OTOcc: Optimal Transport for Occupancy Prediction

Pengteng Li[1,2], **Ying He**[1*], **F. Richard Yu**[1,2], **Pinhao Song**[3],
Xingchen Zhou[1,2,4] and **Guang Zhou**[4]
[1]College of Computer Science and Software Engineering, Shenzhen University, China
[2]Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ)
[3]KU Leuven
[4]Deeproute Inc.
lipengteng2021@email.szu.edu.cn, {heying, yufei}@szu.edu.cn, pinhao.song@kuleuven.be,
{xingchenzhou, maxwell}@deeproute.ai

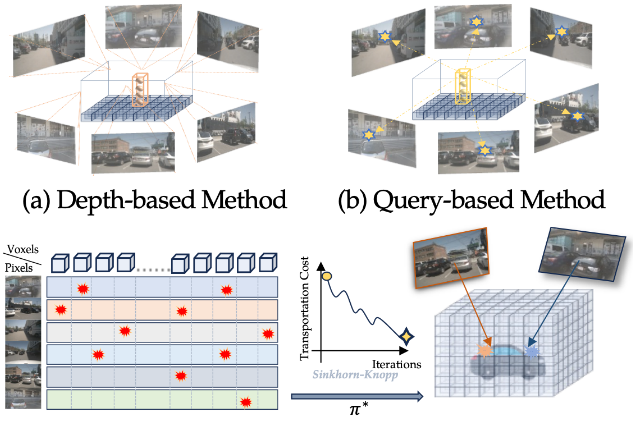
Abstract

The autonomous driving community is highly interested in 3D occupancy prediction due to its outstanding geometric perception and object recognition capabilities. However, previous methods are limited to existing semantic conversion mechanisms for solving sparse ground truths problem, causing excessive computational demands and suboptimal voxels representation. To tackle the above limitations, we propose ***OTOcc***, a novel 3D occupancy prediction framework that models semantic conversion from 2D pixels to 3D voxels as Optimal Transport (OT) problem, offering accurate semantic mapping to adapt to sparse scenarios without attention or depth estimation. Specifically, the unit transportation cost between each demander (voxel) and supplier (pixel) pair is defined as the weighted occupancy prediction loss. Then, we utilize the Sinkhorn-Knopp Iteration to find the best mapping matrices with minimal transportation costs. To reduce the computational cost, we propose a block reading technique with multi-perspective feature representation, which also brings fine-grained scene understanding. Extensive experiments show that ***OTOcc*** not only has the competitive prediction performance but also has about more than 4.58% reduction in computational overhead compared to state-of-the-art methods.

1 Introduction

Vision-based 3D Occupancy Prediction aims to estimate the occupancy state of 3D voxels surrounding the ego-vehicle which provides a comprehensive 3D scene understanding. It is particularly an essential solution for recognizing irregularly shaped objects and also enables the open-set understanding [Tan *et al.*, 2023], further benefiting downstream tasks, like prediction and planning.

Though success, most works still struggle to overcome the semantic sparsity of ground truths, which is the fundamental problem. The information density of such uncompressed representation is low, with numerous regions corresponding to free space in the physical world, resulting in biased 3D presentation construction. Most works are limited to two kinds of semantic conversion mechanisms for alleviating the above-mentioned problem, which can be divided into the following two categories [Qi *et al.*, 2023] : depth-based and query-based methods as shown in Figure 1(a, b). Depth-based methods [Li *et al.*, 2023c; Miao *et al.*, 2023a] leverage the depth estimation to build the multi-view frustums to form the BEV or 3D voxels. Query-based methods [Huang *et al.*, 2023] utilize deformable attention for collecting semantics from images. However, they still suffer from the high computational cost of utilizing heavy decoders or relying on the depth estimation quality due to the sparse scene semantics, which rarely made any fundamental changes to the semantic conversion method. It motivates us to solve its efficiency problems by building the new semantic conversion mechanism without any attention or depth estimation.



(a) Depth-based Method (b) Query-based Method

(c) Our OT-based Method

Figure 1: Comparison of different semantic conversion mechanisms for 3D occupancy prediction. (a) Depth-based methods estimate a latent depth distribution for building 3D representations. (b) Query-based methods pre-define the query (BEV or 3D voxels) to encode semantics from multi-images. (c) Our OT-based method models the mapping strategy from 2D to 3D as the Optimal Transport problem.

[*] Corresponding Author.

In contrast to previous attempts, we address this issue with the Optimal Transport (OT), a well-studied topic in Optimization Theory that directly collects semantics from images by constructing and optimizing the mapping matrix from 2D to 3D via Sinkhorn-Knopp Iteration as shown in Figure 1(c). This method of semantic collection using matrix transformation reduces computational consumption by avoiding repeated learning of partial semantics like attention mechanism and can produce accurate and effective semantic mapping to avoid heavy decoder design. Completely different from current methods, our methods do not use any attention mechanism [Jia *et al.*, 2023], temporal fusion [Yang *et al.*, 2023] or depth estimation [Li *et al.*, 2023a] at all. Specifically, we define each pixel as a supplier that supplies a certain number of semantic marks and define each grid as a demander who needs one unit semantic mark. if a grid receives a sufficient amount of semantic marks from a certain pixel, this grid becomes one important position for that pixel in which the complete semantics will be captured. In this context, the number of semantic marks each pixel supplies can be interpreted as "how many grids that pixels need for better convergence during the training process". The unit transportation cost between each pixel-grid pair is defined as the weighted classification loss of the corresponding grid, where the weight depends on the proportion of semantic marks contribution of each pixel to that grid. After formulation, finding the best semantic conversion strategy is converted to solve the optimal transport plan, which can be quickly and efficiently solved by the off-the-shelf Sinkhorn-Knopp Iteration [Cuturi, 2013].

Our complete framework is as shown in Figure 2, which is named ***OTOcc*** (***O***ptimal ***T***ransport for ***Occ***upancy Prediction). First, we pre-define a group of 3D voxels representation $\mathbf{V}$ containing dense reference point mapping from 3D voxels to 2D images. Motivated by low memory cost design [Huang *et al.*, 2023], we also pre-define tri-perspective view (TPV) mapping matrices for converting multi-view images into TPV features. After applying the camera principle, we utilize the existing reference point mapping relationship to complete various coarse mapping matrices from pixels to grids by inverse projection [Zhou *et al.*, 2022]. Then, coarse mapping matrices are refined via Sinkhorn-Knopp Iteration to seek the best semantic conversion strategy π^*. Multiplying π^* with the image features, we obtain diverse refined TPV features, which can accurately represent current scene information. For further reducing computational cost, we propose a practical block reading technique, which divides the $\mathbf{V}$ equally into four small sets of voxels for optimization in turn. It not only brings Multi-Perspective View (MPV) characteristics to obtain more accurate estimation of 3D scenes but also mitigates the semantic loss caused by tri-plane compression of complex scene information. Finally, we use the lightweight Adaptive Fusion Decoder to process MPV features for obtaining the 3D voxel representation and accurate occupancy prediction. Extensive experiments against state-of-the-art occupancy prediction methods demonstrate that OTOcc outperforms others on two benchmarks, reducing about 4.58% computational cost. Ablation studies further validate the effectiveness of each module within our method. To summarise, our main contributions are as follows:

- We introduce an innovative vision-based 3D occupancy prediction framework named ***OTOcc***. This work represents the first attempt at formulating the semantic conversion from pixels to voxels as the Optimal Transport problem to the best of our knowledge.
- OTOcc designs the dense reference points from voxels to pixels and projects them into multiple mapping matrices inversely. Then, OTOcc utilizes the Sinkhorn-Knopp Iteration to find the optimal mapping matrices. Note that OTOcc doesn't use any attention mechanisms, temporal fusion or depth estimation to show its effectiveness.
- Comprehensive experiments verify that the proposed OTOcc achieves the competitive performance on the Occ3D-nuScene benchmark and also show its effectiveness for vision-based semantic segmentation task.

