Rethinking Temporal Fusion with a Unified Gradient Descent View for 3D Semantic Occupancy Prediction

Supplementary Material

A. Derivation of the Gradients

A.1. Gradient Computation for Scene-Level Temporal Fusion

Consider a normalized transformation function $f_s : \mathbb{R}^{c\times N} \to \mathbb{R}^{c\times N}$ defined as:

$$f_s(\mathbf{X}) = \gamma \odot \mathrm{Norm}(W\mathbf{X} + \mathbf{b}) + \beta + \mathbf{X}, \tag{A.1}$$

where $\gamma, \beta \in \mathbb{R}^{c\times 1}$ are learnable scale and shift parameters, $W \in \mathbb{R}^{c\times c}$ is a weight matrix, $\mathbf{b} \in \mathbb{R}^{c\times 1}$ is a bias vector, and $\odot$ denotes the Hadamard product. $\mathrm{Norm}(\cdot)$ represents the Z-score normalization function across the channel dimension. We seek to minimize the loss function $\mathcal{L}_s$, defined as:

$$\mathcal{L}_s = \|f_s(\mathbf{Q}_1\mathbf{V}^t) - \mathbf{Q}_2\mathbf{V}^t\|_F^2, \tag{A.2}$$

where $\mathbf{Q}_1, \mathbf{Q}_2 \in \mathbb{R}^{c\times c}$ are given matrices and $\mathbf{V}^t \in \mathbb{R}^{c\times N}$. Here, $\|\cdot\|_F$ denotes the Frobenius norm. To facilitate the gradient computation $\mathbf{H}_s^t = -\eta_s \nabla_{\gamma,\beta,W,\mathbf{b}} \mathcal{L}_s$, we introduce the following intermediary terms:

$$\Delta_1 = f_s(\mathbf{Q}_1\mathbf{V}^t) - \mathbf{Q}_2\mathbf{V}^t, \tag{A.3}$$

$$\mathbf{Z} = W\mathbf{Q}_1\mathbf{V}^t + \mathbf{b}\mathbf{1}_n^\top, \tag{A.4}$$

where $\mathbf{1}_n \in \mathbb{R}^{n\times 1}$ denotes a vector of ones. The loss function can then be expressed as:

$$\mathcal{L}_s = \frac{1}{nc} tr(\Delta_1^\top \Delta_1). \tag{A.5}$$

Let $\hat{\mathbf{Z}} = \mathrm{Norm}(\mathbf{Z})$, we have:

$$\hat{\mathbf{Z}} = (\mathbf{Z} - \boldsymbol{\mu}) \oslash \boldsymbol{\sigma}, \tag{A.6}$$

$$\boldsymbol{\mu} = \frac{1}{n}\mathbf{1}_c^\top \mathbf{Z}, \tag{A.7}$$

$$\boldsymbol{\sigma}^2 = \frac{1}{n}\mathbf{1}_c^\top (\mathbf{Z} - \mathbf{1}_c\boldsymbol{\mu})^2, \tag{A.8}$$

where $\oslash$ denotes Hadamard division. The function f_s can then be written as:

$$f_s(\mathbf{Q}_1\mathbf{V}^t) = \gamma \odot \hat{\mathbf{Z}} + \beta + \mathbf{Q}_1\mathbf{V}^t. \tag{A.9}$$

The gradient of the loss with respect to f_s is:

$$\frac{\partial \mathcal{L}_s}{\partial f_s} = 2\Delta_1. \tag{A.10}$$

The gradients with respect to the learnable parameters γ and β are:

$$\frac{\partial \mathcal{L}_s}{\partial \gamma} = 2(\Delta_1 \odot \hat{\mathbf{Z}})\mathbf{1}_n, \quad \frac{\partial \mathcal{L}_s}{\partial \beta} = 2\Delta_1\mathbf{1}_n. \tag{A.11}$$

