AdaOcc: Adaptive Forward View Transformation and Flow Modeling for 3D Occupancy and Flow Prediction

Dubing Chen[1]*, Wencheng Han[1], Jin Fang[2], Jianbing Shen[1]†
[1]SKL-IOTSC, CIS, University of Macau.
[2]Inceptio Tech.
{dobbin.chen, wencheng256, fangjin19900820}@gmail.com, jianbingshen@um.edu.mo

Abstract

In this technical report, we present our solution for the Vision-Centric 3D Occupancy and Flow Prediction track in the nuScenes Open-Occ Dataset Challenge at CVPR 2024. Our innovative approach involves a dual-stage framework that enhances 3D occupancy and flow predictions by incorporating adaptive forward view transformation and flow modeling. Initially, we independently train the occupancy model, followed by flow prediction using sequential frame integration. Our method combines regression with classification to address scale variations in different scenes, and leverages predicted flow to warp current voxel features to future frames, guided by future frame ground truth. Experimental results on the nuScenes dataset demonstrate significant improvements in accuracy and robustness, showcasing the effectiveness of our approach in real-world scenarios. Our single model based on Swin-Base ranks second on the public leaderboard, validating the potential of our method in advancing autonomous car perception systems.

1. Introduction

3D occupancy and flow prediction [10, 16–18] are critical components of autonomous driving perception systems. They involve determining the occupancy status, semantic class, and future position of each voxel in a 3D voxel space. These predictions provide rich semantic and geometric information, crucial for understanding and navigating complex driving environments. The CVPR24 occupancy and flow prediction competition [17] focuses on developing new algorithms that predict occupancy and flow solely from camera input during inference, offering a significant platform for advancing state-of-the-art 3D occupancy and flow prediction algorithms.

Our approach in this competition emphasizes innovative model design. We developed a two-stage framework to separately predict occupancy and flow. In the first stage, we train the occupancy model independently. We propose an adaptive forward view transformation method to enhance the adaptability of depth-based LSS. In the second stage, we train the flow model based on the occupancy model from the first stage. We introduced a novel sequential prediction method, using adjacent frames as inputs for the flow network. We combined regression with classification to predict flow, addressing the issue of varying flow scales in different scenes. Additionally, we use the predicted flow to warp the current frame's voxel features to the future frame and supervise using the future frame's ground truth, further enhancing prediction accuracy.

Our approach achieves an Occ Score of 0.453 without any post-hoc process, achieving 2nd place in this challenge.

2. Our Solution

This section will present our solution in detail. Inspired by the general vision-based occupancy prediction pipeline, our model comprises three main components: voxel feature encoding, occupancy prediction head, and flow prediction head. We elaborate on the design of our model in Sec. 2.1 and Sec. 2.2, and introduce further enhancement techniques in Sec. 2.3.

2.1. Voxel Feature Encoding

We first extract image features, then employ Lift-Splat-Shoot (LSS) [15] to transform the 2D features into 3D space. Previous methods typically use depth probability [8] as weights for LSS, which weakens its adaptability due to its unimodal distribution. To enhance adaptability, we integrate semantic information into the depth probability. We supervise the depth using LiDAR points and apply segmentation loss to the image features. Given the initial sparse nature of voxel features obtained via LSS, we employ the inverse process of trilinear interpolation to densify these 3D

*This work was partly done during internship in Inceptio.

†Corresponding author: *Jianbing Shen*. This work was supported in part by the FDCT grants 0102/2023/RIA2 and 0154/2022/A3.

