Towards Efficient and Effective Interactive 3D Segmentation

Wei Cong[1,2], Yang Cong[3]*, Jiahua Dong[4], Gan Sun[3]
[1]State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang 110016, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]College of Automation Science and Engineering, South China University of Technology, Guangzhou 510640, China
[4]Mohamed bin Zayed University of Artificial Intelligence, UAE
{congwei45, congyang81, dongjiahua1995, sungan1412}@gmail.com

Abstract

Interactive 3D segmentation embodies an advanced human-in-the-loop paradigm, where a model iteratively refines the segmentation of interested objects within a 3D point cloud through user feedback. Existing methods have achieved notable advancements at the expense of substantial resource consumption. To address this challenge, we introduce E^{2}I3D, an efficient and effective model for interactive 3D segmentation. Specifically, we propose a two-stage efficiency-to-effectiveness framework to decouple efficiency and effectiveness, avoiding the high training cost of joint optimization. For efficiency in the first stage, we present heterogeneous pruning, which reliably compresses the model by ranking and pruning the constructed heterogeneous groups separately based on gradient compensation. For effectiveness in the second stage, we design hierarchical click-aware attention that integrates geometric details from high-resolution features with global context from low-resolution features to enhance click-guided interaction. Extensive experiments across public datasets demonstrate that E^{2}I3D exceeds state-of-the-art methods in both efficiency and effectiveness. For instance, on the KITTI-360 dataset, E^{2}I3D boosts the IoU for interactive single-object segmentation from 44.4% to 49.0% with 5 user clicks, while simultaneously reducing parameters from 39.3M to 5.7M.

Introduction

3D instance segmentation (Shin et al. 2024) plays a fundamental role in robotic vision (Cong et al. 2023, 2025), enabling robots to understand and interact with their surroundings by identifying individual objects in complex 3D environments. However, current approaches (Kolodiazhnyi et al. 2024; Roh et al. 2024) rely heavily on large-scale manually labeled datasets, the annotation of which is both labor-intensive and time-consuming. Interactive segmentation techniques (Xu et al. 2016; Sofiiuk et al. 2020; Ding et al. 2020; Kontogianni et al. 2022; Yue et al. 2024) have emerged as a promising alternative to alleviate this challenge, allowing users to guide the segmentation process through point clicks (Xu et al. 2016; Kontogianni et al. 2022), scribbles (Shen et al. 2020), or bounding boxes (Ling et al. 2019). Although interactive 2D segmentation (Xu et al. 2016; Sofiiuk et al. 2020) has achieved remarkable success, its extension to 3D point clouds (Kontogianni et al. 2022; Han et al. 2024; Sun et al. 2023; Yue et al. 2024) remains underexplored.

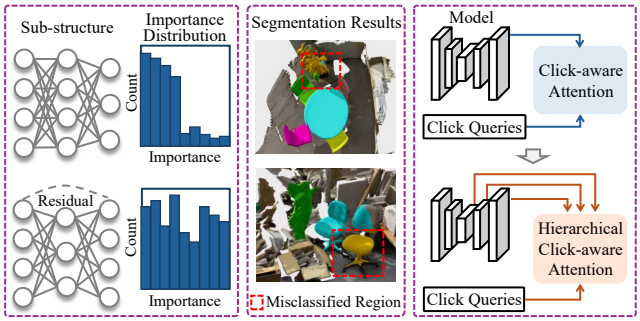

Figure 1: Illustration of challenges. 1) Heterogeneous sub-structures exhibit uneven importance, making uniform pruning problematic. 2) Class-dependent prediction bias leads raw gradients to misrepresent weight importance. 3) The click-aware attention lacks interactions between clicks and multi-scale features, limiting contextual modeling.

Existing interactive 3D segmentation methods (Kontogianni et al. 2022; Han et al. 2024; Sun et al. 2023; Yue et al. 2024) primarily focus on improving segmentation quality, demanding high computing power and memory footprint. To tackle this, we propose a two-stage efficiency-to-effectiveness framework to decouple efficiency and effectiveness, where each stage is functionally distinct yet logically connected to achieve high performance with low resource cost. This design is motivated by the observation that joint optimization (He et al. 2018) often leads to high training costs. In the first stage, we focus on enhancing model efficiency through the lightweight model design. The second stage builds upon this efficient backbone and compensates for the reduced capacity via enhancing interactions with multi-scale scene features. The proposed framework facilitates practical deployment in real-world scenarios by simultaneously achieving high segmentation performance with low resource consumption.

In the first stage, we address the challenge of cumbersome architectures and large parameter counts adopted by recent interactive 3D segmentation methods (Kontogianni et al. 2022; Han et al. 2024). Compared to straightforward

*Corresponding author.
Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

lightweight architectures (Qian et al. 2022) that often compromise accuracy, model compression has become essential to achieve efficient inference without sacrificing segmentation quality. Among various compression techniques, structured pruning (Luo, Wu, and Lin 2017; He, Zhang, and Sun 2017) stands out as a practical and hardware-friendly solution to methods such as knowledge distillation (Han et al. 2015; Kim, Park, and Kwak 2018; Zhu, Gong et al. 2018) and parameter quantization (Shang et al. 2023; Han, Mao, and Dally 2016; Liu et al. 2023). As shown in Fig. 1, existing structured pruning strategies encounter two key challenges: 1) The importance distribution across heterogeneous sub-structures varies significantly, making uniform pruning problematic; 2) The discrepancies in class-wise segmentation performance differ significantly, making raw gradient-based weight importance estimation biased. To address these limitations, we propose a heterogeneous pruning module tailored for interactive 3D segmentation. It models network sub-structures as graphs and clusters them into groups with identical computational topologies. Owing to their structural consistency, intra-group sub-structures exhibit comparable importance, facilitating fair importance evaluation. Then we reweight the loss function based on gradient compensation, enabling a more reasonable estimation of weight importance. This strategy allows for structure-aware model compression that preserves performance while reducing resource cost.