For the normalized activations $\hat{\mathbf{Z}}$, we have:

$$\frac{\partial \mathcal{L}_s}{\partial \hat{\mathbf{Z}}} = 2\gamma \odot \Delta_1 = \Delta_2, \tag{A.12}$$

where we define Δ_2 for notational convenience. For each column i, the gradient of the normalized activations with respect to the pre-normalized activations is:

$$\frac{\partial \hat{\mathbf{Z}}_i}{\partial \mathbf{Z}_i} = \frac{1}{\sigma_i}(\mathbf{I} - \frac{1}{n}\mathbf{1}_c\mathbf{1}_c^\top - \frac{1}{n}\hat{\mathbf{Z}}_i\hat{\mathbf{Z}}_i^\top), \tag{A.13}$$

where $\mathbf{I}$ is the identity matrix. The complete gradient with respect to $\mathbf{Z}$ is:

$$\frac{\partial \mathcal{L}_s}{\partial \mathbf{Z}} = \left(\Delta_2 - \frac{1}{n}\mathbf{1}_c\mathbf{1}_c^\top\Delta_2 - \frac{1}{n}\hat{\mathbf{Z}} \odot (\mathbf{1}_c\mathbf{1}_c^\top(\hat{\mathbf{Z}} \odot \Delta_2))\right) \oslash (\mathbf{1}_c\boldsymbol{\sigma}), \tag{A.14}$$

Let $\Delta_3 = \frac{\partial \mathcal{L}_s}{\partial \mathbf{Z}}$, the gradients with respect to W and $\mathbf{b}$ are:

$$\frac{\partial \mathcal{L}_s}{\partial W} = \Delta_3(\mathbf{Q}_1\mathbf{V}^t)^\top, \quad \frac{\partial \mathcal{L}_s}{\partial \mathbf{b}} = \Delta_3\mathbf{1}_n. \tag{A.15}$$

A.2. Derivation of Sampling Function Jacobian

Given the trilinear interpolation sampling function:

$$\text{Sample}\left(\mathbf{H}_m^{t-1};\ \mathbf{R}^{t\to t-1}(\mathbf{P} + \mathbf{M}^t)\right), \tag{A.16}$$

we aim to compute its Jacobian matrix $\mathbf{J}$ with respect to the sampling coordinates. For a coordinate $\hat{p} = (x, y, z)^\top$ in $\mathbf{R}^{t\to t-1}(\mathbf{P} + \mathbf{M}^t)$, the sampling function is defined as:

$$\text{Sample}\left(\mathbf{H}_m^{t-1};\ \hat{p}\right) = \mathbf{w}(\hat{p})^\top\mathbf{H}_m[\hat{p}], \tag{A.17}$$

where $\mathbf{w}(\hat{p}) \in \mathbb{R}^{8\times 1}$ is the weight vector derived using trilinear interpolation basis functions. $\mathbf{H}_m[\hat{\mathbf{p}}]$ represents the feature matrix at the eight corner points surrounding $\hat{p}$:

$$\mathbf{H}_m[\hat{p}] = \begin{bmatrix} \mathbf{H}_m^{t-1}(i_0,\ j_0,\ k_0)^\top \\ \mathbf{H}_m^{t-1}(i_0+1,\ j_0,\ k_0)^\top \\ \mathbf{H}_m^{t-1}(i_0,\ j_0+1,\ k_0)^\top \\ \mathbf{H}_m^{t-1}(i_0+1,\ j_0+1,\ k_0)^\top \\ \mathbf{H}_m^{t-1}(i_0,\ j_0,\ k_0+1)^\top \\ \mathbf{H}_m^{t-1}(i_0+1,\ j_0,\ k_0+1)^\top \\ \mathbf{H}_m^{t-1}(i_0,\ j_0+1,\ k_0+1)^\top \\ \mathbf{H}_m^{t-1}(i_0+1,\ j_0+1,\ k_0+1)^\top \end{bmatrix}, \tag{A.18}$$

