ClickEnhance: Efficient 3D Interactive Segmentation with Click-Specific Encoder and Contrastive Learning

Yueyang Wen, Yiwen Hou, Shuheng Zhang, Feng Wu

***Abstract*—In interactive point cloud segmentation, users can achieve higher accuracy object masks than in instance segmentation by performing limited positive and/or negative clicks on the objects of interest in the scene. Existing methods often employ sparse click representations, leading the model to focus more on local detail features around the click points and failing to fully exploit the guidance information provided by each click, thus impacting the click effectiveness. We utilize a dense representation that reflects spatial distance relationships, known as the distance map, as the click channel to tackle the sparsity problem of click representation in current approaches. Based on the distance map, we introduce ClickEnhance, which is designed to maximize the guiding impact of each click. The proposed method encompasses the design of a click-specific encoder and the utilization of contrastive learning. The Click-Specific Encoder ensures that the network can adequately consider the influence of individual clicks during the feature encoding phase. Contrastive learning, on the other hand, reduces the feature distance between the click points and the target object, thus simplifying the subsequent segmentation process. Experimental results demonstrate that the ClickEnhance method markedly improves segmentation performance across multiple datasets, exhibiting superior generalization capabilities on challenging datasets compared to the current state-of-the-art methods. This allows for the generation of high-precision object-level masks with fewer interactions, indicating great potential for practical applications.**

***Index Terms*—Interactive point cloud segmentation, user interaction, contrastive learning, click enhance, distance map.**

I. Introduction

SINCE the introduction of PointNet [1], point cloud deep learning technology has developed rapidly. In the field of point cloud segmentation, point cloud semantic segmentation [2], [3] focuses on understanding different object categories in a scene but does not distinguish between different instances of the same category. In contrast, point cloud instance segmentation [4]–[6] not only identifies the category of each point but also further distinguishes different individual instances. Interactive segmentation of the point cloud [7], [8] is similar to instance segmentation in that it aims to obtain masks of objects at the instance level, but it does not rely on specific semantic information. When a model encounters categories not present in the training set, the performance of point cloud instance segmentation is typically poor because instance segmentation is highly dependent on semantic information. The advantage of interactive segmentation lies in its ability to overcome the limitations of semantic categories: an interactively trained model on one dataset can still exhibit good generalization capabilities when faced with another unseen dataset, which is difficult for current models used in point cloud semantic segmentation, instance segmentation, or panoptic segmentation. Additionally, through iterative interaction, interactive segmentation can generate high-precision instance-level object masks. One direct application of this characteristic is point cloud annotation. With a well-trained interactive segmentation model, we can annotate high-precision object labels with just a few clicks. This undoubtedly saves a significant amount of time and cost in 3D annotation, as manual annotation in the 3D domain is both time-consuming and expensive.

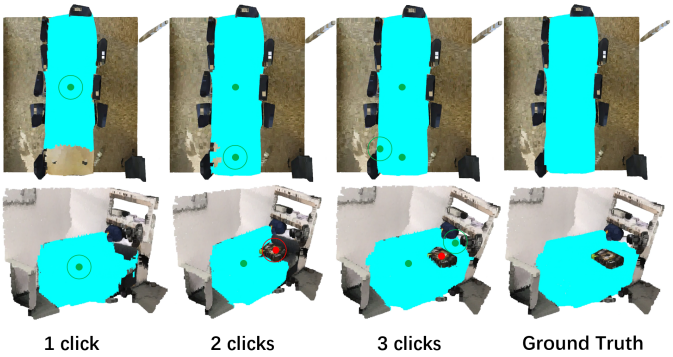

Fig. 1. Interactive object segmentation in point clouds (red/green dots are negative/position clicks and the current click is circled).

Interactive segmentation techniques originated in the image domain [9]–[12] and have been introduced to the 3D point cloud domain in recent years. Currently, this technology is still in its early stages of development, leaving ample room for improvement and expansion. In point cloud interactive segmentation [7], [8], clicking has become the primary means of interaction, as performing similar interactions in 3D space, such as drawing bounding boxes [10], [12]–[14], is much more time-consuming and complex compared to 2D images. In point cloud scenarios, we typically use positive clicks to specify the object of interest and negative clicks to correct misclassified segments, as shown in Fig. 1.

Current research on interactive point cloud segmentation

This work was supported in part by the Major Research Plan of the National Natural Science Foundation of China under Grant 92048301 and in part by the Anhui Provincial Natural Science Foundation under Grant 2208085MF172.

All authors are with School of Computer Science and Technology, University of Science and Technology of China, Hefei, Anhui, China. (E- mail: {wenyueyang, houyiwen, zsh123456}@mail.ustc.edu.cn, wufeng02@ustc.edu.cn).

Feng Wu is the corresponding author.

primarily focuses on two methods: InterObject3D [7] and AGILE3D [8]. InterObject3D [7] extends image-based interactive segmentation techniques to the domain of point cloud interactive segmentation, while AGILE3D [8] achieves superior performance by introducing an attention mechanism [15]. However, we identify two main limitations in these methods: the sparsity of click representation, which leads the model to overly focus on local features around the clicked points, thereby reducing its generalization capability for scenes with significant structural differences; and the insufficient utilization of click guidance information, which limits the full performance potential of each click.

We propose *ClickEnhance*, a method designed to maximize the effectiveness of each click by fully utilizing the information provided by each click. To address the issue of poor generalization performance caused by sparse click representation, we have chosen the distance map as a dense form of click representation. Building on this foundation, we have developed the *Click-Specific Encoder* module and designed a *click loss* using fully supervised contrastive learning [16], [17]. The *Click-Specific Encoder* enables the network to consider the influence of individual clicks, while the *click loss* designed through fully supervised contrastive learning facilitates the model's ability to distinguish between the characteristics within and outside the instance relative to the clicked points.

In the experimental section, we first compare our method with point cloud instance segmentation methods. The results show that our *ClickEnhance* method can surpass the current state-of-the-art(SOTA) instance segmentation methods with only a few clicks. Subsequently, we conduct a comprehensive comparison with the current SOTA interactive segmentation methods. The experimental results demonstrate that our method outperforms the SOTA interactive segmentation methods on multiple datasets, and shows significant advantages on challenging outdoor datasets. Finally, we perform an ablation study and provide a detailed analysis of some performance curves.

The main contributions of this work are summarized as follows:

- We adopt a dense input representation (distance map) to mitigate the sparsity issues present in existing methods.
- We have designed a Click-Specific Encoder module to generate a dedicated input channel for the current click, enabling the network to simultaneously consider information from both distance maps, thereby enhancing the effectiveness of the current click.
- We propose a click loss based on contrastive learning to minimize the feature distance between click points and foreground regions, while maximizing the distance from background regions.
- Experiments demonstrate that our proposed method achieves SOTA performance across multiple datasets, significantly outperforming current approaches on outdoor datasets.

II. Related Work

Current traditional segmentation tasks in the point cloud domain, such as semantic segmentation [2], [3], instance segmentation [4]–[6], and panoptic segmentation [18], have achieved excellent results on specialized datasets. However, even the most advanced methods are still far from reaching their upper limits. For example, in point cloud instance segmentation, the current SOTA methods, including Spherical Mask [19], QueryFormer [20], PBNet [21], and OneFromer3D [5], achieve an average precision (AP) score of around 60 on the ScanNetV2 [22] dataset. This is because these methods perform a one-shot segmentation, and for interactive segmentation, just a few clicks can easily outperform the most advanced instance segmentation methods. Additionally, these instance segmentation methods face limitations in semantic information and generalize poorly to unknown datasets, which are precisely the strengths of interactive segmentation methods.

