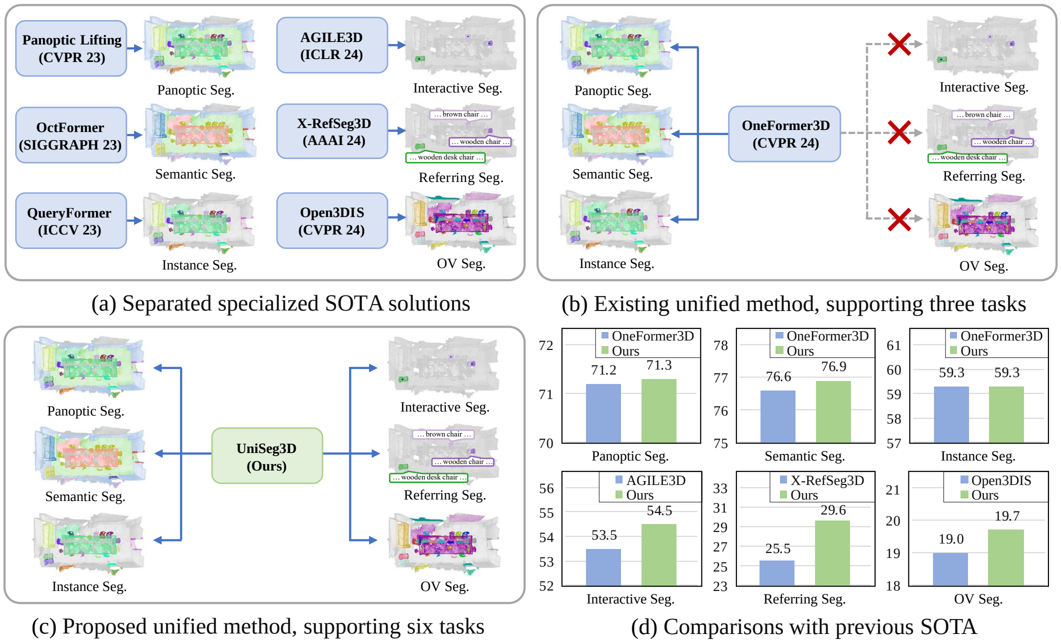
A Unified Framework for 3D Scene Understanding

Wei Xu*, **Chunsheng Shi***, **Sifan Tu**, **Xin Zhou**,
Dingkang Liang, **Xiang Bai**†
Huazhong University of Science and Technology
`{wxu2023, csshi, dkliang, xbai}@hust.edu.cn`



(a) Separated specialized SOTA solutions

(b) Existing unified method, supporting three tasks

(c) Proposed unified method, supporting six tasks

(d) Comparisons with previous SOTA

Figure 1: Comparisons between the proposed method and current SOTA approaches specialized for specific tasks. (a) Representative specialized approaches on six tasks. (b) OneFormer3D, a recent unified framework, achieves SOTA performance on three generic segmentation tasks in one inference. (c) The proposed unified framework achieves six tasks in one inference. (d) Our method outperforms current SOTA approaches across six tasks involving two modalities using a single model.

Abstract

We propose UniSeg3D, a unified 3D scene understanding framework that achieves panoptic, semantic, instance, interactive, referring, and open-vocabulary segmentation tasks within a single model. Most previous 3D segmentation approaches are typically tailored to a specific task, limiting their understanding of 3D scenes to a task-specific perspective. In contrast, the proposed method unifies six tasks into unified representations processed by the same Transformer. It facilitates inter-task knowledge sharing, thereby promoting comprehensive 3D scene understanding. To take advantage of multi-task unification, we enhance performance by establishing explicit inter-task associations. Specifically, we design knowledge distillation and contrastive learning to transfer task-specific knowledge across different tasks. Ex-

*Equal contribution. †Corresponding author.

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

https://doi.org/10.52202/079017-1899

periments on three benchmarks, including ScanNet20, ScanRefer, and ScanNet200, demonstrate that the UniSeg3D consistently outperforms current SOTA methods, even those specialized for individual tasks. We hope UniSeg3D can serve as a solid unified baseline and inspire future work. Code and models are available at https://dk-liang.github.io/UniSeg3D/.

1 Introduction

3D scene understanding has been a foundational aspect of various real-world applications [3, 13, 72, 71, 19], including robotics, autonomous navigation, and mixed reality. Among 3D scene understanding tasks, 3D point cloud segmentation is a crucial component. Generic 3D point cloud segmentation contains panoptic, semantic, and instance segmentation tasks [38, 42, 61, 69, 22], which segment classes annotated in the training set. As a complement, 3D open-vocabulary (OV) segmentation task [41, 54, 16] segments open-vocabulary classes of interest. Another group of works study to utilize user priors. In particular, 3D interactive segmentation task [24, 70] segments instances specified by users. 3D referring segmentation task [15, 45, 59, 58] segments instances described by textual expressions. The above-mentioned tasks are core tasks in 3D scene understanding, drawing significant interest from researchers and achieving great success.

Previous studies [53, 8, 75, 30, 20] in the 3D scene understanding area focus on separated solutions specialized for specific tasks, as shown in Fig. 1(a). These approaches overlook intrinsic connections across different tasks, such as geometric and semantic consistency of objects. They also fail to share knowledge biased toward other tasks, limiting their understanding of 3D scenes to task-specific perspectives. It poses significant challenges for achieving comprehensive and in-depth 3D scene understanding. A recent exploration [23] named OneFormer3D designs an architecture to unify the 3D generic segmentation tasks, as shown in Fig. 1(b). This architecture inputs instance and semantic queries to simultaneously predict the 3D instance and semantic segmentation results. And the 3D panoptic segmentation is subsequently achieved by post-processing these predictions. It is simple yet effective. However, this architecture fails to support the 3D interactive, referring, and open-vocabulary segmentation tasks, which provide complementary scene information, including user priors and open-set classes, should be equally crucial in achieving 3D scene understanding as the generic segmentation tasks. This leads to a natural consideration that *if these 3D scene understanding tasks can be unified in a single framework?*

A direct solution is to integrate separated methods into a single architecture. However, it faces challenges balancing customized optimizations specialized for specific tasks involved in these methods. Therefore, we aim to design a simple and elegant framework without task-specific customized modules. This inspires us to design UniSeg3D, a unified framework processing six 3D segmentation tasks in parallel. Specifically, we use queries to unify representations of six tasks. The 3D generic and open-vocabulary segmentation tasks, only input point clouds without human knowledge, can be processed by sharing the same workflow without worrying about prior knowledge leakage. We use a unified set of queries to extract features of these four tasks for simplification. The interactive segmentation inputs visual point prompts to condition the segmentation. We represent the point prompt information by sampling point cloud features as vision prompt queries, thereby avoiding repeated point feature extraction. The referring segmentation inputs textual expressions persist in a modality gap with point clouds and are hard to unify in previous workflows. To reduce time consumption, we employ a parallel text prompt encoder to extract text features and regard them as text prompt queries. All these queries are decoded using the same mask decoder and share the same output head without the design of task-specific customized structures.

We further enhance performance by taking advantage of the multi-task design. In particular, we empirically find that the interactive segmentation outperforms the rest of the tasks in mask predictions, which is attributable to reliable vision priors. Hence, we design knowledge distillation to distill knowledge from the interactive segmentation to the other tasks. Then, we build contrastive learning between interactive segmentation and referring segmentation to connect these two tasks. The proposed knowledge distillation and contrastive learning promote knowledge sharing across six tasks, effectively establishing inter-task associations. There are three significant strengths of the UniSeg3D: (1) It unifies six 3D scene understanding tasks in a single framework, as shown in Fig. 1(c). (2) It is flexible because it can be easily extended to more tasks by simply feeding additional task-specific queries. (3) The designed knowledge distillation and contrastive learning are only used in the training phase, optimizing performance with no extra inference cost.

We compare the proposed method with task-specific specialized SOTA approaches [50, 56, 36, 70, 45, 40] across six tasks to evaluate its performance. As shown in Fig. 1(d), the UniSeg3D demonstrates superior performance on all the tasks. It is worth noting that our performance on different tasks is achieved by a single model, which is more efficient than running separate task-specific approaches individually. Furthermore, the structure of UniSeg3D is simple and elegant, containing no task-customized modules, while consistently outperforming specialized SOTA solutions, demonstrating a desirable potential to be a solid unified baseline.

Our contributions can be summarized as follows: **First**, we propose a unified framework named UniSeg3D, offering a flexible and efficient solution for 3D scene understanding. It achieves six 3D segmentation tasks in one inference by a single model. To our knowledge, this is the first work to unify six 3D segmentation tasks. **Second**, specialized approaches limit their 3D scene understanding to task-specific perspectives. We facilitate inter-task knowledge sharing to promote comprehensive 3D scene understanding. Specifically, we take advantage of the multi-task unification design, employing knowledge distillation and contrastive learning to establish explicit inter-task associations.