where $i_0 = \lfloor x \rfloor$, $j_0 = \lfloor y \rfloor$, and $k_0 = \lfloor z \rfloor$. We define the fractional parts as $dx = x - i_0$, $dy = y - j_0$, and $dz = z - k_0$. The linear basis functions and their derivatives are:

$$\phi_0(e) = 1 - e, \qquad \phi_1(e) = e, \tag{A.19}$$

$$\phi_0'(e) = -1, \qquad \phi_1'(e) = 1. \tag{A.20}$$

For each coordinate direction:

$$\phi_x = \begin{bmatrix}\phi_0(dx)\\ \phi_1(dx)\end{bmatrix}, \qquad \phi_x' = \begin{bmatrix}\phi_0'(dx)\\ \phi_1'(dx)\end{bmatrix}, \tag{A.21}$$

$$\phi_y = \begin{bmatrix}\phi_0(dy)\\ \phi_1(dy)\end{bmatrix}, \qquad \phi_y' = \begin{bmatrix}\phi_0'(dy)\\ \phi_1'(dy)\end{bmatrix}, \tag{A.22}$$

$$\phi_z = \begin{bmatrix}\phi_0(dz)\\ \phi_1(dz)\end{bmatrix}, \qquad \phi_z' = \begin{bmatrix}\phi_0'(dz)\\ \phi_1'(dz)\end{bmatrix}. \tag{A.23}$$

The weight vector $\mathbf{w}(\hat{p})$ is then constructed using the Kronecker product $\otimes$:

$$\mathbf{w}(\hat{p}) = \phi_x \otimes \phi_y \otimes \phi_z. \tag{A.24}$$

The gradient of the weight vector is:

$$\nabla_{\hat{p}}\mathbf{w}(\hat{p}) = \begin{bmatrix}\phi_x' \otimes \phi_y \otimes \phi_z, & \phi_x \otimes \phi_y' \otimes \phi_z & \phi_x \otimes \phi_y \otimes \phi_z'\end{bmatrix}, \tag{A.25}$$

where each column corresponds to the partial derivative with respect to x, y, and z, respectively. The Jacobian matrix $\mathbf{J}$ at position $\hat{p}$ of the sampling function with respect to $\hat{p}$ is computed as:

$$\mathbf{J}[\hat{p}] = \nabla_{\hat{p}}\text{Sample}\left(\mathbf{H}_m^{t-1};\ \hat{p}\right) = \left(\nabla_{\hat{p}}\mathbf{w}(\hat{p})\right)^\top\mathbf{H}_m[\hat{p}]. \tag{A.26}$$

