S1. CESS Module Architecture

The architecture of CESS is shown in Fig. 6. The scenes with and without applied SCA first pass through different random similarity transformations, such as rotation, scaling and translation. The scene with SCA will be inputted to the student model; whereas, the scene without SCA will be inputted to the teacher model. The outputs of both models are then normalised before computing the loss. We define the pipeline taking the scene with SCA as the supervised branch and the other pipeline as the supervising branch. In this case, we hypothesise that the output in the supervising branch has better quality since the prediction of the teacher model is more stable by nature and the unaugmented input incurs less context break.

S2. Formal Definition of P_G in GCAS

We introduce a probabilistic framework designed to assess the compatibility of a potential class addition within a given scene. Specifically, we deploy a probability estimator, denoted as P_G, to evaluate the likelihood that an object from the proposed class would seamlessly integrate into the existing scene configuration, considering the current assortment of objects. The estimator P_G is formulated to accept inputs comprising object counts within the scene, symbolized as $cnt_1, cnt_2, ..., cnt_{N_K}$, where each cnt_i quantifies the count of objects from class i, as determined by a previously described clustering algorithm. This process yields a probability vector, $P_G(cnt_1, cnt_2, ..., cnt_{N_K}) \in [0, 1]^{N_K}$, with each vector element signifying the likelihood of an additional object from each class fitting into the scene.

To instantiate P_G, we employ a multinomial Naive Bayes classifier, leveraging its efficacy in modelling the probability of class occurrences. The classifier is trained on data reflective of the class distribution observed in the training dataset splits. For each scene representation, we generate a collection of training samples, $(Cnt_i, i) \mid i \in K_s$, wherein K_s is the set of classes with concrete shapes observed within the scene, and Cnt_i represents the adjusted object count distribution for the scene, diminished by one for the considered class i, effectively rendering $Cnt_i = (cnt_1, cnt_2, ..., cnt_{i-1}, cnt_i - 1, ..., cnt_{N_K})$. Noted that, since the class of each object in the target dataset is deemed unknown in the UDA setting, counts are produced based on pseudo-labels for target datasets. This methodology facilitates a nuanced understanding of class compatibility within varied scene configurations, underpinning our approach to scene augmentation.

S3. Cluster Arrangement Algorithms

Algorithms 1 and 2 describe the details of our algorithms for in-room and on-wall cluster centre search here. For in-room clusters centre search, we verify whether a candidate is valid by examining whether placing the new clusters at the position will introduce a collision. For on-well clusters centre search, we detect misalignment of the wall and the window instead. Misalignment is detected by checking the maximum distance between the cluster points and their nearest neighbours in the scene. If this distance is below a threshold d, the candidate centre point and normal form a valid pair $(X_{\text{valid}}, \hat{\mathcal{N}}_{\text{valid}})$.

```
Algorithm 1: In-room clusters centre search
Input  : $X_{\text{scene}}, X_{\text{clr}}, N_p$
Obtain $B_{\text{scene}} \leftarrow$ get2DBoundingBox($X_{\text{scene}}$);
Obtain $z_{\text{floor}} \leftarrow$ getFloorHeight($X_{\text{scene}}$);
for $1, 2, \ldots, N_p$ do
    $(x, y) \leftarrow$ randomXY($B_{\text{scene}}$);
    $X_{\text{cand}} \leftarrow (x, y, z_{\text{floor}})$;
    move($X_{\text{clr}}, X_{\text{cand}}$);
    if  not collide($X_{clr}, X_{scene}$) then
        $X_{\text{valid}} \leftarrow X_{\text{cand}}$;
        break
    end
end
Output: $X_{\text{valid}}$
```

```
Algorithm 2: On-wall clusters centre search algorithm
Input  : $X_{\text{wall}}, X_{\text{clr}}, N_q, d$
Set $X_{\text{valid}}, \hat{\mathcal{N}}_{\text{valid}} \leftarrow$ None, None;
Obtain $\hat{\mathcal{N}}_{\text{wall}} \leftarrow$ estimateNormals($X_{\text{wall}}$);
Obtain $I \leftarrow$ randomInt($1, |\hat{\mathcal{N}}|$, size $= N_q$);
for  i in I do
    $X_{\text{cand}}, \hat{\mathcal{N}}_{\text{cand}} \leftarrow X^{(i)}_{\text{wall}}, \hat{\mathcal{N}}^{(i)}_{\text{wall}}$;
    move($X_{\text{clr}}, X_{\text{cand}}$);
    $\hat{\mathcal{N}}_{\text{clr}} \leftarrow$ getLargestFaceNormal($X_{\text{clr}}$);
    angle $\leftarrow$ arccos $(\hat{\mathcal{N}}_{\text{clr}} \cdot \hat{\mathcal{N}}_{\text{cand}})$;
    rotate($X_{\text{clr}}$, angle);
    $d' \leftarrow X_{\text{clr}}$.distNearestNeighbor($X_{\text{wall}}$);
    if $d' \leq d$ then
        $X_{\text{valid}}, \hat{\mathcal{N}}_{\text{valid}} \leftarrow X_{\text{cand}}, \hat{\mathcal{N}}_{\text{cand}}$;
        break
    end
end
Output: $X_{\text{valid}}, \hat{\mathcal{N}}_{\text{valid}}$
```

