

InSeg: Interactive Refinement via Intent Propagation for Point Cloud Semantic Segmentation

Yudong Liu^{1,2} , Yunfei Li^{1,2} , Ge Gao^{1,2,*} , Han Huang^{1,2} , and Ming Gu^{1,2}

¹ Beijing National Research Center for Information Science and Technology (BNRist), Tsinghua University, Beijing, China

² School of software, Tsinghua University, Beijing, China
liuyd23@mails.tsinghua.edu.cn, liyunfei24@mails.tsinghua.edu.cn,
gaoge@tsinghua.edu.cn, alvin98@foxmail.com, guming@tsinghua.edu.cn

Abstract. Despite recent progress in point cloud semantic segmentation, obtaining deployment-grade predictions on individual scenes remains challenging. Interactive refinement offers a practical solution, yet existing methods commonly interpret user clicks as single-point supervision, leading to fragmented updates and requiring redundant user interaction. We identify this limitation as the *Local Correction Trap*. Our key insight is that a correction click more naturally expresses an intent to revise a semantically coherent region than to relabel an isolated point. Based on this view, we propose **InSeg**, an interactive refinement framework that translates sparse clicks into reliable region-guided virtual supervision. InSeg propagates corrective intent through feature affinity, spatial connectivity, and object-level priors to construct denoised virtual labels. To support stable multi-round refinement, we further introduce a history-aware mechanism that combines uncertainty-guided updates with memory replay, reducing error accumulation and preventing previously corrected regions from being overwritten. Extensive experiments across multiple benchmarks show that InSeg consistently improves segmentation quality and interaction efficiency, achieving strong gains with only a few clicks.

Keywords: 3D point clouds · semantic segmentation · interactive refinement

1 Introduction

Accurate 3D semantic segmentation underpins a wide spectrum of spatial intelligence applications, including digital twin construction, robotic scene understanding, and immersive AR content creation. Although recent models [4, 10, 28, 29] have achieved strong performance on standard benchmarks, practical deployment still faces a persistent “last-mile” problem. In real scenes, off-the-shelf seg-

* Corresponding author.

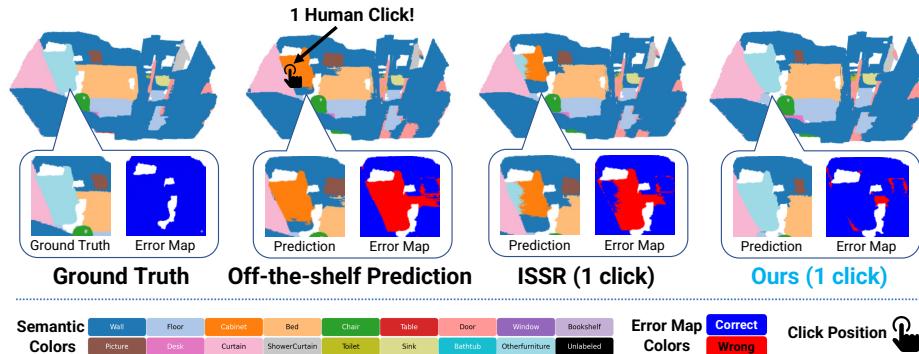


Fig. 1: Local correction trap. Given the same correction click on the same initial prediction, single-point refinement (*e.g.*, ISSR [34]) may remain confined to a local neighborhood, leading to fragmented corrections. InSeg instead propagates the click to an intent-consistent region for more coherent instance-level refinement.

mentors often produce predictions that remain locally inconsistent or instance-level incorrect, even when their overall benchmark scores are strong. Such errors can substantially reduce the utility of segmentation results in downstream workflows such as high-precision 3D reconstruction, computer-aided design, and large-scale annotation. As a result, practical use still demands efficient mechanisms for refining predictions on individual scenes.

Interactive refinement provides a natural solution to this problem by allowing users to correct evident mistakes during inference. However, existing approaches such as ISSR [34] typically treat a click as a single-point constraint and optimize the model only around the clicked location. This often leads to what we term the *Local Correction Trap*: supervision remains confined to a small neighborhood, producing local patching rather than coherent correction of the full error region (Fig. 1).

Our key insight is that a human click is not merely a request to flip one point label; it is a semantic signal indicating how a coherent region should be corrected. In an interactive loop, a click implicitly reflects the user’s holistic intent of correcting an object or an error region. Restricting supervision to isolated points discards this information and may force redundant interactions on the same structure.

This perspective reveals a central challenge for interactive refinement: how to propagate sparse

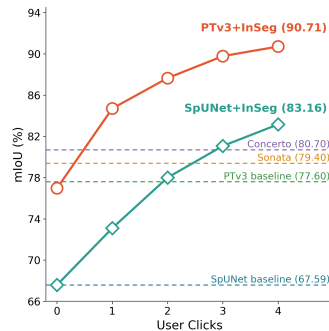


Fig. 2: Interaction efficiency. With only 3 clicks, InSeg enables SparseUNet to surpass large pre-trained models (*e.g.*, Sonata, Concerto) on ScanNet, and further improves performance when applied to PTV3.

corrective intent to semantically coherent regions while maintaining stable behavior across multiple interaction rounds.

To this end, we propose InSeg, an interactive refinement framework for 3D semantic segmentation. InSeg treats each correction click as a semantic seed rather than a single-point supervision. It propagates corrective intent through feature affinity, spatial connectivity, and object-level priors to identify an intent-consistent region, from which reliable virtual clicks are sampled for refinement. To stabilize multi-round refinement, we further introduce a history-aware objective that separates explicit correction from entropy regularization and replays reliable supervision from previously interacted regions.

Extensive experiments on ScanNet [6], S3DIS [1], and SemanticKITTI [2] demonstrate strong interaction efficiency and consistent segmentation improvements. With only 2 clicks, InSeg enables SparseUNet [5] to surpass PTv3 [29] on ScanNet; with 3 clicks, it outperforms large pre-trained models such as Sonata and Concerto [28, 35] (Fig. 2).

Our main contributions are:

1. We propose an intent propagation mechanism that converts sparse clicks into denoised and reliable virtual supervision.
2. We design an interaction- and history-aware stabilization strategy that promotes efficient and stable multi-round refinement.
3. We achieve state-of-the-art interactive performance, enabling lightweight models to outperform heavy pre-trained models with minimal human effort.

