SA3DIP: Segment Any 3D Instance with Potential 3D Priors

Xi Yang[1]**, Xu Gu**[1]**, Xingyilang Yin**[1]*, **Xinbo Gao**[2]
[1]Xidian University, [2]Chongqing University of Posts and Telecommunications
yangx@xidian.edu.cn, {ryangu,yxyl}@stu.xidian.edu.cn, gaoxb@cqupt.edu.cn

Abstract

The proliferation of 2D foundation models has sparked research into adapting them for open-world 3D instance segmentation. Recent methods introduce a paradigm that leverages superpoints as geometric primitives and incorporates 2D multi-view masks from Segment Anything model (SAM) as merging guidance, achieving outstanding zero-shot instance segmentation results. However, the limited use of 3D priors restricts the segmentation performance. Previous methods calculate the 3D superpoints solely based on estimated normal from spatial coordinates, resulting in under-segmentation for instances with similar geometry. Besides, the heavy reliance on SAM and hand-crafted algorithms in 2D space suffers from over-segmentation due to SAM's inherent part-level segmentation tendency. To address these issues, we propose SA3DIP, a novel method for **S**egmenting **A**ny **3D** **I**nstances via exploiting potential 3D **P**riors. Specifically, on one hand, we generate complementary 3D primitives based on both geometric and textural priors, which reduces the initial errors that accumulate in subsequent procedures. On the other hand, we introduce supplemental constraints from the 3D space by using a 3D detector to guide a further merging process. Furthermore, we notice a considerable portion of low-quality ground truth annotations in ScanNetV2 benchmark, which affect the fair evaluations. Thus, we present ScanNetV2-INS with complete ground truth labels and supplement additional instances for 3D class-agnostic instance segmentation. Experimental evaluations on various 2D-3D datasets demonstrate the effectiveness and robustness of our approach. Our code and proposed ScanNetV2-INS dataset are available HERE.

1 Introduction

3D instance segmentation is a fundamental task pivotal to 3D understanding across various domains such as autonomous driving, robotics navigation, and virtual reality applications. State-of-the-art methods [1, 2] are predominantly supervised and rely heavily on precise 3D annotations for training, thus constraining their applications in open-world scenes. Compared to scarce 3D labeled data, the acquisition and annotation of 2D images are more convenient. Recently, 2D foundation models [3–6] trained on large-scale annotated 2D data show impressive performance and strong generalization capabilities in zero-shot scenarios. Recent efforts have sought to leverage Segment Anything Model (SAM) by lifting its class-agnostic 2D segmentation results to 3D tasks [7–10]. Specifically, some methods [7, 8] propose a pipeline that decomposes the 3D scene into geometric primitives and leverages 2D multi-view masks from SAM to calculate pairwise similarity scores as merging guidance. Further well-designed algorithms or Graph Neural Networks (GNNs) are included to ensure multi-view consistency.

*Corresponding author.

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

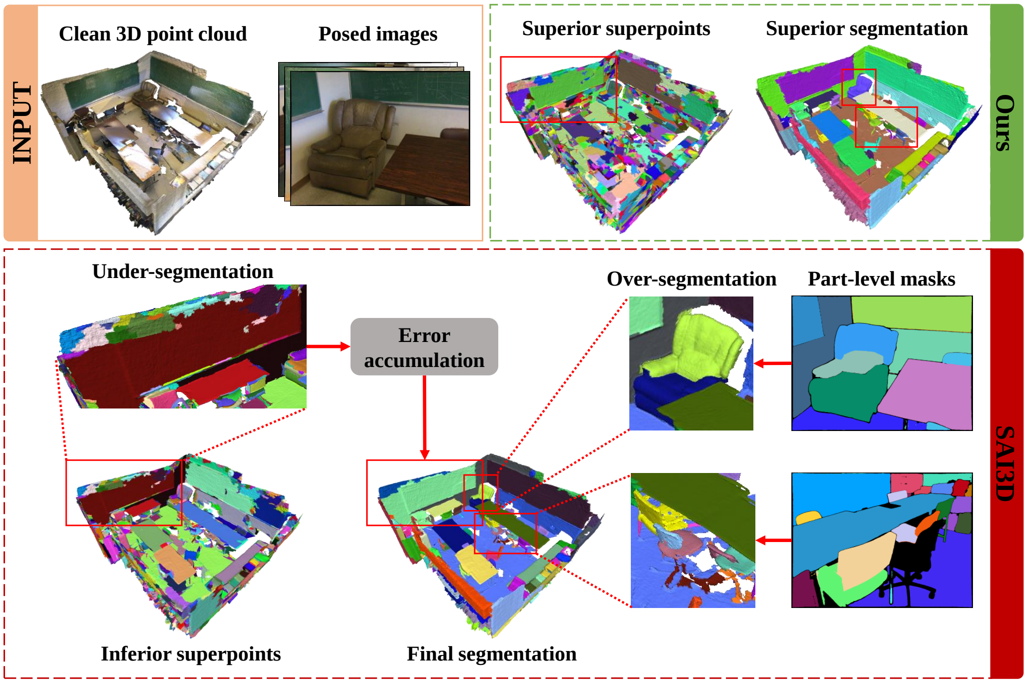

Figure 1: Comparison of our **SA3DIP** with other methods. Methods like SAI3D (bottom) fail to distinguish instances with similar normals when computing superpoints, which accumulate to the final segmentation. Moreover, the part-level 2D segmentation transfers to 3D space, resulting in over-segmented 3D instances. We present a novel pipeline for segmenting any 3D instances, which overcomes the limitations by exploiting additional 3D priors, specifically by incorporating both geometric and textural prior on superpoints computing, and supplementing 3D space constraint provided 3D prior by utilizing a 3D detector.

However, the geometric rudimentary pre-segmentation initialization impedes their ability to group superpoints on points with highly similar normals, such as boards on walls and books on tabletops. As shown in Fig. 1 bottom left, the blackboard and the wall are wrongly allocated within the same superpoint using previous methods. Owing to the coarse-to-fine pattern of the pipeline, errors at this stage propagate to subsequent stages, which the sophisticated merging algorithms fail to rectify. Furthermore, current approaches heavily rely on 2D foundation models and design algorithms or GNNs within 2D space, neglecting the inherent 3D priors of the data. Part-level segmentation in the generated 2D masks by SAM transfers to 3D space and leads to over-segmented 3D instances. As illustrated in Fig. 1 bottom right, the sofa and chairs are segmented at the part level in 2D space, causing over-segmentation in the final results. These limitations primarily stem from the under-exploitation of 3D priors: (1) Complete point cloud data encompasses not only spatial coordinates but also color channels; (2) Constraints provided by 3D space prior to the merging process cannot be neglected.