In the second stage, our aim is to elevate effectiveness after reducing the model parameters. As shown in Fig. 1, previous work (Yue et al. 2024) introduces click-aware attention to model the interaction between user clicks and single-scale scene features. However, such strategies often fall short in complex 3D environments due to their limited capacity to jointly capture fine-grained geometry and global context. In this paper, we design a hierarchical click-aware attention module to facilitate joint interactions between user clicks and scene features across multiple scales. High-resolution features align with clicks for precise boundary localization, while low-resolution features capture global context for robust segmentation. This design effectively fuses local geometry and global semantics, thereby benefiting 3D point clouds with large scale and spatial variations. Moreover, our module adaptively modulates attention across feature levels based on click distribution, alleviating sparsity and ambiguity in user inputs and improving generalization in complex 3D scenes. Our main contributions are as follows:

- We propose an **E**fficient and **E**ffective model to surmount **I**nteractive **3D** Segmentation (E^{2}I3D). In specific, we design a two-stage efficiency-to-effectiveness framework, which is the first attempt to balance both efficiency and effectiveness in the field of interactive 3D segmentation.
- To improve efficiency in the first stage, we develop a heterogeneous pruning module that ranks and prunes heterogeneous groups independently based on gradient compensation, enabling more reliable compression.
- For effectiveness in the second stage, we propose a hierarchical click-aware attention module that explicitly treats user clicks as informative guidance cues, facilitating comprehensive interactions with multi-scale scene features.
- Extensive experiments on several public datasets (*i.e.,* ScanNetV2, S3DIS and KITTI-360) demonstrate that our proposed E^{2}I3D significantly surpasses state-of-the-art methods in terms of both efficiency and effectiveness.

Related Works

Interactive 3D Segmentation. Several methods (Valentin et al. 2015; Shen et al. 2020; Zhi et al. 2022; Kontogianni et al. 2022; Yue et al. 2024) have explored interactive 3D segmentation. SemanticPaint (Valentin et al. 2015) and iLabel (Zhi et al. 2022) mainly target online semantic annotation in 3D environments, without focusing on instance segmentation. Scribble3D (Shen et al. 2020) transfers the user interaction to the 2D image space, assuming the availability of multi-view images with accurate camera calibration. This requirement makes the process of providing feedback across different views both cumbersome and inefficient. InterObject3D (Kontogianni et al. 2022) operates directly on the 3D point cloud, which trains the model by concatenating the 3D point cloud with user clicks as input. AGILE3D (Yue et al. 2024) encodes user clicks as spatial-temporal queries, enabling interactions not only among the click queries themselves but also between the queries and the 3D scene. Despite the success of current interactive 3D segmentation, their practical deployment is often hindered by high memory and computational demands. This motivates our focus on designing lightweight alternatives to improve efficiency.

Model Compression. Model compression (Xu and McAuley 2023) aims to enhance efficiency and deployability without compromising performance. Common techniques include distillation (Han et al. 2015; Kim, Park, and Kwak 2018; Zhu, Gong et al. 2018), quantization (Shang et al. 2023; Han, Mao, and Dally 2016; Liu et al. 2023), pruning (Guo, Yao, and Chen 2016; Luo, Wu, and Lin 2017; He, Zhang, and Sun 2017; Fang et al. 2024), and low-rank factorization (Sainath et al. 2013). Among these, structured pruning (Luo, Wu, and Lin 2017; He, Zhang, and Sun 2017) emerges as a compelling choice due to its simplicity and compatibility with off-the-shelf hardware. Most structured pruning strategies estimate channel/filter importance independently based on local statistics, *e.g.,* L1 norm heuristics (Li et al. 2017), centered filter elimination (He et al. 2019), gradient-based metrics (Liu and Wu 2019) and rank analysis (Lin et al. 2020). In contrast, recent methods (Fang et al. 2023; Wu et al. 2024; Fang et al. 2024) group and prune parameters according to inter-layer dependencies. However, these techniques are primarily developed for general classification tasks. There is still no efficient and effective model specifically tailored for interactive 3D segmentation, leaving an open gap in this area.

Method

Preliminaries

Let $P \in \mathbb{R}^{N \times C}$ denote a 3D scene composed of N points, where each point $p_n \in P$ is associated with the C-dimensional feature (*i.e.,* $C = 3$ for only coordinates *xyz* or $C = 6$ if color *rgb* is included). $S = \{c_1, c_2, \ldots, c_k\}_{k=1}^{K}$ represents the sequence of user clicks, where each c_k corresponds to the k-th user click. The goal of interactive 3D