2 Related Work

2.1 3D Occupancy Prediction

The 3D occupancy prediction task has garnered significant attention due to its enhanced geometry information and superior capabilities in generalized object recognition compared to 3D object detection. Previous works [Yang *et al.*, 2023; Li *et al.*, 2023d] directly utilize the BEV feature for occupancy prediction tasks. Current research in occupancy prediction focuses on dense voxel modelling. OccFormer [Zhang and others, 2023] decomposes the 3D processing into the local and global transformer pathways along the horizontal plane. Due to the high resolution of standard voxel representation and sparse context distribution in 3D scenes, these methods [Zhang and others, 2023; Miao *et al.*, 2023b] face significant computational overhead and efficiency challenges. Some approaches [Huang *et al.*, 2023; Li *et al.*, 2023a] propose to reduce the number of modelled voxels to address this problem. TPVFormer [Huang *et al.*, 2023] proposes a tri-perspective view method for predicting 3D occupancy, leading to performance loss caused by coarse scene semantic compression. VoxFormer [Li *et al.*, 2023a] mitigates computational complexity through depth estimation and modelling only those specific areas. However, the success of this procedure critically depends on the precision of depth estimation. PanoOcc [Wang *et al.*, 2023] presents a coarse-to-fine manner for constructing 3D voxel representation. However, the lack of information from coarse-grained modelling cannot be adequately addressed by the up-sampling process. Some works [Lu *et al.*, 2023; Ouyang *et al.*, 2024] utilize the octree representation for fine-tuning the 3D voxels. Different from above methods, our approach models the semantics conversion from 2D images to 3D representation as the Optimal Transport problem, which finds the best mapping strategy via Sinkhorn-Knopp Iteration. It leverages direct mathematical mapping to avoid redundant and complex semantic collection mechanisms like attention or depth estimation, thereby better adapting to sparse semantic scenes and reducing computational consumption.

2.2 Optimal Transport

Optimal Transport (OT), a well-studied topic in Optimization Theory, efficiently quantifies the minimum "cost" of trans-

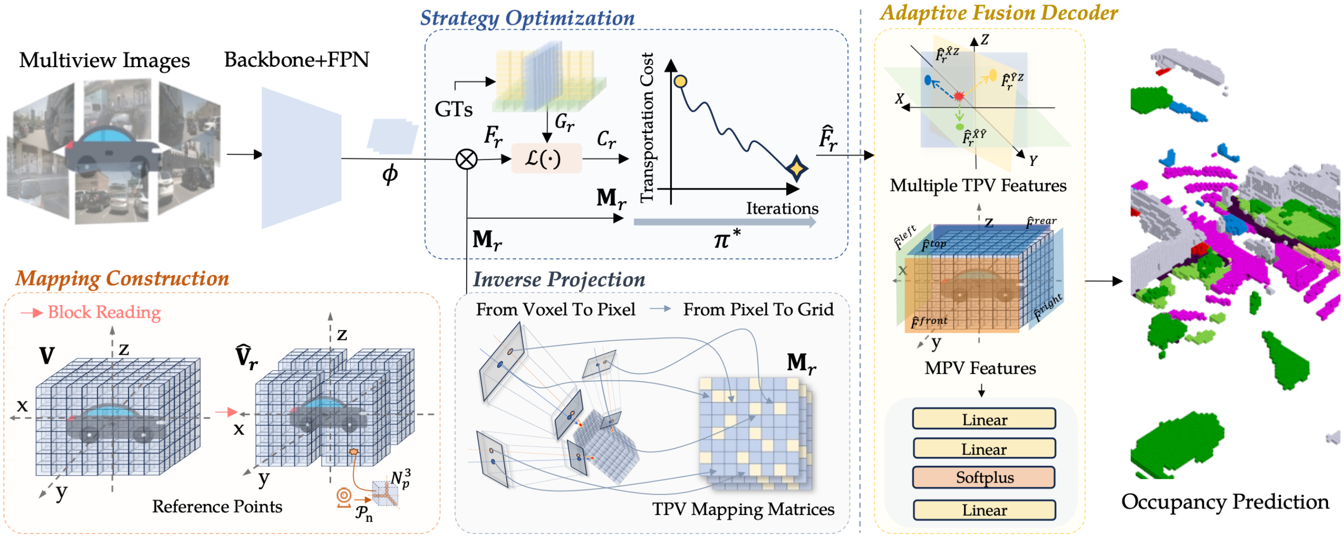


Figure 2: Overall of the proposed OTOcc training framework. "TPV" stands for tri-perspective view (front, side and top). "MPV" stands for multi-perspective view (front, left, right, rear and top). Our method first generates coarse TPV mapping matrices $\mathbf{M}_r$ from pixels to grids for r-th reference points $\hat{\mathbf{V}}_r$. Then, we refine $\mathbf{M}_r$ with Sinkhorn-Knopp Iteration to find the optimal semantic conversion strategy. Finally, the Adaptive Fusion Decoder is designed for constructing MPV features and building accurate 3D representation through simple MLP layer.

porting mass from one distribution to another. It enables measuring the similarity between two distributions, widely applied in various fields, such as object detection [Ge *et al.*, 2021] or domain adaptive [Li *et al.*, 2022a]. Motivated by this well-known theory, we propose a novel framework for 3D occupancy prediction by utilizing OT, which tackles the sub-optimal transformation problem.

3 Method

In this section, we first revisit the definition of the Optimal Transport problem and then demonstrate how we formulate the semantic conversion from 2D images to 3D representation in occupancy prediction into an OT problem as shown in Figure 2. Then, motivated by TPVFormer and MatrixVT [Zhou *et al.*, 2022], we convert the image semantics into TPV features by inverse projection, which utilizes the Sinkhorn-Knopp Iteration to find the best semantic mapping strategy. For reducing the memory cost, we propose the block reading technique. It brings the more accurate estimation of 3D scenes and mitigates the semantic loss caused by tri-plane compression of complex scene information. Finally, we introduce the Adaptive Fusion Decoder for obtaining an accurate 3D voxel representation, which is lightweight and simple, showing the superiority of our OT-based design.

3.1 Optimal Transport

Optimal Transport (OT) describes the following problem: supposing there are n_s suppliers and n_d demanders in a certain area. The α-th supplier holds s_α units of goods while the β-th demander needs d_β units of goods. Transporting cost for each unit of good from supplier α to demander β is denoted by $c_{\alpha\beta}$. The goal of OT problem is to find a transportation plan $\pi^* = \{\pi_{\alpha\beta} \mid \alpha = 1, 2, \dots n_s, \beta = 1, 2, \dots n_d\}$, according to which all goods from suppliers can be transported to demanders at a minimal transportation cost:

$$\begin{aligned} \min_\pi \quad & \sum_{\alpha=1}^{n_s}\sum_{\beta}^{n_d} c_{\alpha\beta}\pi_{\alpha\beta}.\\ \text{s.t.}\quad & \sum_{\alpha=1}^{n_s}\pi_{\alpha\beta}=d_\beta,\quad \sum_{\beta=1}^{n_d}\pi_{\alpha\beta}=s_\alpha\\ & \sum_{\alpha=1}^{n_s}s_\alpha=\sum_{\beta=1}^{n_d}d_\beta,\\ & \pi_{\alpha\beta}\ge 0,\quad \alpha=1,2,\dots n_s, \beta=1,2,\dots n_d.\end{aligned} \tag{1}$$

This is a linear program which can be solved in polynomial time. However, the resulting linear program is large, involving the square of feature dimensions with the semantic conversion from 2D images to 3D representation. Hence, we address this issue by a fast iterative solution named Sinkhorn-Knopp [Cuturi, 2013].