In this paper, we present **SA3DIP** (**S**egment **A**ny **3D** **I**nstance with potential 3D **P**riors), a novel method for segmenting high-quality 3D instances. Specifically, we observe that distinct instances with similar normals often exhibit different colors. Therefore, we incorporate both geometric and textural priors to generate finer-grained complementary primitives. As shown in Fig. 1 top right, our method identifies the boundary between the blackboard and the wall clearly. In this way, the initial errors are minimized, which reduces error accumulation in the subsequent process. Moreover, we exploit the 3D prior at the merging stage to provide constraints on the over-segmented 3D instances, which is implemented by incorporating a 3D detector. This additional 3D prior enables rectification on the over-segmented 3D instances, while preserving the capability in handling fine-grained objects. Therefore, the sofa and chairs maintain their integrity in 3D space by our approach, which is illustrated in Fig. 1 top right. Additionally, we notice that the widely-used benchmark, ScanNet [11], contains a considerable portion of low-quality ground truth annotations for instance segmentation, which leads to biases in assessing model performance. Thus, we propose ScanNetV2-INS, a point-level enhanced version tailored for 3D class-agnostic instance segmentation. The revised dataset contains fewer incomplete labels and fewer missing instances, which better showcases real-world scenarios.

Our contributions are three-fold: (1) We present **SA3DIP**, a novel pipeline for segmenting any 3D instances by exploiting potential 3D priors, which includes incorporating both geometric and color priors on computing 3D superpoints, and introducing constraints provided by 3D prior at the merging stage; (2) We propose a point-level enhanced version of ScanNetV2, specifically for 3D class-agnostic instance segmentation by rectifying incomplete annotations and incorporating more instances; (3) Extensive experiments are conducted on ScanNetV2, ScanNetV2-INS and ScanNet++ [12] datasets, and the competitive results demonstrate the effectiveness and robustness of our method.

2 Related Work

Close-set 3D segmentation. 3D semantic segmentation aims to classify each point into a specific semantic class [13–21]. 3D instance segmentation, on the other hand, assigns unique masks to each distinct instance within the same semantic category [2, 22–26]. Prior research, categorized as Grouping-based [23, 27, 28], Kernel-based [29], and Transformer-based [2, 30, 31] methods, has primarily relied on labeled datasets in a supervised manner. Mask3D [2] proposes the first Transformer-based model for 3D semantic instance segmentation that uses instance queries and Transformer decoders. Spherical Mask [1] achieves state-of-the-art 3D instance segmentation performance on the ScanNetV2 dataset by leveraging a novel coarse-to-fine approach [32, 33] based on spherical representation. However, they all necessitate a significant corpus of annotated 3D data for network training, which is financially burdensome and poses challenges for extending the method to open-world scenarios featuring novel objects from unobserved categories.

Open-set 3D segmentation. 2D foundation models [3–6] have exhibited remarkable efficacy across various tasks. Training on the SA-1B dataset with 11 million images and 1.1 billion masks, Segment Anything model (SAM) serves as a cornerstone for image segmentation, allowing strong zero-shot transfer ability and diverse prompts such as points, boxes, and texts to generate high-quality segmentation masks. Inspired by the generalization capabilities of foundation models, certain works [7–10, 34–37] explore the feasibility of bridging the gap between 2D and 3D, enabling various open-vocabulary 3D tasks. OpenScene [38] and OpenMask3D [25] both rely on transferring knowledge from CLIP, where the former infers CLIP features for each 3D point and classifies them embeddings of class labels, and the latter uses additional pre-trained 3D segmentation model to produce class-agnostic 3D proposals and classifies them based on CLIP scores. SAM3D [10] pioneers the extension of SAM into 3D perception by transferring segmentation information from 2D images to 3D space. SAMPro3D [9] attempts to locate 3D points as natural prompts to align their projected pixel prompts across frames, while its segmentation quality heavily relies on the accuracy of 2D segmentation results. Several other works [7, 8] follow the idea that segments each frame individually and devises a merging algorithm or graph neural network (GNN) to guide the merging process of pre-segmented superpoints. However, the limited use of 3D priors restricts the performance in both superpoints computing and region growth, which results in substandard 3D segmentation. In this paper, we present **SA3DIP**, which incorporates more priors and constraints to minimize error accumulation and over-segmentation, by a thorough exploitation of 3D priors.

3 Methodology

3.1 SA3DIP

Overview. Our overall pipeline is shown in Fig. 2. Given a point cloud $P \in \mathbb{R}^{N\times 6}$ and corresponding 2D data $\{I_m, D_m, K_m, E_m\}_{m=1}^{M}$, which denote the RGB, depth images, camera intrinsic and extrinsic parameters, respectively, the masks of 3D instances in the scene are desired as output. First, we generate complementary 3D primitives of the given point cloud via performing 3D over-segmentation on both geometric and textural priors. Then, we construct the superpoint graph by treating the 3D primitives and their relations as nodes and edges of the graph, respectively. Leveraging the 2D masks generated by 2D foundation segmentators like SAM, we create the affinity matrix which