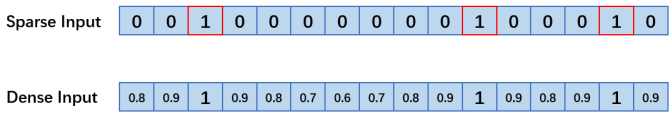

Fig. 2. This figure compares sparse click input representation and dense click input representation. The positions with a value of 1 indicate click locations. Sparse click representation can cause the network to overly focus on the positions with a value of 1, which may affect the model's generalization ability. In contrast, dense click representation reduces the overemphasis on specific click positions, thereby improving the model's generalization ability.

Currently, 2D interactive segmentation has seen widespread application and development, primarily focusing on two main interaction modes: bounding boxes [10], [12], [13] and clicks [9], [23], [24]. Some existing works propose a two-stage approach, such as FocalClick [11] and FocusCut [25], which first obtain a coarse segmentation in the first stage and then refine it in the second stage to achieve a focused effect. Other works [24] use reinforcement learning to automatically predict the next click location, achieving the effect of multiple clicks. A method similar to our work is FCA-Net [26], which considers that the first click is usually at the center of the object. Current image-based interactive models can achieve an Intersection over Union (IoU) of over 80 with just one click. Therefore, FCA-Net [26] fixes the weight of the first click by converting the distance map into a Gaussian map and amplifying the σ value during the Gaussian transformation to enhance the influence of the first click. However, this method heavily relies on the adjustment of the σ value. In contrast, we design a Click-Specific Encoder module to provide the model with the influence of each individual click, not limited to the first click, and our method is free from the constraints of the hyperparameter σ, making it simpler and more efficient.

Point-SAM [27] is a recent prompt-based method that aims to replicate the success of SAM [28] in the 3D domain. This approach integrates multiple datasets to achieve fine-grained instance-level and part-level segmentation. Unlike the current standard practice for interactive segmentation of point clouds—where models are trained on a single dataset but tested on multiple datasets to demonstrate generalization capabilities—although Point-SAM [27] has achieved good results, it utilizes multiple datasets for training and has a larger parameter count, placing it in the category of large models rather than representing an innovation in interactive methods.

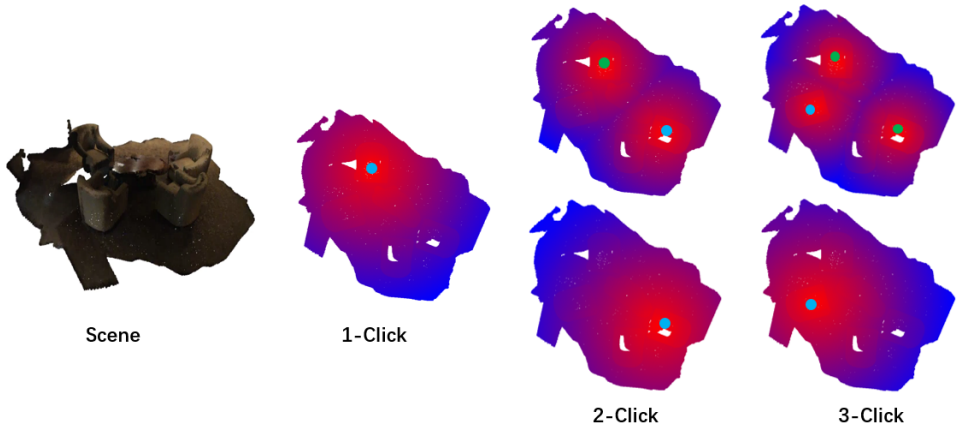

Fig. 3. This is a visualization of the click influence in a scene, with the distance map color-mapped such that red indicates strong influence and blue indicates weak influence. The green points represent historical clicks, and the blue points represent the current click. The top layer shows the overall click influence map, while the bottom layer shows the influence map for a single click.

Current research on interactive segmentation in the field of point clouds primarily includes InterObject3D [7] and AGILE3D [8], both of which utilize the U-Net [29] architecture as their backbone networks. The introduction of InterObject3D [7] marked the extension of interactive segmentation tasks from the image domain to the point cloud domain. This method builds upon traditional semantic segmentation backbone networks by additionally incorporating positive and negative click channels. In contrast, AGILE3D [8] leverages an attention mechanism, associating the click points with queries in the attention mechanism and enabling adaptive interactions through self-attention and cross-attention, allowing the click points to interact with other points in the point cloud space. Additionally, AGILE3D extends interactive point cloud segmentation from single-object to multi-object scenarios. Although these methods improve the interactive representation or expand functionality, they cannot avoid the sparsity issue in their interactive representations and do not explore the information inherent in the click points. Our method changes the sparse representation used in previous works and introduces, for the first time in point cloud segmentation, a dense input representation using a distance map. We also design a *Click-Specific Encoder* and *Click Loss* to enhance the effectiveness of clicks. Compared to previous methods, our approach achieves better results with fewer interactions.

III. Motivation

We briefly review the literature on interactive point cloud segmentation and identify two main deficiencies:

A. Sparse Click Representation

Both methods use a sparse representation of click points. As shown in Fig. 2, this sparsity causes the model to learn overly structured local features, which hinders its ability to generalize to datasets with significant structural variations, failing to fully leverage the advantages of interactive methods. In contrast, dense click representation, as also illustrated in the figure, helps the model avoid over-focusing on specific click positions, leading to better generalization.

The click representation in InterObject3D [7] is sparse because it establishes positive and negative click channels of the same size as the number of points in the point cloud, initially set to zero. Only when a user clicks at a specific location, a cube-sized region is generated at that location, and the corresponding click channel values for the points within this region are set to 1. The number of 1s is far fewer than the number of 0s, resulting in a sparse representation. This causes the model to overly focus on local features around the clicked points, neglecting more general features.

AGILE3D [8] also faces similar sparsity issues. It extracts the user's click points separately and interacts with the entire point cloud space using attention mechanisms. However, compared to the total number of points in the point cloud, this representation is still sparse, causing the model to learn overly structured features around the click points. Consequently, the performance of this method deteriorates when dealing with datasets that have significant structural variations.

B. Insufficient Utilization of Click Guidance

These methods do not fully utilize the guidance information provided by click points. InterObject3D [7] uses simple click channels to reflect click position information, while AGILE3D [8], although using attention mechanisms to allow the model to learn adaptively, does not deeply explore the inherent meaning of the clicks, leading to underutilization of the guiding effect of each click.

These issues motivate us to the following considerations:

- How can we find a denser input representation for point clouds that prevents the model from over-focusing on local details around the click points?
- How can we fully utilize the guidance information provided by each click to maximize the performance?

IV. Method

The overall architecture of our network is shown in Fig. 5. The green arrows represent the backbone, and the blue arrows represent the additional branch module we added for individual click. Together, they form the *Click-Specific Encoder* module.

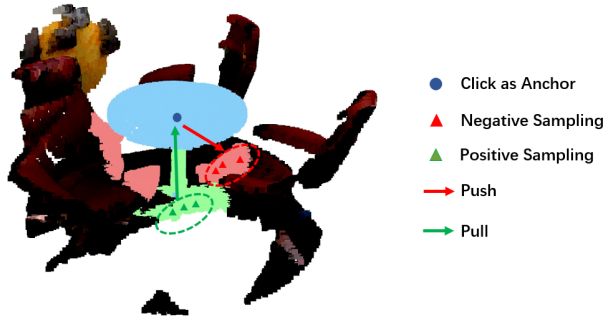

Fig. 4. This figure demonstrates the use of contrastive learning for segmentation. The goal is to segment the table in the scene by clicking on the tabletop (blue point). Currently, the blue and red regions are predicted as the foreground, but we want only the blue and green regions to be the foreground. We aim to use contrastive learning to reduce the feature distance between the click point (blue) and the green region, while increasing the distance from the red region.

In our network architecture, the input is bifurcated into two components. The first input comprises a distance map generated from the original point cloud alongside historical click information, whereas the second input consists of a distance map derived from the point cloud in conjunction with the current click information only. These inputs are then processed through our specially designed Click-Specific Encoder module to extract richer feature representations.

