APPENDIX A
EMBEDDING NETWORK STRUCTURES

Single scale. We adopt the adaptive graph convolutional network (AGCN) [24] as our single scale skeleton embedding network, which is the stack of 9 adaptive graph convolutional (AGC) blocks, global average pooling layer, and a *softmax* classifier, as shown in Fig. 5. For the details of the AGC block and the setting of each block, please refer to [24].

Multi-spatial scale. To construct the multi-spatial scale skeleton, we first build six AGC blocks on the original spatial scale to capture the joint-wise feature representation and then perform the spatial pooling to generate the spatial scale 2 and spatial scale 3 features. To extract the multi-spatial scale skeleton representations, all three spatial streams undertake classification during the pre-training stage, as shown in Fig. 6. Each stream is independently optimized using the cross-entropy loss. The vertically parallel AGC blocks share the same setting, except for the skeleton graph structure.

Multi-temporal scale. Similarly, after processing the first six AGC blocks in the original temporal scale, we perform the temporal pooling to generate the coarser scales (temporal scale 2 and temporal scale 3). To extract the multi-temporal scale skeleton representations, each of the three temporal streams is subject to classification during the pre-training stage and is optimized individually using the cross-entropy loss, as shown in Fig. 7. The vertically parallel AGC blocks share the same setting.

For all types of embedding networks (single scale, multi-spatial scale, and multi-temporal scale), we leverage the output feature maps of 9^{th} block as the skeleton feature representations in the *meta-training* stage.

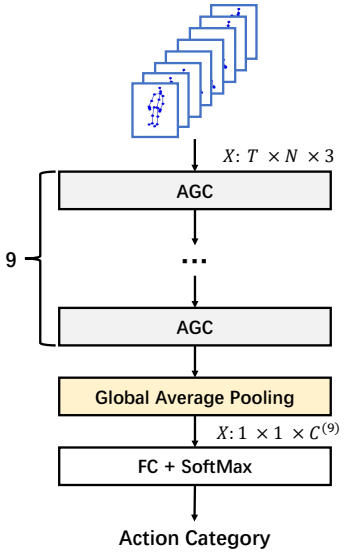

Fig. 5. Illustration of the single-scale embedding network (AGCN [24]). There are a total of 9 AGC blocks, followed by a global average layer and a *softmax* classifier. (*N* denotes the number of skeleton joints, *T* denotes the number of frames, and $C^{(i)}$ denotes the number of output channels at i^{th} block.)

APPENDIX B
ADDITIONAL EXPERIMENTAL RESULTS

More experimental results on different combinations of optimal matching manners: We conduct more experiments on different combinations of optimal matching manners to show the advantages of our proposed optimal matching strategies. Tabs. 7 and 8 show the experimental results under the evaluation protocol 1 and the "Reducing training classes" experiments, respectively. Similar to the conclusion we got in the main manuscript, we can see that including any of our proposed matching strategies (M_s, M_t, C_s, and C_t) improves the one-shot skeleton action recognition performance clearly. Including all four matching strategies perform the best on all three datasets, demonstrating the effectiveness of our proposed optimal matching manners.

More experimental results on different numbers of temporal scales: We conduct additional experiments on different numbers of temporal scales to show the effectiveness of the proposed multi-temporal scale strategy. Specifically, we conduct experiments on two temporal scales (T and $T/2$) and four temporal scales (T, $T/2$, $T/4$, and $T/8$). The experimental results are shown in Tab. 9. It can be seen that the '4-Scale' result is 64.7, which is a little lower than the '3-Scale' (64.9), showing that using 4 temporal scales is redundant. Note that all experiments were conducted on the NTU RGB+D 120 dataset under evaluation protocol 2.

More experimental results on different temporal pooling strategies: We conduct additional experiments on different temporal pooling strategies. Here, we have further expanded our experiments to encompass three additional pooling setups: (1) pooling performed across spans of 2 and 8 consecutive frames (Pooling2_8). (2) pooling performed across spans of 4 and 8 consecutive frames (Pooling4_8). (3) pooling performed across spans of 5 and 10 consecutive frames (Pooling5_10). The experimental results are shown in Tab. 10. We can observe that 'Pooling2_4' performs the best, which is applied the 'Pooling2_4' in our main experiments.