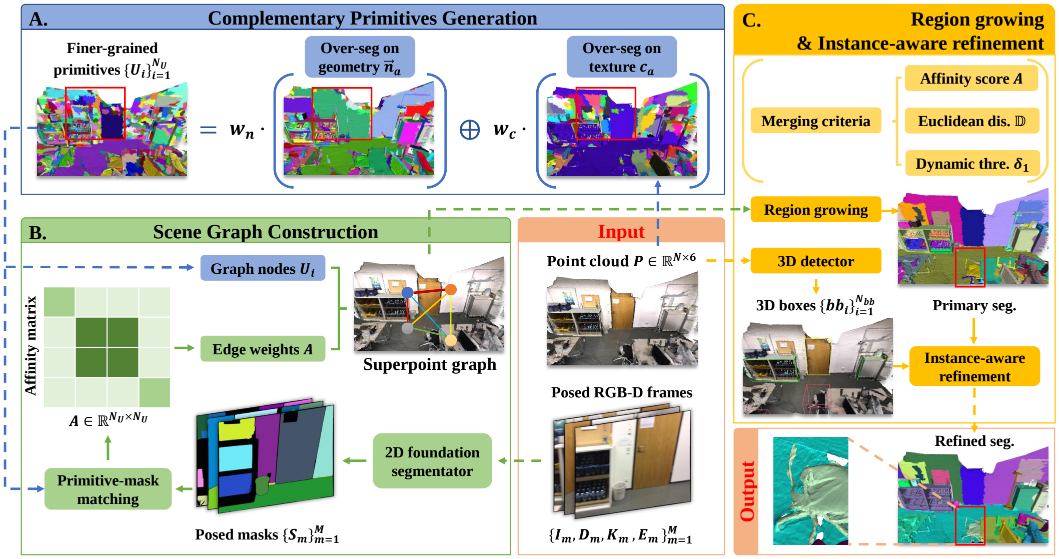
A.
Complementary Primitives Generation
Finer-grained primitives $\{U_i\}_{i=1}^{N_U}$
$= w_n \cdot$
Over-seg on geometry $\vec{n}_a$
$\oplus\ w_c \cdot$
Over-seg on texture c_a
B.
Scene Graph Construction
Graph nodes U_i
Affinity matrix
Edge weights A
Superpoint graph
$A \in \mathbb{R}^{N_U \times N_U}$
Primitive-mask matching
Posed masks $\{S_m\}_{m=1}^{M}$
2D foundation segmentator
Input
Point cloud $P \in \mathbb{R}^{N \times 6}$
Posed RGB-D frames
$\{I_m, D_m, K_m, E_m\}_{m=1}^{M}$
C.
Region growing & Instance-aware refinement
Merging criteria
Affinity score A
Euclidean dis. $\mathbb{D}$
Dynamic thre. δ_1
Region growing
3D detector
3D boxes $\{bb_i\}_{i=1}^{N_{bb}}$
Primary seg.
Instance-aware refinement
Output
Refined seg.

Figure 2: **Overall pipeline.** Our approach first integrates both geometric and textural priors for grouping 3D primitives (step A). Corresponding posed masks are generated using SAM. An affinity matrix is then computed based on these 2D-3D results serving as edge weights (step B). Region growing and instance-aware refinement are conducted on the constructed scene graph, utilizing 3D box constraint to address over-segmentation while maintaining the fine-grained outcomes (step C).

contains the node features and edge weights. Finally, we perform affinity- and distance-aware region growing to merge the 3D primitives. Further merging is introduced by considering the supplemental prior from 3D space, which is implemented using a 3D detector.

Complementary primitives generation. Following the graph-cut algorithm in [39], we calculate complementary primitives by employing over-segmentation on both geometric and textural priors. Previous methods [7, 8] only consider the geometry information during the primitive generation process. As shown in the middle example in Fig. 2-A, under-segmentation occurs in regions where the door, wall, and board exhibit similar normals. Errors at this initial stage propagate and accumulate, adversely affecting the final segmentation. In contrast, we propose to incorporate additional textural prior, as illustrated in the right example in Fig. 2-A, which leads to finer-grained primitives. Specifically, for a 3D scene P, we first treat each point $p_a \in P$ as a node v_a and calculate the edge weights $w(v_a, v_b)$ for each pair of nodes v_a and v_b. We begin by estimating the normal $\mathbf{n}_a$ for all p_a using corresponding 3D coordinates f_a. Next, we extract the additional color information c_a, which previous methods fail to exploit. Note that the combination of $f_a \in \mathbb{R}^3$ and $c_a \in \mathbb{R}^3$ represents the complete point $p_a \in \mathbb{R}^6$. We compute the cosine similarity between normals $\mathbf{n}_a$ and $\mathbf{n}_b$, and normalized Euclidean distance between colors c_a and c_b. The final edge weights $w(v_a, v_b)$ are obtained by a weighted sum of these two dissimilarities:

$$w(v_a, v_b) = w_n \cdot \frac{\mathbf{n}_a \cdot \mathbf{n}_b}{\|\mathbf{n}_a\| \|\mathbf{n}_b\|} + w_c \cdot \sqrt{\sum_{k=1}^{3} (c_{ak} - c_{bk})^2}. \tag{1}$$

Subsequently, we cluster points to the finer-grained primitives $\{U_i\}_{i=1}^{N_U}$ based on each pair of $w(v_a, v_b)$.

Scene graph construction. As shown in Fig. 2-B, we follow the paradigm to construct a superpoint graph for the given scene. The generated primitives serve as nodes and the affinity scores obtained through a matching algorithm serve as edge weights. Specifically, we first obtain the 2D projection $U_i^m \in \mathbb{R}^{H \times W \times 2}$ of the i-th 3D primitive $U_i \in \mathbb{R}^{N_i \times 3}$ on the m-th image by utilizing the common pinhole camera matrix:

$$(U_i^m, 1)^T = K_m \cdot E_m \cdot (U_i, 1)^T. \tag{2}$$

We feed the m-th RGB-image I_m into the 2D foundation segmentator, e.g. SAM, to obtain its masks S_m. The primitive-mask matching algorithm is then performed on 2D projections U_i^m and the masks S_m for computing affinity scores. To be specific, we calculate a normalized histogram vector $\mathbf{e}_{i,m}$ to collect the 2D masks in S_m covered by rendered U_i^m, since multiple labels may be covered due to the ambiguity or inaccuracy in 2D masks. The affinity score between i- and j-th superpoints on the m-th frame is obtained by computing the cosine similarity of their histogram vectors:

$$A_{i,j}^m = \frac{\mathbf{e}_{i,m} \cdot \mathbf{e}_{j,m}}{\|\mathbf{e}_{i,m}\| \|\mathbf{e}_{j,m}\|}. \tag{3}$$

Traversing all M images yields all affinity scores between U_i and U_j. However, primitives may not be visible in all frames, which leads to invalid affinity scores. To address this, we apply a visibility-based filter on the obtained scores. The visibility $\mathbb{V}_i^m \in (0, 1)$ is defined as the ratio of the visible point number of U_i on m-frame to the total point number of that in the scene. Thus, the final affinity score A_{ij} is calculated in a weighted-sum manner as:

$$A_{i,j} = \frac{\sum_{m=1}^{M} \left(\gamma_{i,j}^m A_{i,j}^m\right)}{\sum_{m=1}^{M} \gamma_{i,j}^m}, \tag{4}$$

where the weight $\gamma_{i,j}^m$ is calculated as the product of $\mathbb{V}_i^m$ and $\mathbb{V}_j^m$. Iteratively processing through all superpoints and 2D frames yields the adjacency matrix $A \in \mathbb{R}^{N_U \times N_U}$. Thus, the superpoint graph of the scene is constructed with primitives $\{U_i\}_{i=1}^{N_U}$ as nodes and adjacency matrix A as edge weights.