2 Related Work

2.1 Point Cloud Semantic Segmentation

Recent years have witnessed substantial progress in point cloud semantic segmentation, evolving from point-based methods [10, 19, 20, 25] and voxel-based sparse convolutions [4, 5, 7] to transformer-based architectures [14, 21, 27, 30, 32, 36]. Recent strong models such as PTv3 [29] and large-scale pre-trained frameworks including Sonata [28] and Concerto [35] have achieved impressive performance on standard benchmarks.

Despite these advances, strong benchmark performance does not eliminate the last-mile problem in practical deployment. Predictions on a specific scene may still remain locally inconsistent or object-level incorrect, which can significantly affect downstream reconstruction, editing, or annotation workflows. A straightforward solution is to collect additional labels and retrain or fine-tune the model, but this is often impractical due to annotation and computation costs, especially for large pre-trained backbones. This motivates efficient refinement methods that can improve scene-level predictions with minimal additional supervision.

2.2 Interactive Segmentation and Refinement

Interactive segmentation aims to refine predictions with minimal user effort. In the 2D domain, this field is well-established, with methods leveraging clicks [3, 8], scribbles [22], or bounding boxes [18] to guide segmentation, recently culminating in foundational models like SAM [11]. Extending these paradigms to 3D point clouds remains challenging due to the irregularity and sparsity of the data.

Beyond backbone improvement, some methods refine scene-specific predictions using self-supervised auxiliary tasks [9, 17, 24] or confidence-based objectives computed from model outputs [15, 26, 33] without human feedback. While such methods may improve prediction consistency, they provide limited control over which errors should be corrected. Our work instead focuses on interactive refinement, where sparse user corrections provide explicit and targeted supervision.

Existing 3D interactive approaches generally fall into two categories. The first focuses on interactive object extraction [12, 16, 31], where the goal is to segment an individual object from a scene. Recent methods such as Point-SAM [37] and Easy3D [23] further demonstrate the potential of prompt-driven interaction for 3D point clouds. These methods are valuable for instance selection and object-level interaction, but they are often designed for class-agnostic or promptable foreground extraction rather than dense multi-class semantic refinement of an off-the-shelf segmentor.

The second category targets interactive semantic refinement. In the 2D domain, IA [13] fine-tunes an off-the-shelf model by receiving correction clicks to rectify the segmented foreground objects in the image. In 3D point cloud segmentation, ISSR [34] adopts a predict-click-refine paradigm, where user clicks are treated as point-wise supervision to update model parameters online. While effective in correcting local errors, these methods typically interpret a click as point-wise supervision, limiting how far the corrective signal can propagate.

We argue that this point-wise view underestimates the semantic richness of human interaction. A correction click usually reflects the user’s intention to revise a coherent region rather than an isolated point. By confining supervision to local neighborhoods, existing methods tend to produce fragmented corrections and require repeated interactions on the same structure.

In contrast, our work advocates an intent propagation paradigm, where each click is treated as a semantic seed for coherent region-guided refinement. By combining feature, connectivity, and object-level cues, our method propagates sparse corrective intent to reliable support regions, enabling efficient and stable scene-specific refinement.

3 Problem Definition

Fig. 3 summarizes the interactive refinement loop. Given an input scene with N points, we denote the input as $X = \{(x_i, r_i)\}_{i=1}^N$ with $x_i \in \mathbb{R}^3$, where x_i and r_i represent the point coordinate and its associated attribute, respectively. Let f_θ be a pre-trained off-the-shelf semantic segmentation model. At interaction round

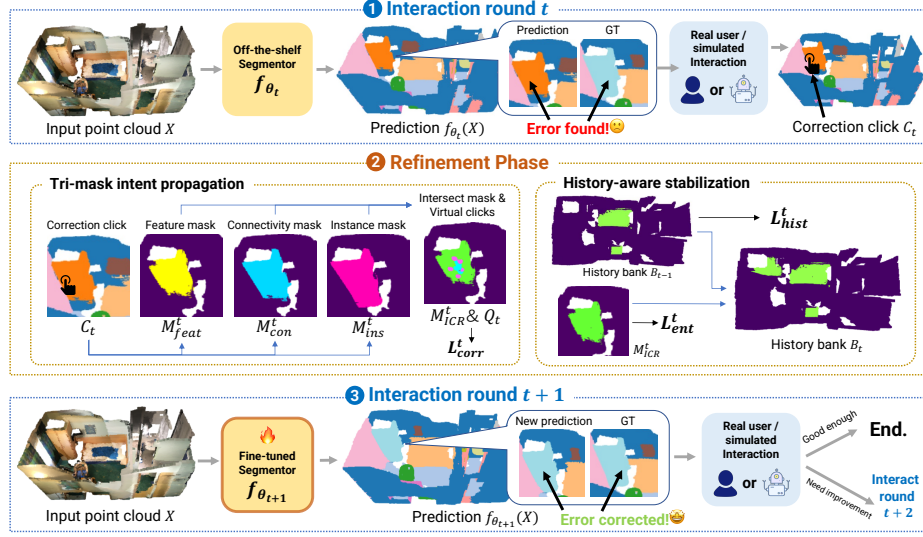


Fig. 3: Overview of InSeg. (1) Given the input point cloud X , at interaction round t , a correction click (index, correct label) is provided by a user or simulator based on the current segmentation $f_{\theta_t}(X)$. (2) The click is then expanded via tri-mask intent propagation to construct an intent-consistent region for virtual supervision, followed by history-aware stabilization. (3) The model f_{θ_t} is further fine-tuned with the combined supervision in (2) to obtain $f_{\theta_{t+1}}$, proceeding to the next round.

t , the model produces per-point predictions $f_{\theta_t}(X) = \{y_i\}_{i=1}^N$, where y_i denotes the predicted label of point i and θ_t denotes the model parameters at round t .

The user or simulator then provides sparse corrective clicks $C_t = \{(u_j, \hat{y}_j)\}_{j=1}^{U_t}$, where u_j is the index of the clicked point and $\hat{y}_j$ is the user-provided correct label. Our objective is to update $f_{\theta_t} \rightarrow f_{\theta_{t+1}}$ conditioned on C_t , and to iteratively refine the model across interaction rounds. The goal is to achieve maximal segmentation improvement with minimal user interactions.