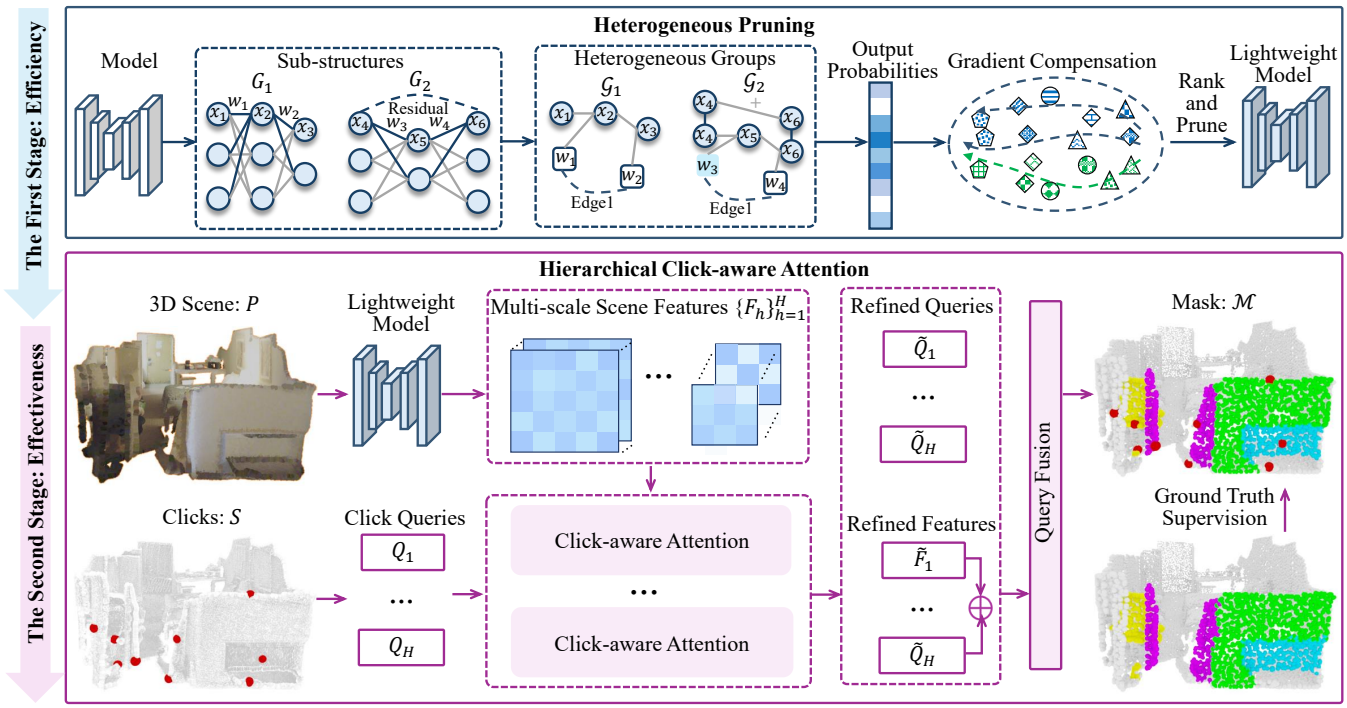

Figure 2: The overview of our E^{2}I3D. It trains with the two-stage efficiency-to-effectiveness framework. In the first stage, it designs heterogeneous pruning to remove redundant sub-structures for efficiency. In the second stage, it proposes hierarchical click-aware attention to improve effectiveness by enhancing interactions between user clicks and multi-scale scene features.

segmentation is to iteratively segment objects of interest with user feedback. Initially, given the 3D scene P and the click sequence S, the model predicts a segmentation mask over the entire 3D point cloud. The user then extends S with corrective clicks, *i.e.,* positive clicks on the desired objects or negative clicks on the background. Specifically, positive clicks recover missing parts of the target object, and negative clicks suppress incorrectly labeled background regions. The model updates the segmentation mask accordingly. This process continues until the user is satisfied with the segmentation quality. In interactive single-object segmentation, all interactions relate to a single target object, and the model predicts a binary mask $\mathcal{M} \in \{0,1\}^N$. In contrast, interactive multi-object segmentation extends this formulation by associating each click with a specific object ID, enabling simultaneous segmentation of multiple objects. In this setting, the model produces a segmentation mask $\mathcal{M} \in \{0,1,\ldots,M\}^N$, where M is the total number of target objects. During training, each sample begins with initial clicks placed at the center of each target object, followed by multiple iterations in which additional clicks are adaptively sampled from the top-ranked error regions. To reduce computational cost, model parameters are updated only in the final iteration. During testing, user behavior is simulated by placing initial clicks at object centers and subsequent ones on the largest error clusters.

Two-stage Efficiency-to-effectiveness Framework

Fig. 2 presents the overview of our proposed E^{2}I3D. To balance segmentation performance and resource cost, we design a two-stage efficiency-to-effectiveness framework. Unlike previous approaches (Kontogianni et al. 2022; Yue et al. 2024) that either rely on heavy architectures or compromise performance for lightweight design (Qian et al. 2022), E^{2}I3D explicitly decouples the goals of efficiency and effectiveness into two sequential yet tightly coupled stages. In the first stage, we adopt the backbone (Yue et al. 2024) pretrained on the ScanNetV2-Train (Dai et al. 2017) dataset. We then perform heterogeneous pruning and retain the resulting lightweight model. In the second stage, we introduce hierarchical click-aware attention to incorporate simulated user clicks as guidance. The entire model is subsequently fine-tuned under the supervision of ground truth. The details of each stage will be elaborated in the following sections.

Heterogeneous Pruning

To tackle the efficiency issue in the first stage, we introduce heterogeneous pruning as shown in Fig. 2, which aims to remove sub-structures in networks. Specifically, we model graphs of sub-structures and divide them into heterogeneous groups based on the heterogeneity of the network topology, where each heterogeneous group corresponds to a unique network topology. Gradient compensation is further designed to compensate for class-wise prediction bias, resulting in a more reliable evaluation of which groups to prune. Then we perform ranking and pruning within each group, which enables structure-aware channel selection.

Graph of Sub-structures A sub-structure refers to the smallest group of parameters with dependencies that can be

entirely removed without compromising the functionality or connectivity of the network. To formally describe the sub-structure, we define a pruning function $p(W_l, d, i)$ as follows:

$$\hat{w}_l = p(W_l, d, i) = \begin{cases} p(W_l, 0, i) = W[i, :, :, :], \text{if pruning } C_{\text{out}}, \\ p(W_l, 1, i) = W[:, i, :, :], \text{if pruning } C_{\text{in}}. \end{cases} \tag{1}$$

Here, $W_l \in \mathbb{R}^{C_{\text{out}} \times C_{\text{in}} \times O \times O}$ denotes the parameters of the l-th convolutional layer, where C_{out}, C_{in} and O represent the number of output channels, input channels, and kernel size, respectively. $\hat{w}_l$ corresponds to the channel marked for removal via slicing operations along the specified axis $d \in \{0, 1\}$, with i indexing the target channel in W_l. Extending this formulation from a single layer to the entire network requires considering inter-layer and intra-layer dependencies (Fang et al. 2023), as coupled parameters across layers should be pruned jointly. In particular, adjacent convolutional layers W_l and W_{l+1} are coupled through C_{out} of W_l and C_{in} of W_{l+1}, forming a minimal sub-structure.

We formally represent each sub-structure as a graph $G = (\mathcal{V}, \mathcal{E})$, where each vertex $v \in \mathcal{V}$ corresponds to a pruning operation $p(W_l, d, i)$ determined by (W_l, d, i), and each edge $e \in \mathcal{E}$ encodes a dependency. The process begins by parameterizing L convolutional layers as $\{W_1, W_2, \ldots, W_L\}$. A seed parameter W_l is randomly selected along with an axis $d \in \{0, 1\}$ to initialize the graph $G = (\mathcal{V} = \{(W_l, d, i)\}, \mathcal{E} = \emptyset)$. The graph is expanded iteratively by identifying dependent triplets $(W_l^{'}, d^{'}, i^{'})$ according to the dependency rules (Fang et al. 2023). Then the discovered dependency between (W_l, d, i) and $(W_l^{'}, d^{'}, i^{'})$ is used to update the set of vertex $\mathcal{V}$ and the set of edge $\mathcal{E}$.

$$\mathcal{V} = \mathcal{V} \cup (W_l^{'}, d^{'}, i^{'}), \quad \mathcal{E} = \mathcal{E} \cup \{((W_l, d, i), (W_l^{'}, d^{'}, i^{'}))\}. \tag{2}$$

They are updated until no new dependencies emerge. Any unassigned parameters trigger new graph constructions, recursively partitioning the network into disjoint sub-structures. Each sub-structure is represented as a graph ensuring synchronized pruning of coupled channels.