Region growing and instance-aware refinement. We perform affinity- and distance-aware region growing on the constructed graph. Previous methods inherit the part-level segmentation tendency from 2D masks output by SAM, often leading to over-segmentation in 3D space. For instance, the chair of the primary segmentation shown in Fig. 2-C is segmented as two distinct parts. To address this issue, we propose to exploit supplemental prior from 3D space by the integration of a 3D detector [40–42] for further merging. As shown in Fig. 2-Output, the constraint provided by additional 3D prior rectifies the over-segmented instances while preserving the capability to handle detailed objects.

Algorithm 1 Instance-aware refinement

```
Input: bounding boxes $\{bb_i\}_{i=1}^{N_{bb}}$ in ascending order of volume, primary 3D segmentation masks $l \in \mathbb{R}^N$, threshold $\delta_2$
Output: Updated 3D segmentation $l'$
$O_i \leftarrow \emptyset$, $l' \leftarrow l$
$L \leftarrow$ {max instance ID in $l$} + 1
for each $bb_i$ in $\{bb_i\}_{i=1}^{N_{bb}}$ do
    $O_i \leftarrow l \cap bb_i$
    for each instance ID $O'_i$ in $O_i$ do
        $\sigma_i \leftarrow \frac{\text{points in } O'_i}{\text{points in } \{l'=O'_i\}}$
        if $\sigma_i > \delta_2$ then
            $\{l' = O'_i\} \leftarrow L$
        end if
    end for
    $L \leftarrow L + 1$
end for
```

At the primary merging stage, we incorporate not only the affinity scores $A_{i,j}$ but also the Euclidean distances $\mathbb{D}_{i,j}$ between nodes U_i and U_j, thus to introduce a certain level of global awareness. Dynamic thresholds $\delta_1 \in \mathbb{R}^{N_t}$ are applied to reduce the initial erroneous merges and subsequent error accumulation. Specifically, we multiply $A_{i,j}$ with decay factor $\epsilon_{\mathbb{D}}$ to get the merging confident score $\delta_{i,j}$ between U_i and U_j:

$$\delta_{i,j} = \epsilon_{\mathbb{D}} \cdot A_{i,j} = \frac{1}{\mathbb{D}_{i,j}} \cdot A_{i,j}, \tag{5}$$

where $\epsilon_{\mathbb{D}}$ is the reciprocal of the Euclidean distance. Then we compare the confident score $\delta_{i,j}$ with the threshold δ_1 to judge whether to merge the nodes U_i and U_j. Therefore, the primary segmentation results are obtained through iterating all pairs of nodes N_t times.

We further introduce supplemental prior from 3D space by employing a detection-based instance-aware refinement. As shown in Algorithm 1, we gather all points O_i within the bounding box bb_i, and assess the proportion σ_i of points belong to instance ID O'_i in O_i to that of the entire scene P. If the ratio σ_i exceeds a specified threshold δ_2, it indicates high confidence that the points with instance ID O'_i represent a portion of an over-segmented instance. We assign a new label to all points that exceed the threshold, thereby rectifying the over-segmented instance. However, there is a possibility that smaller objects which are in close proximity to or situated on larger objects are false-corrected. To

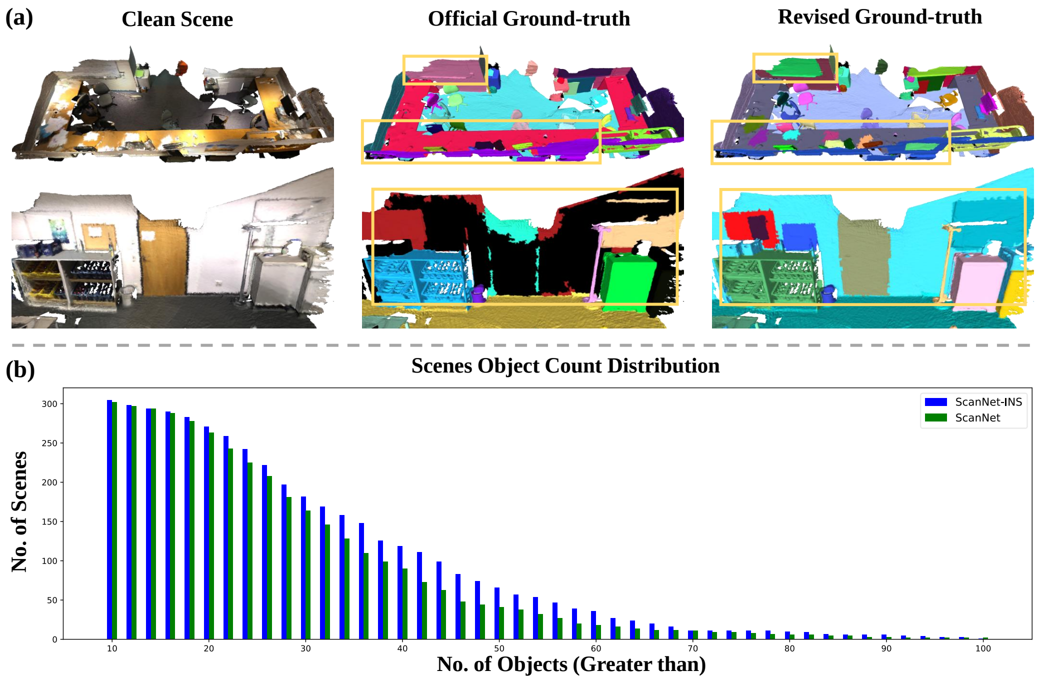

Figure 3: Overview of our proposed ScanNetV2-INS. We present the new benchmark for 3D class-agnostic instance segmentation, which rectifies incomplete annotations and incorporates more instances based on ScanNetV2. Row (a) shows the comparison before and after revision, and row (b) illustrates the object counts per scene between the two benchmarks.

mitigate this issue, we opt to pre-sort the bounding boxes based on size. In this way, the corrections are performed in descending order of bounding box size to ensure that smaller objects retain their independence in the final output.

