3D Open-Vocabulary Panoptic Segmentation with 2D-3D Vision-Language Distillation

Zihao Xiao[1*], Longlong Jing[2], Shangxuan Wu[2], Alex Zihao Zhu[2], Jingwei Ji[2], Chiyu Max Jiang[2], Wei-Chih Hung[2], Thomas Funkhouser[3], Weicheng Kuo[4], Anelia Angelova[4], Yin Zhou[2], and Shiwei Sheng[2*]

[1] Johns Hopkins University, [2] Waymo, [3] Google Research, [4] Google DeepMind

Abstract. 3D panoptic segmentation is a challenging perception task, especially in autonomous driving. It aims to predict both semantic and instance annotations for 3D points in a scene. Although prior 3D panoptic segmentation approaches have achieved great performance on closed-set benchmarks, generalizing these approaches to unseen *things* and unseen *stuff* categories remains an open problem. For unseen object categories, 2D open-vocabulary segmentation has achieved promising results that solely rely on frozen CLIP backbones and ensembling multiple classification outputs. However, we find that simply extending these 2D models to 3D does not guarantee good performance due to poor per-mask classification quality, especially for novel *stuff* categories. In this paper, we propose the first method to tackle 3D open-vocabulary panoptic segmentation. Our model takes advantage of the fusion between learnable LiDAR features and dense frozen vision CLIP features, using a single classification head to make predictions for both base and novel classes. To further improve the classification performance on novel classes and leverage the CLIP model, we propose two novel loss functions: object-level distillation loss and voxel-level distillation loss. Our experiments on the nuScenes and SemanticKITTI datasets show that our method outperforms the strong baseline by a large margin.

Keywords: Autonomous driving · 3D panoptic segmentation · Vision-language

1 Introduction

3D panoptic segmentation is a crucial task in computer vision with many real-world applications, most notably in autonomous driving. It combines 3D semantic and instance segmentation to produce per-point predictions for two different types of objects: *things* (*e.g.*, car) and *stuff* (*e.g.*, road). To date, there has been significant progress in 3D panoptic segmentation [27, 40, 42, 47, 52, 58]. Most recently, methods such as [47] produce panoptic segmentation predictions directly from point clouds by leveraging learned queries to represent objects and

∗ Work done while at Waymo

Transformer-based [45] architectures [2, 4] to perform the modeling. However, existing models only predict panoptic segmentation results for a closed-set of objects. They fail to create predictions for the majority of unseen object categories in the scene, hindering the application of these algorithms to real-world scenarios, especially for autonomous driving. In this work, we focus on segmenting unseen *things* and unseen *stuff* objects in autonomous driving scenarios. We follow [10, 53] and develop models under the open-vocabulary setting: we divide the object categories into base (seen) categories and novel (unseen) categories, and evaluate models that are only trained on base categories.

Such open-world computer vision tasks [3] benefit from the recent advancements in vision-language (V-L) models [22, 39]. In 2D vision, there are many successful methods in open-vocabulary object detection [12, 15, 24] and segmentation [11,50,54]. These methods make predictions in a shared image-text embedding space, where predictions for unseen categories are produced by comparing the similarity of an object with the text embedding of the category. However, these methods are only possible due to the vast amounts of paired image-text data available, making it difficult to train similar models for 3D data.

Instead, researchers have continued to leverage the effectiveness of these 2D vision-language models for 3D with the help of pixel-point correspondences by running inference on 2D images and then aligning with the 3D features. These methods have achieved promising results on open-vocabulary semantic segmentation [10,35,53,55] and instance segmentation [10,43,53], individually. However, there are no methods that address the problem of 3D open-vocabulary panoptic segmentation, *i.e.*, addressing both open-vocabulary semantic segmentation and open-vocabulary instance segmentation at the same time. The challenge lies in how to handle segmentation for novel *things* and *stuff* objects simultaneously.

3D open-vocabulary panoptic segmentation is a challenging problem, due to both the significant domain gaps between the camera and LiDAR modalities and unsolved problems in open-vocabulary segmentation. Many existing open-vocabulary works rely on similarities between text embeddings of class names and pre-trained V-L features to obtain associations between predictions and classes [35, 43, 55]. However, while projecting 2D V-L features to 3D can account for a large part of the scene, there are often many points unaccounted for due to unmatched pixel/point distributions and differing fields of view between sensors. Some 3D open-vocabulary works [10,53] apply contrastive learning to obtain better association between language and points, but they require extra captioning models and do not address the difficulties of detecting novel *stuff* classes.

In this work, we aim to address these two issues with a novel architecture for 3D open-vocabulary panoptic segmentation. Building on existing 3D closed-set panoptic segmentation methods, we train a learned LiDAR feature encoder in parallel with a frozen, pre-trained camera CLIP model. By fusing the 3D LiDAR features with the 2D CLIP features, our model is able to learn rich features throughout the entire 3D sensing volume, even if there are no camera features in certain regions. In addition, we apply a pair of novel distillation losses that allow the 3D encoder to learn both object-level and voxel-level features which

live inside the CLIP feature space. This provides a learned module in 3D space which can directly be compared with text embeddings. These losses also provide useful training supervision to unknown parts of the scene where there would otherwise be no loss gradient.

With the proposed model and loss functions, our method significantly outperforms the strong baseline on multiple datasets. Our contributions are summarized as follows:

- We present the first approach for 3D open-vocabulary panoptic segmentation in autonomous driving.
- We propose two novel loss functions, object-level distillation loss and voxel-level distillation loss to help segment novel *things* and novel *stuff* objects.
- We experimentally show that our proposed method significantly outperforms that strong baseline model on both nuScenes and SemanticKITTI datasets.

2 Related Work

This work is closely related to 3D panoptic segmentation, 2D open-vocabulary segmentation, and 3D open-vocabulary segmentation.

3D panoptic segmentation. The goal of 3D panoptic segmentation is to group 3D points according to their semantics and identities. This is a challenging task and relies on a good representation of the 3D data [1, 20, 36, 37, 44, 46, 48]. Most panoptic segmentation models have separate branches for instance segmentation and semantic segmentation [19, 27, 44, 58]. By following DETR [5], the recently proposed P3Former [47] uses learnable queries and a transformer architecture to obtain state-of-the-art performance on multiple panoptic segmentation benchmarks. Although those closed-set methods achieve incredible results, they cannot predict the labels and masks for novel classes.

