CompetitorFormer: Mitigating Query Conflicts for 3D Instance Segmentation via Competitive Strategy

Supplementary Material

6. Experimental Setup

6.1. Datasets

This section provides detailed specifications for the datasets used to generate the results presented in our main experiments (Section 4). Our selection of these four benchmarks is strategic, designed to comprehensively validate the effectiveness of CompetitorFormer across diverse scenarios. Specifically, ScanNetV2 and S3DIS serve as standard benchmarks for comparison. We chose ScanNet200 to explicitly test our model's performance on fine-grained categories and long-tail distributions, where we argue that resolving competition is particularly beneficial. Furthermore, the high-fidelity scenes in ScanNet++V2 allow us to demonstrate the robustness and scalability of our approach.

ScanNetV2 [4]. This dataset is a standard and large-scale benchmark for indoor 3D scene understanding. It contains 1,613 reconstructed scenes with dense annotations for 20 semantic categories. We follow the official split of 1,201 scenes for training, 312 for validation, and 100 for testing.

S3DIS [1]. The Stanford Large-Scale 3D Indoor Spaces dataset contains 272 scenes from six large-scale areas across three different buildings. It is annotated with 13 semantic categories. For evaluation, we adhere to two standard protocols: training on five areas and testing on Area 5, as well as 6-fold cross-validation where each area is iteratively used for testing.

ScanNet200 [27]. As a challenging extension of ScanNetV2, this dataset introduces a significantly larger set of 200 fine-grained object categories. Its long-tail distribution of instances makes it particularly well-suited for evaluating a model's ability to segment rare or less common objects. The data splits are identical to those of ScanNetV2.

ScanNet++V2 [38]. This is a recent, high-fidelity benchmark that offers substantial improvements over the original ScanNetV2. It features 1,006 indoor scenes captured with high-quality LiDAR scanners, resulting in superior geometric accuracy and denser point clouds. The dataset provides precise annotations for 84 instance categories and presents significant challenges due to its complex, cluttered scenes and a pronounced long-tail category distribution.

6.2. Implementation Details

Our framework is implemented in PyTorch. We build CompetitorFormer upon the strong public codebase of Relation3D [22], ensuring a fair and reproducible comparison. The following details describe the common setup for our experiments across all datasets, with specific variations noted where applicable.

Model Architecture. The input to our model for each point is a 9-dimensional feature vector, comprising its 3D coordinates (XYZ), color information (RGB), and surface normals. We employ a Sparse UNet-based [3] backbone for efficient point cloud feature extraction. Our Transformer decoder consists of 6 layers with a model dimension (d_{model}) of 256 and 8 parallel attention heads per layer. The number of learnable instance queries is adapted to the complexity of each dataset: we use **400** queries for ScanNetV2, **400** for S3DIS, and **800** for the more fine-grained ScanNet200 and ScanNet++V2 benchmarks. We use Fourier features for the absolute positional encoding with a temperature value of 10,000. Our proposed Query Competition Layer , Relative Relationship Encoding, and Rank Cross-Attention modules are integrated into this decoder architecture.

Training and Optimization. We train all models for a total of 512 epochs on a single NVIDIA A100-80G GPU. We use the AdamW [21] optimizer with a base learning rate of 2e-4 and a weight decay of 0.05. The learning rate is managed by a polynomial decay scheduler, which includes a linear warmup phase for the first 5 epochs. For training, we use a batch size of 6 and 8 workers for data loading to maximize GPU utilization.

Loss Configuration. Our loss function is a weighted sum of several components, following the baseline setup. The loss weights for mask prediction (Dice and Focal), classification, and auxiliary losses are set to [0.5, 1.0, 1.0, 0.5, 0.5]. The cost weights for the bipartite matching follow a similar distribution. The weight for the "no object" class is set to 0.1 to maintain a balance between positive and negative samples.