More experimental results on different feature pooling strategies: We conducted this ablation study to evaluate the different feature pooling strategies in cross-scale matching. In the Earth's Mover Distance, the matching score is computed at the channel level, implying that the channel features must be of the same size. In cross-scale matching scenarios, such as cross-temporal scale matching, there's a variation in the size of skeleton features across the three temporal scales. These sizes are denoted as $\mathbf{X}_{t1} \in \mathbb{R}^{C\times N\times T}$, $\mathbf{X}_{t2} \in \mathbb{R}^{C\times N\times T/2}$, and $\mathbf{X}_{t3} \in \mathbb{R}^{C\times N\times T/4}$. We are considering two potential strategies for addressing this:

(1) **Pooling All:** Implementing Average Pooling on the temporal dimension to standardize the three temporal scale features to the shape $\mathbb{R}^{C\times N}$;

(2) **Pooling to the min Dimension:** Adjusting the temporal dimension of $\mathbf{X}_{t1}$ and $\mathbf{X}_{t2}$ through Average Pooling to achieve a size of $4/T$, ensuring compatibility with $\mathbf{X}_{t3}$.

The results from these experiments are presented in Table 11. We can find that both these two pooling strategies achieve very similar results. Therefore, we opt for the **Pooling All** strategy, as it offers an optimal balance between performance and efficiency.

More experimental results on different feature extractors: We conduct an additional ablation study utilizing representations from various backbones. In our main experiments, the representation derived solely from AGCN [24] was

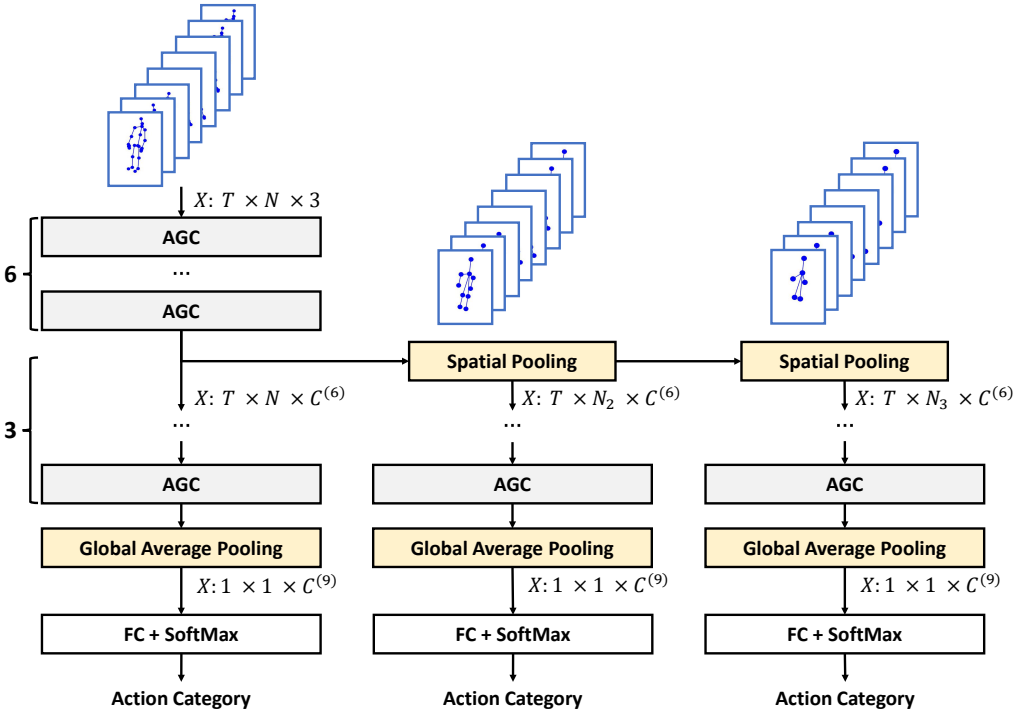

Fig. 6. Illustration of multi-spatial scale embedding network. After processing the first six AGC blocks, we perform spatial pooling to generate the spatial scale 2 and spatial scale 3 features. Finally, the multi-spatial scale features are trained individually and paralleled. For the details of spatial pooling, please refer to Fig. (2)(a) in the main manuscript. (*N* denotes the number of skeleton joints, N_2 denotes the number of nodes for the second-scale spatial graph, N_3 stands for the number of third-scale spatial graph nodes, *T* denotes the number of frames, and $C^{(i)}$ denotes the number of output channels at i^{th} block.)

