Rethinking matching-based few-shot action recognition

Juliette Bertrand [1] Yannis Kalantidis[2] Giorgos Tolias[1]
[1]VRG, FEE, Czech Technical University in Prague
[2]NAVER LABS Europe, Grenoble, France
{bertrjul,toliageo}@fel.cvut.cz {yannis.kalantidis}@naverlabs.com

Abstract

Few-shot action recognition benefits from incorporating temporal information. Prior work either encodes such information in the representation itself and learns classifiers at test time or obtains frame-level features and performs pairwise temporal matching. We first evaluate several recent matching-based approaches using features from spatio-temporal backbones, a comparison missing from the literature, and show that the gap in performance between simple baselines and more complicated methods is significantly reduced. Inspired by this, we propose Chamfer++, a non-temporal matching function that achieves state-of-the-art results in few-shot action recognition.

1. Introduction

Recognizing actions within videos is essential for analyzing trends, enhancing broadcasting experience, or filtering out inappropriate content. However, collecting and annotating enough video examples to train supervised models can be prohibitively time-consuming. It is therefore desirable to recognize new action classes with as few labeled examples as possible. This is the premise behind the task of few-shot learning, where models learn to adapt to a set of unseen classes for which only a few examples are available. In video action recognition, additional challenges arise from the temporal dimension. Recognition methods need to capture the scene's temporal context and temporal dynamics.

One family of approaches is formed by *matching-based* methods [1, 2, 10, 13, 19–21] where each test example or "query" is compared against all support examples of a class to infer a class confidence score. Most existing matching-based methods use frame-level representations, *i.e*. a 2D convolutional backbone that takes a frame as input, and a feature set is formed by encoding multiple frames. Feature extraction is followed by matching the query feature set Q to the support example set X, and a similarity $s_{Q,X}$ between the two is computed. Although each feature represents an individual frame and cannot capture temporal information, the feature sets are usually temporal sequences, and the matching process can exploit such information.

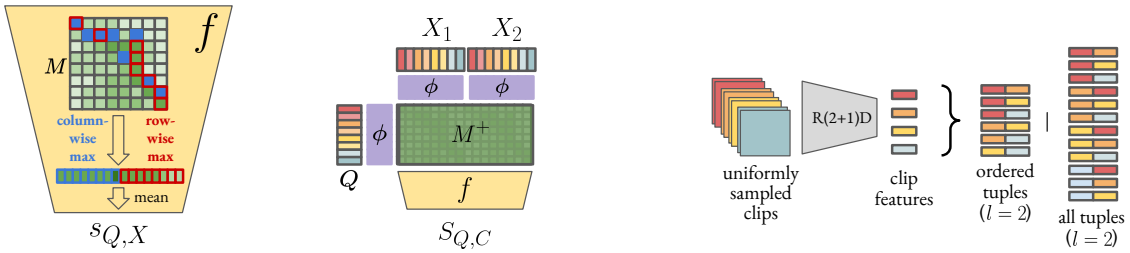
Another family of approaches is formed by methods that learn a conventional linear *classifier* at test time [18, 22], *i.e*. using the handful of examples available. In this case, any temporal context has to be incorporated in the representation. As a representative example, Xian *et al*. [18] adopt the spatio-temporal R(2+1)D architecture [14], where the input is a video *clip*, *i.e*. a sequence of consecutive frames, and convolutions across the temporal dimension enable the features to encode temporal information. Following findings in few-shot learning [3,16], Xian *et al*. further abandon episodic training and instead fine-tune a pre-trained backbone using all training examples of the base classes. Using strong temporal features and by simply learning a linear classifier at test time, they report state-of-the-art results for few-shot action recognition.

Motivated by the two families of approaches presented above, we introduce a new setup that aims at answering the following questions:
1. Do matching-based methods still have something to offer for few-shot action recognition given strong temporal representations? To that end, we level the playing field with respect to representations and evaluate a number of recent matching-based approaches using strong temporal representations. We find that such approaches perform better than training a classifier at test time.
2. Is temporal matching necessary when the features capture temporal information? We show that matching-based methods invariant to the temporal order in the feature sequence (*non-temporal matching*) are performing as good as the ones that do use it (*temporal matching*) on many common benchmarks.

