2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

PinPoint3D: Fine-Grained 3D Part Segmentation from a Few Clicks

Bojun Zhang[1,2], Hangjian Ye[1,2], Hao Zheng[1,2,†], Jianzheng Huang[1],
Zhengyu Lin[1], Zhenhong Guo[1], Feng Zheng[1,2,*]

***Abstract*— Fine-grained 3D part segmentation is crucial for enabling embodied AI systems to perform complex manipulation tasks, such as interacting with specific functional components of an object. However, existing interactive segmentation methods are largely confined to coarse, instance-level targets, while non-interactive approaches struggle with sparse, real-world scans and suffer from a severe lack of annotated data. To address these limitations, we introduce PinPoint3D, a novel interactive framework for fine-grained, multi-granularity 3D segmentation, capable of generating precise part-level masks from only a few user point clicks. A key component of our work is a new 3D data synthesis pipeline that we developed to create a large-scale, scene-level dataset with dense part annotations, overcoming a critical bottleneck that has hindered progress in this field. Through comprehensive experiments, we demonstrate that our method significantly outperforms existing approaches, achieving an average IoU of 55.8% on each object part with only one click and surpassing 71.3% IoU with a few additional click queries. Compared to current state-of-the-art baselines, PinPoint3D yields up to a 16% improvement in IoU and precision, highlighting its effectiveness and high efficiency on challenging, sparse point clouds. Our work represents a significant step towards more nuanced and precise machine perception and interaction in complex 3D environments. Our code, checkpoints and datasets can be found at the project website https://pinpoint3d.online.**

I. INTRODUCTION

The advancement of embodied AI, from household assistants to industrial robots, hinges on perceiving and interacting with the world in fine detail. For instance, a robot opening a cabinet drawer must identify a specific functional *part*—the handle—rather than just the cabinet as a whole. This task requires a hierarchical understanding of scenes (scene → instance → part). However, training systems for such tasks are hindered by a major bottleneck: the prohibitive cost and labor of acquiring dense, part-level 3D annotations.

Interactive 3D segmentation offers a way to ease the annotation burden [14], [33], [25], [35], [37], but current methods focus on coarse, instance-level tasks and are inefficient and inaccurate for part segmentation (Fig. 1). Existing interactive part segmentation approaches are also limited, as they either process objects in isolation [15] or rely on 2D models that lose crucial 3D geometric details [37]. Furthermore, non-interactive methods that work well on clean CAD models perform poorly on sparse, noisy real-world scans [8], [5], [16]. This highlights a critical need for an interactive framework that excels at part segmentation in complex, scene-level data.

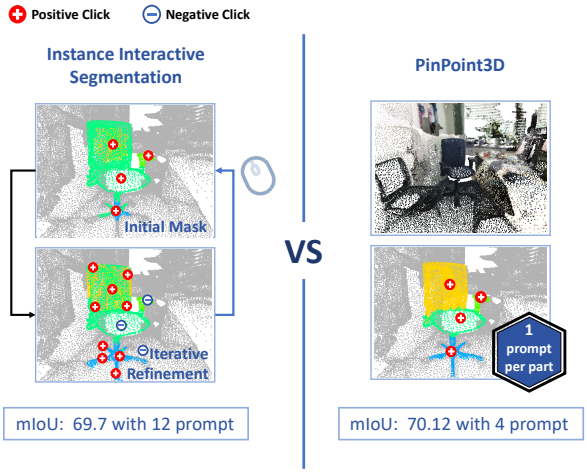


Fig. 1. **Purpose-Built Interactive Part Segmentation.** Instance-level models repurposed for part segmentation (*left*) require extensive user interaction (e.g., 12 clicks). In contrast, our purpose-built framework (*right*) delivers superior accuracy with minimal effort, requiring only a single click per part.

To address this, we introduce **PinPoint3D**, a novel interactive framework for fine-grained, multi-granularity 3D segmentation from sparse, scene-level point clouds. Our model first uses a sparse convolutional network with a lightweight adapter to extract point features. User clicks, encoded as learnable queries with spatial-temporal embeddings, guide a transformer decoder to produce object-level masks. Concurrently, a dedicated part-level branch treats each object as a miniature scene, using a specialized decoder to segment its components. A key innovation is a hierarchical attention mechanism that enables communication between parts and their parent instance. This dual-level decoding is enhanced by an iterative refinement loop with dynamic attention masking, where predicted masks from each step guide the focus of subsequent steps, leading to highly accurate part masks with minimal user input.

Our experimental evaluation across three diverse 3D point cloud datasets demonstrates that PinPoint3D achieves 55.8% average IoU with just a single positive click per target part, exceeding 71.3% IoU with minimal additional user input. Compared to existing interactive segmentation methods, our framework substantially reduces required user effort while improving segmentation accuracy.

Our primary contributions include:

(1) A novel interactive segmentation framework that effi-

†Project Lead.
[1]All authors are with the Department of Computer Science and Engineering, Southern University of Science and Technology. {12211615, 12212012, 11610127, 12532586, 12310817, 12312507 }@mail.sustech.edu.cn
[2]All authors are with Spatialtemporal AI.
*Corresponding author: zfeng02@gmail.com

ciently generates multi-granularity masks from sparse point-based user inputs, bridging the gap between instance and part-level understanding.

(2) A comprehensive 3D data synthesis pipeline, along with a large-scale, scene-level dataset with dense part annotations constructed by this pipeline, addressing the critical data scarcity in this domain.

(3) Robust experimental validation across multiple datasets and unseen scenes, confirming our method's generalization capabilities and practical applicability.