TABLE 7
Evaluation of different combinations of optimal matching manners under Evaluation Protocol 1. The second line shows the baseline result under the single-scale matching manner. (M_t: multi-temporal matching; M_s: multi-spatial matching; C_t: cross-temporal matching; C_s: cross-spatial matching)

M_t	M_s	C_t	C_s	NTU	NTU 120	PKU-MMD
				80.4	81.2	85.7
✓				81.5	82.4	87.4
	✓			81.6	82.3	87.5
✓	✓			82.6	83.5	88.2
✓		✓		82.9	83.5	88.3
	✓		✓	83.0	83.7	88.4
✓	✓	✓	✓	**83.7**	**84.5**	**89.3**

TABLE 8
Evaluation of different combinations of matching manners on the "Reducing training classes" ablation study. The second line shows the baseline result under the single-scale matching manner. (M_t: multi-temporal matching; M_s: multi-spatial matching; C_t: cross-temporal matching; C_s: cross-spatial matching)

M_t	M_s	C_t	C_s	20	40	60	80	100
				37.6	46.5	54.2	58.7	63.2
✓				39.0	50.5	55.1	60.0	65.0
	✓			40.2	50.4	57.8	60.2	65.1
✓	✓			41.2	52.6	59.0	62.4	67.6
✓		✓		40.2	53.3	56.5	63.5	66.7
	✓		✓	42.1	51.7	58.9	61.5	67.3
✓	✓	✓	✓	**44.1**	**55.3**	**60.3**	**64.2**	**68.7**

TABLE 9
Evaluation of our proposed multi-scale temporal matching with different numbers of temporal scales. The experiments were conducted on NTU RGB+D 120 dataset under the Evaluation Protocol 2.

Number of scales	2-Scale	3-Scale	4-Scale
Multi-Temporal Scale	64.4	**65.0**	64.8

TABLE 10
Evaluation of our proposed multi-scale temporal matching with different temporal pooling strategies. The experiments were conducted on NTU RGB+D 120 dataset under the Evaluation Protocol 2.

Temporal Pooling	Pooling2_4	Pooling2_8	Pooling4_8	Pooling5_10
Multi-Temporal Scale	**65.0**	64.7	64.5	64.0

TABLE 11
Evaluation of our cross-scale matching with feature pooling strategies. The experiments were conducted on NTU RGB+D 120 dataset under the Evaluation Protocol 2.

Pooling methods	Pooling All	Pooling to the min Dimension
Cross-Temporal scale	66.7	66.8
Cross-Spatial scale	67.3	67.3
M&C scale	68.7	68.8

considered. In this expanded scope, we have incorporated two more GCN-based methods as feature extractors, namely ST-GCN [25] and CTR-GCN [55].

The experimental results can be found in Table 12. Our proposed method consistently attains state-of-the-art performance across all three backbones, underscoring the efficacy of our multi-scale and cross-scale matching strategies.

Visualization Results: In order to further show the effectiveness of our proposed multi-scale and cross-scale matching strategies, we provide some visualization examples, as shown in Fig. 8. For Fig. 8 (a), two skeleton samples of the same action "wipe face" were performed by different body parts (two hands vs one hand only) Similarity-based