S4. Visualisation of Augmented Scenes

Figure 7 illustrates the visualisation of in-domain and cross-domain augmentations. During the pre-training phase, in-domain augmentation is applied, where both the

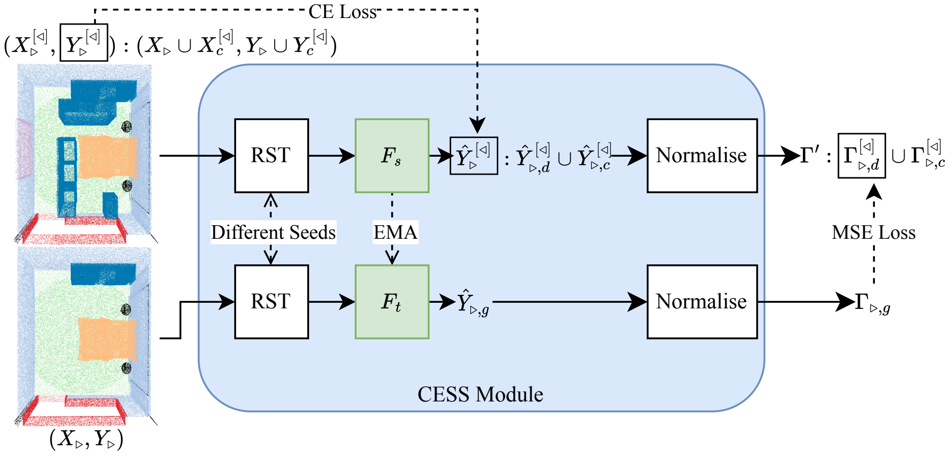

CE Loss
$(X_\triangleright^{[\triangleleft]}, \boxed{Y_\triangleright^{[\triangleleft]}}) : (X_\triangleright \cup X_c^{[\triangleleft]}, Y_\triangleright \cup Y_c^{[\triangleleft]})$
RST
F_s
$\boxed{\hat{Y}_\triangleright^{[\triangleleft]}} : \hat{Y}_{\triangleright,d}^{[\triangleleft]} \cup \hat{Y}_{\triangleright,c}^{[\triangleleft]}$
Normalise
$\Gamma' : \boxed{\Gamma_{\triangleright,d}^{[\triangleleft]}} \cup \Gamma_{\triangleright,c}^{[\triangleleft]}$
Different Seeds
EMA
MSE Loss
RST
F_t
$\hat{Y}_{\triangleright,g}$
Normalise
$\Gamma_{\triangleright,g}$
CESS Module
$(X_\triangleright, Y_\triangleright)$

