ZOPP: A Framework of Zero-shot Offboard Panoptic Perception for Autonomous Driving

Tao Ma[1,2*] **Hongbin Zhou**[2*] **Qiusheng Huang**[2*] **Xuemeng Yang**[2] **Jianfei Guo**[2]

Bo Zhang[2] **Min Dou**[2] **Yu Qiao**[2] **Botian Shi**[2]

Hongsheng Li[1,3]

[1]Multimedia Laboratory, The Chinese University of Hong Kong
[2]Shanghai Artificial Intelligence Laboratory [3]CPII

taoma@link.cuhk.edu.hk, hsli@ee.cuhk.edu.hk
{zhouhongbin, huangqiusheng, yangxuemeng, shibotian}@pjlab.org.cn

Abstract

Offboard perception aims to automatically generate high-quality 3D labels for autonomous driving (AD) scenes. Existing offboard methods focus on 3D object detection with closed-set taxonomy and fail to match human-level recognition capability on the rapidly evolving perception tasks. Due to heavy reliance on human labels and the prevalence of data imbalance and sparsity, a unified framework for offboard auto-labeling various elements in AD scenes that meets the distinct needs of perception tasks is not being fully explored. In this paper, we propose a novel multi-modal Zero-shot Offboard Panoptic Perception (ZOPP) framework for autonomous driving scenes. ZOPP integrates the powerful zero-shot recognition capabilities of vision foundation models and 3D representations derived from point clouds. To the best of our knowledge, ZOPP represents a pioneering effort in the domain of multi-modal panoptic perception and auto labeling for autonomous driving scenes. We conduct comprehensive empirical studies and evaluations on Waymo open dataset to validate the proposed ZOPP on various perception tasks. To further explore the usability and extensibility of our proposed ZOPP, we also conduct experiments in downstream applications. The results further demonstrate the great potential of our ZOPP for real-world scenarios. Code will be released at https://github.com/PJLab-ADG/ZOPP.

1 Introduction

Comprehensive perception and understanding of 3D scenes are important for autonomous driving (AD). We have witnessed the evolution of machine perception at different levels within a short period: from single-modal [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12] to multi-modal inputs [13, 14, 15, 16, 17, 18], from limited categories to open set [19, 20, 21, 22, 23, 24, 25], from 3D box to 3D occupancy [26, 27, 28, 29, 30, 31, 32], and from low-level detection to high-level understanding. Though remarkable, to train a model for different AD perception tasks, huge amounts of high-quality data and labels are still required, which is time-consuming and expensive. Therefore, it is essential to come up with an efficient solution to cope with such rapid changes.

*Equally contributed to the work.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Table 1: Comparisons of recent onboard and offboard perception models. *Seg.*, *Det.*, *Occ.* represent 3D segmentation, 3D object detection, occupancy prediction, respectively. *HLF* means training in a human-label-free manner. *Grounding* highlights models that can respond with text prompts. *Zero.* stands for the zero-shot capability for unseen classes.

Method	LiDAR	Image	Seg.	Det.	Occ.	HLF	Grounding	Zero.
3DAL [33]	✓	✗	✗	✓	✗	✗	✗	✗
CTRL [34]	✓	✗	✗	✓	✗	✗	✗	✗
DetZero [35]	✓	✗	✗	✓	✗	✗	✗	✗
LabelFormer [36]	✓	✗	✗	✓	✗	✗	✗	✗
UniSeg [14]	✓	✓	✓	✗	✗	✗	✗	✗
LidarMultiNet [37]	✓	✗	✓	✓	✗	✗	✗	✗
SAM3D [38]	✓	✗	✗	✓	✗	✓	✗	✓
ZOPP (ours)	✓	✓	✓	✓	✓	✓	✓	✓

Recently, offboard detection and auto-labeling have gained significant attention in the field of AD, which focuses on alleviating the burdens of human labor and the cost of labeling huge amounts of data. These methods [33, 34, 35] have showcased impressive performance for point clouds based 3D object detection with closed-set taxonomy (*e.g.*, predefined categories of vehicles, pedestrians, and cyclists) compared to humans. However, their modular fashion always needs high-quality human labels as a prerequisite for training the whole pipeline, which places the auto labeling as a chicken-or-egg problem. Due to the data sparsity and imbalance, the supervised training fashion on limited seen categories also struggles to effectively perform auto-labeling in open-set settings. In particular, the compensated points of small and distant objects (*e.g.*, traffic cone, traffic light) over the entire sequence are still extremely sparse, so the auto-labeling models will lose effectiveness during object-centric prediction. Furthermore, these auto-labeling models might not flexibly generalize well due to unavoidable domain shifts arising from different types of 3D sensors. To sum up, we found that all these shortages greatly limit the broad application prospects, and the development of a unified framework for offboard auto labeling that effectively meets the distinct needs of each perception task has not been fully explored.

To tackle this challenge, we propose ZOPP, which is a novel pioneering **Z**ero-shot **O**ffboard **P**anoptic **P**erception framework with multi-modal data input and supports a wide range of perception tasks in AD scenes. The core of ZOPP is a compact and lightweight pipeline to achieve panoptic perception without any human-label-based model training.

Specifically, ZOPP first extends SAM-Track [39] to multi-view images to achieve open-set 2D detection for object tracking and instance segmentation. Based on the aligned correspondence between point clouds and multi-view images, we can obtain robust semantic and instance segmentation for each 3D point with the proposed parallax occlusion and noise filtering module. The points belonging to a specific object can be aggregated via the pose matrix, and then fed into the proposed point completion module to generate dense point clouds. Equipped with such dense and high-quality object points (especially for dynamic objects), we can acquire precise 3D bounding boxes in a human-label-free manner. Furthermore, to achieve 3D occupancy prediction, unlike straightforward voxel feature generation from image features or solely using BEV feature as in previous literature [26, 30, 29], ZOPP employs neural rendering based reconstruction [40] to decode 3D occupancy from the reconstructed scenes. All the instance and semantic information are fused and leads to 4D occupancy flow as the final output.

We conduct comprehensive empirical studies and evaluations on the large-scale Waymo open dataset, to validate the proposed ZOPP on various perception tasks, *i.e.*, 2D/3D semantic and panoptic segmentation, 2D/3D detection and tracking, 4D occupancy flow prediction. It is noteworthy that ZOPP not only produces 3D bounding boxes for the common object categories, but also integrates the open-set detection capabilities into the 3D object detection task, which shows a more profound significance for small and distant objects. Extensive ablation studies and generalization experiments show that each proposed module of ZOPP performs well with their respective functions.

To further explore the generalization of our proposed ZOPP, we also conduct experiments in downstream applications and demonstrate ZOPP's great potential. ZOPP can be utilized as a quick

cold-start paradigm for existing auto-labeling methods. The completed dense object points can not only further boost the performance of their object-centric refining fashion, but also be used for generative assets modeling in simulation.

2 Related Work

Open-set 2D&3D Object Detection Open-set object detection is trained using existing bounding box annotations and aims at detecting arbitrary classes with the help of language generalization. Current image-based 2D open-set detectors often employ CLIP [41] to encode the text embedding as queries to decode the category-specific boxes [24], or as knowledge distillation to learn region embeddings containing the language semantics [42]. Leveraging additional data to train the model in grounding [43] and captioning [44] fashions, can also improve the generalization ability.

For 3D point clouds, transferring image or vision-language pre-trained models is very challenging. PointCLIP [22] achieves open-vocabulary point-cloud recognition via projecting point-cloud into multi-view images. Explorations of data augmentation [45] and construction [46] are conducted to improve open-set point cloud learning. Multi-model pre-trained models are also employed to enable open-set 3D detection for indoor scenes [47, 25]. MLUC [23] combines metric learning and unsupervised clustering for limited unknown categories in the outdoors. However, these approaches are still far from large-scale open-set settings for outdoor driving scenarios.

3D Segmentation and Occupancy Prediction 3D segmentation includes semantic [48, 14] and panoptic [49, 50] segmentation tasks by involving point clouds or multi-modal fusion with images [51, 52, 53, 54]. Some work [55, 56, 54] also associates features from previous frames to establish 4D panoptic segmentation. Meanwhile, zero-shot segmentation is explored by implicitly estimating the distribution of unseen features [20], or visual feature guidance [21].

Occupancy prediction recently arises with proposed benchmarks [28, 27]. Visual features are leveraged to construct dense 3D occupancy with semantic labels [26, 30]. However, these methods are all trainable with human labels or point clouds supervision [29]. In our offboard setting, we can employ 3D reconstruction and neural rendering [40, 57] in our pipeline, to concentrate more on the quality of the scene geometry and visual appearance.

