OPUS: Occupancy Prediction Using a Sparse Set

Jiabao Wang, Zhaojiang Liu, Qiang Meng, Liujiang, Yan, Ke Wang,

Jie Yang, Wei Liu, Qibin Hou, Mingming Cheng

VCIP, College of Computer Science, Nankai University

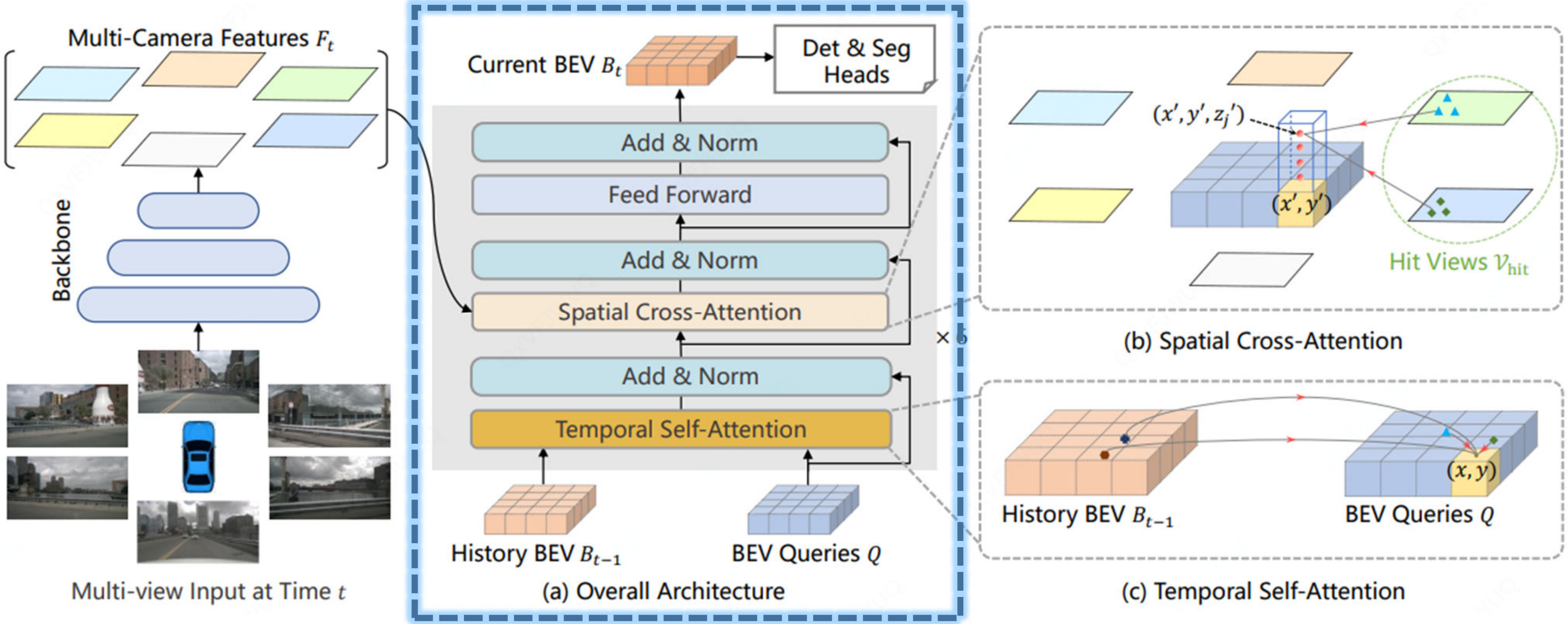
Shanghai Jiaotong University, KargoBot

Contents

Introduction

☐ Dense Occupancy Prediction

Construct dense 3D voxel features by backward projection

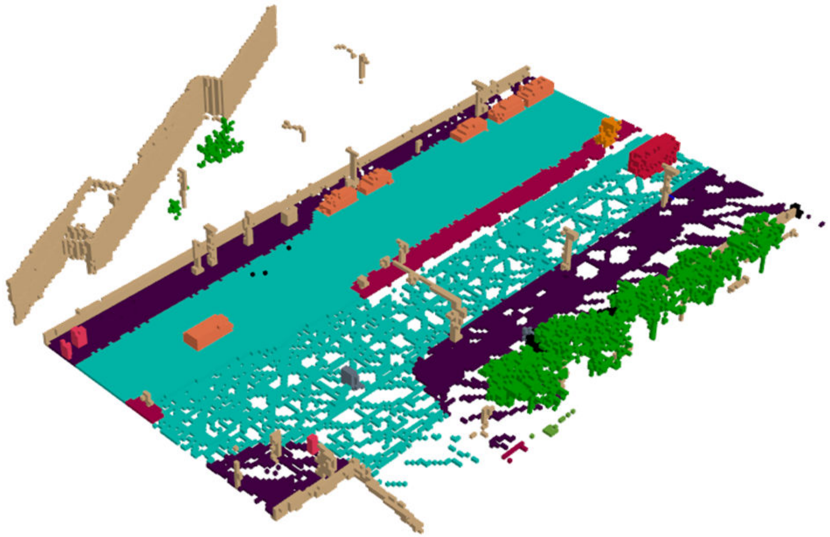


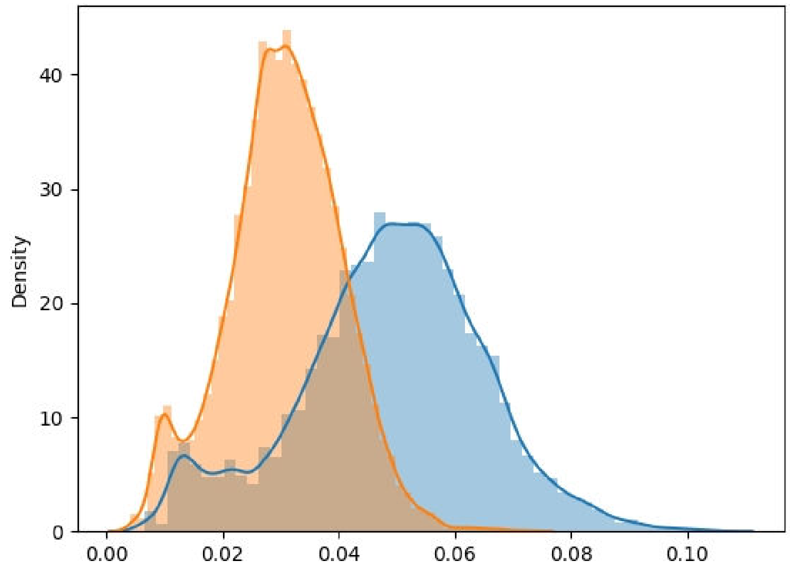
BEVFormer: Learning Bird's-Eye-View Representation from Multi-Camera Images via Spatiotemporal Transformers (ECCV 2022)