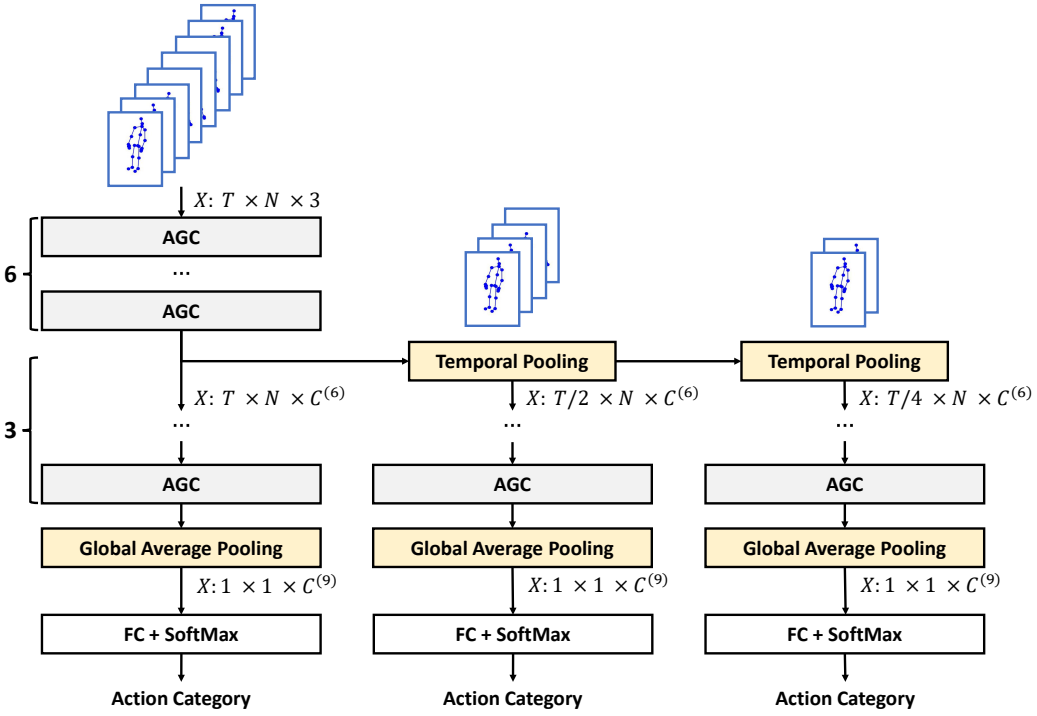

Fig. 7. Illustration of multi-temporal scale embedding network. First, we build six AGC blocks on the original temporal scale to capture the feature representation and then perform the temporal pooling to generate the coarser scales' features. please refer to Fig. (2)(b) in the main manuscript for the details of temporal pooling. (*N* denotes the number of skeleton joints, *T* denotes the number of frames, and $C^{(i)}$ denotes the number of output channels at i^{th} block.)

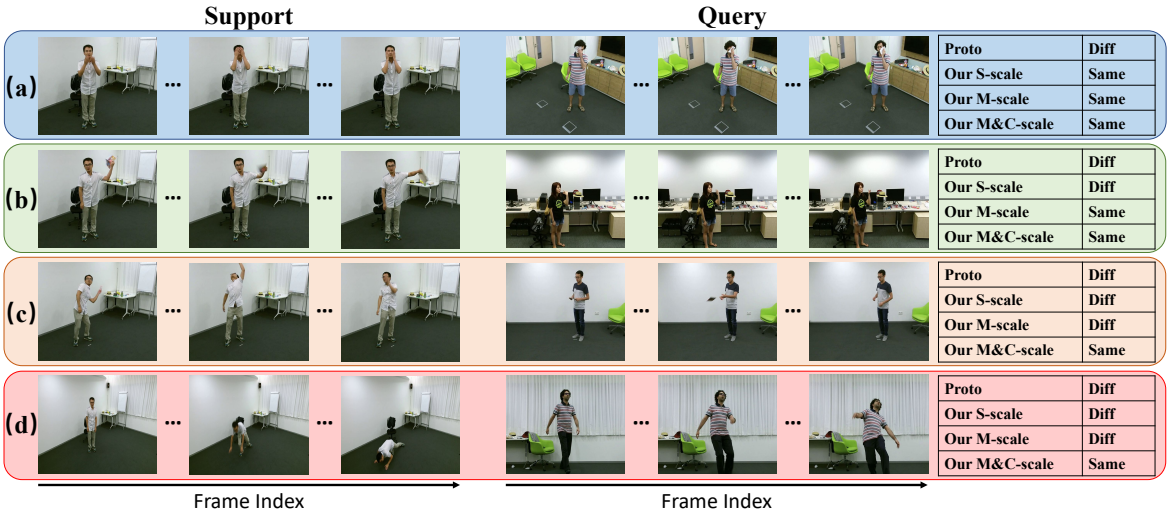

Fig. 8. Visualization of comparison results between different matching strategies. For better visualization, we here show the RGB videos, instead of skeleton sequences. ('Same' means that model predicts the support and query samples are from the same action category; 'Diff' stands for that the predicted action classes for support and query samples are different.)

TABLE 12
Experiments with different feature extractors. The experiments were conducted on NTU RGB+D 120 dataset under the Evaluation Protocol 2.