4 Method

4.1 Tri-Mask Intent Propagation

Our refinement process starts from the predictions $f_{\theta_t}(X)$ of an off-the-shelf segmentation model on an input scene. We explicitly model each correction click as a semantic seed and propagate its corrective intent to a structurally consistent region that shares the same error pattern. We refer to this region as the Intent-Consistent Region (ICR). The objective is to identify a reliable subset of points within the ICR and assign them the same corrective label as the clicked point, while minimizing the risk of label mispropagation. To achieve this, we enforce consistency from three complementary perspectives.

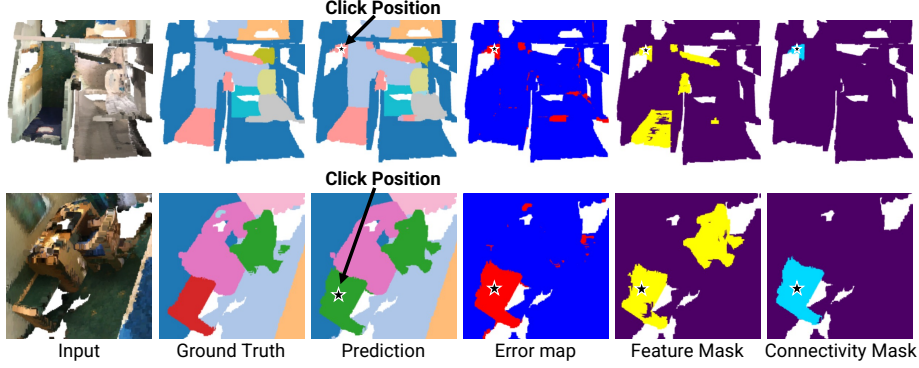


Fig. 4: Connectivity mask for cross-semantic control. Propagating supervision purely by feature affinity may spill corrections to previously correct regions; for example, correcting a wall point may inadvertently spread the “wall” label to a nearby door (top), while correcting a table point may likewise affect nearby chair points (bottom). The spatial connectivity mask restricts propagation to the connected prediction component containing the clicked point, reducing such leakage.

Mask-1: Feature similarity mask. We begin with feature similarity since the network-produced point features encode high-level semantic affinity, providing an effective cue for retrieving points that are likely to share the same error pattern as the clicked point. Given normalized per-point features e_{u_j} of the clicked point u_j , and let $\mathcal{I} = \{1, \dots, N\}$ denote the index of point cloud X , we identify points whose features are highly similar to the clicked point, which captures semantic affinity in feature space:

$$M_{\text{feat}}^j = \{i \in \mathcal{I} \mid \cos(e_i, e_{u_j}) \geq \tau_{\text{sim}}\}. \quad (1)$$

However, feature affinity alone is insufficiently selective, since distinct categories or neighboring structures may also exhibit similar features to the clicked point, as illustrated in Fig. 4. Consequently, a purely feature-based propagation can cause corrective signals to extend beyond the intended region. This motivates a second constraint.

Mask-2: Spatial connectivity mask. To prevent cross-semantic leakage, we restrict propagation to the spatially connected component that contains the clicked point, computed on the subset of points sharing its current predicted label. Let y_j denote the model’s predicted label at the clicked point u_j . Formally, we first consider all points whose predicted labels equal y_j , i.e., $\{i \in \mathcal{I} \mid y_i = y_j\}$. Among these, we retain only the 3D connected component that contains u_j :

$$M_{\text{con}}^j = \text{Comp}(u_j, \{i \in \mathcal{I} \mid y_i = y_j\}) \quad (2)$$

Here, $\text{Comp}(u_j, \cdot)$ denotes the spatial connected component containing u_j , computed via density-based clustering (DBSCAN) on the class-filtered subset.

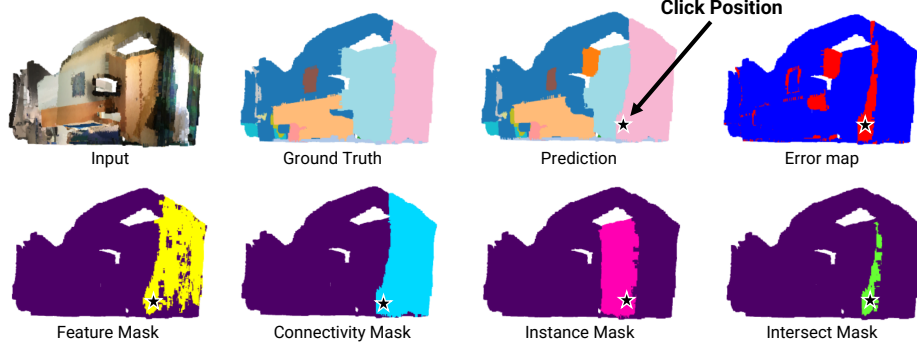


Fig. 5: Instance mask for cross-instance control. Propagating supervision using only feature and connectivity masks may spill corrections across instance boundaries; for example, correcting a furniture point near a curtain boundary may inadvertently spread the “furniture” label to nearby curtain regions. The instance mask restricts propagation to the target instance, reducing such leakage.

This constraint localizes corrective propagation to the region around the clicked point, rather than allowing it to spread globally through feature similarity. As a result, it reduces unintended correction transfer across distinct predicted objects that are semantically similar in the feature space, as shown in Fig. 4 (right).

The spatial connectivity constraint is effective when the predicted component corresponds to a single instance. In such cases, propagating the corrective label within this component is safe. However, these regions may not respect true instance boundaries. As illustrated in Fig. 5, connectivity mask and feature mask alone may still cross true instance boundaries. To reduce such cross-instance contamination, we further introduce an instance prior constraint.