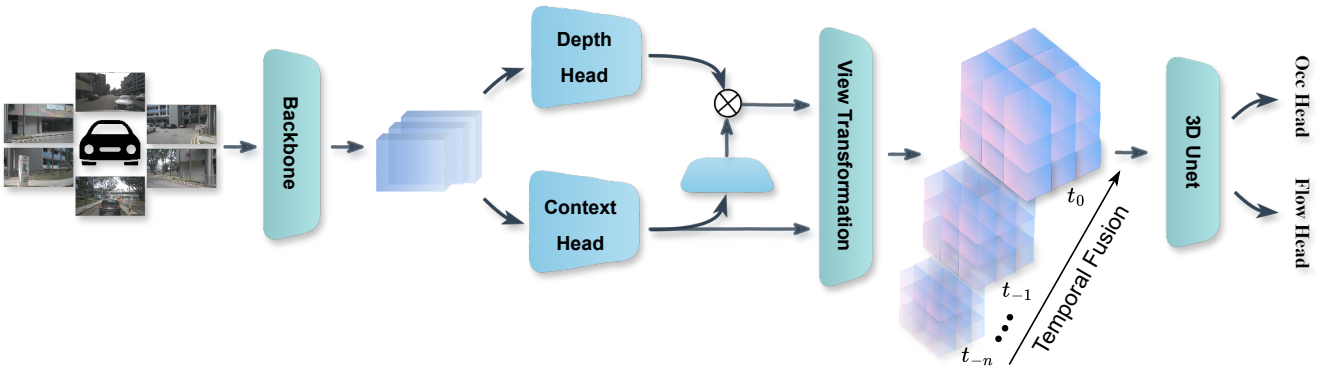


Figure 1. An illustration of our overall pipeline, including image backbone, view transformation from 2D to 3D, Unet for 3D encoding, and task-specific heads.

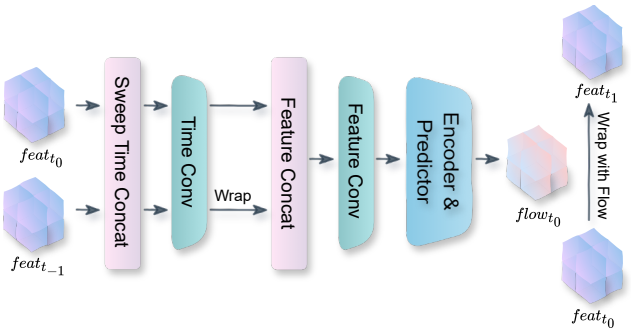


Figure 2. Framework of the flow head. **Note** that the feature of the last frame is sequentially predicted without extra computation. The voxel feature is wrapped to the coordinates of the next frame with the predicted flow for further supervision.

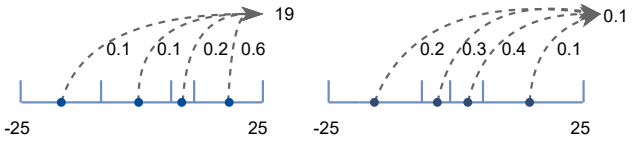


Figure 3. An illustration of aggregating adaptive bins and adaptive weights to flows.

voxel features, further enhancing the adaptability of the forward view transformation. Temporal information from history frames is then fused into these voxel features, based on the sequential temporal fusion method [14]. We utilize a 3D Unet [4] similar to BEVDet [7] for encoding 3D features. Fig. 1 is an illustration of our whole framework.

2.2. Occupancy and Flow Prediction Heads

For the occupancy prediction head, we adopt a per-mask classification approach similar to Mask2Former [3]. The mask for each category is predicted and supervised with binary cross-entropy and dice loss. We use a similar loss function for image semantic segmentation prediction.

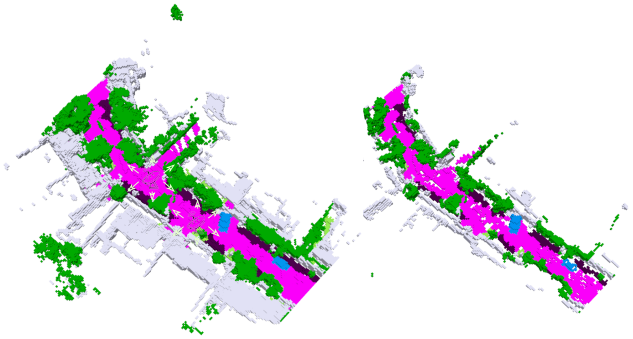
In the flow prediction head, we predict flow using the current and previous frames as input. Rather than computing voxel features for both frames during a single inference, we sequentially utilize the previous frame's features, reducing computational overhead. Considering the significant scale variations of flow in different scenes (e.g., the flow values range from a maximum of 19.11 to a minimum of -22.73 in the OpenOcc dataset), neural networks struggle to adapt to data with such high variance. Predicting flow values for all instances becomes challenging for the network. Therefore, we transform the regression problem into a combination of classification and regression, alleviating the prediction burden on the network. We model the flow predictions within a scene into discrete adaptive bin predictions [1] and adaptive weights prediction. We first average the features in the scene to predict the bin centers. After defining the number of bins n, we predict n probabilities using softmax and calculate each bin's center using cumulative probability:

$$c(b_i) = f_{\text{min}} + (f_{\text{max}} - f_{\text{min}}) \left(\frac{b_i}{2} + \sum_{j=1}^{i-1} b_j \right),$$

where b_i is the bin probability, and f_{min} and f_{max} are the pre-defined flow range. We then predict the probability p_k for each flow bin and compute the final flow prediction by weighting these bin centers:

$$f = \sum_{k=1}^{N} c(b_k) p_k.$$

As shown in Fig. 3, the prediction of different scales of flows can be transferred to different bins and weights prediction. We supervise the flow scale and direction by minimizing the L2 loss and maximizing cosine similarity with



(a) Groundtruth Occupancy (b) Masked Occupancy

Figure 4. Visualization of the ray visible mask. (a): Groundtruth occupancy map; (b): Traffic-critical regions of the groundtruth occupancy map.

the ground truth values, respectively. Using the predicted flow, we transform the current frame's features to the next frame and supervise them with the ground truth occupancy of the future frame. To address the issue of gradient discontinuity when mapping features by coordinates, we again use the trilinear interpolation's inverse process. Supervision is applied using simple cross-entropy. Fig. 2 is an illustration of the flow head.

We observed that jointly predicting occupancy and flow significantly degraded occupancy performance. Thus, we adopted a two-stage training strategy: first, we trained the feature encoding module and occ head jointly with occupancy prediction, then finetuned the flow head with the occupancy backbone frozen.

2.3. Further Improvements

Ray Visible Mask[1]**.** Beyond traffic-related factors on the road, there are numerous unnecessary elements such as buildings and vegetation far away from the ego vehicle. RayIoU [10] increases the focus on important traffic targets by setting a virtual LiDAR along the vehicle's driving path and only assessing the regions scanned by this virtual LiDAR. Therefore, we adopt this idea during training, directing more attention to the critical factors of traffic. We follow the calculation of RayIoU by setting multiple LiDAR origins along the driving path and calculating the visible regions from these origins to obtain a ray visible mask. To differentiate occupied voxels from surrounding points, we also consider voxels within 2 meters of ray-visible occupied points as critical regions. Per-frame training losses are only calculated on the critical regions with the mask. The critical regions are visualized in Fig. 4. During training, we implement a hard example mining strategy, focusing on the training of difficult voxels based on their uncertainty.

1 https://drive.google.com/file/d/10jB08Z6MLT3JxkmQfxgPVNq5Fu4lHs_h/view

Stronger Setting. We upgrade our base model from ResNet-50 [5] to Swin Base [12], increasing voxel feature channel sizes and using larger image input resolution.

3. Experiment

In this section, we discuss the dataset and metrics, provide implementation details, and present the results of ablation studies. Tab. 3 displays the final results on the test server.