2 Related Work

3D segmentation. The generic segmentation consists of panoptic, semantic, and instance segmentation. The panoptic segmentation [38, 60] is the union of instance segmentation [9, 33, 2, 64, 55] and semantic segmentation [44, 42, 5, 73]. It contains instance masks from the instance segmentation and stuff masks from the semantic segmentation. These 3D segmentation tasks rely on annotations, segmenting classes labeled in the training set. The open-vocabulary segmentation [40, 54] extends 3D segmentation to novel classes. Another group of works explores 3D segmentation conditioned by human knowledge. Specifically, the interactive segmentation [24, 70] segments instances specified by the point clicks. The referring segmentation [15, 45, 57] segments objects described by textual expressions. Most previous researches [65, 4, 25, 32] focus on specific 3D segmentation tasks, limiting their efficiency in multi-task scenarios, such as the domotics, that require multiple task-specific 3D segmentation approaches to be applied simultaneously. This work proposes a framework to achieve the six above-mentioned tasks in one inference.

Unified vision models. Unified research supports multiple tasks in a single model, facilitating efficiency and attracting extensive attention in the 2D area [43, 37, 29, 18]. However, rare works study the unified 3D segmentation architecture. It might be attributed to the higher dimension of the 3D data, which leads to big solution space, making it challenging for sufficient unification across multiple 3D tasks. Recent works [11, 35] explore outdoor unified 3D segmentation architectures, and some others [76, 14, 17] delve into unified 3D representations. So far, only one method, OneFormer3D [23], focuses on indoor unified 3D segmentation. It extends the motivation proposed in OneFormer [18] to the 3D area and proposes an architecture to achieve three 3D generic segmentation tasks in a single model. We note that the supported tasks in OneFormer3D can be achieved in one inference through post-processing predictions of a panoptic segmentation model. In contrast, we propose a simple framework that unifies six tasks, including not only generic segmentation but also interactive, referring, and open-vocabulary segmentation, into a single model. Additionally, we establish explicit associations between these unified tasks to promote knowledge sharing, contributing to effective multi-task unification.

3 Methodology

The framework of UniSeg3D is depicted in Fig. 2. It mainly consists of three modules: a point cloud backbone, prompt encoders, and a mask decoder. We illustrate their structures in the following.

3.1 Point Cloud Backbone and Prompt Encoders

Point cloud backbone. We represent a set of N input points as $\mathbf{P} \in \mathbb{R}^{N\times 6}$, where each point is characterized by three-dimensional coordinates x, y, z and three-channel colors r, g, b. These input points are then fed into a sparse 3D U-Net, serving as the point cloud backbone, to obtain point-wise

https://doi.org/10.52202/079017-1899

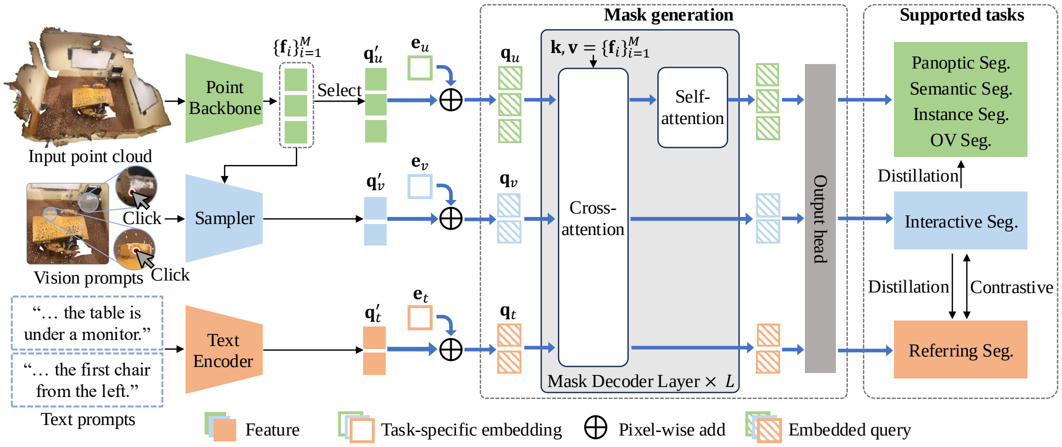


Figure 2: The framework of UniSeg3D. This is a simple framework handling six tasks in parallel without any modules specialized for specific tasks. We take advantage of the multi-task unification design and enhance performance by building associations between the supported tasks. Specifically, knowledge distillation transfers insights from interactive segmentation to the other tasks, while contrastive learning establishes connections between interactive segmentation and referring segmentation.

features $\mathbf{F} \in \mathbb{R}^{N \times d_{in}}$, where d_{in} denotes the feature dimension. Processing dense points individually in 3D scene understanding can be time-consuming. Therefore, we downsample the 3D scenario into M superpoints and pool the point features within each superpoint to form superpoint features $\mathbf{F}_s = \{\mathbf{f}_i\}_{i=1}^M$, where each $\mathbf{f}_i \in \mathbb{R}^{d_{in}}$ and $\mathbf{F}_s \in \mathbb{R}^{M \times d_{in}}$. This procedure exhibits awareness of the edge textures [27] while reducing computation cost.

Vision prompt encoder. Click is a clear and convenient visual interaction condition widely employed in previous works [21, 24, 70]. We formulate clicks as vision prompts, as illustrated in Fig. 2. In practice, a click is first indicated by the spatially nearest point. Then, we sample a superpoint containing this point and employ its superpoint feature as a vision prompt feature $\mathbf{f}_v \in \mathbb{R}^{d_{in}}$ to represent point prompt information, thus avoiding repeated feature extraction and maintaining feature consistency with the point clouds.

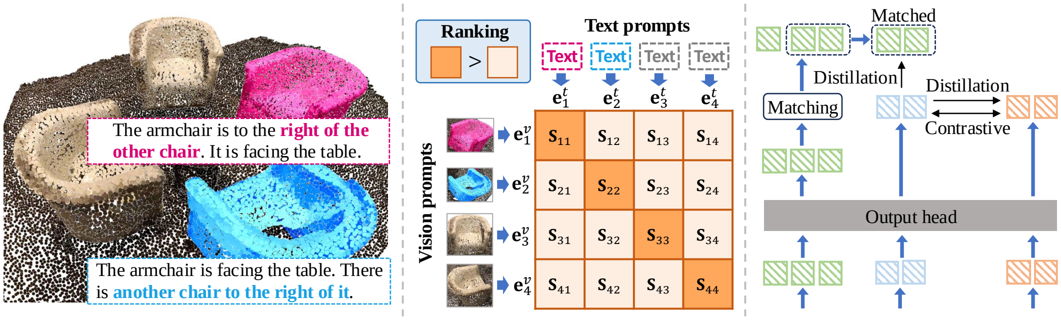
Text prompt encoder. UniSeg3D is able to segment instances described by textual expressions. The initial step of processing a text prompt involves tokenizing the text sentence to obtain its string tokens $\mathbf{T} \in \mathbb{R}^{l \times c}$, where l is the sentence length, and c represents the token dimension. These tokens are then fed into a frozen CLIP [46] text encoder to produce a C-dimensional text feature $\mathbf{f}_t \in \mathbb{R}^C$. This feature is subsequently projected into dimension of d_{in} using two linear layers, obtaining $\mathbf{f}_t \in \mathbb{R}^{d_{in}}$, aligning the dimension of the point features for subsequent processing.

3.2 Mask Generation

We employ a single mask decoder to output predictions of six 3D scene understanding tasks. The generic and open-vocabulary segmentation share the same input data, *i.e.*, the point cloud without user knowledge. Therefore, we randomly select m features from M superpoint features to serve as unified queries $\mathbf{q}'_u \in \mathbb{R}^{m \times d_{in}}$ for both the generic and open-vocabulary segmentation tasks. During training, we set $m < M$ to reduce computational costs, while for inference, we set $m = M$ to enable the segmentation of every region.

The prompt information is encoded into prompt features as discussed in Sec. 3.1. We employ the prompt features as prompt queries, which can be written as $\mathbf{q}'_v = \{\mathbf{f}_{v,i}\}_{i=1}^{K_v}$, $\mathbf{q}'_t = \{\mathbf{f}_{t,i}\}_{i=1}^{K_t}$, where $\mathbf{q}'_v \in \mathbb{R}^{K_v \times d_{in}}$, $\mathbf{q}'_t \in \mathbb{R}^{K_t \times d_{in}}$. K_v and K_t are the number of the point and text prompts, respectively. $\mathbf{q}'_u$, $\mathbf{q}'_v$, $\mathbf{q}'_t$ are three types of queries containing information from various aspects. Feeding them forward indiscriminately would confuse the mask decoder for digging task-specific information. Thus, we add task-specific embeddings $\mathbf{e}_u$, $\mathbf{e}_v$, and $\mathbf{e}_t$ before further processing:

$$\mathbf{q}_u = \mathbf{q}'_u + \mathbf{e}_u, \quad \mathbf{q}_v = \mathbf{q}'_v + \mathbf{e}_v, \quad \mathbf{q}_t = \mathbf{q}'_t + \mathbf{e}_t, \tag{1}$$


The armchair is to the right of the other chair. It is facing the table.
The armchair is facing the table. There is another chair to the right of it.
Ranking
>
Text prompts
Text
Text
Text
Text
$\mathbf{e}_1^t$
$\mathbf{e}_2^t$
$\mathbf{e}_3^t$
$\mathbf{e}_4^t$
Vision prompts
$\mathbf{e}_1^v$
$\mathbf{e}_2^v$
$\mathbf{e}_3^v$
$\mathbf{e}_4^v$
$\mathbf{s}_{11}$ $\mathbf{s}_{12}$ $\mathbf{s}_{13}$ $\mathbf{s}_{14}$
$\mathbf{s}_{21}$ $\mathbf{s}_{22}$ $\mathbf{s}_{23}$ $\mathbf{s}_{24}$
$\mathbf{s}_{31}$ $\mathbf{s}_{32}$ $\mathbf{s}_{33}$ $\mathbf{s}_{34}$
$\mathbf{s}_{41}$ $\mathbf{s}_{42}$ $\mathbf{s}_{43}$ $\mathbf{s}_{44}$
Matched
Distillation
Matching
Distillation
Contrastive
Output head