Method	mIoU	$mIoU_D$	IoU	others	barrier	bicycle	bus	car	cons. veh.	motor.	pedes.	tfc. cone	trailer	truck	drv. surf.	other flat	sidewalk	terrain	manmade	vegetation
BEVFormer [6]	39.2	37.2	-	5.0	44.9	26.2	59.7	55.1	27.9	29.1	34.3	29.6	29.1	50.5	44.4	22.4	21.5	19.5	39.3	31.1
OSP [15]	41.2	37.0	-	11.0	49.0	27.7	50.2	56.0	23.0	31.0	30.9	30.3	35.6	41.2	82.1	42.6	51.9	55.1	44.8	38.2
UniOCC [11]	39.7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
SurroundSDF [9]	42.4	36.2	-	13.9	49.7	27.8	44.6	53.0	30.0	29.0	28.3	31.1	35.8	41.2	83.6	44.6	55.3	58.9	49.6	43.8
FlashOCC [18]	32.0	24.7	65.3	6.2	39.6	11.3	36.3	44.0	16.3	14.7	16.9	15.8	28.6	30.9	78.2	37.5	47.4	51.4	36.8	31.4
COTR [10]	44.5	38.6	75.0	13.3	52.1	32.0	46.0	55.6	32.6	32.8	30.4	34.1	37.7	41.8	84.5	46.2	57.6	60.7	52.0	46.3
ViewFormer [5]	41.9	35.0	70.2	12.9	50.1	28.0	44.6	52.9	22.4	29.6	28.0	29.3	35.2	39.4	84.7	49.4	57.4	59.7	47.4	40.6
OPUS [17]	36.2	33.3	54.0	11.9	43.5	25.5	41.0	47.2	23.9	25.9	21.3	29.1	30.1	35.3	73.1	41.1	47.0	45.7	37.4	35.3
BEVDetOcc-SF [4, 13]	41.9	34.4	75.1	12.1	50.0	22.1	43.9	53.9	**29.1**	23.8	25.8	28.5	34.9	41.8	84.3	44.4	57.5	61.0	53.1	46.7
BEVDetOcc-GF	**43.6**	**36.0**	**77.8**	**12.6**	**51.5**	**24.0**	**46.2**	**55.8**	26.8	**26.3**	**27.3**	**30.8**	**37.6**	**43.4**	**84.7**	**46.8**	**58.4**	**62.1**	**56.9**	**50.7**
FB-Occ [7]	39.8	34.2	69.9	13.8	44.5	27.1	46.2	49.7	24.6	27.4	28.5	28.2	33.7	36.5	81.7	44.1	52.6	56.9	42.6	38.1
FB-Occ-GF	**41.7**	**35.8**	**73.2**	**14.1**	**47.6**	**27.5**	**46.8**	**52.0**	**26.8**	**28.1**	**29.8**	**31.5**	**36.1**	**39.3**	**82.5**	**46.2**	**54.2**	**58.5**	**46.3**	**42.3**
ALOcc	45.5	39.3	75.3	15.3	52.5	30.8	47.2	55.9	**32.7**	33.3	32.4	36.2	**38.9**	43.7	**84.9**	48.5	**58.8**	61.9	53.5	47.3
ALOcc-GF	**46.5**	**40.2**	**77.4**	**15.7**	**53.1**	**32.6**	**48.5**	**57.7**	30.6	**34.1**	**33.6**	**38.8**	38.8	**45.2**	84.8	**49.1**	58.7	**62.4**	**55.8**	**49.9**

Table A.1. **Detailed per-class 3D semantic occupancy prediction results.** GDFusion consistently improves IoU for each class.

Method	Backbone	Input Size	RayIoU	$RayIoU_{1m, 2m, 4m}$		
RenderOcc [12]	Swin-Base	512×1408	19.5	13.4	19.6	25.5
SparseOcc [8]	ResNet-50	256×704	36.1	30.2	36.8	41.2
Panoptic-FlashOcc [19]	ResNet-50	256×704	38.5	32.8	39.3	43.4
OPUS [17]	ResNet-50	256×704	41.2	34.7	42.1	46.7
BEVDetOcc-SF [4, 13]	ResNet-50	256×704	35.2	31.2	35.9	38.4
BEVDetOcc-GF	ResNet-50	256×704	**36.6** ↑1.4	**32.6** ↑1.4	**37.3** ↑1.4	**39.9** ↑1.5
FB-Occ [7]	ResNet-50	256×704	39.0	33.0	40.0	44.0
FB-Occ-GF	ResNet-50	256×704	**40.6** ↑1.6	**35.0** ↑2.0	**41.5** ↑1.5	**45.3** ↑1.3
ALOcc [2]	ResNet-50	256×704	43.7	37.8	44.7	48.8
ALOcc-GF	ResNet-50	256×704	**44.1** ↑0.4	**38.2** ↑0.4	**45.0** ↑0.3	**49.2** ↑0.4

Table A.2. **Evaluation of 3D semantic occupancy prediction on the Occ3D dataset without using the camera-visible mask, assessed using RayIoU metrics.** Relative improvements are highlighted with red arrows ↑. The integration of GDFusion demonstrates consistent and substantial performance enhancements across the baseline methods.

