

Supplementary Material for What Makes for Effective Few-shot Point Cloud Classification?

Chuangguan Ye¹, Hongyuan Zhu², Yongbin Liao¹, Yanggang Zhang¹, Tao Chen¹, Jiayuan Fan³

¹School of Information Science and Technology, Fudan University, China

²Institute for Infocomm Research, A*STAR, Singapore

³Academy for Engineering and Technology, Fudan University, China

{cgye19, 19210720121, ygzhang19, eetchen, jyfan}@fudan.edu.cn, hongyuanzhu.cn@gmail.com

A. Details about 3D Backbones

To study the influence of different backbone architectures on FSL, we select three types of current state-of-the-art 3D networks as the support backbones for features extraction. In this section, we will introduce more structural details about backbones employed in Section 3 and 5. The input point cloud instances consist of 512 points with 3d coordinates and the backbones output a feature vector with 1024 dimensions.

Pointwise MLP Networks: **PointNet** contains five MLP layers (64,64,64,128,1024) with learnable parameters, and batch normalization is used for all MLP layers with ReLU. After that, we use maxpooling function to aggregate a global feature vector. Note that we remove the transform layers in original PointNet [10] framework for simplicity and efficiencies. **PointNet++** consists of 3-level PointNet Set Abstractions with single scale grouping (SSG), which have the same setting in [11]. We remove the fully connected (FC) layers and take the last PointNet Set Abstraction’s output as the global feature vector.

Convolution Networks: There are 4 X-conv layers (48,96,192,384) with ReLU in **PointCNN**. The last X-conv layer outputs a 384-dimension feature vector and we use 2 FC layers (512,1024) to extend the dimension to 1024 for fair comparisons. **RSCNN** contains 3 RC-Conv layers (128,512,1024) with single scale neighborhood (SSN) grouping. Other settings of RC-Conv are same with [6]. **DensePoint** contains 3 P-Conv layers, 2 P-Pooling layers and 1 global pooling layer. The settings of these layers are same with [5]. We remove the FC layers in RSCNN and DensePoint for outputting the 1024-dimension global feature vectors too.

Graph-based Network: **DGCNN** is the embedding network of our baseline for 3DFSL, consisting of 4 Edge-Conv layers (64,64,128,256). The outputs of each Edge-Conv will be concatenated as a 512-dimension feature map. Then the feature map will be fed into an MLP layer to ex-

tend its dimension to 1024. At last, a maxpooling function is used to aggregate the global features and outputs a 1024-dimension feature vector. Figure 6 illustrates the network architecture of DGCNN.

B. Details about FSL Baselines

In this section, we will introduce more adapting details about FSL algorithms for few-shot point cloud classification in Section 3 and 5.

Metric-based methods: We take Squared Euclidean Distance as metric function and use Cross-Entropy loss in ProtoNet. For RelationNet, we first construct support-query pair features $f_{sq} \in \mathbb{R}^{2 \times 1024}$ by concatenating the support feature vector $f_s \in \mathbb{R}^{1 \times 1024}$ and query feature vector $f_q \in \mathbb{R}^{1 \times 1024}$. Then the pair features f_{sq} are fed into a relation module, which contains two convolutional blocks ((1x1 conv 128 filters, BN, ReLU), (1x1 conv 1 filters, BN, ReLU)) and two FC layers (128,L) (L denotes the number of classes at a meta-task). After that, relation module outputs the predicted relation score and Mean Square Error loss is used to regress relation score to ground truth. For FSLGNN, we construct 4 GNN layers with the same setting in [15], and use Cross-Entropy loss as loss function.

Optimization-based methods: We use two FC layers (256, K) as the classifier in Meta-learner and MAML. Batch normalization is used for the first FC layer with ReLU and Dropout operation. There is a 2-layer LSTM in Meta-learner following the same configuration in [12], where the first layer is a normal LSTM and the second layer is meta-learner LSTM for gradient state updating. The meta learning rate in MAML is 0.1 for 5-shot and 0.5 for 1-shot. The classification head in MetaOptNet is the multi-class SVM presented in [1]. We also take Cross-Entropy loss as loss function in these methods.