2D open-vocabulary segmentation. 2D open-vocabulary segmentation aims to group image pixels according to their semantics or identities for base (seen) or novel (unseen) categories. The prediction on novel categories is usually done by leveraging large V-L models [22,39]. There are many works that focus on open vocabulary semantic segmentation [14, 17, 26, 29, 31, 34, 49, 51, 56, 57, 59]. Some work has also explored open-vocabulary panoptic segmentation [11, 38, 50]. Recently, FC-CLIP [54] proposes a single-stage framework based on a frozen convolutional CLIP backbone [21, 32, 39] for 2D open-vocabulary panoptic segmentation that achieves state-of-the-art performance. However, due to the camera-LiDAR domain gap, we show that simply extending it to 3D leads to poor performance.

3D open-vocabulary segmentation. 3D open-vocabulary segmentation is less explored due to the lack of 3D point-to-text association. One common practice is to utilize V-L models and use 2D-3D pairings to obtain rich, structured information in 3D [7, 8, 10, 16, 18, 35, 41, 43, 53, 55]. Notably, CLIP2Scene [7] proposes a semantic-driven cross-modal contrastive learning framework. PLA [10] leverages images as a bridge and builds hierarchical 3D-caption pairs for contrastive learning. OpenScene [35] extracts per-pixel CLIP features using a pre-trained V-L model [14, 26] then derives dense 3D features by projecting 3D points onto

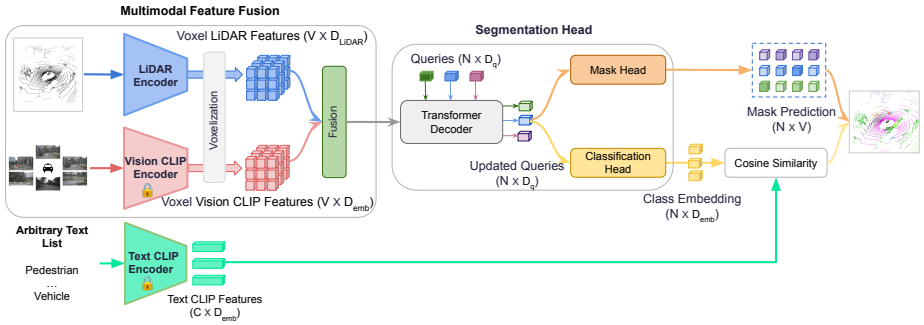

Fig. 1: Overview of our method. Given a LiDAR point cloud and the corresponding camera images, LiDAR features are extracted with a learnable LiDAR encoder, while vision features are extracted by a frozen CLIP vision model. The extracted LiDAR features and the frozen CLIP vision features are then fused and fed to a query-based transformer model to predict instance masks and semantic classes.

image planes. One concurrent work, RegionPLC [53], utilizes regional visual prompts to create dense captions and perform point-discriminative contrastive learning, which is used for semantic segmentation or instance segmentation, individually. In contrast, our work does not rely on any captioning model or extra contrastive learning, but only depends on pre-trained CLIP features. Our model also handles semantic segmentation and instance segmentation simultaneously.

3 Method

This section is organized as follows. First, we define the 3D open-vocabulary panoptic segmentation task. Then we provide detailed descriptions of the model architecture as well as the proposed loss functions. The overview of our method is presented in Fig. 1, and the two proposed loss functions are illustrated in Fig. 2 (a) and Fig. 2 (b).

3.1 Problem Definition

In 3D panoptic segmentation, the goal is to annotate every point in a point cloud. For *stuff* classes, (*e.g.* road, vegetation), a category label is assigned according to its semantics. For *things* classes (*e.g.* cars, pedestrians), an instance label is assigned to an object in addition to its semantic label.

In open-vocabulary panoptic segmentation, the models are trained on C_B base(seen) categories. At test time, besides these C_B base categories, the data will contain C_N novel(unseen) categories. Following the settings of prior work [15, 24, 54], we assume the availability of the name of the novel categories during inference, but the novel categories are not present in the training data and their names are not known. Note that we do not apply any prompt engineering, as

this is not the focus of this paper. We follow OpenScene [35] to obtain the CLIP text embedding for each category.

3.2 3D Open-Vocabulary Panoptic Segmentation

Most of the previous 3D open-vocabulary works only address semantic segmentation [7, 8, 10, 16, 18, 35, 41, 53, 55] or instance segmentation [43, 53] separately, and there is no existing work for the 3D open-vocabulary panoptic segmentation task, which handles novel *things* and novel *stuff* objects simultaneously. A natural idea would be extending the 2D open vocabulary segmentation methods to build the 3D counterpart. We start with P3Former [47], a state-of-the-art transformer-based 3D closed-set panoptic segmentation model, and add the essential components to support open-vocabulary capability by following FC-CLIP [54], a 2D open-vocabulary segmentation model that achieves state-of-the-art performance on multiple datasets. However, we found that this simple extension leads to poor performance in our experiments, and in this work we propose several new features to improve the performance of our model. More implementation details for this baseline can be found in the supplementary material.

In order to improve the open vocabulary capability of our model, we propose significant changes to the P3Former architecture, as well as two new loss functions. The architecture of our method is shown in Fig. 1 and mainly consists of multimodal feature fusion, a segmentation head, and input text embeddings for open-vocabulary classification.

Multimodal feature fusion. The core idea of many recent 2D open-vocabulary works is to leverage the features of large-scale vision-language models [22, 39]. These methods [54] mainly rely on frozen CLIP features and use a transformer model to perform the 2D panoptic segmentation task. However, this is not optimal for 3D tasks since many points do not have corresponding valid camera pixels, leading to invalid features preventing meaningful predictions. To fully exploit the power of the CLIP vision features and learn complementary features from both CLIP features from camera and features from LiDAR, we generate predictions from the fusion of CLIP features extracted by a frozen CLIP model and learned LiDAR features from a LiDAR encoder.

