FB-OCC: 3D Occupancy Prediction based on Forward-Backward View Transformation

Zhiqi Li[1,2], Zhiding Yu[1], David Austin[1], Mingsheng Fang[2], Shiyi Li[1], Jan Kautz[1], Jose M. Alvarez[1]
[1]NVIDIA [2]Nanjing University

Abstract

This technical report summarizes the winning solution for the 3D Occupancy Prediction Challenge, which is held in conjunction with the CVPR 2023 Workshop on End-to-End Autonomous Driving and CVPR 23 Workshop on Vision-Centric Autonomous Driving Workshop. Our proposed solution FB-OCC builds upon FB-BEV, a cutting-edge vision-based 3D object detection method. On the basis of FB-BEV, we study novel designs and optimization tailored to the 3D occupancy prediction task, including joint depth-semantic pretraining, joint voxel-BEV representation, model scaling up, and effective post-processing strategies. These designs and optimization result in a state-of-the-art mIoU score of 54.19% on the nuScenes dataset and ranks the 1st place in the challenge. Code will be released at: https://github.com/NVlabs/FB-BEV.

1. Introduction

3D occupancy prediction, which refers to predicting the occupancy status and semantic class of every voxel in a 3D voxel space, is an important task in autonomous vehicle (AV) perception. Predicting 3D occupancy is important to the development of safe and robust self-driving systems by providing rich information to the planning stack [6]. The challenge track requires participants to developing occupancy prediction algorithms that solely utilize camera input during inference. In addition, the challenge permits the use of open-source datasets and models, which facilitates the exploration of data-driven algorithms and large-scale models. The impact of this challenge is significant by providing a playground for the latest state-of-the-art 3D occupancy prediction algorithms in real-world scenarios.

In the context of challenge, besides our efforts in model structure design, we emphasize the importance of both model scale and model pre-training techniques. This focus stems from several motivations. First, it should be mentioned that there have been a number of state-of-the-art BEV-based solutions. These solutions can be adapted to 3D occupancy prediction with certain modifications. However, there is still limited knowledge regarding the impact of large-scale models and pre-training on the occupancy prediction task. As will be reported in this work, the use of large-scale models and pre-training techniques stands as crucial factors contributing to our success.

2. Our Solution

In this section, we will present our solution in details with the following aspects covered. Section 2.1 will elaborate on the design of our model structure. Section 2.2 will discuss the efforts in model pre-training and scaling up. Finally, Section 2.3 will outline the post-processing techniques employed in our study.

2.1. Model Structure Designs

Our solution, FB-OCC, builds upon a 3D object detection method termed FB-BEV. Here, we provide a brief introduction to facilitate a better understanding of FB-OCC. The central module of the camera-only 3D perception model is the view transformation module. This module encompasses two prominent view transformations: forward projection (represented by List-Splat-Shoot [16]) and backward projection (represented by BEVFormer [12]). FB-BEV provides a unified design that leverages both methods, promoting the benefits from each method with improved perception results while overcoming their limitations. In the case of FB-OCC, we use forward projection to generate the initial 3D voxel representation. We then condense the 3D voxel representations into a flattened BEV feature map. The BEV feature map is treated as queries within the BEV space and attends the image encoder features to acquire dense geometry information. The fusion features of the 3D voxel representation and the optimized BEV representations are then fed into the subsequent task head.

In the forward projection module, we adhere to the principles of LSS [16] to account for the uncertainty in the depth estimation of each pixel. This allows us to project the image features into the 3D space based on their corresponding depth values. In contrast to LSS, which models BEV features, we directly model 3D voxel representations to capture more detailed information in the 3D space. Additionally, we adopt the approach of BEVDepth [11] to uti-

lize point clouds in generating accurate depth ground truth, which helps supervise the depth prediction of our model for improved accuracy. LSS tends to produce relatively sparse 3D representations. To tackle this issue, we incorporate a backward projection method to optimize these sparse 3D representations. Considering the computational burden, we employ BEV representation instead of 3D voxel representation at this stage. The backward projection method draws inspiration from BEVFormer [12]. However, unlike BEVFormer, which employs randomly initialized parameters as BEV queries, we compress the obtained 3D voxel representation into a BEV representation, thereby incorporating stronger semantic priors. Furthermore, our backward projection method leverages the depth distribution during the projection phase, enabling more precise modeling of projection relationships.