Introduction

☐ Occupancy Sparsity

Occupancy in Occ3d-nuSc

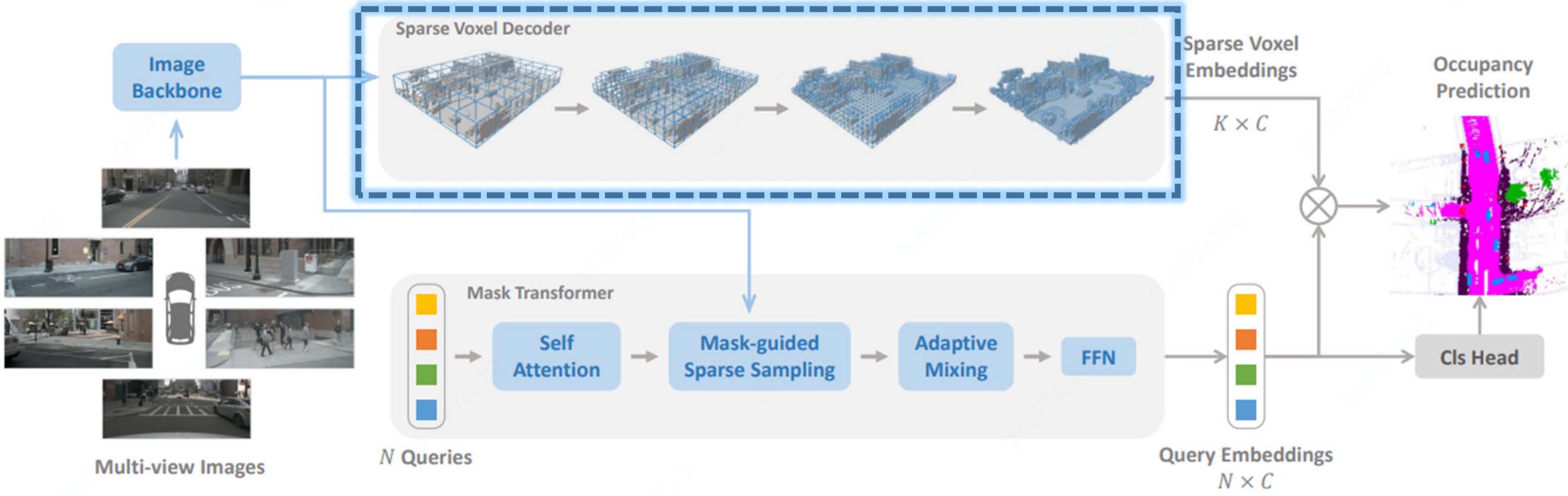




Introduction

◻ Sparse Occupancy Prediction

Filter out empty voxels according to previous decoder classification



SparseOcc: Fully Sparse 3D Occupancy Prediction (Arxiv)

Complicated manual 3D space model !!!