Backbone	ST-GCN [25]	CTR-GCN [55]	AGCN [24]
ProtoNet [32]	59.2	61.5	60.4
FEAT [33]	59.8	62.1	61.5
S-Scale (Ours)	61.0	63.8	63.2
M-Scale (Ours)	65.1	68.8	67.6
M&C-Scale (Ours)	**66.2**	**70.3**	**68.7**

method (proto) fails in recognition, while our matching-based methods (*e.g.*, '**S-scale**', '**M-scale**', '**M&C-scale**') can recognize the correct category, showing the effectiveness of the matching-based method. As shown in Fig. 8 (b), two 'use a fan' action samples were performed in different ways. The S-scale model fails in recognition since the joint-level representations are different. However, if we focus on the limb level, these two samples all can be seen as 'frequent shaking of arms toward the torso'. Therefore, our multi-scale matching-based methods (*e.g.*, '**M-scale**', '**M&C-scale**') successfully recognize the action category, demonstrating the advantage of our proposed multi-scale matching manner.

Fig. 8 (c) shows two skeleton action samples (belonging to 'throw') were performed at different motion magnitudes, and Fig. 8 (d) shows two samples (belonging to 'falling ') were performed at different speeds. Our '**M&C-scale**' method can still succeed in recognition in the above challenging two situations, showing that our designed cross-scale matching strategy is able to handle the challenging scenarios where the samples of the same action category can be performed at different magnitudes and different motion paces.

APPENDIX C
IMPLEMENTATION DETAILS OF COMPARED FEW-SHOT LEARNING METHODS

The implementation for the pre-training stage is the same as our single-scale setting in the main manuscript. In this section, we focus on the setting for the meta-training stage. To achieve the ***best results*** for those few-shot learning methods [32], [33], [34], [35], we use different settings. There are also slight differences in the settings of the same method on different datasets.

ProtoNet [32]. We utilize the AGCN (a single-scale model, detailed in Figure 5 of the Appendix) as the backbone for ProtoNet. During the meta-learning stage, features for ProtoNet learning are extracted using the pre-trained models, specifically from the global pooled feature of the 9th AGCN block, sized at $1 \times 1 \times C^{(9)}$. Regarding training specifics, for both NTU RGB+D and NTU RGB+D 120 datasets, we set the learning rate to 0.001. For the PKU-MMD dataset, the initial learning rate is set at 0.0005. Across all three datasets, we reduce the learning rate by half every 10 epochs. For every 5-way 1-shot task, there's a single prototype for each class, meaning each meta task encompasses 5 prototypes.

FEAT [33]. We utilize the AGCN (detailed in Figure 5 of the Appendix) as the backbone for FEAT. During the meta-learning stage, features for FEAT learning are extracted using the pre-trained models, specifically from the global pooled feature of the 9th AGCN block, sized at $1 \times 1 \times C^{(9)}$. Diving into the training specifics, across the NTU RGB+D, NTU RGB+D 120, and PKU-MMD datasets, our initial learning rate is set at 0.0005, which we reduce by half every 10 epochs. Additionally, there is a weight value to balance the contrastive term in the learning objective. Here we set the balance value to 0.1 for NTU RGB+D and NTU RGB+D 120 datasets and 0.01 for the PKU-MMD dataset.

Subspace [34]. Similar to ProtoNet and FEAT, we also leverage the single-scale model as the backbone for Subspace and the feature for Subspace model learning is also in the size of $1 \times 1 \times C^{(9)}$. For NTU RGB+D and NTU RGB+D 120 datasets, we set the learning rate to 0.005. For the PKU-MMD dataset, the initial learning rate is set at 0.0005. We cut the learning rate to half every 5 epochs for all three datasets.

Dynamic Filter [35]. Consistent with ProtoNet, FEAT, and Subspace, we employ the single-scale model as the backbone for Dynamic Filter [38]. Notably, while Dynamic Filter [38] operates based on feature maps, the input feature for Dynamic Filter learning is in size of $H^{(9)} \times W^{(9)} \times C^{(9)}$, which represents the feature prior to global pooling. For NTU RGB+D, NTU RGB+D 120, and PKU-MMD datasets, we set the learning rate to 0.05 and cut the rate by half every 10 epochs. There's a weighting factor to balance the few-shot classification and global classification objectives. This balance value is set at 0.2 for NTU RGB+D and NTU RGB+D 120 datasets, and 0.1 for the PKU-MMD dataset.