Mask-3: Instance prior mask. To further mitigate cross-instance risk, we introduce a click-conditioned instance prior:

$$G_{\text{ins}} : (X, u_j) \mapsto M_{\text{ins}}^j. \quad (3)$$

The instance prior module estimates an object-level region associated with the clicked point. It operates as an interface and is not restricted to a specific architecture (*e.g.*, could be supervised/unsupervised instance segmentation, interactive object extraction). The key role of this module is to provide an object-aware structural constraint that complements feature and connectivity cues. As depicted in Fig. 5 (bottom-right), the instance prior counterbalances the potential cross-instance bias of the feature and connectivity masks, resulting in more precise identification of misclassified regions near instance boundaries.

The final intent-consistent region for click u_j is defined as the intersection of the three masks $M_{\text{ICR}}^j = M_{\text{feat}}^j \cap M_{\text{con}}^j \cap M_{\text{ins}}^j$. Even if the instance prior or feature grouping is imperfect, the intersection acts as a denoising operator. This process can be viewed as supervision denoising, where noisy propagation candidates are progressively filtered, leaving a high-confidence support region.

Virtual click generation and correction loss. Given the intent-consistent region M_{ICR}^j , we convert a single human click into a set of virtual clicks for dense supervision. We randomly sample m points from M_{ICR}^j to form S_{uni}^j to cover the spatial extent of the region. Meanwhile, to emphasize points that are most representative of the clicked instance, we additionally select the top- n points in M_{ICR}^j with the highest feature affinity to the clicked point u_j , forming S_{top}^j .

All virtual clicks inherit the same corrective label $\hat{y}_j$ from the original click. We define the virtual click set for click j as

$$V_j = \{(i, \hat{y}_j) \mid i \in S_{\text{uni}}^j \cup S_{\text{top}}^j\}. \quad (4)$$

For a multi-click round with U_t clicks, we aggregate the supervision as $Q_t = C_t \cup \bigcup_{j=1}^{U_t} V_j$, where $C_t = \{(u_j, \hat{y}_j)\}_{j=1}^{U_t}$ denotes the set of human clicks at round t . Correspondingly, for multi-click rounds we define the round-level intent-consistent region as the union of per-click regions, $M_{\text{ICR}}^t = \bigcup_{j=1}^{U_t} M_{\text{ICR}}^j$. Finally, the correction loss at round t is defined as a cross-entropy loss over the union of human and virtual clicks:

$$\mathcal{L}_{\text{corr}}^t = \sum_{(i, \hat{y}_i) \in Q_t} \text{CE}(z_i^t, \hat{y}_i), \quad (5)$$

where z_i^t denotes the model logits at point i in round t . The total correction loss at round t is further accumulated over all rounds up to t .

4.2 History-Aware Stabilization

While the proposed correction loss becomes substantially more effective by leveraging a larger set of virtual clicks, it also makes online updates more aggressive. In previous work ISSR [34], the stabilization term is based on per-round entropy changes with point-wise hard filtering, which is inherently short-sighted: regions corrected in early rounds may be unintentionally overwritten by subsequent updates. To address this, instead of simply utilizing “single-round thresholded entropy minimization”, we propose a two-branch objective that is both interaction-aware and history-aware.

Interaction-aware entropy minimization. Let $p_{i,c}^t = \text{softmax}(z_{i,c}^t)$ denote the predicted probability of class c at point i in round t , and

$$H(p_i^t) = - \sum_c p_{i,c}^t \log p_{i,c}^t \quad (6)$$

be the predictive entropy. Following ISSR, we first obtain a set of points P^t for entropy minimization using their entropy-change criterion. Different from ISSR, we exclude the current interaction region M_{ICR}^t from entropy minimization, so that points receiving explicit corrective supervision are optimized only by the correction loss. This decouples correction from stabilization in the most uncertain and boundary-confused areas, and avoids conflicting gradients caused by forcing

low entropy on points whose labels are being actively corrected:

$$\mathcal{L}_{\text{ent}}^t = \sum_{i \in \mathcal{E}^t} H(p_i^t), \quad \mathcal{E}^t = P^t \setminus M_{\text{ICR}}^t \quad (7)$$

History-aware replay loss. To prevent catastrophic forgetting, we maintain a history bank that stores pseudo-labeled supervision from *previously interacted* regions. Specifically, after interaction round r , for every point $k \in M_{\text{ICR}}^r$ we store its index k , a pseudo label $\tilde{y}_k$ (the predicted label after refinement), and a confidence score $s_k = \max_c(p_{k,c})$. At the beginning of round t , the bank contains all records accumulated from rounds $0, \dots, t-1$:

$$\mathcal{B}_t = \{(k, \tilde{y}_k, s_k)\}, \quad \text{where } (k, \tilde{y}_k, s_k) \text{ are collected from } M_{\text{ICR}}^{0:t-1}. \quad (8)$$

We then select a reliable subset by requiring both high confidence and label consistency with the current prediction,

$$\bar{\mathcal{B}}_t = \{(k, \tilde{y}_k) \mid (k, \tilde{y}_k, s_k) \in \mathcal{B}_t, s_k \geq \tau_{\text{conf}}, \tilde{y}_k = y_k^t\}, \quad (9)$$

and apply a replay cross-entropy loss using $\tilde{y}_k$ as the pseudo label:

$$\mathcal{L}_{\text{hist}}^t = \sum_{(k, \tilde{y}_k) \in \bar{\mathcal{B}}_t} \text{CE}(z_k^t, \tilde{y}_k). \quad (10)$$

This replay branch anchors the model on previously corrected and confident regions, reducing the chance that later aggressive updates overwrite earlier improvements.

Finally, the overall loss term at interaction round t is defined as:

$$\mathcal{L}^t = \mathcal{L}_{\text{corr}}^t + \beta \mathcal{L}_{\text{ent}}^t + \gamma \mathcal{L}_{\text{hist}}^t \quad (11)$$

Together, these components enable a refinement process that balances effective correction propagation with stable multi-round updates. We validate the contribution of each component through targeted ablations on mask design, interaction-aware entropy minimization, and history replay.

5 Experiment

5.1 Experiment Configurations

Backbones and datasets. Following ISSR [34], we evaluate on ScanNet [6], S3DIS [1], and SemanticKITTI (SKITTI) [2]. For fair comparison, we adopt the same backbone settings as ISSR: SparseUNet (SpUNet) [5] on ScanNet, Point Transformer (PT) [36] on S3DIS, and RandLA-Net [10] on SKITTI. We further evaluate InSeg with Point Transformer V3 (PTv3) [29] to assess its effectiveness on a SOTA backbone.