Heterogeneous Groups of Sub-structures Unlike single-layer importance measures (He, Zhang, and Sun 2017), current structured pruning methods (Fang et al. 2023; Chen et al. 2023) typically compute a holistic importance metric by aggregating individual layer scores across all parameter triplets (W_l, d, i) within a sub-structure. Although intuitive, these methods conflate parameters from diverse computational topologies, leading to biased comparisons. For instance, a large sub-structure spanning multiple embedding layers may dominate aggregated scores over a small sub-structure with only two linear layers, even if the latter contributes critically to the model. The issue arises from the inherent heterogeneity among sub-structures. Applying a unified ranking metric to such heterogeneous sub-structures fails to reflect their true relative importance. To enable fair comparisons, we divide sub-structures into heterogeneous groups based on their parameter configurations and topological logic. Two sub-structures are assigned to one heterogeneous group if they satisfy two criteria: 1) vertex label consistency, where corresponding vertices share identical layer types (*i.e.,* Conv or Linear) and pruning dimension (*i.e.,* $d = d^{'}$); 2) edge connectivity equivalence, ensuring identical dependency patterns in their computational graphs.

Gradient Compensation Unlike previous methods (Fang et al. 2023, 2024), our objective is to compensate for the gradient bias caused by performance differences across different classes, where the gradient measurement $\mathcal{T}_n$ (Wang et al. 2019, 2021) for the point n with respect to the neuron $\mathcal{N}_{Y_n}$ corresponding to the ground truth Y_n in the last output layer is calculated as follows:

$$\mathcal{T}_n = \frac{\partial \mathcal{L}}{\partial \mathcal{N}_{Y_n}} = O_{Y_n} - 1, \tag{3}$$

where O_{Y_n} is the Y_n-th sigmoid output probability of the point n. $\mathcal{L}$ is the loss function. Then the rectified weight ψ_n of the point n in Ψ is obtained by normalizing $\mathcal{T}_n$. Given a set of sub-structures $\{G_1, G_2, \ldots, G_U\}$ within a heterogeneous group $\mathcal{G}$, we independently evaluate their aggregated importance scores based on gradient compensation:

$$I^*(\mathcal{G}) = \sum_{(W_l, d, i) \in \mathcal{G}} || \frac{\partial (\Psi \mathcal{L})}{\partial g(W_l, d, i)} g(W_l, d, i) ||_2. \tag{4}$$

$I^*(\cdot)$ means the aggregated importance function.

Ranking and Pruning Heterogeneous Groups According to Eq. (4), a heterogeneous group $\mathcal{G}$ yields a U-dimensional vector $[I^*(G_1), I^*(G_2), \ldots, I^*(G_U)]$, where each element quantifies the global significance of the corresponding sub-structure. The sub-structures are then ranked based on these scores, and the least important $p\%$ are pruned. This strategy ensures reliable comparisons, as all sub-structures within a heterogeneous group share identical parameter scales, architectural configurations, and computational dependencies. By limiting comparisons to within-group sub-structures and estimating importance based on gradient compensation, the biases generated from network heterogeneity and class-wise segmentation performance are eliminated.

Hierarchical Click-aware Attention

To enhance fine-grained scene understanding in the second stage, we propose a hierarchical click-aware attention to facilitate comprehensive interactions between click queries and multi-scale scene features. Given the point cloud P of a 3D scene, our lightweight backbone extracts multi-scale scene features across H scales from different layers, denoted as $\{F_h\}_{h=1}^{H}$. $F_h \in \mathbb{R}^{N_h \times D_h}$ corresponds to the feature representation at the h-th scale, where N_h denotes the number of points at this scale and D_h is the feature dimension associated with each point. Simultaneously, user clicks serve as another input to hierarchical click-aware attention. The user clicks S at scale h are encoded into click queries Q_h, which selectively focuses on spatial regions at the h-th scale by initializing from the nearest voxel feature in F_h.

$$Q_h = \text{ClickMap}(S, F_h). \tag{5}$$

$\text{ClickMap}(S, F_h)$ means to obtain the query corresponding to S in F_h. Subsequently, both Q_h and F_h are fed into the

Method	Train→Eval	#Params	IoU@5 ↑	IoU@10 ↑	IoU@15 ↑	NoC@80 ↓	NoC@85 ↓	NoC@90 ↓
InterObject3D (Kontogianni et al. 2022)	ScanNetV2→ScanNetV2	37.9M	72.4	79.9	82.4	8.9	11.2	14.2
AGILE3D (Yue et al. 2024)		39.3M	**79.9**	**83.7**	**85.0**	**7.1**	**9.6**	**12.9**
E^{2}I3D (Ours)		40.5M	**79.9**	83.6	**85.0**	**7.1**	**9.6**	**12.9**
E^{2}I3D (Ours)		36.9M	**79.9**	83.6	**85.0**	**7.1**	**9.6**	**12.9**
E^{2}I3D (Ours)		32.8M	**79.9**	83.6	84.9	**7.1**	**9.6**	**12.9**
E^{2}I3D (Ours)		28.0M	79.7	83.5	84.9	7.2	9.7	13.0
E^{2}I3D (Ours)		24.2M	79.6	83.4	84.9	7.2	9.7	13.0
E^{2}I3D (Ours)		20.4M	79.4	83.3	84.8	7.3	9.8	13.1
E^{2}I3D (Ours)		16.4M	79.1	83.2	84.7	7.4	9.9	13.2
E^{2}I3D (Ours)		13.0M	78.8	83.1	84.6	7.5	10.0	13.2
E^{2}I3D (Ours)		8.6M	76.9	82.1	84.0	8.2	10.6	13.7
E^{2}I3D (Ours)		5.7M	72.9	80.2	82.8	9.4	11.8	14.7
InterObject3D (Kontogianni et al. 2022)	ScanNetV2→S3DIS-A5	37.9M	72.4	83.6	88.3	6.8	8.4	11.0
AGILE3D (Yue et al. 2024)		39.3M	83.5	88.2	89.5	4.8	6.4	9.5
E^{2}I3D (Ours)		40.5M	**84.2**	88.6	**89.9**	**4.4**	**6.1**	**9.1**
E^{2}I3D (Ours)		36.9M	83.7	88.5	**89.9**	4.5	6.2	**9.1**
E^{2}I3D (Ours)		32.8M	**84.2**	**88.7**	89.8	4.5	6.2	**9.1**
E^{2}I3D (Ours)		28.0M	84.0	88.5	89.8	4.5	6.2	**9.1**
E^{2}I3D (Ours)		24.2M	84.0	**88.7**	**89.9**	4.5	**6.1**	**9.1**
E^{2}I3D (Ours)		20.4M	83.6	88.3	89.7	4.6	6.4	9.2
E^{2}I3D (Ours)		16.4M	83.7	88.4	89.8	4.6	6.3	9.2
E^{2}I3D (Ours)		13.0M	83.4	88.3	89.7	4.7	6.5	9.2
E^{2}I3D (Ours)		8.6M	82.1	87.9	89.4	5.1	6.8	9.7
E^{2}I3D (Ours)		5.7M	77.7	85.8	88.4	6.3	8.5	11.4
InterObject3D (Kontogianni et al. 2022)	ScanNetV2→KITTI-360	37.9M	14.3	26.3	35.0	19.1	19.4	19.7
AGILE3D (Yue et al. 2024)		39.3M	44.4	49.6	54.9	14.2	15.5	16.8
E^{2}I3D (Ours)		40.5M	**49.6**	52.2	54.9	13.4	14.7	16.1
E^{2}I3D (Ours)		36.9M	48.7	51.7	55.2	13.3	14.5	15.9
E^{2}I3D (Ours)		32.8M	47.4	50.6	53.5	14.1	15.3	16.6
E^{2}I3D (Ours)		28.0M	47.2	50.8	53.9	14.0	15.1	16.4
E^{2}I3D (Ours)		24.2M	46.0	48.5	50.7	14.2	15.5	16.8
E^{2}I3D (Ours)		20.4M	47.5	51.7	54.9	13.5	14.8	16.2
E^{2}I3D (Ours)		16.4M	46.7	51.6	54.9	13.9	15.0	16.1
E^{2}I3D (Ours)		13.0M	47.8	**53.2**	56.6	13.7	14.8	16.0
E^{2}I3D (Ours)		8.6M	46.0	52.3	**57.5**	14.1	15.2	16.4
E^{2}I3D (Ours)		5.7M	49.0	50.3	53.4	**11.5**	**12.9**	**14.7**