Offboard Auto Labeling Relying on the serialized point cloud datasets, offboard 3D detection approaches often follow a modular pipeline design [33, 58, 34, 35], and leverage off-the-shelf 3D detectors [1, 2, 3, 5], trackers [59, 60, 61], and object-centric refining, to boost high-quality bounding boxes for auto-labeling. LidarMultiNet [37] unifies 3D segmentation and detection in one network, achieving performance gains on both tasks. Unfortunately, these methods only focus on 3D object detection, their modules still require huge amounts of data with high-quality annotations, and lack the capabilities of open-set and zero-shot settings.

In this paper, we focus on addressing the zero-shot offboard panoptic perception, and integrate the aforementioned perception tasks with an offboard running manner into the outdoor AD scenes.

3 Methodology

In this section, we introduce the general framework and workflow of our proposed ZOPP in detail, which generates multiple robust perception results from multi-view images and point clouds. As shown in Fig. 1, our method comprises four stages: (1) generating multi-view object mask tracks by Multi-view SAM-Track in Sec. 3.1, (2) Point Cloud Segmentation with aligned spatial correspondence and parallax occlusion filtering in Sec. 3.2, (3) 3D Box Interpretation after completing the partial points in Sec. 3.3, and (4) 4D Occupancy Reconstruction with neural rendering in Sec. 3.4.

3.1 Multi-view Mask Track Generation

Taking as input multi-view images and text prompts, we generate 2D panoptic segmentation and tracking results with the proposed Multi-view SAM-Track.

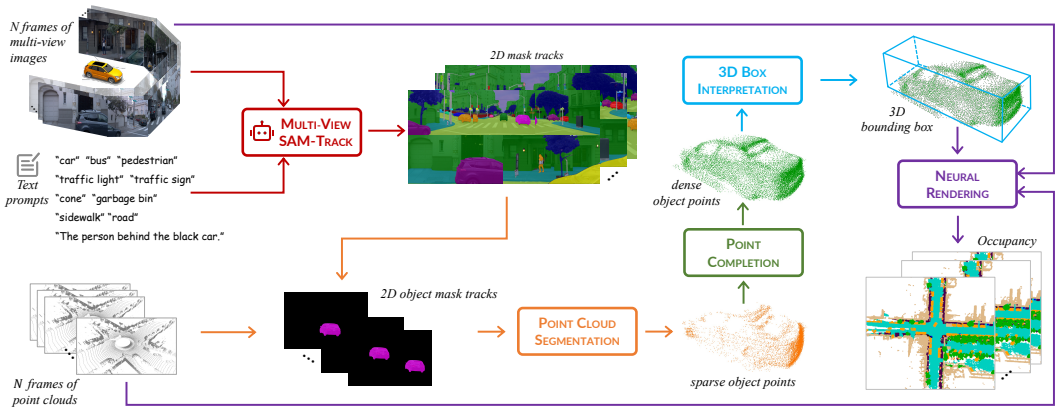


Figure 1: Overview of our proposed ZOPP. The core of ZOPP is a complete pipeline to achieve offboard panoptic perception of AD scenes, including multi-view mask track generation (**Red**), 3D semantic and instance segmentation (**Orange**), point cloud completion (**Green**), 3D detection (**Blue**), and 4D occupancy reconstruction (**Purple**).

3.1.1 Single-view Mask Tracking

We employ SAM-Track [39] to establish interactive open-set 2D object detection for segmenting and tracking in outdoor AD scenes. Specifically, SAM-Track first includes a powerful open-set object detector, Grounding-DINO [19], to detect objects in each frame according to predetermined text prompts (*e.g.*, "car", "the woman in a red dress"). Then, SAM [62] is leveraged to obtain segmentation masks for each object in the frame, serving as a reference input for DeAOT [63], a highly efficient multi-object tracking model. DeAOT hierarchically propagates the extracted visual embeddings and ID embeddings for each object from past to current frames based on the segmentation reference, to establish object tracking frame-by-frame.

3.1.2 Multi-view SAM-Track

Considering the prevalent use of multi-view cameras in AD, we design a simple yet effective similarity cost to measure the semantic and instance consistency among objects across all the views. This cost involves the computation of appearance and location similarities to facilitate object association.

We first apply the aforementioned process to each image sequence of different views, yielding independent tracking results. Simultaneously, we obtain the appearance information of each object by extracting the visual features of Grounding-DINO and DeAOT with the 2D boxes. So the appearance similarity is compared across different objects by computing the cosine distance of the visual features. In contrast, the location similarity is derived by concatenating the images of all viewpoints in a panoramic order, followed by normalizing the pixel distances along the horizontal axis for each object. Hence, objects with large similarity scores would be associated together with the same instance ID.

As illustrated in Fig. 2, the use of appearance similarity allows for the discrimination of objects that are spatially close but exhibit significant visual differences. Meanwhile, distance similarity serves to prevent the matching of objects with similar appearances but substantial spatial separation. This comprehensive design thereby enhances the robustness and accuracy in multi-view settings.

For the sake of the grounding ability, we preserve the interactive mode to select target objects through natural language. For the automatic mode, we output all the object categories arising at the driving surroundings, to establish multi-view panoptic segmentation. Finally, we directly output the tracked object masks with a unique ID and corresponding categories as the final 2D semantic and instance segmentation results. Note that background objects (*e.g.*, buildings, trees) only have semantic segmentation results.

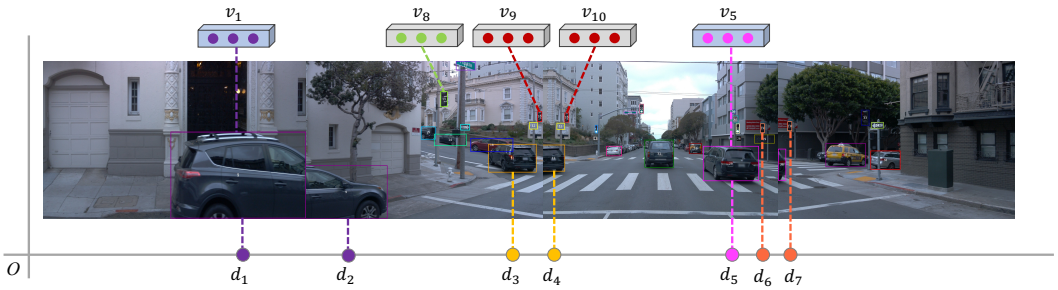


Figure 2: Overview of our object association across multiple views. Multi-view images are concatenated in a panoramic order. The visual features and horizontal pixel coordinates of each object are drawn at the top and bottom of the images, respectively. Visual features v_1 and v_5 are very similar, so the location differences d_1 and d_5 contribute to the matching determination. The visual features of traffic lights are almost the same (v_8, v_9, v_{10}), so we can associate them with location similarities (d_6, d_7).

3.2 Point Cloud Segmentation

In this section, point cloud data is introduced to be well-aligned with multi-view image planes to obtain corresponding semantic and instance information. Then, we can extract points belonging to each foreground object based on the instance ID, for subsequent 3D box interpretation. The extraction is carefully established by our proposed parallax occlusion and noise filtering module.

3.2.1 Multi-modal Spatial Alignment

We denote a frame of point cloud as $\mathcal{P}^{\mathrm{L}} = \{p_1^{\mathrm{L}}, p_2^{\mathrm{L}}, ...\}$, where L represents the LiDAR coordinate system. For each 3D point $p_i^{\mathrm{L}} = (x_i, y_i, z_i)^T \in \mathbb{R}^3$, we denote its corresponding pixel coordinate on the image plane as $q_i = (u_i, v_i)^T \in \mathbb{R}^2$. The point and the pixel can be correlated by the calibration process in two steps. Firstly, p_i^{L} is transformed to the camera coordinate system C as $p_i^{\mathrm{C}} \in \mathbb{R}^3$ through $p_i^{\mathrm{C}} = \mathbf{R} \cdot p_i^{\mathrm{L}} + \mathbf{t}$ ($\mathbf{R}$ and $\mathbf{t}$ represent the rotation and translation between LiDAR and multi-view cameras). Next, p_i^{C} is projected onto the image plane through a projection function: $q_i = \mathbf{K}(p_i^{\mathrm{C}})$ ($\mathbf{K} : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ is defined with the camera intrinsic parameter for each specific view).

3.2.2 Parallax Occlusion and Noise Filtering

Given well-aligned point-to-pixel correspondence, we can easily obtain the instance ID and semantic categories for most projected 3D points within the 2D object masks. This strategy is leveraged by most of the previous methods to obtain 3D mask annotations [64]. However, LiDARs are always equipped much higher than multi-view cameras on autonomous vehicles, leading to serious parallax occlusion issues. As shown in Fig. 3, the 3D points belonging to backgrounds (green) are projected into the pixel regions of the car (orange). Because disparity occlusion commonly arises at regions around the upper edges of foreground objects, we thereby propose to filter out these background points from the foreground pixels with an algorithm akin to a convolution filtering operation.

