Supplementary Material for Compound Prototype Matching for Few-shot Action Recognition

Yifei Huang[0000−0001−8067−6227], Lijin Yang[0000−0003−3787−6658], and Yoichi Sato[0000−0003−0097−4537]

Institute of Industrial Science, the University of Tokyo, Tokyo, Japan
{hyf,yang-lj,ysato}@iis.u-tokyo.ac.jp

1 Additional Experiment and Analysis

1.1 Many shot results

Due to the space limit, we put the standard 5-shot 5-way action recognition result here in Table 1. We can see similar result comparison with Table 1 of the main submission, indicating the superior performance of our method. Thus, similar conclusions as the main submission can be derived.

Table 1. Result comparison of 5-way 5-shot experiments on 5 dataset splits. Methods marked with * indicates results of our implementation with the original reported results shown in parenthesis. The bottom and upper block are results with and without object features, respectively.

Method	SSv2°	SSv2♯	Kinetics	HMDB	UCF
CMN [7]	48.8	-	78.9	-	-
ARN [4]	-	-	82.4	60.6	83.1
OTAM [1]	-	52.3	85.8	-	-
TRN* [6,1]	46.7	49.5 (48.9)	82.2 (82.0)	70.2	85.4
ITA-Net* [5]	52.2	63.0	84.3	75.8	93.7
TRX* [2]	59.3 (59.1)	64.5 (64.6)	85.9 (85.9)	75.3 (75.6)	**96.0 (96.1)**
CPMT (Ours)	**61.6**	**66.7**	**86.4**	**77.0**	91.0
TRN+*	47.6	51.0	85.1	72.5	88.3
ITA-Net+*	55.0	67.2	85.5	76.8	95.5
TRX+*	62.9	66.8	87.3	78.4	**96.5**
CPMT (Ours)	**69.0**	**73.5**	**87.9**	**85.1**	92.3

1.2 Additional visualization

We place another visualization figure in Fig. 1. We can see similar results in the main submission. In the example to the left, focused prototypes $\boldsymbol{p}^a_{f,2}$ and $\boldsymbol{p}^b_{f,1}$

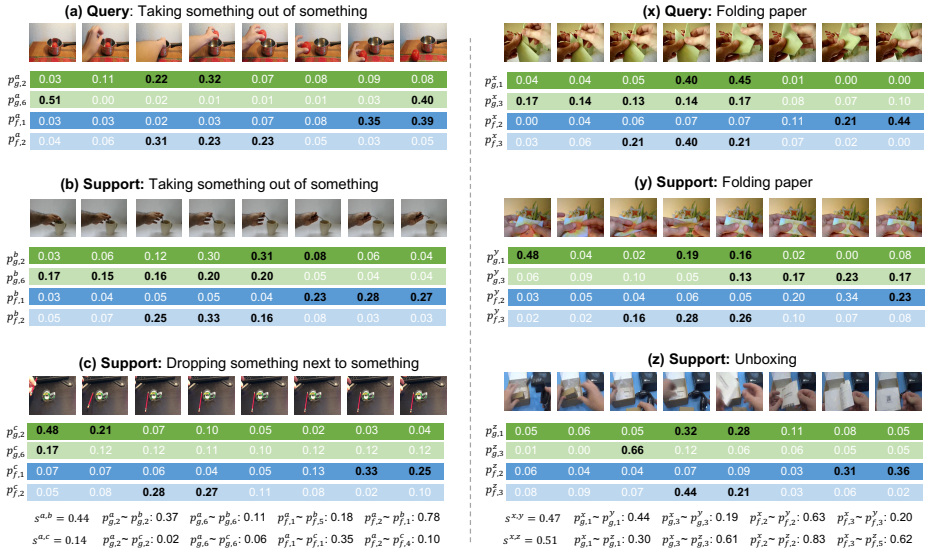

Fig. 1. Visualization of the self-attention weight of two global prototypes and two focused prototypes on each timestamp of the input. Attention weights higher than average (0.125) are marked in black. Example to the left comes from the SSv2♯ dataset and the example to the right (False positive) is from Kinetics. Video similarity scores s and similarity scores of matched prototypes $\boldsymbol{p}_* \sim \boldsymbol{p}_*$ are shown at the bottom.

are matched and high similarity score (0.78) is given. From the figure, we can see that $\boldsymbol{p}^a_{f,2}$ has high attention on the middle frames, and $\boldsymbol{p}^b_{f,1}$ focus more on the ending frames. The focus of both prototypes capture the moment when the object is being taken out of the container, thus high similarity score is given. In the example to the right, our model wrongly classifies the query action of "folding paper" as "unboxing". This is because $\boldsymbol{p}^x_{f,2}$ and $\boldsymbol{p}^z_{f,2}$ give high similarity. In fact, when we manually inspect the unboxing action in video z, we found that in the last half of the video, the video recorder was reading a book taken out of the box. Since reading a book contains actions similar to "folding paper", we think this may be the main reason that our method gives wrong prediction.