Table 1: Quantitative comparison results on interactive single-object segmentation.

corresponding click-aware attention module (Yue et al. 2024), producing a refined $\widetilde{F}_h$. The multi-scale refined features are then aggregated to obtain a unified representation F_{agg}:

$$F_{agg} = \sum_{h=1}^{H} \widetilde{F}_h = \sum_{h=1}^{H} \text{ClickAttn}(F_h, Q_h). \quad (6)$$

$\text{ClickAttn}(F_h, Q_h)$ refers to the click-aware attention module that takes F_h and Q_h as input. The aggregated feature F_{agg} is fed into a query fusion module (Yue et al. 2024) to predict the segmentation mask $\mathcal{M}$, supervised by the ground truth.

Experiments

Experimental Setup

Datasets The proposed E^{2}I3D model is trained on ScanNetV2-Train (Dai et al. 2017) and evaluated on ScanNetV2-Val (Dai et al. 2017), S3DIS (Armeni et al. 2016) and KITTI-360 (Behley et al. 2019).

Implementation Details We extract feature maps of sizes $N_1 \times 96$ and $N_2 \times 96$ from the intermediate and final layers (i.e., $H = 2$). Other settings follow AGILE3D (Yue et al. 2024). We use four NVIDIA RTX 4090 GPUs.

Evaluation Metrics In the single-object setting, IoU@k ↑ measures the mean IoU after k clicks (≤ 20 per object), and NoC@q% ↓ denotes the average clicks required to reach a q% IoU. For multi-object evaluation, $\overline{\text{IoU}}$@k ↑ and $\overline{\text{NoC}}$@q% ↓ represent their averaged counterparts across all objects. A total budget of $M \times 20$ clicks is shared among M objects instead of fixing clicks per object.

Baseline Our method is evaluated against AGILE3D (Yue et al. 2024) and InterObject3D (Kontogianni et al. 2022).

Comparison on Single-object Segmentation

In Tab. 1, all models are evaluated on multiple datasets (Dai et al. 2017; Armeni et al. 2016; Behley et al. 2019) to assess domain generalization capabilities. On the in-distribution

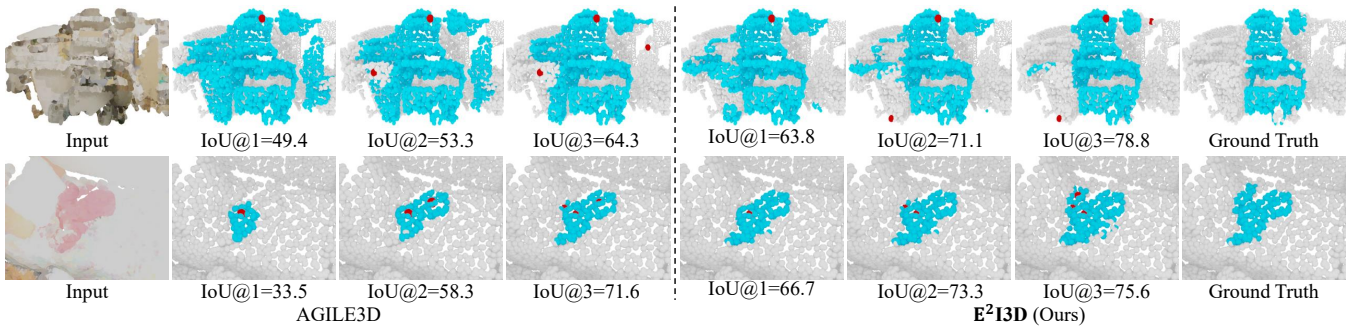

Figure 3: Visualization of interactive single-object segmentation on the ScanNetV2 dataset.