Experiments for all these methods [32], [33], [34], [35] are optimized using the SGD optimizer with Nesterov momentum (0.9), and the training last for 100 epochs.

TABLE 13
Notations and Definitions.

Notations	Definitions
S	Support set
Q	Query set
D_{train}	Meta-training set
D_{test}	Meta-testing set
$s(\cdot,\cdot)$	The semantic relevance score between two skeleton features
$\mathcal{X}$	Suppliers
$\mathcal{Y}$	Demanders
x_i	The i_{th} supplier
y_j	The j_{th} demander
$OT(\cdot,\cdot)$	The optimal transportation cost between two sets of representations
π	The optimal matching flow between two distributions
r_i	The weight for i_{th} node in suppliers
c_j	The weight for j_{th} node in demanders
d_{ij}	The pair-wise distance between i_{th} supplier and j_{th} demander
$D_{emd}(\cdot,\cdot)$	The Earth Mover's Distance between two feature maps
N	The number of skeleton joints
T	The number of frames

APPENDIX D
NOTATIONS AND DEFINITIONS

We have included a notation table that delineates each important mathematical symbol and its corresponding definition, thereby enhancing clarity for the reader. This table is presented as Tab. 13.

APPENDIX E
SPATIAL POOLING

We adopt 3 spatial scales in our work: the joint-level scale, the part-level scale, and the limb(super-part)-level scale, as shown in Fig. 9. As all three datasets (NTU RGB+D, NTU RGB+D 120, and PKU-MMD) collected skeleton data, which consists of 3D locations of 25 body joints, we consider those 25 skeleton joints for spatial scale 1. Additionally, in spatial scale 2 and spatial scale 3, we consider 10 parts and 6 super-parts, respectively. Pooling details from spatial scale 1 to spatial scale 2 and from spatial scale 2 to spatial scale 3 can be found in Tabs. 14 and 15, respectively.

APPENDIX F
DATASET SPLITTING

NTU RGB+D 120 [1]. We follow the official one-shot setting as described in the NTU RGB+D 120 paper [1]. The action

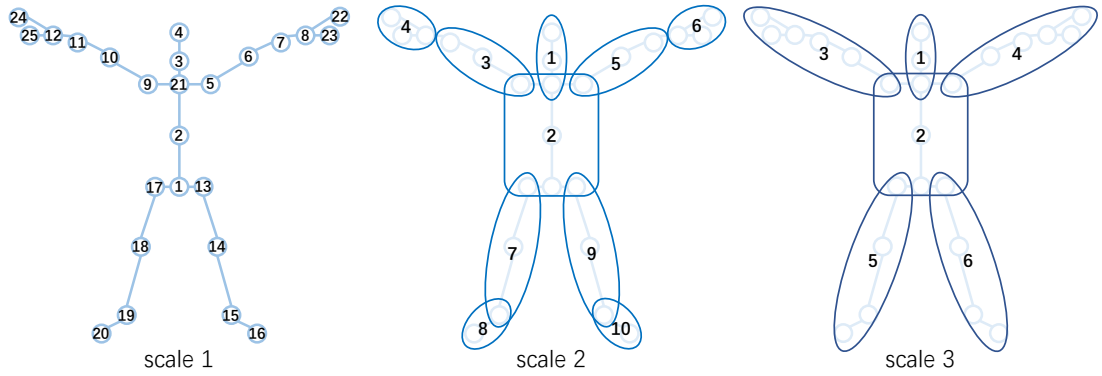

Fig. 9. Three spatial scales on NTU RGB+D, NTU RGB+D 120, and PKU-MMD datasets. In spatial scale 1, we consider 25 skeleton joints. In spatial scale 2 and spatial scale 3, we consider 10 parts and 6 super-parts, respectively.

TABLE 14
The pooling details from spatial scale 1 to spatial scale 2.

Part No.	Name	Joint No.
1	Neck	3, 4, 21
2	Trunk	1, 2, 5, 9, 13, 17
3	Right arm	9, 10, 11
4	Right hand	12, 24, 25
5	Left arm	5, 6, 7
6	Left hand	8, 22, 23
7	Right leg	17, 18, 19
8	Right foot	19, 20
9	Left leg	13, 14, 15
10	Left foot	15, 16

TABLE 15
The pooling details from spatial scale 2 to spatial scale 3.