1.3 Analysis of bipartite matching

In the main submission, we introduced our method with includes two groups of prototypes, where the focused prototypes are matched via bipartite matching. Since bipartite matching matches each support prototype to a query prototype, there will inevitably exist wrong matchings. However, we found that these wrong matchings are necessary for training the model. Since our decoder is based on a transformer architecture, each prototype is generated by attending to all the frames. With the positional encoding, the generated prototypes implicitly contains the temporal ordering of the whole video, but with different emphasizes

Table 2. Result comparison of 5-way 1-shot experiments when using different thresholds to filter the matched focused prototypes. "best" means we only use the best matched focused prototype pair and filter all others. "average" indicates only keeping the matched prototype pairs whose similarity scores are above average.

Method	Matching	p	SSv2$^\sharp$	Kinetics
CPMT (Ours)	best	0	23.5	59.2
CPMT (Ours)	average	0	24.3	63.6
CPMT (Ours)	best	0.5	29.3	59.7
CPMT (Ours)	average	0.5	30.6	65.4
CPMT (Ours)	all	N/A	**59.6**	**81.0**

(*e.g.*, a prototype will look at all frames but emphasize only some frames). Thus, through training, the optimization process will encourage the correct matchings to generate high similarity scores, and keep the similarity scores of the wrong matchings low.

To test whether filtering out subset of the matchings will also get good results, we conduct the following experiment with different methods to filter the focused prototypes. Specifically, since each prototype is not necessarily guaranteed to have its correct matching, we filter the matched prototypes by the following thresholds: 1) choose only the best match as the overall similarity score of two videos, 2) choose only the matchings with similarity scores greater than average. To allow the gradient flow over the non-matched prototypes, we use a leaky relu-like method to suppress the scores of the non-matched prototypes by:

$$\text{score} = \begin{cases} \text{score} & \text{if matching meets threshold} \\ p * \text{score} & \text{otherwise} \end{cases} \tag{1}$$

Here if $p = 0$, matchings that do not meet the threshold are discarded. If $0 < p < 1$, these matchings are suppressed. We equip this filtering method and test it with different values of p and show the results in Table 2.

From Table 2, we can see that suppressing ($p = 0.5$) or ignoring ($p = 0$) a subset of matched prototypes cannot perform well, since we find it is even hard for model to fit on the training set. This proves our claim that the wrong matchings are essential to train our model.

1.4 Class improvement of multi-relation encoding

In the main submission, we show the class improvement when using both groups of prototypes compared with using only one single group of prototype. These experiments are done using $m_g = m_f = 8$. Additionally, to see the impact of each different relation in our multi-relation encoder, we show the class improvement when encoding all 3 relations (global-global, global-object, object-object) compared with encoding only one of the 3 relations. In Figure 2, yellow bars show the class accuracy difference between encoding all relations and encoding

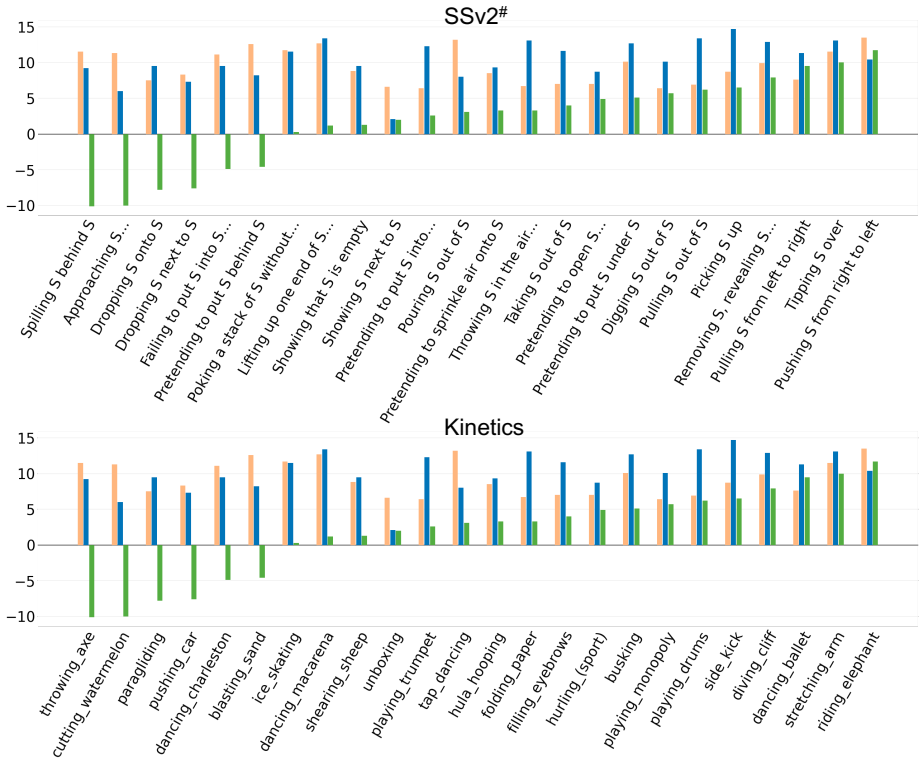

Fig. 2. Class accuracy improvement when encoding 3 relations compared with encoding only one relation. The colors indicate the improvement when compared with: global-global only (yellow), global-object only (blue), object-object only (green).