In the decoder segment of our network, dual outputs are produced. One output is designated for segmentation loss calculation, facilitating the delineation between foreground and background areas. Concurrently, another output mechanism is activated exclusively during the training phase. This involves an innovative adaptation of the triplet loss concept into what we term as click loss. Through this mechanism, challenging points within regions misidentified by the model predictions are sampled. Consequently, this process adjusts the feature distances among click points and positive/negative samples, either by drawing them closer or pushing them apart. Such a strategy significantly enhances the neural network's capability to distinguish between foreground and background features effectively.

A. Dense Input Representation

Our research suggests that a distance map, a dense representation, is more suitable. Since point clouds inherently possess spatial relationships, using a distance map as the click representation can fully reflect the distance information of all points in the point cloud space relative to the click location. The magnitude of the distance values reflects the proximity, and this representation is dense, avoiding the issue of having many zeros.

1) Input Representation: As shown in Fig. 5, our model has two inputs. The blue arrow represents the additional input for individual click, which forms a branch of the network, while the green arrow represents the main pathway. Our input structure is similar to that of InterObject3D [7] , where the point cloud and click channels are concatenated.

Consider a 3D scene $\mathbf{P} \in \mathbb{R}^{N \times C}$, where N represents the number of points in the point cloud, and C represents the feature dimension of each point, typically including the xyz position features and rgb color features. We augment this representation by adding two distance maps $\mathbf{D}_{\text{pos}} \in \mathbb{R}^{N \times 1}$ and $\mathbf{D}_{\text{neg}} \in \mathbb{R}^{N \times 1}$, where $\mathbf{D}_{\text{pos},i}$ represents the distance of point $\mathbf{P}_i$ to the nearest positive click, and $\mathbf{D}_{\text{neg},i}$ represents the distance of point $\mathbf{P}_i$ to the nearest negative click.

Therefore, the input representation for the main network module is $\mathbb{R}^{N \times (C+2)}$, where C is 6-dimensional (3 for xyz and 3 for rgb), and the additional 2 channels represent the distance maps for positive and negative clicks.

For the branch module, we exclude the rgb channels and retain only the current click point information in the click channel.

2) Click Distance Maps for Point Clouds: We adopt a method similar to InterObject3D [7] by adding positive and negative click channels, but instead of simply setting the values to 1 or 0, the values in each channel reflect the distance of each point in the point cloud space from the click point. Smaller values indicate closer proximity, and larger values indicate greater distance. This relationship reflects the influence of the click point on each point in the point cloud, with closer points having a stronger influence and farther points having a weaker influence. The entire click distance map can be seen as a click influence map. This representation allows the network to intuitively perceive the influence of surrounding click points on each point in the space, rather than just knowing the location of the clicks, as in InterObject3D [7] or AGILE3D [8], which can cause the model to focus on overly structured local features, affecting its generalization performance.

To compute the click distance maps, we proceed as follows:

First, define the binary mask $\mathbf{M}_j$ for each point $\mathbf{P}_j$, where $\mathbf{M}_j = 1$ if there exists a click at point $\mathbf{P}_j$, and $\mathbf{M}_j = 0$ otherwise. Specifically, $\mathbf{M}_j$ can be either $\mathbf{M}_{\text{pos},j}$ indicating a positive click or $\mathbf{M}_{\text{neg},j}$ indicating a negative click.

Next, compute the distance from each point's xyz coordinates to its nearest clicked point's xyz coordinates:

$$d_i = \min_{j:\mathbf{M}_j=1} \|\mathbf{P}_{i,xyz} - \mathbf{P}_{j,xyz}\|_2 \tag{1}$$

where $\mathbf{P}_{i,xyz}$ and $\mathbf{P}_{j,xyz}$ represent the xyz coordinates of points $\mathbf{P}_i$ and $\mathbf{P}_j$, d_i represents the distance between each point in the point cloud space and the nearest click point,respectively.

Then, construct the distance map $\mathbf{D} \in \mathbb{R}^{N \times 1}$:

$$\mathbf{D} = [d_1, d_2, \ldots, d_N]^T \tag{2}$$

Finally, normalize the distance map to the range $[0, 1]$:

$$\mathbf{D}_{\text{norm}} = \frac{\mathbf{D} - \min(\mathbf{D})}{\max(\mathbf{D}) - \min(\mathbf{D})} \tag{3}$$

D_{norm} is the distance map channel that we finally use as network input.

B. Maximizing Click Effectiveness

1) Click-Specific Encoder: In selecting the distance map as the click representation, we also considered how to maximize the effectiveness of each click with limited click information. We found that the influence range of a single click in the

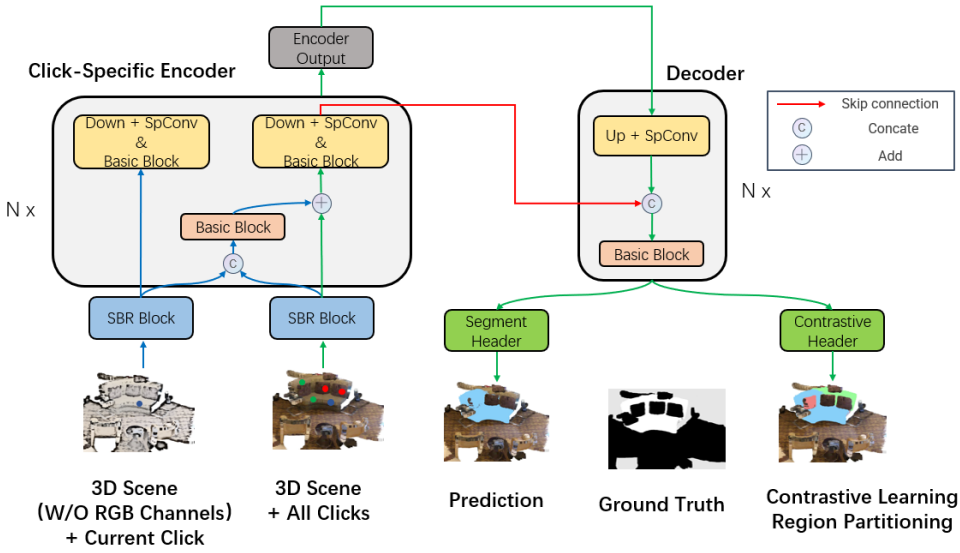

Fig. 5. This figure shows our overall network architecture. The green arrows represent the original backbone. In the encoder, we add the blue arrow components to form the Click-Specific Encoder module. The decoder's output is connected to both a segmentation head and a contrastive head for training on two tasks.

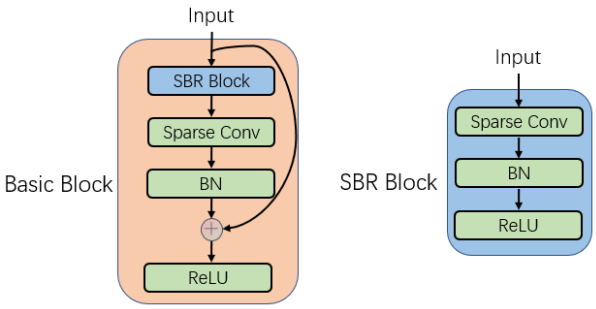

Fig. 6. Structure of the Basic Block and SBR Block.

point cloud space is the largest. As the number of clicks increases, the influence of the current click is actually affected by historical clicks, making the distance map more detailed but weakening the influence of the current click, as shown in Fig. 3. To ensure that the neural network does not overlook the information of individual clicks, we designed a *Click-Specific Encoder* module to generate a separate input channel for the current click, allowing the network to consider both types of distance map information to enhance the effect of the current click.

We designed the *Click-Specific Encoder* to fully leverage the strong influence of individual clicks, allowing the neural network to combine both single-click and multi-click features, thereby enhancing the click effectiveness.