(a) A challenging case

(b) Ranking-based contrastive learning

(c) Explicit inter-task associations

Figure 3: Illustration of the inter-task association. (a) A challenging case requiring distinction of position information within textual descriptions. (b) A contrastive learning matrix for paired vision-text features, where a ranking rule is employed to suppress incorrect pairings. (c) Knowledge distillation across multiple tasks.

where $\mathbf{e}_u \in \mathbb{R}^{d_{in}}$, $\mathbf{e}_v \in \mathbb{R}^{d_{in}}$, $\mathbf{e}_t \in \mathbb{R}^{d_{in}}$, and are broadcasted into $\mathbb{R}^{m \times d_{in}}$, $\mathbb{R}^{K_v \times d_{in}}$, and $\mathbb{R}^{K_t \times d_{in}}$, respectively. The mask decoder comprises L mask decoder layers, which contain self-attention layers integrating information among queries. Prompt priors involving human knowledge are unavailable for generic segmentation during inference. Therefore, in the training phase, we should prevent human knowledge from leaking to the generic segmentation. In practice, the prompt queries are exclusively fed into the cross-attention layers. Output queries of the last mask decoder layer are sent into an output head consisting of MLP layers to project dimensions of the output queries from d_{in} to d_{out}. In general, the mask generation process can be formally defined as:

$$\mathbf{F}_{out} = \mathrm{MLP}\left(\mathrm{MaskDecoder}\left(\mathbf{q} = \mathrm{Concat}\left(\mathbf{q}_u, \mathbf{q}_v, \mathbf{q}_t\right); \mathbf{k} = \mathbf{F}_s; \mathbf{v} = \mathbf{F}_s\right)\right), \tag{2}$$

where $\mathbf{F}_{out} = \{\mathbf{f}_{out,i}\}_{i=1}^{m+K_v+K_t}$ represents output features, with $\mathbf{f}_{out,i} \in \mathbb{R}^{d_{out}}$ and $\mathbf{F}_{out} \in \mathbb{R}^{(m+K_v+K_t) \times d_{out}}$.

Subsequently, we can process the output features to obtain class and mask predictions. For class predictions, a common practice involves replacing class names with class IDs [23]. However, for our method to support referring segmentation, the class names are crucial information that should not be overlooked. Hence, we encode the class names into text features $\mathbf{e}_{cls} \in \mathbb{R}^{K_c \times d_{out}}$ using a frozen CLIP text encoder and propose to regress the class name features instead, where K_c denotes the number of categories. Specifically, we formulate the mask predictions $\mathbf{mask}_{pred}$ and class predictions $\mathbf{cls}_{pred}$ as follows:

$$\mathbf{mask}_{pred} = \mathbf{F}_{out} \cdot \mathrm{MLP}\left(\mathbf{F}_s\right)^{\top}, \quad \mathbf{cls}_{pred} = \mathrm{Softmax}\left(\mathbf{F}_{out} \cdot \mathbf{e}_{cls}^{\top}\right), \tag{3}$$

where $\mathbf{mask}_{pred} = \{\mathbf{mask}_i\}_{i=1}^{m+K_v+K_t}$ and $\mathbf{cls}_{pred} = \{\mathbf{cls}_i\}_{i=1}^{m+K_v+K_t}$, with $\mathbf{mask}_{pred} \in \mathbb{R}^{(m+K_v+K_t) \times M}$ and $\mathbf{cls}_{pred} \in \mathbb{R}^{(m+K_v+K_t) \times K_c}$. $\mathbf{mask}_i \in \mathbb{R}^M$ and $\mathbf{cls}_i \in \mathbb{R}^{K_c}$ represent the mask outcome and category probability predicted by the i-th query, respectively. The MLP projects $\mathbb{R}^{d_{in}}$ into $\mathbb{R}^{d_{out}}$. Given that $\mathbf{mask}_{pred}$ and $\mathbf{cls}_{pred}$ are derived from superpoints, we map the segmentation outputs for each superpoint back to the input point cloud to generate point-wise mask and class predictions.

3.3 Explicit Inter-task Association

Previous studies have overlooked connections among 3D scene understanding tasks, resulting in task-specific approaches that fail to leverage cross-task knowledge. This limitation restricts the understanding of 3D scenes to a task-specific perspective, hindering comprehensive 3D scene understanding. We establish explicit inter-task associations to overcome these constraints.

Specifically, on the one hand, as shown in Fig. 3(a), the referring segmentation is challenging when multiple individuals of identical shapes are arranged adjacently. It requires the method to distinguish the location variations inserted in the text prompts, such as “right of the other chair” vs. “another chair to the right of it.” However, the modality gap between 3D points and linguistic texts sets significant obstructions. We propose ranking-based contrastive learning between the vision and text features to reduce the modality gap and optimize the referring segmentation.

Table 1: Mask prediction performance of instance and interactive segmentation.

Tasks	mIoU
Instance Seg.	68.1
Interactive Seg.	**76.0** (+7.9)

On the other hand, as shown in Tab. 1, we evaluate our baseline framework built in Sec. 3.1 and Sec. 3.2 on instance and interactive segmentation tasks. Essentially, the main difference between the instance and interactive segmentation is w/o or w/ vision prompts. The mIoU metric, which directly measures the quality of mask predictions, indicates that the interactive segmentation surpasses the instance segmentation by a notable margin of 7.9%. It suggests that vision prompts provide reliable position priors, boosting the interactive segmentation to perform superior mask prediction performance. We design a knowledge distillation to share insights from interactive segmentation across unified tasks. The core idea of this approach is to leverage the task of predicting best-quality masks to guide the other tasks, *i.e.*, using a teacher to guide students.

3.3.1 Ranking-based Contrastive Learning

We set the vision and text prompts specifying the same individual instances into pairs and align their pairwise features by employing contrastive learning.

Assuming B vision-text pairs within a training mini-batch, the corresponding vision and text output features are $\left\{\mathbf{f}_{out,i}^{v}\right\}_{i=1}^{B}$ and $\left\{\mathbf{f}_{out,i}^{t}\right\}_{i=1}^{B}$. $\mathbf{f}_{out,i}^{v} \in \mathbb{R}^{d_{out}}$ and $\mathbf{f}_{out,i}^{t} \in \mathbb{R}^{d_{out}}$ are selected from output features $\{\mathbf{f}_{out,i}\}_{i=m+1}^{m+K_v}$ and $\{\mathbf{f}_{out,i}\}_{i=m+K_v+1}^{m+K_v+K_t}$, respectively. We normalize these selected features to obtain vision and text metric embeddings $\{\mathbf{e}_i^v\}_{i=1}^{B}$ and $\{\mathbf{e}_i^t\}_{i=1}^{B}$, where $\mathbf{e}_i^v \in \mathbb{R}^{d_{out}}$ and $\mathbf{e}_i^t \in \mathbb{R}^{d_{out}}$. Then, the contrastive learning can be formulated as $\mathcal{L}_{con} = \mathcal{L}_v + \mathcal{L}_t$, with:

$$\mathcal{L}_v = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp\left(\mathbf{e}_i^v \cdot \mathbf{e}_i^t/\tau\right)}{\sum_{j=1}^{B}\exp\left(\mathbf{e}_i^v \cdot \mathbf{e}_j^t/\tau\right)}, \quad \mathcal{L}_t = -\frac{1}{B}\sum_{i=1}^{B}\log\frac{\exp\left(\mathbf{e}_i^t \cdot \mathbf{e}_i^v/\tau\right)}{\sum_{j=1}^{B}\exp\left(\mathbf{e}_i^t \cdot \mathbf{e}_j^v/\tau\right)}, \tag{4}$$

where τ is a learnable temperature parameter. The pairwise similarity is illustrated in Fig. 3(b), where we denote $\mathbf{e}_i^v \cdot \mathbf{e}_j^t$ as $\mathbf{s}_{i,j}$ for simplification. To distinguish the target instances from adjacent ones with identical shapes, we introduce a ranking rule inspired by the CrowdCLIP [31] that the diagonal elements are greater than the off-diagonal elements, which can be described as:

$$\mathcal{L}_{rank} = \frac{1}{B}\sum_{i=1}^{B}\sum_{j=1}^{B}\max\left(0, \mathbf{s}_{\mathrm{i,j}} - \mathbf{s}_{\mathrm{i,i}}\right). \tag{5}$$

3.3.2 Knowledge Distillation

As shown in Fig. 3(c), we transfer knowledge from the interactive segmentation task to the generic and referring segmentation tasks to guide their training phases.