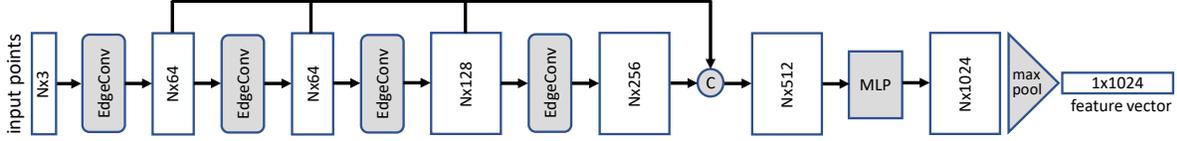


Figure 6. The architecture of DGCNN for feature embedding. The details of EdgeConv could be find in [19].

C. Details about Channel Interaction Module

The architecture of Channel Interaction Module is shown in Figure 7. Query-vector q and key-vector k are generated from the input feature f with two linear embedding functions φ and γ . Then the channel relation score map can be denoted as $R = q^T k$. After that, we can obtain the reweighted feature $v = fR'$, where $R' = \text{soft max}(R)$. Finally, for compensating the discarded information, we combine v and f to get updated feature $f' = v + f$.

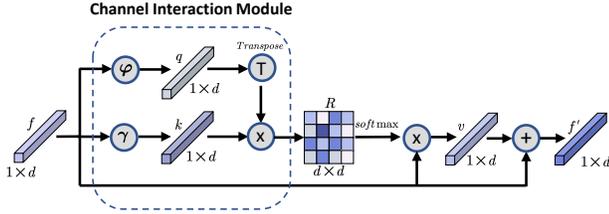


Figure 7. Details of Channel Interaction Module in Self-Channel Interaction Module introduced in Section 4.2.1.

D. Details about Cross Instance Fusion Module

Cross-Instance Fusion (CIF) module is proposed to address the low-data and high intra-class variances issues in Section 4.2.2, which can enrich prototypical information and rectify feature distribution by fusing prototype features and query features with a meta-learner.

As illustrated in Figure 8, for updating prototype features f_p , we first concatenate each prototype feature with K_1 query features with the highest cosine similarity and get the concatenated feature Z_{f_p} . Then we employ two simple 1×1 Conv layers as a meta-learner to learn cross-instance interactions and output updated prototype features f'_p . Concretely, the first layer is designed to encode Z_{f_p} to generate a d -dim feature interaction Z'_{f_p} , and the second layer is used to adjust the interaction's dimension so as to generate a reweight matrix W_{f_p} for Z_{f_p} . Finally, we update the prototype features by fusing the concatenated feature Z_{f_p} based on W_{f_p} . Similarly, we could also easily update the query features f_q by CIF module. Furthermore, we use the validation set to determine the value of K_1 and h , and set them to 45 and 64 respectively according to the results shown in Figure 9.

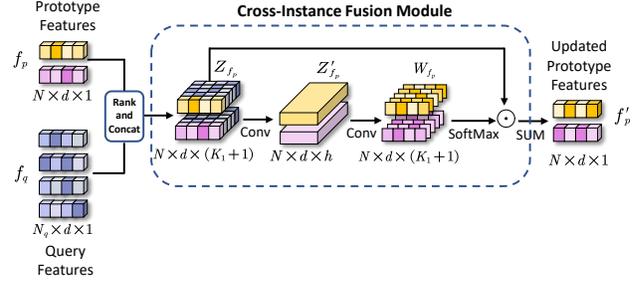


Figure 8. The illustration of updating prototype features by CIF module. Here we set $K_1 = N_q$. \odot is the instance-wise product.

E. Extra Experimental Results

We also conduct extra experiments to further explore the effects of our proposed network in two scenarios, including fine-grained few-shot classification and 2D image few-shot classification.