3.2 OT for Semantic Conversion

As shown in Figure 2, we present a coarse-to-fine manner for finding the semantic conversion strategy. We define the semantic conversion from 2D images to 3D voxels as the OT problem. Specifically, We instantiate the semantic conversion strategy as a mapping matrix, which defines each pixel as a supplier who supplies a certain number of semantic marks, and defines each voxel as a demander who needs one unit semantic mark. The pair-wise transportation cost is defined as the weighted occupancy prediction loss. With Sinkhorn-Knopp optimization, we can find the optimal mapping matrix. In this section, we will introduce how to generate the mapping matrix, calculate the transportation cost and optimize the semantic conversion strategy via Sinkhorn-Knopp Iteration.

Mapping Construction. We first define a group of 3D voxels representation $\mathbf{V} \in \mathbb{R}^{X\times Y\times Z}$, where X, Y, Z are the spatial resolution of the voxel space. For reducing the computional cost, we divide the $\mathbf{V}_r$ equally into four small $\hat{\mathbf{V}} \in \mathbb{R}^{\hat{X}\times\hat{Y}\times Z}$, where $\hat{X} = \frac{X}{2}, \hat{Y} = \frac{Y}{2}, r \in \{1, 2, 3, 4\}$. Each grid cell in the voxel corresponds to a box in the real-world size of (e_x, e_y, e_z) meters. We uniformly sample N_p points within a certain range along a dimension of the real world. The real position of a reference point located at voxel grid (i, j, k) in the ego-vehicle frame is (x_i^m, y_j^m, z_k^m), where $i, j, k \in \{1, 2, \cdots, N_p\}$, $m \in \{1, 2, \cdots, N_p^3\}$. The projection between m-th projected reference point $\mathbf{Ref}_{i,j,k}^m = (x_i^m, y_j^m, z_k^m)$ and its corresponding 2D reference point n-th view can be formulate as:

$$d_{ijk}^{n,m} \cdot [u_{ijk}^{n,m}, v_{ijk}^{n,m}, 1] = \mathbf{P_n} \cdot [x_i^m, y_j^m, z_k^m, 1]^T, \tag{2}$$

where $\mathbf{P}_n \in \mathbb{R}^{3\times4}$ is the projection matrix of the n-th camera. $n \in \{0, 1, ..., N_c - 1\}$ and N_c denotes the number of cameras. $(u_{ijk}^{n,m}, v_{ijk}^{n,m})$ denotes the m-th 2D reference point on n-th image view. $d_{ijk}^{n,m}$ is the depth in the camera frame. For the sake of simplicity, we denote the process of projecting from three-dimensional coordinates to pixels coordinates as $\{\tilde{\mathbf{Ref}}_{i,j,k}^m\} = \mathcal{P}_\mathbf{n}(\{\mathbf{Ref}_{i,j,k}^m\})$. Since not all cameras can capture the reference points of $\mathbf{v}$, we can further reduce computation by removing invalid sets from $\{\tilde{\mathbf{Ref}}_{i,j,k}^m\}$ if none of the reference points falls onto the image captured by the corresponding camera.

Inverse Projection. For further reducing memory cost, we decide to utilize the Tri-Perspective View (TPV) shape matrices as our mapping matrices from 2D images to 3D voxels for storing the current mapping strategy and represent the 3D voxel semantics as TPV features. In detailed, for r-th $\hat{\mathbf{V}}$, we first define its mapping matrices as follows:

$$\begin{aligned} &\mathbf{M}_r = [\mathbf{M}_r^{\hat{X}\hat{Y}}, \mathbf{M}_r^{\hat{Y}Z}, \mathbf{M}_r^{\hat{X}Z}],\ \mathbf{M}_r^{\hat{X}\hat{Y}} \in \mathbb{R}^{N_cHW\times\hat{X}\hat{Y}}, \\ &\mathbf{M}_r^{\hat{Y}Z} \in \mathbb{R}^{N_cHW\times\hat{Y}Z},\ \mathbf{M}_r^{\hat{X}Z} \in \mathbb{R}^{N_cHW\times\hat{X}Z}, \end{aligned} \tag{3}$$

where H, W denotes the weight and width of image feature maps and $r \in \{1, 2, 3, 4\}$. All elements of $\mathbf{M}_r$ are filled with 0. Since the mappings contained in $\mathbf{M}_r$ and $\{\tilde{\mathbf{Ref}}_{i,j,k}^m\}$ are exactly opposite, we obtain $\mathbf{M}_r$ through $\{\tilde{\mathbf{Ref}}_{i,j,k}^m\}$ by inverse projection $\mathcal{P}_I$:

$$\mathbf{M}_r = \mathcal{P}_I(\{\tilde{\mathbf{Ref}}_{i,j,k}^m\}). \tag{4}$$

Specifically, for $\tilde{\mathbf{Ref}}_{i,j,k}^m = (u_{ijk}^{n,m}, v_{ijk}^{n,m})$, we first encode it as $\{\tilde{\mathbf{Ref}}_{i,j,k}^m\} = \{(p_{ijk}^{n,m})\}$ for obtaining its position of the feature map after flattening:

$$p_{ijk}^{n,m} = u_{ijk}^{n,m} + v_{ijk}^{n,m}W + nHW, \tag{5}$$

where $p_{ijk}^{n,m} \in [1, N_cHW]$. Each pixel coordinates $(u_{ijk}^{n,m}, v_{ijk}^{n,m})$ in each voxel can be expressed as a value $p_{ijk}^{n,m}$ in one dimension. Then, we set the element in $\mathbf{M}_r$ to 1 if its corresponding point exists in $\{\tilde{\mathbf{Ref}}_{i,j,k}^m\}$ and obtain the coarse mapping matrix $\mathbf{M}_r$ that converts 2D image semantics into TPV features. Next, we map the one fused feature map $\phi \in \mathbb{R}^{C\times N_cHW}$ with 1/16 of the input resolution to coarse TPV features $F_r = [F_r^{\hat{X}\hat{Y}}, F_r^{\hat{X}Z}, F_r^{\hat{Y}Z}]$ with $\mathbf{M}_r$ as follows:

$$F_r^{\hat{X}\hat{Y}/\hat{X}Z/\hat{Y}Z} = f_{mlp}(\phi * \mathbf{M}_r^{\hat{X}\hat{Y}/\hat{X}Z/\hat{Y}Z}) \in \mathbb{R}^{C\times\hat{X}\hat{Y}/\hat{X}Z/\hat{Y}Z}, \tag{6}$$

where C is the channel number and $f_{mlp}(\cdot)$ denotes the multilayer perception layer (MLP). This MLP layer learns the characteristics of Sinkhorn Iteration for better optimization and fine-tune. The coarse mapping matrices $\mathbf{M}_r$ and TPV features F_r will be used to calculate initial transportation cost and finding the best semantic conversion strategy by refining $\mathbf{M}_r$ in the following module.