To generic segmentation. Define predictions decoded from the unified queries as $Pred_u = \{\mathbf{mask}_i, \mathbf{cls}_i\}_{i=1}^{m}$. We employ the Hungarian algorithm, utilizing the Dice and cross-entropy metrics as matching cost criteria, to assign $Pred_u$ with interactive segmentation labels $GT_v = \{\mathbf{mask}_{gt,i}, \mathbf{cls}_{gt,i}\}_{i=1}^{K_v}$. The matched predictions are selected as positive samples $Pos_u = \{\mathbf{mask}_{pos,i}, \mathbf{cls}_{pos,i}\}_{i=1}^{K_v}$. We denote mask predictions among the positive samples as $\mathbf{mask}_{pos} = \{\mathbf{mask}_{pos,i}\}_{i=1}^{K_v}$, with $\mathbf{mask}_{pos} \in \mathbb{R}^{K_v \times M}$. The predicted masks of interactive segmentation can be formulated as $\mathbf{mask}_v = \{\mathbf{mask}_i\}_{i=m+1}^{m+K_v}$, where $\mathbf{mask}_v \in \mathbb{R}^{K_v \times M}$. We select the pixels with top $k\%$ scores of $\mathbf{mask}_v$ as learning region $\mathbf{R}$, and depict the knowledge transfer process from the interactive segmentation to the generic segmentation tasks as:

$$\mathcal{L}_{v \to g} = \mathcal{L}_{BCE}\left(\mathbf{mask}_{pos}\left(\mathbf{R}\right), \mathbf{mask}_v\left(\mathbf{R}\right)\right), \tag{6}$$

where $\mathbf{mask}_{pos}\left(\mathbf{R}\right)$ and $\mathbf{mask}_v\left(\mathbf{R}\right)$ represent the predicted mask values within the region $\mathbf{R}$, gathering from the positive samples and the interactive segmentation predictions, respectively.

To referring segmentation. Define pairwise class probabilities predicted by the vision and text prompt queries as $\mathbf{cls}_v \in \mathbb{R}^{B \times K_c}$ and $\mathbf{cls}_t \in \mathbb{R}^{B \times K_c}$ selected from $\{\mathbf{cls}_i\}_{i=m+1}^{m+K_v}$ and $\{\mathbf{cls}_i\}_{i=m+K_v+1}^{m+K_v+K_t}$, respectively. We formulate a knowledge transfer process from the interactive segmentation to the referring segmentation task as:

$$\mathcal{L}_{v \to t} = \mathcal{L}_{BCE}\left(\mathrm{Sigmoid}\left(\mathbf{cls}_t\right), \mathrm{Sigmoid}\left(\mathbf{cls}_v\right)\right). \tag{7}$$

3.4 Training Objectives

Open-set pseudo mask labels. For open-vocabulary tasks, we train models on close-set data. To enhance segmentation performance on open-set data, we use SAM3D [67] to generate segmentation masks with undetermined categories as pseudo mask labels (open-set masks). While training, we assign predictions of the unified queries with ground-truth masks (close-set masks). The assigned and miss-assigned predictions are divided into positive and negative samples. The positive samples are supervised to regress the close-set masks. We match the negative samples with the pseudo mask labels and supervise the matched ones to regress the open-set masks. Note that the SAM3D is an unsupervised method and does not rely on ground-truth annotations, eliminating worries of label leakage. This process is exclusively applied in the training phase, incurring no extra inference cost.

Loss function. Training losses contain two components: (1) Basic losses, formulated as $\mathcal{L}_{base} = \mathcal{L}_{mask} + \mathcal{L}_{cls}$. $\mathcal{L}_{mask}$ stands for pixel-wise mask loss, comprising the BCE and Dice losses. $\mathcal{L}_{cls}$ indicates the classification loss, where we use the cross-entropy loss. (2) Losses used to build inter-task associations, summarized as $\mathcal{L}_{inter} = \mathcal{L}_{v\to g} + \mathcal{L}_{v\to t} + \mathcal{L}_{con} + \mathcal{L}_{rank}$. The final loss function is $\mathcal{L} = \mathcal{L}_{base} + \lambda\mathcal{L}_{inter}$, where λ is a balance weight set as 0.1.

4 Experiments

Datasets. We evaluate the UniSeg3D on three benchmarks: ScanNet20 [6], ScanNet200 [47], and ScanRefer [1]. ScanNet20 provides RGB-D images and 3D point clouds of $1,613$ scenes, including 18 instance categories and 2 semantic categories. ScanNet200 uses the same source data as ScanNet20, while it is more challenging for up to 198 instance categories and 2 semantic categories. ScanRefer contains $51,583$ natural language expressions referring to $11,046$ objects selected from 800 scenes.

Experimental setups. We train our method on the ScanNet20 training split, and referring texts are collected from the ScanRefer. d_{in} and d_{out} are set as 32 and 256, respectively. m ranges $[50, 100]$ percent of M with an upper limit of $3,500$. We set k as 10 and L as 6. For data augmentations, input point clouds are randomly rotated around the z-axis, elastic distorted, and scaled; the referring texts are augmented using public GPT tools following [63, 7]. We adopt the AdamW optimizer with the polynomial schedule, setting an initial learning rate as 0.0001 and the weight decay as 0.05. All models are trained for 512 epochs on a single NVIDIA RTX 4090 GPU and evaluated per 16 epochs on the validation set to find the best-performed model. To stimulate models' performance, we propose a two-stage fine-tuning trick, which fine-tunes the best-performed model, setting the learning rate and weight decay 0.001 times the initial values for 40 epochs. The proposed framework achieves end-to-end generic, interactive, and referring segmentation tasks. We divide the open-vocabulary segmentation task into mask prediction and class prediction. Specifically, we employ the proposed UniSeg3D to predict masks and then follow the Open3DIS [40] to generate class predictions.

We use PQ, mIoU, and mAP metrics to evaluate performance on the generic segmentation tasks following [38, 42, 69]. Then, we use AP and mIoU metrics for the interactive and referring segmentation tasks, respectively, following [70, 45]. For the open-vocabulary segmentation task, we train our model on the ScanNet20 and evaluate it on the ScanNet200 using AP metric, following [54]. The Overall metric represents the average performance across six tasks intended to reflect the model's unified capability.

4.1 Comparison to SOTA Methods

The proposed method achieves six 3D scene understanding tasks in a single model. We demonstrate the effectiveness of our method by comparing it with SOTA approaches specialized for specific tasks. As shown in Tab. 2, the proposed method outperforms the specialized SOTA methods Panoptic-NDT [49], OctFormer [56], MAFT [26], AGILE3D [70], X-RefSeg3D [45], and Open3DIS [40] on the panoptic, semantic, instance, interactive, referring, and open-vocabulary (OV) segmentation tasks by 12.1 PQ, 1.2 mIoU, 0.9 mAP, 1.0 AP, 4.1 mIoU, 0.7 AP, respectively. Even when compared with the competitive 3D unified method, *i.e.*, OneFormer3D [23], the proposed UniSeg3D achieves 0.1 PQ improvement on the panoptic segmentation task, and 0.3 mIoU improvement on the semantic segmentation task. More importantly, the OneFormer3D focuses on three generic segmentation tasks. It fails to understand user prompts, which limits its application prospects. In contrast, UniSeg3D unifies six tasks and presents desirable performance, demonstrating UniSeg3D a powerful architecture.

https://doi.org/10.52202/079017-1899

Table 2: Comparisons on ScanNet20 [6], ScanRefer [1], and ScanNet200 [47]. The best results are highlighted in **bold**, and the second-best results are underscored. "∗" indicates the use of the two-stage fine-tuning trick. "-/-" denotes training on filtered or complete ScanRefer datasets.

Datasets		ScanNet20				ScanRefer	ScanNet200
3D scene understanding tasks		Pan.	Sem.	Inst.	Inter.	Ref.	OV
Method	Reference	PQ	mIoU	mAP	AP	mIoU	AP
SceneGraphFusion [60]	CVPR 21	31.5	-	-	-	-	-
TUPPer-Map [68]	IROS 21	50.2	-	-	-	-	-
Panoptic Lifting [50]	CVPR 23	58.9	-	-	-	-	-
PanopticNDT [49]	IROS 23	59.2	-	-	-	-	-
PointNeXt-XL [44]	NeurIPS 22	-	71.5	-	-	-	-
PointMetaBase-XXL [34]	CVPR 23	-	72.8	-	-	-	-
MM-3DScene [66]	CVPR 23	-	72.8	-	-	-	-
PointTransformerV2 [62]	NeurIPS 22	-	75.4	-	-	-	-
ADS [10]	ICCV 23	-	75.6	-	-	-	-
OctFormer [56]	SIGGRAPH 23	-	75.7	-	-	-	-
SoftGroup [55]	CVPR 22	-	-	45.8	-	-	-
PBNet [74]	ICCV 23	-	-	54.3	-	-	-
ISBNet [39]	CVPR 23	-	-	54.5	-	-	-
SPFormer [52]	AAAI 23	-	-	56.3	-	-	-
Mask3D [48]	ICRA 23	-	-	55.2	-	-	-
MAFT [26]	ICCV 23	-	-	58.4	-	-	-
QueryFormer [36]	ICCV 23	-	-	56.5	-	-	-
OneFormer3D [23]	CVPR 24	71.2	76.6	**59.3**	-	-	-
InterObject3D [24]	ICRA 23	-	-	-	20.9	-	-
AGILE3D [70]	ICLR 24	-	-	-	53.5	-	-
TGNN [15]	AAAI 21	-	-	-	-	24.9/27.8	-
X-RefSeg3D [45]	AAAI 24	-	-	-	-	25.5/29.9	-
OpenScene [41] with [48]	CVPR 23	-	-	-	-	-	8.5
OpenMask3D [54]	NeurIPS 23	-	-	-	-	-	12.6
SOLE [28]	CVPR 24	-	-	-	-	-	18.7
Open3DIS [40]	CVPR 24	-	-	-	-	-	19.0
UniSeg3D (**ours**)	-	**71.3**	76.3	59.1	54.1	29.5/-	19.6
UniSeg3D* (**ours**)	-	**71.3**	**76.9**	**59.3**	**54.5**	**29.6/-**	**19.7**

The proposed method achieves six tasks in one training, which is elegant while facing an issue for fair comparison. Specifically, partial labels in the referring segmentation benchmark (10, 115 objects, 27.6% of the complete ScanRefer training set) annotate novel classes of the open-vocabulary segmentation task. Obviously, these labels should not be used for training to avoid label leakage. Thus, we filter out these labels and only employ the filtered ScanRefer training set to train our model. As shown in Tab. 2, our model uses 72.4% training data to achieve closing performance with X-RefSeg3D [45] (29.6 vs. 29.9). Moreover, while reproducing the X-RefSeg3D using official code on our filtered training data, the performance drops to 4.1 mIoU lower than UniSeg3D, demonstrating our model's effectiveness.