II. RELATED WORKS

A. 3D Interactive Segmentation

There are only a few methods that support interactive 3D segmentation with explicit user input. Valentin et al. [28] and Zhi et al. [36] introduced early systems for online 3D scene labeling, focusing on semantic annotation. Shen et al. [24] projected the interaction to 2D by allowing users to annotate multi-view images through scribbles, but providing feedback from multiple viewpoints is cumbersome. More recent approaches operate directly on 3D point clouds: Kontogianni et al. [14] proposed InterObject3D, which segments objects from user clicks but can only handle a single object each time. Yue et al. [34] developed an attention-guided model that segments multiple objects simultaneously by encoding user clicks as spatial queries, achieving higher accuracy with fewer clicks. Zhou et al. [37] adapts the Segment-Anything concept to 3D point clouds for promptable segmentation and demonstrates strong generalization across domains.

B. Multi-granularity segmentation

Current research on 3D segmentation has largely progressed on scene-level representations that enable multi-scale structure driven by language or interaction. Distillation approaches such as OpenScene [21] and ConceptFusion[9] yield sufficiently fine-grained features but lack explicit hierarchical structure and controllable granularity. GARField[12] represents scene elements using an affinity field and directly adjusts grouping granularity via an extra control parameter. A series of works lifts 2D SAM [13] 's multi-granularity masks into 3D: SAI3D [32] relies on multi-view consistency and region growing to obtain instance/part masks, while SAGA [1] employs *scale-gated affinity* features for promptable and granularity-adjustable 3D segmentation.

C. 3D Part Segmentation

3D part segmentation involves two main tasks: semantic segmentation and instance segmentation. Many 3D networks [22], [23], [27], [29] predict a semantic label for each point or voxel in a 3D shape. Existing learning-based approaches tackle instance segmentation by incorporating various point grouping and clustering strategies [3], [7], [10] or by generating part proposals with 3D bounding boxes and region proposal networks [30], [31].

Recently, vision-language and other foundation models now enable 3D part segmentation with minimal manual labels. PartSLIP [18] transfers knowledge from a pretrained 2D image-language model to detect part regions in multi-view renderings of a 3D object, achieving open-vocabulary zero-shot part segmentation via 2D-to-3D label lifting. PartSTAD [11] adapts 2D SAM [13] to 3D point clouds via few-shot training, improving part segmentation by leveraging 2D pretrained priors.

III. METHOD

A. Overall Architecture

We propose a novel *Two-Level Forward Mask Prediction Architecture* that simultaneously produces object-level and part-level segmentation masks within a unified decoder pipeline, as illustrated in Fig 2. Compared to existing interactive 3D segmentation methods, our framework is specifically designed with **(1) a frozen backbone plus a lightweight adapter**, which preserves stable object-level semantics while enabling fine-grained adaptation to part-level segmentation, and **(2) a dedicated part-level decoder**, which builds upon object-level predictions and refines them into fine-grained part masks. The additional components introduce only a small parameter overhead, see Tab. III.

In order to enhance **hierarchical consistency**, we introduce the Targeted Attention Mask (TAM), which constrains part queries to operate only within the designated target region. This target is not limited to a single geometric object, but can also correspond to a *composite object* formed by grouping multiple instances. By enforcing this hard constraint, TAM effectively suppresses cross-object interference and encodes the intended object→part hierarchy, while in practice it is reinforced during training by simulating clicks within the chosen target region.

B. Data Generation

Our approach requires a large-scale point cloud with part annotations for training. Due to the scarcity of such annotated datasets, we construct a novel training set by integrating data from ScanNet and PartNet. The dataset construction consists of two steps: generating pseudo labels for ScanNet, and adapting PartNet data to ScanNet.

1) Generate Pseudo Labels: ScanNet is an indoor scene dataset offering 3D point clouds with instance-level annotations [4]. For each object instance that can be decomposed into semantic parts, we augment it by generating part-level pseudo-labels using PartField [17]. PartField captures the general concept of parts and their hierarchical structure, generating a continuous feature field for a given object. These learned point-wise representations are then clustered to produce part segmentations. This procedure yields a part segmentation that is both compact and well-separated, aligning with the object's intrinsic structure.

2) Synthetic Data Generation: To overcome the potential imprecision of pseudo-labels derived from ScanNet and to enhance the diversity of our training data, we augment our training data with high-quality assets from the PartNet dataset [20]. PartNet provides fine-grained, hierarchically consistent part annotations for a diverse range of 3D objects. This hybrid data strategy allows our model to learn from both