Data Preprocessing. For all ScanNet-based datasets and S3DIS, we voxelize the input point clouds with a voxel size of 0.02m. During training, we apply standard data augmentation techniques, including random rotation, scaling, and elastic distortion, to improve model robustness.

Inference. During evaluation, we do not use any data augmentations. For each scene, instance mask predictions are generated from the decoder outputs. We apply a score threshold of 0.0 and a minimum point threshold of 100 to filter out low-quality or empty predictions. The maximum number of predicted instances per scene is also tailored to the dataset: we report results based on the top **1200** instances for ScanNetV2 and S3DIS, and the top **3200** instances for the denser ScanNet200 and ScanNet++V2.

Table 4. Comprehensive analysis of using Top-k competitors on the ScanNetV2 validation set. The Top-1 strategy provides the best combination of accuracy, efficiency, and optimization stability (query utilization).

Method	Performance			Time (ms) ↓	Query Utilization (%)				
	mAP ↑	mAP_{50} ↑	mAP_{25}↑		L1 ↑	L2 ↑	L3 ↑	L4 ↑	L5 ↑
Relation3D [22]	62.2	80.2	86.9	**1176**	67.0	83.1	88.0	90.9	94.6
Top-1 (Ours)	**63.4**	**81.6**	**88.4**	1289	**77.5**	**88.5**	**91.4**	**92.7**	**95.0**
Top-3	63.0	80.9	87.7	1412	75.1	84.4	90.7	92.2	94.8
Top-5	62.3	80.6	87.3	1653	70.2	83.6	88.2	91.6	94.8

7. Analysis of Competitor Selection Strategy

7.1. Objective and Methodology

A core design choice in our Query Competition Layer is the selection of a single primary competitor (Top-1). This ablation study is designed to rigorously validate this choice by answering a key question: Does considering more competitors (Top-k) provide a richer signal that improves performance, or does it introduce counterproductive noise?

To implement the Top-k strategy, we extend our QCL module's embedding structure. For each query, we now utilize **one dominant embedding** and **k distinct, rank-aware subordinate embeddings**. The process is as follows: for a given query, we first identify its top-k competitors, sorted by mask IoU. The competitive landscape feature is then constructed by aggregating k individual interactions. For the m-th competitor in the sorted list (where $m \in \{1, ..., k\}$), the query's single dominant embedding is paired with the dedicated m-th subordinate embedding. These k feature pairs are processed and aggregated to form the final competitive landscape, which is then fused into the query's representation. Our analysis compares this approach for $k = 1, 3, 5$ against the Relation3D baseline.

7.2. Defining the Query Utilization Metric

To quantitatively measure optimization stability, we adopt the query utilization metric introduced in MP-Former [41]. The calculation involves two steps. First, for each decoder layer i, we perform bipartite matching between the N queries and the O ground-truth instances. The result is stored in an index vector $V^i = \{V_0^i, V_1^i, ..., V_{N-1}^i\}$, where each element V_n^i is defined as:

$$V_n^i = \begin{cases} o, & \text{if } Q_n^i \text{ matches } GT_o, \\ -1, & \text{otherwise.} \end{cases} \tag{10}$$

This vector effectively records which ground-truth instance is assigned to each query at a specific layer.

Second, the query utilization of layer i, denoted Util^i, is calculated as the proportion of ground-truth instances that are matched by the same query in layer i as in the final layer l. This is formally expressed as:

$$\text{Util}^i = \frac{1}{O} \sum_{n=0}^{N-1} \mathbb{1}\{V_n^i = V_n^l \wedge V_n^i \neq -1\}, \tag{11}$$

where $\mathbb{1}(\cdot)$ is the indicator function. The term $V_n^i = V_n^l$ checks for a stable match assignment for query n between the current and final layers. The condition $V_n^i \neq -1$ ensures we only count queries that are matched to a valid ground-truth instance. In essence, a high utilization rate signifies that queries lock onto their final targets early, indicating stable and efficient optimization.