Following the acquisition of the 3D voxel representations and optimized BEV representation, we combine them through the process of expanding the BEV features, resulting in the final 3D voxel representations. The voxel encoder and the occupancy prediction head, as depicted in Figure 1 and Figure 2, are outlined below.

To train the model, we use a distance-aware Focal loss function L_{fl} inspired by M2BEV [21], Dice loss L_{dl}, affinity loss L_{scal}^{geo} and L_{scal}^{sem} from MonoScene [2], lovasz-softmax loss L_{ls} from OpenOccupancy [20]. In addition, we also need a depth supervision loss L_d and a 2D semantic loss L_s, which will be introduced in the next section.

2.2. Scaling up and Pretraining

Scaling the model size has traditionally been a convenient approach to improving model accuracy. However, in the field of 3D vision-only perception, researchers have discovered that employing a more powerful 2D backbone often leads to overfitting [7]. For instance, on the nuScenes 3D object detection task, the largest backbone, such as VIT-L [4] with approximately 300M parameters, and commonly used backbones like ConvNext-B [13] and VoVNet-99 [10] with around 100M parameters, tend to encounter this issue. To address this challenge, we explore the utilization of the 1B-parameter backbone, InternImage-H [19], for multi-camera 3D perception tasks. However, directly applying this backbone would result in severe overfitting due to the limited number of samples available for training, specifically the 40K samples in the nuScenes dataset [1]. To overcome this limitation, we leverage the opportunity provided by this competition, which allows participants to utilize additional public data. By augmenting our data resources, we can train our large-scale model more effectively. Building upon the open-source InternImage-H checkpoint, we conduct model training on the Object365 dataset [17], which is a vast 2D object detection dataset comprising 2 million images. This pre-training on large-scale 2D detection tasks enhances the model's semantic perception capabilities. However, there still exists a certain domain gap when applying the pre-trained model to downstream 3D perception tasks. Therefore, we further perform targeted pre-training on the model specifically for 3D perception tasks. An effective approach for pre-training is to enhance the model's geometric awareness through depth estimation tasks. Consequently, we conduct extensive pre-training on the nuScenes dataset, primarily focusing on depth estimation. It is worth noting that depth pre-training lacks semantic-level supervision. To mitigate the risk of the model becoming excessively biased towards depth information, potentially leading to the loss of semantic priors (especially given the large-scale nature of the model, which is prone to overfitting), we simultaneously aim to predict the 2D semantic segmentation labels alongside the depth prediction task, as shown in Figure 3.

However, nuScenes does not provide semantic segmentation labels for 2D images. To address this issue, we employ the popular Segment Anything Model (SAM) [9] for automatic labeling. For thing categories with bounding box annotations provided by nuScenes, we utilize box prompts to generate high-quality semantic masks for each object. Unfortunately, for stuff categories such as road surfaces or buildings, bounding box annotations are not available. Nonetheless, nuScenes offers corresponding point cloud semantic segmentation labels for these categories.

To generate semantic masks for stuff categories, we project the LiDAR points belonging to these categories onto the image. For each category, we randomly select three points as point prompts to generate the corresponding semantic masks. The quality of the produced stuff masks is satisfactory. With the 2D image semantic mask labels and the ground truth depth maps, we train the model using a joint depth estimation task and semantic segmentation task. This pre-training task closely aligns with the final occupancy prediction task, enabling the direct generation of 3D occupancy results using depth values and semantic labels. The pre-trained model serves as an improved starting point for the subsequent training of the occupancy prediction task.

2.3. Post-Processing

2.3.1 Test-Time Augmentation

During the inference phase, we employ several techniques to improve the prediction accuracy. Firstly, we horizontally flip the input image and horizontally and vertically flip the 3D space, resulting in a total of eight prediction results for the current frame. The final prediction result is obtained by calculating the mean average of all these results. Additionally, we utilize a temporal Test-Time Augmentation (TTA) strategy. We observed that the accuracy of occupancy prediction significantly deteriorates with distance. To mitigate this issue, for static voxels, we can leverage the predicted