Table 1: Main comparison under the standard setting. We report scene-averaged mIoU/mAcc after 5, 10, and 20 clicks. Initial performance: 45.6/78.9 (SpUNet@ScanNet), 55.8/84.3 (PT@S3DIS), and 29.6/70.7 (RLA@SKITTI).

Backbone @Dataset	Method	mIoU			mAcc		
		@5	@10	@20	@5	@10	@20
SpUNet @ScanNet	IA	58.0	61.2	63.9	87.1	88.9	90.4
	ISSR	75.1	80.7	85.8	87.8	90.7	93.1
	InSeg	83.6	87.2	88.8	90.5	92.5	93.6
PT @S3DIS	IA	62.4	65.2	69.2	89.2	90.6	92.2
	ISSR	81.5	85.3	88.8	91.3	93.5	95.0
	InSeg	82.0	86.0	89.8	91.6	93.5	95.2
RLA @SKITTI	IA	37.8	40.5	46.3	76.1	74.9	72.1
	ISSR	53.1	63.1	68.7	76.7	78.7	80.4
	InSeg	62.1	66.0	69.4	77.5	79.2	81.0

Table 2: Completion rate under different NoC thresholds. We report the percentage of scenes that reach the target mIoU within 5, 10, or 20 clicks.

Backbone @Dataset	Method	Completion Rate		
		≤ 5	≤ 10	≤ 20
SpUNet @ScanNet (NoC@75)	IA	20.5	29.5	39.0
	ISSR	57.5	73.7	87.3
	InSeg	78.5	92.5	97.0
PT @S3DIS (NoC@80)	IA	14.7	18.7	33.3
	ISSR	69.3	81.3	86.7
	InSeg	73.3	84.0	90.6
RLA @SKITTI (NoC@60)	IA	5.8	11.4	26.0
	ISSR	35.7	62.9	74.1
	InSeg	59.2	67.6	75.0

Compared methods. We compare InSeg with two representative interactive baselines. ISSR is the closest 3D counterpart, as it performs interactive semantic refinement for off-the-shelf point cloud semantic segmentation networks under the same multi-class setting. To the best of our knowledge, no other methods in the 3D domain explicitly address this task. Therefore, we additionally adapt IA [13], a 2D interactive refinement method that can be applied to off-the-shelf models without task-specific modification.

Evaluation metrics. Following ISSR, we report scene-level mIoU and mAcc by computing mIoU/mAcc for each individual scene and then averaging these per-scene scores over all test scenes. We also report the completion rate following ISSR, i.e., the percentage of scenes reaching the target mIoU within a given click budget. All click budgets are counted per scene.

Implementation details. In practical scenarios, InSeg supports manual user-provided correction clicks. However, to ensure a fair comparison, we adopt the same simulated click generation strategy as ISSR in all experiments. We also follow ISSR for the warm-up scheme, entropy-

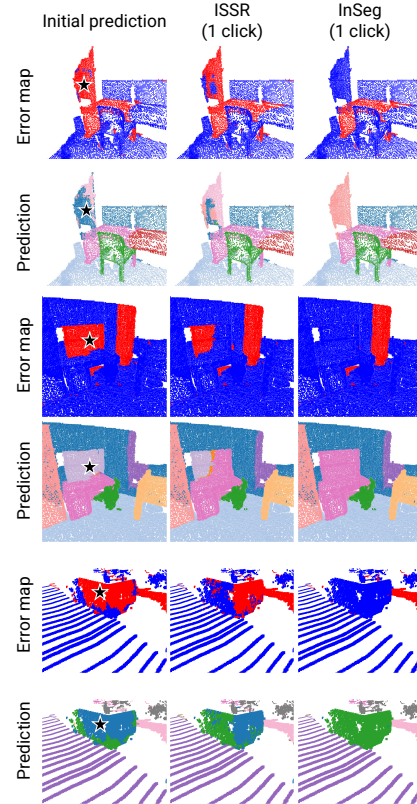


Fig. 6: Single click refinement. The same user click (black star) is provided to both ISSR and InSeg.