7.3. Results and Analysis

The comprehensive results in Table 4 provide robust, multi-faceted evidence for our design choice.

Performance and Efficiency Trade-off. The results first highlight a clear accuracy-efficiency trade-off. Our Top-1 strategy achieves the highest mAP (63.4), delivering a substantial +1.2 mAP gain over the Relation3D baseline. The Top-3 approach degrades performance to 63.0 mAP, and the Top-5 model falls to 62.3 mAP—nearly erasing the gains over the baseline—while incurring significant computational costs (1412ms and 1653ms, respectively). This establishes that a targeted Top-1 approach is the most effective and efficient design.

Causal Explanation via Query Utilization. The query utilization metrics provide a compelling causal explanation for this performance trend. The most dramatic impact is observed in the earliest decoder layer. Our Top-1 model achieves a remarkable **77.5%** utilization in Layer 1, a massive **+10.5%** improvement over the baseline's 67.0%. This indicates that our explicit, targeted competition mechanism allows queries to lock onto their final ground-truth targets much earlier in the refinement process, leading to a more stable and efficient optimization trajectory.

Crucially, the data also demonstrates why adding more competitors is counterproductive. Both the Top-3 (75.1%) and Top-5 (70.2%) strategies exhibit lower early-layer utilization than our Top-1 model. This suggests that providing a query with multiple competitor signals introduces ambiguity rather than useful information. The query's optimiza-

Method	Competitor-Former	ScanNetV2 Val			S3DIS Area 5		
		mAP	mAP_{50}	mAP_{25}	mAP	mAP_{50}	mAP_{25}
Mask3D [28]		55.2	73.7	83.5	57.8	71.9	77.2
Mask3D [28]	✓	56.0 (+0.8)	73.8 (+0.1)	83.9 (+0.4)	58.0 (+0.2)	72.0 (+0.1)	77.2 (-)
SPFormer [30]◁		56.3	73.9	82.9	50.2	66.8	75.3
SPFormer [30]	✓	59.0 (+2.7)	77.1 (+3.2)	85.1 (+2.2)	52.6 (+2.4)	69.2 (+2.4)	78.1 (+2.8)
MAFT [16]◁		58.3	75.9	84.5	47.1	62.8	71.2
MAFT [16]	✓	59.8 (+1.5)	77.4 (+1.5)	85.4 (+0.9)	48.4 (+1.3)	67.1 (+4.3)	74.8 (+3.6)
OneFormer3D [14]◁		59.3	78.1	86.4	58.7	72.0	78.5
OneFormer3D [14]	✓	59.5 (+0.2)	78.7 (+0.6)	87.0 (+0.6)	59.5 (+0.8)	72.9 (+0.9)	79.4 (+0.9)

Table 5. Performance of CompetitorFormer when integrated into various Transformer-based frameworks. The checkmark (✓) indicates the model is enhanced with our proposed modules. We report consistent improvements across all architectures on both the ScanNetV2 validation set and the S3DIS Area 5 benchmark. Gains are shown in parentheses. ◁ denotes the reproduced result.

tion target becomes less stable as it hesitates between conflicting signals, disrupting the learning process and explaining the subsequent drop in mAP.

This analysis confirms a core principle of our work: effective competition modeling hinges on decisiveness. The superior performance of the Top-1 strategy stems from its ability to provide a clean, unambiguous, and powerful signal that stabilizes the optimization process. In contrast, the Top-k approaches introduce noise and ambiguity, which destabilizes query assignments and ultimately degrades performance. This insight validates our design and underscores the value of targeted, rather than distributed, relationship modeling in query-based architectures.

8. Generalizability Across Different Frameworks

A key advantage of our proposed method is its modularity and architectural independence. While the main paper presents results based on a single strong baseline (Relation3D), we conducted further experiments to validate that CompetitorFormer is a general-purpose module capable of enhancing a variety of Transformer-based architectures. To this end, we integrated our core components, QCL, RRE, and RCA, into four prominent and publicly available frameworks: Mask3D [28], SPFormer [30], MAFT [16], and OneFormer3D [14]. The integration was straightforward, requiring minimal modifications to the baseline decoders.