Contents

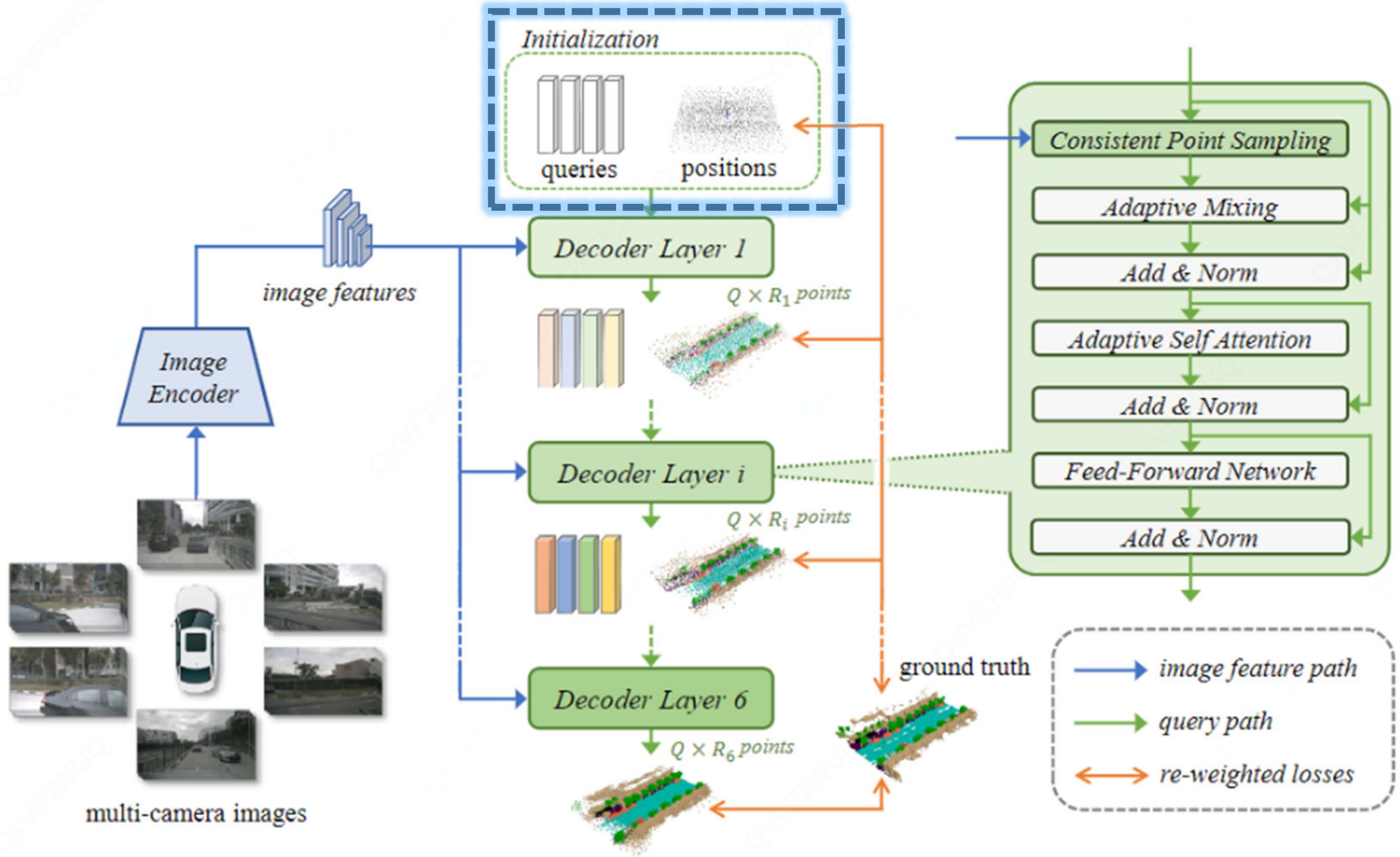
Methods

- Overall Architecture

1. Setup queries

$\mathbb{Q}_0$: $Q \times C$

$\mathbb{P}_0$: $Q \times R_0 \times 3$



Methods

Overall Architecture