Table 3: Ablation on task unification.

ScanNet200	ScanRefer	ScanNet20			
OV	Ref.	Inter.	Pan.	Sem.	Inst.
AP	mIoU	AP	PQ	mIoU	mAP
✗	✗	✗	**71.0**	76.2	**59.0**
✗	✗	56.8	**71.0**	**76.4**	58.7
✗	29.1	56.0	70.3	76.3	58.4
19.7	29.1	54.5	70.4	76.2	58.0

4.2 Analysis and Ablation

We conduct ablation studies and analyze key insights of our designs. All models are evaluated on multiple tasks to show the effectiveness of the proposed components on a broad scope.

Table 4: Ablation on components. “Distillation”, “Rank-Contrastive”, and “Trick” denote the knowledge distillation, ranking-based contrastive learning, and two-stage fine-tuning trick, respectively.

Datasets			ScanNet20				ScanRefer	ScanNet200	Overall
Components			Pan.	Sem.	Inst.	Inter.	Ref.	OV	
Distillation	Rank-Contrastive	Trick	PQ	mIoU	mAP	AP	mIoU	AP	
-	-	-	70.4	76.2	58.0	54.5	29.1	19.7	51.3
✔	-	-	70.9	76.2	58.6	**55.3**	29.2	19.6	51.6
-	✔	-	70.8	76.4	58.4	54.1	**29.6**	**19.9**	51.5
✔	✔	-	**71.3**	76.3	59.1	54.1	29.5	19.6	51.7
✔	✔	✔	**71.3**	**76.9**	**59.3**	54.5	**29.6**	19.7	**51.9**

Table 5: Ablation on different designs of the proposed components. “$v \to g$” and “$v \to t$” denote the knowledge distillation from the interactive segmentation to the generic segmentation and the referring segmentation, respectively. “Contrastive” and “Rank” denote the contrastive learning and the ranking rule, respectively.

(a) Ablation on designs for knowledge distillation.

Datasets		ScanNet20				ScanRefer	ScanNet200	Overall
Components		Pan.	Sem.	Inst.	Inter.	Ref.	OV	
$v \to g$	$v \to t$	PQ	mIoU	mAP	AP	mIoU	AP	
-	-	70.8	**76.4**	58.4	**54.1**	29.6	19.9	51.5
✔	-	71.2	76.3	59.0	54.0	29.5	19.8	51.6
-	✔	70.7	76.2	58.6	**54.1**	**29.7**	**20.0**	51.6
✔	✔	**71.3**	76.3	**59.1**	**54.1**	29.5	19.6	**51.7**

(b) Ablation on designs for ranking-based contrastive learning.

Datasets		ScanNet20				ScanRefer	ScanNet200	Overall
Components		Pan.	Sem.	Inst.	Inter.	Ref.	OV	
Contrastive	Rank	PQ	mIoU	mAP	AP	mIoU	AP	
-	-	70.9	76.2	58.6	**55.3**	29.2	19.6	51.6
✔	-	71.0	**76.3**	59.0	54.5	29.4	19.7	**51.7**
-	✔	71.0	76.2	58.7	54.6	**29.5**	**19.8**	51.6
✔	✔	**71.3**	**76.3**	**59.1**	54.1	**29.5**	19.6	**51.7**

The challenge of multi-task unification. We discuss the challenge of unifying multiple tasks in a single model. Specifically, we simply add interactive, referring, and open-vocabulary segmentation into our framework to build a unification baseline, as shown in Tab. 3. We observe a continuous performance decline on the panoptic, instance, and interactive segmentation tasks, indicating a significant obstacle in balancing different tasks. Even so, we believe that unifying multiple tasks within a single model is worth exploring, as it can reduce computation consumption and benefit real-world applications. Thus, this paper proposes to eliminate performance decline by delivering inter-task associations, and the following experiments demonstrate that this could be a valuable step.

Design of inter-task associations. Our approach uses knowledge distillation and contrastive learning to connect supported tasks. As shown in Tab. 4, when applying the knowledge distillation, *i.e.* row 2, the performance of instance and interactive segmentation increase to 58.6 mAP and 55.3 AP, respectively. We believe the improvement on the instance task is because of the reliable knowledge distilled from the interactive segmentation, and the improvement on the interactive segmentation task is attributed to the intrinsic connections between the two tasks. Then, we ablate the ranking-based contrastive learning, *i.e.* row 3. We observe improvements on five tasks, including the generic segmentation and the referring segmentation, while a slight performance drop on the interactive segmentation. This phenomenon suggests that contrastive learning is effective on most tasks, but there is a huge struggle to align point and text modalities, which weakens the interactive segmentation performance. Overall metric measures multi-task unification performance. We choose models and

Table 6: Ablation on hyper-parameter λ.

Datasets	ScanNet20				ScanRefer	ScanNet200	Overall
Hyper-parameter	Pan.	Sem.	Inst.	Inter.	Ref.	OV	
λ	PQ	mIoU	mAP	AP	mIoU	AP	
0.05	70.7	76.2	58.9	**54.4**	29.5	**19.6**	51.6
0.1	**71.3**	76.3	**59.1**	54.1	29.5	**19.6**	**51.7**
0.2	70.8	**76.6**	58.6	52.3	**29.8**	19.5	51.3
0.3	70.6	75.7	58.4	51.6	29.6	19.3	50.9

checkpoints with higher Overalls in our experiments. In practical applications, checkpoints can be chosen based on preferred tasks while maintaining good performance across other tasks. Applying knowledge distillation and ranking-based contrastive learning obtains comparable performance on most tasks, performing higher Overall than rows 2 and 3, indicating complementarity of the two components. We employ the two-stage fine-tuning trick, consistently improving various tasks.

Detailed ablation on the components is shown in Tab. 5. It is observed that knowledge distillation to various tasks brings respective improvements. As for contrastive learning, comparing row 2 and row 4 in Tab. 5(b), the ranking rule suppresses confusing point-text pairs, boosting contrastive learning to be more effective. λ controls the strength of the explicit inter-task associations. We empirically find that setting λ to 0.1 obtains the best performance, as shown in Tab. 6.

Table 7: Ablation on vision prompts.

Strategy	mIoU	AP	AP_{50}	AP_{25}
From [70]	78.8	54.5	79.4	93.2
Instance center	**79.6**	**56.6**	**82.1**	**94.9**
$r_d = 0.1$	79.1	55.9	81.1	94.4
$r_d = 0.2$	78.7	55.1	80.0	93.4
$r_d = 0.3$	78.0	53.8	78.5	92.4
$r_d = 0.4$	77.5	53.0	77.4	91.7
$r_d = 0.5$	76.6	52.1	76.2	90.6
$r_d = 0.6$	75.9	51.2	74.6	90.0
$r_d = 0.7$	74.9	50.1	72.9	88.1
$r_d = 0.8$	73.4	48.2	71.1	86.5
$r_d = 0.9$	71.0	45.3	66.6	82.1
$r_d = 1.0$	62.7	36.4	54.8	70.2
Random	76.0	51.3	75.2	89.6

Influence of vision prompts. We empirically find that vision prompts affect the interactive segmentation performance. To ensure a fair comparison, we adopt the same vision prompts generation strategy designed in AGILE3D [70] to evaluate our interactive segmentation performance.

We ablate 3D spatial distances between the vision prompts and instance centers. Specifically, assuming an instance containing n points, we denote the mean coordinate of these points as the *instance center* and order the n points based on their distances to the instance center. Then, we evaluate the interactive segmentation performance while employing the $\lfloor r_d \times n \rfloor$-th nearest point as the vision prompt, as shown in Tab. 7. When the vision prompt is positioned at the instance center, the interactive segmentation achieves an upper-bound performance of 56.6 AP, exhibiting a substantial performance gap of up to 20.2 AP compared to when the vision prompt is located at the object's edge ($r_d = 1.0$), illustrating considerable room for improvement. We also observe an unusual performance decline while increasing r_d from 0.9 to 1.0, which we attribute to the ambiguity in distinguishing the edge points of adjacent instances. As we know, this is the first work ablating the influence of vision prompts, and we will explore this issue in depth in future work.