B. Additional Resutls

B.1. Detailed Per-Class Semantic Occupancy Prediction

As shown in Tab. A.1, we present the IoU for all categories. GDFusion consistently improves the performance of the three baselines across most categories, demonstrating the broad applicability of our approach. In particular, our method achieves significant improvements in background categories such as *vegetation* and *manmade*, while also providing considerable gains for dynamic object categories like *car* and *pedestrian*.

B.2. Performance Evaluation with RayIoU

Recently, Liu et al. [8] proposed the use of RayIoU to evaluate semantic occupancy prediction, providing an alternative perspective on the evaluation system in Occ3D [16]. The experiments in Tab. A.2 showcase the significant impact of GDFusion on advancing 3D semantic occupancy prediction under training conditions without a camera-visible mask. The results in Tab. A.2 clearly demonstrate the effectiveness of integrating GDFusion, which consistently improves the performance of baseline models across RayIoU metrics, as indicated by the red arrows marking relative improvements. These findings underscore the effectiveness of GDFusion in leveraging valuable information embedded within temporal cues, thereby enhancing both the geometric coherence and semantic precision of reconstructed scenes. As a result, GDFusion enables more reliable and accurate occupancy predictions.

Method	mIoU	$mIoU_D$	IoU
Baseline	38.0	31.0	71.1
Our Vox his	41.8	34.0	76.5
RWKV	38.2	31.3	72.2
RWKV + Our Scene, Motion, Geometry His	41.9	35.0	76.1
xLSTM	39.9	32.7	74.4
xLSTM + Our Scene, Motion, Geometry His	40.9	33.6	76.1
Mamba	41.2	33.9	75.0
Mamba + Our Scene, Motion, Geometry His	42.4	34.6	76.8
RWKV + Our Vox his	41.7	33.5	76.5
RWKV + Our All His	43.3	35.8	77.8
xLSTM + Our Vox his	42.2	34.7	76.4
xLSTM + Our All his	43.3	35.7	77.5
Mamba + Our Vox his	41.9	34.4	76.4
Mamba + Our All his	43.1	35.2	77.8
Our Full	43.3	35.3	77.8

Table A.3. **Extension study on integrating modern RNN methods with our method.**

Position	mIoU	$mIoU_D$	IoU
Before Depth Net	42.1	34.6	76.6
Before Voxel-level His Fusion	42.4	34.8	76.8
After Voxel-level His Fusion	42.5	34.8	77.0
After Volume Encoder	42.1	34.5	76.6
After Voxel-level His Fusion + Before Voxel-level His Fusion	42.4	34.6	77.2
After Voxel-level His Fusion + After Volume Encoder	42.4	34.7	77.1
All	42.0	34.2	76.9

Table A.4. **Ablation study *w.r.t.* the position of scene-level history fusion in the framework.**

B.3. Results on Modern RNNs

Our voxel-level history fusion module can be directly replaced with modern RNN methods such as RWKV [14], xLSTM [1], and Mamba [3]. In Tab. A.3, we evaluate the performance of combining these modern RNN methods with our approach. For RWKV, xLSTM, and Mamba, we used a single-layer model for each respective structure. From the table, we draw several conclusions: First, a standalone modern RNN method cannot outperform our voxel-level history fusion module. Second, combining our proposed auxiliary temporal modules with a modern RNN method yields significant improvements. Third, while integrating modern RNN methods with our voxel-level fusion approach and other temporal fusion modules does result in improvements, it does not outperform the configuration without modern RNNs (*i.e.*, Our Full). We hypothesize that modern RNNs, which are designed for long-sequence context understanding in tasks like natural language processing, do not exhibit clear advantages in the short sequences of the nuScenes dataset when used solely for temporal fusion. Utilizing modern RNNs to integrate both spatial and temporal dimensions could be explored as a direction for future research.