2. Update queries in each decoder $\mathbb{Q}_{i-1} \rightarrow \mathbb{Q}_i$

- Consistent Point Sampling

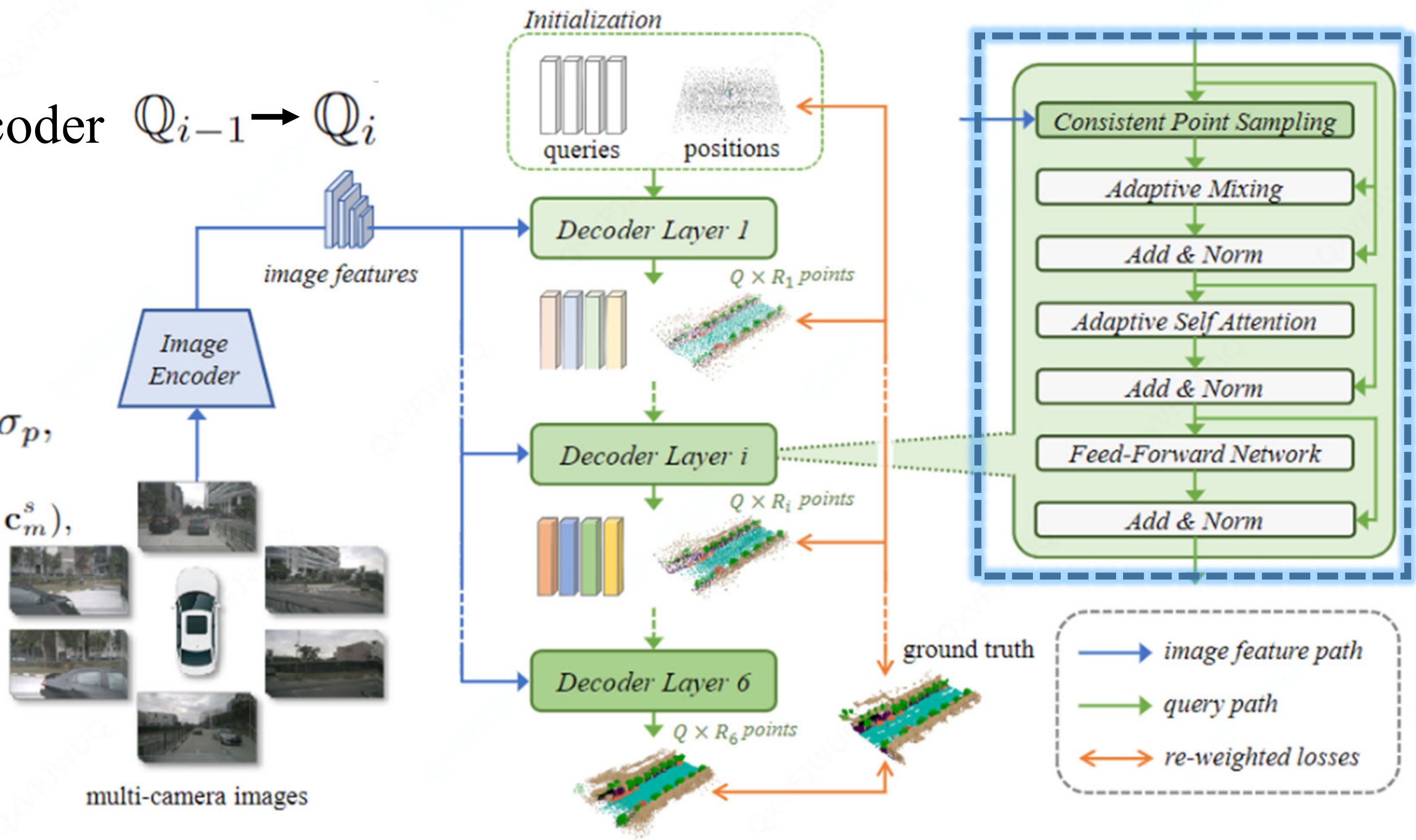
$\mathbb{P}_{i\text{-}1} : Q \times R_{i\text{-}1} \times 3$