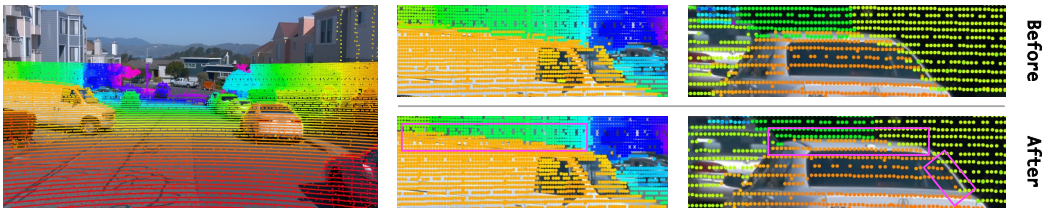


Figure 3: Point clouds are projected into the image plane, and visualized in a color map based on the depth values (**Near** to **Far**). On the right, we compare the effect before (top) and after (bottom) our proposed parallax occlusion. Please zoom in the highlighted pink boxes to see the filtering points.

Algorithm 1: Parallax Occlusion Filtering

Input: projected points $\mathcal{P}^{\mathrm{I}}$, 2D instance segmentation masks $\mathcal{M}$, kernel size k, horizontal and vertical step size s_h, s_v, depth threshold θ, image resolution h, w
Output: accurate object-specific points $\mathcal{P}_i^{\mathrm{L}}$

for $\mathcal{M}_i \leftarrow$ *near to far* **do**
  cnt_h $= 0$, cnt_w $= 0$;
  $\mathcal{P}^{\mathrm{I}} \leftarrow$ `SpatialAligment`$(\mathcal{M}_i, \mathcal{P}^{\mathrm{L}})$;
  while cnt_h $< h$, cnt_w $< w$ **do**
    $p \leftarrow$ `SampleDepthPixel`$(\mathcal{P}^{\mathrm{I}},$ cnt_h, cnt_w, $k)$;
    if $\frac{\max(p)-\min(p)}{\min(p)} > \theta$ **then**
      $p^{\text{near}}, p^{\text{far}} \leftarrow$ `SplitDepthPixel`$(p,\ \theta)$;
      if len$(\mathrm{p}^{near}) > 1$ **then** Rect $\leftarrow$ `LocalRectConstruct()`;
      else Rect $\leftarrow$ `LocalRectConstruct()`;
      if p^{far} *in* Rect **then**
        $\mathcal{P}_i^{\mathrm{L}} \leftarrow$`FilterOut`$(p^{\mathrm{far}})$
      end
    end
    cnt_h+=s_h, cnt_w+=s_w
  end
end

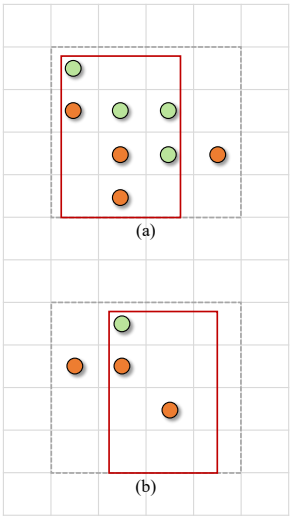


Figure 4: Two cases of constructing the local rectangle regions in our proposed algorithm. Projected points with large depth values p^{far} (orange) inside the local rectangle region will be filtered out.

In specific, we first take a subset of the projected points that fall within a pre-defined filter kernel, and then calculate the maximum depth differences among them. If the depth difference exceeds a threshold θ, we identify all points that surpass θ as p^{far}, and p^{near} otherwise. Then, we will construct a local rectangle region $\boldsymbol{\mathcal{R}}$ based on the numbers of p^{near}.

As illustrated in Fig. 4, there are two cases to be considered: (1) If the number of p^{near} is greater than one, we identify the maximum and minimum pixel coordinates along the horizontal axis among p^{near} to define the left and right boundaries of $\boldsymbol{\mathcal{R}}$, and (2) if there is only one p^{near}, we record its coordinate as the left boundary and assume a pseudo coordinate along the horizontal direction as the right boundary. Meanwhile, the top and bottom boundaries are determined by the minimum vertical pixel coordinate among p^{near} and the maximum vertical coordinate of the location covered by the operation kernel, respectively. Finally, p^{far} that fall within $\boldsymbol{\mathcal{R}}$ will be filtered out as background points to be assigned with relevant semantic categories (*e.g.*, road, sidewalk, wall).

In practice, we execute the filtering operation for each object mask after sorting their depth values in ascending order. The kernel size and stride of this operation kernel can be adjusted to different LiDAR types. Detailed steps are presented in Alg. 1. Note that the projected points may be located in two valid 2D object masks from neighboring views. Thanks to our multi-view consistency design, we could directly combine the object points together with the same instance ID. In addition, we filter out isolated outliers and noise points by clustering the re-projected object points in 3D space with DBSCAN [65] technique.

With the processing of all these methods, we can extract points belonging to foreground objects and assign accurate semantic categories and instance IDs for them, resulting in the final 3D semantic and instance segmentation outputs.

3.3 3D Box Interpretation

In this section, we aim to interpret the precise 3D bounding boxes with the instance segmented points in a human-label-free manner, especially for the foreground objects.

However, super sparse point clouds of objects are very common in driving environments, typically manifesting in two scenarios: (1) LiDARs often struggle to obtain dense scanning results for small

or distant objects, and (2) due to the constraints of fixed configurations, it becomes challenging to capture scans of objects from all views. Hence, it is challenging to precisely characterize the geometry shape even after compensating the object points of the entire sequence. To meet the requirement of deprecating human labels, it is intuitive to firstly complete point clouds from partial inputs.

3.3.1 Point Completion

Inspired by recent remarkable progress in the field of point cloud completion, we design a simple and effective network to precisely capture the structural information of 3D shapes and predict complete point clouds with highly detailed geometries.

The whole network consists of three models, the point encoder, geometry generator, and point generator. Specifically, a PointNet-structure [66] encoder first extracts a shape embedding from the partial point cloud input to capture both local structural details and the global context of the object. To better take advantage of semantic information, we leverage pre-trained CLIP [41] text encoder to generate object category embedding. Then, the geometry generator aims to produce a sparse but complete geometric structure, based on decoding the shape and category embeddings. The final point generator receives the shape embedding, the geometric structure, and the category embedding as input, and generates the dense fine-grained point clouds. All the point cloud modules leverage self-attention layers to adaptively aggregate information and reveal detailed spatial relations among the unordered partial points. Please refer to the Appendix for more details of acquiring partial-complete data pairs and model training process.

3.3.2 Box Interpretation

We first classify the motion state of the objects as static or dynamic based on the segmented points of each object track. For static objects, we transform the object points of each frame to the global coordinate with the pose matrix, and combine them together. We then apply L-Shape fitting to derive an initial 3D box representing the geometric shape. Noisy points and outliers outside the initial box are removed, and we randomly select a set of points with FPS sampling. L-Shape fitting is then performed for these selected points to generate a refined 3D box, which is subsequently transformed back to each frame as the final result.

In contrast to the combination operation for static objects, we process each sample of the dynamic object tracks on a frame-by-frame basis. We first subsample a set of points with FPS sampling and fit the initial 3D box for each object sample from each frame. Based on the distributions of these initial boxes, we generate anchors that stabilize the refined 3D box through L-Shape fitting. Finally, the trajectory is smoothed by linear fitting and Kalman filter in the global coordinate, and the results are then updated to each frame.

3.4 4D Occupancy Flow

Eventually, multi-view images, point clouds, and the generated 3D boxes are all fed into a neural rendering model to reconstruct the 3D scenes, which are used to decode occupancy grids as our 4D occupancy flow output.

In particular, we aim to build a compositional scene representation that models the 3D world including the dynamic objects and static scene, by leveraging StreetSurf [40]. The core is to render the well-suit geometry representation with signed distance functions (SDF), by disentangling a 3D space volume into a static background and a set of foreground objects [57, 67] which are determined by the input 3D boxes. Please refer to the Appendix or the original paper for more details.

With the implicit surface being reconstructed, we obtain a continuous representation of scene geometry that has infinitesimal granularity. Subsequently, we can decode high-resolution occupancy grids out of the reconstructed implicit surface. The semantic and instance information of each grid can still be preserved based on the inside LiDAR points.