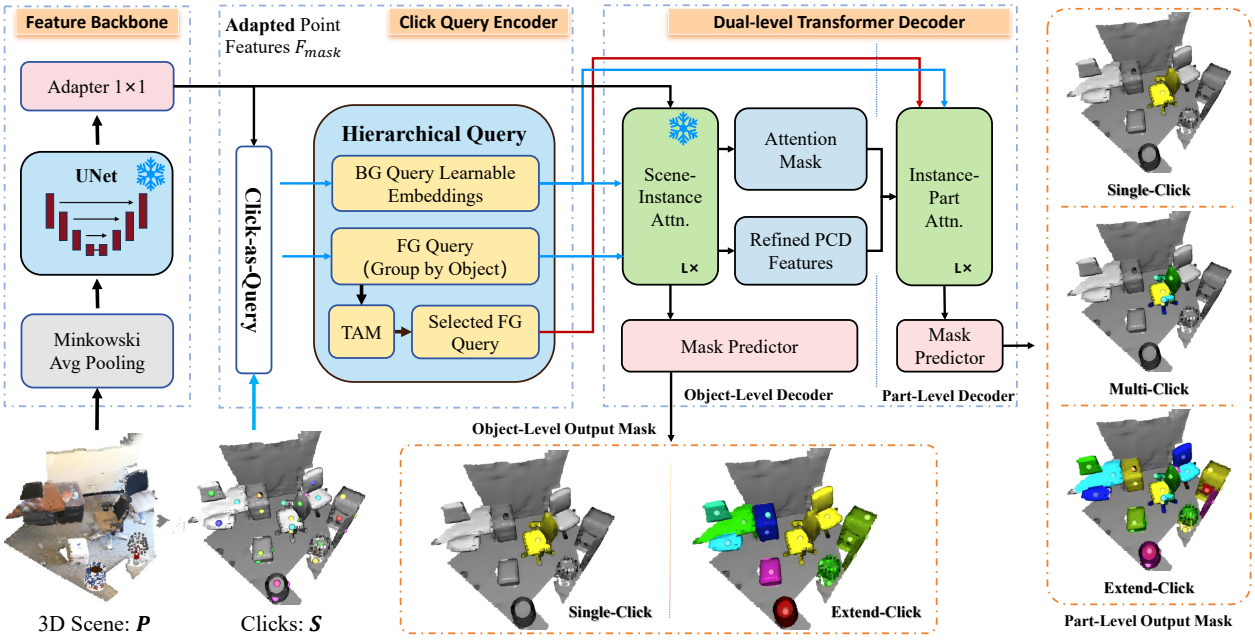


Fig. 2. Hierarchical interactive segmentation pipeline. Given a 3D scene P and user clicks S, the **Feature Backbone** (Minkowski U-Net with a 1×1 Adapter) extracts per-point features. The **Click Query Encoder** forms hierarchical queries and TAM selects queries in the target object. The **Dual-level Transformer Decoder** refines features via Scene–Instance and Instance–Part attention to predict masks. Two heads are provided: an **optional Object-Level Decoder** for holistic masks, and a **Part-Level Decoder** that is trained for fine-grained part segmentation while remaining object-consistent in low-click regimes, yielding object-level masks when required, even without the optional object head.

the high-fidelity part supervision of PartNet and the complex spatial arrangements of real-world indoor scenes from ScanNet, which is crucial for enhancing its generalization capabilities.

Our data integration process involves carefully selecting and placing PartNet objects into ScanNet scenes. We selected 12 object categories from PartNet based on three primary criteria: (1) high prevalence in indoor scenes, (2) scale compatibility with ScanNet environments, and (3) possession of distinct, semantically meaningful part structures. The selected categories from PartNet are: *Table, Refrigerator, StorageFurniture, Chair, Dishwasher, Microwave, Bag, Mug, Bottle, Lamp, Vase*, and *Faucet*.

To align the PartNet objects with the scale and density of ScanNet scenes, we first analyzed key statistical properties, including size, volume, and point density across both datasets. Based on these statistics, we determined an appropriate scaling factor for each PartNet category to match the geometric properties of ScanNet counterparts. We then scaled and downsampled each PartNet object point cloud accordingly. Each object was scaled and downsampled using farthest point sampling, applied independently to each part to preserve local structure and part-level details.

The processed PartNet objects were then inserted into ScanNet scenes to create synthetic indoor environments enriched with accurate part annotations. Specifically, for a given ScanNet scene, we first estimate the floor plane and partition it into a 2D grid. Objects are randomly placed into unoccupied grid cells, one at a time, until either the available space is exhausted or a predefined object limit is reached. This procedure yields augmented 3D scenes that contain both the original ScanNet geometry with pseudo-labeled parts and PartNet objects with accurate part annotations.

C. Model Architecture

Our framework is an attention-based two-stage 3D segmentation model that takes a sparse point cloud and a few user-provided point prompts as input, and produces fine-grained part segmentation masks. It consists of (a) a 3D sparse convolutional **backbone** for feature extraction, (b) a **click query encoder** that generates learnable query embeddings from user clicks, and (c) a dual-level transformer **decoder** with Targeted Attention Masking (TAM) for instance and part decoding.

1) Feature Backbone: We adopt a 3D sparse convolutional backbone [2] for point cloud feature extraction. Operating on a sparse voxel grid, it efficiently encodes the N input points into a feature tensor $\mathbf{F}_{\text{pcd}} \in \mathbb{R}^{N\times C_{\text{bb}}}$ with rich geometric context. To adapt the feature backbone for fine-grained 3D segmentation tasks, we insert a lightweight *residual adapter* that specializes the object-level representations for *part-level* semantics. Inspired by He et al. [6], the adapter applies a bottleneck-style update $\Delta(\cdot)$ and a controlled residual scaling. Formally, given the backbone features $\mathbf{F}_{\text{pcd}}$, the adapter produces

$$\mathbf{F}_{\text{mask}} = \mathbf{F}_{\text{pcd}} + \alpha \cdot \text{Conv}^2_{1\times1}\Big(\text{ReLU}\big(\text{Conv}^1_{1\times1}(\mathbf{F}_{\text{pcd}})\big)\Big), \quad (1)$$

where $\text{Conv}^1_{1\times1}$ reduces channels and $\text{Conv}^2_{1\times1}$ expands them; $\alpha \in (0, 1]$ stabilizes residual updates. This design keeps

object-level geometry while endowing $\mathbf{F} \in \mathbb{R}^{N \times D}$ with part-sensitive cues for decoding.

2) Click Query Encoder: We encode the user's point prompts (clicks) as a set of learnable query embeddings. We encode each click using Fourier positional encoding [26]. Furthermore, for interactive settings where clicks are provided sequentially, we add a 1-dimensional temporal encoding to each query, indicating the click order. After the encoding steps, queries are sent into a dual-level transformer decoder module for further refinement.