Figure 6. The architecture of CESS. RST stands for random similarity transformations. Similar to Fig. 4 in the main paper, we use $\triangleleft$ and $\triangleright$ for generality. $X_c^{\triangleleft}$ are point cloud added to $X_\triangleright$ in the SCA module (i.e., objects from $\triangleleft$ domain). $Y_c^{\triangleleft}$ and $\hat{Y}_c^{\triangleleft}$ are the label and the prediction of $X_c^{\triangleleft}$. $\hat{Y}_{\triangleright,d}^{[\triangleleft]}$ and $\hat{Y}_{\triangleright,g}^{[\triangleleft]}$ are the prediction of $X_\triangleright$ from the augmented and unaugmented scenes, respectively.

object bank and the scene originate from the same domain, resulting in visually similar object styles but enhancing the diversity of room layouts. In contrast, during the self-training phase, cross-domain augmentation is employed, where the object bank and scene come from different domains (e.g., the object bank is generated from 3D-FRONT while the scene is sourced from ScanNet, or vice versa). This leads to more heterogeneous objects in the augmented scene, effectively creating an intermediate domain that bridges the gap between the source and target domains.

S5. Implementation Details

In our implementation, we utilize the same backbone as in DODA [10] for a fair comparison, a sparse-convolution-based U-Net backbone [6, 17] which serves as our semantic parser F. We follow the protocols outlined in DODA [10] including dataset preparation and splitting for validation and hyperparameter tuning.

S5.1. Module Hyper-parameters

In the SCA module, we employ DBSCAN [11] for 3D point cloud clustering. Note that this can be any clustering algorithm. We chose DBSCAN simply because it has been implemented by the Python package Open3D [85], which we used to process the point cloud in our experiments. The density parameter is set to 0.1, and the minimum number of points required for a cluster to be considered as a vocabulary is set to 50 in our experiments. For each scene, we generate 10 additional clusters (N_c') using the SCA module. When arranging in-room clusters, we sample 10 candidate poses (N_p) to prevent over-crowding the training scenes. However, for on-wall clusters, we sample 20 candidate poses (N_q) since the estimated normals of walls tend to be noisy, leading to a higher likelihood of invalid candidate poses. To determine the validity of candidate poses for on-wall clusters, we use a threshold of 0.3 meters (d).

In the CESS module, α is set to 0.999 and the pseudo labels update frequency N_u to 5 in our experiments.

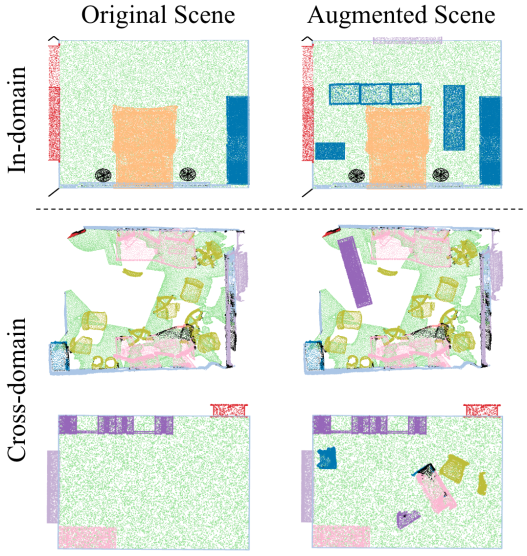


Figure 7. Examples of scenes augmented with SCA.

S5.2. Optimisation Hyper-parameters

In terms of the weights for our objective terms, we use λ_s =1.0, λ_t =0.5, and λ_c =5.0. Note that λ_c is assigned a greater value compared to the others since the distribution of the normalized output from F is relatively confined, resulting in smaller values for $\mathcal{L}_c$ compared to $\mathcal{L}_s$ and $\mathcal{L}_t$. For the optimisor, we use SGD with 0.0001 for weight decay and 0.9 for momentum.

S6. Discussion on Failure Cases

As discussed in the main paper, we observe that on the 3D-FRONT → ScanNet setting, the performance on the wall class becomes worse after adding the CESS module. We attribute this change to the inaccurate pseudo-labels and the characteristics of the CESS module.