4 Experiments

In this section, we first introduce the dataset details and evaluation metrics. We then provide a detailed performance of ZOPP on different perception tasks. The ablation studies and analysis are presented

to convince each component of our entire approach. Please refer to Appendix for detailed quantitative results, more qualitative results, and application experiments.

4.1 Dataset

Following the experimental setting of previous offboard perception methods [33, 35, 34, 36], we conduct extensive experiments on the large-scale Waymo open dataset [68]. The dataset provides 20-second point cloud and 5-view image data for each scene with a sampling frequency at 10Hz. Considering that the environmental conditions would affect the quality of neural rendering (*e.g.*, weather conditions, image blurring), we select a set of sequences from the validation set to conduct all the experiments.

4.2 Main Results

We present a comprehensive evaluation of 3D object detection, 3D segmentation, and occupancy prediction. Note that there are only 5 cameras on WOD, we hence calculate the performance (indicated by †) of each perception task by excluding regions outside the field of view of multiple cameras.

3D Detection As illustrated in Tab. 2, we report the performance of our ZOPP on the validation set of WOD. The Average Precision (AP) and Recall performance are calculated using different matching criteria (IoU and BEV distance). Meanwhile, we compare the performance with several methods across different distance ranges in Tab. 3. As the distance increases, the performance of all methods decreases. Specifically, VoxelRCNN shows a decline in L1 AP of 14.37% and 36.12% for the distance ranges of *30-50m* and *50+m*, compared to *0-30m*. PVRCNN experiences decreases of 15.52% and 37.05%, while our method demonstrates reductions of 16.94% and 29.35%. This improvement can be attributed to our mask tracking module, which effectively utilizes the entire temporal information in the point cloud sequence with generated object IDs. Consequently, our method mitigates the impact of distance more effectively than other onboard methods, particularly at farther ranges. Furthermore, we visualize the 3D object detection results in Fig. 5, where the red and blue boxes are ground-truth and predicted ones, respectively.

Table 2: Verifying 3D object detection ability of our ZOPP on WOD val set. Metrics are 3D AP of L2 difficulties for *Vehicle*, *Pedestrian*, and *Cyclist*.

	Vehicle		*Pedestrian*		*Cyclist*	
Criterion	AP	Recall	AP	Recall	AP	Recall
IoU†	35.6	48.8	34.5	46.7	11.2	22.9
Distance†	48.1	61.6	46.7	58.5	21.8	34.0

Table 3: Comparisons of fully-supervised detectors and human-label-free methods. We re-implement these methods and report the AP performance (IoU criterion) of *Vehicle* within camera FOVs across different distance ranges.

		Total		*0-30m*		*30-50m*		*50+m*	
Method	Training Data	L1	L2	L1	L2	L1	L2	L1	L2
Centerpoint [5]	*train set*	73.04	64.72	88.17	86.81	72.12	66.50	51.24	39.72
VoxelRCNN [69]	*train set*	76.29	67.05	89.27	87.84	76.44	69.68	57.03	44.37
PVRCNN [6]	*train set*	75.53	66.77	89.03	87.63	75.21	68.33	56.04	43.33
DetZero [35]	*train set*	89.49	83.34	96.64	95.90	88.84	84.37	78.32	66.77
SAM3D [38]	-	6.90	5.88	19.51	19.05	0.029	0.026	0.0	0.0
ZOPP† (ours)	-	37.56	35.61	42.31	41.16	35.14	33.86	29.89	28.67

Segmentation We present the results of semantic segmentation and panoptic segmentation for both 2D images and 3D point clouds in Fig. 5. In addition to common objects such as vehicle, the segmentation results for tree, pole, traffic light, and sign, are also impressive. This demonstrates that

we not only retain the dense semantic and instance information from the foundation models, but also establish carefully aligned correspondence by the proposed parallax occlusion and noise filtering. The quantitative results of our ZOPP are shown in Tab. 4, along with the performance of SOTA methods for reference. We achieve comparable performance, particularly on foreground objects (*e.g.*, Vehicle, pedestrian, bicyclist), showcasing the powerful potentials of our point clouds segmentation module. Note that we merge all the categories belonging to *car, truck, bus, other vehicle* together as *Vehicle* (its performance is the average of these four categories). Additionally, categories that cannot be fully recognized are excluded from the results.

Table 4: Comparisons of ZOPP and state-of-the-art LiDAR semantic segmentation methods.

Method	Vehicle	motorcyclist	bicyclist	pedestrian	sign	traffic light	pole	Cons. Cone	bicycle	motorcycle	building	vegetation	tree trunk	curb	road	lane marker	other ground	walkable	sidewalk
P-Transformer [70]	59.7	0.0	67.9	85.5	72.3	36.2	71.4	66.4	58.7	54.3	93.7	90.0	64.7	65.2	90.4	48.2	42.8	74.5	71.7
UniSeg [14]	68.8	0.0	73.2	89.0	75.7	43.3	76.1	70.2	75.5	80.8	95.2	91.0	68.2	68.7	92.6	53.9	48.3	78.8	75.8
ZOPP† (ours)	54.2	-	49.6	77.3	29.7	34.2	51.7	33.1	21.8	35.4	75.5	73.6	-	-	81.8	-	-	-	61.2

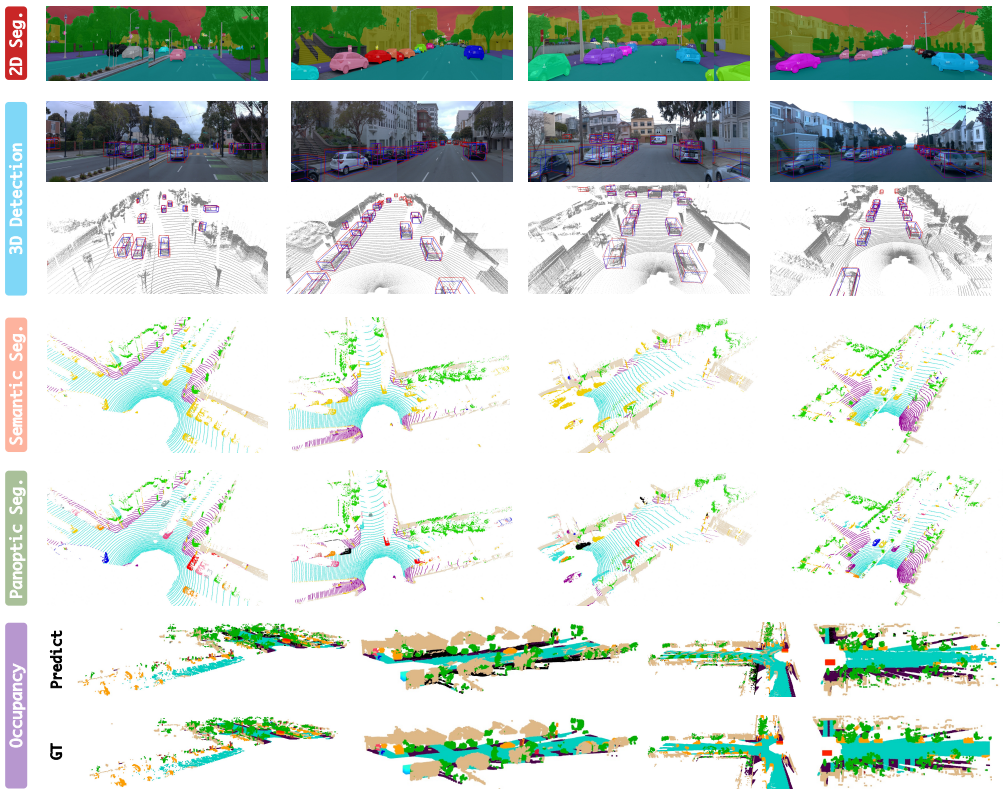


Figure 5: Qualitative results of our proposed ZOPP on various perception tasks in AD scenes, including **2D segmentation**, **3D detection**, **3D semantic segmenation**, **3D panoptic segmentation**, and **occupancy predition**.

Occupancy We leverage Occ3D [28] as ground truth to evaluate the performance. Our approach is based on neural rendering reconstruction, which operates in an offboard fashion, prioritizing reconstruction quality over running efficiency. The performance of other training-based methods is reported as a reference, rather than for detailed comparison. As shown in Tab. 5, we achieve promising results compared to previous methods, especially for slim objects (*e.g.*, traffic light, pole) and flat objects (*e.g.*, road). Object categories not recognized in the selected sequences are excluded (*e.g.*, tree trunk). Notably, ZOPP can outperform a LiDAR-only baseline (supervised training) in

some object categories with limited training samples (*e.g.*, traffic light, sign, cone), by utilizing the zero-shot capabilities of foundation models. Meanwhile, as shown in Fig. 5, the reconstructed scenes are highly complete and spatially coherent, *e.g.*, the predicted road is highly complete and accurately well-defined, demonstrating that our reconstruction method can preserve the detailed 3D geometry effectively.