3.1. Dataset and Metrics

The 3D Occupancy and Flow Prediction Challenge Dataset [18] is built upon the nuScenes dataset [2]. It comprises 700 sequences for training, 150 for validation, and 150 for testing. Each frame includes six surround-view images with a resolution of 900×1600. Occupancy and flow annotations are provided for each frame within the range of [-40m, -40m, -1m, 40m, 40m, 5.4m], with voxel resolution set at 0.4 meters. The dataset includes flow annotations for the x and y axes and semantic annotations for 17 categories (including "unoccupied"). No external data is utilized in our method. The final evaluation metrics are the class-averaged RayIoU [10] across all classes and the mAVE for foreground classes. The overall evaluation score is computed as

$$\text{Occ Score} = 0.9 \cdot \text{RayIoU} + 0.1 \cdot \max(1 - \text{mAVE}, 0).$$

3.2. Implementation Details

Training Strategies. We conducted preliminary experiments using BEVDet as the baseline model, training on 8 NVIDIA A100 GPUs. We used the AdamW optimizer [13] with a learning rate of 2e-4, a weight decay of 0.05, and a total batch size of 32. We adopted exponential moving average (EMA) [7] for updating model weights. When using ResNet-50 [5] as the backbone, we trained the baseline model without temporal fusion for 24 epochs. For the baseline with temporal fusion, we followed the methodologies of SOLOFusion [14] and FB-BEV to apply CBGS [14] for 12 epochs. For the Swin Base [12] model, we trained the occ head with a total batch size of 8 and trained with CBGS for 5 epochs. The flow head was then fine-tuned for an additional 5 epochs.

Network Details. We used common data augmentation strategies [7], including random flip, scale, and rotation for image and flip augmentation for voxel features. We initialized our networks using publicly available models. For ResNet-50, we used a BEVDet [6] detection pre-trained model, and for the Swin Base, we initialized with GeoMIM [11] pre-trained on Occ3d [16]. The voxel size in training is set to $200 \times 200 \times 16$. When using ResNet-50 as the backbone, the input image size was 256×704, with voxel feature channels set to 32. The image size is set to 640×1600 with

	Method	IoU@1	IoU@2	IoU@4	RayIoU
-	Competition Baseline [9]	0.196	0.248	0.284	0.243
1	Baseline [Bevdet, ResNet-50, 256×704]	0.268	0.325	0.362	0.318
2	Exp#1+Mask-Based Loss	0.296	0.376	0.430	0.367
3	Exp#2+Segmentation Supervision, Depth Semantic Fusion	0.310	0.391	0.444	0.382
4	Exp#3+History Fusion	0.365	0.444	0.490	0.433
5	Exp#3+Ray Visible Mask V1	0.325	0.399	0.447	0.39
6	Exp#3+Ray Visible Mask V2	0.326	0.404	0.453	0.394
7	Exp#6+History Fusion, Swin Base, 640×1600, Channel 100	0.407	0.483	0.523	0.471

Table 1. The occupancy prediction results w.r.t. different settings on the nuScenes-OpenOcc `val` set.

Experiment	OCC Score	RayIoU	mAVE@TP	mAVE@LQ	mAVE@Per-voxel
Swin Base w/o Flow	-	0.471	-	-	-
Swin Base + Flow Joint Train	0.443	0.43.5	0.486	0.955	0.508
Swin Base (fix) + Flow	0.479	0.471	0.467	0.914	0.541
Swin Base (fix) + AdaFlow, Future frame Sup	0.481	0.471	0.457	0.908	0.534

Table 2. The flow prediction results w.r.t. different settings on the nuScenes-OpenOcc `val` set. mAVE@TP is the original flow metric of the challenge. We introduced two additional metrics: per-voxel mAVE (mAVE@Per-Voxel), and mAVE for all points queried from the LiDAR origin (mAVE@LQ).

OCC Score	RayIoU@1	RayIoU@2	RayIoU@4	RayIoU	mAVE
0.453	0.398	0.459	0.496	0.451	0.529

Table 3. Final results on the nuScenes-OpenOcc `test` set.

100 voxel channels for the Swin Base model. We used 16 history frames for temporal fusion, following a sequentially based fusion pipeline [14].