3) Dual-level Transformer Decoder: We design a dual-level decoder that contains a Scene-Instance decoder, a Targeted Attention Masking(TAM) module, and an Instance-Part decoder. This transformer decoder takes the scene features and user query embeddings as input, producing refined representations for downstream mask prediction at both instance and part levels.

a) Scene-Instance decoder: The Scene-Instance decoder processes the foreground and background queries C_f and C_b, together with the input scene feature F_{pcd}, to obtain updated query embeddings and updated point cloud features. A Scene-Instance Attention Module allows *bidirectional* interaction between point prompt queries and point cloud features, as in AGILE3D [33]. Attention flows from instance queries into scene features to gather contextual information, then conversely flows from scene features back to instance queries to update the scene representation. Finally, attention is applied among instance queries for direct information exchange.

b) Mask Prediction and Targeted Attention Masking(TAM): The refined queries generated by the Scene-Instance decoder are fed into a mask prediction module to produce binary masks for each object query. This module uses a learned MLP to map the query embedding vectors to scalar mask logits for each point. Specifically, let $\{\mathbf{w}\}$ denote the learned mask embeddings (one for each query). The logit for query i at point p is computed as the dot product between the point's feature $\mathbf{F}_{\text{mask}}(p)$ and the query's embedding $\mathbf{w}_i$:

$$z_{i,p} = \mathbf{F}_{\text{mask}}(p)^\top \mathbf{w}_i.$$

These logits are then converted to per-point class assignments via a $\max$ operation.

Built on AGILE3D's object-level semantics [33], TAM converts the current objectness signals into an attention constraint for the part decoder. The target is not restricted to a single instance but may also correspond to a *composite object* formed by grouping several instances. In this case, TAM not only suppresses cross-object interference but also encourages the decoder to learn how to partition parts consistently across the entire composite region, rather than conflating them with the background.

Given object-level predictions $\hat{y}_n \in \{0, \ldots, M\}$, we derive a binary mask $\mathbf{A}^{(t)} \in \{0,1\}^{Q \times N}$ for target instance t, where $\mathbf{A}^{(t)}_{q,n} = 0$ if query q is allowed to attend to point n, and 1 otherwise. During part decoding, querypoint attention is restricted as

$$\alpha_{q,n} \;=\; \frac{\exp(s_{q,n}) \cdot \mathbb{1}[\mathbf{A}^{(t)}_{q,n} = 0]}{\sum_{n':\mathbf{A}^{(t)}_{q,n'}=0} \exp(s_{q,n'})} \tag{2}$$

This module encodes the object→part hierarchy: foreground queries for object t attend only to its interior, while background queries attend only to background regions. In practice, $\mathbf{A}^t$ is passed as the transformer *memory mask* between the object-level decoder and the part-level decoder (Fig. 2).

c) Instance-Part Decoder: For the selected object i, a variable number of part queries $\mathbf{P}^i \in \mathbb{R}^{N^i_p \times d}$ interact with the object-restricted point set $\mathbf{O}^i$ (points gated by TAM). Each layer applies masked cross-attention and self-attention, followed by intra-part self-attention and FFNs with residual connections and normalization. The decoder supports iterative refinement: users can add clicks to update $\mathbf{P}^i$ and re-run masked attention, yielding refined part hypotheses $\mathbf{P}^i_{\text{final}}$ and precise part masks.

D. Training and Evaluation Protocol

1) Iterative Multi-Part Training Protocol: We follow the interactive training paradigm of AGILE3D [33], where users iteratively provide simulated clicks. At each refinement step, positive clicks are sampled from parts of the selected objects, while negative clicks are drawn from background regions. Unlike protocols restricted to a single object, we simulate a more realistic scene-level interaction by randomly selecting parts across all objects present in the scene. The number of active parts per interaction is capped at 10, ensuring that each forward pass maintains a tractable scope while still covering diverse object–part configurations. This random yet bounded sampling strategy promotes balanced supervision across scenes of varying complexity, while preserving the realism of simulated annotation. As in AGILE3D, gradients are only backpropagated at the final refinement step to reduce computational overhead.

2) Click Simulation for Scene-Level Parts: In contrast to object-centric settings, our framework simulates part clicks at the scene level. For evaluation, we construct two validation protocols for each object: (i) *multi-part*, where a random number of parts between 1 and the maximum available parts are annotated; and (ii) *full-part*, where all parts of the object are annotated. This design allows us to assess model performance under different annotation budgets, ranging from minimal user input to complete coverage.

3) Optimization Objective: Both object-level and part-level predictions are supervised by a combination of cross-entropy and Dice loss, encouraging precise boundary alignment in cases of class imbalance. During training, the backbone network remains frozen while the adapter and two decoders are being updated.

$$\mathcal{L}_{\text{part}} = \frac{1}{N_{\text{part}}} \sum_{p \in P_{\text{part}}} w_p \left(\lambda_{\text{CE}}\, \mathcal{L}_{\text{CE}}(p) + \lambda_{\text{Dice}}\, \mathcal{L}_{\text{Dice}}(p)\right), \tag{3}$$

This stabilizes object-level semantics while enabling fine-grained adaptation to part-level segmentation.

TABLE I
COMPARISON OF METHODS ON SCENE PART SEGMENTATION ABILITY.