In the real scenes, due to the errors in reconstruction, walls are often uneven with fluctuations. However, in the synthetic scenes, the walls are flat and connected to the floor. Uneven patterns that adhere to the wall and are connected to the floor in the synthetic scene only occur when doors are presented. Therefore, when producing the pseudo label of the real data by predicting with the model pre-trained on the synthetic data, uneven walls are often pseudo-labelled as doors (see Fig 8).

Points that occur before applying SCA will have a larger impact on the loss with the CESS module. This is because only those points will be supervised by both the (pseudo-)labels and the supervising branch; whereas, the points added to the scene via SCA will only be supervised by the (pseudo-)labels. This characteristic will further amplify the negative impact of the inaccurate pseudo labels, which cause the performance to drop on walls after adopting CESS.

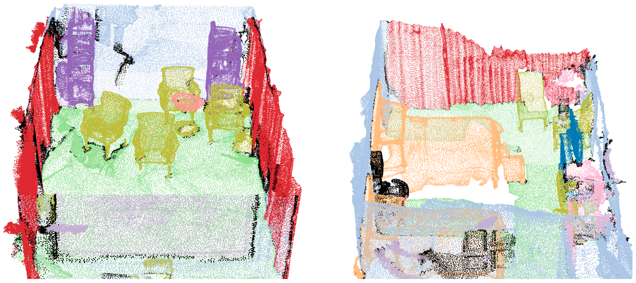
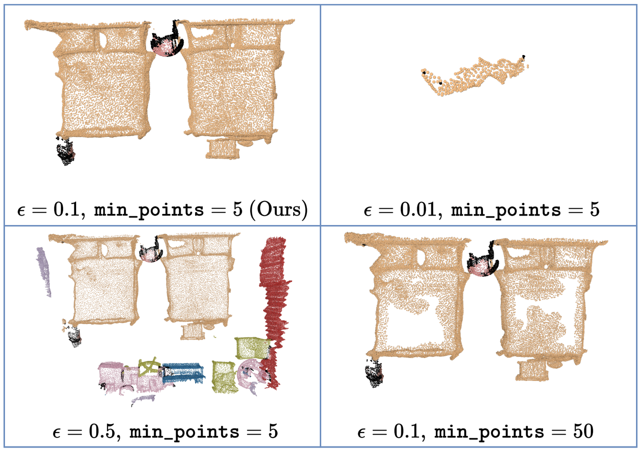


Figure 8. Predictions on ScanNet scenes. Points predicted as walls and doors are coloured in blue and red, respectively. Walls are often mislabelled as doors since the door frame in the real data is not significant due to the inherent noise of the wall.

S7. Efficiency of the Method

We conduct experiments on a computing platform with NVIDIA A10 (25GB VRAM/GPU), 30 vCPUs, 200 GB RAM, and 1.4 TB SSD. The inference efficiency stays the same as DODA [10] (86.6 ms/sample) since our modules are only enabled during training. Training with and without SCA takes 604.8 and 450.9 seconds per epoch on average, respectively.



$\epsilon = 0.1$, $\texttt{min\_points} = 5$ (Ours)

$\epsilon = 0.01$, $\texttt{min\_points} = 5$

$\epsilon = 0.5$, $\texttt{min\_points} = 5$

$\epsilon = 0.1$, $\texttt{min\_points} = 50$

Figure 9. Clusters of a bed constructed with different DBSCAN parameters.

S8. Influence of DBSCAN

Although DBSACN in our experiment can be replaced by any suitable out-of-box clustering algorithms and our research does not focus on DBSACN, we provide the influence of hyper-parameters of DBSCAN to clustering results to facilitate future research in this community. In Fig. 9, an object is grouped into several clusters with a smaller ϵ=0.05; whereas, furniture in a scene is grouped as one cluster with a larger ϵ=0.5; with large min_points (the minimum number of points to form a cluster), object parts with lower density will be missed during clustering.