As shown in Fig. 1, there are three major components for the multimodal feature fusion including a LiDAR encoder, a vision CLIP encoder, and voxel-level feature fusion. The LiDAR encoder is a model which takes an unordered set of points as input and extracts per-point features. We apply voxelization to the features from the LiDAR encoder, producing output features $F_{lidar} \in \mathbb{R}^{V \times D_{lidar}}$, where V is the number of the voxels and D_{lidar} is the dimension of the learned LiDAR feature. The Vision CLIP encoder is a pre-trained V-L segmentation model [14] which extracts pixel-wise CLIP features from each camera image. Within each voxel, every LiDAR point is projected into the camera image plane based on the intrinsic and extrinsic calibration parameters to index into the corresponding vision CLIP features, then the vision CLIP features of all the points belonging to each voxel are averaged to represent that voxel. Zero padding is used for points which do not have any valid corresponding camera pixels. The

voxel CLIP features will be referred as $F_{vclip} \in \mathbb{R}^{V \times D_{emb}}$, where V is the number of voxels after voxelization and D_{emb} is the dimension of the CLIP features. Finally, the learned per-voxel LiDAR features and frozen per-voxel vision CLIP features are concatenated together to be used as input into the transformer decoder in the segmentation head. This feature fusion enables our model to learn complementary information from both the LiDAR and CLIP features, allowing us to fine-tune our backbone for each dataset's specific data distribution.

Segmentation head. The segmentation head is a transformer [45] model that takes the LiDAR-Vision fused feature as input to produce panoptic segmentation results. Prior works, including existing 2D open-vocabulary works such as FC-CLIP [54], typically use learnable queries q to represent each instance or thing, and they contain a mask prediction head f_{mask} to produce the corresponding mask for each individual object and a classification head f_{cls} to predict the per-mask class score for each known class. However, as a result, they also need to rely on another classifier to handle novel categories. Our goal is to use a single model to handle the prediction for both base and novel categories. Thus, we predict a class embedding instead of a class score for each mask. During training, the model learns to regress an analogy to the CLIP vision embedding for each mask, and the category prediction can be obtained by calculating its similarity with the CLIP text embedding of text queries during the inference stage. The class embedding f_{cls} prediction is defined as:

$$v_q = f_{cls}(q) \in \mathbb{R}^{D_{emb}}, \tag{1}$$

where v_q is in the CLIP embedding space. The predicted class logits are then computed from the cosine similarity between the predicted class embedding and the text embedding of every category name from the evaluation set using a frozen CLIP model. The classification logits are defined as:

$$s_{v_q} = \frac{1}{T}[\cos(v_q, t_1), \cos(v_q, t_2), \ldots, \cos(v_q, t_C)] \tag{2}$$

where $t_i \in \mathbb{R}^{D_{emb}}$, $i \in \{1, 2, \ldots, C\}$ is the text embedding, C is the number of categories (C_B in training and C_B+C_N in testing), and T is a learnable temperature term that controls the concentration of the distribution.

Query assignment. A common practice [9, 54] for transformer-based panoptic segmentation models is to utilize a single set of queries to make predictions for both *things* and *stuff* classes jointly. In contrast, P3Former uses one query set to represent *things* classes after bipartite matching and one fixed query set for *stuff* classes. We have found that this separation of *things* queries and *stuff* queries makes our model converge faster and improve overall performance, and similar pattern has been observed in other tasks [28]. However, the fixed set of queries for *stuff* classes is not applicable to the open-vocabulary setting due to the unknown number of novel stuff classes. To take advantage of the benefits of separating *things* queries and *stuff* queries, we propose to predict the base *stuff* classes with a fixed set of queries and utilize a set of learnable queries to target base *things* classes and all novel (*things* and *stuff*) classes. More details of the query assignment can be found in the supplementary materials.

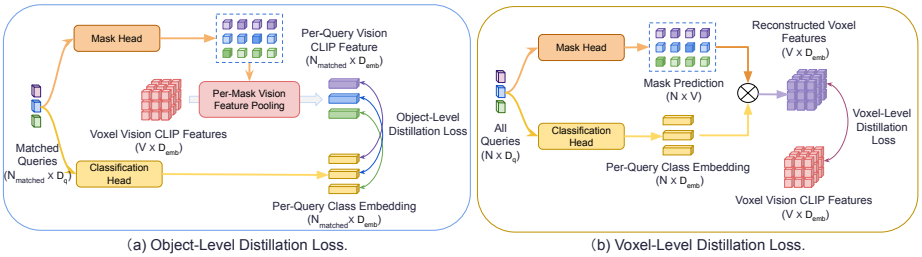

Fig. 2: (a) the proposed object-level distillation loss, and (b) the proposed voxel-level distillation loss.

3.3 Loss Function

Closed-set panoptic segmentation models [47] are typically optimized with objective functions consisting of a classification loss L_{cls} and a mask prediction loss L_{mask}. We follow P3Former [47] for these two losses: the classification loss L_{cls} optimizes the focal loss [30] between the class predictions and the category labels, while the mask loss L_{mask} optimizes the voxel-query classification loss. Besides the two standard loss functions, we propose two simple yet effective losses to apply distillation from the CLIP model at different levels.

Object-level distillation loss. Similar to previous methods [50, 54], we use the cosine similarity between predicted class embeddings and class text CLIP embeddings to produce classification scores. However, the classification loss applied to Eq. (2) only enforces similarity to known classes. In this work, we make the assumption that the frozen CLIP features are discriminative with respect to open-vocabulary classes and have good out-of-distribution generalization. We propose an additional training loss which forces our predicted object-level class embeddings to be similar to the CLIP embeddings within their corresponding masks after matching. Similar to [54], we utilize voxel vision CLIP features to get an embedding for each query q by mask pooling Vision CLIP features:

$$w_q = \frac{1}{|M_q|} \sum_p \mathbb{1}(p \in M_q) F_{vclip}(p) \tag{3}$$

where M_q is the set of points p belonging to the mask for query q. Our object-level distillation loss is then defined as:

$$L_O = \frac{1}{|Q_{matched}|} \sum_{q \in Q_{matched}} 1 - \cos(v_q, w_q), \tag{4}$$

where $Q_{matched}$ is the set of queries matched with ground truth objects during training, v is the set of predicted class embeddings, and w is the set of mask-pooled CLIP embeddings. This loss forces the model to directly distill object-level camera CLIP features and improves model performance for novel *things* classes. We also experimented with applying L_O to all predicted masks, but we

found that this slightly reduced model performance, likely due to the presence of masks that do not correspond to any objects in the scene.