Table 5: Comparison of 3D occupancy prediction performance.

Method	GO	vehicle	bicyclist	pedestrian	sign	traffic light	pole	Cons. Cone	bicycle	motorcycle	building	vegetation	tree trunk	road	sidewalk	mIoU
TPVFormer [29]	3.89	17.86	12.03	5.67	13.64	8.49	8.90	9.95	14.79	0.32	13.82	11.44	5.8	73.3	51.49	16.76
BEVFormer [8]	3.48	17.18	13.87	5.9	13.84	2.7	9.82	12.2	13.99	0.0	13.38	11.66	6.73	74.97	51.61	16.76
BEVFormer-Fusion	5.11	64.61	52.35	21.52	32.74	17.1	42.62	27.75	13.36	0.05	63.65	60.51	35.64	81.89	66.84	39.05
LiDAR-only	1.01	57.41	35.31	20.33	11.7	13.01	36.21	7.81	0.13	0.0	57.83	54.71	27.07	69.15	54.47	29.74
ZOPP† (ours)	0.08	49.68	10.63	6.44	12.33	21.73	32.75	19.87	9.41	0.07	41.14	46.22	-	69.07	32.34	25.13

5 Conclusion

In this work, we have proposed ZOPP, a novel framework of zero-shot offboard panoptic perception for autonomous driving. Foundation models empower our ZOPP comprehensive capability of language understanding to establish various perception tasks for open-set settings in a zero-shot manner. We enhance SAM-Track to ensure semantic and instance consistency among object mask tracks across multiple views. The proposed parallax occlusion and noise filtering can produce robust 3D semantic and panoptic segmentation results after the well-aligned correspondence between point clouds and multi-view image planes. Equipped with the proposed point completion module, we can generate dense completed points and subsequently interpret precise 3D bounding boxes. These modules cooperate to make the 3D segmentation and detection more accurate and consistent, especially for dynamic foreground objects. Finally, we decode high-quality 4D occupancy by concentrating on the geometry quality and visual appearance with neural rendering reconstruction fashion. Extensive experimental results not only demonstrate that ZOPP substantially advances promising open-set perception results in offboard manner for outdoor AD scenes, but also show a profound significance in industry auto-labeling applications.

6 Limitations and Broader Impacts

While foundation models have endowed our ZOPP with open-set capabilities, the annotated categories in the existing dataset still incorporate expressions that lack universality, which may hinder the effective recognition of similar object categories. Additionally, neural rendering may encounter numerous challenges in street-view scenes, influenced by practice factors (adverse weather conditions, sensor imaging issues). Moreover, ZOPP may raise concerns about data capturing, abuse, privacy, and legal implications in driving surroundings. Nonetheless, ZOPP still offers a high degree of flexibility, allowing for seamless integration with SOTA models to meet diverse application requirements, showcasing resilience and applicability in both industry and daily lives. We believe that advancements in technology and the development of regulatory frameworks can pave the way for unified AD systems.

Acknowledgements

This project is funded in part by National Key R&D Program of China Project 2022ZD0161100, by Shanghai Artificial Intelligence Laboratory (Grant No. 2022ZD0160104), by the Centre for Perceptual and Interactive Intelligence (CPII) Ltd under the Innovation and Technology Commission (ITC)'s InnoHK, by General Research Fund of Hong Kong RGC Project 14204021, by Smart Traffic Fund PSRI/76/2311/PR. Hongsheng Li is a PI of CPII under the InnoHK.

References

[1] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[2] Yan Yan, Yuxing Mao, and Bo Li. Second: Sparsely embedded convolutional detection. In *Sensors*, 2018.

[3] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[4] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[5] Tianwei Yin, Xingyi Zhou, and Philipp Krahenbuhl. Center-based 3d object detection and tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[6] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[7] Shaoshuai Shi, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. From points to parts: 3d object detection from point cloud with part-aware and part-aggregation network. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2020.

[8] 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. *arXiv preprint arXiv:2203.17270*, 2022.

[9] Tao Ma, Zhiwei Zheng, Hongbin Zhou, Xinyu Cai, Xuemeng Yang, Yikang Li, Botian Shi, and Hongsheng Li. Velovox: A low-cost and accurate 4d object detector with single-frame point cloud of livox lidar. In *Proceedings of the IEEE Conference on Robotics and Automation (ICRA)*, pages 1992–1998. IEEE, 2024.

[10] Yeqi Bai, Ben Fei, Youquan Liu, Tao Ma, Yuenan Hou, Botian Shi, and Yikang Li. Rangeperception: taming lidar range view for efficient and accurate 3d object detection. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

[11] Xuesong Chen, Shaoshuai Shi, Benjin Zhu, Ka Chun Cheung, Hang Xu, and Hongsheng Li. Mppnet: Multi-frame feature intertwining with proxy points for 3d temporal object detection. In *European Conference on Computer Vision*, pages 680–697. Springer, 2022.

[12] Xuesong Chen, Shaoshuai Shi, Chao Zhang, Benjin Zhu, Qiang Wang, Ka Chun Cheung, Simon See, and Hongsheng Li. Trajectoryformer: 3d object tracking transformer with predictive trajectory hypotheses. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18527–18536, 2023.

[13] Xin Li, Tao Ma, Yuenan Hou, Botian Shi, Yuchen Yang, Youquan Liu, Xingjiao Wu, Qin Chen, Yikang Li, Yu Qiao, and Liang He. Logonet: Towards accurate 3d object detection with local-to-global cross-modal fusion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[14] Youquan Liu, Runnan Chen, Xin Li, Lingdong Kong, Yuchen Yang, Zhaoyang Xia, Yeqi Bai, Xinge Zhu, Yuexin Ma, Yikang Li, Yu Qiao, and Yuenan Hou. Uniseg: A unified multi-modal lidar segmentation network and the openpcseg codebase. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2023.

[15] Yingjie Cai, Xuesong Chen, Chao Zhang, Kwan-Yee Lin, Xiaogang Wang, and Hongsheng Li. Semantic scene completion via integrating instances and scene in-the-loop. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 324–333, 2021.

[16] Tao Ma, Zhizheng Liu, Guohang Yan, and Yikang Li. Crlf: Automatic calibration and refinement based on line feature for lidar and camera in road scenes. *arXiv preprint arXiv:2103.04558*, 2021.

[17] Tao Ma, Zhizheng Liu, and Yikang Li. Perception entropy: A metric for multiple sensors configuration evaluation and design. *arXiv preprint arXiv:2104.06615*, 2021.

[18] Licheng Wen, Daocheng Fu, Xin Li, Xinyu Cai, Tao Ma, Pinlong Cai, Min Dou, Botian Shi, Liang He, and Yu Qiao. Dilu: A knowledge-driven approach to autonomous driving with large language models, 2023.

[19] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023.

[20] Jianan Li and Qiulei Dong. Open-set semantic segmentation for point clouds via adversarial prototype framework. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[21] Yuhang Lu, Qi Jiang, Runnan Chen, Yuenan Hou, Xinge Zhu, and Yuexin Ma. See more and know more: Zero-shot point cloud segmentation via multi-modal visual data. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2023.

[22] Renrui Zhang, Ziyu Guo, Wei Zhang, Kunchang Li, Xupeng Miao, Bin Cui, Yu Qiao, Peng Gao, and Hongsheng Li. Pointclip: Point cloud understanding by clip. *arXiv preprint arXiv:2112.02413*, 2021.

[23] Jun Cen, Peng Yun, Junhao Cai, Michael Yu Wang, and Ming Liu. Open-set 3d object detection. *arXiv preprint arXiv:2112.01135*, 2021.

[24] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[25] Dongmei Zhang, Chang Li, Ray Zhang, Shenghao Xie, Wei Xue, Xiaodong Xie, and Shanghang Zhang. Fm-ov3d: Foundation model-based cross-modal knowledge blending for open-vocabulary 3d detection. *arXiv preprint arXiv:2312.14465*, 2023.

[26] 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 (CVPR)*, 2023.

[27] Chonghao Sima, Wenwen Tong, Tai Wang, Li Chen, Silei Wu, Hanming Deng, Yi Gu, Lewei Lu, Ping Luo, Dahua Lin, and Hongyang Li. Scene as occupancy. 2023.

[28] 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.

[29] Yuanhui Huang, Wenzhao Zheng, Yunpeng Zhang, Jie Zhou, and Jiwen Lu. Tri-perspective view for vision-based 3d semantic occupancy prediction. *arXiv preprint arXiv:2302.07817*, 2023.