Fine-Grained Few-Shot Classification. We study the proposed network in a fine-grained classification scenario to evaluate its ability to distinguish similar categories. We first train the baselines and the proposed network on meta-train data of ShapeNet70-FS, and test them on seven subcategories of “Airplane” in meta-test data under a 5way-1shot-15query setting. The mean accuracy of each class are listed in Table 9. One can see that, our proposed network outperforms other baselines most of the time and improve the mean accuracy more than 3%.

Ablation Studies of Residual Design in SCI Module. We design the SCI as a residual update to compensate the discarded information, because the Softmax operation can highlight the weight of discriminative channels, but it also may discard some information from the original features. The results of ablation studies listed in Table 10 show that this residual design can gain performance improvement.

Comparison Results of More-way k-shot Setting. We conduct the experiments on ShapeNet70-FS with $N=\{5,10,15\}$, $K=\{1,5\}$, and the results are in Table 11. A larger N-way setting is more challenging with significantly performance dropping. One possible explanation is that larger classes with few support examples make training harder, leading to obscurer class boundaries.

Effects of CIA for 2D FSL. While the CIA model is designed for few-shot point cloud classification task, we also

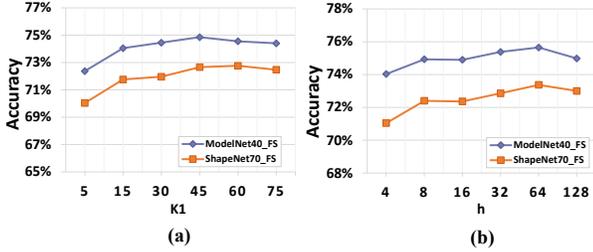


Figure 9. (a) and (b) are the ablative results of different values of K_1 and h in Cross-Instance Fusion module respectively. We conduct these ablation experiments under the 5way-1shot-15query scenario on ModelNet40_FS and ShapeNet70_FS.

Methods	Airline	Jet	Fighter	Swept Wing	Propeller Plane	Bomber	Delta Wing	Mean
ProtoNet [16]	49.59	25.21	27.64	41.30	35.76	20.73	46.24	35.22
RelationNet [17]	49.28	24.55	30.36	41.03	28.26	24.05	44.79	34.63
FSLGNN [15]	48.53	23.56	21.01	31.75	24.29	23.58	58.49	33.08
Meta-learner [12]	29.76	21.92	30.94	27.83	27.02	24.96	30.22	27.53
MAML [2]	56.03	26.65	34.37	21.90	16.43	25.30	15.71	28.12
MetaOptNet [4]	46.81	24.72	29.05	39.74	24.42	25.17	38.23	32.60
Ours	48.23	25.51	35.16	41.88	46.63	26.06	48.02	38.80

Table 9. 5way-1shot-15query classification results (accuracy %) on fine-grained classes in ShapeNet70-FS.

	ModelNet40-FS		ShapeNet70-FS	
	5w-1s	5w-5s	5w-1s	5w-5s
w/o Res	74.64	86.81	72.98	82.87
w/ Res	75.70	87.15	73.57	83.24

Table 10. The ablation study of residual design in SCI module.

	ShapeNet70-FS					
	5w-1s	5w-5s	10w-1s	10w-5s	15w-1s	15w-5s
ProtoNet	<u>65.96</u>	<u>78.77</u>	50.57	<u>67.29</u>	43.15	<u>59.39</u>
RelationNet	65.88	76.25	<u>50.93</u>	63.14	43.04	53.31
MetaOpt	65.08	<u>77.81</u>	48.97	64.50	40.83	56.10
Ours	69.36	80.31	54.26	67.69	47.38	60.48

Table 11. The comparison results under larger-way settings.

study the effects of CIA model for the case of 2D image few-shot classification on *miniImagenet* and *tieredImagenet* with ResNet12 as backbone. The results shown in Table 12 indicate that the CIA model also can improve the classification performance of ProtoNet [16], especially for 1-shot setting, and achieves competitive performance compared with state-of-the-art 2D FSL approaches.