S9. Comparison with Other Data-mixing Methods

Mix3D [48] is a data-mixing method that augments a scene by adding the entire point cloud of another scene into the same spatial domain, resulting in overlap between scenes. This approach has been shown to significantly improve segmentation quality. To evaluate the effectiveness of our proposed SCA, we reprint a subset of the results from DODA [10] for direct comparison. Our data-mixing method outperforms Mix3D by 7.17%.

S10. Ablation on point grouping

As discussed in Section 3.1, CINMix [83] employs a different design choice compared to our SCA when grouping points for vocabulary bank construction. To evaluate the

Table 6. Ablation study of data-mixing methods on 3D-FRONT → ScanNet. TACM is the augmentation algorithm proposed in DODA [10].

Method	mIoU
Mix3D [48]	48.62
TACM [10]	51.42
Ours (SCA)	**55.79**

impact of this difference, we conducted an experiment by replacing our Euclidean-based, semantic-agnostic grouping mechanism in the SCA module with their semantic-aware approach. As shown in Table 7, the results demonstrate that preserving local context using our method yields better performance in the task, compared to the potential benefits of increased augmentation diversity from CINMix's semantic-aware grouping.

Table 7. Ablation study of point-grouping methods on 3D-FRONT → ScanNet. Semantic-aware is the strategy employed by CINMix [83]. The experiment was conducted in the pre-train stage.

Method	mIoU
Semantic-aware (CINMix [83])	39.88
Semantic-agnostic (Ours)	**43.48**

S11. Limitations and Future Work

In the context of unsupervised domain adaptation, the present method is constrained by its ability to only segment classes common to both the source and target domains, owing to the absence of labels in the target domain. This limitation impedes the generalizability of the semantic segmentation framework. To address this challenge, future investigations could explore the integration of large language models (LLMs) to endow the segmentation model with the capability for open-vocabulary segmentation, thus enhancing its adaptability across diverse domains.

Regarding the methodology for incorporating windows into walls, one potential issue is that windows might protrude beyond the confines of the wall boundary. Although this phenomenon was not observed in our evaluation of augmented scenes, it remains a conceivable outcome. Preventative measures could include the decrement of the parameter d within the clustering algorithm or the establishment of a distinct distance constraint d_{up} for the upper margin of wall-mounted objects, ensuring their containment within wall boundaries.

This study endeavours to maintain the simplicity and applicability of the proposed augmentation technique for large-scale dataset enhancement. While ensuring the logical coherence of augmented scenes was a priority, the incorporation of improvisational layout augmentations—such as the placement of a lamp atop a table—was deferred, due to the complexities associated with validating the feasibility of new layouts. Nevertheless, this concept presents a promising avenue for future research, with the potential to significantly enrich the diversity and realism of augmented scenes.

S12. More Literature Review

3D Semantic Segmentation Domain Adaptation is important for many applications like autonomous vehicle, semantic mapping and construction site monitoring, with a particular focus on enclosing discrepancies induced by different LiDAR sensors or positions. Some methods [2, 54] employ a hand-crafted approach to achieve domain-invariant representations, such as normalization of the input feature spaces of different mounting positions. Domain mapping [4,34,47,57,58] is another common approach that is often used in dataset-to-dataset applications, in which the labelled source data is usually transformed to appear more like the target data, creating a pseduo-labeled target dataset. Domain-invariant feature learning is done by constructing a common feature representation space for both source and target domains. Similar to UDA, they can also be categorized into divergence minimization [23, 46, 73] and adversarial learning-based approaches [24, 72, 81].

3D Indoor Semantic Segmentation. In the 3D indoor semantic segmentation task, early approaches [9, 62] focus on exploring 3D volumes. Instead of using volumes, point-based models [20, 22, 41, 51, 71] directly learn from unordered point clouds and construct point representation. To address point cloud on a large scale, approaches [29, 33, 53] such as super-points and structured hierarchical data are proposed. Another direction is to circumvent processing 3D data directly. Multiple-view approaches [8, 32, 35] transfer 2D segmentation into 3D, which can potentially scale better to large-scale scenes.

