We do appreciate all reviewers' careful reviews, constructive comments, and positive recognitions. We firstly answer common questions and then response each reviewer. All the concerns will be addressed in the final version paper.

Common Questions Response:

[CQ1] Class-wise accuracy. Actually, we observed that the imbalanced training samples do affect the performance of our model and other domain adaptation (DA) models. *Chair*, *Table*, and *Sofa* (easily confusing with *Bed*) cover more than 60% samples in M-to-S scenario which causes the drop of certain classes (*e.g.*, *Bed* and *Sofa*). After re-balancing samples, most classes show satisfactory improvements with our model (seen in the Table below in which each class equally contains 300 samples for training), except for *Lamp*, which indicate the following weakness of our model. 1) Neglect of scale information; 2) When different classes share very similar local structures, our local alignment has the possibility to align similar self-adaptive (SA) nodes across those classes (*e.g.*, large columns contained by both *Lamps* and round *Tables*). We will discuss more details in the final version and explore solutions in our future work.

	Bathtub	Bed	Bookshelf	Cabinet	Chair	Keyboard	Lamp	Laptop	Sofa	Table	Avg
MCD	74.7	28.3	10.7	16.2	75.0	99.2	**83.8**	99.3	5.8	76.3	56.9
Ours	**79.1**	**37.1**	**12.0**	**30.3**	**75.1**	**100.0**	79.2	**99.5**	**10.1**	**77.7**	**60.0**

[CQ2] Discuss previous papers. It is a good suggestion. Our method is similar but different compared with these previous works. [Lai *et al.*, IJRR10] adopts the model trained by Web data to the 3D data scanned by LiDAR for object detection, while it is a supervised fashion which requires labeled target samples for training. In the works of [Wu *et al.*, arXiv18] and [Saleh *et al.*, arXiv19], 3D LiDAR point clouds are firstly projected/transferred into a 2D image and then forwarded to 2D-based conventional algorithms. It is not a general 3D approach which requires specific 3D data and 3D-2D transmission of single view. Our method could directly process point cloud data without any strict limitations.

[CQ3] Open-source. We will definitely release our code and all implementation details after the paper is accepted. Moreover, the well arranged dataset will also be released as a solid benchmark to benefit the whole research community.

Response To Reviewer 1:

[Q1] Comparison with bilateral filtering (BF). Good suggestion. 1) Kernel: BF uses Gaussian Kernel while ours uses a learnable 1×1 convolutional kernel. 2) Different Objects: BF applies to pixel intensity (RGB) and coordinates to get the weights. Our method applies to high-level features to get the weights. 3) Different weights: In BF, learned weights are for aggregating pixels (*i.e.*, intensity). In ours, learned weights are for aggregating edges (*i.e.*, coordinate).

[Q2] Analysis of complexity. Good question. Based on the backbone of PointNet, the parameters of our method is around 12M with 981 MFLOPs/sample. Our method is able to process 1k objects per second on a Nvidia TitanXp GPU.

[Q3] Mean, stddev and simple size. Good question. The experiments are repeated over three times with the average one reported on Tab. 2. We further collected stddev in the range of 0.6-1.4, and every object is sampled in 1,024 points.

[Q4] Results of fine-tuning models. It's a good suggestion. By selecting 30% labeled target samples in training set, the fine-tuning results on M-to-S and S-to-M are 75.1% and 49.6% respectively. It gains about 4%-5% improvements.

[Q5 & Q6] Weaknesses and **Comparison with reference papers.** Please see the answer above **[CQ1]** and **[CQ2]**.

Response To Reviewer 2:

[Q1] Effects of our method on different classes. It's a good observation. Please see the answer above on **[CQ1]**.

[Q2] The difference between local alignment and re-weighting of PointNet features. It's a good question. Local alignment aligns region-level features. Then, aligned region features with the attribute of common domain knowledge are interpolated back to point features, which are updated to bridge the domain gap rather than just re-weighting original PointNet features. In our experiments, the performance has dropped from 69.3% to 62.2% on M-to-S by replacing local alignment with an attention module on PointNet features which indicates re-weighting method doesn't work for DA.

[Q3] Why some classes profit from DA. It is a good question. In general, same-class objects, which look similar across domains but discriminative with other class objects, are easier to be aligned. In previous global alignment, the similarity of objects is measured by global semantic features in the feature space. While in 3D data, the global semantic features are often confused between many similar classes, but their geometric local structures could be varied a lot. So in our local alignment module, we try to align the similar local structure representations across domains apart from global alignment, which works especially well for the 3D data containing rich geometric information. For instance, *Cabinets* often have diversified structures, and profit a lot from our SA nodes focusing on unique local structures alignment.

[Q3] Public Codes & Datasets. Practical question. Yes, we will do that and please see our reply in **[CQ3]** above.

[Q4] Minor fixes. Many thanks for the reviewer's patient reading and careful checking. We do apologize for the inconvenience caused by these problems. We will revise all the typos and correct the formulas in the final version paper.

Response To Reviewer 3:

[Q1 & Q2] Comparisons and **performance on different classes.** Good observation. Please see reply **[CQ1 & CQ2]**.

[Q3] Extend to other 3D tasks. Good question. Classification is the first step and we are planning to extend the method to 3D object detection and segmentation. It is a promising field and we will continuously work in this direction.

[Q4] Relationship between global alignment and MCD. Good question. We adopted exactly the same setting for global alignment as those of MCD. Drop of *Bed* is mainly caused by data imbalance detailedly explained in **[CQ1]**.

[Q5] Section 3.5. We will provide a more detailed theoretical analysis in the final version and move it to supplementary.