B.4. Experiments on the Position of Scene-Level History Fusion in the Framework

In Tab. A.4, we investigate the impact of the position of scene-level history fusion on network performance. The baseline model employs voxel-level history fusion. Specifically, we consider several positions within the vision-based semantic occupancy network architecture: before the depth network, before voxel-level history fusion, after voxel-level history fusion, and after the volume encoder, as well as multiple positions for scene-level fusion. The results in the table indicate that scene-level fusion before the depth network or after the volume encoder performs worse compared to fusion before or after voxel-level history fusion. The poor performance of fusion before the depth network is attributed to the fusion occurring in the 2D modality, where the difference in data structure limits its effectiveness compared to direct fusion in the 3D modality. The inferior performance of fusion after the volume encoder is primarily because scene-level history fusion acts similarly to domain adaptation, which is beneficial for generating domain-independent features, favoring subsequent network encoding.

η_m	0.001	0.01	0.1
mIoU	42.5	42.5	42.4
mIoU_D	32.4	32.4	32.4
IoU	76.6	76.5	76.4

Table A.5. **Parameter study on η_m.**

Method	η_s	mIoU	mIoU_D	IoU
Scene His	0.1	42.5	34.8	77.0
	1.0	42.4	34.2	77.3
	10	42.3	34.1	77.5
	100	42.1	34.3	77.2
w/o Gradient on γ, β	0.1	42.3	34.4	77.1
	1.0	42.4	34.5	76.9
	10	42.5	34.9	77.0
	100	42.1	34.3	76.6
Linear $\rightarrow$ MLP	0.01	42.5	35.1	76.9
	0.1	42.4	34.9	77.0

Table A.6. **Ablation study *w.r.t.* the structure of the scene-level history fusion module.** The first row indicates the selected structure for scene-level history fusion. The second row shows the case where only linear layer parameters $\mathbf{W}$ and $\mathbf{b}$ are updated during scene-level history fusion. The third row represents extending the linear layer to an MLP by adding additional parameters.

Therefore, fusion before the volume encoder, which is a densely encoding module, proves to be more advantageous. We also experimented with multiple positions for scene-level fusion, but the results showed no significant advantage over a single fusion position. Consequently, we only perform fusion after the voxel-level history fusion module in the final method.

B.5. Module Structure and Impact of Parameters η_s and η_m

In Tab. A.6, we explore different structural choices for scene-level history fusion. The baseline model utilizes voxel-level temporal fusion. Jointly applying history fusion to both linear and LayerNorm parameters improves IoU to some extent, likely due to the influence of LN parameters on background categories, which occupy a large proportion of the scene. In the third row, we replace the linear layer with an MLP, which means using more parameters to store historical scene information. However, experimental results indicate no significant gains, so we ultimately use only the linear and LN layers to store scene information.

In Tab. A.6 and Tab. A.5, we evaluate the impact of the learning rate parameters η_s and η_m in scene-level temporal fusion and temporal motion fusion on model performance. The results demonstrate that our method is relatively insensitive to these hyperparameters.

B.6. Qualitative Analysis

Fig. A.1 presents the qualitative results of our method. Notably, none of the three baselines was able to predict the presence of the car in the image, and even the ground truth lacks annotations for this car. However, after incorporating GDFusion, all three approaches successfully detected the car, demonstrating the robustness and generalizability of our approach, even in scenarios where the ground truth is incomplete.

References

[1] Maximilian Beck, Korbinian Pöppel, Markus Spanring, Andreas Auer, Oleksandra Prudnikova, Michael Kopp, Günter Klambauer, Johannes Brandstetter, and Sepp Hochreiter. xlstm: Extended long short-term memory. *arXiv preprint arXiv:2405.04517*, 2024. 4

[2] Dubing Chen, Jin Fang, Wencheng Han, Xinjing Cheng, Junbo Yin, Chengzhong Xu, Fahad Shahbaz Khan, and Jianbing Shen. Alocc: adaptive lifting-based 3d semantic occupancy and cost volume-based flow prediction. *arXiv preprint arXiv:2411.07725*, 2024. 3