The results of these experiments on the ScanNetV2 validation set and the S3DIS Area 5 benchmark are summarized in Table 5. The table clearly demonstrates that integrating CompetitorFormer provides consistent performance gains across all tested frameworks and on both datasets.

As shown in the table, our method acts as a reliable performance booster. For instance, on SPFormer and MAFT, which are superpoint-based methods, CompetitorFormer

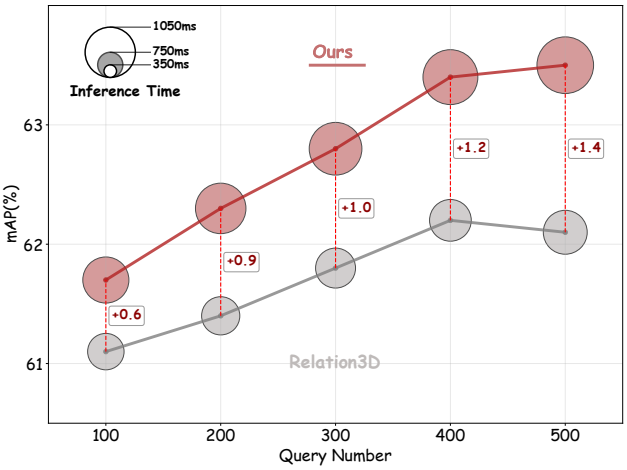

Figure 7. **Analysis of mAP and inference time versus the number of queries on ScanNetV2.** Circle size is proportional to the inference time per scene. Our method (Ours) not only consistently outperforms the baseline (Relation3D), but the performance gap also widens as the number of queries grows. This demonstrates that CompetitorFormer is particularly effective at mitigating the negative effects of intensified inter-query competition in high-query regimes.

delivers substantial improvements, boosting mAP by up to +2.7 on ScanNetV2 and +4.3 mAP_{50} on S3DIS. On voxel-based methods like Mask3D and OneFormer3D, the gains are also consistent, demonstrating the versatility of our competition modeling. The magnitude of the improvement varies, which is expected as different baselines may have their own implicit mechanisms for handling query interactions. However, the consistent positive trend across these diverse architectures strongly supports our claim that explicitly modeling the competitive landscape is a fundamental

Method	mAP	mAP_{50}
Relation3D [22]	62.2	80.2
Relation3D + EASE-DETR [7]	62.4	80.5
Ours (CompetitorFormer)	**63.4**	**81.6**

Table 6. Comparison with the Relation3D integrated with EASE.

and broadly applicable strategy for improving query-based 3D instance segmentation.

9. Impact of Query Count on Performance and Competition.

To investigate the relationship between the number of instance queries and the severity of inter-query competition, we evaluated our method and the Relation3D baseline with varying numbers of queries, from 100 to 500. The results, presented in Figure 7, provide strong evidence for our central hypothesis.

First, we observe the baseline model's performance. The mAP initially improves as the query count increases from 100 to 400. This is expected, as more queries provide greater capacity to capture all object instances in a scene. However, performance begins to degrade when the query count is increased to 500. This decline is consistent with our hypothesis that a surplus of queries intensifies destructive competition, leading to fragmented predictions that harm overall accuracy. The model becomes unable to effectively manage the redundant queries.

In contrast, CompetitorFormer demonstrates a more robust performance trend. Our method consistently outperforms the baseline across all query counts. Crucially, the performance gap between our method and the baseline widens as the number of queries increases, growing from **+0.6** mAP at 100 queries to **+1.4** mAP at 500 queries. This trend strongly indicates that our explicit competition modeling is not just a general improvement, but a targeted solution that becomes increasingly beneficial as the root problem, inter-query competition, becomes more severe.