3.2 ScanNetV2-INS

ScanNetV2 has served as a standard benchmark for evaluating model performance. However, it includes a notable proportion of low-quality ground truth labels, potentially leading to misleading results. To address this issue, we introduce a refined version of the dataset, termed ScanNetV2-INS, wherein annotations are enhanced at the point level.

Imperfection in vanilla ScanNetV2. The original ScanNetV2 exhibits imperfections in its ground truth annotations, primarily manifesting in two aspects. Firstly, certain obvious instances remain unmarked. For example, as illustrated in Fig. 3-a top row, the board on the wall and papers on the desk are neglected. Secondly, some instances are incompletely annotated. For instance, doors and boards in Fig. 3-a bottom row that are clearly visible in clean point clouds feature large areas of black (indicating "unlabeled") in the annotations. The prevalence of these significantly impacts the accuracy of evaluation metrics and leads to erroneous estimations of model performance. Thus, corrective measures are imperative.

Revision of ScanNetV2. With the aid of a recently released annotation tool AGILE3D proposed in [43], we perform point-level updates on the ground truth annotations for all 312 scenes in the validation set efficiently. The revision primarily addresses two aforementioned deficiencies, as shown in Fig. 3-a right column. Firstly, we re-label the instances where the ground truth was obscured by unlabeled black points, such as the door and boards. Secondly, we assign class-agnostic labels to certain instances that were clearly discernible to the human perception but were not originally annotated in the ground truth, such as the papers on the desk and the poster on the wall.

Table 1: Instance number within varying point range of ScanNetV2 and ScanNetV2-INS dataset.

Dataset	Point number of the instance					
	<500	500-1000	1000-2000	2000-5000	5000-10000	>10000
ScanNetV2	252	452	1119	1690	567	284
ScanNetV2-INS	692	748	1366	1873	626	291

Table 2: Instance count of ScanNetV2 and ScanNetV2-INS dataset.

Dataset	Instance Count			
	Min	Max	Avg	Total
ScanNetV2	2	47	14	4364
ScanNetV2-INS	2	54	17	5596

Statistic analysis and limitation of ScanNetV2-INS. Fig. 3-b demonstrates how many scenes hold more than (10, 20, ..., 100) instances. In Tab. 1 we show the number of instances with varying point counts within specified ranges for two datasets. ScanNetV2-INS dataset features more smaller objects, which requires the model to have finer-grained instance perception capabilities. As a result, the instance count of our new dataset, as shown in Tab. 2, significantly increases. Therefore, it better reflects and poses more challenges on the model performance. However, our dataset consists of only the revised version of 312 scenes in the validation set, focusing on the evaluation use of 3D class-agnostic instance segmentation in the context of no-training methods.

4 Experiments

In this section, we quantitatively evaluate our SA3DIP on ScanNet series (including the vanilla ScanNetV2 [11], our ScanNetV2-INS, and more challenging ScanNet++ [12]), Matterport3D [44] and Replica [45] datasets to demonstrate its effectiveness and robustness in 3D instance segmentation. Qualitative visualizations for ScanNet series datasets are also provided for a more intuitive comparison with other methods.

4.1 Experiment settings

Datasets. ScanNet [11] integrates a comprehensive array of 2D and 3D data sourced from indoor environments, facilitated by an iPad application in tandem with depth sensors. This dataset includes RGB and depth images, along with 3D point cloud data, all meticulously annotated with semantic and instance labels. It encompasses an extensive collection of over 2.5 million views derived from more than 1500 scans. In contrast, ScanNet++ [12] represents a recently introduced indoor dataset exhibiting a similar composition to ScanNet but boasting higher-resolution 3D geometry and more detailed data annotations. ScanNet++ data is captured using advanced equipment, including the Faro Focus Premium laser scanner, iPhone 13 Pro, and a DSLR camera equipped with a fisheye lens. Our proposed ScanNet-INS encompasses a revised version of all 312 validation scenes in ScanNetV2, while maintaining consistency with ScanNetV2 in terms of data and label format. It provides a more accurate metric and fairer comparison between methods.

Parameter settings. We conduct all experiments on a single RTX4090. The weights for geometry and texture used in the complementary primitives generation are set as $w_n = 0.96$ and $w_c = 0.04$. This is because texture prior such as RGB values are not robust enough when being used solely due to lighting conditions, reflections, shadows, and noise collected by sensors. We conduct detailed ablation study on the choice of the two weights in the later section. The threshold δ_1 in the region growing is empirically set as $[0.9, 0.8, 0.7, 0.6, 0.5]$ for ScanNetV2 and ScanNetV2-INS, $[0.9, 0.8, 0.7]$ for ScanNet++, and the threshold δ_2 in instance-aware refinement is set as 0.75 experimentally.

Evaluation metrics. We evaluate the quantitative results with the widely used Average Precision score. Following [2, 7, 8, 25], we report AP with thresholds of 25% and 50% (denoted as $\mathbf{AP}_{25}$, $\mathbf{AP}_{50}$) and averaged over all overlaps between [50% and 95%] at 5% steps ($\mathbf{mAP}$). Since the 2D foundation segmentation model produces class-agnostic masks, we ignore semantic class labels in the evaluation and consider only the accuracy of the instance masks themselves.

Table 3: Class-agnostic 3D instance segmentation comparison on ScanNetV2, ScanNetV2-INS, and ScanNet++ datasets.