[30] 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:2302.13540*, 2023.

[31] Jiakang Yuan, Bo Zhang, Kaixiong Gong, Xiangyu Yue, Botian Shi, Yu Qiao, and Tao Chen. Reg-tta3d: Better regression makes better test-time adaptive 3d object detection. In *European conference on computer vision*, pages 197—-213. Springer, 2024.

[32] Xiangchao Yan, Runjian Chen, Bo Zhang, Jiakang Yuan, Xinyu Cai, Botian Shi, Wenqi Shao, Junchi Yan, Ping Luo, and Yu Qiao. Spot: Scalable 3d pre-training via occupancy prediction for autonomous driving. *arXiv preprint arXiv:2309.10527*, 2023.

[33] Charles R. Qi, Yin Zhou, Mahyar Najibi, Pei Sun, Khoa Vo, Boyang Deng, and Dragomir Anguelov. Offboard 3d object detection from point cloud sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[34] Lue Fan, Yuxue Yang, Yiming Mao, Feng Wang, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Once detected, never lost: Surpassing human performance in offline lidar based 3d object detection. In *ICCV*, 2023.

[35] Tao Ma, Xuemeng Yang, Hongbin Zhou, Xin Li, Botian Shi, Junjie Liu, Yuchen Yang, Zhizheng Liu, Liang He, Yu Qiao, Yikang Li, and Hongsheng Li. Detzero: Rethinking offboard 3d object detection with long-term sequential point clouds. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2023.

[36] Anqi Joyce Yang, Sergio Casas, Nikita Dvornik, Sean Segal, Yuwen Xiong, Jordan Sir Kwang Hu, Carter Fang, and Raquel Urtasun. Labelformer: Object trajectory refinement for offboard perception from lidar point clouds. *arxiv preprint*, 2023.

[37] Dongqiangzi Ye, Zixiang Zhou, Weijia Chen, Yufei Xie, Yu Wang, Panqu Wang, and Hassan Foroosh. Lidarmultinet: Towards a unified multi-task network for lidar perception. *arXiv preprint arXiv:2209.09385*, 2022.

[38] Dingyuan Zhang, Dingkang Liang, Hongcheng Yang, Zhikang Zou, Xiaoqing Ye, Zhe Liu, and Xiang Bai. Sam3d: Zero-shot 3d object detection via segment anything model. *arxiv preprint*, 2023.

[39] Yangming Cheng, Liulei Li, Yuanyou Xu, Xiaodi Li, Zongxin Yang, Wenguan Wang, and Yi Yang. Segment and track anything. *arXiv preprint arXiv:2305.06558*, 2023.

[40] Jianfei Guo, Nianchen Deng, Xinyang Li, Yeqi Bai, Botian Shi, Chiyu Wang, Chenjing Ding, Dongliang Wang, and Yikang Li. Streetsurf: Extending multi-view implicit surface reconstruction to street views. *arXiv preprint arXiv:2306.04988*, 2023.

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[42] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021.

[43] Renrui Zhang Teli Ma Rongyao Fang Yongfeng Zhang Hongsheng Li Yu Qiao Peng Gao, Shijie Geng. Clip-adapter: Better vision-language models with feature adapters. *arXiv preprint arXiv:2110.04544*, 2021.

[44] Lewei Yao, Jianhua Han, Youpeng Wen, Xiaodan Liang, Dan Xu, Wei Zhang, Zhenguo Li, Chunjing Xu, and Hang Xu. Detclip: Dictionary-enriched visual-concept paralleled pre-training for open-world detection. *arXiv preprint arXiv:2209.09407*, 2022.

[45] Jie Hong, Shi Qiu, Weihao Li, Saeed Anwar, Mehrtash Harandi, Nick Barnes, and Lars Petersson. Pointcam: Cut-and-mix for open-set point cloud learning. *arXiv preprint arXiv:2212.02011*, 2022.

[46] Antonio Alliegro, Francesco Cappio Borlino, and Tatiana Tommasi. 3dos: Towards 3d open set learning – benchmarking and understanding semantic novelty detection on point clouds. *arXiv preprint arXiv:2207.11554*, 2022.

[47] Yuheng Lu, Chenfeng Xu, Xiaobao Wei, Xiaodong Xie, Masayoshi Tomizuka, Kurt Keutzer, and Shanghang Zhang. Open-vocabulary point-cloud object detection without 3d annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2023.

[48] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[49] Fangzhou Hong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. Lidar-based panoptic segmentation via dynamic shifting network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.

[50] Jinke Li, Xiao He, Yang Wen, Yuan Gao, Xiaoqiang Cheng, and Dan Zhang. Panoptic-phnet: Towards real-time and high-precision lidar panoptic segmentation via clustering pseudo heatmap. *arXiv preprint arXiv:2205.07002*, 2022.

[51] Zhuangwei Zhuang, Rong Li, Kui Jia, Qicheng Wang, Yuanqing Li, and Mingkui Tan. Perception-aware multi-sensor fusion for 3d lidar semantic segmentation. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2021.

[52] Georg Krispel, Michael Opitz, Georg Waltner, Horst Possegger, and Horst Bischof. Fuseseg: Lidar point cloud segmentation fusing multi-modal data. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, 2020.

[53] Khaled El Madawy, Hazem Rashed, Ahmad El Sallab, Omar Nasr, Hanan Kamel, and Senthil Yogamani. Rgb and lidar fusion based 3d semantic segmentation for autonomous driving. In *Proceedings of the IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019.

[54] Ali Athar, Enxu Li, Sergio Casas, and Raquel Urtasun. 4d-former: Multimodal 4d panoptic segmentation. *arXiv preprint arXiv:2311.01520*, 2023.

[55] Mehmet Aygün, Aljoša Ošep, Mark Weber, Maxim Maximov, Cyrill Stachniss, Jens Behley, and Laura Leal-Taixé. 4d panoptic lidar segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[56] Minghan Zhu, Shizhong Han, Hong Cai, Shubhankar Borse, Maani Ghaffari, and Fatih Porikli. 4d panoptic segmentation as invariant and equivariant field prediction. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2023.

[57] Ze Yang, Yun Chen, Jingkang Wang, Sivabalan Manivasagam, Wei-Chiu Ma, Anqi Joyce Yang, and Raquel Urtasun. Unisim: A neural closed-loop sensor simulator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[58] Bin Yang, Min Bai, Ming Liang, Wenyuan Zeng, and Raquel Urtasun. Auto4d: Learning to label 4d objects from sequential point clouds. *arxiv preprint*, 2021.

[59] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3d multi-object tracking: A baseline and new evaluation metrics. In *Proceedings of the IEEE Conference on Intelligent Robots and Systems (IROS)*, pages 10359–10366. IEEE, 2020.

[60] Ziqi Pang, Zhichao Li, and Naiyan Wang. Simpletrack: Understanding and rethinking 3d multi-object tracking. *arxiv preprint*, 2021.

[61] Qitai Wang, Yuntao Chen, Ziqi Pang, Naiyan Wang, and Zhaoxiang Zhang. Immortal tracker: Tracklet never dies. *arxiv preprint*, 2021.

[62] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. *arxiv preprint*, 2023.

[63] Zongxin Yang and Yi Yang. Decoupling features in hierarchical propagation for video object segmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[64] Yijie Zhou, Likun Cai, Xianhui Cheng, Zhongxue Gan, Xiangyang Xue, and Wenchao Ding. Openannotate3d: Open-vocabulary auto-labeling system for multi-modal 3d data. *arXiv preprint arXiv:2310.13398*, 2023.

[65] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of International Conference on Knowledge Discovery and Data Mining*, 1996.

[66] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2017.

[67] Xiaoyu Zhou, Zhiwei Lin, Xiaojun Shan, Yongtao Wang, Deqing Sun, and Ming-Hsuan Yang. Drivinggaussian: Composite gaussian splatting for surrounding dynamic autonomous driving scenes. *arXiv preprint arXiv:2312.07920*, 2023.

[68] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[69] Jiajun Deng, Shaoshuai Shi, Peiwei Li, Wengang Zhou, Yanyong Zhang, and Houqiang Li. Voxel r-cnn: Towards high performance voxel-based 3d object detection. *arXiv preprint arXiv:2012.15712*, 2020.

[70] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip Torr, and Vladlen Koltun. Point transformer. *arXiv preprint arXiv:2012.09164*, 2020.

[71] 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. *arXiv preprint arXiv:2103.14030*, 2021.

[72] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[73] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*, 2019.

Appendix

A Overview

This document is the supplementary material of our ZOPP. We provide more details of models, experiments and analysis results in this document.