Method	**Eval**	$\text{IoU}_1 \uparrow$	$\text{IoU}_3 \uparrow$	$\text{IoU}_5 \uparrow$	$\text{NoC}_{50} \downarrow$	$\text{NoC}_{65} \downarrow$	$\text{NoC}_{80} \downarrow$	$\text{AP}_{25\%} \uparrow$	$\text{AP}_{50\%} \uparrow$
PointSAM	SyntheticData(random-part)	46.2	50.1	51.4	-	-	-	72.8	49.9
AGile3D		39.8	58.4	64.9	3.07	5.38	7.89	94.4	73.5
PinPoint3D(Ours)		**50.0**	**65.9**	**69.7**	**2.12**	**3.92**	**6.92**	**95.1**	**81.5**
PointSAM	SyntheticData(all-part)	48.4	52.6	52.7	-	-	-	74.1	51.0
AGile3D		39.1	61.1	66.7	2.67	5.18	8.12	96.7	78.2
PinPoint3D(Ours)		**55.8**	**68.4**	**71.3**	**1.68**	**3.46**	**6.43**	**96.9**	**85.7**
PointSAM	MultiScan(random-part)	**44.4**	54.9	58.1	-	-	-	81.7	57.6
AGile3D		40.8	59.3	66.5	2.88	5.24	7.75	93.4	74.9
PinPoint3D(Ours)		44.0	**60.8**	**66.8**	**2.71**	**4.93**	**7.74**	**94.0**	**77.3**
PointSAM	MultiScan(all-part)	**44.9**	54.0	56.1	-	-	-	80.8	58.3
AGile3D		42.1	61.2	67.5	2.62	4.90	7.85	93.7	76.5
PinPoint3D(Ours)		44.4	**62.7**	**68.1**	**2.28**	**4.53**	**7.66**	**93.9**	**78.9**

IV. EXPERIMENT

We evaluate PinPoint3D on the fine-grained 3D scene part segmentation task. Lacking a standardized protocol, we adapt the evaluation methodology from AGILE3D[34]. Our evaluation covers both in-domain and cross-domain data to assess performance and generalization.

Datasets We trained PinPoint3D on our dataset PartScan, a synthesized dataset with part-level ground truth and real data with pseudo labels, as mentioned in section III-B. A dedicated test split is reserved to ensure no overlap with the training set, which provides human-verified part annotations for 12 categories derived from PartNet [20]. To assess the generalization ability of our model, we further adopt the MultiScan [19] dataset. MultiScan is a large-scale RGB-D dataset with part-level annotations, containing 10,957 objects and 5,129 parts, making it well-suited for evaluating cross-domain generalization.

Evaluation metrics We evaluate using (1) $\text{NoC}_q \downarrow$, the average number of clicks required to reach $q\%$ IoU, and (2) $\text{IoU}_k \uparrow$, the average IoU after k user clicks per part (capped at 10). We also report IoU@k, which measures IoU after k clicks per object.

For comparison with non-interactive methods, we further report weighted mIoU, computed as the IoU averaged across parts with weights proportional to the number of points in each part. This provides a more balanced evaluation when part sizes vary significantly.

Baseline We compare PinPoint3D with two baselines. PointSAM [37] is an interactive point-cloud segmentation model designed for object-level segmentation but applicable to part-level tasks. AGILE3D, originally developed for object segmentation, can also produce part-level results via repeated user clicks.

A. Evaluation on Scene-Part Segmentation

Table I compares results on scene part segmentation. We adopt two testing strategies: the random-part strategy, where a randomly selected part of an object is segmented, and the all-part strategy, where all parts of an object are segmented. We report the $\text{NoC}_q\%$, IoU_k, and AP metrics for our model and the baselines. For PointSAM, the NoC_q metric is not reported, as it often fails to reach the required IoU threshold on our test set.

In random-part segmentation tasks, our model achieves higher final IoU and requires fewer user clicks to reach a given IoU threshold. This advantage becomes even more pronounced in the all-part segmentation setting. This suggests PinPoint3D effectively captures the structural relationships among object parts, handling complex segmentation scenarios more efficiently, as further illustrated by representative annotation results in Fig. 3. We also conducted a cross-dataset generalization test on the KITTI Odometry dataset, with an example result shown in Fig. 4.

TABLE II
COMPARISON OF METHODS ON OBJECT SEGMENTATION ABILITY

Method	Test dataset	$\text{IoU}_1 \uparrow$	$\text{IoU}_3 \uparrow$	$\text{IoU}_5 \uparrow$
AGILE3D	PartScan	83.6	96.9	97.7
PinPoint3D (ours)		**86.7**	**97.0**	**98.0**
AGILE3D	Multiscan	**58.5**	**75.0**	**81.0**
PinPoint3D (ours)		57.1	72.3	78.6

B. Evaluation on Object Segmentation

In this part, we evaluate the object segmentation capability of our model. Since our method is an extension of AGILE3D, we compare it with the original baseline. The results are presented in Table II and show that our model achieves improved object segmentation performance on the synthetic dataset, while on the MultiScan generalization test set, the difference compared with AGILE3D is marginal. This indicates that our approach preserves the object-level segmentation ability of AGILE3D, while further enabling multi-granularity segmentation capability.