F. More Visualization Analysis

Visualization Analysis of Feature Heatmap. We further visualize the feature heatmap of point cloud instances to qualitatively evaluate the proposed modules in Section 4.2. Figure 10 are the comparative results before and after incorporating SCI module and CIF module respectively. Deeper color means higher feature responding in this region. We could observe that the SCI module pays more attention on the discriminative fine-grained parts of differ-

Backbone	Method	<i>miniImagenet</i>		<i>tieredImagenet</i>	
		5w-1s	5w-5s	5w-1s	5w-5s
ResNet12	SNAIL [8]	55.71	68.88	-	-
	TADAM [9]	58.50	76.70	-	-
	ECM [13]	59.00	77.46	63.99	81.97
	TPN [7]	59.46	75.65	59.91	73.30
	MetaOptNet [4]	62.64	78.63	65.99	81.56
	CAN [3]	63.85	79.44	69.89	84.23
	ProtoNet [16]	60.37	79.02	65.65	83.40
	ProtoNet [16]+CIA	63.05	80.02	70.10	83.73

Table 12. Comparisons of the classification results after incorporating CIA Module into ProtoNet [16] on *miniImagenet* [18] and *tieredImagenet* [14] with ResNet12 as backbone.

ent classes, such the "cap" of Bottle and the "legs" of Stool, while the CIF module could activate more diverse regions, which could help to generate more informative features. More comparative visualizations are shown in Figure 11 and 12. One can see that, the CIA module can highlight more discriminative parts and structures, which enriches the information learned from point cloud instances.

G. Dataset Split

ModelNet40-FS is a new split of ModelNet40, containing 30 training classes with 9,204 examples and 10 disjoint testing classes with 3,104 examples. Statistics of ModelNet40-FS are reported in Table 13, and details of training set split and testing set split are listed in Table 15.

	Train	Test	Total
Classes	30	10	40
Instances	9,204	3,104	12,308

Table 13. Statistics of ModelNet40-FS dataset.

ShapeNet70-FS is adapted from ShapeNetCore, including 50 base classes from 34 categories and 20 novel classes from 14 categories. Statistics of the ShapeNet70-FS are reported in Table 14, and details of training set split and testing set split are listed in Table 16 and Table 17 respectively.

	Train	Test	Total
Categories	34	14	48
Classes	50	20	70
Instances	21,722	8,351	30,073

Table 14. Statistics of ShapeNet70-FS dataset.

References

- [1] Koby Crammer and Yoram Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of machine learning research*, 2(Dec):265–292, 2001.
- [2] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *ICML*, 2017.

Training Set				Testing Set			
Class	Num	Class	Num	Class	Num	Class	Num
chair	989	car	297	radio	124	bookshelf	672
sofa	780	desk	286	xbox	123	vase	575
airplane	725	dresser	286	bath tub	156	bottle	435
bed	615	glass_box	271	lamp	144	piano	331
monitor	565	guitar	255	stairs	144	night_stand	286
table	492	bench	193	door	129	range_hood	215
toilet	444	cone	187	stool	110	flower_pot	169
mantel	384	tent	183	wardrobe	107	keyboard	165
tv_stand	367	laptop	169	cup	99	sink	148
plant	339	curtain	157	bowl	84	person	108

Table 15. The training and testing classes of ModelNet40-FS dataset.