Algorithm 1 Optimal Transport for Strategy Optimization

```
Input:
    $\phi \in \mathbb{R}^{C\times N_c\times H_f\times W_f}$ is the feature map
    $F \in \mathbb{R}^{C\times H_t\times W_t}$ is the coarse TPV feature
    $\mathbf{M} \in \mathbb{R}^{N_cH_fW_f\times H_tW_t}$ is the coarse mapping matrix
    $\mathcal{A} \in \mathbb{R}^{H_tW_t}$ is the sum of each column of $\mathbf{M}$
    $G$ is the ground truth for $F$
Parameters:
    $C$ is the number of the channel
    $N_c$ is the number of the camera view
    $(H_f, W_f)$ is resolution of the feature map
    $(H_t, W_t)$ is resolution of the TPV feature
    $\gamma$ is the regularization intensity in Sinkhorn-Knopp Iter.
    $T$ is the number of iterations in Sinkhorn-Knopp Iter.
Output:
    $\pi^*$ is the optimal strategy
$P \leftarrow$ Forward$(F)$
$s_\alpha(\alpha = 1, 2...., N_cH_fW_f) \leftarrow \sum_{j=1}^{H_tW_t}(\mathbf{M}[\alpha, j])$
$d_\beta(\beta = 1, 2, ..., H_tW_t) \leftarrow$ OnesInit
classification cost: $C \in \mathbb{R}^{H_t\times W_t} \leftarrow$ CE$(P, G)$
$C$ is flattened and broadcast into $\hat{C} \in \mathbb{R}^{N_cH_fW_f\times H_tW_t}$
transportation cost: $\tilde{C} = \mathbf{M} \cdot \hat{C}/\mathcal{A}$
$v^0, u^0 \leftarrow$ OnesInit
for t=0 to $T$ do:
    $v^{t+1}, u^{t+1} \leftarrow$ SinkhornIter$(c, v^t, u^t, s, d)$
obtain $\pi^*$ from SinkhornIter$(c, v^T, u^T, s, d)$
return $\pi^*$
```

Strategy Optimization. Supposing there are n_s image pixels and n_d grids (across all mapping matrices $\mathbf{M}_r$) for multi-view features ϕ, we view each pixel as a supplier who holds s_α units of semantics marks (*i.e.*, $\alpha = 1, 2, .., N_cHW$), and each grid as a demander who needs d_β units of semantic mark (*i.e.*, $\beta = 1, 2, .., \eta$), where $\eta \in \{\hat{X}\hat{Y}, \hat{X}Z, \hat{Y}Z\}$. The cost c_{p_α,g_β}^r for transporting one unit of semantic marks from pixel p_α to grid g_β is defined as the weighted occupancy prediction loss corresponding to g_β:

$$c_{p_\alpha,g_\beta}^r = \frac{1}{\sum_{i=1}^{N_CHW}\mathbf{M_r}[i, \beta]}\mathcal{L}(P(g_\beta), G_{g_\beta}), \tag{7}$$

where $P(g_\beta)$ denotes the predicted classification score and G_{g_β} denotes ground truth class for g_β. $\mathcal{L}(\cdot)$ generally stands for cross entropy loss. Specifically, we use top plane mapping matrix $\mathbf{M}_r^{\hat{X}\hat{Y}}$ to instantiate the above how to transportation cost for instance. We first use its corresponding top plane feature $F_j^{\hat{X}\hat{Y}}$ to calculating the classification loss:

$$C_r^{\hat{X}\hat{Y}} = \mathcal{L}(F_j^{\hat{X}\hat{Y}}, G^{\hat{X}\hat{Y}}) \in \mathbb{R}^{\hat{X}\times\hat{Y}}. \tag{8}$$

Then, we flatten the $C_r^{\hat{X}\hat{Y}}$ into a vector and broadcast the

Method	mIoU	others ■	barrier ■	bicycle ■	bus ■	car ■	const. veh. ■	motorcycle ■	pedestrian ■	traffic cone ■	trailer ■	truck ■	drive. suf. ■	other flat ■	sidewalk ■	terrain ■	manmade ■	vegetation ■
MonoScene [Cao and others, 2022]	6.06	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
BEVDet [Huang *et al.*, 2021]	11.73	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
BEVStereo [Li *et al.*, 2023b]	24.51	5.73	38.41	7.88	38.70	41.20	17.56	17.33	14.69	10.31	16.84	29.62	54.08	28.92	32.68	26.54	18.74	17.49
OccFormer [Zhang and others, 2023]	21.93	5.94	30.29	12.32	34.40	39.17	14.44	16.45	17.22	9.27	13.90	26.36	50.99	30.96	34.66	22.73	6.76	6.97
RenderOcc [Pan *et al.*, 2023]	26.11	4.84	31.72	10.72	27.67	26.45	13.87	18.20	17.67	17.84	21.19	23.25	**63.20**	36.42	**46.21**	**44.26**	19.58	**20.70**
BEVFormer [Cortinhal and others, 2020]	26.88	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	36.00	28.06	20.04	17.69
TPVFormer [Huang *et al.*, 2023]	27.83	7.22	38.90	13.67	**40.78**	**45.90**	17.23	19.99	18.85	14.30	**26.69**	34.17	55.65	35.47	37.55	30.70	19.40	16.78
CTF-Occ [Zhu *et al.*, 2021]	28.53	**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**	18.00
OTOcc	**29.10**	7.81	39.23	19.45	39.54	42.45	**18.15**	**25.72**	**23.81**	20.13	24.56	**34.73**	54.45	**37.12**	37.23	34.56	18.53	17.10

Table 1: 3D occupancy prediction performance on the Occ3D-nuScenes dataset. Our OTOcc achieves competitive prediction performance compared with state-of-the-art methods.

Method	Input Modality	mIoU	barrier ■	bicycle ■	bus ■	car ■	const. veh. ■	motorcycle ■	pedestrian ■	traffic cone ■	trailer ■	truck ■	drive. suf. ■	other flat ■	sidewalk ■	terrain ■	manmade ■	vegetation ■
RangeNet++ [Milioto *et al.*, 2019]	LiDAR	65.5	66.0	21.3	77.2	80.9	30.2	66.8	69.6	52.1	54.2	72.3	94.1	66.6	63.5	70.1	83.1	79.8
PolarNet [Zhang *et al.*, 2020]	LiDAR	71.0	74.7	28.2	85.3	90.9	35.1	77.5	71.3	58.8	57.4	76.1	96.5	71.1	74.7	74.0	87.3	85.7
Salsanext [Cortinhal and others, 2020]	LiDAR	72.2	74.8	34.1	85.9	88.4	42.2	72.4	72.2	63.1	61.3	76.5	96.0	70.8	71.2	71.5	86.7	84.4
Cylinder3D++ [Zhu *et al.*, 2021]	LiDAR	76.1	76.4	40.3	91.2	**93.8**	51.3	78.0	78.9	64.9	62.1	84.4	96.8	71.6	**76.4**	75.4	90.5	87.4
RPVNet [Xu *et al.*, 2021]	LiDAR	**77.6**	**78.2**	43.4	92.7	93.2	49.0	**85.7**	**80.5**	**66.0**	66.9	84.0	**96.9**	**73.5**	75.9	**76.0**	**90.6**	**88.9**
BEVFormer-Base [Li *et al.*, 2022b]	Camera	56.2	54.0	22.8	76.7	74.0	45.8	53.1	44.5	24.7	54.7	65.5	88.5	58.1	50.5	52.8	71.0	63.0
TPVFormer-Base [Huang *et al.*, 2023]	Camera	68.9	70.0	40.9	93.7	85.6	49.8	68.4	59.7	38.2	65.3	83.0	93.3	64.4	64.3	64.5	81.6	79.3
Occformer [Zhang and others, 2023]	Camera	70.4	70.3	**43.8**	93.2	85.2	52.0	59.1	67.6	45.4	64.4	84.5	93.8	68.2	67.8	68.3	82.1	80.4
PanoOcc-Base [Wang *et al.*, 2023]	Camera	70.7	73.7	42.6	**94.1**	87.1	**56.4**	62.4	64.7	36.7	**69.3**	86.4	94.9	69.8	67.1	67.9	80.3	77.0
OTOcc	Camera	70.9	73.0	43.2	93.9	88.5	53.5	66.5	62.5	41.0	68.5	**87.2**	90.5	69.4	66.0	66.0	80.5	81.7