3.3. Ablation Study for Occ and Flow Training

We conduct preliminary experiments using the ResNet-50 backbone to quickly validate the efficacy of the proposed components, then scale up using the validated effective methods. We found that training occ and flow together decreased RayIoU by about two points. Thus, we adopted a two-stage training strategy: first train occ, then train flow. Tab. 1 presents the ablation study results on occ training, we incrementally added modules to the BEVDet baseline, each significantly improving performance. In baseline 2, we adopted the mask-based loss and hard example mining strategy following Mask2Former [3], achieving significant improvements. Similarly, in baseline 5, we used the ray visible mask to focus the network on important traffic areas, improving RayIoU. In baseline 6, performance (particularly IoU@4) is further enhanced by including voxels within 2 meters of the ray termination point in training. Additionally, in baseline 3, image segmentation loss and the injection of semantic information into depth alleviated the unimodal distribution, enhancing adaptability. Finally, our method surpassed the official baseline method with a RayIoU of 0.151 under the same image output size and backbone settings. In baselines 4 and 7, we leveraged historical frame information and larger models, which naturally led to improvements.

Tab. 2 presents the evaluation results of flow head training. When fine-tuning both occ and flow on a trained occ network, we observed a significant drop in occ prediction performance. Therefore, we froze the occ network and the encoder, fine-tuning only the flow head. This preserved the occ performance while effectively predicting flow. To provide a more comprehensive comparison of flow performance, we introduced two additional metrics: one calculates a per-voxel mAVE (mAVE@Per-Voxel), and the other calculates mAVE for all points queried from the LiDAR origin (mAVE@LQ). We found that fine-tuning all parameters resulted in a decrease in the per-voxel mAVE, but did not provide gains for the other two metrics. Additionally, our proposed AdaBin method and the use of future frames for supervision improved flow prediction.

4. Final Results and Conclusion

In this report, we describe our solution in detail for the CVPR 2024 Autonomous Grand Challenge Track On Occupancy and Flow. The model of our final submission employs Swin Base as the image backbone. The image resolution is set to 640×1600. The encoded voxel size is $200 \times 200 \times 16$, with a channel size of 100. We use 16 historical frames. No post-processing techniques, such as test-time augmentation or model ensembling, are applied. This model achieves an Occ Score of 0.453 on the test server, ranking 2nd on the test server.

References

[1] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4009–4018, 2021. 2

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

[3] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 2, 4

[4] Özgün Çiçek, Ahmed Abdulkadir, Soeren S Lienkamp, Thomas Brox, and Olaf Ronneberger. 3d u-net: learning dense volumetric segmentation from sparse annotation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19*, pages 424–432. Springer, 2016. 2

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[6] Junjie Huang and Guan Huang. Bevdet4d: Exploit temporal cues in multi-camera 3d object detection. *arXiv preprint arXiv:2203.17054*, 2022. 3

[7] Junjie Huang, Guan Huang, Zheng Zhu, Yun Ye, and Dalong Du. Bevdet: High-performance multi-camera 3d object detection in bird-eye-view. *arXiv preprint arXiv:2112.11790*, 2021. 2, 3

[8] 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. 1

[9] 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. 4

[10] Haisong Liu, Haiguang Wang, Yang Chen, Zetong Yang, Jia Zeng, Li Chen, and Limin Wang. Fully sparse 3d panoptic occupancy prediction. *arXiv preprint arXiv:2312.17118*, 2023. 1, 3

[11] Jihao Liu, Tai Wang, Boxiao Liu, Qihang Zhang, Yu Liu, and Hongsheng Li. Geomim: Towards better 3d knowledge transfer via masked image modeling for multi-view 3d understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17839–17849, 2023. 3

[12] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 3

[13] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 3

[14] 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. 2, 3, 4

[15] Jonah Philion and Sanja Fidler. Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIV 16*, pages 194–210. Springer, 2020. 1

[16] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yucheng Mao, Huitong Yang, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *Advances in Neural Information Processing Systems*, 36, 2024. 1, 3

[17] Wenwen Tong, Chonghao Sima, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, et al. Scene as occupancy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8406–8415, 2023. 1

[18] Xiaofeng Wang, Zheng Zhu, Wenbo Xu, Yunpeng Zhang, Yi Wei, Xu Chi, Yun Ye, Dalong Du, Jiwen Lu, and Xingang Wang. Openoccupancy: A large scale benchmark for surrounding semantic occupancy perception. *arXiv preprint arXiv:2303.03991*, 2023. 1, 3