Training Set					
ID	Class	Num	ID	Class	Num
04256520	sofa	1520	04037443	race car	323
03179701	desk	1226	20000011	garage cabinet	307
04401088	phone	1089	03948459	handgun	307
02738535	armchair	1051	04285965	sport utility	300
02924116	bus	939	03928116	piano	239
02808440	bath tub	856	02818832	bed	233
02992529	radiotelephone	831	04330267	stove	218
03891251	park bench	823	03100240	convertible	208
03063968	coffee table	763	04285008	sports car	197
20000027	club chair	748	02880940	bowl	186
02858304	boat	741	03141065	cruiser	181
04250224	sniper rifle	717	02961451	carbine	172
03046257	clock	651	04004475	printer	166
03991062	pot	602	03761084	microwave	152
03593526	jar	596	04225987	skateboard	152
03237340	dresser	482	04460130	tower	133
04380533	table lamp	464	20000020	cantilever chair	125
03642806	laptop	460	02801938	basket	113
04166281	sedan	429	02814533	beach wagon	108
03624134	knife	424	02946921	can	108
20000037	rectangular table	421	03938244	pillow	96
03119396	coupe	418	03594945	jeep	95
04373704	swivel chair	398	03207941	dishwasher	93
20000010	desk cabinet	356	04099429	rocket	85
03790512	motorcycle	337	02773838	bag	83

Table 16. The training classes of ShapeNet70-FS dataset. "ID" corresponds to WordNet synset offset.

- [3] Ruibing Hou, Hong Chang, Bingpeng Ma, Shiguang Shan, and Xilin Chen. Cross attention network for few-shot classification. In *NeurIPS*, 2019.
- [4] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *CVPR*, 2019.
- [5] Yongcheng Liu, Bin Fan, Gaofeng Meng, Jiwen Lu, Shiming Xiang, and Chunhong Pan. Densepoint: Learning densely contextual representation for efficient point cloud processing. In *CVPR*, pages 5239–5248, 2019.
- [6] Yongcheng Liu, Bin Fan, Shiming Xiang, and Chunhong Pan. Relation-shape convolutional neural network for point cloud analysis. In *CVPR*, pages 8895–8904, 2019.
- [7] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho

Testing Set					
ID	Class	Num	ID	Class	Num
03211117	display	1093	03337140	file cabinet	298
02690373	airline	1054	20000001	swept wing	271
03467517	guitar	797	03797390	mug	214
03325088	faucet	744	04554684	washer	169
03595860	jet	675	03513137	helmet	162
03335030	fighter	597	04012084	propeller plane	137
02876657	bottle	498	02867715	bomber	130
02871439	bookshelf	452	03174079	delta wing	121
04468005	train	389	02942699	camera	113
02747177	ashcan	343	03710193	mailbox	94

Table 17. The testing classes of ShapeNet70-FS dataset. "ID" corresponds to WordNet synset offset.

- Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. In *ICLR*, 2019.
- [8] Nikhil Mishra, Mostafa Rohaninejad, Xi Chen, and Pieter Abbeel. A simple neural attentive meta-learner. In *ICLR*, 2018.
 - [9] Boris N. Oreshkin, Pau Rodríguez, and Alexandre Lacoste. TADAM: task dependent adaptive metric for improved few-shot learning. In *NeurIPS*, 2018.
 - [10] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *CVPR*, 2017.
 - [11] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *NeurIPS*, 2017.
 - [12] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2017.
 - [13] Avinash Ravichandran, Rahul Bhotika, and Stefano Soatto. Few-shot learning with embedded class models and shot-free meta training. *arXiv preprint arXiv:1905.04398*, 2019.
 - [14] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *ICLR*, 2018.
 - [15] Victor Garcia Satorras and Joan Bruna Estrach. Few-shot learning with graph neural networks. In *ICLR*, 2018.
 - [16] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *NeurIPS*, 2017.
 - [17] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *CVPR*, 2018.
 - [18] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *NeurIPS*, 2016.
 - [19] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *Acm Transactions On Graphics (tog)*, 38(5):1–12, 2019.



Figure 10. The heatmap of point cloud instances before and after using CIA module (SCI and CIF). Deeper color means higher feature responding in this region. The classes of each row are 'Bottle', 'Airplane' and 'Stool' respectively.