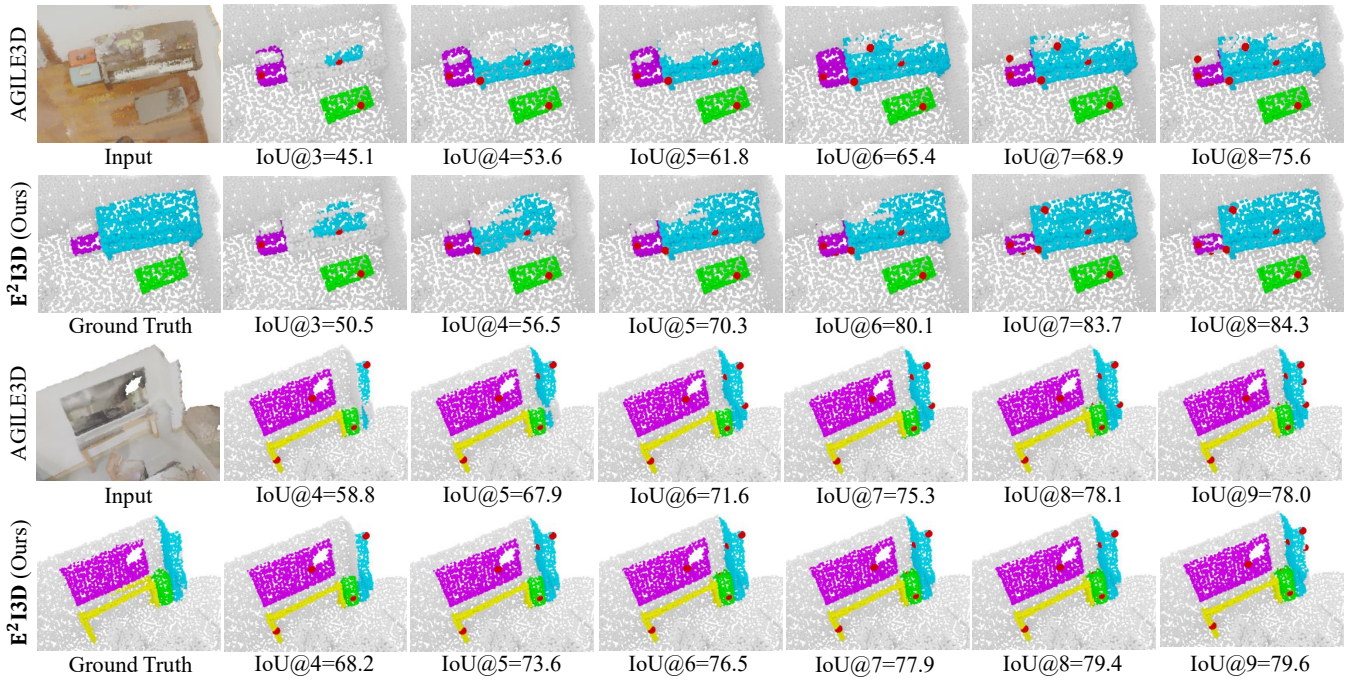

Figure 4: Visualization of interactive multi-object segmentation on the ScanNetV2 dataset.

ScanNetV2 dataset (Dai et al. 2017), our method achieves competitive performance over existing baselines despite a significant reduction in parameters. To evaluate robustness under domain shifts, we further test the models on the S3DIS dataset (Armeni et al. 2016) (indoor scenes captured with different sensors). Our method consistently demonstrates strong generalization, achieving a 0.2% IoU improvement using 15 user clicks, with a parameter reduction from 39.3 M to 13.0M. In particular, our method exhibits exceptional adaptability to the challenging KITTI-360 dataset (Behley et al. 2019) (outdoor LiDAR scenes). For example, when targeting an IoU of 90%, our method achieves this with 14.7 clicks on average using just 5.7M parameters, significantly outperforming AGILE3D (Yue et al. 2024). The consistent performance gap across all metrics and environments confirms the ability of our method to handle distribution shifts. The visualization results show that our method achieves gains with the same (see Fig. 3) clicks, highlighting its efficiency and effectiveness.

Comparison on Multi-object Segmentation

Our evaluation demonstrates the advantages of our method in interactive multi-object segmentation, as quantitatively validated in Tab. 2 and Fig. 4. On the KITTI-360 (Behley et al. 2019) dataset, our method outperforms AGILE3D (Yue et al. 2024) by an average of 3.0%, 1.9% and 1.4% IoU under 5, 10 and 15 user clicks, respectively, with a drastic parameter reduction from 39.3 M to 5.7M, demonstrating its robustness on interactive multi-object segmentation.

Efficiency Comparison

To evaluate inference efficiency, we measure the time taken after an average of 5, 10, and 15 user interactions per object across different scenes in Tab. 3. Our model not only maintains less memory footprint to that of existing methods, but also achieves faster inference. This is due to our carefully designed heterogeneous pruning.

Method	Train→Eval	#Params	$\overline{\text{IoU}}@\bar{5}\uparrow$	$\overline{\text{IoU}}@\bar{10}\uparrow$	$\overline{\text{IoU}}@\bar{15}\uparrow$	$\overline{\text{NoC}}@\bar{80}\downarrow$	$\overline{\text{NoC}}@\bar{85}\downarrow$	$\overline{\text{NoC}}@\bar{90}\downarrow$
InterObject3D (Kontogianni et al. 2022)	ScanNetV2→ScanNetV2	37.9M	75.1	80.3	81.6	10.2	13.5	16.6
AGILE3D (Yue et al. 2024)		39.3M	82.3	**85.0**	**86.0**	6.3	**10.0**	**14.4**
E^{2}I3D (Ours)		40.5M	**82.5**	**85.0**	**86.0**	**6.0**	10.3	**14.4**
E^{2}I3D (Ours)		36.9M	82.3	84.9	**86.0**	6.2	10.1	14.5
E^{2}I3D (Ours)		32.8M	82.3	84.9	85.9	6.2	10.3	**14.4**
E^{2}I3D (Ours)		28.0M	82.2	84.8	86.0	6.2	10.3	14.5
E^{2}I3D (Ours)		24.2M	82.1	84.7	85.8	6.3	10.5	14.5
E^{2}I3D (Ours)		20.4M	81.9	84.5	85.7	6.5	10.6	14.6
E^{2}I3D (Ours)		16.4M	81.7	84.5	85.7	6.6	10.7	14.6
E^{2}I3D (Ours)		13.0M	81.3	84.2	85.5	6.8	10.8	14.8
E^{2}I3D (Ours)		8.6M	80.0	83.6	84.9	7.4	11.4	15.0
E^{2}I3D (Ours)		5.7M	77.3	82.1	83.9	8.9	12.5	15.8
InterObject3D (Kontogianni et al. 2022)	ScanNetV2→S3DIS-A5	37.9M	76.9	85.0	87.3	6.8	8.8	13.5
AGILE3D (Yue et al. 2024)		39.3M	86.3	88.3	90.3	3.4	5.7	9.6
E^{2}I3D (Ours)		40.5M	88.7	90.8	91.5	**2.6**	**4.3**	**8.0**
E^{2}I3D (Ours)		36.9M	**89.0**	**91.1**	**91.6**	2.7	**4.3**	**8.0**
E^{2}I3D (Ours)		32.8M	88.7	90.9	91.4	2.8	**4.3**	8.3
E^{2}I3D (Ours)		28.0M	88.8	90.7	91.3	3.0	4.4	8.6
E^{2}I3D (Ours)		24.2M	88.7	90.9	**91.6**	**2.6**	4.4	8.5
E^{2}I3D (Ours)		20.4M	88.6	90.8	91.5	2.8	4.5	8.4
E^{2}I3D (Ours)		16.4M	88.9	91.0	91.5	2.8	**4.3**	**8.0**
E^{2}I3D (Ours)		13.0M	88.3	90.7	91.4	2.9	4.6	8.6
E^{2}I3D (Ours)		8.6M	86.9	90.3	91.2	3.3	5.3	9.2
E^{2}I3D (Ours)		5.7M	83.5	88.4	90.1	4.7	6.8	11.4
InterObject3D (Kontogianni et al. 2022)	ScanNetV2→KITTI-360	37.9M	10.5	22.1	31.0	19.8	19.8	19.9
AGILE3D (Yue et al. 2024)		39.3M	40.5	44.3	48.2	17.4	18.3	18.8
E^{2}I3D (Ours)		40.5M	44.4	**47.6**	49.5	17.1	18.1	18.7
E^{2}I3D (Ours)		36.9M	43.6	46.4	49.5	17.0	**17.7**	18.6
E^{2}I3D (Ours)		32.8M	44.0	46.8	49.3	17.3	18.1	18.6
E^{2}I3D (Ours)		28.0M	43.5	46.8	49.0	17.2	17.9	18.6
E^{2}I3D (Ours)		24.2M	41.3	43.0	45.1	17.6	18.2	18.7
E^{2}I3D (Ours)		20.4M	**44.5**	47.3	49.5	**16.6**	**17.7**	18.4
E^{2}I3D (Ours)		16.4M	42.7	45.9	48.5	17.0	17.9	18.6
E^{2}I3D (Ours)		13.0M	43.0	45.4	48.2	17.2	18.1	18.7
E^{2}I3D (Ours)		8.6M	42.8	47.4	50.7	17.3	18.1	18.5
E^{2}I3D (Ours)		5.7M	43.5	46.2	**49.6**	15.2	16.8	**17.9**