Table 2: LiDAR segmentation results on nuScenes validation dataset. Our OTOcc achieves comparable performance with state-of-the-art vision-based methods.

vector into the matrix $\hat{C}_r^{\hat{X}\hat{Y}} \in \mathbb{R}^{N_c HW \times \hat{X}\hat{Y}}$, so that we can calculate the initial transportation cost of $\mathbf{M}_r^{\hat{X}\hat{Y}}$ as follows:

$$\tilde{C}_r^{\hat{X}\hat{Y}} = \mathbf{M}_r^{\hat{X}\hat{Y}} \cdot \hat{C}_r^{\hat{X}\hat{Y}} / \mathcal{A}_r^{\hat{X}\hat{Y}} \in \mathbb{R}^{N_c HW \times \hat{X}\hat{Y}}, \tag{9}$$

where $\mathcal{A}_r^{\hat{X}\hat{Y}} \in \mathbb{R}^{\hat{X}\hat{Y}}$ denotes the sum of each column of $\mathbf{M}_r^{\hat{X}\hat{Y}}$, which is the vector. In a standard OT problem, the total supply must be equal to the total demand. Hence, for each supplier g_β, we sum up all elements in its corresponding row of $\mathbf{M}_r^{\hat{X}\hat{Y}}$ as its holding semantic marks. As we already have cost matrix $\tilde{C}^{\hat{X}\hat{Y}}$, supplying vector $s_\alpha \in \mathbb{R}^{N_c HW}$ and demanding vector $d_\beta \in \mathbb{R}^{\hat{X}\hat{Y}}$, the optimal transportation plan $\pi_r^{*,\hat{X}\hat{Y}} \in \mathbb{R}^{N_c HW \times \hat{X}\hat{Y}}$ can be obtained by solving this OT problem via the off-the-shelf Sinkhorn-Knopp Iteration [Cuturi, 2013]. After T iteration, we get $\pi_r^{*,\hat{X}\hat{Y}}$. One can decode the corresponding semantic conversion strategy by directly multiplying it with feature map by Eq. (6), which obtains the better top plane $\hat{F}_r^{\hat{X}\hat{Y}} = f_{mlp}(\phi \cdot \pi_r^{*,\hat{X}\hat{Y}})$. The similar procedure is repeated for all mapping matrices $\mathbf{M}_r$ as shown in Algorithm 1. Here, we get the refined TPV features $\hat{F}_r$ from each segmented voxel $\hat{\mathbf{V}}$, which contains the pure semantics from 2D images, facilitating accurate scene representation. Note that the optimization process of OT problem only contains some matrix multiplications which can be accelerated by GPU devices. Hence, the optimization module only increases the total training time.

3.3 Adaptive Fusion Decoder

After sending four small $\hat{\mathbf{V}}$ through the above modules, we obtain the refining TPV features $\hat{F}_r$ for each $\hat{\mathbf{V}}_r$, which collects pure semantics from 2D images. We first need to combine these features to form five diverse planes to represent the front, left, right, rear and top features as follows:

$$\begin{aligned}
&\hat{F}^{\hat{X}\hat{Y}}_{front} = \mathbb{C}(\hat{F}_1^{\hat{X}\hat{Y}}, \hat{F}_2^{\hat{X}\hat{Y}}), \hat{F}^{\hat{X}\hat{Y}}_{rear} = \mathbb{C}(\hat{F}_3^{\hat{X}\hat{Y}}, \hat{F}_4^{\hat{X}\hat{Y}}),\\
&\hat{F}^{top} = \mathbb{C}(\hat{F}^{\hat{X}\hat{Y}}_{front}, \hat{F}^{\hat{X}\hat{Y}}_{rear}), \hat{F}^{front} = \mathbb{C}(\hat{F}_1^{\hat{X}Z}, \hat{F}_2^{\hat{X}Z}),\\
&\hat{F}^{rear} = \mathbb{C}(\hat{F}_3^{\hat{X}Z}, \hat{F}_4^{\hat{X}Z}), \hat{F}^{left} = \mathbb{C}(\hat{F}_3^{\hat{Y}Z}, \hat{F}_1^{\hat{Y}Z}),\\
&\hat{F}^{right} = \mathbb{C}(\hat{F}_4^{\hat{Y}Z}, \hat{F}_2^{\hat{Y}Z}),
\end{aligned} \tag{10}$$

where $\mathbb{C}(\cdot)$ stands for the concat operation. Compared to ordinary TPV features setting [Huang *et al.*, 2023], our methods perform a more comprehensive representation of the 3D scene representation by introducing the Multi-Perspective View (MPV) features. Due to the top plane $\hat{F}^{top}$ containing comprehensive and significant labeling of scene category position relationships, we propose to decouple $\hat{F}^{top}$ on other planes $\hat{F}'$ (front, rear, left and right) for condensing voxel semantics. Specifically, we use several convolutional layers to highly aggregate plane feature semantics and perform dual fusion:

$$\hat{F}' \leftarrow \hat{F}' * \mathcal{P}_a(\hat{F}^{top}) + \mathcal{P}_b(\hat{F}') * \hat{F}^{top}, \tag{11}$$

where $\mathcal{P}_a(\cdot)$ is composed of three convolutional layers, a linear layer and an upsampled layer. $\mathcal{P}_b(\cdot)$ consists of two con-

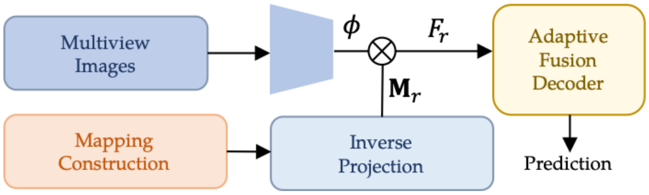


Figure 3: Illustration of the inference pipeline of OTOcc. Note that we don't need the Strategy Optimization with Sinkhorn-Knopp Iteration in this pipeline.

Method	Memory	Latency	FPS	mIoU
TPVFormer-Base	33.5G / 7.1G	268ms	3.7	68.9
PanoOcc-Base	24.0G / 6.0G	**203ms**	**4.8**	70.7
OTOcc (Ours)	**22.9G / 5.3G**	227ms	4.4	**70.9**

Table 3: Model efficiency comparison in LiDAR Segmentation. We report the train/inference memory consumption in the experiment.

volutional layers, a linear layer, an upsampled layer and a relu layer. For dense voxel features, we actively broadcast each plane $\hat{F}'$ along the corresponding orthogonal direction to produce three feature tensors of the same upsampled size $C \times X' \times Y' \times Z'$ and aggregate them by summation to obtain the full-scale voxel features. To conduct fine-grained prediction or segmentation tasks, we apply a lightweight MLP on voxel features to predict their semantic labels, which is instantiated by only three linear layers, and an intermediate activation layer.