We observed that as the number of clicks increases, the distance map becomes more refined, but the influence of the current click actually diminishes, being affected by previous clicks, as shown in Fig. 3. To enable the network to consider the impact of each individual click, we modified the encoder part of the original 3D U-Net to support both single-click and multi-click inputs.

As illustrated by the blue arrows in Fig. 5, compared to the main network input, the branch channel retains only the current click point and removes the color information from the point cloud (this design choice is made because some point cloud scenes do not have RGB information).

The single-click branch module extracts features through multiple levels of downsampling and sparse convolution operations. These extracted features from the single-click are then provided to the main network's encoder (represented by the green arrows) via residual connections [30]. This results in a fusion of single-click and multi-click features, with the residual connections ensuring that the final click performance is at least as good as before the addition of the single-click pathway. The features obtained from each layer of the *Click-Specific Encoder* are passed to the corresponding decoder layer via skip connections.

2) Contrastive Click Loss for Improved Interactive Segmentation: The *Click-Specific Encoder* can enhance the influence of the current click, but it is not sensitive to the boundary features of the object being segmented because the model does not have information about the specific contours of the object. To improve this, we propose enabling the model to learn the differences between the click points and the features inside and outside the object, thereby achieving more accurate segmentation.

We believe that interactive segmentation is essentially a process guided by interaction points, which can be viewed as a classification based on similarity. The model makes decisions about the classification of each point by evaluating the similarity between the point features in the point cloud space and the positive and negative interaction point features. Due to this similarity-based judgment, we propose using a method to strengthen the association between the features of the interaction points and the features of all points in the point cloud space in the feature space.

Each time we click, the neural network classifies points with features similar to the clicked point into the same category. As the number of clicks increases, points within the influence

range of the clicks are classified based on their similarity to nearby clicked points. This idea inspired us to use a contrastive learning approach.

We adopt a fully supervised contrastive learning approach to minimize the feature distance between the current positive click point and points within the target instance, while maximizing the feature distance between the current positive click point and points outside the target instance.

While the neural network without the contrastive loss adaptively learns the relationship between the clicked point's features and the features of points within or outside the instance, the contrastive learning acts as a boosting mechanism. It forces the network to bring the features of points within the instance closer to the clicked point's features with each click, thereby enhancing the performance of each click.

We modify the triplet loss [31] to serve as our click loss, which acts as an auxiliary task, while the main task remains the binary classification for foreground and background segmentation. We make the following improvements to the triplet loss, considering the characteristics of the interactive segmentation task: - The current positive click point serves as the anchor. - Points in the incorrectly predicted regions are considered as hard negative samples. As shown in Fig. 4, during the current click, we aim to classify the green region (false negatives) into the same category as the blue click point, while we do not want the red region (false positives) to be predicted as the foreground. Therefore, our hard negative samples are randomly sampled from these two regions.

We propose a modified triplet loss, termed Click Loss, specifically designed for interactive point cloud segmentation. In this formulation, the current click point serves as the anchor, and we sample K positive examples from the false positive region (Region P) and K negative examples from the false negative region (Region N).

The Click $loss$ is defined as:

$$L_{\text{click}}(A, \{P_k\}_{k=1}^{K}, \{N_k\}_{k=1}^{K}) = \max\left(0, \frac{1}{K}\sum_{k=1}^{K} d(A, P_k) - \frac{1}{K}\sum_{k=1}^{K} d(A, N_k) + \alpha\right) \quad (4)$$

where: A is the anchor point (current click point), $\{P_k\}_{k=1}^{K}$ are the K positive samples sampled from Region P, $\{N_k\}_{k=1}^{K}$ are the K negative samples sampled from Region N, d is the Euclidean distance, α is a predefined margin, typically a small positive number, K is the number of positive and negative samples. Both P and N represent regions that are currently predicted incorrectly. The distinction lies in the fact that P denotes a portion of the foreground, whereas N represents a segment of the background.

This loss function aims to minimize the average Euclidean distance between the anchor point and the positive samples while maximizing the average Euclidean distance between the anchor point and the negative samples, thereby enhancing the discriminative power of the network in interactive segmentation tasks.

C. Implementation Details

a) Loss: We use two types of loss functions for training: the commonly used segmentation losses, Dice loss [32] and CrossEntropyLoss, and an improved click loss using contrastive learning. The segmentation task is the primary task, and we use the click loss as an auxiliary training objective. The total loss L is defined as:

$$L = \lambda_1(L_{\text{CE}} + L_{\text{Dice}}) + \lambda_2 L_{\text{click}}$$

In our training process, we set λ_1 to 0.8 and λ_2 to 0.2. Additionally, in the click loss, we set α to 0.1, the margin value α is used to ensure that the distance between positive pairs is smaller than the distance between negative pairs by at least this margin.

b) Iterative Training and Testing: During the training and testing process, we simulate human clicks by generating positive or negative clicks at the center of the region with the maximum prediction error [9]. Since actual human clicks are not available during training and testing, we adopt a test strategy that simulates human clicks (at the center of the region with the maximum error for each click).

c) Training Details: In our training process, the model was trained for 60 epochs on the ScanNetV2 dataset. We used the SGD [33] optimizer with an initial learning rate of 0.005. The learning rate was dynamically adjusted using the OneCycleLR [34] scheduler, which is known for its effectiveness in finding optimal learning rates and improving convergence.

We observed that the model typically achieved stable performance around 52 epochs on the ScanNetV2 dataset. Our strategy involves training the model exclusively on ScanNetV2 and evaluating it on the same dataset to select the best-performing model. This best model is then tested on other datasets (SemanticKITTI, KITTI-360, and S3DIS) without additional fine-tuning. The entire training process took approximately 80 hours on an NVIDIA GeForce RTX 4090 GPU.

d) Backbone: Our backbone adopts a U-Net architecture, which is consistent with InterObject3D [7] and AGILE3D [8]. However, while they use the Minkowski Engine [35] library implementation, we use the SpConv [36] library. We chose SpConv [36] because it is easier to install and faster than Minkowski Engine [35].

V. Experiments

Our experiments first aim to demonstrate that our interactive method can surpass the current SOTA instance segmentation methods with only a limited number of clicks. Next, we conduct a comprehensive comparison with existing interactive segmentation methods. Not only do we achieve SOTA performance on multiple datasets, but we also show significantly better generalization on challenging datasets compared to current methods. Additionally, we perform a comparative experiment to illustrate the impact of sparse click representation and dense click representation on generalization, as shown in Fig. 8. Finally, we conduct an ablation study to evaluate the performance of each module in our method.

A. Experiment Settings

a) Datasets.: We conducted experiments on four datasets: two indoor datasets, ScanNetV2 [22] and S3DIS [37], and two outdoor datasets, SemanticKITTI [38] and KITTI-360 [39]. In the ScanNetV2 [22] dataset, the training set and validation set (which serves as the test set in ScanNetV2) can be divided into seen and unseen categories. The seen categories are the 20 benchmark classes in ScanNetV2 [22], and the remaining classes are considered unseen. We train our model on the ScanNetV2-Train set and test it on the ScanNetV2-Val set and the other three datasets. In our experiments, we refer to the 20 benchmark classes in the ScanNetV2 [22] dataset as ScanNet20, and the entire dataset with all classes as ScanNet40.

In our approach, we leverage the instance label information from the original four datasets. For a given scene, the target instance selected for segmentation is labeled as the foreground (assigned a label of 1), whereas all other instances are considered as the background (assigned a label of 0). Using this method of instance segmentation, we generate an interactive segmentation dataset specifically designed for individual object segmentation tasks.

To further enhance the robustness and adaptability of our model, we simulate human interactions during both training and testing phases. Specifically, we iteratively simulate human clicks to generate new click point data. These newly generated data points are then incorporated into the dataset, thereby enriching its content and diversity.

b) Baseline.: In terms of average precision (AP) metrics, we compare our method with the SOTA instance segmentation methods, OneFormer3D [5], Mask3D [6] and Competitor-MAFT [40].