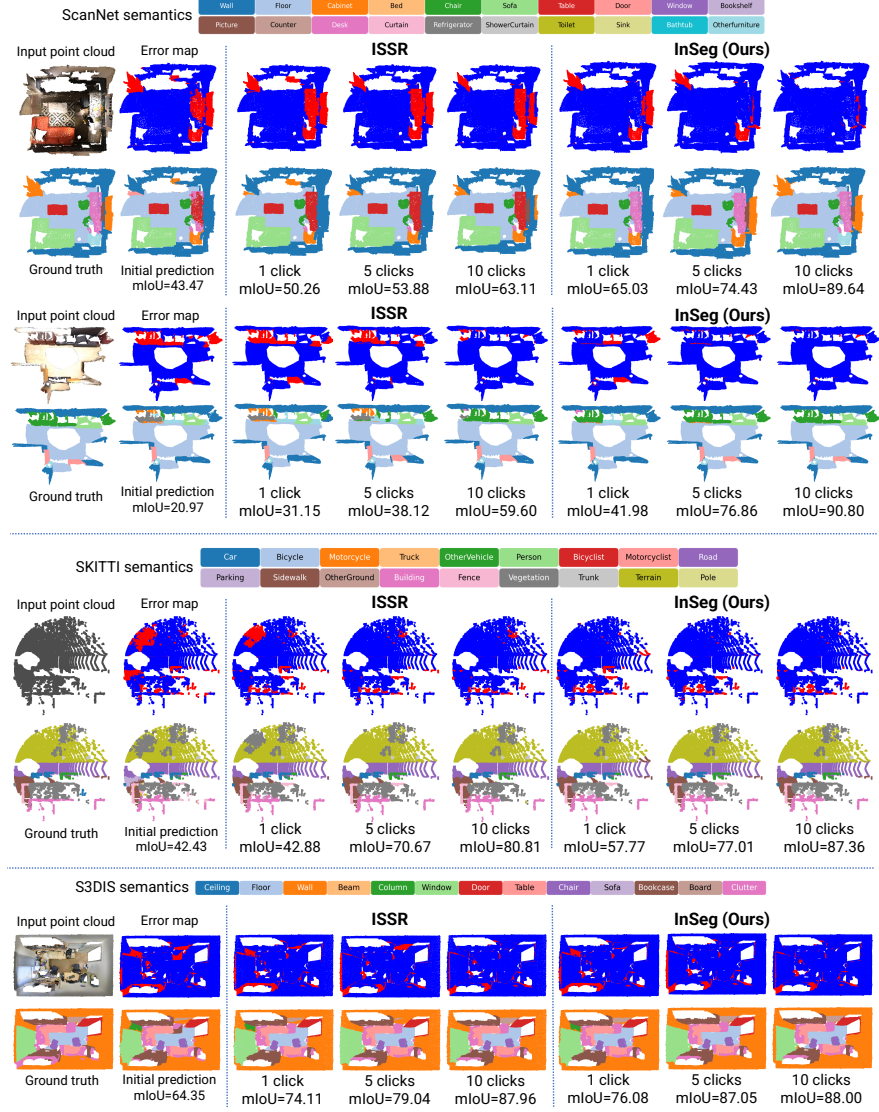


Fig. 7: Qualitative Results. Qualitative comparison between ISSR and InSeg on SpUNet@ScanNet, RLA@SKITTI, and PT@S3DIS.

filter threshold, β , and DBSCAN clustering. We set $\tau_{\text{sim}} = \tau_{\text{conf}} = 0.9$ throughout. For the instance prior, we use NPISeg3D [16], which segments individual instances from point cloud scenes in a class-agnostic manner. For virtual clicks, we set $(m, n, \gamma) = (5, 15, 2)$ on ScanNet/S3DIS and $(2, 5, 20)$ on SKITTI. We use the same optimizer as ISSR and perform 5 update steps per interaction. All experiments are conducted on a single RTX 4090 GPU with 24 GB memory.

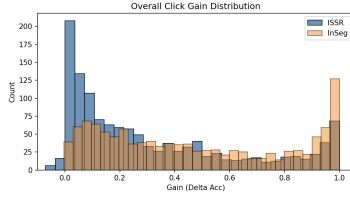


Fig. 8: Distribution of per-click gain.

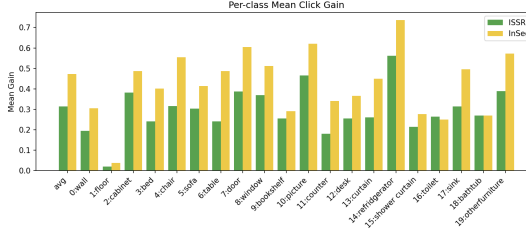


Fig. 9: Category-wise averaged per-click gain.

5.2 Main Results

Quantitative results. As shown in Tab. 1, InSeg consistently outperforms IA and ISSR across all datasets. On ScanNet with SpUNet, it improves mIoU@5 from 75.1 to 83.6 and maintains clear gains at 10 and 20 clicks. Similar improvements are observed on S3DIS and SKITTI, indicating that the proposed refinement strategy is effective across both indoor and outdoor scenes. Tab. 2 further shows that InSeg achieves higher completion rates under all NoC thresholds. For example, on ScanNet, the completion rate at $\text{NoC}@75 \leq 5$ increases from 57.5% to 78.5%. Overall, InSeg improves both final segmentation quality and interaction efficiency.

Qualitative results. Fig. 6 shows the refinement results with a single click. Given the same user click (the black star), ISSR mainly corrects errors in the immediate neighborhood of the click, often falling into the *local correction trap*. In contrast, InSeg leverages intent-aware propagation to extend corrections across the local region, leading to more complete error recovery. Fig. 7 further compares ISSR and InSeg under multi-round interactions. At different interaction stages, InSeg consistently corrects target errors while preserving previously refined regions, demonstrating stable long-range effectiveness. Notably, in some cases, InSeg with 5 clicks achieves performance comparable to, or slightly better than, ISSR with 10 clicks, indicating higher interaction efficiency.

Per-click gain analysis. To better quantify the benefit of each interaction, we compute the instance-level accuracy gain on ScanNet using ground-truth instance masks. Specifically, for each click, we define the gain as the change in segmentation accuracy within the ground-truth mask of the clicked instance before and after that interaction. Fig. 8 presents the distribution of per-click gains for ISSR and InSeg. Compared with ISSR, InSeg exhibits substantially fewer low-gain clicks and a distribution that shifts toward higher gains. Fig. 9 further reports the category-wise averaged gains, showing a consistent advantage of InSeg across semantic classes. In addition, we analyze the number of repeatedly clicked instances N_{inst} and the

Table 3: Instance-level repeated-click statistics.

Method	N_{inst}	N_{scene}
ISSR	233	204
InSeg	170	149

Table 4: Cross-dataset transfer evaluation. The model is trained on ScanNet and evaluated on S3DIS over the 8 classes shared by both datasets, using the scene-averaged metric. Initial performance is 66.88/77.31 (mIoU/mAcc).

Backbone @Dataset	Method	@1	@2	mIoU			@1	@2	mAcc		
				@3	@4	@5			@3	@4	@5
SpUNet	ISSR	70.09	73.06	75.13	77.21	78.35	79.65	81.53	83.22	84.80	85.63
@S3DIS	InSeg	71.82	75.39	78.69	80.65	82.66	81.28	84.23	86.89	88.55	90.27

Table 5: Results with a SOTA backbone. We compare ISSR and InSeg on SpUNet and PTV3 under the class-averaged metric. Initial performance is 67.59/76.22 for SpUNet and 76.97/84.15 for PTV3 (mIoU/mAcc).

Backbone @Dataset	Method	@1	@2	mIoU			@1	@2	mAcc		
				@3	@4	@5			@3	@4	@5
SpUNet	ISSR	72.18	74.94	76.77	78.53	79.67	79.67	81.88	83.22	84.68	85.57
@ScanNet	InSeg	73.10	78.00	81.07	83.16	84.70	80.58	84.70	86.91	88.62	89.72
PTv3	ISSR	83.25	87.05	89.17	89.73	90.80	88.94	91.60	93.22	93.66	94.67
@ScanNet	InSeg	84.70	87.65	89.78	90.71	91.38	90.05	92.06	93.90	94.52	94.81

number of scenes containing such instances N_{scene} on ScanNet, as summarized in Tab. 3. Overall, InSeg achieves higher per-click gains and demonstrates a clearer instance-level correction behavior, significantly reducing repeated interactions on the same instance compared with ISSR.