Method	ScanNetV2			ScanNetV2-INS			ScanNet++		
	mAP	**AP**$_{50}$	**AP**$_{25}$	**mAP**	**AP**$_{50}$	**AP**$_{25}$	**mAP**	**AP**$_{50}$	**AP**$_{25}$
Closed-vocabulary									
Mask3D [2]	31.1	44.9	58.0	29.1	43.9	56.3	9.9	17.3	25.8
Open-vocabulary									
Felzenszwalb [39]	5.0	12.7	38.9	2.8	6.5	24.0	4.1	9.2	25.3
SAM3D [10] (w/o ensemble)	12.4	28.7	57.4	12.5	28.9	57.8	1.1	4.5	15.4
SAM3D [10] (w/ ensemble)	20.1	33.3	52.1	20.0	33.2	52.2	7.2	14.2	29.4
SAM-graph [7]	24.1	40.3	65.9	23.1	39.5	64.1	12.9	25.3	43.6
SAI3D [8]	30.8	50.5	70.6	28.9	49.2	69.7	17.1	31.1	49.5
SAMPro3D [9]	33.7	56.2	75.3	32.5	54.8	73.4	18.9	33.7	51.6
SA3DIP (ours)	**41.6**	**64.6**	**81.3**	**36.1**	**58.6**	**76.3**	**21.4**	**36.4**	**53.6**

Baselines. We compare our approach with both closed-vocabulary and open-vocabulary methods. Mask3D [2] trained on ScanNetV2 serves as the closed-vocabulary baseline. Recent methods based on leveraging 2D foundation models, including SAM3D [10] (with and without ensemble process), SAM-graph [7], SAI3D [8], and SAMPro3D [9] are compared as open-vocabulary methods. In addition, we compare with the traditional point grouping method proposed by Felzenszwalb [39].

4.2 Results on ScanNet series

Tab. 3 shows the quantitative results of our approach in comparison with other methods on ScanNetV2, ScanNetV2-INS, and ScanNet++ datasets. Our method achieves the best performance among all three datasets, showing the effectiveness of our approach. Specifically, our SA3DIP outperform 7.9% $\mathbf{mAP}$, 8.4% $\mathbf{AP}_{50}$, and 6.0% $\mathbf{AP}_{25}$ on ScanNetV2, 3.6% $\mathbf{mAP}$, 3.8% $\mathbf{AP}_{50}$, and 2.9% $\mathbf{AP}_{25}$ on ScanNetV2-INS. On the challenging ScanNet++, our method still obtains 2.5% $\mathbf{mAP}$, 2.7% $\mathbf{AP}_{50}$, and 2.0% $\mathbf{AP}_{25}$ gain. Note that all methods except SAM3D experience a drop in precision on ScanNetV2-INS dataset compared with the vanilla ScanNetV2. This indicates that our proposed ScanNetV2-INS poses more challenges in identifying fine-grained objects and yields fairer metrics. SAM3D, due to its limited segmentation capability, tends to produce a significantly higher number of instances than the actual objects in the scene. Consequently, its metrics do not show noticeable changes on finer-grained ScanNetV2-INS.

We also present qualitative results in Fig. 4. The visual comparison further proves the effectiveness of our method. As shown in the first two rows of Fig. 4, our method maintains a better instance awareness and is capable of identifying the tables as a whole. Moreover, by utilizing more accurate 3D primitives, our approach is the only one able to segment the door out from the wall, as shown in the third row of Fig. 4. This showcases the significance of exploiting the potential 3D priors.

4.3 Ablation studies

We conduct detailed ablation studies on prior with varying weights w_n and w_c. We report the metrics in Tab. 4. We assigned several weights for geometry and texture to test their contribution. Specifically, we conduct one with configuration of $w_n = 0.4$ and $w_c = 0.6$ which yields the similar number of 3D primitives as the primitives used in SAM3D, SAI3D and others, for a fair comparison.

It can be observed that texture prior are not robust enough when being used solely due to the influence of shadows, reflection and so on. Therefore, we choose to assign less weight to the textural prior, thus to exploit it while minimizing its negative impact. The experiments show that the setting with $w_n = 0.96$ and $w_c = 0.04$ suits best for our approach.

However, it is noticed that incorporating only complementary primitives yields a slight drop in average precision on both datasets. It is related to the definition of the metric AP (ratio of correctly

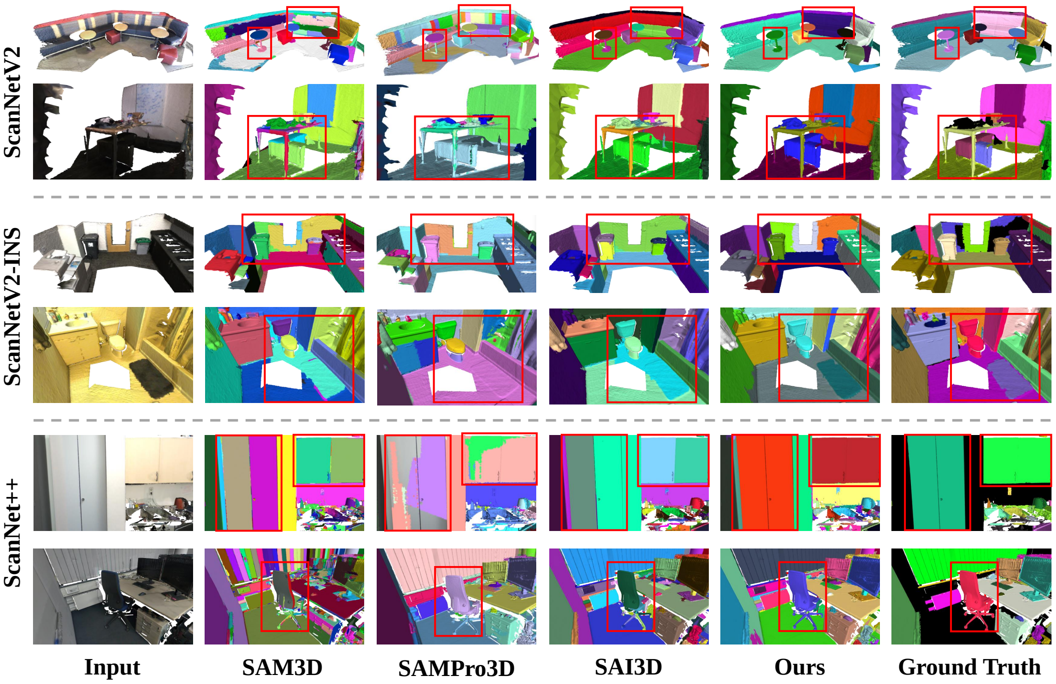

Figure 4: Visual comparison between our method with SAM3D [10], SAMPro3D [9], and SAI3D [8] on ScanNetV2, ScanNetV2-INS, and ScanNet++ dataset. Among all datasets, our method shows the most robust and accurate segmentation.