Voxel-level distillation loss. While the object-level distillation loss distills the per-object features from CLIP model, it does not provide any supervision for the mask prediction head, which would otherwise only receive supervision for known classes. We found this particularly problematic for unknown *stuff* classes, which tend to be more spread out and cover larger and more diverse parts of the scene. In addition, it is only being applied to queries with relatively accurate mask predictions in order to learn useful CLIP features. To target these issues, we propose the voxel-level distillation loss to explicitly learn voxel-level CLIP features, which do not depend on any labels and can be applied on all queries. In particular, the voxel-level distillation loss is defined as:

$$F_{rec} = M_Q^T F_{Qemb} \tag{5}$$

where Q is the number of queries, $F_{Qemb} \in \mathbb{R}^{Q \times D_{emb}}$ is the predicted embedding for all queries and $M_Q \in \mathbb{R}^{Q \times V}$ is the predicted per-voxel mask probabilities for all queries. The reconstructed features can be regarded as the weighted sum of all queries for each voxel. We supervise these features with the voxel CLIP features:

$$L_V = L_1(F_{rec}, F_{vclip}) \tag{6}$$

Unlike the object-level distillation loss, which is only applied to queries with matched ground truth, this loss is applied to all predicted mask scores and queries. In our experiments, we found that this loss significantly improves performance on novel *stuff* categories in particular, likely as it does not require exact matches with the ground truth, which can be difficult for large *stuff* classes. However, this loss is still susceptible to noisy or low quality mask scores, and we found that larger weights for this loss can disrupt training.

To summarize, L_O helps get rid of the ensemble of classifiers in [14, 15, 24, 50, 54] and enables open-vocabulary ability with one trainable classifier. L_V uses a scene-level representation represented by the embedding of all queries, while previous methods only consider object-level representation. Combining L_O with L_V enables segmenting novel *things* and novel *stuff* objects simultaneously. Our final objective function can be written as:

$$L = w_\alpha * L_{cls} + w_\beta * L_{mask} + w_\lambda * L_O + w_\gamma * L_V \tag{7}$$

, where w_α, w_β, w_λ, w_γ, are weights for the corresponding objective functions.

3.4 Implementation Details

For the LiDAR encoder and segmentation head, we follow the implementation of the state-of-the-art closed-set 3D panoptic segmentation method P3Former [47]. For the Vision CLIP encoder, we use OpenSeg [14], due to its remarkable performance on the recent open-vocabulary 3D semantic segmentation task [35]. For the Text CLIP encoder, we use CLIP [39] with ViT-L/14 [45] backbone, following other state-of-the-art open vocabulary works [35].

4 Experiments

4.1 Experimental Setting

Following the state-of-the-art closed-set 3D panoptic segmentation work [27, 40, 42, 47, 52, 58], we conduct experiments and ablation studies on the nuScenes [4] and SemanticKITTI [2, 13] datasets.

nuScenes. The nuScenes dataset [4] is a public benchmark for autonomous driving. It consists of 1000 run segments and is further divided into prescribed train/val/test splits. We use all key frames with panoptic labels in the training set(28130 frames) to train the model. Following the most recent state-of-the-art model P3Former [47], we evaluate the models on the validation set(6019 frames). There are 16 semantic classes, including 10 *things* classes and 6 *stuff* classes.

SemanticKITTI. SemanticKITTI [2, 13] is the first large dataset for LiDAR panoptic segmentation for autonomous driving. We conduct experiments on the training and validation sets, where panoptic segmentation labels are available. 3D open-vocabulary methods often require point and pixel pairing. In the SemanticKITTI dataset, however, the ego-vehicle is only equipped with frontal cameras. Thus, we filter out the points that are not visible in the camera view based on the provided camera parameters for both training and evaluation. There are 19 semantic classes, including 8 *things* classes and 11 *stuff* classes.

Data split. Both the nuScenes and SemanticKITTI datasets do not provide official base and novel class splits. Following the state-of-the-art 3D open-vocabulary segmentation work [6, 10, 53], we randomly split the classes into base and novel, while keeping the ratio between base and novel classes around 3 : 1. For nuScenes, the number of class for base and novel split are 12 and 4 respectively, and this setting will be referred as B12/N4. For SemanticKITTI, the number of class for base and novel split are 14 and 5, and this setting will be referred as B14/N5. We use the same splits in the main comparison with prior methods, and provide the results of more variations in the ablation studies and supplementary materials.

Training details. We follow most of the architecture configurations in the official P3Former [47] implementation. We set $w_\alpha = 1$, $w_\beta = 1$, $w_\lambda = 1$, $w_\gamma = 0.1$ for both datasets. We use the AdamW [23, 33] optimizer with a weight decay of 0.01. We set the initial learning rate as 0.0008 with a multi-step decay schedule. The models are trained for 40 epochs, and we use the checkpoint of the last epoch for evaluation. To avoid ambiguous class names and better utilize the CLIP text embedding, we follow [25, 35, 54] and apply multi-label mapping for the text queries. During inference, if there are multiple labels for one class, we derive the class score by getting the maximum scores among these labels.