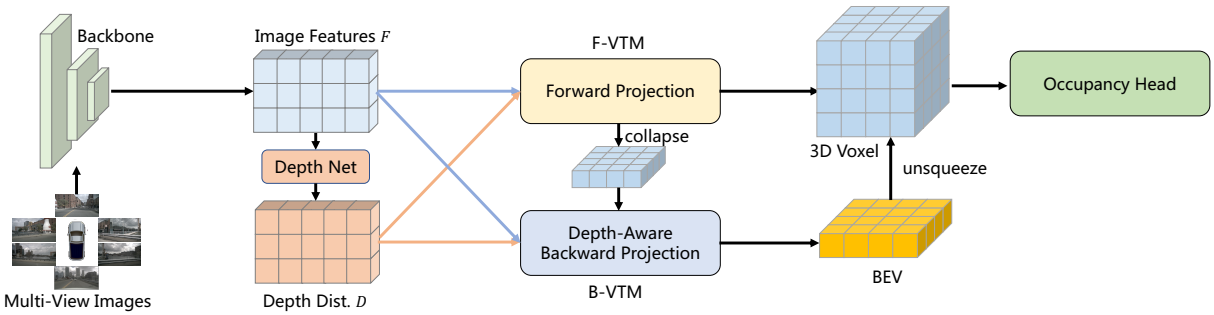

Figure 1: **Overall architecture of FB-OCC. The F-VTM is based on the forward projection strategy (LSS), and the B-VTM is based on the backward projection strategy (BEVFormer).**

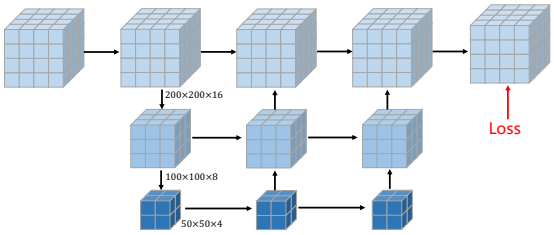

Figure 2: **The architecture of the occupancy prediction head of FB-OCC.**

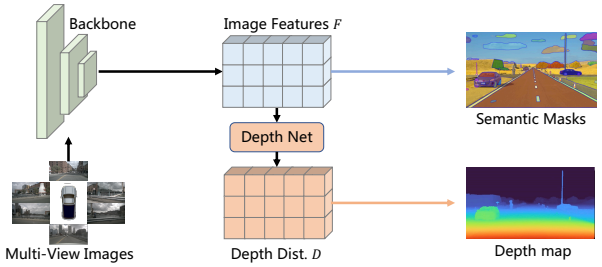

Figure 3: **The joint depth and semantic pretraining.**

voxels that are close to the ego car in previous frames to replace the voxels in the same location of the current frame, where the distance to the ego car is greater. By incorporating these strategies, we aim to enhance the overall prediction accuracy of the model.

2.3.2 Ensemble

For our ensemble strategy, we perform a weighted sum of all independent results. The weight of each voxel is determined by two factors. The first factor is the model weight, which is related to the overall mIoU of each result. The second factor is the specific category weight, which is related to the IoU of this voxel's category. We use NNI [14] to search the different weight values automatically.

3. Experiments

3.1. Datasets and Metrics

Dataset The occupancy dataset is built based on the existing nuScenes dataset [1, 18]. For each frame, they provide occupancy annotations within the range of [-40m, -40m, -1m, 40m, 40m, 5.4m], and the resolution of each voxel is 0.4m. The dataset contains 18 classes, where one indicates a free voxel that is occupied by nothing. The dataset also provides the camera mask to indicate whether the voxel is visible from any cameras.

Metrics For this challenge, we mainly evaluate our models based on mIoU, which can be formulated as follows:

$$mIoU = \frac{1}{C}\sum_{c=1}^{C}\frac{TP_c}{TP_c + FP_c + FN_c}, \quad (1)$$

where TP_c, FP_c, and FB_c correspond to the number of true positive, false positive, and false negative predictions for class c, and C is the total number of classes.

3.2. Implementation Details

Training Strategies. For training large-scale models, we use a batch size of 32 on 32 NVIDIA A100 GPUs, AdamW optimizer with a learning rate of 1×10^{-4} and a weight-decay of 0.05. The learning rate of the backbone is 10 times smaller. We train our models around 50 epochs for occupancy tasks. The temporal windows used by every model are determined based on the GPU memory. For the Intern-H backbone, we use 6 previous frames. When GPU memory is sufficient, we use up to 16 historical frames. Following SOLOFusion [15], we use online temporal sequences during training which is much more efficient.