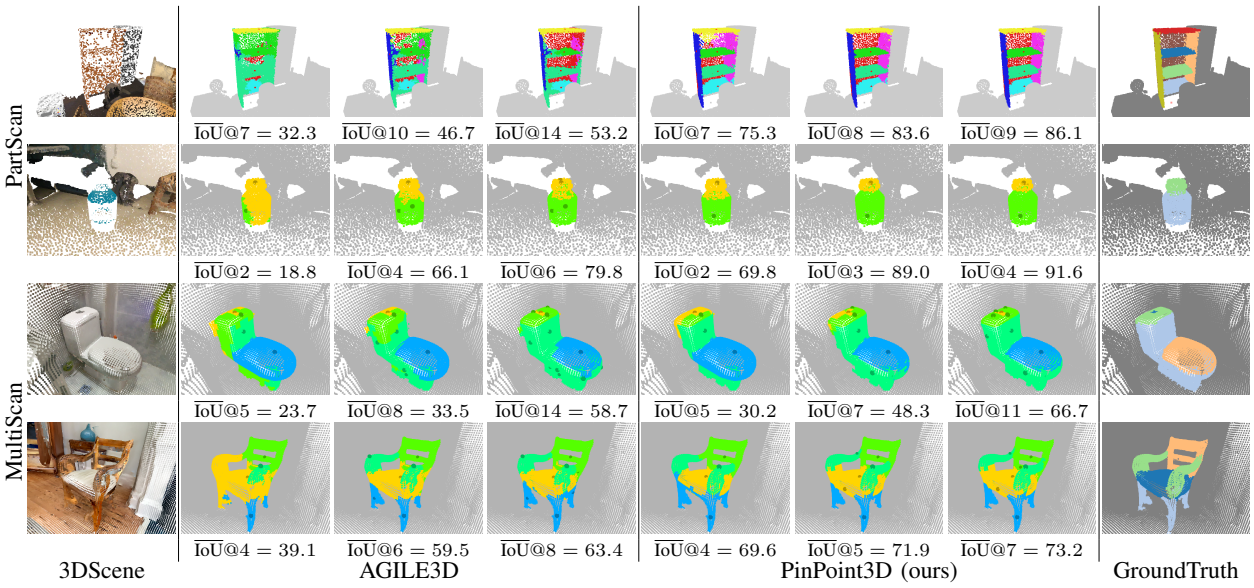


Fig. 3. **Qualitative comparison on interactive part segmentation.**

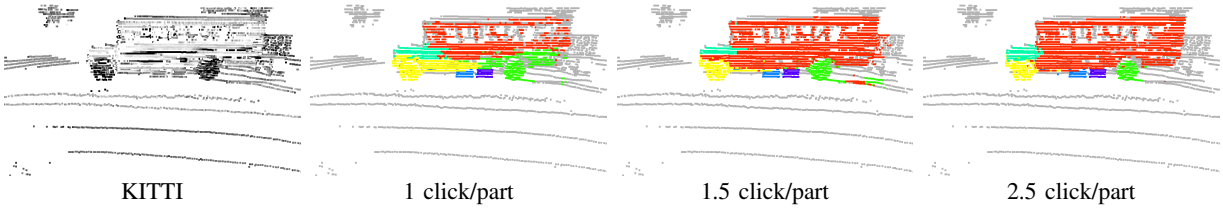


Fig. 4. **Cross-dataset generalization test on KITTI Odometry Dataset**

TABLE III
PARAMETER GROWTH OF PINPOINT3D COMPONENTS

Model	**Params (M)**	**Δ**
Base Model	39.31	–
+ Adapter	0.03	+0.09%
+ Part-level Transformer	1.39	+3.53%
+ Part Mask Head	0.03	+0.08%
Final Model	40.76	+3.70%

V. ABLATION STUDY

A. Architecture design

We first examine two key architectural design choices: the adapter in the feature encoder and a dedicated part transformer branch.

For the adapter, we compare direct fine-tuning of the backbone with using a frozen backbone plus lightweight adapters. As shown in Tab. IV and Tab. VI, direct fine-tuning yields only marginal part-level gains (IoU@1) while destabilizing object-level representations. In contrast, combining the adapter with the frozen backbone preserves the semantic feature space and maintains strong object-level accuracy, while still providing significant improvements in part segmentation.

We also evaluate the necessity of the instance-part transformer decoder. Previous methods on Interactive Instance Segmentation can be adapted to part segmentation by adding more point prompts for filtering. Thus, we remove the part transformer and train a unified decoder, directly utilizing the mask module to predict part masks from the output of the object-level decoder. The **Architecture** block of Table. IV indicates that the model's part segmentation accuracy drops sharply without the part transformer, since partitioning relies only on coarse object-level features. Including the part transformer enables a progressive refinement of representations from object-level to part-level, yielding improvements in fine-grained part segmentation performance.

B. Training Strategy

In previous part segmentation tasks, we train on a single object setting, meaning only one targeted object is partitioned at a time. Such a protocol naturally ensures that part predictions are strictly constrained within the object. In contrast, scene-level part segmentation introduces multiple objects simultaneously, which raises another training protocol: part learning across different objects.

Single-Object vs. Multi-Object Training. We compare two training protocols for the part decoder: single-object, where only one object's parts are clicked per iteration, and multi-object, where clicks from different objects appear jointly. Since the decoder still receives a single target ID, the latter forces it to interpret dispersed clicks as a "composite

TABLE IV

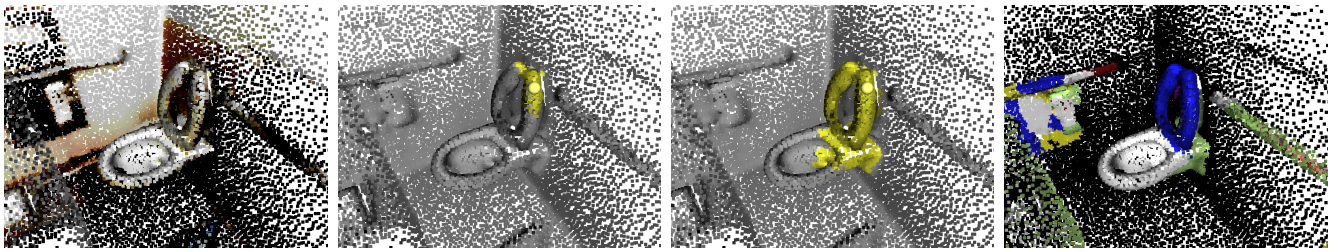
ABLATIONS ON ARCHITECTURE AND TRAINING STRATEGY (EVALUATED ON PARTSCAN).