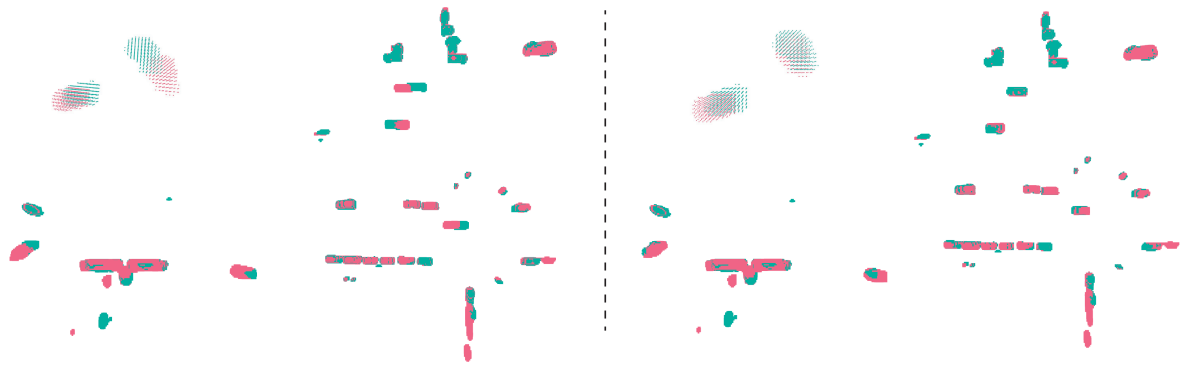
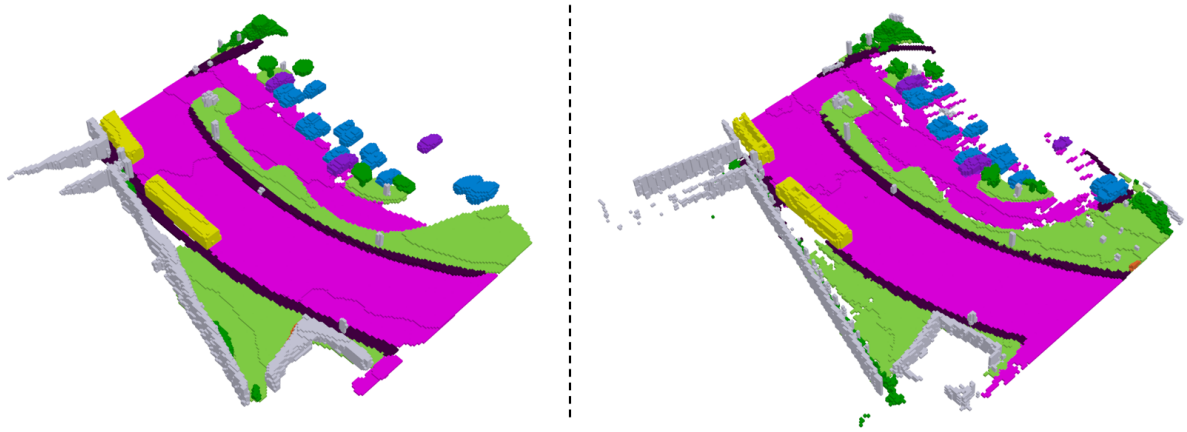


Figure A.5. An illustration of our flow prediction.



(a) Predicted occupancy (b) GT occupancy

Figure A.6. A comprising of the predicted (left) and groundtruth occupancy map (right).

Apendix

A. Visulization

As shown in Fig. A.5 and Fig. A.6, we visualized the comparison between the predicted and ground truth occupancy, as well as the predicted flow and its application in transforming the current frame to the next frame. The visualizations demonstrate the accuracy of our model in predicting occupancy status and semantic class. The alignment between predicted and actual occupancies confirms the effectiveness of our dual-stage framework and AdaBin integration. For flow prediction, the visual transformation shows our model's capability to capture temporal dynamics and spatial continuity, validating the robustness of our approach.