Network Details For large-scale models, the image features from the backbone are downsampled with a stride of 16. The input image scale is 640×1600. We use commonly used data augmentation strategies, including flip, and rotation on both image and 3D space. The depth net predicts

Method	Input	others	barrier	bicycle	bus	car	construction vehicle	motorcycle	pedestrian	traffic cone	trailer	truck	driveable surface	other flat	sidewalk	terrain	manmade	vegetation	mIoU
MonoScene [2]	C	1.75	7.23	4.26	4.93	9.38	5.67	3.98	3.01	5.90	4.45	7.17	14.91	6.32	7.92	7.43	1.01	7.65	6.06
BEVDet [8]	C	2.09	15.29	0.0	4.18	12.97	1.35	0.0	0.43	0.13	6.59	6.66	52.72	19.04	26.45	21.78	14.51	15.26	11.73
BEVFormer [12]	C	5.85	37.83	17.87	40.44	42.43	7.36	23.88	21.81	20.98	22.38	30.70	55.35	28.36	36.0	28.06	20.04	17.69	26.88
CTF-Occ [18]	C	8.09	39.33	20.56	38.29	42.24	16.93	24.52	22.72	21.05	22.98	31.11	53.33	33.84	37.98	33.23	20.79	18.0	28.53
Version A	C	0.04	37.15	16.81	34.17	38.22	13.41	16.97	19.69	18.94	11.65	21.94	55.94	26.98	29.65	26.92	10.24	14.33	23.12
Version B	C	0.03	40.94	21.16	39.22	40.75	20.57	23.85	23.6	24.95	16.63	26.36	59.42	27.57	31.39	29.03	16.69	18.42	27.09
Version C	C	0.02	45.18	25.26	44.55	47.38	22.63	26.24	26.92	27.91	26.4	32.1	76.97	37.2	44.84	47.81	37.0	32.64	35.36
Version D	C	12.17	44.83	25.73	42.61	47.97	23.16	25.17	25.77	26.72	31.31	34.89	78.83	41.42	49.06	52.22	39.07	34.61	37.39
Version E	C	13.57	44.74	27.01	45.41	49.1	25.15	26.33	27.86	27.79	32.28	36.75	80.07	42.76	51.18	55.13	42.19	37.53	39.11
Version F	C	13.66	45.88	28.26	44.91	49.78	26.21	28.84	28.27	27.89	32.75	37.56	81.2	43.46	52.13	56.35	42.79	38.1	39.89
Version G	C	14.41	45.77	29.19	45.29	50.53	27.86	29.01	28.15	28.61	32.89	37.86	81.76	45.52	53.99	58.69	43.49	38.75	40.69
Version H	C	14.30	49.71	30.0	46.62	51.54	29.3	29.13	29.35	30.48	34.97	39.36	83.07	47.16	55.62	59.88	44.89	39.58	42.06

Table 1: 3D occupancy prediction performance of different settings on the Occ3D-nuScenes dataset [18].

Method	params.	others	barrier	bicycle	bus	car	construction vehicle	motorcycle	pedestrian	traffic cone	trailer	truck	driveable surface	other flat	sidewalk	terrain	manmade	vegetation	mIoU
Version H	67.8M	14.30	49.71	30.0	46.62	51.54	29.3	29.13	29.35	30.48	34.97	39.36	83.07	47.16	55.62	59.88	44.89	39.58	42.06
Version I	130.8M	14.26	57.02	38.34	57.69	62.12	34.35	39.43	38.82	39.42	42.91	50.02	86.04	50.24	60.06	62.54	52.36	45.68	48.90
Version J	428.8M	16.74	55.33	39.77	58.94	61.79	32.04	42.63	40.51	39.06	43.72	51.33	87.34	53.77	62.63	66.06	56.63	49.74	50.47
Version K	1200.0M	28.28	56.70	44.35	51.37	61.81	35.12	47.38	41.56	39.88	57.96	48.39	86.66	56.97	64.66	61.23	62.78	52.35	52.79

Table 2: Results of models in different scales.

80 discrete depth categories covering the depth from 2m to 42m. The resolution of generated 3D voxel features is $200 \times 200 \times 16$. The backward projection module uses 1 layer since the input BEV queries already contain meaningful information. During the training phase, we ignore the invisible voxels from cameras.