3.4 Model Optimization

In the training stage of the OTOcc, the whole loss function $\mathcal{L}$ consists of three main components:

$$\mathcal{L} = \mathcal{L}_{CE} + \mathcal{L}_{Lovasz} + \lambda \mathcal{L}_{OT}, \tag{12}$$

where $\mathcal{L}_{CE}$ denotes the cross entropy loss (all voxels), $\mathcal{L}_{Lovasz}$ denotes the Lovasz loss [Berman *et al.*, 2018] (voxels containing LiDAR points) for voxel prediction and $\mathcal{L}_{OT}$ denotes the mean of the initial transportation cost of TPV mapping matrices M_r. We utilize the hyperparameter λ for controlling the sensitivity of $\mathcal{L}_{OT}$.

3.5 Model Inference

The whole test pipeline of our proposed OTOcc is as shown in Figure 3. Caused by the time cost of the Sinkhorn-Knopp Iteration, we abandon the Strategy Optimization module from the training framework, which promotes the inference speed significantly. Moreover, the f_{mlp} also plays a similar role as Strategy Optimization for fine-tuning the coarse TPV features, which contains sufficient inherent semantic knowledge from TPV features and voxels presentation.

4 Experiments

4.1 Datasets

Occ3D-nuScenes[1] [Tian *et al.*, 2023] contains 700 training scenes and 150 validation scenes. The occupancy scope is defined as $-40m$ to $40m$ for x-axis or y-axis, and $-1m$ to $5.4m$ for the z-axis in the ego coordinate. The voxel size is $0.4m \times 0.4m \times 0.4m$ for the occupancy label. The semantic labels contain 17 categories.

nuScenes [Caesar *et al.*, 2020] is a large-scale autonomous driving dataset, collected in Boston and Singapore. It includes 1000 driving sequences from various scenes, split into 700 in the training set, 150 in the validation set, and 150 in the test set. Each sequence is captured at 20Hz frequency with 20 seconds duration. Each sample contains RGB images from 6 cameras with 360°horizontal FOV and point cloud data from 32 beam LiDAR sensor.

4.2 Experimental Settings

We adopt TPVFormer as our baseline and use its framework for developing our method. Following the setting in previous works, we adopted ResNet101-DCN [Dai *et al.*, 2017] as the image backbone and trained the model on 8 NVIDIA A100 GPUs with a batch size of 1 per GPU. We also employ both cross entropy loss and Lovasz-softmax loss [Berman *et al.*, 2018]. In the implementation, both two loss use voxel predictions as input. During training, we utilize the AdamW optimizer with an initial learning rate of 2×10^{-4} and the cosine schedule. Additionally, we employ photo-metric distortion as the data augmentation technique. The initial resolution of the (X, Y, Z) is $(100, 100, 8)$ and the upsampled voxel features have dimensions of (X', Y', Z') is $(200, 200, 16)$. We also set $N_p = 4$, $T = 40$, $\lambda = 0.2$ and $C = 256$. Note that our proposed OTOcc is trained with only 15 epochs for efficiency.

4.3 Main Results

3D Occupancy Prediction. Evaluation results for 3D occupancy prediction are recorded in Table 1. OTOcc can achieve SOTA prediction performance with a 29.10% mIoU, outperforming existing counterparts significantly. Moreover, OTOcc surpasses CTF-Occ (28.53% mIoU), TPVFormer (28.34% mIoU) and BEVFormer (26.88% mIoU) with 0.57%, 0.76%, 2.22% mIoU gains, showing the advantage in terms of accurate semantic conversion. After that, our method can better detect easily confused objects such as the *motorcycle* (25.72%) or *pedestrian* (23.81%) than SOTAs.

3D Semantic Segmentation. We utilize the voxel predictions on sparse LiDAR points for the semantic segmentation evaluation. As shown in Table 2, we evaluate the semantic segmentation performance on the nuScenes validation set. OTOcc surpasses TPVFormer-Base, and Occformer with 2.0%, 0.5% mIoU, verifying its better ability to effectively sample semantics than these non-linear strategies. Moreover, OTOcc can achieve comparable performance to methods with LiDAR as input modality. For the model efficiency as shown in Table 3, we compare the performance and efficiency of our method with TPVformer and PanoOcc, under the same experimental setup. Our model still exhibits lower memory consumption with 22.9G training memory consumption (inference with 5.3G), showing the effectiveness of our methods. Moreover, our methods also have similar and competitive inference latency or FPS performance compared to PanoOcc.

[1] https://github.com/Tsinghua-MARS-Lab/Occ3D/

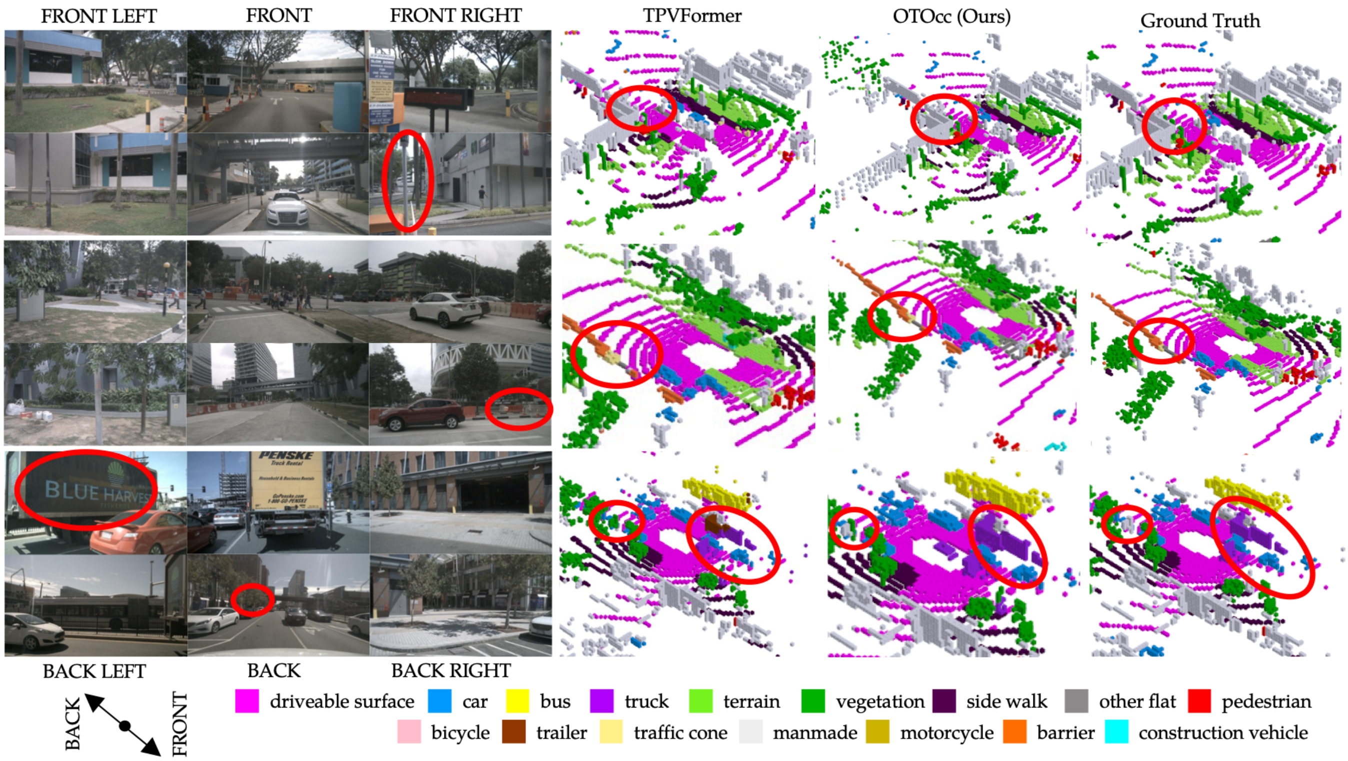


Figure 4: Qualitative results about LiDAR segmentation on nuScenes among TPVFormer, the proposed OTOcc and Ground-truth (GT). Red circles denote the instance of baseline failure detection. Zooming in best views.