Table 2: Quantitative comparison on interactive multi-object segmentation.

Method	#Params	t@$\bar{5}$	t@$\bar{10}$	t@$\bar{15}$
InterObject3D	37.9M	1.2	2.4	3.6
AGILE3D	39.3M	0.5	1.0	1.6
E^{2}I3D (Ours)	5.7M	0.1	0.2	0.3

Table 3: Efficiency comparison.

Method	IoU@5 ↑	IoU@10 ↑	NoC@80 ↓	NoC@85 ↓
E^{2}I3D-w/o HCA	46.8	50.0	13.8	15.1
E^{2}I3D (Ours)	48.7	51.7	13.3	14.5

Table 4: Ablation Study.

Ablation Study

Tabs. 1 and 2 show the ablation study of heterogeneous pruning (HP). E^{2}I3D with 40.5M parameters represents our model without HP, which outperforms AGILE3D (Yue et al. 2024) in almost every scenario. In addition, although our model is pruned to different degrees, its performance remains stable, which proves the rationality of our heterogeneous pruning. The ablation study of hierarchical click-aware attention (HCA) is shown in Tab. 4. HCA improves IoU by 1.9% using 5 user clicks, showing the effectiveness of enhancing interactions between clicks and multi-scale features.

Conclusion

This work addresses the critical trade-off between efficiency and effectiveness in interactive 3D segmentation by introducing the E^{2}I3D model. Specifically, we propose a two-stage efficiency-to-effectiveness framework. For efficiency in the first stage, we design the heterogeneous pruning, which selectively removes redundant sub-structures based on network heterogeneity and gradient compensation. For effectiveness in the second stage, we develop the hierarchical click-aware attention. It strengthens the interaction between user clicks and multi-scale scene features. Experiments demonstrate the efficiency and effectiveness of our E^{2}I3D.

Acknowledgments

This work is supported by National Key R&D Program of China (2023YFB4704800), Guangdong S&T Program (2025B1111130001), National Nature Science Foundation of China under Grant (62225310, 62127807) and the Fundamental Research Funds for the Central Universities (2024ZYGXZR024).

References

Armeni, I.; Sener, O.; Zamir, A. R.; Jiang, H.; Brilakis, I.; Fischer, M.; and Savarese, S. 2016. 3d semantic parsing of large-scale indoor spaces. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 1534–1543.

Behley, J.; Garbade, M.; Milioto, A.; Quenzel, J.; Behnke, S.; Stachniss, C.; Gall, J.; and Semantickitti. 2019. A dataset for semantic scene understanding of lidar sequences. In *Int. Conf. Comput. Vis.*, 9297–9307.

Chen, T.; Liang, L.; Ding, T.; Zhu, Z.; and Zharkov, I. 2023. Otov2: Automatic, generic, user-friendly. *arXiv preprint arXiv:2303.06862*.

Cong, W.; Cong, Y.; Dong, J.; Sun, G.; and Ding, H. 2023. Gradient-Semantic Compensation for Incremental Semantic Segmentation. *IEEE Trans. Multimedia*.

Cong, W.; Cong, Y.; Liu, Y.; and Sun, G. 2025. Cs^2K: Class-Specific and Class-Shared Knowledge Guidance for Incremental Semantic Segmentation. In *Eur. Conf. Comput. Vis.*, 244–261.

Dai, A.; Chang, A. X.; Savva, M.; Halber, M.; Funkhouser, T.; and Nießner, M. 2017. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 5828–5839.

Ding, H.; Cohen, S.; Price, B.; and Jiang, X. 2020. Phraseclick: toward achieving flexible interactive segmentation by phrase and click. In *Eur. Conf. Comput. Vis.*, 417–435.

Fang, G.; Ma, X.; Mi, M. B.; and Wang, X. 2024. Isomorphic pruning for vision models. In *Eur. Conf. Comput. Vis.*, 232–250. Springer.

Fang, G.; Ma, X.; Song, M.; Mi, M. B.; and Wang, X. 2023. Depgraph: Towards any structural pruning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 16091–16101.

Guo, Y.; Yao, A.; and Chen, Y. 2016. Dynamic network surgery for efficient dnns. In *Adv. Neural Inform. Process. Syst.*, volume 29.