Super-Part No.	Name	Part No.	Joint No.
1	Neck	1	3, 4, 21
2	Trunk	2	1, 2, 5, 9, 13, 17
3	Right upper limb	3, 4	9, 10, 11, 12, 24, 25
4	Left upper limb	5, 6	5, 6, 7, 8, 22, 23
5	Right lower limb	7, 8	17, 18, 19, 20
6	Left lower limb	9, 10	13, 14, 15, 16

classes of the two sets are distinct, which include 100 classes for training and 20 for testing. The testing set consists of 20 novel classes (i.e. *A1, A7, A13, A19, A25, A31, A37, A43, A49, A55, A61, A67, A73, A79, A85, A91, A97, A103, A109, A115*), and one sample from each novel class is picked as the exemplar.

The following 20 categories are selected: A1 (drink water), A7 (throw), A13 (tear up paper), A19 (take off glasses), A25 (reach into pocket), A31 (pointing to something with finger), A37 (wipe face), A43 (falling), A49 (use a fan (with hand or paper)/feeling warm), A55 (hugging other person), A61 (put on headphone), A67 (hush (quite)), A73 (staple book), A79 (sniff (smell)), A85 (apply cream on face), A91 (open a box), A97 (arm circles), A103 (yawn), A109 (grab other person's stuff), A115 (take a photo of other person).

As suggested by the original dataset paper [1], the following 20 samples are selected as the exemplars: 'S001C003P008R001A001', 'S001C003P008R001A007', 'S001C003P008R001A013', 'S001C003P008R001A019', 'S001C003P008R001A025', 'S001C003P008R001A031', 'S001C003P008R001A037', 'S001C003P008R001A043', 'S001C003P008R001A049', 'S001C003P008R001A055', 'S018C003P008R001A061', 'S018C003P008R001A067', 'S018C003P008R001A073', 'S018C003P008R001A079', 'S018C003P008R001A085', 'S018C003P008R001A091', 'S018C003P008R001A097', 'S018C003P008R001A103', 'S018C003P008R001A109', 'S018C003P008R001A115'.

NTU RGB+D [51]. We select 10 novel classes and 10 exemplars from the NTU RGB+D 120 one-shot setting, of which the action label's no. is smaller than 60, as the novel classes and exemplars for the NTU RGB+D dataset.

The following 10 categories are selected: A1 (drink water), A7 (throw), A13 (tear up paper), A19 (take off glasses), A25 (reach into pocket), A31 (pointing to something with finger), A37 (wipe face), A43 (falling), A49 (use a fan (with hand or paper)/feeling warm), A55 (hugging other person).

The following 10 samples are selected as the exemplars: 'S001C003P008R001A001', 'S001C003P008R001A007', 'S001C003P008R001A013', 'S001C003P008R001A019', 'S001C003P008R001A025', 'S001C003P008R001A031', 'S001C003P008R001A037', 'S001C003P008R001A043', 'S001C003P008R001A049', 'S001C003P008R001A055'.

PKU-MMD [52]. Similarly, we split PKU-MMD dataset into two parts: the training set (41 classes) and the testing set (10 classes). The testing set consists of 10 novel classes, and one sample from each novel class is picked as the exemplar.

The following 10 categories are the novel classes: A1 (bow), A6 (clapping), A11 (falling), A16 (hugging other person), A21 (pat on back of other person), A26 (punching/slapping other person), A31 (rub two hands together), A36 (take off glasses), A41 (throw), A46 (typing on a keyboard).

The following 10 samples are the exemplars: '0003-L_A_1', '0003-L_A_6', '0002-L_A_11', '0005-L_A_16', '0005-L_A_21', '0005-L_A26', '0002-L_A_31', '0003-L_A_36' , '0002-L_A_41', '0003-L_A_46'.

The videos in the PKU-MMD dataset are untrimmed, so we need to trim videos to the one-action segment level based on the given starting time and ending time. While the videos' filenames contain only the part before the first '_' of

exemplars' filenames, take the '0003-L_A_1' as an example, the original filename is '0003-L'. Since we trim the video, we add the action category number in the filename, here 'A_1' in '0003-L_A_1' means the corresponding segment of action category **1** in the video '0003-L'.