5 Conclusion and Discussion

We propose a unified framework named UniSeg3D, which provides a flexible and efficient solution for 3D scene understanding, supporting six tasks within a single model. Previous task-specific approaches fail to leverage cross-task information, limiting their understanding of 3D scenes to task-specific perspectives. In contrast, we take advantage of the multi-task design and enhance performance by building inter-task associations. Specifically, we employ knowledge distillation and ranking-based contrastive learning to facilitate cross-task knowledge sharing. Experiments demonstrate that the proposed framework is a powerful method, achieving SOTA performance across six unified tasks.

Limitation. UniSeg3D aims to achieve unified 3D scene understanding. However, it works on indoor tasks and lacks explorations in outdoor scenes. Additionally, we observe that UniSeg3D performs worse interactive segmentation performance when the vision prompt is located away from the instance centers, limiting the reliability of the UniSeg3D and should be explored in future work.

Acknowledgements

This work is supported in part by the National Natural Science Foundation of China (Grant. No. 62225603 and 623B2038).

References

[1] Dave Zhenyu Chen, Angel X Chang, and Matthias Nießner. Scanrefer: 3d object localization in rgb-d scans using natural language. In *Proc. of European Conference on Computer Vision*, pages 202–221, 2020.

[2] Shaoyu Chen, Jiemin Fang, Qian Zhang, Wenyu Liu, and Xinggang Wang. Hierarchical aggregation for 3d instance segmentation. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 15467–15476, 2021.

[3] Shizhe Chen, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Sugar: Pre-training 3d visual representations for robotics. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 18049–18060, 2024.

[4] Silin Cheng, Xiwu Chen, Xinwei He, Zhe Liu, and Xiang Bai. Pra-net: Point relation-aware network for 3d point cloud analysis. *IEEE Transactions on Image Processing*, 30:4436–4448, 2021.

[5] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 3075–3084, 2019.

[6] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. ScanNet: Richly-annotated 3D reconstructions of indoor scenes. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017.

[7] Haixing Dai, Zhengliang Liu, Wenxiong Liao, Xiaoke Huang, Yihan Cao, Zihao Wu, Lin Zhao, Shaochen Xu, Wei Liu, Ninghao Liu, et al. Auggpt: Leveraging chatgpt for text data augmentation. *arXiv preprint arXiv:2302.13007*, 2023.

[8] Lei Han, Tian Zheng, Lan Xu, and Lu Fang. Occuseg: Occupancy-aware 3d instance segmentation. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 2940–2949, 2020.

[9] Tong He, Chunhua Shen, and Anton Van Den Hengel. Dyco3d: Robust instance segmentation of 3d point clouds through dynamic convolution. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 354–363, 2021.

[10] Cheng-Yao Hong, Yu-Ying Chou, and Tyng-Luh Liu. Attention discriminant sampling for point clouds. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 14429–14440, 2023.

[11] Fangzhou Hong, Lingdong Kong, Hui Zhou, Xinge Zhu, Hongsheng Li, and Ziwei Liu. Unified 3d and 4d panoptic segmentation via dynamic shifting networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[12] Ji Hou, Angela Dai, and Matthias Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 4421–4430, 2019.

[13] Jinghua Hou, Zhe Liu, Dingkang Liang, Zhikang Zou, Xiaoqing Ye, and Xiang Bai. Query-based temporal fusion with explicit motion for 3d object detection. In *Proc. of Advances in Neural Information Processing Systems*, volume 36, 2023.

[14] Di Huang, Sida Peng, Tong He, Honghui Yang, Xiaowei Zhou, and Wanli Ouyang. Ponder: Point cloud pre-training via neural rendering. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 16089–16098, 2023.

[15] Pin-Hao Huang, Han-Hung Lee, Hwann-Tzong Chen, and Tyng-Luh Liu. Text-guided graph neural networks for referring 3d instance segmentation. In *Proc. of the AAAI Conf. on Artificial Intelligence*, pages 1610–1618, 2021.

[16] Zhening Huang, Xiaoyang Wu, Xi Chen, Hengshuang Zhao, Lei Zhu, and Joan Lasenby. Openins3d: Snap and lookup for 3d open-vocabulary instance segmentation. In *Proc. of European Conference on Computer Vision*, 2024.

[17] Muhammad Zubair Irshad, Sergey Zakharov, Vitor Guizilini, Adrien Gaidon, Zsolt Kira, and Rares Ambrus. Nerf-mae: Masked autoencoders for self-supervised 3d representation learning for neural radiance fields. In *Proc. of European Conference on Computer Vision*, 2024.

[18] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 2989–2998, 2023.

[19] Maximilian Jaritz, Jiayuan Gu, and Hao Su. Multi-view pointnet for 3d scene understanding. In *Porc. of IEEE Intl. Conf. on Computer Vision Workshops.*, pages 0–0, 2019.

https://doi.org/10.52202/079017-1899

[20] Li Jiang, Hengshuang Zhao, Shaoshuai Shi, Shu Liu, Chi-Wing Fu, and Jiaya Jia. Pointgroup: Dual-set point grouping for 3d instance segmentation. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 4867–4876, 2020.

[21] 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 *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 4015–4026, 2023.

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

[23] Maxim Kolodiazhnyi, Anna Vorontsova, Anton Konushin, and Danila Rukhovich. Oneformer3d: One transformer for unified point cloud segmentation. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2024.

[24] Theodora Kontogianni, Ekin Celikkan, Siyu Tang, and Konrad Schindler. Interactive object segmentation in 3d point clouds. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2891–2897, 2023.

[25] Xin Lai, Jianhui Liu, Li Jiang, Liwei Wang, Hengshuang Zhao, Shu Liu, Xiaojuan Qi, and Jiaya Jia. Stratified transformer for 3d point cloud segmentation. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2022.

[26] Xin Lai, Yuhui Yuan, Ruihang Chu, Yukang Chen, Han Hu, and Jiaya Jia. Mask-attention-free transformer for 3d instance segmentation. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 3693–3703, 2023.

[27] Loic Landrieu and Martin Simonovsky. Large-scale point cloud semantic segmentation with superpoint graphs. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 4558–4567, 2018.

[28] Seungjun Lee, Yuyang Zhao, and Gim Hee Lee. Segment any 3d object with language. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2024.

[29] Xiangtai Li, Haobo Yuan, Wei Li, Henghui Ding, Size Wu, Wenwei Zhang, Yining Li, Kai Chen, and Chen Change Loy. Omg-seg: Is one model good enough for all segmentation? In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2024.

[30] Dingkang Liang, Tianrui Feng, Xin Zhou, Yumeng Zhang, Zhikang Zou, and Xiang Bai. Parameter-efficient fine-tuning in spectral domain for point cloud learning. *arXiv preprint arXiv:2410.08114*, 2024.

[31] Dingkang Liang, Jiahao Xie, Zhikang Zou, Xiaoqing Ye, Wei Xu, and Xiang Bai. Crowdclip: Unsupervised crowd counting via vision-language model. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 2893–2903, 2023.

[32] Dingkang Liang, Xin Zhou, Wei Xu, Xingkui Zhu, Zhikang Zou, Xiaoqing Ye, Xiao Tan, and Xiang Bai. Pointmamba: A simple state space model for point cloud analysis. In *Proc. of Advances in Neural Information Processing Systems*, 2024.

[33] Zhihao Liang, Zhihao Li, Songcen Xu, Mingkui Tan, and Kui Jia. Instance segmentation in 3d scenes using semantic superpoint tree networks. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 2783–2792, 2021.

[34] Haojia Lin, Xiawu Zheng, Lijiang Li, Fei Chao, Shanshan Wang, Yan Wang, Yonghong Tian, and Rongrong Ji. Meta architecture for point cloud analysis. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 17682–17691, 2023.

[35] Youquan Liu, Lingdong Kong, Xiaoyang Wu, Runnan Chen, Xin Li, Liang Pan, Ziwei Liu, and Yuexin Ma. Multi-space alignments towards universal lidar segmentation. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2024.

[36] Jiahao Lu, Jiacheng Deng, Chuxin Wang, Jianfeng He, and Tianzhu Zhang. Query refinement transformer for 3d instance segmentation. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 18516–18526, 2023.

[37] Li Minghan, Li Shuai, Zhang Xindong, and Zhang Lei. Univs: Unified and universal video segmentation with prompts as queries. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2024.

[38] Gaku Narita, Takashi Seno, Tomoya Ishikawa, and Yohsuke Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 4205–4212, 2019.

[39] Tuan Duc Ngo, Binh-Son Hua, and Khoi Nguyen. Isbnet: a 3d point cloud instance segmentation network with instance-aware sampling and box-aware dynamic convolution. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 13550–13559, 2023.

[40] Phuc DA Nguyen, Tuan Duc Ngo, Chuang Gan, Evangelos Kalogerakis, Anh Tran, Cuong Pham, and Khoi Nguyen. Open3dis: Open-vocabulary 3d instance segmentation with 2d mask guidance. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2024.

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

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

[43] Lu Qi, Lehan Yang, Weidong Guo, Yu Xu, Bo Du, Varun Jampani, and Ming-Hsuan Yang. Unigs: Unified representation for image generation and segmentation. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, 2024.

[44] 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 *Proc. of Advances in Neural Information Processing Systems*, pages 23192–23204, 2022.

[45] Zhipeng Qian, Yiwei Ma, Jiayi Ji, and Xiaoshuai Sun. X-refseg3d: Enhancing referring 3d instance segmentation via structured cross-modal graph neural networks. In *Proc. of the AAAI Conf. on Artificial Intelligence*, pages 4551–4559, 2024.