For the interactive segmentation task, we compare our method with the SOTA point cloud interactive segmentation methods, InterObject3D [7] and AGILE3D [8]. InterObject3D [7] is trained on ScanNet20, while AGILE3D [8] is trained on ScanNet40. To ensure a fair comparison, we train our model on both ScanNet20 and ScanNet40, aligning the settings with those of InterObject3D and AGILE3D, respectively.

c) Evaluation Metrics.: We evaluate the performance of our model using two commonly used evaluation metrics in the 2D domain [13], [23], [41], [42]: **Number of Clicks (NOC)@q%**, the average number of clicks needed to reach q% Intersection over Union (IoU) between predicted and ground-truth masks on every object instance (thresholded at 20 clicks). The lower the NOC value the better, and **IoU@k**, the average IoU for k number of clicks per object instance. In addition, we also use the **mAP** metric to compare with point cloud instance segmentation (ignoring semantic information).

B. Comparison with Instance Segmentation Methods

First, we compare our method with current SOTA fully supervised instance segmentation methods [5], [6], [40]. As shown in Table I, our interactive method outperforms the fully supervised instance segmentation methods on the ScanNet [22] benchmark classes with just 3 clicks. After multiple iterations of clicking, our method achieves an AP score of 100.

TABLE I
COMPARISON WITH FULLY-SUPERVISED. WE COMPARE OUR METHOD WITH THE SOTA FULLY SUPERVISED INSTANCE SEGMENTATION METHOD.

	Method	AP	$AP_{50\%}$	$AP_{25\%}$
Benchmark Classes	Competitor-MAFT	61.8	81.6	89.6
	OneFormer3D	56.6	80.1	89.6
	Mask3D	51.5	77.0	90.2
	ClickEnhance(Ours) (1 click)	28.2	46.2	67.0
	ClickEnhance(Ours) (2 clicks)	56.5	79.9	**91.4**
	ClickEnhance(Ours) (3 clicks)	**66.7**	**86.8**	97.4
	ClickEnhance(Ours) (5 clicks)	79.2	94.2	99.4
	ClickEnhance(Ours) (10 clicks)	88.0	98.4	99.7
	ClickEnhance(Ours) (20 clicks)	91.4	99.0	100.0
Unseen Classes	Mask3D	5.3	13.1	24.7
	ClickEnhance(Ours) (1 click)	**23.3**	**42.5**	**72.5**
	ClickEnhance(Ours) (2 clicks)	40.8	64.5	86.0
	ClickEnhance(Ours) (3 clicks)	47.9	74.0	90.6
	ClickEnhance(Ours) (5 clicks)	60.2	85.3	96.1
	ClickEnhance(Ours) (10 clicks)	75.6	95.5	98.7
	ClickEnhance(Ours) (20 clicks)	84.7	97.7	100.0

In contrast, fully supervised instance segmentation methods perform poorly when encountering unseen classes during testing. Our interactive segmentation method, however, is able to effectively segment these unseen classes. On unseen classes, our method can achieve 4 times the precision of Mask3D [6] with just one click, and a few more clicks can produce high-quality object masks. These results demonstrate the significant advantages of our interactive method over one-shot instance segmentation methods. Moreover, our method is not limited by semantic categories. Specifically, the performance of instance segmentation methods drops by 89.7% when encountering unseen categories, while our method's performance only decreases by 17.3%.

C. Comparison with Interactive Segmentation Methods

1) Comparison on ScanNet20: We conducted a comprehensive comparison with InterObject3D [7] on ScanNet20 [22]. For in-domain comparisons, we divided the ScanNet-val dataset into seen classes and unseen classes, maintaining the same settings as InterObject3D. As shown in Table II, our method requires at least 3 fewer clicks than InterObject3D [7] to achieve the specified IoU score.

For out-of-domain results, as shown in Table III, our method shows some improvement over InterObject3D [7] on the S3DIS [37] dataset, but the improvement is less significant compared to the ScanNet dataset. We attribute this to the characteristics of the datasets. The point cloud density of the S3DIS [37] dataset is 10 times higher than that of the ScanNet dataset. InterObject3D uses a Cube size input representation, which makes the model focus more on the structured information around the clicked points. This learned feature performs better on the denser point clouds of the S3DIS dataset. However, our method, which is based on distance map, is less sensitive to changes in point cloud density, leading to less significant improvements on the S3DIS dataset.

Additionally, as shown in Table III, the structured information learned by InterObject3D [7] performs poorly on the SemanticKITTI [38] dataset, which is an outdoor dataset. It

TABLE II
INTERACTIVE 3D OBJECT SEGMENTATION ON SCANNETV2 VALIDATION IN WITHIN-DOMAIN EVALUATION.

ScanNetV2 val-	Seen			Unseen			All		
NOC @ k % IoU	80%	85%	90%	80%	85%	90%	80%	85%	90%
InterObject3D	8.3	10.6	13.6	11.7	14.1	16.5	9.6	11.8	14.6
ClickEnhance(Ours)	**5.2**	**6.9**	**9.6**	**7.8**	**10.0**	**13.0**	**6.2**	**8.0**	**10.9**

TABLE III
INTERACTIVE 3D OBJECT SEGMENTATION SCORES ON S3DIS AND SEMANTICKITTI IN OUT-OF-DOMAIN EVALUATION.

(trained on ScanNet)	S3DIS Area 5			SemanticKITTI		
NOC @ k % IoU	80%	85%	90%	80%	85%	90%
InterObject3D	6.8	8.9	11.8	19.4	19.5	19.6
ClickEnhance(Ours)	**6.0**	**7.5**	**9.8**	**8.8**	**9.6**	**10.5**

almost loses the advantage of interactive segmentation (i.e., generalization to unseen semantic categories). In contrast, our method shows impressive results on these challenging datasets, requiring at least 10 fewer clicks than InterObject3D to achieve the same level of precision.

2) Comparison on ScanNet40: In this section, we conduct a comprehensive comparison with AGILE3D [8] and the re-implemented InterObject3D++ by the authors of AGILE3D. As shown in Table IV, our method outperforms all current methods in most cases. In the ScanNet [22] dataset, our method achieves a higher average IoU with only 10 clicks compared to the previous methods with 15 clicks.

Despite performing slightly worse on the S3DIS dataset, our method still surpasses the current SOTA in the NoC@90 metric. We previously analyzed that this is due to the characteristics of the S3DIS dataset. Both S3DIS and ScanNet are indoor datasets with similar structures, but the point cloud density of S3DIS [37] is 10 times higher than that of ScanNet. Therefore, methods like InterObject3D [7] and AGILE3D [8], which use sparse click representations to learn structured information, perform better on the high-density, structurally similar S3DIS dataset.

However, when facing the challenging Kitti-360 outdoor dataset, the performance of InterObject3D and AGILE3D is markedly different from their performance on the S3DIS dataset. Our method demonstrates a significant advantage on Kitti-360 [39], achieving over 20 points higher IoU scores compared to the current SOTA methods, and requiring at least 4 fewer clicks.

D. Performance Curve Analysis

We conducted four additional experiments using the model trained on ScanNet40 [22] to generate four types of curves, which comprehensively measure the average performance of each method after 20 clicks. This provides a more intuitive view of the strengths and weaknesses of each method and module. The first experiment compares the overall performance of InterObject3D [7], InterObject3D++, AGILE3D [8], and ClickEnhance (Ours). The second experiment evaluates the performance of the same backbone but with different input representations: sparse input representation (represented by cube size) and dense input representation (represented by distance map) on various datasets. The third experiment aims to study the performance changes of the click strategy under different temperature parameters T. The four experiment is an ablation study of the different modules in our method.