Evaluation metrics. We use panoptic quality (PQ) as the major evaluation metric for the panoptic segmentation task. PQ is formulated as:

$$\mathrm{PQ} = \underbrace{\frac{\sum_{TP} \mathrm{IoU}}{|TP|}}_{\mathrm{SQ}} \times \underbrace{\frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|}}_{\mathrm{RQ}}. \tag{8}$$

PQ is the multiplication of segmentation quality (SQ) and recognition quality (RQ). We report all the three metrics (PQ, RQ, SQ) for all classes. We also

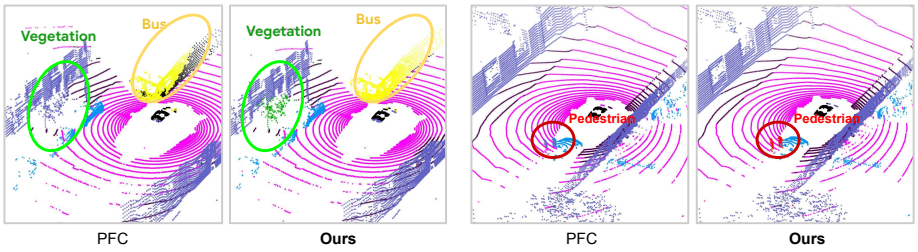

Fig. 3: Open-vocabulary panoptic segmentation results from PFC and our method on nuScenes. PFC predicts inaccurate category and masks for the novel pedestrian (red), bus (yellow) and vegetation (green), while ours makes correct predictions.

report PQ, RQ, SQ for novel *things* objects and novel *stuff* objects separately. In particular, PQ_N^{Th} means PQ for novel *things* classes and PQ_N^{St} stands for PQ for novel *stuff* classes. We also report the mean Intersection over Union (mIoU) for all classes to measure semantic segmentation quality.

4.2 P3Former-FC-CLIP Baseline

As a baseline for novel-class panoptic segmentation, we construct a model from a fusion of P3Former [47] and FC-CLIP [54]. This baseline will be called P3Former-FC-CLIP (PFC). The baseline model takes the frozen voxel vision CLIP features as input, and the final prediction is obtained by geometric ensembling [14,15,24, 50,54] of the results from the classification head f_{cls} and another frozen classifier based on the similarity between the average-pool class embedding w_q and the CLIP text embedding. Following FC-CLIP [54], the same set of learnable queries were used to represent both *things* and *stuff* classes. In summary, this baseline provides a comparison against our proposed method without the multimodal feature fusion module, the unified segmentation head, and the distillation losses. More information on this baseline can be found in the supplementary material.

4.3 Main Results

Since there are no existing methods for the 3D open-vocabulary panoptic segmentation task, we mainly compare with three methods to demonstrate the capability of our method: (1) the strong open-vocabulary baseline method PFC to fairly demonstrate the strength of our method, (2) the closed-set state-of-the-art 3D panoptic segmentation method P3Former to understand the headroom of our method, and (3) the open-set, zero-shot state-of-the-art method for 3D semantic segmentation, OpenScene [35]. Comparisons on the nuScenes and SemanticKITTI datasets are shown in Tab. 1 and Tab. 3.

Results on nuScenes dataset. Table 1 shows the quantitative comparison on the validation set of the nuScenes dataset. Our method significantly outperforms

Table 1: Quantitative results of panoptic segmentation on nuScenes. We compare the performance of open-vocabulary and fully supervised models. All open vocabulary models share the same randomly picked base/novel split: B12/N4. The novel *things* classes are bus, pedestrian and motorcycle. The novel *stuff* class is vegetation.

Model	Type	Supervision	PQ	PQ^{Th}_{N}	PQ^{St}_{N}	RQ	RQ^{Th}_{N}	RQ^{St}_{N}	SQ	SQ^{Th}_{N}	SQ^{St}_{N}	mIoU
P3Former [47]	closed-set	full	75.9	85.1	82.9	84.7	89.9	95.9	89.8	94.7	86.5	76.8
OpenScene [35]	open-voc	zero-shot	-	-	-	-	-	-	-	-	-	42.1
PFC	open-voc	partial	54.8	37.3	0.5	63.6	42.1	0.8	84.2	**89.3**	60.4	55.5
Ours	open-voc	partial	**62.0**	**49.6**	**35.2**	**70.9**	**55.6**	**46.0**	**87.0**	89.1	**76.7**	**60.1**

Table 2: Performance for base classes on nuScenes. We report the performance on base classes for models in Tab. 1. A gap still exists between open and closed-set methods for base classes. We show that this is due to lack of supervision of the whole scene as P3Former achieves similar performance when only trained on base categories.

Model	Supervision	Training Data	Base *Things*			Base *Stuff*		
			PQ^{Th}_{B}	RQ^{Th}_{B}	SQ^{Th}_{B}	PQ^{St}_{B}	RQ^{St}_{B}	SQ^{St}_{B}
P3Former [47]	full	base+novel	73.4	80.5	90.9	73.9	85.3	85.9
P3Former [47]	partial	base	65.2	71.3	88.0	64.2	77.4	81.8
PFC	partial	base	65.6	73.3	89.0	61.0	75.4	83.7
Ours	partial	base	66.7	73.7	89.8	69.2	82.1	83.7

the strong baseline PFC across all metrics. PFC works relatively well for the novel *things* classes, but performance on the novel *stuff* class collapses. This is likely because *stuff* classes tend to cover large parts of the scene, leading to diverse per-voxel CLIP features which may not be good representatives for their respective classes. Qualitative comparison is provided in Fig. 3.

To further understand the headroom of our method, we also compare our model with the closed-set P3Former. Note that the comparison here is deliberately unfair since the supervision signals are different. Compared with the closed-set P3Former, our segmentation quality(SQ) is good while there is a large gap on mask classification quality(RQ). The gap is largely due to regressions in the novel classes, where precise supervision is not available for open-vocabulary models. For base classes, as shown in Tab. 2, the gap is relatively small except for a drop in RQ^{Th}_{B}. We believe the closed-set P3Former sees ground truth supervision for the entire scene, while open-set methods do not receive supervision in the 'unknown class' regions. In fact, when P3Former is only trained on base categories, the performance is worse than our proposed method. Besides the comparison with the closed-set method, we also compare with the zero-shot state-of-the-art method OpenScene [35] which does not use any labels for training. In this comparison, our model significantly outperforms OpenScene in the mIoU metric for semantic segmentation. Note that this comparison is not en-

Table 3: Quantitative results of panoptic segmentation on SemanticKITTI. We compare the performance different models. All open vocabulary models share the same randomly picked base/novel split: B14/N5. The novel *things* classes are bicycle and truck. The novel *stuff* classes are sidewalk, building and trunk.