Finally, this analysis highlights the trade-off between accuracy and computational cost. As indicated by the marker size in Figure 7, inference time scales with the number of queries. Our method allows for the use of a larger query set to achieve higher accuracy without the performance degradation seen in the baseline, offering a better accuracy-cost trade-off. This makes our approach more scalable and effective for complex scenes that may benefit from a larger number of instance queries.

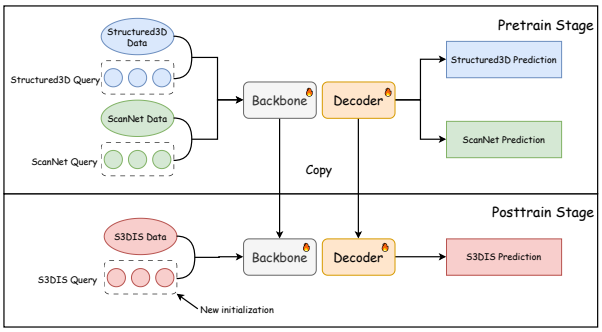

Figure 8. An overview of our two-stage training strategy for S3DIS. The model is first pre-trained on Structured3D and ScanNet. The learned backbone and decoder weights are then transferred and fine-tuned on S3DIS with newly initialized queries, enabling an efficient superpoint-based pipeline.

Method	Pipeline	mAP_{50}	mAP_{25}
Mask3D [28]	Voxel-based	71.9	77.2
OneFormer3D [14]	Voxel-based	72.0	-
Relation3D [22]	Voxel-based	72.5	78.5
IKNE [26]	Voxel-based	73.0	-
Ours	**Superpoint-based**	**73.8**	**80.6**

Table 7. Performance comparison on the S3DIS Area 5 benchmark. Our superpoint-based adaptation of OneFormer3D achieves state-of-the-art results, highlighting the effectiveness of our pre-training strategy and efficient pipeline.

10. Comparison with 2D Method (EASE-DETR)

We compare our method with an implicit competition model, adapting the core mechanism of EASE-DETR [7] to our 3D baseline. As shown in Table 6, EASE-DETR provides a +0.2 mAP gain, which is modest compared to the +1.2 mAP from our method. EASE-DETR employs a global scalar bias applied uniformly across all query relationships. In contrast, our approach uses paired, role-aware embeddings (Eq. 3) that are directly fused into query features based on explicit competitor identification. This targeted design enables more decisive conflict resolution for the high-overlap spatial conflicts characteristic of dense 3D point cloud scenes, where explicit pairing proves more effective than distributed global adjustments.

11. Superpoint-based Pre-training Strategy for S3DIS

A significant practical challenge in applying Transformer-based methods to the S3DIS dataset lies in its data representation. Prominent open-source frameworks, such as

Mask3D and OneFormer3D, typically rely on a voxel-based pipeline. However, the large scale and varying point densities of S3DIS scenes result in an exceptionally high number of voxels, creating substantial computational and memory bottlenecks that complicate the training process. To overcome this limitation, we propose an efficient and effective superpoint-based pre-training strategy, adapting the powerful OneFormer3D architecture.

Our approach, illustrated in Figure 8, consists of a two-stage pre-training and fine-tuning process. **In the pre-training stage**, we first build a strong foundational model by jointly training the shared backbone and decoder on two large-scale datasets: Structured3D and ScanNet. This step allows the model to learn a rich and generalizable understanding of 3D geometry and semantics from diverse indoor environments, establishing a robust feature representation. **In the post-training stage**, we transfer the learned weights of the backbone and decoder to the S3DIS task. Crucially, we discard the dataset-specific queries from the pre-training phase and introduce a new, randomly initialized set of queries tailored for S3DIS. The entire model is then fine-tuned on the S3DIS dataset using our efficient superpoint-based pipeline, which is better suited to the sparse nature of point clouds.