$$\mathbf{c}_m = \mathbf{T}_m \mathbf{r}, \text{ where } \mathbf{r} = \mathbf{m}_p + \phi(\mathbf{q}) \cdot \sigma_p,$$

$$f^s = \frac{1}{\sum_{m=1}^{M} |\mathbb{V}_m|} \sum_{s=1}^{S} \sum_{m=1}^{M} w_{s,m} \cdot v_m^s \cdot \mathcal{B}(F_m, \mathbf{c}_m^s),$$

- Adaptive Mixing

- Adaptive Self Attention



Methods

Overall Architecture

3. Update Occupancy predictions

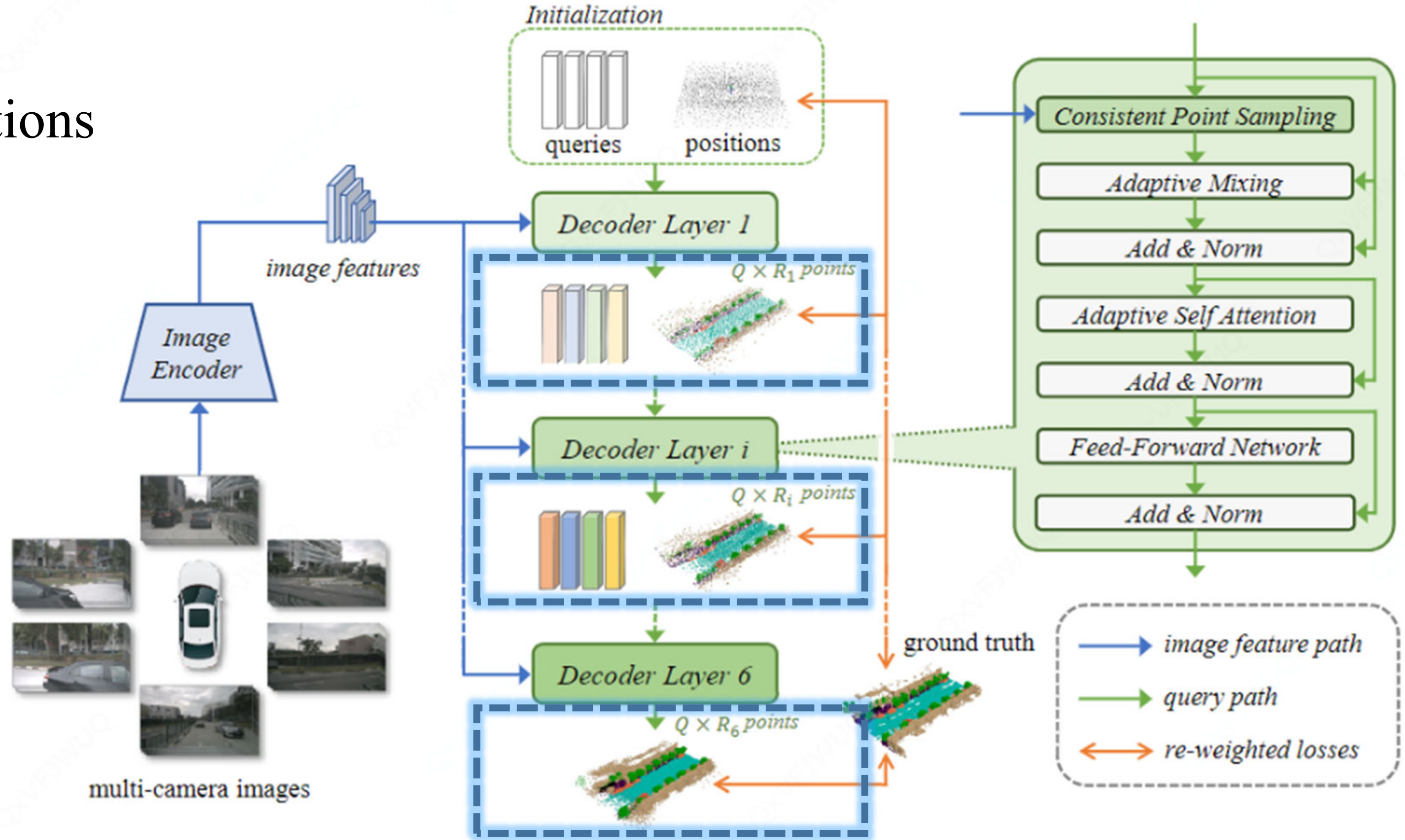