[46] 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 *Proc. of Intl. Conf. on Machine Learning*, pages 8748–8763, 2021.

[47] David Rozenberszki, Or Litany, and Angela Dai. Language-grounded indoor 3d semantic segmentation in the wild. In *Proc. of European Conference on Computer Vision*, pages 125–141, 2022.

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

[49] Daniel Seichter, Benedict Stephan, Söhnke Benedikt Fischedick, Steffen Müller, Leonard Rabes, and Horst-Michael Gross. Panopticndt: Efficient and robust panoptic mapping. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 7233–7240, 2023.

[50] Yawar Siddiqui, Lorenzo Porzi, Samuel Rota Buló, Norman Müller, Matthias Nießner, Angela Dai, and Peter Kontschieder. Panoptic lifting for 3d scene understanding with neural fields. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 9043–9052, 2023.

[51] 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.

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

[53] Weiwei Sun, Daniel Rebain, Renjie Liao, Vladimir Tankovich, Soroosh Yazdani, Kwang Moo Yi, and Andrea Tagliasacchi. Neuralbf: Neural bilateral filtering for top-down instance segmentation on point clouds. In *Proc. of IEEE Winter Conf. on Applications of Computer Vision*, pages 551–560, 2023.

[54] Ayca Takmaz, Elisabetta Fedele, Robert Sumner, Marc Pollefeys, Federico Tombari, and Francis Engelmann. Openmask3d: Open-vocabulary 3d instance segmentation. In *Proc. of Advances in Neural Information Processing Systems*, 2023.

[55] Thang Vu, Kookhoi Kim, Tung M Luu, Thanh Nguyen, and Chang D Yoo. Softgroup for 3d instance segmentation on point clouds. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 2708–2717, 2022.

[56] Peng-Shuai Wang. Octformer: Octree-based transformers for 3d point clouds. *ACM Transactions ON Graphics*, 42(4):1–11, 2023.

[57] Changli Wu, Qi Chen, Jiayi Ji, Haowei Wang, Yiwei Ma, You Huang, Gen Luo, Hao Fei, Xiaoshuai Sun, and Rongrong Ji. Rg-san: Rule-guided spatial awareness network for end-to-end 3d referring expression segmentation. In *Proc. of Advances in Neural Information Processing Systems*, 2024.

[58] Changli Wu, Yihang Liu, Jiayi Ji, Yiwei Ma, Haowei Wang, Gen Luo, Henghui Ding, Xiaoshuai Sun, and Rongrong Ji. 3d-gres: Generalized 3d referring expression segmentation. In *Proc. of ACM Multimedia*, 2024.

[59] Changli Wu, Yiwei Ma, Qi Chen, Haowei Wang, Gen Luo, Jiayi Ji, and Xiaoshuai Sun. 3d-stmn: Dependency-driven superpoint-text matching network for end-to-end 3d referring expression segmentation. In *Proc. of the AAAI Conf. on Artificial Intelligence*, volume 38, pages 5940–5948, 2024.

[60] Shun-Cheng Wu, Johanna Wald, Keisuke Tateno, Nassir Navab, and Federico Tombari. Scenegraphfusion: Incremental 3d scene graph prediction from rgb-d sequences. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 7515–7525, 2021.

https://doi.org/10.52202/079017-1899

[61] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 9621–9630, 2019.

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

[63] Yanmin Wu, Qiankun Gao, Renrui Zhang, and Jian Zhang. Language-assisted 3d scene understanding. In *Proc. of the AAAI Conf. on Artificial Intelligence*, 2024.

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

[65] Chenfeng Xu, Bichen Wu, Zining Wang, Wei Zhan, Peter Vajda, Kurt Keutzer, and Masayoshi Tomizuka. Squeezesegv3: Spatially-adaptive convolution for efficient point-cloud segmentation. In *Proc. of European Conference on Computer Vision*, pages 1–19, 2020.

[66] Mingye Xu, Mutian Xu, Tong He, Wanli Ouyang, Yali Wang, Xiaoguang Han, and Yu Qiao. Mm-3dscene: 3d scene understanding by customizing masked modeling with informative-preserved reconstruction and self-distilled consistency. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 4380–4390, 2023.

[67] Yunhan Yang, Xiaoyang Wu, Tong He, Hengshuang Zhao, and Xihui Liu. Sam3d: Segment anything in 3d scenes. In *Porc. of IEEE Intl. Conf. on Computer Vision Workshops.*, 2023.

[68] Zhiliu Yang and Chen Liu. Tupper-map: Temporal and unified panoptic perception for 3d metric-semantic mapping. In *Proc. of the IEEE Int. Conf. on Intelligent Robots and Systems*, pages 1094–1101, 2021.

[69] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J Guibas. Gspn: Generative shape proposal network for 3d instance segmentation in point cloud. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 3947–3956, 2019.

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

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

[72] Dingyuan Zhang, Dingkang Liang, Zhikang Zou, Jingyu Li, Xiaoqing Ye, Zhe Liu, Xiao Tan, and Xiang Bai. A simple vision transformer for weakly semi-supervised 3d object detection. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 8373–8383, 2023.

[73] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 16259–16268, 2021.

[74] Weiguang Zhao, Yuyao Yan, Chaolong Yang, Jianan Ye, Xi Yang, and Kaizhu Huang. Divide and conquer: 3d point cloud instance segmentation with point-wise binarization. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 562–571, 2023.

[75] Xin Zhou, Dingkang Liang, Wei Xu, Xingkui Zhu, Yihan Xu, Zhikang Zou, and Xiang Bai. Dynamic adapter meets prompt tuning: Parameter-efficient transfer learning for point cloud analysis. In *Proc. of IEEE Intl. Conf. on Computer Vision and Pattern Recognition*, pages 14707–14717, 2024.

[76] Ziyu Zhu, Xiaojian Ma, Yixin Chen, Zhidong Deng, Siyuan Huang, and Qing Li. 3d-vista: Pre-trained transformer for 3d vision and text alignment. In *Porc. of IEEE Intl. Conf. on Computer Vision*, pages 2911–2921, 2023.

https://doi.org/10.52202/079017-1899

Appendix

In this appendix, we provide additional content to complement the main manuscript:

- Appendix A: Comparisons employing more metrics on specific tasks.
- Appendix B: Inference time analysis of the proposed UniSeg3D.
- Appendix C: Qualitative visualizations illustrating model effectiveness.

A Comparisons employing more metrics on specific tasks.

The experiments presented in the main manuscript primarily use overarching metrics to measure performance on each task. This section provides more comprehensive comparisons of our method on each task using detailed metrics. We train the model on ScanNet20 and assess its open-vocabulary segmentation performance on ScanNet200. Following [40], 51 classes in ScanNet200 that are semantically similar to annotated classes in ScanNet20 are grouped as `Base` classes, while the remaining classes are divided as `Novel` classes. The model is then directly tested on Replica [51] to evaluate its zero-shot segmentation performance.

Table I: Comparison with existing instance segmentation methods on ScanNet20. UniSeg3D achieves highly competitive performance.

Method	Reference	mAP_{25}	mAP_{50}	mAP
3D-SIS[12]	CVPR 19	35.7	18.7	-
GSPN[69]	CVPR 19	53.4	37.8	19.3
PointGroup[20]	CVPR 20	71.3	56.7	34.8
OccuSeg[8]	CVPR 20	71.9	60.7	44.2
DyCo3D[9]	CVPR 21	72.9	57.6	35.4
SSTNet[33]	ICCV 21	74.0	64.3	49.4
HAIS[2]	ICCV 21	75.6	64.4	43.5
DKNet[64]	ICCV 22	76.9	66.7	50.8
SoftGroup[55]	CVPR 22	78.9	67.6	45.8
PBNet[74]	ICCV 23	78.9	70.5	54.3
ISBNet[39]	CVPR 23	82.5	73.1	54.5
SPFormer[52]	AAAI 23	82.9	73.9	56.3
Mask3D[48]	ICRA 23	83.5	73.7	55.2
MAFT[26]	ICCV 23	-	75.9	58.4
QueryFormer[36]	ICCV 23	83.3	74.2	56.5
OneFormer3D[23]	CVPR 24	**86.4**	**78.1**	**59.3**
UniSeg3D (**ours**)	-	86.1	77.0	**59.3**

Table II: Comparison with previous 3D interactive segmentation methods on ScanNet20. UniSeg3D presents remarkable performance in terms of three metrics.

Method	Reference	AP	AP_{50}	AP_{25}
InterObject3D [24]	ICRA 23	20.9	38.0	67.2
AGILE3D [70]	ICLR 24	53.5	75.6	91.3
UniSeg3D (**ours**)	-	**54.5**	**79.4**	**93.2**

Table III: Comparison with existing 3D referring segmentation methods on ScanRefer. UniSeg3D demonstrates notable performance in terms of mIoU and acc@0.25.

Method	Reference	mIoU	acc@0.5	acc@0.25
TGNN [15]	AAAI 21	24.9	28.2	33.2
X-RefSeg3D [45]	AAAI 24	25.5	**28.6**	34.0
UniSeg3D (**ours**)	-	**29.6**	28.0	**41.5**

Table IV: Comparison with previous open-vocabulary segmentation methods on ScanNe200 and Replica. Our method outperforms existing approaches in terms of AP.

