G-FARS: Gradient-Field-based Auto-Regressive Sampling for 3D Part Grouping Supplementary

Junfeng Cheng
Imperial College London
London, UK
junfeng.cheng20@imperial.ac.uk

Tania Stathaki
Imperial College London
London, UK
t.stathaki@imperial.ac.uk

1. Datasets

As stated in the main paper, we applied three types of shapes, chair, table and lamp from PartNet [3] dataset in our experiments. We use the random mixing method discussed in the main paper to create our training and testing data. In this section, we discuss more details about our datasets.

Data statistics In the main paper, we have described our approach to generate 3D part grouping datasets. We provide detailed statistics of our 3D part grouping datasets in the table below:

Dataset	Mixed Num.	Train Mixed Set	Test Mixed Set
Chair	2	2381	329
	3	1032	139
	total	3413	468
Table	2	3160	471
	3	1351	208
	total	4511	679
Lamp	2	1352	209
	3	620	90
	total	1972	299

Table 1. The detailed statistics of our mixed part datasets.

Statistics for the Number of Parts We present the detailed statistics for the number of parts in the constructed mixed part sets in Fig. 1. In this figure, the statistics for both training and testing datasets of the three shapes are shown. The horizontal axis represents the number of parts in a single mixed part set, while the vertical axis indicates the quantity of the corresponding part sets in the datasets. For the chair and table datasets, the maximum number of parts exceeds 50; however, for the lamp dataset, this number is only 30. Most part sets in the chair and table datasets contain 10 to 35 parts, while the majority of part sets in the lamp dataset include 5 to 17 parts.

2. More Details about Baselines

We roughly introduce our baselines in the main body, and more details about them are discussed here.

2.1. GRU-Mask

In this baseline, we employ the same 3D encoding technique (*i.e.*, PointNet [4]) as G-FARS uses. Following the PointNet, we utilize a GRU to sequentially encode the input parts. The GRU enables us to capture the relationships among all parts. Finally, we apply an MLP to generate a mask that represents all the selection methods for the parts.

The quantitative and qualitative comparisons demonstrate that this method achieves the goal of 3D part grouping to some extent. It can generate all the selection vectors without relying on auto-regressive inference. However, a drawback of this method is the necessity to predetermine the output size of the MLP during network design, which limits the number of groups to this predetermined number.

2.2. Comp-Net

The idea of Comp-Net is to compare two parts and identify whether they can be grouped together. To implement this concept, we employ a 'dual PointNet' structure. The first PointNet encodes all the input parts, while the second one is tasked with the goal of part comparison. The output of the second PointNet is a boolean value, indicating whether two parts can be grouped together.

Based on the results discussed in the main paper and supplementary materials, we see that this method is a feasible approach for the 3D part grouping task. However, a disadvantage of this approach is that Comp-Net is only trained to compare any two parts. This means it struggles to understand the relationships among multiple parts (more than two parts).