Model	Type	Supervision	PQ	PQ^{Th}_N	PQ^{St}_N	RQ	RQ^{Th}_N	RQ^{St}_N	SQ	SQ^{Th}_N	SQ^{St}_N	mIoU
P3Former [47]	closed-set	full	62.1	65.9	74.2	71.3	74.8	86.8	77.1	88.3	83.9	61.6
PFC	open-voc	partial	33.7	12.0	0.4	40.1	15.0	0.6	67.6	81.1	47.3	33.4
Ours	open-voc	partial	**42.2**	**13.1**	**17.8**	**50.4**	**16.2**	**26.7**	**73.0**	**84.0**	**67.2**	**44.6**

Table 4: Impact of each component. We evaluate the impact of each component using the base/novel split in Tab. 1. We observe that each component can provide improvements over the PCF baseline. Noticeably, L_V brings the biggest improvement.

Components				PQ	PQ^{Th}_N	PQ^{St}_N	RQ	RQ^{Th}_N	RQ^{St}_N	SQ	SQ^{Th}_N	SQ^{St}_N	mIoU
QA	Fusion	L_O	L_V										
				54.8	37.3	0.5	63.6	42.1	0.8	84.2	**89.3**	60.4	55.5
✓				55.5	35.7	0.4	64.0	40.8	0.7	84.3	87.4	56.5	56.6
✓	✓			56.4	38.1	0.4	65.0	43.5	0.6	84.6	87.4	61.3	56.4
✓	✓	✓		56.3	43.8	0.2	64.8	49.2	0.3	85.1	88.9	64.0	54.0
✓	✓	✓	✓	**62.0**	**49.6**	**35.2**	**70.9**	**55.6**	**46.0**	**87.0**	89.1	**76.7**	**60.1**

tirely fair, as our method is trained with partial labels. Instead, the comparison is useful to understand the gap between the two types of open-vocabulary methods. The concurrent work RegionPLC [53] also reports open-vocabulary results for the semantic segmentation task on the nuScenes dataset. However, we cannot directly compare with this method since it removes one class (other-flat) and does not provide its base/novel split.

Results on SemanticKITTI dataset. To demonstrate the generalization ability of our method across different datasets, we report the results on SemanticKITTI dataset in Tab. 3. Overall, we observe similar patterns as on the nuScenes dataset. The baseline achieves relatively poor overall performance and struggles with the novel *stuff* classes. Using our architecture and loss functions, our model significantly outperforms PFC on PQ, with the largest margin for novel *stuff* classes. Note that the gap between the open-vocabulary methods (ours and PFC) and the closed-set method is larger on SemanticKITTI, likely due to the smaller dataset limiting performance.

4.4 Ablation Studies and Analysis

To better understand the effectiveness of each component, we conduct ablation studies for each design choice and loss function on the nuScenes dataset. These results are shown in Tab. 4. We conduct five sets of experiments, starting with the PFC baseline and build upon it four ablations with different combinations.

Table 5: Performance on a different split. We compare the performance with a split with 5 novel classes (B11/N5). The novel *things* classes are bicycle, car and construction vehicle. The novel *stuff* classes are terrain and man-made. Our method consistently outperforms the PFC baseline across all the metrics by a large margin.

Model	Type	Supervision	PQ	PQ_N^{Th}	PQ_N^{St}	RQ	RQ_N^{Th}	RQ_N^{St}	SQ	SQ_N^{Th}	SQ_N^{St}	mIoU
P3Former [47]	closed-set	full	75.8	70.5	71.7	83.8	76.4	85.5	90.1	91.6	83.6	75.0
PFC	open-voc	partial	43.9	27.7	0.6	51.7	33.2	1.0	80.2	82.4	62.7	45.2
Ours	open-voc	partial	**52.8**	**56.0**	**16.4**	**60.5**	**61.8**	**22.6**	**84.9**	**89.7**	**68.7**	**49.9**

Impact of query assignment. Starting from the PFC baseline model, we add our proposed fixed query assignment for *stuff* categories. As shown in the second row of Tab. 4, with query assignment, the overall PQ improves by 0.7. The performance for the novel classes drop slightly, but improvement on the base classes overcomes this for the overall PQ.

Impact of feature fusion. The third row of Tab. 4 shows the impact of feature fusion. Without feature fusion, our model already achieves 55.5 PQ, demonstrating the power of the CLIP vision features. The third row shows that the performance with feature fusion for the model input improves the overall PQ by 0.9. This slightly improved the overall performance, but the improvement on the novel *things* class is the most significant, demonstrating that the learned LiDAR features and CLIP vision features are indeed complementary for the task.

Impact of object-level distillation loss. The fourth row of the results in Tab. 4 shows the impact of the proposed object-level distillation loss. Note that for models with the object-level distillation loss, we remove the frozen class classification head and the ensemble in the PFC baseline, consolidating to a single class embedding head. Although the RQ_N^{St} slightly dips by 0.3 for the novel *stuff* classes, this loss can significantly improve the RQ_N^{Th} for the novel *things* class by 5.7.

Impact of voxel-level distillation loss. We study the impact of the voxel-level distillation loss to see if it can further improve the performance given all of our designs. The results are shown in the last row of Tab. 4. With this loss function, PQ significantly improves by 5.7. The improvement on the novel split is particularly large, especially for the novel *stuff* classes. The PQ_N^{St} of the novel *stuff* class improves from 0.2 to 35.2, which demonstrates the importance of the voxel-level supervision to the performance of the novel *stuff* class.

Performance of different splits. To validate the generalizability of our method, we conduct experiments on a different split (B11/N5) for the nuScenes dataset. As shown in Tab. 5, our proposed method consistently and significantly outperforms the strong baseline method. This again demonstrates the effectiveness of our design and the proposed loss functions.