Sec. B.1 provides the details of acquiring partial-complete data pairs and model training process. Sec. B.2 introduces more details about the neural rendering reconstruction method. Sec. C shows the implementation details of the network setting and training process. Sec. D provides more experiment results and analyses in detail. In specific, we compare the distribution of predicted bounding boxes with different distance thresholds in Sec. D.1. The effectiveness of our method to overcome the influence of occlusion is shown in Sec. D.2. The qualitative and quantitative results to show the effectiveness of parallax occlusion and noise filtering are presented in Sec. D.3. And the effectiveness of point completion in our whole pipeline is shown in Sec. D.4. We also illustrate the open-set detection capabilities in Sec. D.5, and the failure pattern analysis in Sec. D.6.

B Method Details

B.1 Point Completion

We implement an automatic algorithm to determine whether the extracted object points are completed or not, selectively filtering out those with intact shapes. The automatic selection is mainly based on the ratio of occupied grids. During the training process, we randomly remove part of the points to generate the partial inputs, based on the constraints of geometric projection principles, inducing realistic structural incompleteness data pairs. For the sparse but complete geometric points, we randomly sample a set of points based on the FPS sampling strategy to promise the geometric structure. We also use random rotations sampled from a uniform distribution $[-\pi/2, \pi/2]$ and random linear transformations sampled from a standard Gaussian distribution to translate the point coordinates. Chamfer distance is utilized as the metric distance of points to formulate the supervision between partial input and real dense complete object point clouds. We don't require any human labels in this procedure.

B.2 Neural Rendering

We aim to build a compositional scene representation that models the 3D world including the dynamic objects and static scene, by leveraging a neural rendering method [40]. A 3D space volume is first defined over the entire scene. The volume consists of a static background and a set of dynamic objects, determined by the input 3D boxes. Such that separate neural feature fields and feature grids can be used to model them, respectively.

The static background is delimited into three parts, close-range, distant view, and sky. This design can well-suit geometry representation by signed distance functions (SDF). Then, three neural rendering models are employed for these three parts to jointly render a differential pixel color by querying samples for each ray. The queried samples are combined from near to far for the subsequent volume rendering.

The dynamic foreground objects are transformed to their local box coordinates (centroid of the box), and their feature grids are at the world coordinate to compose with the background. allowing us to disentangle the 3D motion of each object and focus on representing shape and appearance [57, 67].

C Implementation Details

For parallax occlusion filtering, the kernel size is 15, the steps in horizontal and vertical directions are 10 and 5 respectively. The depth ratio threshold is set to 0.25. For Grounding-DINO, we keep the same setting to leverage pre-trained Swin-L [71] as image backbone, and BERT-base [72] from Hugging Face [73] as text backbones. The point completion network is tuned on WOD training set, with our proposed data preparation mentioned before. The whole training includes 100 epochs because of the limited amounts of objects, while the learning rate is initialized to $1e-4$ and decayed by

0.7 every 40 epochs with Adam optimizer. The batch size is set to 32. For occupancy reconstruction, we train the model for around 15000 iterations and add additional cross-entropy supervision after 5000 iterations, compared to the original version [40]. In one batch, we use 8192 rays. The entire pipeline does not rely on too many computation resources, the point completion module and the reconstruction module need to train the network. We utilize four NVIDIA A100 to accelerate the reconstruction with multi-processing settings.

D More Experiments and Ablations

D.1 3D Detection Analysis

In this section, we show a detailed analysis of our 3D detection performance shown in Tab. 6. Specifically, we first match the predicted boxes with the ground-truth boxes that have the smallest center distance up to a certain threshold. Then the performance (Recall) is the statistics for the matched part of all ground-truth boxes in the FOV of cameras. The final results are averaged over the matching thresholds of $(0.5, 1, 2, 4)$ meters. We can draw several conclusions:

1) Over 50% of objects (*Vehicle* and *Pedestrian*) are recalled in the 1m range compared to ground truths, showing the accuracy of our multi-modal spatial alignment and parallax occlusion and noise filtering.
2) Almost 70% to 80% of objects are recalled in the 4m range, demonstrating vision foundation models possess sufficient capability to provide detection proposals, and the majority of objects not recalled are primarily due to occlusion (cameras are installed at a lower position relative to the LiDAR).
3) There exists a distance gap between predictions and ground-truths (almost 20% of objects are in the range of 1$\sim$4m), which is mainly due to the processing pipeline of our box interpretation. Previous box prediction of 3D detection models are always separately to predict the components, *e.g.*, the CenterHead of CenterPoint predicts the box center, geometry size, and heading direction with different network layers. Different from them, our box interpretation would first predict the geometry size, and then calculate the center with the half of length, height, and width. Therefore, if the geometry size is inaccurate, the box center will also not be precise.

Table 6: Detailed performance of 3D object detection on WOD val set. Metrics are Recall (with BEV distance criterion) of L2 difficulties for *Vehicle*, *Pedestrian*, and *Cyclist*. All results are in the FOV of camera views.

	Avg.	0.5m	1m	2m	4m
Vehicle	61.6	39.2	54.9	70.7	81.6
Pedestrian	58.5	42.5	57.2	64.5	70.1
Cyclist	34.0	25.4	32.8	37.5	40.4

D.2 Performance of Occlusions

We report the L1 AP performance of the overall and the occlusion part on WOD validation set to compare with other methods. The occlusion levels are defined based on whether the objects are obscured in the image perspective, which are provided by WOD.

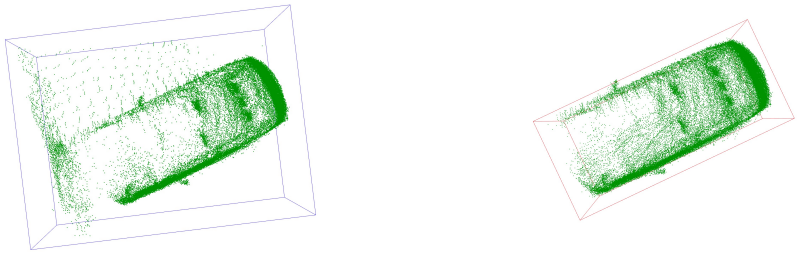
As shown in Tab. 7, compared to the overall performance, the occlusion part of CenerPoint, VoxelR-CNN and PVRCNN exhibit decreases of 18.81%, 18.78% and 19.07% respectively, while SAM3D shows a decrease of 31.30%. In contrast, our method demonstrates a decrease of only 11.02%. This improvement is attributed to our mask tracking module, which effectively leverages temporal context to mitigate the influence of occlusion.

D.3 Parallax Occlusion and Noise Filtering

We present the effectiveness of our parallax occlusion and noise filtering module by comparing the box interpretation results before and after the filtering operation. As shown in Fig. 6, if we assign

Table 7: Performance comparison of occlusion on WOD val set. Metrics are L1 AP (with IoU criterion) for *Vehicle*. All results are in the FOV of camera views.

	Training Data	Overall	Occluded
CenterPoint	*train set*	73.04	59.30
VoxelRCNN	*train set*	76.29	61.96
PVRCNN	*train set*	75.53	61.13
SAM3D	-	6.90	4.74
ZOPP (ours)	-	37.56	33.42



(a) (b)

Figure 6: (a) Before the parallax occlusion and noise filtering, our box interpretation would produce inaccurate box dimensions based on the wrong object points. (b) After filtering, we will output 3D boxes with precise dimensions.

the instance and category information to the points that are directly projected to the image plane, some background points would be classified as the foreground objects, resulting in incorrect box interpretation. Our method could filter out the background and noise points, which significantly reduces the burden of our box interpretation module.

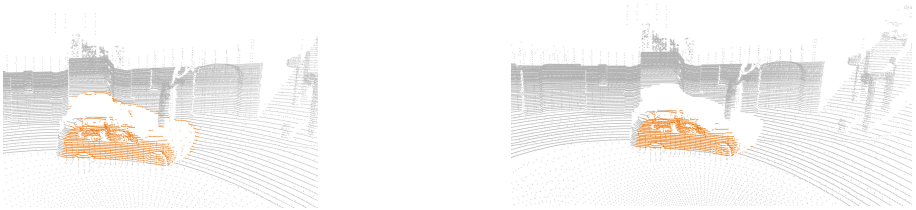
In addition, we evaluate its effect on the segmentation task. We first report the performance of semantic segmentation in Tab. 8, which shows a significant quantitative improvement for all foreground objects and backgrounds. For the objects that always appear at higher altitudes (*e.g.*, sign, traffic light), the parallax occlusion issue is not serious, hence the performance is maintained the same after the filtering module. We also show the visualization comparison of segmentation results in Fig. 7. As we can see, the background points may be located in the boundary regions of the foreground car, hence the corresponding categories are all incorrectly assigned as car. Our filtering module can filter out these points and better align the relation between LiDAR points and image pixels, producing more accurate segmentation results.