This strategy proves highly effective, as demonstrated in Table 7. Our superpoint-based adaptation not only alleviates the computational burden associated with voxelization but also achieves new state-of-the-art performance on the S3DIS Area 5 benchmark. Specifically, our method surpasses previous voxel-based approaches on both mAP_{50} and mAP_{25} metrics. This result strongly suggests that the combination of robust, multi-dataset pre-training and an efficient, representation-aware fine-tuning pipeline is a superior approach for achieving top performance on the S3DIS.

12. Additional Qualitative Analysis

To further illustrate the practical benefits of our approach, we provide extensive qualitative comparisons on both the ScanNetV2 and S3DIS datasets. These visualizations serve as direct visual evidence for the claims made in our main paper, highlighting how explicit competition modeling leads to more coherent and accurate instance segmentation, particularly in complex and cluttered scenes.

As shown in Figure 9, our method consistently produces superior results on ScanNetV2. In cluttered scenes with many nearby objects (e.g., Scene 1 and Scene 2), the baseline model (Relation3D) often produces fragmented or incorrectly merged instances. This is a direct symptom of unresolved inter-query competition. In contrast, our model correctly separates adjacent chairs and produces complete masks for beds and cabinets. This demonstrates that by resolving conflicts, our method maintains instance integrity and accurately delineates object boundaries even in close proximity.

Method	mAP (%)	mAP_{50} (%)
OneFormer3D [14]	59.3	78.1
OneFormer3D + EASE-DETR [7]	59.5	78.7
OneFormer3D + Ours (Adapted)	**59.6**	**78.9**

Table 8. Adapting CompetitorFormer to an adaptive query model.

Similar improvements are observed on the S3DIS dataset, as shown in Figure 10. The baseline model struggles with large structural elements and repetitive objects, often merging distinct wall panels or fragmenting door frames (e.g., Scene 1 and Scene 4). Our method, by establishing a clear competitive hierarchy among queries, successfully segments these challenging structures. It correctly separates individual objects within repetitive patterns and generates complete, coherent masks. These results across diverse scenes and datasets visually confirm that our approach effectively mitigates the negative effects of inter-query competition, leading to more robust and precise 3D instance segmentation.

13. Limitation

While CompetitorFormer demonstrates significant improvements, we identify two primary areas for future work: computational scalability and architectural dependency.

First, the explicit, pairwise competitor identification at the core of our method has a computational complexity of $O(N^2)$ with respect to the number of queries, N. Although this overhead is manageable on current benchmarks it presents a scalability challenge for dense scenes or real-time applications that may require thousands of queries. Future research could explore more efficient competitor identification schemes, such as sparse attention mechanisms or approximate nearest-neighbor search, to mitigate this quadratic scaling.

Second, our method's effectiveness is tied to the fixed-query architecture. The modest gains on dynamic query systems like OneFormer3D (+0.3 mAP, shown in Table 8) highlight this dependency. Our model is designed around competition landscape, where stable query identities allow for progressive refinement and the tracking of competitive relationships across decoder layers. Dynamic query systems, by their nature, disrupt this temporal consistency. This suggests that a different philosophy of competition modeling may be required for such architectures, one based on transient, *population-level dynamics* rather than persistent individual pairings. Developing such a framework would be a valuable and non-trivial extension, potentially broadening the applicability of competition modeling to a wider range of modern Transformer architectures.