Variant	$IoU_1 \uparrow$	$IoU_3 \uparrow$	$IoU_5 \uparrow$	$NoC_{50} \downarrow$	$NoC_{65} \downarrow$	$NoC_{80} \downarrow$
Baseline (PinPoint3D)	55.8	**68.4**	**71.6**	**1.70**	**3.83**	**7.10**
Architecture						
No Part-Transformer	43.7	60.2	65.0	2.54	5.46	8.36
No Adapter (Unfreezing Backbone)	**56.3**	67.5	70.4	1.73	3.97	7.40
Training Strategy						
Single-Object	52.5	67.5	70.8	1.79	3.96	7.38

TABLE V

ABLATIONS ON TRAINING DATA (EVALUATED ON MULTISCAN).

Data Variant	$IoU_1 \uparrow$	$IoU_3 \uparrow$	$IoU_5 \uparrow$	$NoC_{50} \downarrow$	$NoC_{65} \downarrow$	$NoC_{80} \downarrow$
PartScan (ours)	**44.4**	**62.7**	**68.1**	**2.37**	**4.86**	8.06
PartNet (in-scene)	43.3	62.4	67.8	2.50	4.91	8.10
ScanNet-PartField	43.1	61.6	67.4	2.65	5.10	**8.05**



(a) ScanNet (b) Single-Object Training (c) Multi-Object Training (d) Pseudo parts on ScanNet

Fig. 5. **Qualitative comparison of training protocols.** Single-object training often suffers from excessive competition and coupling within self-attention, causing each click to cover only a small region and producing fragmented masks. In contrast, multi-object training alleviates this over-coupling, cover a larger extent with each click, resulting in more complete part coverage.

object". As shown in the Training Strategy block of Tab. IV, multi-object training consistently yields higher part IoU.

We attribute this improvement to three factors: (i) multi-object sampling introduces more dispersed and heterogeneous click distributions, forcing the model to rely on local evidence rather than assuming all clicks belong to a single connected region; (ii) the frozen object decoder tends to produce over-merged masks, which compels the part decoder to learn corrective splitting, transferring to single-object evaluation as more conservative mask growth and sharper boundaries; and (iii) queries sampled from more distant and heterogeneous regions thereby reduce excessive coupling (see Fig. 5), and under the combined influence of loss constraints and residual coordination in the shared scene, undergo soft competition that makes them act as specialized local experts—achieving more complete coverage while maintaining boundaries as clean as possible.

C. Training Dataset Composition

We study the impact of training dataset composition on segmentation performance. We compare training the model on: (i) PartNet-only, using synthetic objects with fine part annotations integrated into scenes; (ii) pseudo-label only, using real scanned scenes with pseudo-labeled parts generated by PartField; and (iii) a combined dataset PartScan containing both sources. All models are evaluated identically on the MultiScan validation set. Table V shows different performance.

TABLE VI

ADAPTER/BACKBONE STRATEGY FOR OBJECT SEGMENTATION.

Protocol	Obj $IoU_1 \uparrow$	Obj $IoU_3 \uparrow$	$AP_{25\%} \uparrow$	$AP_{50\%} \uparrow$
A+F	**86.7**	**97.0**	**90.6**	**89.9**
NA+UF	58.9	92.2	88.5	85.6

A+F = Adapter + Frozen Backbone;
NA+UF = No Adapter + Unfrozen Backbone.

ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China (Grant No. 2024YFE0203100).

REFERENCES

[1] Jiazhong Cen, Jiemin Fang, Chen Yang, Lingxi Xie, Xiaopeng Zhang, Wei Shen, and Qi Tian. Segment any 3d gaussians. *arXiv preprint arXiv:2312.00860*, 2025.

[2] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 4d spatio-temporal convnets: Minkowski convolutional neural networks. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3070–3079, 2019.

[3] Ruihang Chu, Yukang Chen, Tao Kong, Lu Qi, and Lei Li. ICM-3D: Instantiated category modeling for 3d instance segmentation. *IEEE Robotics and Automation Letters*, 7(1):57–64, 2021.

[4] 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 conference on computer vision and pattern recognition*, pages 5828–5839, 2017.

[5] Junming Fan, Pai Zheng, and Carman K. M. Lee. A multi-granularity scene segmentation network for human-robot collaboration environment perception. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2105–2110, 2022.

[6] Ruidan He, Linlin Liu, Hai Ye, Qingyu Tan, Bosheng Ding, Liying Cheng, Jiawei Low, Lidong Bing, and Luo Si. On the effectiveness of adapter-based tuning for pretrained language model adaptation. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2208–2222, Online, August 2021. Association for Computational Linguistics.

[7] Tong He, Dong Gong, Zhi Tian, and Chunhua Shen. Learning and memorizing representative prototypes for 3d point cloud semantic and instance segmentation. In *Proceedings of the European Conference on Computer Vision*, pages 564–580, 2020.

[8] Wei-Ling Hsu, Qiangui Huang, Song-Chun Zhu, and Yixin Zhu. OPD: Single-view 3d openable part detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 262–278, 2022.

[9] Krishna Murthy Jatavallabhula, Alihusein Kuwajerwala, Qiao Gu, Mohd Omama, Tao Chen, Alaa Maalouf, Shuang Li, Ganesh Iyer, Soroush Saryazdi, Nikhil Keetha, Ayush Tewari, Joshua B. Tenenbaum, Celso M. de Melo, Madhava Krishna, Liam Paull, Florian Shkurti, and Antonio Torralba. Conceptfusion: Open-set multimodal 3d mapping. In *Robotics: Science and Systems (RSS)*, 2023.