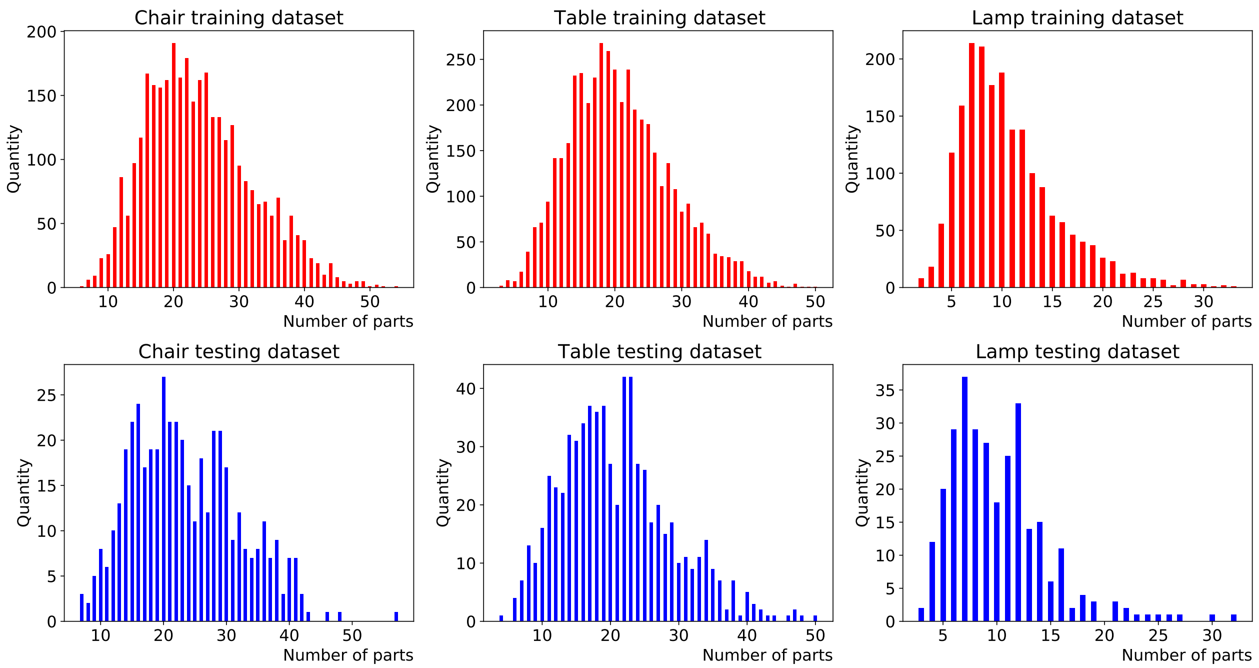

Figure 1. The statistics for the number of parts in the mixed part set. The horizontal axis is the number of parts in one mixed part set, and the vertical axis represents the quantity of the mixed part sets which include the corresponding number of parts.

2.3. Variants of G-FARS

We present three variants of G-FARS in our main paper: G-FARS-CG, G-FARS-R, and G-FARS-T. As stated in the main paper, we modify the approach for modeling the score function for these variants. Specifically, we attempt to model the score function as $S_\theta = \nabla_c \log p_t(c_n^m \mid GNN(F_P^n, f_m))$, where f_m represents the encoded feature for the m^{th} part in the part set, and c_n^m denotes the corresponding selection boolean value for this single part. The GNN is implemented using an EdgeConv-based structure. G-FARS-CG, G-FARS-R, and G-FARS-T apply an MLP, ResNet [1], and Transformer [6], respectively, to learn the new score function. In these variants, we separate the GNN from the score function, aiming to determine whether the G-FARS framework can be effectively adapted to score functions modeled in this manner. Furthermore, we seek to explore whether applying better architectures can enhance the network's performance under this new modeling approach.

3. Experimental details

Training details In our experiments, the optimizer applied for training is Adam [2]. The learning rate is set as 10^{-3}, and the batch size is set as 16. In the training procedure, we select the best checkpoints for each dataset.

Sampling details As mentioned in our main paper, we use Predictor-Corrector sampler [5] for both selection vector sampling and pose matrix sampling. The parameters for both samplers are set as $T = 1.0, \sigma = 25.0, C = 1$. The sampling step N is set as 500.

4. Additional Experiments

4.1. Category Mixing Testing

In this experiment, we mix all three categories (chair, table, and lamp) and test the performance of G-FARS on the mixed-category dataset. The results are shown in Table 2. Surprisingly, we find that the performance on this mixed-category dataset is even better than that on single-category data. We infer that this improvement is due to two main reasons: 1. The mixing of categories results in a larger dataset, which may lead to better generalization; 2. Parts become more distinguishable when mixed together (e.g., chair parts versus lamp parts).