3.3. Ablations

Training large-scale models requires huge computing resources. In our exploration phase, we verify the effect of different methods on a smaller model scale. For this smaller model, the input scale is 256×704, and the resolution is $100 \times 100 \times 8$ and the image backbone is ResNet-50 [5]. We list the milestones of our exploration in Table. 1. Version A is our naive baseline. In Version B, we use depth supervision following BEVDepth. For Version C, we ignore the invisible voxels from cameras during the training phase. Version D model fixed several serious bugs of Version C, especially for the abnormal IoU of **other** category. For Version E, we used temporal information from the previous 16 frames. We leverage the joint depth and semantic pretraining in Version F. For Version G, we optimized the loss function by adding Dice loss and using 3D transformation to align voxel features from different timesteps. The results of Version H is the test-time augmentation results of Version G.

3.4. Scaling Up

After exploring the basic design of FB-OCC, we scale up the model size by using larger backbones and image input size, as shown in Table 2. Compared to Version H, version I uses VoVnet-99 backbone and other modifications, including using 960×1760 image input and a voxel resolution of 0.4m. Compared to Version I, Version J leverages a ViT-L backbone and ViT-adapter [3]. For our most powerful Version K, we scale up the model to over 1 billion parameters with the InternImage-H backbone.

3.5. Ensemble

In our final submission, we employed our ensemble strategy with seven models to improve accuracy. The main difference between different models is the use of different backbones. By combining the results of all the models through ensemble techniques, we were able to achieve our best result on the test set with a mIoU score of 54.19%.

References

[1] Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020) 2, 3

[2] Cao, A.Q., de Charette, R.: Monoscene: Monocular 3d semantic scene completion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3991–4001 (2022) 2, 4

[3] Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision transformer adapter for dense predictions. arXiv preprint arXiv:2205.08534 (2022) 4

[4] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 2

[5] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016) 4

[6] Hu, Y., Yang, J., Chen, L., Li, K., Sima, C., Zhu, X., Chai, S., Du, S., Lin, T., Wang, W., et al.: Planning-oriented autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 17853–17862 (2023) 1

[7] Huang, J., Huang, G., Zhu, Z., Du, D.: Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. arXiv preprint arXiv:2112.11790 (2021) 2

[8] Hunag, J.: Bevdet codebase. https://github.com/HuangJunJie2017/BEVDet/ (2022) 4

[9] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv preprint arXiv:2304.02643 (2023) 2

[10] Lee, Y., Hwang, J.w., Lee, S., Bae, Y., Park, J.: An energy and gpu-computation efficient backbone network for real-time object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019) 2

[11] Li, Y., Ge, Z., Yu, G., Yang, J., Wang, Z., Shi, Y., Sun, J., Li, Z.: BEVDepth: Acquisition of reliable depth for multi-view 3d object detection. arXiv preprint arXiv:2206.10092 (2022) 1

[12] Li, Z., Wang, W., Li, H., Xie, E., Sima, C., Lu, T., Yu, Q., Dai, J.: Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. arXiv preprint arXiv:2203.17270 (2022) 1, 2, 4

[13] Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11976–11986 (2022) 2

[14] Microsoft: Neural Network Intelligence (Jan 2021), https://github.com/microsoft/nni 3

[15] Park, J., Xu, C., Yang, S., Keutzer, K., Kitani, K., Tomizuka, M., Zhan, W.: Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. arXiv preprint arXiv:2210.02443 (2022) 3

[16] Philion, J., Fidler, S.: Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In: European Conference on Computer Vision. pp. 194–210. Springer (2020) 1

[17] Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., Sun, J.: Objects365: A large-scale, high-quality dataset for object detection. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 8430–8439 (2019) 2

[18] Tian, X., Jiang, T., Yun, L., Wang, Y., Wang, Y., Zhao, H.: Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. arXiv preprint arXiv:2304.14365 (2023) 3, 4

[19] Wang, W., Dai, J., Chen, Z., Huang, Z., Li, Z., Zhu, X., Hu, X., Lu, T., Lu, L., Li, H., et al.: Internimage: Exploring large-scale vision foundation models with deformable convolutions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14408–14419 (2023) 2

[20] Wang, X., Zhu, Z., Xu, W., Zhang, Y., Wei, Y., Chi, X., Ye, Y., Du, D., Lu, J., Wang, X.: Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. arXiv preprint arXiv:2303.03991 (2023) 2

[21] Xie, E., Yu, Z., Zhou, D., Philion, J., Anandkumar, A., Fidler, S., Luo, P., Alvarez, J.M.: M^2BEV: Multi-camera joint 3d detection and segmentation with unified birds-eye view representation. arXiv preprint arXiv:2204.05088 (2022) 2