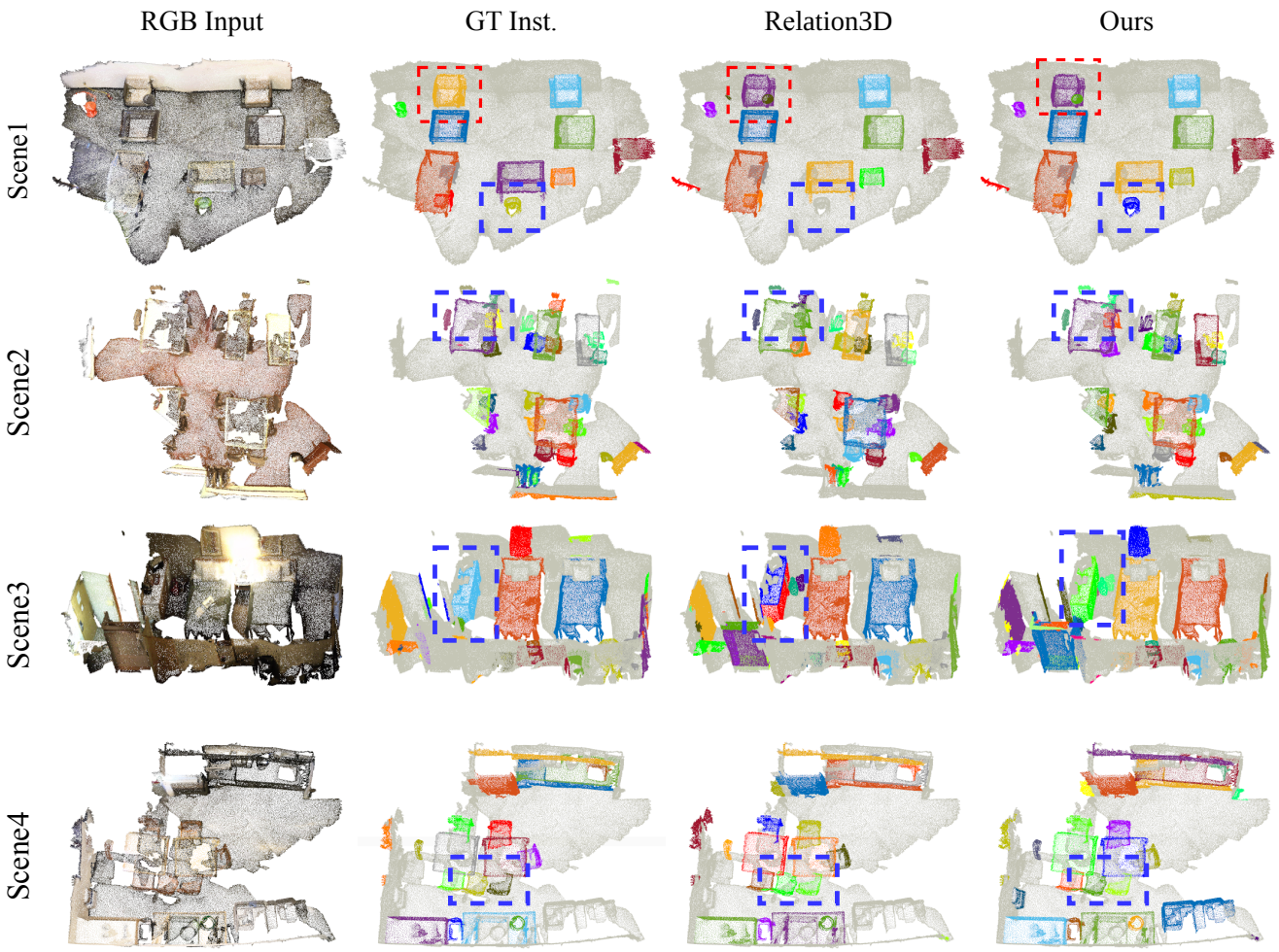

Figure 9. Qualitative comparisons of 3D Instance Segmentation performance on the ScanNetV2 [4] validation set. We visualize Instance masks of Relation3D [22] and ours based on both architecture with Ground Truth (GT) masks. The critical differences are highlighted using colored boxes for better comparison.