1) Overall Performance Comparison: We first compare the average performance of InterObject3D [7], InterObject3D++, AGILE3D [8], and ClickEnhance (Ours) over 20 clicks, as shown in Fig. 7. By generating performance curves, we can intuitively see the performance of each method at different click counts. We observe that the current SOTA method, AGILE3D [8], has a significant advantage in the initial few clicks, with a substantial performance improvement. However, its performance gains diminish with subsequent clicks. For interactive segmentation tasks, obtaining high-precision masks with the fewest clicks is crucial, and it is evident that our ClickEnhance method has a more pronounced advantage.

In the S3DIS [37] dataset, as previously mentioned, our method is based on dense input representation using distance maps. This contrasts with the sparse input representations used by InterObject3D [7] and AGILE3D [8], which are better suited to learning the structural features of the indoor ScanNet [22] dataset. These learned features perform better when applied to the S3DIS [37] dataset, which has a much higher point density than ScanNet [22]. Our method shows a slight decrease in performance, but as the number of clicks increases, our performance curve eventually matches that of the SOTA methods.

For challenging outdoor datasets, the gap between current methods and our approach is significant. Our model is better at learning a general representation that is not limited to indoor datasets, which is a key advantage of interactive segmentation methods.

2) Impact of Input Representation: Through our experiments, we found that a sparse input representation using cube size (where most positions in the click channel are zero) causes the network to learn overly structured features. This leads to poor generalization performance when the model encounters outdoor datasets with significantly different structures. In contrast, our method based on dense input representation using distance map (which reflect the distance of each point in the point cloud to the nearest click point and do not have many zeros) effectively mitigates this issue.

We trained two models on ScanNet40 [22] with comparable performance and evaluated their performance on S3DIS [37] and KITTI-360 [39]. As shown in Fig. 8, the sparse representation using cube size performs better on the indoor dataset S3DIS [37] compared to the dense representation using distance maps. However, on the KITTI-360 [39] dataset, which has a significantly different structure, the sparse representation performs much worse. This validates our hypothesis.

3) Click Strategy Analysis: To analyze the robustness of click placement strategies, we propose a temperature-controlled probabilistic sampling method. Given a set of candidate points $\mathcal{P} = \{p_1, ..., p_N\}$ across all error regions, where $p_{\text{ref}} \in \mathcal{P}$ denotes the reference point (typically the centroid of the maximum-error region), the selection probability for point p_i is:

TABLE IV
COMPARISON OF INTERACTIVE SEGMENTATION METHODS

Method	Train → Eval	IoU@5 ↑	IoU@10 ↑	IoU@15 ↑	NoC@80 ↓	NoC@85 ↓	NoC@90 ↓
InterObject3D	ScanNet40 → ScanNet40	72.4	79.9	82.4	8.9	11.2	14.2
InterObject3D++		78.0	82.9	84.2	7.7	10.0	13.2
AGILE3D		79.9	83.7	85.0	7.1	9.6	12.9
ClickEnhance(Ours)		**81.2**	**85.9**	**87.7**	**5.9**	**7.8**	**10.7**
InterObject3D	ScanNet40 → S3DIS-A5	72.4	83.6	88.3	6.8	8.4	11.0
InterObject3D++		80.8	**89.2**	**91.5**	5.2	6.7	9.3
AGILE3D		**83.5**	88.2	89.5	**4.8**	**6.4**	9.5
ClickEnhance(Ours)		80.7	87.9	90.2	5.5	7.0	**9.2**
InterObject3D	ScanNet40 → KITTI-360	14.3	26.3	35.0	19.1	19.4	19.7
InterObject3D++		19.9	40.6	55.1	17.0	17.7	18.4
AGILE3D		44.4	49.6	54.9	14.2	15.5	16.8
ClickEnhance(Ours)		**62.6**	**76.2**	**81.7**	**9.8**	**11.1**	**12.7**

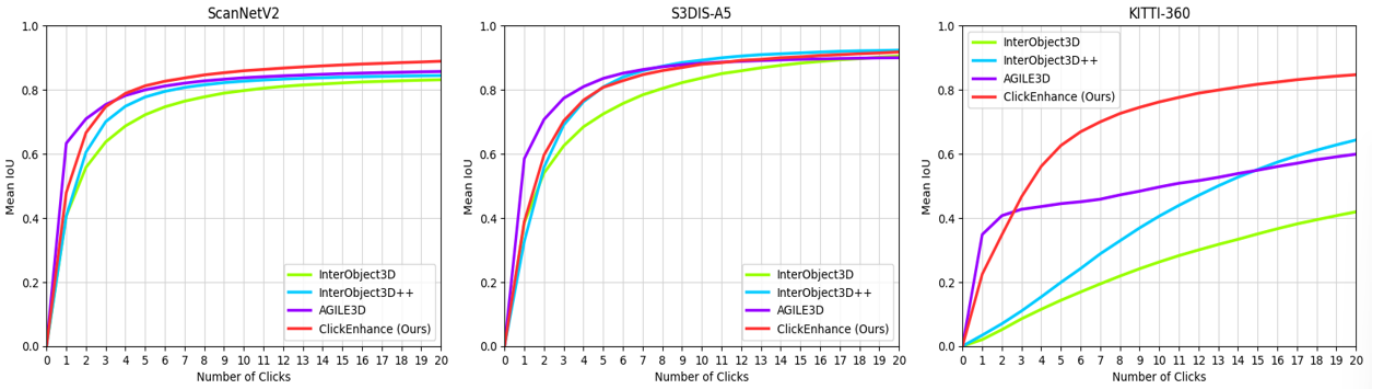

Fig. 7. Performance curves for different methods over 20 clicks.

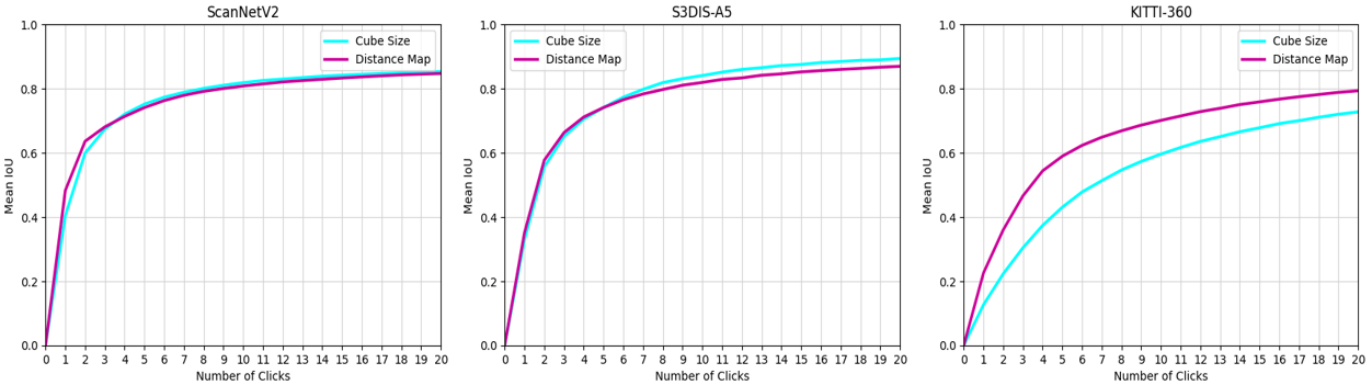

Fig. 8. Performance curves for different input representations on various datasets.

$$P(i) = \frac{\exp(-d_i \cdot (1-T))}{\sum_{j \in \mathcal{P}} \exp(-d_j \cdot (1-T))}, \quad T \in [0,1] \tag{5}$$

where:

- $d_i = \|p_i - p_{\text{ref}}\|_2$: Euclidean distance from candidate p_i to reference p_{ref}
- T: Exploration temperature controlling sampling diversity
- $j \in \mathcal{P}$: Index over all candidate points

The temperature parameter T governs a smooth transition between two distinct sampling behaviors:

Distance selection mechanism (When $T = 0$):

$$P(i) \propto \exp(-d_i)$$

This configuration assigns exponentially higher probabilities to candidate points proximal to p_{ref}, effectively implementing a soft $\arg\min$ operation. The resultant distribution strongly biases sampling toward the geometrically nearest neighbors in the configuration space, p_{ref} has the highest probability of being sampled as $T \to 0$.