[10] 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.

[11] Hyunjin Kim and Minhyuk Sung. PartSTAD: 2d-to-3d part segmentation task adaptation. In *Proceedings of the European Conference on Computer Vision*, 2024.

[12] Minhyuk Kim, Baifeng Wu, Jacob Kerr, Ken Goldberg, Matthew Tancik, and Angjoo Kanazawa. Garfield: Group anything with affinity field. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[13] 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.

[14] Theodora Kontogianni, Ecenur Celikkan, Shuaicheng Tang, and Konrad Schindler. Interactive object segmentation in 3d point clouds. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, pages 2891–2897, 2023.

[15] Itai Lang, Fei Xu, Dale Decatur, Sudarshan Babu, and Rana Hanocka. iSeg: Interactive 3d segmentation via interactive attention. In *ACM SIGGRAPH Asia Conference Papers*. ACM, 2024.

[16] Liulei Li, Tianfei Zhou, Wenguan Wang, Jianwu Li, and Yi Yang. Deep hierarchical semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1244–1254, 2022.

[17] Minghua Liu, Mikaela Angelina Uy, Donglai Xiang, Hao Su, Sanja Fidler, Nicholas Sharp, and Jun Gao. Partfield: Learning 3d feature fields for part segmentation and beyond. *arXiv preprint arXiv:2504.11451*, 2025.

[18] Minghua Liu, Yinhao Zhu, Hong Cai, Shizhong Han, Zhan Ling, Fatih Porikli, and Hao Su. PartSLIP: Low-shot part segmentation for 3d point clouds via pretrained image-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[19] Yongsen Mao, Yiming Zhang, Hanxiao Jiang, Angel Chang, and Manolis Savva. Multiscan: Scalable rgbd scanning for 3d environments with articulated objects. *Advances in neural information processing systems*, 35:9058–9071, 2022.

[20] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019.

[21] Songyou Peng, Kyle Genova, Chiyu "Max" Jiang, Andrea Tagliasacchi, Marc Pollefeys, and Thomas Funkhouser. Openscene: 3d scene understanding with open vocabularies. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[22] Charles R. 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 (NeurIPS)*, volume 30, 2017.

[23] Guocheng Qian, Yuchen Li, Houwen Peng, Jinjie Mai, Hasan Hammoud, Mohamed Elhoseiny, and Bernard Ghanem. PointNeXt: Revisiting pointnet++ with improved training and scaling strategies. *arXiv preprint arXiv:2206.04670*, 2022.

[24] Tianchang Shen, Jun Gao, Amlan Kar, and Sanja Fidler. Interactive annotation of 3d object geometry using 2d scribbles. In *European Conference on Computer Vision (ECCV)*, pages 751–767. Springer, 2020.

[25] Wenshan Sun, Zhongjie Luo, Yunjie Chen, Hao Li, Julian Marcato, Wesley N. Gonçalves, and Jizhong Li. A click-based interactive segmentation network for point clouds. *IEEE Transactions on Geoscience and Remote Sensing*, 61(12):1–13, 2023.

[26] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ramamoorthi, Jonathan Barron, and Ren Ng. Fourier features let networks learn high frequency functions in low dimensional domains. *Advances in neural information processing systems*, 33:7537–7547, 2020.

[27] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, Francois 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.

[28] Julien Valentin, Vibhav Vineet, Ming-Ming Cheng, David Kim, Jamie Shotton, Pushmeet Kohli, Matthias Nießner, Antonio Criminisi, Shahram Izadi, and Philip Torr. Semanticpaint: Interactive 3d labeling and learning at your fingertips. *ACM Transactions on Graphics (TOG)*, 34(5):154:1–154:17, 2015.

[29] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E. Sarma, Michael M. Bronstein, and Justin M. Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics*, 38(5):1–12, 2019.

[30] Bo Yang, Jianan Wang, Ronald Clark, Qingyong Hu, Sen Wang, Andrew Markham, and Niki Trigoni. Learning object bounding boxes for 3d instance segmentation on point clouds. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[31] Li Yi, Wang Zhao, He Wang, Minhyuk Sung, and Leonidas J. Guibas. GSPN: Generative shape proposal network for 3d instance segmentation in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3947–3956, 2019.

[32] Yingda Yin, Yuzheng Liu, Yang Xiao, Daniel Cohen-Or, Jingwei Huang, and Baoquan Chen. Sai3d: Segment any instance in 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[33] Yiming Yue, Shravya Mahadevan, Jonas Schult, Francis Engelmann, Bastian Leibe, Konrad Schindler, and Theodora Kontogianni. AGILE3D: Attention guided interactive multi-object 3d segmentation. *arXiv preprint arXiv:2306.00977*, 2023.

[34] 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 (ICLR)*, 2024.

[35] Peng Zhang, Tong Wu, Jiaming Sun, Weikai Li, and Zhizhong Su. Refining segmentation on-the-fly: An interactive framework for point cloud semantic segmentation. *arXiv preprint arXiv:2403.06401*, 2024.

[36] Shuaifeng Zhi, Edgar Sucar, André Mouton, Iain Haughton, Tristan Laidlow, and Andrew J. Davison. iLabel: Revealing objects in neural fields. *IEEE Robotics and Automation Letters*, 8(2):832–839, 2022.

[37] Yuchen Zhou, Jiayuan Gu, Tung-Yen Chiang, Fanbo Xiang, and Hao Su. Point-sam: Promptable 3d segmentation model for point clouds. In *International Conference on Learning Representations (ICLR)*, 2025.