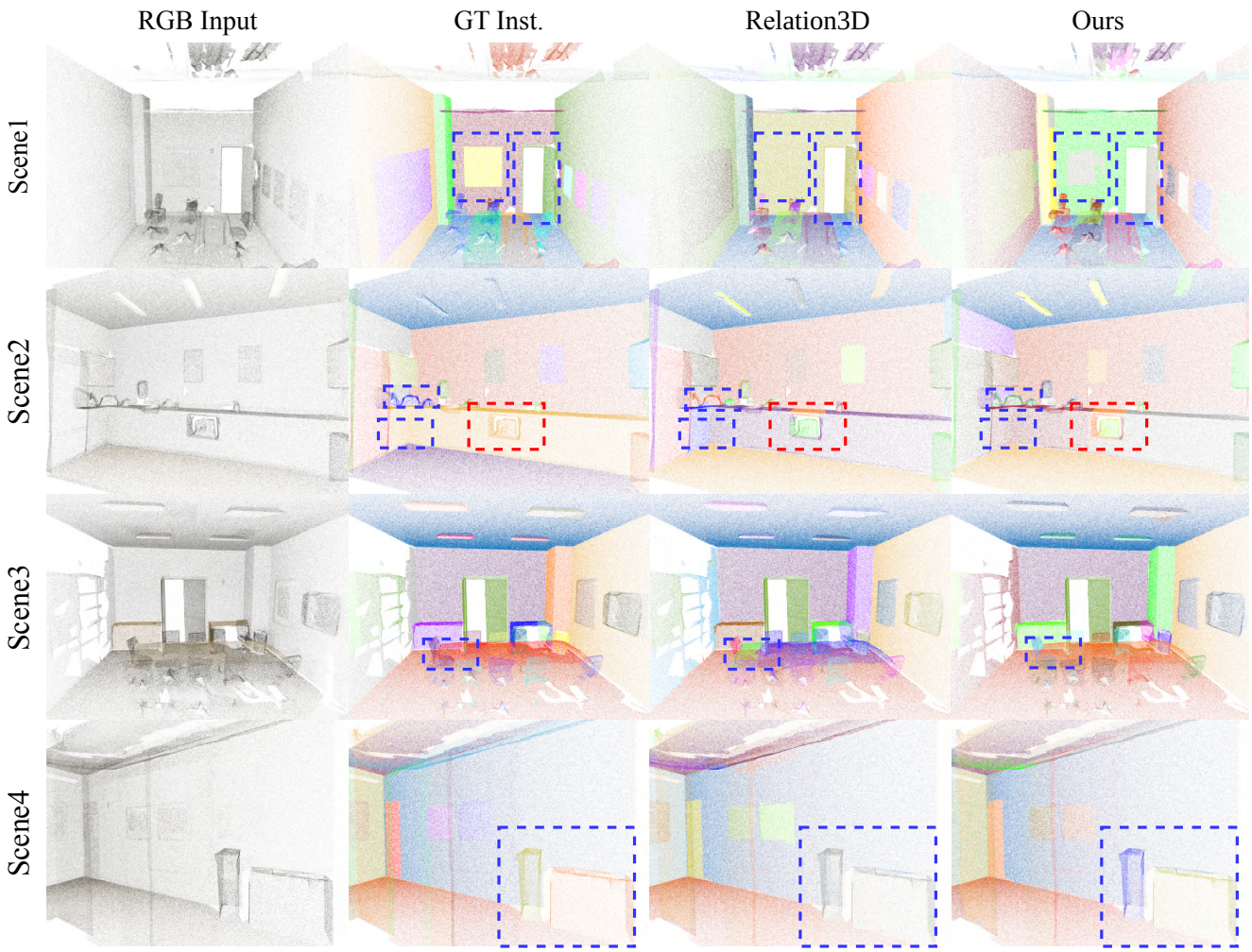

Figure 10. Qualitative comparisons of 3D Instance Segmentation performance on the S3DIS [4] validation set. We visualize Instance masks of Relation3D [22] and ours based on both architecture with Ground Truth (GT) masks. The critical differences are highlighted using colored boxes for better comparison.