$\mathbb{P}_{i\text{-}1}$: Q×R_{i-1}×3

$\mathbb{P}_i$: Q×R_i×3

$\mathbb{C}_i$: Q×R_i×N

Apply coarse-to-fine strategy, where $R_{i-1} < R_i$

$$\mathbf{p}_i = \bar{\mathbf{p}}_{i-1} + \Delta \mathbf{p}_i$$



Methods

☐ Training Strategy

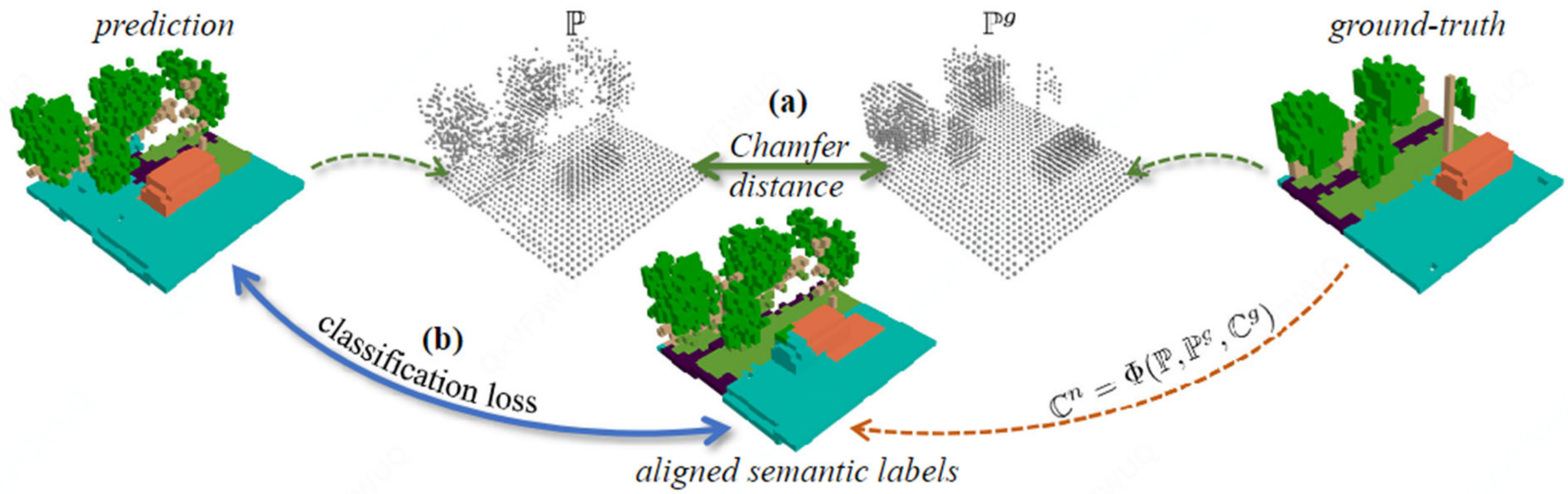
Having a *O(n3)* time complexity and a *O(n2)* space complexity, the Hungarian algorithm is unable to tackle tremendous voxels.