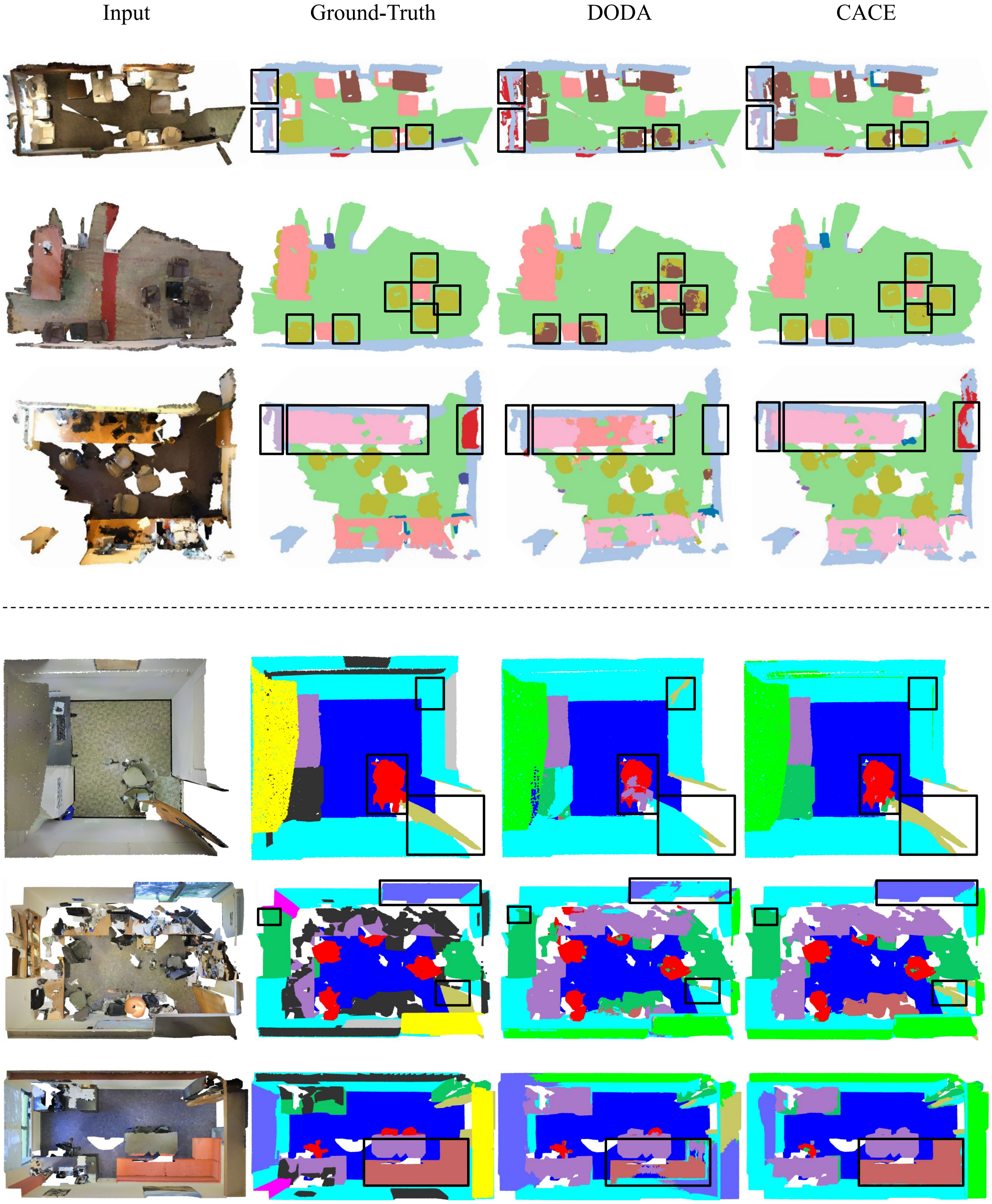


Figure 10. The first three and the last three rows are the qualitative comparisons on FRONT-3D → ScanNet and FRONT-3D → S3DIS, respectively. The bounding boxes highlight the parts that our method significantly outperform the SOTA method DODA [10].