[3] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 4

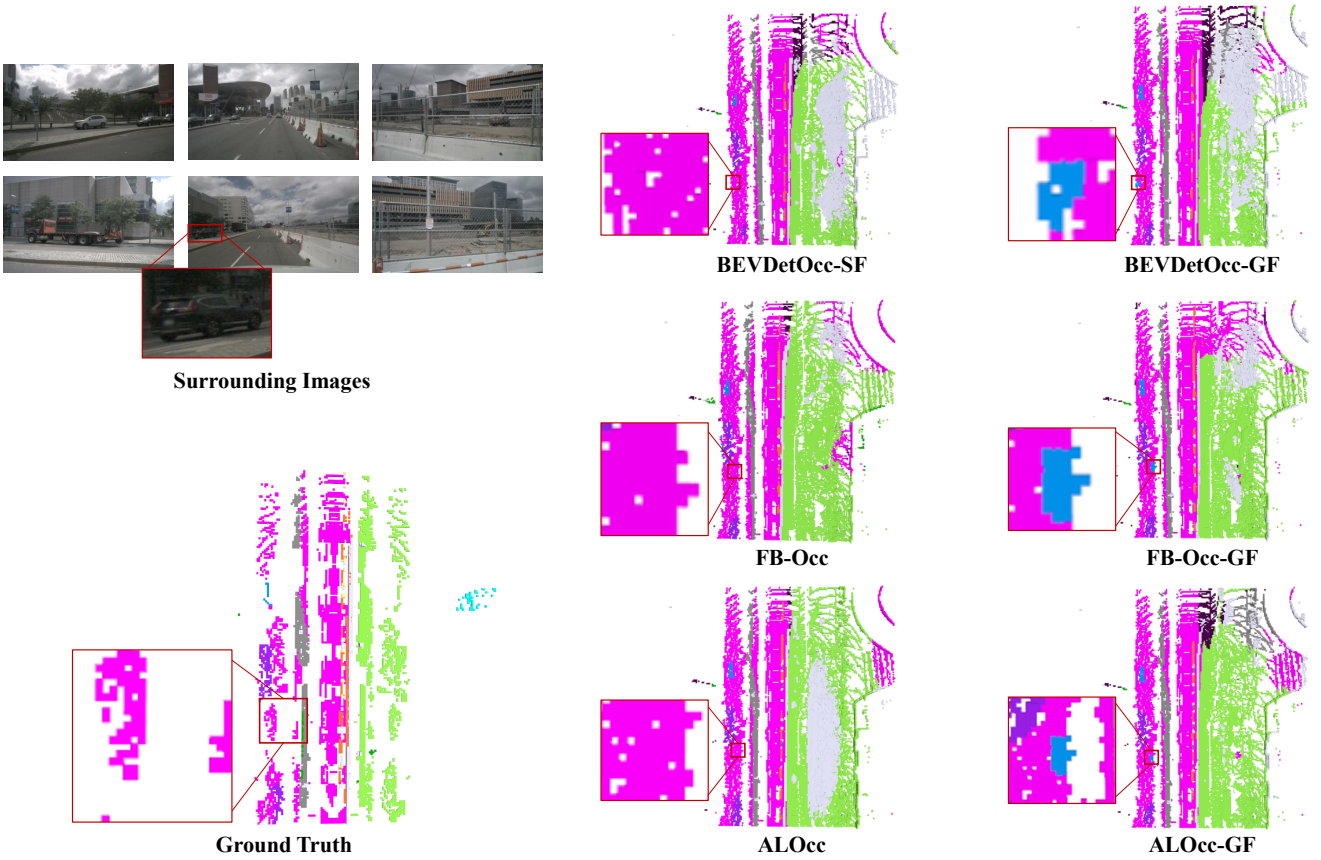

Figure A.1. **Qualitative comparison between BEVDetOcc-SF, FB-Occ, and ALOcc versus their versions enhanced with our GDFusion.** The top row in the leftmost column shows the input images, presented in the following order: camera front left, camera front, camera front right, camera back left, camera back, and camera back right. The bottom row in the leftmost column displays the ground-truth semantic occupancy. The middle section illustrates the results of the three baselines, while the rightmost column presents the results after incorporating our method. Key areas are highlighted with red boxes for emphasis.