Precision	Recall	F1
0.896 / 0.866	0.804 / 0.792	0.84 / 0.827

Table 2. Results on all mixed categories data

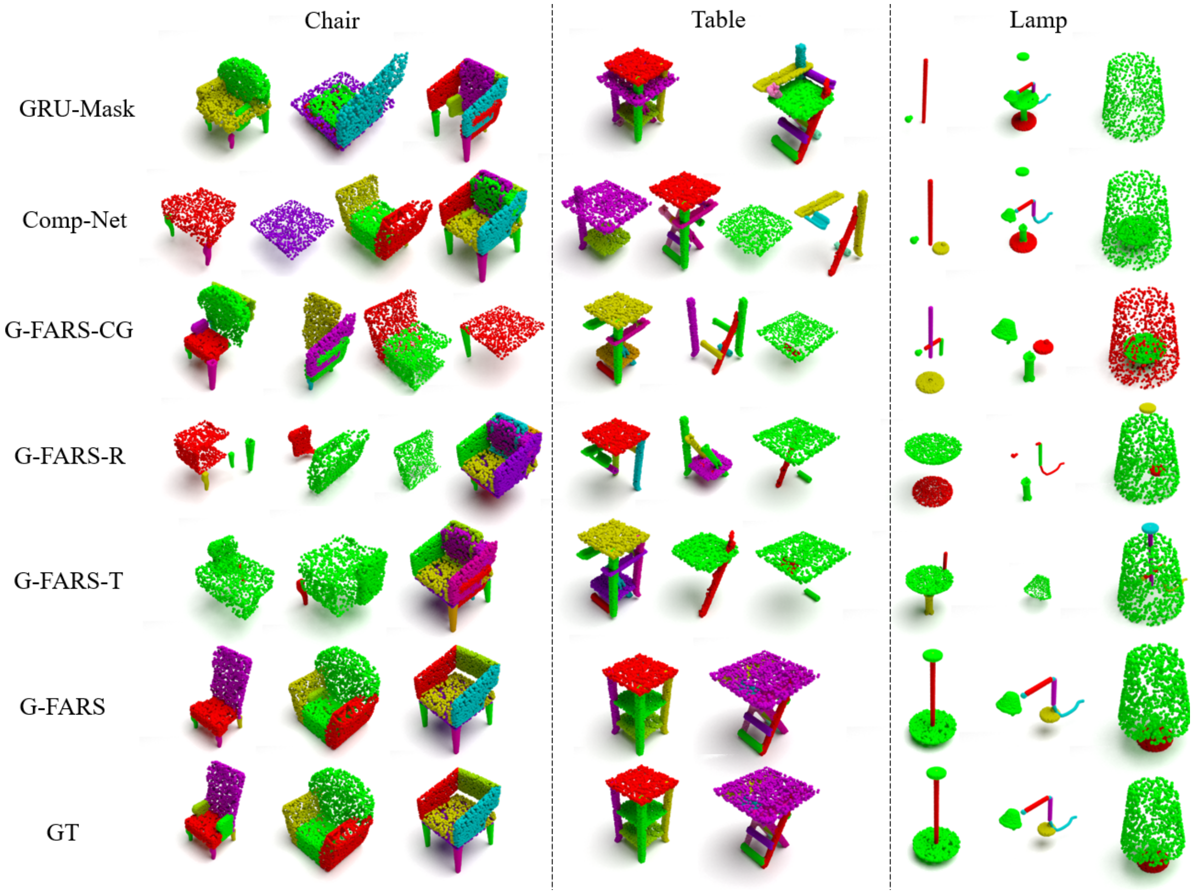

Figure 2. The full comparison for Fig. 4 of the main paper.

4.2. Generalization to Unseen Categories

To further verify the generalization ability of our algorithm for unseen category objects, we conducted an experiment, the results of which are shown in Table 3. In this experiment, the model is trained on the chair dataset but tested on the table dataset. Although the performance is lower than that of the model trained and tested on the same table dataset, it is still capable of grouping parts from unseen categories. This demonstrates that our algorithm can generalize to a certain extent to object types it has not previously encountered.

Precision	Recall	F1
0.766 / 0.716	0.697 / 0.685	0.717 / 0.7

Table 3. Testing on Table with the model trained on Chair.

4.3. More Qualitative Results

In Fig. 2, we present the full comparison for the Fig. 4 of the main paper. The full results indicate that the baseline methods struggle to accurately group the 3D parts. Besides, we also present additional qualitative comparisons in Fig. 3. The figure shows that our framework is able to correctly group most part sets, while it is difficult for other baselines to obtain the correct groups. This result proves the effectiveness of our proposed method.