Cross-dataset transfer setting. To further examine the robustness of interactive refinement beyond the standard in-domain setting, we transfer a SpUNet trained on ScanNet to S3DIS and evaluate it on their 8 overlapping categories. As shown in Tab. 4, InSeg maintains a clear advantage over ISSR throughout the interaction process, with a widening margin as clicks increase, suggesting improved generalization when refining models in unseen domains.

Incorporating SOTA backbone. We further apply InSeg to a state-of-the-art backbone, Point Transformer V3 (PTv3). For consistency with the PTv3 evaluation protocol, we report class-averaged mIoU and mAcc. For reference, we also include the corresponding class-averaged metrics obtained with SpUNet. The results in Tab. 5 show that InSeg still provides consistent accuracy improvements over ISSR even when applied to a strong backbone. In particular, with only four clicks, the mIoU increases from approximately 77% to 90%, indicating that the proposed refinement strategy remains effective for modern high-performance models.

5.3 Ablation Studies

Intent-Consistent Region. Tab. 6 validates the design of the tri-mask Intent-Consistent Region. Starting from point-wise supervision only, adding the feature mask already brings a clear gain (75.10→78.89 mIoU), showing the benefit of propagating a click according to feature similarity. Introducing the connectivity

Table 6: Intent-consistent region ablation.

Feature	Connectivity	Instance	mIoU	mAcc
$\times$	$\times$	$\times$	75.10	87.80
$\checkmark$	$\times$	$\times$	78.89	87.53
$\checkmark$	$\checkmark$	$\times$	81.39	89.28
$\checkmark$	$\checkmark$	$\checkmark$	83.62	90.55

Table 8: Replay branch ablation. Effect of the replay loss weight γ .

γ	mIoU	mAcc
0	82.20	90.27
1	82.94	90.33
2	83.62	90.55
4	82.91	90.20

Table 7: Virtual clicks ablation. We consider 5 region clicks (N_{reg}) and 15 similarity clicks (N_{sim}) as default setting.

N_{reg}	mIoU	mAcc	N_{sim}	mIoU	mAcc
0	82.95	90.15	0	82.62	89.73
2	83.19	90.41	5	83.13	90.19
4	83.56	90.50	10	83.27	90.26
5	83.62	90.55	15	83.62	90.55
6	83.50	90.42	20	82.87	90.23

Table 9: Interaction-aware entropy minimization ablation.

Exclude M_{ICR}^t	mIoU	mAcc
$\times$	82.91	90.16
$\checkmark$	83.62	90.55

mask further improves performance to 81.39 mIoU by suppressing cross-semantic leakage. Further adding the instance prior yields the best result, reaching 83.62 mIoU. These results confirm that the three masks play complementary roles, and that reliable intent propagation requires jointly enforcing semantic affinity, structural consistency, and object-level constraints.

Virtual clicks. Tab. 7 analyzes the effect of virtual-click composition via controlled single-factor ablations. For each block, we vary only one type of virtual click while fixing the other to its default value, thereby isolating its individual contribution without changing the overall refinement framework. We observe that both region clicks and similarity clicks yield positive improvements when moderately increased, indicating that each provides useful yet distinct supervision signals. The optimal setting is 5 region clicks and 15 similarity clicks, suggesting that effective refinement requires a balance between region-level coverage and feature-aligned emphasis.

History-aware stabilization. Tab. 8 and Tab. 9 verify the proposed history-aware stabilization. The replay branch improves performance from 82.20 to 83.62 mIoU when γ increases from 0 to 2, while a larger weight degrades accuracy, suggesting that replay should regularize rather than dominate online updates. In addition, excluding the current intent-consistent region from entropy minimization improves mIoU from 82.91 to 83.62. This supports our design choice to decouple explicit correction from confidence regularization, leading to more stable multi-round refinement.

6 Limitation and Future Work

As a plug-and-play interactive refinement framework, InSeg’s absolute gain can correlate with the underlying backbone quality and its feature representation:

stronger models may leave less room for improvement, while weaker ones may benefit more from intent propagation. In addition, the current pipeline performs interaction and online updates on a single scene and re-initializes the model when moving to a new scene. This design keeps the refinement strictly scene-specific, but does not yet exploit the potential benefit of accumulating knowledge across scenes. A natural extension is to introduce a memory mechanism (*e.g.*, selective replay or meta-style adaptation) so that corrections from previous scenes can benefit future scenes while maintaining robustness. Moreover, superpoints provide pre-constructed local geometric regions that could serve as efficient local priors for intent propagation; incorporating superpoint-based propagation to simplify the current tri-mask design is an interesting direction for future work.

7 Conclusion

In this work, we presented InSeg, an interactive refinement framework to improve the performance of off-the-shelf point cloud segmentors on individual scenes. InSeg interprets each correction click as region-level intent and generates denoised virtual supervision via tri-mask propagation that integrates feature affinity, spatial connectivity, and instance priors. To stabilize multi-round updates, we introduce a history-aware objective with interaction-aware entropy regularization and memory replay to preserve previous fixes. Experiments across multiple benchmarks demonstrate consistent gains with only a few clicks. Looking forward, we aim to leverage cross-scene memory for continual improvement across scenes.

Acknowledgements

The corresponding author is Ge Gao. This work was supported in part by the National Key Research and Development Program of China under Project No. 2024YFC3809102.