Han, C.; Yu, X.; Xie, Y.; Liu, Y.; Mao, S.; Zhou, S.; Xiong, R.; and Wang, Y. 2024. Scale disparity of instances in interactive point cloud segmentation. In *Int. Conf. Intell. Robots Syst.*, 2660–2667. IEEE.

Han, S.; Mao, H.; and Dally, W. J. 2016. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *Int. Conf. Learn. Represent.*

Han, S.; Pool, J.; Tran, J.; and Dally, W. 2015. Learning both weights and connections for efficient neural network. *Adv. Neural Inform. Process. Syst.*, 28.

He, Y.; Kang, G.; Dong, X.; Fu, Y.; and Yang, Y. 2018. Soft filter pruning for accelerating deep convolutional neural networks. In *IJCAI*.

He, Y.; Liu, P.; Wang, Z.; Hu, Z.; and Yang, Y. 2019. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 4340–4349.

He, Y.; Zhang, X.; and Sun, J. 2017. Channel pruning for accelerating very deep neural networks. In *Int. Conf. Comput. Vis.*, 1389–1397.

Kim, J.; Park, S.; and Kwak, N. 2018. Paraphrasing complex network: Network compression via factor transfer. *Adv. Neural Inform. Process. Syst.*, 31.

Kolodiazhnyi, M.; Vorontsova, A.; Konushin, A.; and Rukhovich, D. 2024. Oneformer3d: One transformer for unified point cloud segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 20943–20953.

Kontogianni, T.; Celikkan, E.; Tang, S.; and Schindler, K. 2022. Interactive Object Segmentation in 3D Point Clouds. *Int. Conf. Robot. Autom.*

Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; and Graf, H. 2017. Pruning Filters for Efficient ConvNets. In *Int. Conf. Learn. Represent.*

Lin, M.; Ji, R.; Wang, Y.; Zhang, Y.; Zhang, B.; Tian, Y.; and Shao, L. 2020. Hrank: Filter pruning using high-rank feature map. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 1529–1538.

Ling, H.; Gao, J.; Kar, A.; Chen, W.; and Fidler, S. 2019. Fast interactive object annotation with curve-gcn. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 5257–5266.

Liu, C.; and Wu, H. 2019. Channel pruning based on mean gradient for accelerating convolutional neural networks. *Signal Processing*, 156: 84–91.

Liu, J.; Niu, L.; Yuan, Z.; Yang, D.; Wang, X.; and Liu, W. 2023. PD-Quant: Post-Training Quantization Based on Prediction Difference Metric. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 24427–24437.

Luo, J.-H.; Wu, J.; and Lin, W. 2017. Thinet: A filter level pruning method for deep neural network compression. In *Int. Conf. Comput. Vis.*, 5058–5066.

Qian, G.; Li, Y.; Peng, H.; Mai, J.; Hammoud, H.; Elhoseiny, M.; and Ghanem, B. 2022. Pointnext: Revisiting pointnet++ with improved training and scaling strategies. *Adv. Neural Inform. Process. Syst.*, 35: 23192–23204.

Roh, W.; Jung, H.; Nam, G.; Yeom, J.; Park, H.; Yoon, S. H.; and Kim, S. 2024. Edge-Aware 3D Instance Segmentation Network with Intelligent Semantic Prior. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 20644–20653.

Sainath, T. N.; Kingsbury, B.; Sindhwani, V.; Arisoy, E.; and Ramabhadran, B. 2013. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *IEEE international conference on acoustics, speech and signal processing*, 6655–6659.

Shang, Y.; Yuan, Z.; Xie, B.; Wu, B.; and Yan, Y. 2023. Post-Training Quantization on Diffusion Models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 1972–1981.

Shen, T.; Gao, J.; Kar, A.; and Fidler, S. 2020. Interactive annotation of 3D object geometry using 2D scribbles. In *Eur. Conf. Comput. Vis.*, 751–767.

Shin, S.; Zhou, K.; Vankadari, M.; Markham, A.; and Trigoni, N. 2024. Spherical mask: Coarse-to-fine 3d point cloud instance segmentation with spherical representation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 4060–4069.

Sofiiuk, K.; Petrov, I.; Barinova, O.; and Konushin, A. 2020. f-brs: Rethinking backpropagating refinement for interactive segmentation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 8623–8632.

Sun, W.; Luo, Z.; Chen, Y.; Li, H.; Junior, J. M.; Gonéalves, W. N.; and Li, J. 2023. A click-based interactive segmentation network for point clouds. *IEEE Trans. Geosci. Remote Sens.*, 61: 1–12.

Valentin, J.; Vineet, V.; Cheng, M.-M.; Kim, D.; Shotton, J.; Kohli, P.; Nießner, M.; Criminisi, A.; Izadi, S.; and Torr, P. 2015. Semanticpaint: Interactive 3d labeling and learning at your fingertips. *ACM Trans. Graph.*, 34(5): 1–17.

Wang, L.; Xu, S.; Wang, X.; and Zhu, Q. 2019. Eavesdrop the composition proportion of training labels in federated learning. *arXiv preprint arXiv:1910.06044*.

Wang, L.; Xu, S.; Wang, X.; and Zhu, Q. 2021. Addressing class imbalance in federated learning. In *Proc. AAAI Conf. Artif. Intell.*, volume 35, 10165–10173.

Wu, X.; Gao, S.; Zhang, Z.; Li, Z.; Bao, R.; Zhang, Y.; Wang, X.; and Huang, H. 2024. Auto-train-once: Controller network guided automatic network pruning from scratch. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 16163–16173.

Xu, C.; and McAuley, J. 2023. A survey on model compression and acceleration for pretrained language models. In *Proc. AAAI Conf. Artif. Intell.*, volume 37, 10566–10575.

Xu, N.; Price, B.; Cohen, S.; Yang, J.; and Huang, T. S. 2016. Deep interactive object selection. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 373–381.

Yue, Y.; Mahadevan, S.; Schult, J.; Engelmann, F.; Leibe, B.; Schindler, K.; and Kontogianni, T. 2024. AGILE3D: Attention Guided Interactive Multi-object 3D Segmentation. In *Int. Conf. Learn. Represent.*

Zhi, S.; Sucar, E.; Mouton, A.; Haughton, I.; Laidlow, T.; and Davison, A. J. 2022. ilabel: Revealing objects in neural fields. *IEEE Robot. Autom. Lett.*, 8(2): 832–839.

Zhu, X.; Gong, S.; et al. 2018. Knowledge distillation by on-the-fly native ensemble. *Adv. Neural Inform. Process. Syst.*, 31.