Table 4: Comparison of Hungarian algorithm and our label assignment scheme.

Number of Points	Time (ms) Hungarian Algorithm	Time (ms) Ours	GPU (Mb) Hungarian Algorithm	GPU (Mb) Ours
100	0.52	0.12	39	14
1,000	78.34	0.13	81	14
10,000	24,216.35	1.25	2,304	15
100,000	-	28.85	-	39

Methods

Training Strategy

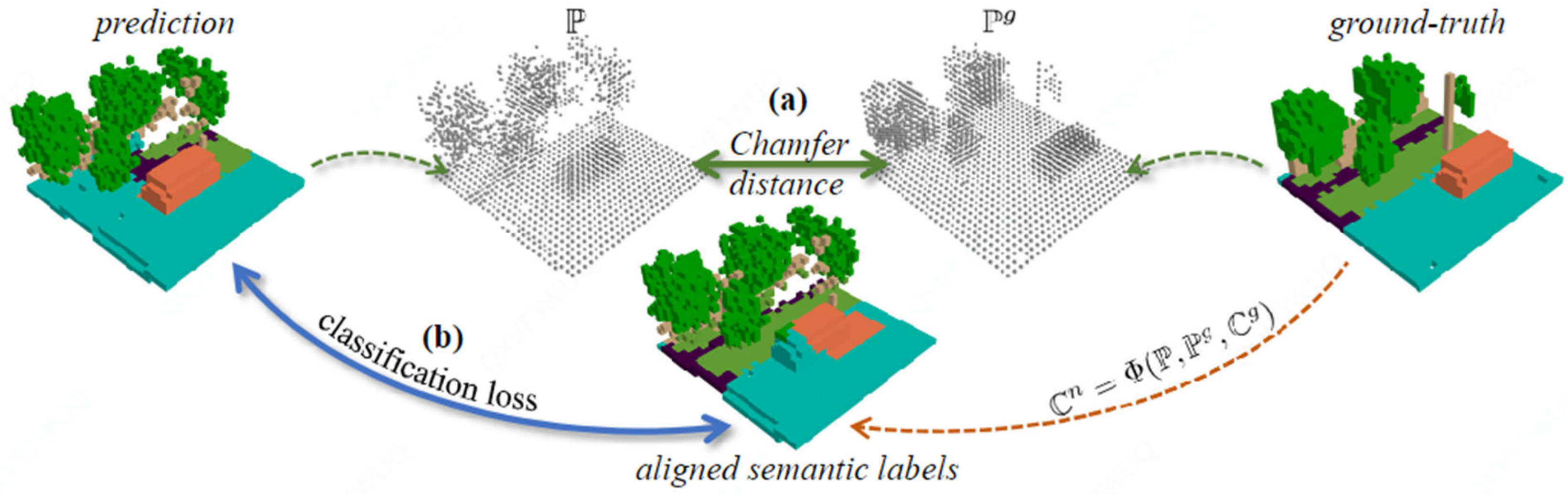


(a) $$\mathrm{CD}(\mathbb{P}, \mathbb{P}^g) = \frac{1}{|\mathbb{P}|}\sum_{\mathbf{p}\in\mathbb{P}} D(\mathbf{p}, \mathbb{P}^g) + \frac{1}{|\mathbb{P}^g|}\sum_{\mathbf{p}^g\in\mathbb{P}^g} D(\mathbf{p}^g, \mathbb{P}), \text{ where } D(\mathbf{x}, \mathbb{Y}) = \min_{\mathbf{y}\in\mathbb{Y}} ||\mathbf{x}-\mathbf{y}||_1.$$

(b) $$\{\mathbb{C}^n, \mathbb{P}^n\} = \left\{\arg\min_{\{\mathbf{c}^g, \mathbf{p}^g\}\in\{\mathbb{C}^g, \mathbb{P}^g\}} ||\mathbf{p}^g - \mathbf{p}||_2, \quad \mathbf{p}\in\mathbb{P}\right\}.$$