Open-vocabulary exploration. In previous experiments, we follow other 3D open-vocabulary works [6, 10, 53] and provide analytical results on pre-defined object categories, mainly due to the limited categories in current panoptic segmentation datasets. In practice, our model goes beyond detecting these object categories: we can take class embeddings v_q in Eq. (1) and compute the cosine

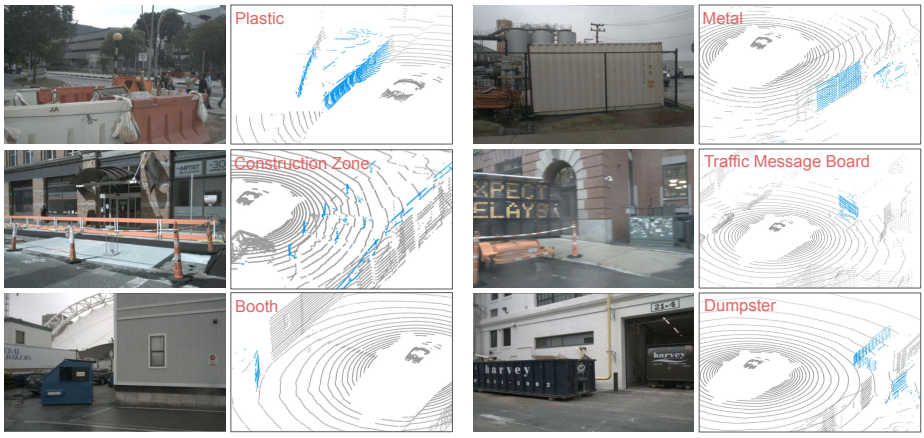

Fig. 4: Open-vocabulary exploration. We show the novel materials/objects in blue color. The orientation of the ego vehicle is fixed in the LiDAR point visualization while the reference images come from on of the surrounding cameras of the ego vehicle.

similarity with CLIP embedding of any text. Fig. 4 shows that we can detect novel materials/objects that are not in the predefined category list. Note that the concept of open vocabulary is very different from domain adaptation, as open vocabulary refers to the ability to deal with novel inputs in a scene while domain adaptation addresses the difference in data distributions in different scenes.
Limitations. Our models are only evaluated on current autonomous driving panoptic segmentation benchmarks, with limited number of category annotations. To further evaluate open-vocabulary performance, a large-scale autonomous driving benchmark with more diverse object categories is greatly desired.

5 Conclusion

In this paper, we present the first approach for the open-vocabulary 3D panoptic segmentation task in autonomous driving by leveraging large vision-language models. We experimentally verified that simply extending the 2D open-vocabulary segmentation method into 3D does not yield good performance, and demonstrated that our proposed model design and loss functions significantly boost performance for this task. Our method significantly outperformed the strong baseline on multiple well-established benchmarks. We hope our work can shed light on the future studies of the 3D open-vocabulary panoptic segmentation.

Acknowledgements. We would like to thank Mahyar Najibi, Chao Jia, Zhenyao Zhu, Yolanda Wang, Charles R. Qi, Dragomir Anguelov, Tom Ouyang, Ruichi Yu, Chris Sweeney, Colin Graber, Yingwei Li, Sangjin Lee, Weilong Yang, and Congcong Li for the help to the project.

References

1. Alonso, I., Riazuelo, L., Montesano, L., Murillo, A.C.: 3d-mininet: Learning a 2d representation from point clouds for fast and efficient 3d lidar semantic segmentation. IEEE Robotics and Automation Letters **5**(4), 5432–5439 (2020)
2. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences. In: ICCV (2019)
3. Bendale, A., Boult, T.: Towards open world recognition. In: CVPR (2015)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: CVPR (2020)
5. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: ECCV (2020)
6. Cen, J., Yun, P., Zhang, S., Cai, J., Luan, D., Wang, M.Y., Liu, M., Tang, M.: Open-world semantic segmentation for LIDAR point clouds. In: ECCV (2022)
7. Chen, R., Liu, Y., Kong, L., Zhu, X., Ma, Y., Li, Y., Hou, Y., Qiao, Y., Wang, W.: Clip2scene: Towards label-efficient 3d scene understanding by clip. In: CVPR (2023)
8. Chen, Z., Li, B.: Bridging the domain gap: Self-supervised 3d scene understanding with foundation models. arXiv preprint arXiv:2305.08776 (2023)
9. Cheng, B., Schwing, A., Kirillov, A.: Per-pixel classification is not all you need for semantic segmentation. In: NeurIPS (2021)
10. Ding, R., Yang, J., Xue, C., Zhang, W., Bai, S., Qi, X.: Pla: Language-driven open-vocabulary 3d scene understanding. In: CVPR (2023)
11. Ding, Z., Wang, J., Tu, Z.: Open-vocabulary universal image segmentation with maskclip. In: ICML (2023)
12. Du, Y., Wei, F., Zhang, Z., Shi, M., Gao, Y., Li, G.: Learning to prompt for open-vocabulary object detection with vision-language model. In: CVPR (2022)
13. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In: CVPR (2012)
14. Ghiasi, G., Gu, X., Cui, Y., Lin, T.Y.: Scaling open-vocabulary image segmentation with image-level labels. In: ECCV (2022)
15. Gu, X., Lin, T.Y., Kuo, W., Cui, Y.: Open-vocabulary object detection via vision and language knowledge distillation. ICLR (2022)
16. Ha, H., Song, S.: Semantic abstraction: Open-world 3d scene understanding from 2d vision-language models. In: CoRL (2022)
17. He, W., Jamonnak, S., Gou, L., Ren, L.: Clip-s4: Language-guided self-supervised semantic segmentation. In: CVPR (2023)
18. Hegde, D., Valanarasu, J.M.J., Patel, V.M.: Clip goes 3d: Leveraging prompt tuning for language grounded 3d recognition. arXiv preprint arXiv:2303.11313 (2023)
19. Hong, F., Zhou, H., Zhu, X., Li, H., Liu, Z.: Lidar-based panoptic segmentation via dynamic shifting network. In: CVPR (2021)
20. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A.: Learning semantic segmentation of large-scale point clouds with random sampling. IEEE Transactions on Pattern Analysis and Machine Intelligence **44**(11), 8338–8354 (2021)
21. Ilharco, G., Wortsman, M., Wightman, R., Gordon, C., Carlini, N., Taori, R., Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., Schmidt, L.: Openclip (Jul 2021). `https://doi.org/10.5281/zenodo.5143773`, `https://doi.org/10.5281/zenodo.5143773`