Inspired by the findings above, we further introduce **Chamfer++**, a novel, parameter-free matching approach that employs Chamfer matching and is able to achieve a new state-of-the-art for one-shot action recognition on three common benchmarks.

2. Method

Video representation A clip c_i is a sequence of L consecutive RGB frames in the form of a $L \times H \times W \times 3$ tensor.



(a) Chamfer matching (b) Joint matching (c) Clip tuples

Figure 1. **Details of the proposed matching approach**. a) The **Chamfer** matching function f_{QS} from Eq.(3). b) Jointly matching multiple examples per class (Chamfer+). c) using clip tuples as representation; one can use only ordered tuples or all tuples; in both cases, the matching part remains *non-temporal*, *i.e.* invariant to the temporal order of features.

A deep video backbone b takes a clip c_i as input and maps it to a d-dimensional vector $\mathbf{q}_i$. We use the R(2+1)D backbone architecture [14] as in the work of Xian *et al*. [18], which uses efficient and effective separated spatio-temporal convolutions. A video Q is represented by a set of clip features. The clips are uniformly sampled over the temporal dimension, with possible overlap.

Episodic protocol We adopt the commonly used setup [2, 10]. The classes of the train and test sets are non-overlapping. To simulate the limited annotated data, test episodes are randomly sampled from the test set. Each episode corresponds to a different classification task and comprises query and support examples for a fixed set of classes, where labels of support examples are known, while labels of query examples are unknown. Only k labeled examples per class, also named shots, are available in the support set, with k typically ranging from 1 to 5. The performance is evaluated via classification accuracy on the query examples averaged over all test episodes.

2.1. Classifier and matching-based approaches

We identify two dominant families of approaches proposed in the recent few-shot action recognition literature: the classifier-based and the matching-based methods [1]. Unfortunately, discrepancies in setup and architecture between existing approaches make it difficult to compare them fairly. We propose to follow the same representation learning strategy and start from a common frozen backbone.

Classifier-based approaches The classifier-based approaches [18, 22] train the video representation using a classifier in the form of a linear layer. This is similar to the corresponding work on few-shot learning in the image domain [4]. Xian *et al*. [18] depart from episodic training and propose a *Two-stage Learning* (*TSL*) process. During the first stage, a R(2+1)D video backbone and a classifier are learned jointly using all the labeled examples of the train set. During the second stage, the backbone remains fixed to avoid overfitting, and a newly initialized classifier needs to be trained per test episode using the support examples. In both stages, a linear classifier with a soft-max function denoted by $h : \mathbb{R}^d \to \mathbb{R}^C$, where C is the number of classes, is added to the output of the backbone. Training is performed by optimizing the class probabilities with the cross-entropy loss ($\mathcal{L}_{cls}$), while inference is performed by sum-pooling of the classifier output across clips.

Matching-based approaches The matching-based approaches estimate the similarity between the query and all the support examples of each class to obtain class probabilities. Training is performed on episodes sampled from the train set, which are meant to imitate the episodes of the test set. Let Q and X be two videos with $|Q| = |X| = n$ constant across videos. We form the *temporal similarity matrix* for the ordered video pair (Q, X) denoted by M, with elements $m_{ij} = \phi(\mathbf{q}_i)^\top \phi(\mathbf{x}_j)$, where $\phi : \mathbb{R}^d \to \mathbb{R}^D$ is a learnable projection head. The function ϕ consists of a linear layer, a layer normalization, and a ℓ_2 normalization to guarantee bounded similarity values m_{ij}. We consider the family of matching approaches that infer a video-to-video similarity $S_{Q,X} = f(M)$, with $f : \mathbb{R}^{n \times n} \to \mathbb{R}$, named the *matching function*. By definition, the result of function f only depends on the strength and position of the pairwise similarities m_{ij} and does not directly depend on the features themselves. The function f can either be hand-crafted or include learnable parts. Some matching functions use temporal information by leveraging the position, either absolute or relative, of the pairwise similarities m_{ij}. The matching functions that use temporal information are called *temporal*, whereas the others are called *non temporal*. The pairwise video-to-video similarities between query and support examples are averaged per class to obtain class probabilities. During training, the class probabilities are optimized with the cross-entropy loss ($\mathcal{L}_{cls}$). Inference is also performed by estimating the similarity between query and all support examples. This is a form of a k-nearest-neighbor classifier.