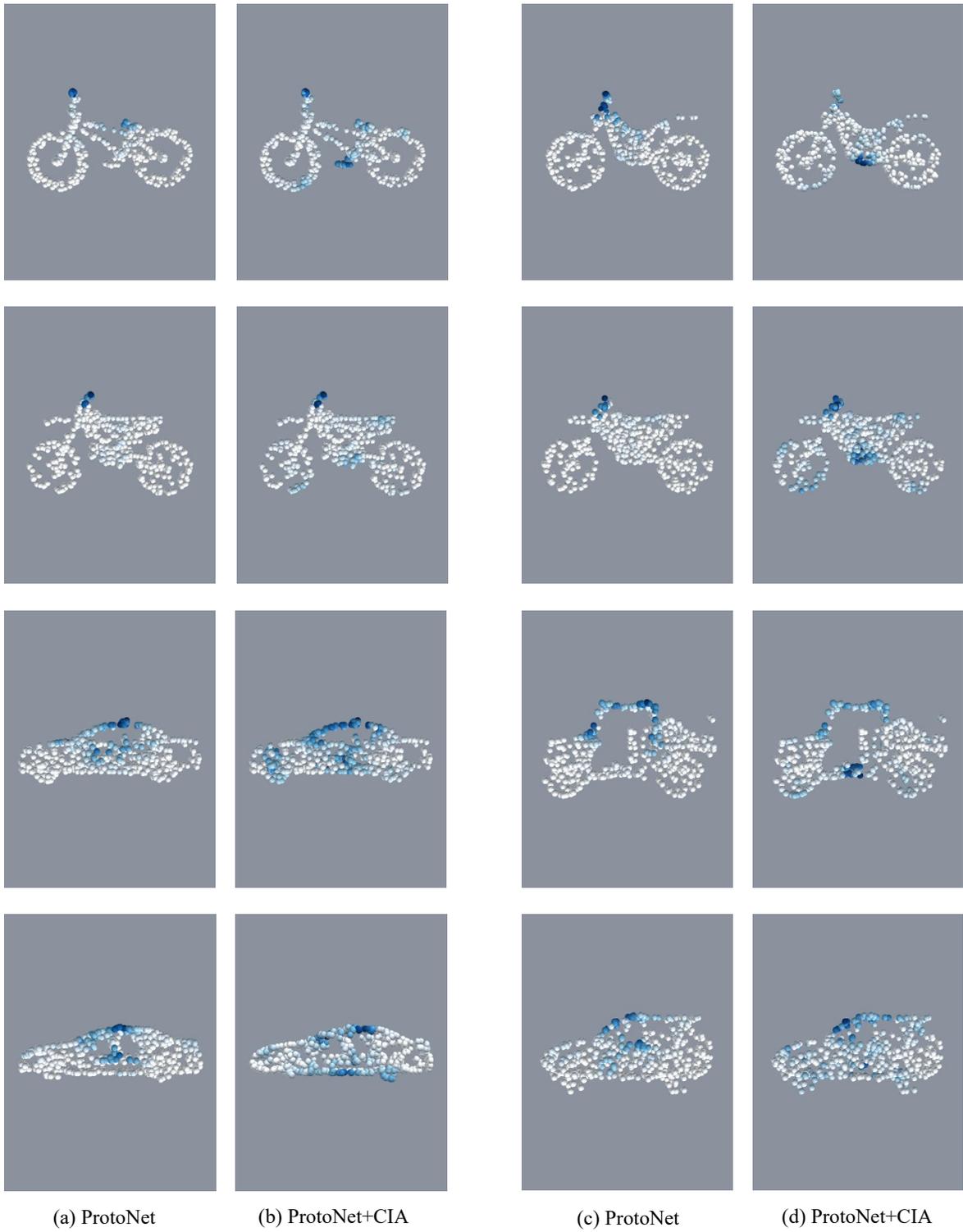


Figure 11. The heatmap of point cloud instances. Deeper color means higher feature responding in this region. Column(a) and column(c) are the results of ProtoNet. Column(b) and column(d) are the results of ProtoNet incorporating with CIA module.

