We thank all the reviewers for their positive comments and constructive suggestions. We will incorporate the suggested changes and add the relevant references in the final revision, if accepted. Below, we address the primary questions.

Response To Reviewer #1

Q1: The number of iterations? We have conducted experiments of our full algorithm with 3 to 7 iterations, and observed worse performance after 5 iterations (Part accuracy: 45.51/49.51/44.6 for 3/5/7 iterations on the Table category), so we finally set that number to 5 in our paper. We find empirically that having too many iterations, makes the network harder to train and difficult to convergence. We will add a discussion in the final revision.

Q2: Different number of points per part? We have tried to sample 500 points per part. The performance remains roughly unchanged (PA: 48.78) with 500 points compared to the setting of sampling 1000 points (PA: 49.51) in our paper. This indicates that as long as the point samples are sufficient to cover the part geometric details, the performance does not change too much with increased sampling.

Response To Reviewer #2

Q1: Comparison to average pooling? We observed no significant performance difference between the average and max pooling (PA: 48.92 v.s. 49.51), indicating that both ways work. We will add the comparison in the revision.

Q2: Relation between an aggregated part and the other parts? We demonstrate this relation via the visualization in Fig.3 (the odd steps). For odd iterations, the relation graph is updated from the sparse node set, where the geometrically-equivalent parts, such as legs, are aggregated to a single node, and thus have relatively more influence as compared to the other parts. This is done the opposite way for the even iterations. We will make this clearer in the final paper.

Q3: Is"Our backbone w. relation reasoning" the model with aggregation module? No. It refers to the dynamic relation reasoning module mentioned in Sec 3.2. Further ablation study about the part aggregation is in the supplementary.

Q4: Discuss more about the results in Fig.3? The part relationship edges in our part graph are directional, so the relation from part A to part B can be different from the relation from B to A. So, in your examples, empirically we find that seat(arm) is more influential to back(leg), but it's not the case in the opposite direction. Legs are usually symmetric, so it is very likely that they share the same/similar part orientations, which explains their strong correlations.

Q5: Known part count or semantics? Previous works usually assume a known part count or semantics for the entire category of shapes (e.g., for chairs, four semantic parts: back, seat, leg and arm), while our algorithm does not rely on such an assumption. Our input is a set of parts, and the part count is computed purely from the geometry of the parts, so it can vary across different shapes. We will explain this more fully in the final paper.

Response To Reviewer #3

Q1: Experimental details for baselines, training strategies? The baseline methods are trained using the same losses and the same termination strategy as our method. We stop training when they achieve the best scores on the validation set. We will add more details in the final paper and release the code to the public.

Q2: Missing ablation of training losses? Experimentally, we find that removing any of the three losses (PA: w/o $\mathcal{L}_t/\mathcal{L}_r/\mathcal{L}_s$: 24.48/46.94/48.23) is less satisfactory compared to the full version (PA: 49.51).

Q3: Discussion on part connectivity? In Sec.4 of supplementary, we presented and discussed some failure examples with part dis-connection artifacts. Even though our method implicitly considers part relationships, we agree that more explicit handling of the part connectivity may further improve the results. We leave this as a future work.

Response To Reviewer #4

Q1: Adapt existing 3D shape generation works? Our task of part assembly takes as inputs a given set of parts to assemble and outputs the part poses for each part. This is quite different from the 3D shape generation tasks, where the part geometry is generated as outputs. However, there are some shared techniques, e.g., handling of part relationships.

Q2: Timings on network inference? The network inference roughly takes 0.3 second per example, which is very fast. In the revision, we can add more thorough comparisons on timings if requested.

Q3: Replace the 6-DoF pose prediction with 9-DoF? Our problem is motivated by the practical scenario of assembling an IKEA furniture, where the scale for each part is fixed by definition. However, we agree with the reviewer that it could be interesting to apply our method for other analogous problems that require 9-DoF prediction.

Q4: Only compare with the ground truth? As indicated in Line 226-228, when we evaluated the performance of these methods, we generated multiple possible assembly outputs with different Gaussian random noise, and measured the minimum distance from the assembly predictions to the ground truth, which allows the models to generate shapes that are different from ground truth while similar to real-world objects. We can make this clearer in the final paper.