4.4. Additional Results of Noisy Part Removal

We demonstrate additional results of noisy part removal task in Fig. 4. The figure shows that our framework can remove noisy parts from the given part sets in a zero-shot manner.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[2] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[3] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tri-

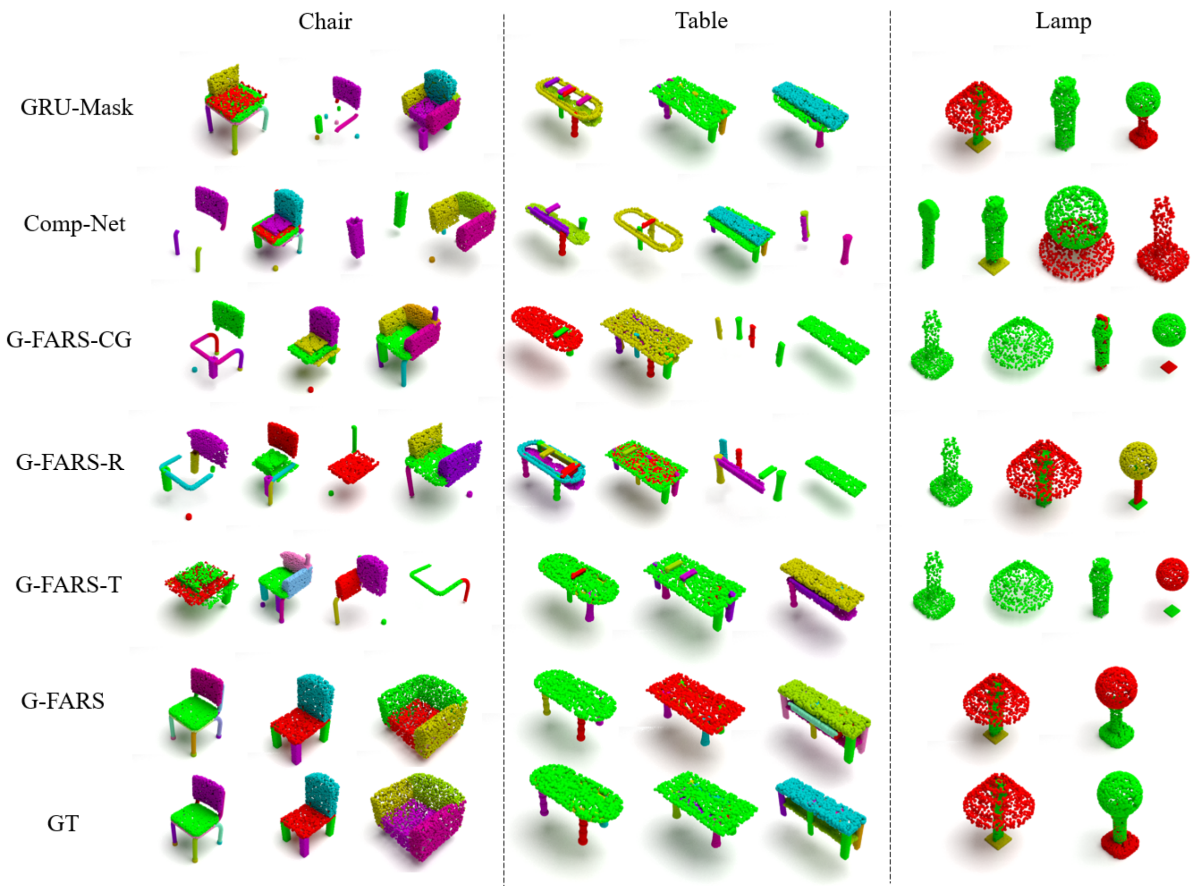

Figure 3. More qualitative comparisons on chair, table and lamp datasets.

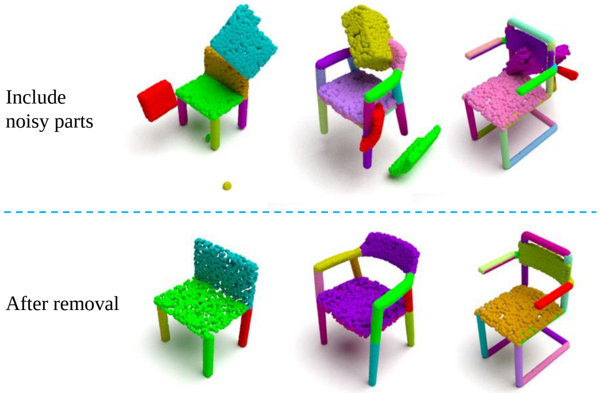

Figure 4. More qualitative results of noisy part removal task. Our framework can remove the unnecessary parts from the given set of parts.

pathi, Leonidas J Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 909–918, 2019. 1

[4] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 1

[5] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *arXiv preprint arXiv:2011.13456*, 2020. 2

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2