán, J.; and García, F. 2020. Birdnet+: End-to-end 3d object detection in lidar bird's eye view. In *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 1–6. IEEE.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11621–11631.

Chen, S.; Xu, Q.; Ma, Y.; Qiao, Y.; and Wang, Y. 2023. Attentive Snippet Prompting for Video Retrieval. *IEEE Transactions on Multimedia*.

Chen, Z.; Li, Z.; Zhang, S.; Fang, L.; Jiang, Q.; and Zhao, F. 2022. BEVDistill: Cross-Modal BEV Distillation for Multi-View 3D Object Detection. *arXiv preprint arXiv:2211.09386*.

Dalal, N.; and Triggs, B. 2005. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, 886–893. Ieee.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Ge, C.; Chen, J.; Xie, E.; Wang, Z.; Hong, L.; Lu, H.; Li, Z.; and Luo, P. 2023. Metabev: Solving sensor failures for 3d detection and map segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8721–8731.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 16000–16009.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Huang, J.; and Huang, G. 2022. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*.

Huang, J.; Huang, G.; Zhu, Z.; and Du, D. 2021. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*.

Huang, P.; Liu, L.; Zhang, R.; Zhang, S.; Xu, X.; Wang, B.; and Liu, G. 2022. TiG-BEV: Multi-view BEV 3D Object Detection via Target Inner-Geometry Learning. *arXiv preprint arXiv:2212.13979*.

Jiang, Y.; Zhang, L.; Miao, Z.; Zhu, X.; Gao, J.; Hu, W.; and Jiang, Y.-G. 2022. Polarformer: Multi-camera 3d object detection with polar transformers. *arXiv preprint arXiv:2206.15398*.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12697–12705.

Lee, Y.; and Park, J. 2020. Centermask: Real-time anchor-free instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13906–13915.

Li, G.; Zheng, H.; Liu, D.; Wang, C.; Su, B.; and Zheng, C. 2022a. Semmae: Semantic-guided masking for learning masked autoencoders. *arXiv preprint arXiv:2206.10207*.

Li, X.; Wang, W.; Yang, L.; and Yang, J. 2022b. Uniform masking: Enabling mae pre-training for pyramid-based vision transformers with locality. *arXiv preprint arXiv:2205.10063*.

Li, Y.; Bao, H.; Ge, Z.; Yang, J.; Sun, J.; and Li, Z. 2023. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, 1486–1494.

Li, Y.; Ge, Z.; Yu, G.; Yang, J.; Wang, Z.; Shi, Y.; Sun, J.; and Li, Z. 2022c. Bevdepth: Acquisition of reliable depth for multi-view 3d object detection. *arXiv preprint arXiv:2206.10092*.

Li, Z.; Wang, W.; Li, H.; Xie, E.; Sima, C.; Lu, T.; Qiao, Y.; and Dai, J. 2022d. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IX*, 1–18. Springer.

Li, Z.; Yao, Y.; Quan, Z.; Yang, W.; and Xie, J. 2021. Sienet: Spatial information enhancement network for 3d object detection from point cloud. *arXiv preprint arXiv:2103.15396*.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Lin, X.; Lin, T.; Pei, Z.; Huang, L.; and Su, Z. 2022. Sparse4D: Multi-view 3D Object Detection with Sparse Spatial-Temporal Fusion. *arXiv preprint arXiv:2211.10581*.

Liu, Y.; Wang, T.; Zhang, X.; and Sun, J. 2022a. Petr: Position embedding transformation for multi-view 3d object detection. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVII*, 531–548. Springer.

Liu, Y.; Yan, J.; Jia, F.; Li, S.; Gao, A.; Wang, T.; and Zhang, X. 2023. Petrv2: A unified framework for 3d perception from multi-camera images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 3262–3272.

Liu, Z.; Tang, H.; Amini, A.; Yang, X.; Mao, H.; Rus, D.; and Han, S. 2022b. BEVFusion: Multi-Task Multi-Sensor

Fusion with Unified Bird's-Eye View Representation. *arXiv preprint arXiv:2205.13542*.

Ma, R.; Chen, C.; Yang, B.; Li, D.; Wang, H.; Cong, Y.; and Hu, Z. 2022a. CG-SSD: Corner guided single stage 3D object detection from LiDAR point cloud. *ISPRS Journal of Photogrammetry and Remote Sensing*, 191: 33–48.

Ma, Y.; Cun, X.; He, Y.; Qi, C.; Wang, X.; Shan, Y.; Li, X.; and Chen, Q. 2023a. MagicStick: Controllable Video Editing via Control Handle Transformations. *arXiv preprint arXiv:2312.03047*.