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

[5] Jinke Li, Xiao He, Chonghua Zhou, Xiaoqiang Cheng, Yang Wen, and Dan Zhang. Viewformer: Exploring spatiotemporal modeling for multi-view 3d occupancy perception via view-guided transformers. In *Proceedings of European Conference on Computer Vision*, 2024. 3

[6] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Qiao Yu, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. *arXiv preprint arXiv:2203.17270*, 2022. 3

[7] Zhiqi Li, Zhiding Yu, David Austin, Mingsheng Fang, Shiyi Lan, Jan Kautz, and Jose M Alvarez. Fb-occ: 3d occupancy prediction based on forward-backward view transformation. *arXiv preprint arXiv:2307.01492*, 2023. 3

[8] Haisong Liu, Haiguang Wang, Yang Chen, Zetong Yang, Jia Zeng, Li Chen, and Limin Wang. Fully sparse 3d occupancy prediction. In *Proceedings of European Conference on Computer Vision*, 2024. 3

[9] Lizhe Liu, Bohua Wang, Hongwei Xie, Daqi Liu, Li Liu, Zhiqiang Tian, Kuiyuan Yang, and Bing Wang. Surroundsdf: Implicit 3d scene understanding based on signed distance field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 3

[10] Qihang Ma, Xin Tan, Yanyun Qu, Lizhuang Ma, Zhizhong Zhang, and Yuan Xie. Cotr: Compact occupancy transformer for vision-based 3d occupancy prediction. *arXiv preprint arXiv:2312.01919*, 2023. 3

[11] Mingjie Pan, Li Liu, Jiaming Liu, Peixiang Huang, Longlong Wang, Shanghang Zhang, Shaoqing Xu, Zhiyi Lai, and Kuiyuan Yang. Uniocc: Unifying vision-centric 3d occupancy prediction with geometric and semantic rendering. *arXiv preprint arXiv:2306.09117*, 2023. 3

[12] Mingjie Pan, Jiaming Liu, Renrui Zhang, Peixiang Huang, Xiaoqi Li, Hongwei Xie, Bing Wang, Li Liu, and Shanghang Zhang.

Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2024. 3

[13] Jinhyung Park, Chenfeng Xu, Shijia Yang, Kurt Keutzer, Kris Kitani, Masayoshi Tomizuka, and Wei Zhan. Time will tell: New outlooks and a baseline for temporal multi-view 3d object detection. *arXiv preprint arXiv:2210.02443*, 2022. 3

[14] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Stella Biderman, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023. 4

[15] Yiang Shi, Tianheng Cheng, Qian Zhang, Wenyu Liu, and Xinggang Wang. Occupancy as set of points. In *Computer Vision–ECCV 2024: 18th European Conference*, 2024. 3

[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. In *Proceedings of Advances in Neural Information Processing Systems*, 2024. 3

[17] Jiabao Wang, Zhaojiang Liu, Qiang Meng, Liujiang Yan, Ke Wang, Jie Yang, Wei Liu, Qibin Hou, and Mingming Cheng. Opus: occupancy prediction using a sparse set. *arXiv preprint arXiv:2409.09350*, 2024. 3

[18] Zichen Yu, Changyong Shu, Jiajun Deng, Kangjie Lu, Zongdai Liu, Jiangyong Yu, Dawei Yang, Hui Li, and Yan Chen. Flashocc: Fast and memory-efficient occupancy prediction via channel-to-height plugin. *arXiv preprint arXiv:2311.12058*, 2023. 3

[19] Zichen Yu, Changyong Shu, Qianpu Sun, Junjie Linghu, Xiaobao Wei, Jiangyong Yu, Zongdai Liu, Dawei Yang, Hui Li, and Yan Chen. Panoptic-flashocc: An efficient baseline to marry semantic occupancy with panoptic via instance center. *arXiv preprint arXiv:2406.10527*, 2024. 3