only global-global relation. Blue and green bars indicate the difference between encoding all relations and encoding only global-object relation, encoding object-object relation, respectively. We can see that generally, object-object relation can greatly help to improve the classification performance, while for classes like "tipping something over", "pushing something from right to left" and "pulling something from left to right", benefit from the combination of all the three relations.

1.5 Pretraining using semantic labels

To compare with most of previous works, we do not use pretraining on the backbone encoder, but directly use ImageNet pretrained ResNet50 as the backbone [1]. Meanwhile, many recent works have shown the usefulness of conducting pretraining using the semantic labels of the training set [8,3,5]. Thus, for a more complete comparison, we conduct another set of experiments where we first pretrain the backbone network on the training dataset using their semantic labels,

following PAL [8]. We show the result comparison in Tab. 3. From this setting we can still see the superiority of our proposed method.

Table 3. Result comparison of 5-way 1-shot experiments on 4 datasets. In this table, the backbone network is first pretrained using the semantic labels of the training set. Results of previous methods are the authors' reported results in each paper.

Method	SSv2$^\sharp$	Kinetics	UCF	HMDB
SRPN [3]	-	75.2	86.5	61.6
PAL [8]	46.4	74.2	85.3	60.9
ITA-Net [5]	49.2	73.6	-	-
CPMT-single (Ours)	**53.3**	**75.4**	85.1	**62.3**
CPMT-full (Ours)	**61.0**	**85.2**	**88.1**	**85.0**

2 Details for Training and Baseline Implementation

2.1 Model training

For each dataset, we use slightly different training parameters. Specifically, for the Kinetics and UCF datasets, we fix the backbone encoder. For other datasets, we do not fix the full backbone encoder, but fix all the batch normalization layers except the first one. When training the model with backbone, the learning rate of the backbone network is multiplied by 10. For the Kinetics dataset alone, we set the hyperparameter $\lambda_1 = 0.1$.

2.2 Baseline implementation details

As stated in the main submission, we give the baseline methods the same input as our model for a fair comparison (denoted as "method+"). Here we describe more details for the implementation of these baselines.

For TRX+, the original TRX generate frame-wise tuples with different cardinalities. We extend this tuple construction on the B object features each frame. To construct pairwise object tuples of frames i, j, we construct $B \times B$ object tuples by associating each object feature of frame i with all object features of frame j. Triplet tuples are constructed in the same fashion. As in TRX, the tuples are all considered together to generate the class-specific prototypes. For ITA-Net+, the implicit temporal alignment is used to align both frame-wise and object features, so the input is the concatenated feature $\boldsymbol{F}_{input} \in \mathbb{R}^{(B+2)T \times d}$. The code will be made publicly available including the implementation of baselines.

References

1. Cao, K., Ji, J., Cao, Z., Chang, C.Y., Niebles, J.C.: Few-shot video classification via temporal alignment. In: CVPR (2020) 1, 4
2. Perrett, T., Masullo, A., Burghardt, T., Mirmehdi, M., Damen, D.: Temporal-relational crosstransformers for few-shot action recognition. In: CVPR (2021) 1
3. Wang, X., Ye, W., Qi, Z., Zhao, X., Wang, G., Shan, Y., Wang, H.: Semantic-guided relation propagation network for few-shot action recognition. In: ACM MM (2021) 4, 5
4. Zhang, H., Zhang, L., Qi, X., Li, H., Torr, P.H., Koniusz, P.: Few-shot action recognition with permutation-invariant attention. In: ECCV (2020) 1
5. Zhang, S., Zhou, J., He, X.: Learning implicit temporal alignment for few-shot video classification. IJCAI (2021) 1, 4, 5
6. Zhou, B., Andonian, A., Oliva, A., Torralba, A.: Temporal relational reasoning in videos. In: ECCV (2018) 1
7. Zhu, L., Yang, Y.: Label independent memory for semi-supervised few-shot video classification. TPAMI (2020) 1
8. Zhu, X., Toisoul, A., Perez-Rua, J.M., Zhang, L., Martinez, B., Xiang, T.: Few-shot action recognition with prototype-centered attentive learning. BMVC (2021) 4, 5