Uniform Sampling Regime (When $T = 1$):

$$P(i) = \frac{1}{|\mathcal{P}|}$$

This degenerate case induces all candidate points in the feasible set $\mathcal{P}$ receive equal selection probability. The temperature-scaled distribution thereby degenerates to uniform randomness, ensuring unbiased exploration of the entire error tolerance region.

As shown in Fig. 9, on the ScanNet dataset, when $T \leq 0.3$, the maximum IoU drop is 5 points, but after 20 clicks, the final performance reaches 98.4% of the Base Model. When $T = 1$, the maximum IoU drop is 15 points, and the final performance after 20 clicks is 96.5% of the Base Model. On the S3DIS dataset, when $T \leq 0.3$, the maximum IoU drop is 2 points, and the final performance after 20 clicks is 98.8% of the Base Model; when $T = 1$, the maximum IoU drop is 13 points, and the final performance is 96.8% of the Base Model. On the KITTI-360 dataset, when $T \leq 0.3$, the maximum IoU drop is 2 points, and the final performance after 20 clicks is 98.7% of the Base Model; when $T = 1$, the maximum IoU drop is 6 points, and the final performance is 94.6% of the Base Model.

These results indicate that when $T \leq 0.3$, i.e., there is a high probability that the click position tends towards p_{ref}, the robustness of the click strategy across all three datasets is better. However, when T increases to 1, indicating completely random click positions, the overall performance exhibits significant fluctuations.

4) Ablation Study: In this ablation study, we evaluate the contribution of each module in our method, as shown in Fig. 10. The results demonstrate the impact of individual modules on the overall performance.

First, we tested the base model (Base) and the models with different modules added as follows.

- Distance Map Module: Our dense input representation of the distance map has good generalization capabilities. The experimental results show that the base model's performance does not drastically decline across the three datasets, indicating that this module effectively enhances the robustness and generalization of the model.
- CSE Module: Our CSE module enhances the effect of each click, especially in the first 3 clicks, where the performance improvement is significant. This indicates that the CSE module effectively improves the segmentation accuracy of early clicks.
- Click Loss: Our click loss module aims to enhance the model's ability to distinguish between inside and outside features of objects. The experimental results show that adding the click loss module further improves the overall performance of the model. This shows that this module effectively boosts the overall performance of the model.

In summary, the experimental results demonstrate that the addition of each module significantly improves the overall performance. Particularly, the distance map module and the CSE module excel in enhancing the model's generalization capabilities and the accuracy of early clicks, while the click loss module further raises the upper limit of the model's performance. These results validate the effectiveness and superiority of our method in interactive segmentation tasks.

E. Discussion on Limitations

As shown in Fig. 11, we visualize some challenging segmentation examples in indoor scenes. We found that the difficulty in indoor scenes lies in the segmentation of small objects, meaning our method does not perform well on these smaller objects. For outdoor scenes, segmenting road objects is particularly challenging. This is mainly due to the dispersed point clouds captured by LiDAR, which are not tightly connected in space, and the presence of other occluding objects (e.g., pedestrians, vehicles) on the road, leading to highly scattered final scans.

Future work could consider the following improvements:

- **Enhanced Context-Aware Modules**: Introduce more powerful context-aware modules to better handle the segmentation of small objects.
- **Enhance the Model's Robustness to Interference**: Enable the model to learn more structured information, such that even when occlusion causes spatial dispersion of an object, its various parts should still exhibit highly similar structural features and should not be treated as separate objects.

F. Inference Time

We conducted inference time tests using the ScanNetV2-val [22] dataset, comparing the baseline methods InterObject3D [7] and AGILE3D [8] with our ClickEnhance model. The results show that the average inference speed of the InterObject3D [7] model is 0.07 seconds per data sample, the AGILE3D [8] model has an average inference speed of 0.03 seconds per data sample, and our ClickEnhance model achieves an inference speed of 0.02 seconds per data sample. These results indicate that our model has the fastest inference speed, meeting the real-time requirements for interactive tasks.

VI. Conclusion

We propose a simple yet highly effective method for point cloud interactive segmentation. Our approach uses dense click input representations to address the poor generalization of sparse click input representations on challenging datasets. By fully utilizing current click information, our method significantly enhances the effect of each click. The Click-Specific Encoder module enables the model to fully consider the influence of individual clicks, while the click loss function based on contrastive learning helps the model better distinguish points at boundaries that belong to different instances. Our model outperforms SOTA methods on multiple datasets, especially on complex outdoor datasets. Additionally, our model achieves SOTA inference speed, meeting the real-time requirements of interactive segmentation tasks.

Despite the significant progress made by our method, it still faces some performance limitations. As shown in Fig. 6, even with 20 clicks, our method remains far from achieving 100% Intersection over Union (IoU), and the improvement rate gradually decreases. This is primarily because the model struggles to handle fine-grained regions in the later stages of interaction. We hope that future work can effectively address this issue by developing more advanced techniques to enhance the model's ability to handle small detail regions.

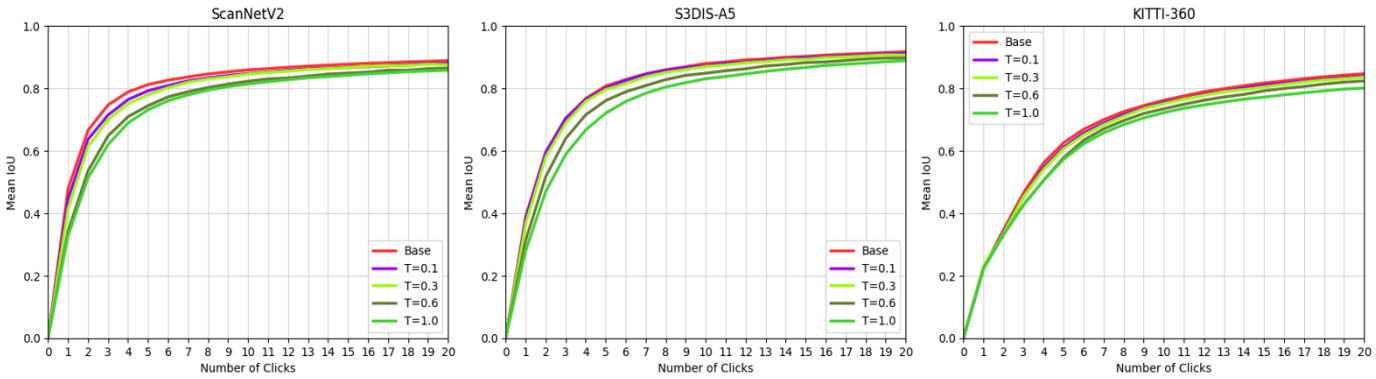

Fig. 9. Performance Comparison of Click Strategies under Different Temperature Parameters: Base denotes clicking the reference point (i.e., the original strategy of clicking the center of the maximum error region).