Table 4: Ablation studies on prior we exploit with varying weights.

w_n	w_c	3D Space Prior	ScanNetV2			ScanNetV2-INS		
			mAP	**AP$_{50}$**	**AP$_{25}$**	**mAP**	**AP$_{50}$**	**AP$_{25}$**
1	0	×	30.8	50.5	70.6	28.9	49.2	69.7
0	1	×	10.4	18.1	32.5	9.5	17.0	31.1
0.4	0.6	×	27.3	47.4	69.8	25.6	46.3	69.4
0.96	0.04	×	29.3	49.2	70.5	27.4	48.3	70.4
1	0	✓	40.8	63.6	80.7	35.9	57.8	75.4
0	1	✓	12.7	22.1	37.2	11.0	19.7	34.1
0.4	0.6	✓	39.1	62.7	80.2	33.5	56.3	75.0
0.96	0.04	✓	**41.6**	**64.6**	**81.3**	**36.1**	**58.6**	**76.3**

identified instances to the total number of identified instance). This AP metric is more in favor of under-segmentation rather than over-segmentation, since the former yields high precision and fewer false positive cases, while the latter gives fewer precision and higher recall.

4.4 More experiments

We conduct further experiments and corresponding ablation studies on both Matterport3D [44] and Replica [45] dataset to test the robustness and generalization ability of our method. Matterport3D dataset contains 194,400 RGB-D images of 90 building-scale indoor scenes and exhibits more view changes on its 2D frames compared to ScanNet. Replica dataset incorporates 18 highly photo-realistic 3D indoor scene reconstructions with dense geometry, high resolution and dynamic range textures. Our method clearly gives better quantitative results, as shown in Tab. 5.

Table 5: Experiments on Matterport3D and Replica datasets.

	w_n	w_c	3D Space Prior	**AP**	**AP**$_{25}$	**AP**$_{50}$
Matterport3D dataset [44]						
OpenMask3D [25]	/	/	×	15.3	28.3	43.3
OVIR-3D [37]	/	/	×	6.6	15.6	28.3
SAM3D w/ ensemble [10]	1	0	×	10.1	19.4	36.1
SAI3D [8]	1	0	×	18.9	35.6	56.5
Ablation of ours						
Ours #1	1	0	✓	19.8	36.6	56.2
Ours #2	0.9	0.1	×	18.1	35.7	**62.3**
Ours #3	0.9	0.1	✓	**20.6**	**38.3**	*61.0*
Replica dataset [45]						
SAM3D w/o ensemble [10]	/	/	×	11.9	22.9	38.4
SAM3D w/ ensemble [10]	1	/	×	12.4	20.0	32.0
SAMPro3D [9]	1	0	×	13.1	25.2	44.7
SAI3D [8]	1	0	×	20.4	30.7	42.9
Ablation of ours						
Ours #1	1	0	✓	21.0	31.5	43.6
Ours #2	0.9	0.1	×	20.8	32.8	46.2
Ours #3	0.9	0.1	✓	**22.6**	**34.2**	**47.1**

4.5 Limitations

Due to the trade-off between efficiency and accuracy, we choose to compute 3D superpoints based on only 3D priors. This yields an extremely short time of execution within a few seconds, while it may lead to an overwhelming number of superpoints which introduces challenges in the merging process. Moreover, for high-resolution point clouds with vivid light and shade effects, the superpoints generated based on geometric and texture is not enough yet. One approach is to design a more sophisticated pre-segmentation model with semantic awareness. Besides, though constraints provided by 3D prior are introduced, the affinity matrix based on 2D masking still relies heavily on the accuracy of 2D foundation segmentators. Designing a more robust merging algorithm or better leveraging various 2D foundation models shows promise in the future.

5 Conclusion

In this paper, we introduce a novel method for segmenting any 3D instances by exploiting the potential 3D priors. The key idea is to incorporate more 3D priors into the 2D foundation model guided pipeline and leverage not only knowledge transferred from 2D space but also features in 3D space. We first generate complementary 3D superpoint primitives based on both geometric and textural priors to reduce the initial errors that accumulate in subsequent procedures. Then we introduce supplemental constraints from the 3D space by using a 3D detector. Along with the constructed affinity matrix by using 2D masks, the region growing and refinement process is performed on the 3D primitives. Furthermore, we propose ScanNetV2-INS with complete ground truth labels and supplement additional instances for 3D class-agnostic instance segmentation, which produces unbiased metrics on comparing different methods. Experimental evaluations on ScanNetV2, ScanNetV2-INS, and ScanNet++ datasets demonstrate the effectiveness of our approach. We believe that we pioneer at exploiting the importance of 3D priors in the 2D foundation model guided pipeline, and it should draw attention toward future research that methods trying to extend 2D foundation models into 3D space should not overlook the role of inherent 3D priors.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62372348, Grant 62441601, Grant 62176195, Grant U21A20514; in part by the Key Research and Development Program of Shaanxi under Grant 2024GX-ZDCYL-02-10; in part by Shaanxi

Outstanding Youth Science Fund Project under Grant 2023-JC-JQ-53; in part by the Fundamental Research Funds for the Central Universities under Grant QTZX24080.

References

[1] Sangyun Shin, Kaichen Zhou, Madhu Vankadari, Andrew Markham, and Niki Trigoni. Spherical mask: Coarse-to-fine 3d point cloud instance segmentation with spherical representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

[2] Jonas Schult, Francis Engelmann, Alexander Hermans, Or Litany, Siyu Tang, and Bastian Leibe. Mask3d: Mask transformer for 3d semantic instance segmentation. In *Proceedings of the International Conference on Robotics and Automation*, pages 8216–8223, 2023.

[3] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4015–4026, 2023.

[4] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, pages 8748–8763, 2021.

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021.

[6] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*, 2023.

[7] Haoyu Guo, He Zhu, Sida Peng, Yuang Wang, Yujun Shen, Ruizhen Hu, and Xiaowei Zhou. Sam-guided graph cut for 3d instance segmentation. *arXiv preprint arXiv:2312.08372*, 2023.

[8] Yingda Yin, Yuzheng Liu, Yang Xiao, Daniel Cohen-Or, Jingwei Huang, and Baoquan Chen. Sai3d: Segment any instance in 3d scenes. *arXiv preprint arXiv:2312.11557*, 2023.

[9] Mutian Xu, Xingyilang Yin, Lingteng Qiu, Yang Liu, Xin Tong, and Xiaoguang Han. Sampro3d: Locating sam prompts in 3d for zero-shot scene segmentation. *arXiv preprint arXiv:2311.17707*, 2023.