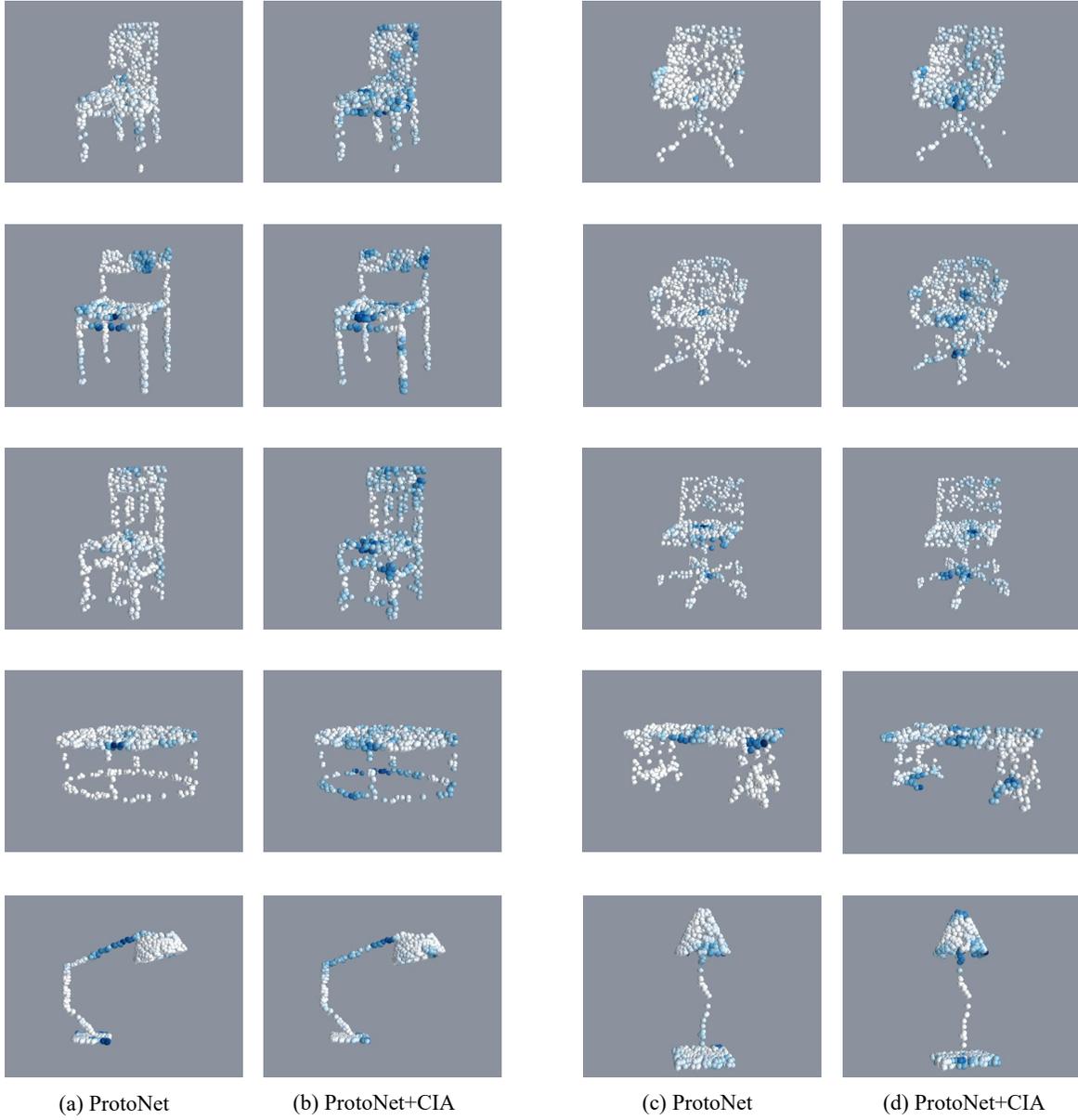


Figure 12. The heatmap of point cloud instances. Deeper color means higher feature responding in this region. Column(a) and column(c) are the results of ProtoNet. Column(b) and column(d) are the results of ProtoNet incorporating with CIA module.