Methods

Training Strategy



$$L_{\mathrm{OPS}} = \mathrm{CD}_R(\mathbb{P}_0, \mathbb{P}^g) + \sum_{i=1}^{6} (\mathrm{CD}_R(\mathbb{P}_i, \mathbb{P}^g) + \mathrm{FocalLoss}_R(\mathbb{C}_i, \mathbb{C}_i^n)),$$

Contents

Experiments

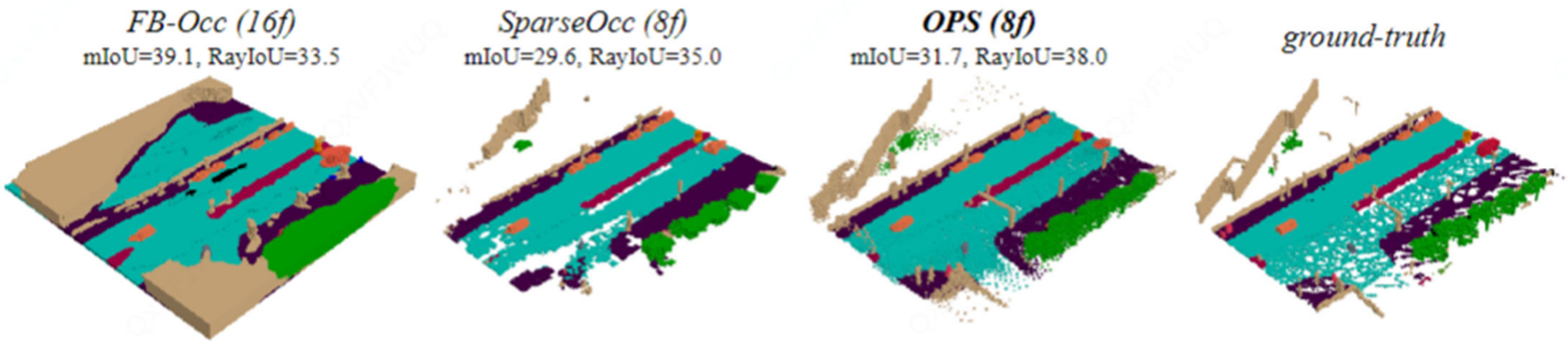
- Compare with SOTA

Table 1: Occupancy prediction performance on Occ3D-nuScenes [31]. "8f" and "16f" denote models fusing temporal information from 8 or 16 frames, respectively. Baseline results are directly copied from their corresponding papers or the SparseOcc [19]. FPS results are measured on an A100 GPU.

Methods	Backbone	Image Size	mIoU	RayIoU_{1m}	RayIoU_{2m}	RayIoU_{4m}	RayIoU	FPS
RenderOcc [28]	Swin-B	1408×512	24.5	13.4	19.6	25.5	19.5	-
BEVFormer [13]	R101	1600×900	39.3	26.1	32.9	38.0	32.4	3.0
BEVDet-Occ [7]	R50	704×256	36.1	23.6	30.0	35.1	29.6	2.6
BEVDet-Occ (8f) [7]	R50	704×384	39.3	26.6	33.1	38.2	32.6	0.8
FB-Occ (16f) [7]	R50	704×256	39.1	26.7	34.1	39.7	33.5	10.3
SparseOcc (8f) [19]	R50	704×256	-	28.0	34.7	39.4	34.0	17.3
SparseOcc (16f) [19]	R50	704×256	30.6	29.1	35.8	40.3	35.1	12.5
OPS-tiny (8f)	R50	704×256	30.6	29.6	36.7	41.4	35.9	24.9
OPS-S (8f)	R50	704×256	31.2	31.0	38.1	42.8	37.3	23.7
OPS-M (8f)	R50	704×256	31.7	31.7	38.8	43.4	38.0	14.7
OPS-L (8f)	R50	704×256	32.4	32.7	39.7	44.3	38.9	7.5
OPS-L (16f)	R50	704×256	33.1	33.7	40.9	45.5	40.0	5.6

Experiments

Compare with SOTA



mIoU cannot reflect the real quality of occupancy prediction!

Paper

Codes

Thanks !