N_p^x	N_p^y	N_p^z	Occ. mIoU	LiDAR Seg. mIoU
3	3	3	27.45	65.79
3	3	4	28.76	66.98
3	4	4	28.36	68.34
4	4	4	29.10	70.87
4	4	5	29.02	**70.92**
4	5	5	**29.21**	69.78
5	5	5	28.57	68.36
6	6	6	26.31	66.43

Table 4: Different number of sampling points $N_p^i, i \in \{x, y, z\}$ settings on each voxel grid for building reference points.

$\mathbf{M}_r^{\hat{X}\hat{Y}}$	$\mathbf{M}_r^{\hat{Y}Z}$	$\mathbf{M}_r^{\hat{X}Z}$	Occ. mIoU	LiDAR Seg. mIoU
-	-	-	26.04	65.21
✓	-	-	27.98	67.87
-	✓	-	27.34	66.89
-	-	✓	27.82	66.21
✓	✓	-	**29.25**	70.13
✓	-	✓	28.44	70.68
-	✓	✓	27.23	69.14
✓	✓	✓	29.10	**70.87**

Table 5: Different mapping matrices $\mathbf{M}_r^*, * \in \{\hat{X}\hat{Y}, \hat{X}Z, \hat{Y}Z\}$ choices for strategy optimization.

4.4 Ablation Studies

We present detailed ablation studies of Occupancy Prediction on Occ3D-nuScenes and LiDAR segmentation on the validation sets of nuScenes respectively.

Ablation for Sampling Points

Our framework relies on the quality of mapping matrices. The sampling point setting is essential for constructing our initial reference points. As shown in Table 4, we perform analysis on different numbers of points settings on each voxel. $N_p^i, i \in \{x, y, z\}$ denotes the number of points sampled along i-axis in a voxel. We can conclude that the sparse point setting like $N_p^i = 3$ causes a worse result because sparse points are struggling to capture scene semantics. Meanwhile, too dense point setting like $N_p^i = 6$ leads to significant performance degradation because the diversity of the mapping relationship has approached its limit, which brings more noise confusing our model. $N_p^i = 4$ can strike the right balance between projecting appropriate image semantics and sampling noise, which prevents our model from overfitting catastrophe.

Ablation for Strategy Optimization

OTOcc presents a coarse-to-fine semantic conversion strategy. We compare different settings of the iteration number T, optimization object $\mathbf{M}_r$ choice of each $\hat{\mathbf{v}}$ and sensitive parameter λ in Sinkhorn-Knopp Iteration for exploring their impact on semantic conversion.

Mapping Matrices Optimization. As shown in Table 5, we

T	Time	λ	Occ. mIoU	LiDAR Seg. mIoU
0	1.207s	0	26.04	65.21
30	1.427s	1.0	27.04	68.41
		0.5	27.67	68.29
		0.2	28.13	69.43
40	1.545s	1.0	28.43	69.54
		0.5	28.97	70.05
		0.2	29.10	**70.87**
50	1.682s	1.0	27.62	69.31
		0.5	**29.21**	69.64
		0.2	28.65	70.12

Table 6: Various number of iterations T and hyperparameter λ settings for Sinkhorn-Knopp module. "Time" denotes the time cost of each training iter.

can conclude that if we don't optimize any mapping matrices and directly utilize the coarse $\mathbf{M}_r$ for semantic conversion, it leads to serious performance degradation (like 3.06% decrease in occupancy prediction). Even though, it still demonstrates strong prediction accuracy compared to BEVFormer, thanks to the superiority of MPV features representation. Then, we find that compared with optimizing other plane features, optimizing the top plane $\mathbf{M}_r^{\hat{X}\hat{Y}}$ can significantly enhance the semantic representation of 3D scenes, benefiting from its comprehensive spatial information representation.

Sinkhorn-Knopp Iteration Setting. As shown in Table 6, we observe that a higher λ ($\lambda = 1$) setting compromises the model performance, which focuses on the loss of a single plane is inconsistent with the loss optimization direction of the overall calculation of 3D voxels. Then, higher T will bring better performance, but also increase the computational cost. Like $T = 30$, the Sinkhorn-Knopp Iteration can bring more than 1.00% mIoU gains on occupancy prediction or 3.20% mIoU gains on LiDAR segmentation compared to $T = 0$, while the training time increases by more than 16% due to the large scale optimization. When $T = 50$, performance on two benchmarks generally decreases caused of the optimization limits from the Sinkhorn-Knopp Iteration. To balance the computational cost and prediction performance, we set the T to 40 and λ to 0.2.

4.5 Qualitative Results

We present the comparisons of LiDAR segmentation results among TPVFormer, OTOcc and ground truth, which are shown in Figure 4. OTOcc can reduce class errors in LiDAR segmentation, such as the *truck* in the third row while TPVFormer classifies it as a *trailer*. We also observe that OTOcc has a better perception of small and blurry objects like the less occupied *traffic cone* in the second row or the *manmade* in the third row, demonstrating the advantages of finding the optimal semantic transport strategy for significant prediction.

5 Discussion

Motivation. From another perspective, calculating the optimal semantic conversion strategy is equivalent to performing auxiliary task learning, not limited to occupancy prediction. OTOcc utilizes transportation loss to calibrate the MPV feature semantics, thereby providing a guarantee for building accurate voxel representation.

Limitation. We admit that the deformable attention [Zhu *et al.*, 2020] mechanism has a better computation speed than OT-based methods no matter from theoretical derivation or experiments as shown in the speed comparison results with PanoOcc from Table 3. However, it only samples several points on key, which easily aggregates semantics within a certain range leading to the slower convergence. In future works, we will investigate more efficient methods in OT-based deeply. After that, the Sinkhorn Iteration still brings the high training time cost and the instability prediction performance, easily leading to the cherry picking. Then, OTOcc only uses a lightweight decoder to explore the effectiveness of applying OT methods, which motivates us to design a more powerful decoder for accurate occupancy prediction.

6 Conclusion

In this paper, we propose a novel framework named OTOcc —— Optimal Transport for Occupancy Prediction. It represents the semantic conversion from 2D images to 3D voxels as the Optimal Transport problem, which breaks the barrier of existing approaches relying on attention or depth estimation to collect information in sparse semantic scenes. It generates initial mapping matrices from the designed dense reference points, which are refined via Sinkhorn-Knopp Iteration to seek the best semantic conversion strategy. Then, it adopts an Adaptive Fusion Decoder for obtaining accurate voxel representation. Extensive experiments on two benchmarks show the effectiveness of our proposed method.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (NSFC) under Grants 62271324, 62231020, 62371309 and 61976142, Shenzhen Science and Technology Program under Grant ZDSYS20220527171400002 and the Hetao Shenzhen-HongKong Science and Technology Innovation Cooperation Zone (HZQB-KCZYZ-2021055). This work is also funded by Shenzhen Deeproute.ai Co., Ltd.