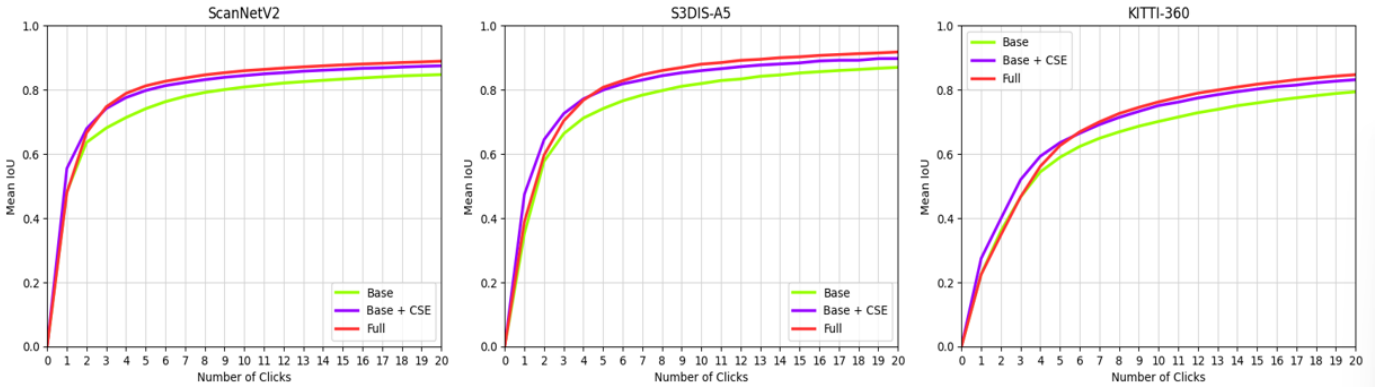

Fig. 10. Ablation study results showing the impact of individual modules. The models compared are: (1) Base (backbone + distance map), (2) Base + CSE (Click-Specific Encoder), and (3) Full (Base + CSE + Click loss).

Fig. 11. Failure cases in indoor and outdoor scenes. The top three images show scenarios where the model struggles in indoor settings, while the bottom three images depict challenges in outdoor environments.

REFERENCES

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[2] S. Yang, M. Hou, and S. Li, "Three-dimensional point cloud semantic segmentation for cultural heritage: a comprehensive review," *Remote Sensing*, vol. 15, no. 3, p. 548, 2023.

[3] A. Xiao, J. Huang, W. Xuan, R. Ren, K. Liu, D. Guan, A. El Saddik, S. Lu, and E. P. Xing, "3d semantic segmentation in the wild: Learning generalized models for adverse-condition point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 9382–9392.

[4] B. Zhang and P. Wonka, "Point cloud instance segmentation using probabilistic embeddings," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 8883–8892.

[5] M. Kolodiazhnyi, A. Vorontsova, A. Konushin, and D. Rukhovich, "Oneformer3d: One transformer for unified point cloud segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 20 943–20 953.

[6] J. Schult, F. Engelmann, A. Hermans, O. Litany, S. Tang, and B. Leibe, "Mask3d: Mask transformer for 3d semantic instance segmentation," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 8216–8223.

[7] T. Kontogianni, E. Celikkan, S. Tang, and K. Schindler, "Interactive object segmentation in 3d point clouds," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2023, pp. 2891–2897.

[8] Y. Yue, S. Mahadevan, J. Schult, F. Engelmann, B. Leibe, K. Schindler, and T. Kontogianni, "Agile3d: Attention guided interactive multi-object 3d segmentation," *arXiv preprint arXiv:2306.00977*, 2023.

[9] S. Mahadevan, P. Voigtlaender, and B. Leibe, "Iteratively trained interactive segmentation," *arXiv preprint arXiv:1805.04398*, 2018.

[10] N. Xu, B. Price, S. Cohen, J. Yang, and T. S. Huang, "Deep interactive object selection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 373–381.

[11] X. Chen, Z. Zhao, Y. Zhang, M. Duan, D. Qi, and H. Zhao, "Focalclick: Towards practical interactive image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1300–1309.

[12] K.-K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool, "Deep extreme cut: From extreme points to object segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 616–625.

[13] R. Benenson, S. Popov, and V. Ferrari, "Large-scale interactive object segmentation with human annotators," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 11 700–11 709.

[14] S. Zhang, J. H. Liew, Y. Wei, S. Wei, and Y. Zhao, "Interactive object segmentation with inside-outside guidance," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 12 234–12 244.

[15] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[16] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *2006 IEEE computer society conference on computer vision and pattern recognition (CVPR'06)*, vol. 2. IEEE, 2006, pp. 1735–1742.

[17] W. Wang, T. Zhou, F. Yu, J. Dai, E. Konukoglu, and L. Van Gool, "Exploring cross-image pixel contrast for semantic segmentation," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 7303–7313.

[18] D. Robert, H. Raguet, and L. Landrieu, "Scalable 3d panoptic segmentation as superpoint graph clustering," in *2024 International Conference on 3D Vision (3DV)*. IEEE, 2024, pp. 179–189.

[19] S. Shin, K. Zhou, M. Vankadari, A. Markham, and N. Trigoni, "Spherical mask: Coarse-to-fine 3d point cloud instance segmentation with spherical representation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 4060–4069.

[20] J. Lu, J. Deng, C. Wang, J. He, and T. Zhang, "Query refinement transformer for 3d instance segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 18 516–18 526.

[21] W. Zhao, Y. Yan, C. Yang, J. Ye, X. Yang, and K. Huang, "Divide and conquer: 3d point cloud instance segmentation with point-wise binarization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 562–571.

[22] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, "Scannet: Richly-annotated 3d reconstructions of indoor scenes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5828–5839.

[23] Z. Li, Q. Chen, and V. Koltun, "Interactive image segmentation with latent diversity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 577–585.

[24] G. Song, H. Myeong, and K. M. Lee, "Seednet: Automatic seed generation with deep reinforcement learning for robust interactive segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1760–1768.

[25] Z. Lin, Z.-P. Duan, Z. Zhang, C.-L. Guo, and M.-M. Cheng, "Focuscut: Diving into a focus view in interactive segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 2637–2646.

[26] Z. Lin, Z. Zhang, L.-Z. Chen, M.-M. Cheng, and S.-P. Lu, "Interactive image segmentation with first click attention," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 13 339–13 348.

[27] Y. Zhou, J. Gu, T. Y. Chiang, F. Xiang, and H. Su, "Point-sam: Promptable 3d segmentation model for point clouds," *arXiv preprint arXiv:2406.17741*, 2024.

[28] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[29] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*. Springer, 2015, pp. 234–241.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[31] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[32] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li, "Dice loss for data-imbalanced nlp tasks," *arXiv preprint arXiv:1911.02855*, 2019.

[33] S.-i. Amari, "Backpropagation and stochastic gradient descent method," *Neurocomputing*, vol. 5, no. 4-5, pp. 185–196, 1993.

[34] L. N. Smith, "Cyclical learning rates for training neural networks," in *2017 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2017, pp. 464–472.

[35] C. Choy, J. Gwak, and S. Savarese, "4d spatio-temporal convnets: Minkowski convolutional neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 3075–3084.

[36] S. Contributors, "Spconv: Spatially sparse convolution library," https://github.com/traveller59/spconv, 2022.

[37] I. Armeni, O. Sener, A. R. Zamir, H. Jiang, I. Brilakis, M. Fischer, and S. Savarese, "3d semantic parsing of large-scale indoor spaces," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1534–1543.

[38] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "Semantickitti: A dataset for semantic scene understanding of lidar sequences," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 9297–9307.

[39] Y. Liao, J. Xie, and A. Geiger, "Kitti-360: A novel dataset and benchmarks for urban scene understanding in 2d and 3d," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 3, pp. 3292–3310, 2022.

[40] D. Wang, J. Liu, H. Gong, Y. Quan, and D. Wang, "Competitorformer: Competitor transformer for 3d instance segmentation," *arXiv preprint arXiv:2411.14179*, 2024.

[41] W.-D. Jang and C.-S. Kim, "Interactive image segmentation via backpropagating refinement scheme," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5297–5306.

[42] J. Liew, Y. Wei, W. Xiong, S.-H. Ong, and J. Feng, "Regional interactive image segmentation networks," in *2017 IEEE international conference on computer vision (ICCV)*. IEEE, 2017, pp. 2746–2754.