References

1. Armeni, I., Sener, O., Zamir, A.R., Jiang, H., Brilakis, I., Fischer, M., Savarese, S.: 3d semantic parsing of large-scale indoor spaces. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1534–1543 (2016)
2. Behley, J., Garbade, M., Milioto, A., Quenzel, J., Behnke, S., Stachniss, C., Gall, J.: Semantickitti: A dataset for semantic scene understanding of lidar sequences. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 9297–9307 (2019)
3. Chen, X., Zhao, Z., Zhang, Y., Duan, M., Qi, D., Zhao, H.: Focalclic: Towards practical interactive image segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 1300–1309 (2022)
4. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 3075–3084 (2019)

5. Contributors, S.: Spconv: Spatially sparse convolution library. <https://github.com/traveller59/spconv> (2022), accessed: 2026-06-25
6. Dai, A., Chang, A.X., Savva, M., Halber, M., Funkhouser, T., Nießner, M.: Scannet: Richly-annotated 3d reconstructions of indoor scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5828–5839 (2017)
7. Graham, B., Engelcke, M., Van Der Maaten, L.: 3d semantic segmentation with submanifold sparse convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9224–9232 (2018)
8. Hao, Y., Liu, Y., Wu, Z., Han, L., Chen, Y., Chen, G., Chu, L., Tang, S., Yu, Z., Chen, Z., et al.: Edgeflow: Achieving practical interactive segmentation with edge-guided flow. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1551–1560 (2021)
9. Hatem, A., Qian, Y., Wang, Y.: Point-tta: Test-time adaptation for point cloud registration using multitask meta-auxiliary learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 16494–16504 (2023)
10. Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A.: Randla-net: Efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11108–11117 (2020)
11. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 4015–4026 (2023)
12. Kontogianni, T., Celikkan, E., Tang, S., Schindler, K.: Interactive object segmentation in 3d point clouds. arXiv preprint arXiv:2204.07183 (2022)
13. Kontogianni, T., Gygli, M., Uijlings, J., Ferrari, V.: Continuous adaptation for interactive object segmentation by learning from corrections. In: European Conference on Computer Vision. pp. 579–596. Springer (2020)
14. Lai, X., Liu, J., Jiang, L., Wang, L., Zhao, H., Liu, S., Qi, X., Jia, J.: Stratified transformer for 3d point cloud segmentation. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8500–8509 (2022)
15. Liang, J., Hu, D., Feng, J.: Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation. In: International conference on machine learning. pp. 6028–6039. PMLR (2020)
16. Liu, J., Zhou, P., Xiao, Z., Shen, J., Yin, W., Sonke, J.J., Gavves, E.: Probabilistic interactive 3d segmentation with hierarchical neural processes. arXiv preprint arXiv:2505.01726 (2025)
17. Liu, Y., Kothari, P., Van Delft, B., Bellot-Gurlet, B., Mordan, T., Alahi, A.: Ttt++: When does self-supervised test-time training fail or thrive? Advances in Neural Information Processing Systems **34**, 21808–21820 (2021)
18. Maninis, K.K., Caelles, S., Pont-Tuset, J., Van Gool, L.: Deep extreme cut: From extreme points to object segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 616–625 (2018)
19. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
20. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
21. Robert, D., Raguet, H., Landrieu, L.: Efficient 3d semantic segmentation with superpoint transformer. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17195–17204 (2023)

22. Sambaturu, B., Gupta, A., Jawahar, C., Arora, C.: Scribblenet: Efficient interactive annotation of urban city scenes for semantic segmentation. *Pattern Recognition* **133**, 109011 (2023)
23. Simonelli, A., Müller, N., Kotschieder, P.: Easy3d: A simple yet effective method for 3d interactive segmentation. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. pp. 24707–24716 (2025)
24. Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A., Hardt, M.: Test-time training with self-supervision for generalization under distribution shifts. In: *International conference on machine learning*. pp. 9229–9248. PMLR (2020)
25. Thomas, H., Qi, C.R., Deschaud, J.E., Marcotegui, B., Goulette, F., Guibas, L.J.: Kpconv: Flexible and deformable convolution for point clouds. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 6411–6420 (2019)
26. Wang, D., Shelhamer, E., Liu, S., Olshausen, B., Darrell, T.: Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726* (2020)
27. Wang, P.S.: Octformer: Octree-based transformers for 3d point clouds. *ACM Transactions on Graphics (TOG)* **42**(4), 1–11 (2023)
28. Wu, X., DeTone, D., Frost, D., Shen, T., Xie, C., Yang, N., Engel, J., Newcombe, R., Zhao, H., Straub, J.: Sonata: Self-supervised learning of reliable point representations. In: *Proceedings of the Computer Vision and Pattern Recognition Conference*. pp. 22193–22204 (2025)
29. Wu, X., Jiang, L., Wang, P.S., Liu, Z., Liu, X., Qiao, Y., Ouyang, W., He, T., Zhao, H.: Point transformer v3: Simpler faster stronger. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. pp. 4840–4851 (2024)
30. Wu, X., Lao, Y., Jiang, L., Liu, X., Zhao, H.: Point transformer v2: Grouped vector attention and partition-based pooling. *Advances in Neural Information Processing Systems* **35**, 33330–33342 (2022)
31. Yue, Y., Mahadevan, S., Schult, J., Engelmann, F., Leibe, B., Schindler, K., Kontogianni, T.: Agile3d: Attention guided interactive multi-object 3d segmentation. *arXiv preprint arXiv:2306.00977* (2023)
32. Yue, Y., Robert, D., Wang, J., Hong, S., Wegner, J.D., Rupprecht, C., Schindler, K.: Litept: Lighter yet stronger point transformer. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 24173–24184 (2026)
33. Zhang, M., Levine, S., Finn, C.: Memo: Test time robustness via adaptation and augmentation. *Advances in neural information processing systems* **35**, 38629–38642 (2022)
34. Zhang, P., Wu, T., Sun, J., Li, W., Su, Z.: A general framework for interactive semantic segmentation refinement of point clouds. *Engineering Applications of Artificial Intelligence* **163**, 112919 (2026)
35. Zhang, Y., Wu, X., Lao, Y., Wang, C., Tian, Z., Wang, N., Zhao, H.: Concerto: Joint 2d-3d self-supervised learning emerges spatial representations. *arXiv preprint arXiv:2510.23607* (2025)
36. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: *Proceedings of the IEEE/CVF international conference on computer vision*. pp. 16259–16268 (2021)
37. Zhou, Y., Gu, J., Chiang, T.Y., Xiang, F., Su, H.: Point-sam: Promptable 3d segmentation model for point clouds. *arXiv preprint arXiv:2406.17741* (2024)