Method	Reference	ScanNet200			Replica		
		AP	$AP_{\texttt{Base}}$	$AP_{\texttt{Novel}}$	AP	AP_{50}	AP_{25}
OpenScene [41] with [48]	CVPR 23	8.5	11.1	7.6	10.9	15.6	17.3
OpenMask3D[54]	NeurIPS 23	12.6	14.3	11.9	13.1	18.4	24.2
SOLE[28]	CVPR 24	18.7	17.4	**19.1**	-	-	-
Open3DIS[40]	CVPR 24	19.0	**25.8**	16.5	18.5	**24.5**	28.2
UniSeg3D (**ours**)	-	**19.7**	24.4	18.0	**19.1**	24.1	**29.2**

B Inference time analysis of the proposed UniSeg3D.

This work proposes a unified framework, achieving six tasks in one inference, which would be more efficient than running six task-specific approaches individually. We present the inference time of the proposed method for efficiency analysis. Tab. V illustrates that our method achieves effective unification across six tasks while maintaining highly competitive inference times compared to previous methods.

Table V: Inference time and instance segmentation performance on the ScanNet20 validation split.

Method	Component	Device	Component time, ms	Total time, ms	mAP
PointGroup [20]	Backbone	GPU	48	372	34.8
	Grouping	GPU+CPU	218		
	ScoreNet	GPU	106		
HAIS [2]	Backbone	GPU	50	256	43.5
	Hierarchical aggregation	GPU+CPU	116		
	Intra-instance refinement	GPU	90		
SoftGroup [55]	Backbone	GPU	48	266	45.8
	Soft grouping	GPU+CPU	121		
	Top-down refinement	GPU	97		
SSTNet [33]	Superpoint extraction	CPU	168	400	49.4
	Backbone	GPU	26		
	Tree Network	GPU+CPU	148		
	ScoreNet	GPU	58		
Mask3D [48] w/o clustering	Backbone	GPU	106	221	54.3
	Mask module	GPU	100		
	Query refinement	GPU	15		
Mask3D [48]	Backbone	GPU	106	19851	55.2
	Mask module	GPU	100		
	Query refinement	GPU	15		
	DBSCAN clustering	CPU	19630		
SPFormer [52]	Superpoint extraction	CPU	168	215	56.3
	Backbone	GPU	26		
	Superpoint pooling	GPU	4		
	Query decoder	GPU	17		
OneFormer3D [23]	Superpoint extraction	CPU	168	221	59.3
	Backbone	GPU	26		
	Superpoint pooling	GPU	4		
	Query decoder	GPU	23		
UniSeg3D (**ours**)	Superpoint extraction	CPU	168	230.03	59.3
	Backbone	GPU	33		
	Text encoder	GPU	0.03		
	Mask decoder	GPU	29		

C Qualitative visualizations illustrating model effectiveness.

We provide qualitative results in this section. In Fig. I, visualizations of multi-task segmentation results are presented, showcasing point clouds, ground truth, and predictions within each scene. In Fig. II, we present visualizations of predictions from UniSeg3D and current SOTA methods. In Fig. III, we test our model on open-set classes not included in training data to evaluate the model's open capability. Furthermore, we even replace the class names with attribute descriptions in the open vocabulary, and impressively, we observe the preliminary reasoning capabilities of our approach.

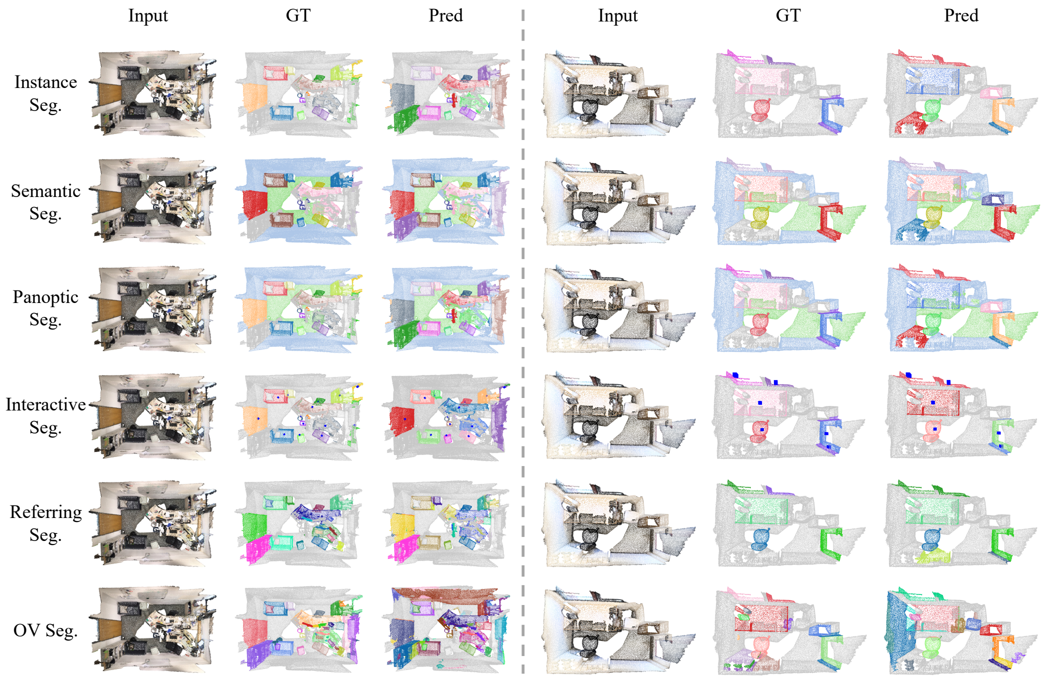


Figure I: Visualization of segmentation results obtained by UniSeg3D on ScanNet20 validation split.

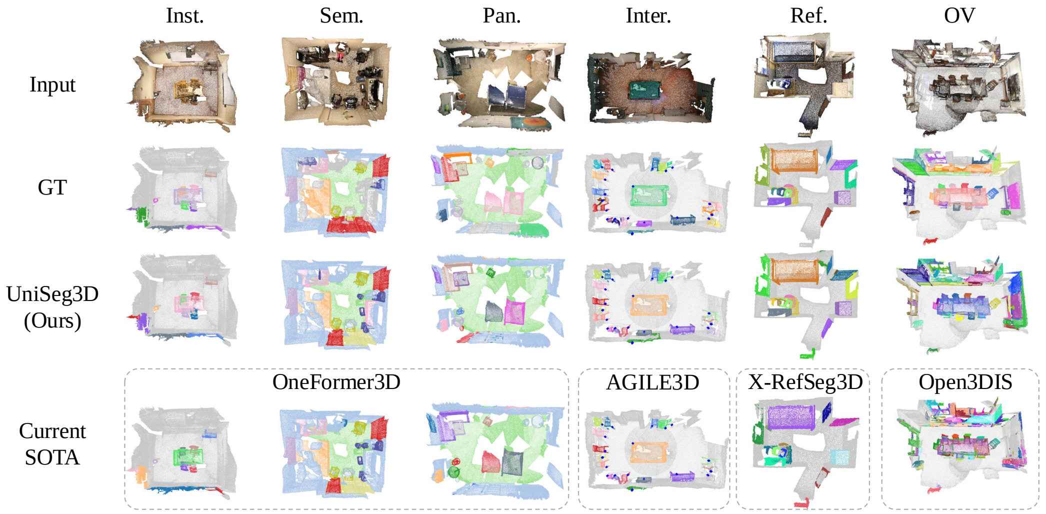


Figure II: Visualization of segmentation results obtained by UniSeg3D and current SOTA methods on ScanNet20 validation split.

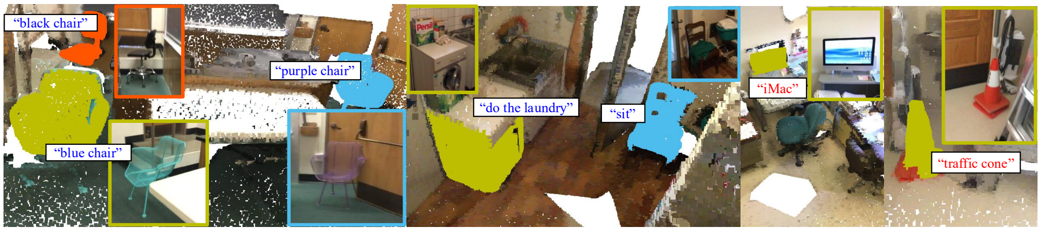


Figure III: Visualization of open capabilities. Red prompts involve categories not presented in the ScanNet200 annotations, while blue prompts describe attributes of various objects, such as affordances and color.

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: See abstract and introduction.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: See limitation part.

   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: [Yes]

https://doi.org/10.52202/079017-1899

Justification: See experiments.

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: See experiments part.

   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?

https://doi.org/10.52202/079017-1899

Answer: [NA]

Justification: The code will be made 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: See experiments part.

   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: [Yes]

   Justification: See experiments part.

   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.

https://doi.org/10.52202/079017-1899

- 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: See experiments part.

   Guidelines:

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

9. **Code Of Ethics**

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

   Answer: [Yes]

   Justification: We follow the NeurIPS Code of Ethic.

   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: [NA]

    Justification: no societal impacts.

    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

https://doi.org/10.52202/079017-1899

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: 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 will release the code.

Guidelines:

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

13. **New Assets**

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

Answer: [NA]

Justification: We use the public assets.

Guidelines:

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

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

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

Answer: [NA]

Justification: [NA]

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: [NA]

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.

https://doi.org/10.52202/079017-1899