Table 8: Comparisons of ZOPP on semantic segmentation before and after the proposed parallax occlusion and noise filtering.

Method	Vehicle	motorcyclist	bicyclist	pedestrian	sign	traffic light	pole	Cons. Cone	bicycle	motorcycle	building	vegetation	tree trunk	curb	road	lane marker	other ground	walkable	sidewalk
ZOPP† (before)	51.6	-	47.7	76.1	29.5	34.0	49.6	32.7	21.2	34.2	73.8	72.3	-	-	80.5	-	-	-	60.4
ZOPP† (after)	54.2	-	49.6	77.3	29.7	34.2	51.7	33.1	21.8	35.4	75.5	73.6	-	-	81.8	-	-	-	61.2

D.4 Point completion

We first evaluate the performance of point completion, we visualize the generated dense and completed object points shown in Fig. 8. The input object points are always sparse (1st, 5th columns) and uncompleted (2nd, 3rd, 4th columns), *e.g.*, the bus in the 3rd column only has points at the top of the



(a) (b)

Figure 7: (a) Before the parallax occlusion and noise filtering, we would assign the instance ID or semantic category of foreground objects to background points. (b) After the filtering operation, the segmentation results would be more accurate.

side surface. As a comparison, the generated dense points contain much more geometric structures, which would contribute much to interpreting precise 3D bounding boxes. The high-quality results can also be used for generative assets modeling in simulation applications.

Afterward, to better evaluate the effectiveness of point completion in our pipeline, we visualize the interpreted 3D bounding boxes based on the object points processed before and after the point completion in Fig. 9. It is crucial to first generate complete point clouds for our human-label-free box interpretation module, resulting in accurate geometric sizes (length, width, height) prediction.

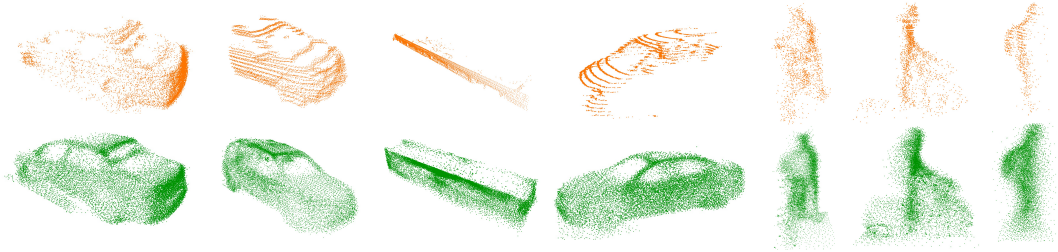
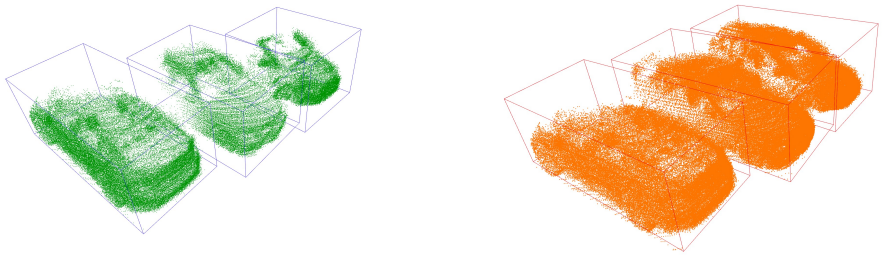


Figure 8: Visual comparisons of point cloud completion. Compared with the sparse inputs (**Top**), we can produce fine-grained geometric structures of dense point clouds (**Bottom**).



(a) (b)

Figure 9: (a) The object points are always sparse and partial, which would lead to inaccurate box interpretation. (b) After point completion for each object, we will predict 3D boxes with precise dimensions.

Furthermore, we report the quantitative results that would reflect the improvements of both the filtering and completion modules. We calculate the Recall performance based on IoU criterion, which considers the accuracy of box shapes. As shown in Tab. 9, after the point completion process, the Recall is gained with 26.2, 12.5, and 11.4 points on the three categories. Because vehicles are always larger than the other two categories, it is more likely to produce sparse and incomplete point clouds. So, our completion module shows an impressive effect for our 3D box interpretation module.

Table 9: Verifying the effect of parallax noise filtering and point completion for 3D bounding box interpretation on WOD val set. Metrics are Recall of L2 difficulties for *Vehicle*, *Pedestrian*, and *Cyclist* with IoU criterion. The results are in the FOV of the cameras.

	Vehicle	*Pedestrian*	*Cyclist*
Before	22.6	34.2	11.5
After	48.8	46.7	22.9

D.5 Open-set 3D Detection

As shown in Fig. 10, our ZOPP can output the open-set 3D detection results of traffic sign and traffic light (represented with red color bounding boxes).

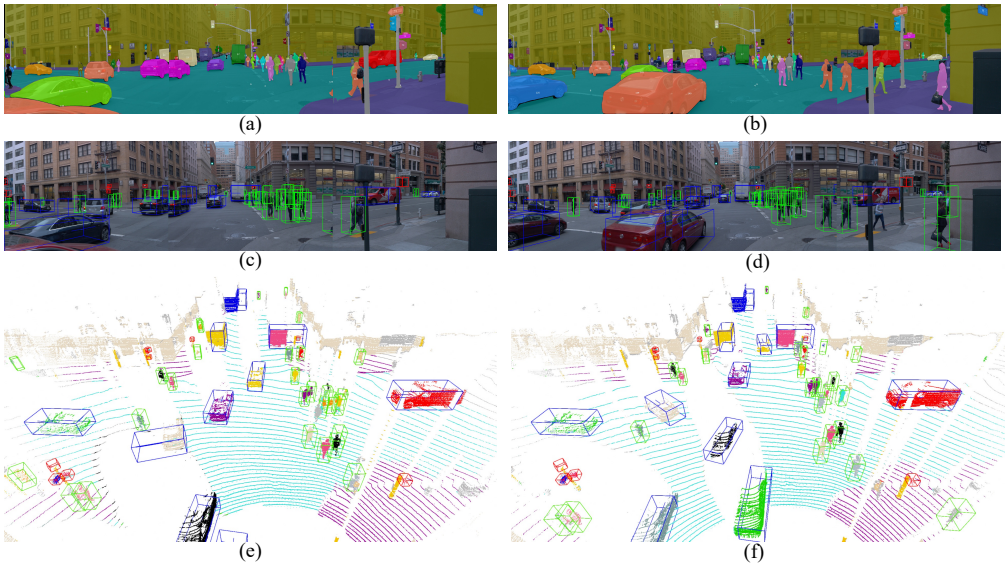


Figure 10: The open-set detection results of ZOPP on WOD in both 2D and 3D spaces for consecutive frames.

D.6 Failure Pattern Analysis

We have briefly summarized some representative challenging scenarios in Sec. 6 of our main contents. Firstly, our method would fail to effectively recognize similar object categories (e.g., construction vehicle, truck, trailer) and some uncommon object categories (e.g., tree trunk, lane marker) with the foundation models (Grounding-DINO). Since this is the first stage of our entire method, it will result in subsequent stages lacking the output of corresponding perception results, such as 3D segmentation and occupancy prediction. Secondly, neural rendering methods may encounter numerous challenges in street-view scenes, constrained by practice factors (adverse weather conditions, sensor imaging issues), such as camera overexposure. Our occupancy decoding will fail in these scenarios where it is impossible to generate geometrically plausible 3D reconstructions. Please refer to Fig. 11 for qualitative visualizations.

(a) (b)

(c) (d)

Figure 11: The illustration of the failure cases. It indicates that the image data are influenced by the lighting conditions at night (a), rainy weather conditions (a), and the camera's overexposure condition (c). Then we could not generate accurate detection and segmentation results (b), and reconstruction with lower quality (d).

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The motivations and contributions are well depicted and summarized in the abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations of the work are performed in Section 6.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [NA]

Justification: No theoretical results are included.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The implementation of our method and the dataset details are clearly and fully presented in the main content and the Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The open access to the code is provided in Abstract.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The full details are presented in Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: Our method mainly focuses on offboard perception tasks in autonomous driving, which naturally includes several test time augmentation and ensemble techniques with multiple inference times.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).

- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The information about computation resources is illustrated in the implementation details of Appendix.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We make sure that the research conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed the potential societal impacts in Section 6.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have cited the original paper that produced the dataset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: This paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.