Ma, Y.; He, Y.; Cun, X.; Wang, X.; Shan, Y.; Li, X.; and Chen, Q. 2023b. Follow Your Pose: Pose-Guided Text-to-Video Generation using Pose-Free Videos. *arXiv preprint arXiv:2304.01186*.

Ma, Y.; Wang, Y.; Wu, Y.; Lyu, Z.; Chen, S.; Li, X.; and Qiao, Y. 2022b. Visual knowledge graph for human action reasoning in videos. In *Proceedings of the 30th ACM International Conference on Multimedia*, 4132–4141.

Ma, Y.; Yang, T.; Shan, Y.; and Li, X. 2022c. SimVTP: Simple Video Text Pre-training with Masked Autoencoders. *arXiv preprint arXiv:2212.03490*.

Ma, Z.; Li, J.; Li, G.; and Huang, K. 2022d. Cmal: A novel cross-modal associative learning framework for vision-language pre-training. In *Proceedings of the 30th ACM International Conference on Multimedia*, 4515–4524.

Mohapatra, S.; Yogamani, S.; Gotzig, H.; Milz, S.; and Mader, P. 2021. BEVDetNet: bird's eye view LiDAR point cloud based real-time 3D object detection for autonomous driving. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2809–2815. IEEE.

Philion, J.; and Fidler, S. 2020. 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*, 194–210. Springer.

Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8): 9.

Roddick, T.; Kendall, A.; and Cipolla, R. 2018. Orthographic feature transform for monocular 3d object detection. *arXiv preprint arXiv:1811.08188*.

Tan, H.; Lei, J.; Wolf, T.; and Bansal, M. 2021. Vimpac: Video pre-training via masked token prediction and contrastive learning. *arXiv preprint arXiv:2106.11250*.

Tong, Z.; Song, Y.; Wang, J.; and Wang, L. 2022. Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training. *arXiv preprint arXiv:2203.12602*.

Vincent, P.; Larochelle, H.; Bengio, Y.; and Manzagol, P.-A. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, 1096–1103.

Wang, L.; Huang, B.; Zhao, Z.; Tong, Z.; He, Y.; Wang, Y.; Wang, Y.; and Qiao, Y. 2023. VideoMAE V2: Scaling Video Masked Autoencoders with Dual Masking. *arXiv preprint arXiv:2303.16727*.

Wang, R.; Chen, D.; Wu, Z.; Chen, Y.; Dai, X.; Liu, M.; Jiang, Y.-G.; Zhou, L.; and Yuan, L. 2022a. Bevt: Bert pretraining of video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14733–14743.

Wang, T.; Zhu, X.; Pang, J.; and Lin, D. 2021. Fcos3d: Fully convolutional one-stage monocular 3d object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 913–922.

Wang, Y.; Chao, W.-L.; Garg, D.; Hariharan, B.; Campbell, M.; and Weinberger, K. Q. 2019. 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*, 8445–8453.

Wang, Y.; Guizilini, V. C.; Zhang, T.; Wang, Y.; Zhao, H.; and Solomon, J. 2022b. Detr3d: 3d object detection from multi-view images via 3d-to-2d queries. In *Conference on Robot Learning*, 180–191. PMLR.

Wei, C.; Fan, H.; Xie, S.; Wu, C.-Y.; Yuille, A.; and Feichtenhofer, C. 2022. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 14668–14678.

Xu, S.; Zhou, D.; Fang, J.; Yin, J.; Bin, Z.; and Zhang, L. 2021. Fusionpainting: Multimodal fusion with adaptive attention for 3d object detection. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 3047–3054. IEEE.

Yang, C.; Chen, Y.; Tian, H.; Tao, C.; Zhu, X.; Zhang, Z.; Huang, G.; Li, H.; Qiao, Y.; Lu, L.; et al. 2022. BEVFormer v2: Adapting Modern Image Backbones to Bird's-Eye-View Recognition via Perspective Supervision. *arXiv preprint arXiv:2211.10439*.

Yin, T.; Zhou, X.; and Krähenbühl, P. 2021. Multimodal virtual point 3d detection. *Advances in Neural Information Processing Systems*, 34: 16494–16507.

Zhou, Y.; Sun, P.; Zhang, Y.; Anguelov, D.; Gao, J.; Ouyang, T.; Guo, J.; Ngiam, J.; and Vasudevan, V. 2020. End-to-end multi-view fusion for 3d object detection in lidar point clouds. In *Conference on Robot Learning*, 923–932. PMLR.

Zhou, Y.; and Tuzel, O. 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4490–4499.

Zhu, Z.; Zhang, Y.; Chen, H.; Dong, Y.; Zhao, S.; Ding, W.; Zhong, J.; and Zheng, S. 2023. Understanding the Robustness of 3D Object Detection With Bird's-Eye-View Representations in Autonomous Driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 21600–21610.