References

[Berman *et al.*, 2018] Maxim Berman, Amal Rannen Triki, and Matthew B Blaschko. The lovász-softmax loss: A tractable surrogate for the optimization of the intersection-over-union measure in neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, 2018.

[Caesar *et al.*, 2020] 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.

[Cao and others, 2022] Cao et al. Monoscene: Monocular 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3991–4001, 2022.

[Cortinhal and others, 2020] Cortinhal et al. Salsanext: Fast, uncertainty-aware semantic segmentation of lidar point clouds. In *Advances in Visual Computing: 15th International Symposium, ISVC 2020, San Diego, CA, USA, October 5–7, 2020, Proceedings, Part II 15*, pages 207–222. Springer, 2020.

[Cuturi, 2013] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in Neural Information Processing Systems*, 26, 2013.

[Dai *et al.*, 2017] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 764–773, 2017.

[Ge *et al.*, 2021] Zheng Ge, Songtao Liu, Zeming Li, Osamu Yoshie, and Jian Sun. Ota: Optimal transport assignment for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 303–312, 2021.

[Huang *et al.*, 2021] 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.

[Huang *et al.*, 2023] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9223–9232, 2023.

[Jia *et al.*, 2023] Yupeng Jia, Jie He, Runze Chen, Fang Zhao, and Haiyong Luo. Occupancydetr: Making semantic scene completion as straightforward as object detection. *arXiv preprint arXiv:2309.08504*, 2023.

[Li *et al.*, 2022a] Wuyang Li, Xinyu Liu, and Yixuan Yuan. Scan++: Enhanced semantic conditioned adaptation for domain adaptive object detection. *IEEE Transactions on Multimedia*, 2022.

[Li *et al.*, 2022b] Zhiqi Li, Wenhai Wang, Hongyang Li, Enze Xie, Chonghao Sima, Tong Lu, Yu Qiao, and Jifeng Dai. Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers. In *European Conference on Computer Vision*, pages 1–18. Springer, 2022.

[Li *et al.*, 2023a] Yiming Li, Zhiding Yu, Christopher Choy, Chaowei Xiao, Jose M Alvarez, Sanja Fidler, Chen Feng, and Anima Anandkumar. Voxformer: Sparse voxel transformer for camera-based 3d semantic scene completion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9087–9098, 2023.

[Li *et al.*, 2023b] Yinhao Li, Han Bao, Zheng Ge, Jinrong Yang, Jianjian Sun, and Zeming Li. Bevstereo: Enhancing depth estimation in multi-view 3d object detection with temporal stereo. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1486–1494, 2023.

[Li *et al.*, 2023c] 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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1477–1485, 2023.

[Li *et al.*, 2023d] 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.

[Lu *et al.*, 2023] Yuhang Lu, Xinge Zhu, Tai Wang, and Yuexin Ma. Octreeocc: Efficient and multi-granularity occupancy prediction using octree queries. *arXiv preprint arXiv:2312.03774*, 2023.

[Miao *et al.*, 2023a] Ruihang Miao, Weizhou Liu, Mingrui Chen, Zheng Gong, Weixin Xu, Chen Hu, and Shuchang Zhou. Occdepth: A depth-aware method for 3d semantic scene completion. *arXiv preprint arXiv:2302.13540*, 2023.

[Miao *et al.*, 2023b] Ruihang Miao, Weizhou Liu, Mingrui Chen, Zheng Gong, Weixin Xu, Chen Hu, and Shuchang Zhou. Occdepth: A depth-aware method for 3d semantic scene completion. *arXiv preprint arXiv:2302.13540*, 2023.

[Milioto *et al.*, 2019] Andres Milioto, Ignacio Vizzo, Jens Behley, and Cyrill Stachniss. Rangenet++: Fast and accurate lidar semantic segmentation. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4213–4220. IEEE, 2019.

[Ouyang *et al.*, 2024] Wenzhe Ouyang, Xiaolin Song, Bailan Feng, and Zenglin Xu. Octocc: High-resolution 3d occupancy prediction with octree. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 4369–4377, 2024.

[Pan *et al.*, 2023] Mingjie Pan, Jiaming Liu, Renrui Zhang, Peixiang Huang, Xiaoqi Li, Li Liu, and Shanghang Zhang. Renderocc: Vision-centric 3d occupancy prediction with 2d rendering supervision. *arXiv preprint arXiv:2309.09502*, 2023.

[Qi *et al.*, 2023] Zhangyang Qi, Jiaqi Wang, Xiaoyang Wu, and Hengshuang Zhao. Ocbev: Object-centric bev transformer for multi-view 3d object detection. *arXiv preprint arXiv:2306.01738*, 2023.

[Tan *et al.*, 2023] Zhiyu Tan, Zichao Dong, Cheng Zhang, Weikun Zhang, Hang Ji, and Hao Li. Ovo: Open-vocabulary occupancy. *arXiv preprint arXiv:2305.16133*, 2023.

[Tian *et al.*, 2023] Xiaoyu Tian, Tao Jiang, Longfei Yun, Yue Wang, Yilun Wang, and Hang Zhao. Occ3d: A large-scale 3d occupancy prediction benchmark for autonomous driving. *arXiv preprint arXiv:2304.14365*, 2023.

[Wang *et al.*, 2023] Yuqi Wang, Yuntao Chen, Xingyu Liao, Lue Fan, and Zhaoxiang Zhang. Panoocc: Unified occupancy representation for camera-based 3d panoptic segmentation. *arXiv preprint arXiv:2306.10013*, 2023.

[Xu *et al.*, 2021] Jianyun Xu, Ruixiang Zhang, Jian Dou, Yushi Zhu, Jie Sun, and Shiliang Pu. Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16024–16033, 2021.

[Yang *et al.*, 2023] Chenyu Yang, Yuntao Chen, Hao Tian, Chenxin Tao, Xizhou Zhu, Zhaoxiang Zhang, Gao Huang, Hongyang Li, Yu Qiao, Lewei Lu, et al. Bevformer v2: Adapting modern image backbones to bird's-eye-view recognition via perspective supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17830–17839, 2023.

[Zhang and others, 2023] Zhang et al. Occformer: Dual-path transformer for vision-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2304.05316*, 2023.

[Zhang *et al.*, 2020] Yang Zhang, Zixiang Zhou, Philip David, Xiangyu Yue, Zerong Xi, Boqing Gong, and Hassan Foroosh. Polarnet: An improved grid representation for online lidar point clouds semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9601–9610, 2020.

[Zhou *et al.*, 2022] Hongyu Zhou, Zheng Ge, Zeming Li, and Xiangyu Zhang. Matrixvt: Efficient multi-camera to bev transformation for 3d perception. *arXiv preprint arXiv:2211.10593*, 2022.

[Zhu *et al.*, 2020] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.

[Zhu *et al.*, 2021] Xinge Zhu, Hui Zhou, Tai Wang, Fangzhou Hong, Yuexin Ma, Wei Li, Hongsheng Li, and Dahua Lin. Cylindrical and asymmetrical 3d convolution networks for lidar segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9939–9948, 2021.