22. Jia, C., Yang, Y., Xia, Y., Chen, Y.T., Parekh, Z., Pham, H., Le, Q., Sung, Y.H., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML (2021)
23. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
24. Kuo, W., Cui, Y., Gu, X., Piergiovanni, A., Angelova, A.: F-vlm: Open-vocabulary object detection upon frozen vision and language models. In: ICLR (2023)
25. Lambert, J., Liu, Z., Sener, O., Hays, J., Koltun, V.: Mseg: A composite dataset for multi-domain semantic segmentation. In: CVPR (2020)
26. Li, B., Weinberger, K.Q., Belongie, S., Koltun, V., Ranftl, R.: Language-driven semantic segmentation. In: ICLR (2022)
27. Li, J., He, X., Wen, Y., Gao, Y., Cheng, X., Zhang, D.: Panoptic-phnet: Towards real-time and high-precision lidar panoptic segmentation via clustering pseudo heatmap. In: CVPR (2022)
28. Li, Z., Wang, W., Xie, E., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P., Lu, T.: Panoptic segformer: Delving deeper into panoptic segmentation with transformers. In: CVPR (2022)
29. Liang, F., Wu, B., Dai, X., Li, K., Zhao, Y., Zhang, H., Zhang, P., Vajda, P., Marculescu, D.: Open-vocabulary semantic segmentation with mask-adapted clip. In: CVPR (2023)
30. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollar, P.: Focal loss for dense object detection. In: ICCV (2017)
31. Liu, Q., Wen, Y., Han, J., Xu, C., Xu, H., Liang, X.: Open-world semantic segmentation via contrasting and clustering vision-language embedding. In: ECCV (2022)
32. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: CVPR (2022)
33. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: ICLR (2019)
34. Ma, C., Yang, Y., Wang, Y., Zhang, Y., Xie, W.: Open-vocabulary semantic segmentation with frozen vision-language models. BMVC (2022)
35. Peng, S., Genova, K., Jiang, C., Tagliasacchi, A., Pollefeys, M., Funkhouser, T., et al.: Openscene: 3d scene understanding with open vocabularies. In: CVPR (2023)
36. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: CVPR (2017)
37. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. NeurIPS (2017)
38. Qin, J., Wu, J., Yan, P., Li, M., Yuxi, R., Xiao, X., Wang, Y., Wang, R., Wen, S., Pan, X., et al.: Freeseg: Unified, universal and open-vocabulary image segmentation. In: CVPR (2023)
39. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: ICML (2021)
40. Razani, R., Cheng, R., Li, E., Taghavi, E., Ren, Y., Bingbing, L.: Gp-s3net: Graph-based panoptic sparse semantic segmentation network. In: ICCV (2021)
41. Rozenberszki, D., Litany, O., Dai, A.: Language-grounded indoor 3d semantic segmentation in the wild. In: ECCV (2022)
42. Sirohi, K., Mohan, R., Büscher, D., Burgard, W., Valada, A.: Efficientlps: Efficient lidar panoptic segmentation. IEEE Transactions on Robotics **38**(3), 1894–1914 (2021)
43. Takmaz, A., Fedele, E., Sumner, R.W., Pollefeys, M., Tombari, F., Engelmann, F.: Openmask3d: Open-vocabulary 3d instance segmentation. In: NeuRIPS (2023)

44. Tang, H., Liu, Z., Zhao, S., Lin, Y., Lin, J., Wang, H., Han, S.: Searching efficient 3d architectures with sparse point-voxel convolution. In: ECCV (2020)
45. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: NeurIPS (2017)
46. Wu, W., Fuxin, L., Shan, Q.: Pointconvformer: Revenge of the point-based convolution. In: CVPR (2023)
47. Xiao, Z., Zhang, W., Wang, T., Loy, C.C., Lin, D., Pang, J.: Position-guided point cloud panoptic segmentation transformer. arXiv preprint (2023)
48. Xu, J., Zhang, R., Dou, J., Zhu, Y., Sun, J., Pu, S.: Rpvnet: A deep and efficient range-point-voxel fusion network for lidar point cloud segmentation. In: ICCV (2021)
49. Xu, J., De Mello, S., Liu, S., Byeon, W., Breuel, T., Kautz, J., Wang, X.: Groupvit: Semantic segmentation emerges from text supervision. In: CVPR (2022)
50. Xu, J., Liu, S., Vahdat, A., Byeon, W., Wang, X., De Mello, S.: Open-vocabulary panoptic segmentation with text-to-image diffusion models. In: CVPR (2023)
51. Xu, M., Zhang, Z., Wei, F., Lin, Y., Cao, Y., Hu, H., Bai, X.: A simple baseline for open-vocabulary semantic segmentation with pre-trained vision-language model. In: ECCV (2022)
52. Xu, S., Wan, R., Ye, M., Zou, X., Cao, T.: Sparse cross-scale attention network for efficient lidar panoptic segmentation. In: AAAI (2022)
53. Yang, J., Ding, R., Wang, Z., Qi, X.: Regionplc: Regional point-language contrastive learning for open-world 3d scene understanding. In: CVPR (2024)
54. Yu, Q., He, J., Deng, X., Shen, X., Chen, L.C.: Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip. In: NeurIPS (2023)
55. Zhang, J., Dong, R., Ma, K.: Clip-fo3d: Learning free open-world 3d scene representations from 2d dense clip. In: ICCV (2023)
56. Zhou, C., Loy, C.C., Dai, B.: Extract free dense labels from clip. In: ECCV (2022)
57. Zhou, Z., Lei, Y., Zhang, B., Liu, L., Liu, Y.: Zegclip: Towards adapting clip for zero-shot semantic segmentation. In: CVPR (2023)
58. Zhou, Z., Zhang, Y., Foroosh, H.: Panoptic-polarnet: Proposal-free lidar point cloud panoptic segmentation. In: CVPR (2021)
59. Zou, X., Dou, Z.Y., Yang, J., Gan, Z., Li, L., Li, C., Dai, X., Behl, H., Wang, J., Yuan, L., et al.: Generalized decoding for pixel, image, and language. In: CVPR (2023)