[10] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. *arXiv preprint arXiv:2306.03908*, 2023.

[11] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE/CVF Conference Computer Vision and Pattern Recognition*, 2017.

[12] Chandan Yeshwanth, Yueh-Cheng Liu, Matthias Nießner, and Angela Dai. Scannet++: A high-fidelity dataset of 3d indoor scenes. In *Proceedings of the International Conference on Computer Vision*, 2023.

[13] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[14] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, volume 30, 2017.

[15] Benjamin Graham, Martin Engelcke, and Laurens Van Der Maaten. 3d semantic segmentation with submanifold sparse convolutional networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9224–9232, 2018.

[16] Mutian Xu, Runyu Ding, Hengshuang Zhao, and Xiaojuan Qi. Paconv: Position adaptive convolution with dynamic kernel assembling on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3173–3182, 2021.

[17] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. In *Advances in Neural Information Processing Systems*, volume 35, pages 23192–23204, 2022.

[18] Xingyilang Yin, Xi Yang, Liangchen Liu, Nannan Wang, and Xinbo Gao. Point deformable network with enhanced normal embedding for point cloud analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6738–6746, 2024.

[19] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 16259–16268, 2021.

[20] Xiaoyang Wu, Yixing Lao, Li Jiang, Xihui Liu, and Hengshuang Zhao. Point transformer v2: Grouped vector attention and partition-based pooling. In *Advances in Neural Information Processing Systems*, volume 35, pages 33330–33342, 2022.

[21] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler, faster, stronger. *arXiv preprint arXiv:2312.10035*, 2023.

[22] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, pages 3075–3084, 2019.

[23] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2783–2792, 2021.

[24] Jiahao Sun, Chunmei Qing, Junpeng Tan, and Xiangmin Xu. Superpoint transformer for 3d scene instance segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 2393–2401, 2023.

[25] Ayca Takmaz, Elisabetta Fedele, Robert Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

[26] Hugues Thomas, Charles R Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J Guibas. Kpconv: Flexible and deformable convolution for point clouds. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6411–6420, 2019.

[27] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 15467–15476, 2021.

[28] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, pages 4867–4876, 2020.

[29] Yizheng Wu, Min Shi, Shuaiyuan Du, Hao Lu, Zhiguo Cao, and Weicai Zhong. 3d instances as 1d kernels. In *European Conference on Computer Vision*, pages 235–252, 2022.

[30] Xin Lai, Yuhui Yuan, Ruihang Chu, Yukang Chen, Han Hu, and Jiaya Jia. Mask-attention-free transformer for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3693–3703, 2023.

[31] Jiahao Lu, Jiacheng Deng, Chuxin Wang, Jianfeng He, and Tianzhu Zhang. Query refinement transformer for 3d instance segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18516–18526, 2023.

[32] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern recognition*, pages 4421–4430, 2019.

[33] Maksim Kolodiazhnyi, Anna Vorontsova, Anton Konushin, and Danila Rukhovich. Top-down beats bottom-up in 3d instance segmentation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3566–3574, 2024.

[34] Youquan Liu, Lingdong Kong, Jun Cen, Runnan Chen, Wenwei Zhang, Liang Pan, Kai Chen, and Ziwei Liu. Segment any point cloud sequences by distilling vision foundation models. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

[35] Runyu Ding, Jihan Yang, Chuhui Xue, Wenqing Zhang, Song Bai, and Xiaojuan Qi. Pla: Language-driven open-vocabulary 3d scene understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7010–7019, 2023.

[36] Jihan Yang, Runyu Ding, Weipeng Deng, Zhe Wang, and Xiaojuan Qi. Regionplc: Regional point-language contrastive learning for open-world 3d scene understanding. *arXiv preprint arXiv:2304.00962*, 2023.

[37] Shiyang Lu, Haonan Chang, Eric Pu Jing, Abdeslam Boularias, and Kostas Bekris. Ovir-3d: Open-vocabulary 3d instance retrieval without training on 3d data. In *Proceedings of the Conference on Robot Learning*, pages 1610–1620, 2023.

[38] Songyou Peng, Kyle Genova, Chiyu Jiang, Andrea Tagliasacchi, Marc Pollefeys, Thomas Funkhouser, et al. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 815–824, 2023.

[39] Pedro F Felzenszwalb and Daniel P Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59:167–181, 2004.

[40] Yichao Shen, Zigang Geng, Yuhui Yuan, Yutong Lin, Ze Liu, Chunyu Wang, Han Hu, Nanning Zheng, and Baining Guo. V-detr: Detr with vertex relative position encoding for 3d object detection. In *International Conference on Learning Representations*, 2023.

[41] 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/CVF conference on computer vision and pattern recognition*, pages 1190–1199, 2023.

[42] Yang Cao, Zeng Yihan, Hang Xu, and Dan Xu. Coda: Collaborative novel box discovery and cross-modal alignment for open-vocabulary 3d object detection. In *Advances in Neural Information Processing Systems*, volume 36, 2024.

[43] Yuanwen Yue, Sabarinath Mahadevan, Jonas Schult, Francis Engelmann, Bastian Leibe, Konrad Schindler, and Theodora Kontogianni. Agile3d: Attention guided interactive multi-object 3d segmentation. In *International Conference on Learning Representations*, 2024.

[44] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. *arXiv preprint arXiv:1709.06158*, 2017.

[45] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

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 research topic of our work is 3D instance segmentation by exploit additional 3D priors. In the introduction, we have clearly presented the existing problems in this field and outlined our contributions.

   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: We have discussed the limitation and promising solutions of our method in Section 4.4.

   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: There is no theory assumptions or theoretical results in this paper.

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: We present all the information needed to reproduce the results. The formulas and experimental settings are listed in Section 3.1 and 4.1.

   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 code and newly proposed dateset will be available.

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 experimental settings are listed in Section 4.1.

   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: The overall pipeline described in the paper requires no training, thus the experimental results are consistent for each time.

   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 compute resource used is described in Section 4.1.

   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: The research conducted in the paper conform 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 discuss the broader impact at the end of the conclusion section.

    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: Our research conducts experiment on widely-used datasets. Therefore, it 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 properly credited the codes and data used in this paper at the reference section.

    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: [Yes]

    Justification: The proposed method and dataset is well documented as Section 3.1, 3.2, and 4.1.

    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: Our 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: Our 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.