A common starting point We follow the training stage of TSL to learn the R(2+1)D backbone parameters using all the annotated examples. We freeze this backbone, and treat the resulting model as a feature extractor. This is our starting point for both classifier-based and matching-based approaches which enable us to fairly compare the two families of approaches. Specifically, matching-based methods

[1] Prototypical networks can be seen as an extension of matching based methods [10, 13, 21]. Hence we group them with the matching-based family.

only learn the feature projection function ϕ and the matching parameters when needed in a test-agnostic way. Unlike classifier-based methods, which need to train a classifier for every testing episode, matching-based approaches require no learning or adaptation at test time. They only need the pairwise matching between the query and each one of the support examples.

2.2. Chamfer++

In this section, we introduce a new matching function, Chamfer++, which is non temporal and achieves top performance while being parameter-free and intuitive.

Chamfer The Chamfer matching function f_Q is given by

$$f_Q(M) := \frac{1}{n}\sum_i \max_j m_{ij}. \quad (1)$$

It implies that each clip sampled from the query example contributes to the similarity score by matching its closest clip in the support example. One can transpose the temporal similarity matrix and derive the symmetric process where each clip from the support example needs to match a query clip. Then, the matching function becomes

$$f_S(M) := \frac{1}{n}\sum_j \max_i m_{ij}. \quad (2)$$

Summing the two gives a symmetric Chamfer variant, where all clips from both the query and the support example are required to match:

$$f_{QS}(M) := f_Q(M) + f_S(M). \quad (3)$$

We refer to this symmetric variant as simply *Chamfer matching* in the context of few-shot action recognition.

Joint-matching The standard option to compute the query-to-class probabilities is by averaging the pairwise similarity score between the query example and all the support examples belonging to the class. Instead, we propose to match all support examples jointly. We concatenate the temporal similarity matrices between the query and all support examples. It creates the joint temporal similarity matrix M^+, with $M^+ \in \mathbb{R}^{n\times kn}$. Then, we compute the matching function on top of M^+ to obtain video-to-class similarity $S_{Q,c} := s(M^+)$.

Clip tuples Inspired by Perret *et al*. [10], we extend Chamfer to enable matching of *clip* feature tuples formed by any clip subset of fixed length l. A clip feature tuple $\mathbf{t}^l$ contains l clip features, non-necessarily consecutive, but with the same relative order as in $Q = \{\mathbf{q}_i\}$. For example, $\mathbf{t}^2 = \{(q_i, q_{j>i})\}$. Each clip feature tuple is concatenated and fed to the learnable projection head $\phi : \mathbb{R}^{ld} \to \mathbb{R}^D$. The resulting temporal similarity matrix is $M^{++} \in \mathbb{R}^{n'\times kn'}$, with $n' = \binom{n}{l}$. The clip tuples are sub-sequence representations on top of single-clip representations. By definition, clip tuples add additional temporal information to the representation. But the matching function remains non-temporal.

We also define non-temporally ordered clip tuples as the permutations of l clip features. The resulting temporal similarity matrix is $M^{++} \in \mathbb{R}^{n'\times kn'}$ with $n' = n!$. Unless otherwise stated, we use ordered clip tuples.

3. Experiments

We report results on the three most commonly used benchmarks for few-shot action recognition, *i.e*. Kinetics-100 [20], Something-Something V2 (SS-v2) [5], and UCF-101 [11]. We use the train/val/test splits from [20] for all three datasets, containing 64/12/24 classes, respectively. We learn the parameters of the R(2+1)D backbone using the train split, similar to [18]. For matching-based approaches, we learn the projection ϕ together with any learnable parameters in the matching function f using episodic training also on the train split. We use the val split for hyper-parameter tuning and early stopping.

We evaluate on the common 1-shot and 5-shot setups. Unless otherwise stated, we use 5-way classification tasks. Training episodes are randomly sampled from the train set. We use the same fixed, predefined set of 10k test episodes sampled from the test set for all methods (prior work randomly samples them each time [2, 10, 22]). We always evaluate *three* trained models and report mean.

Frame or clip-based features? In Figure 3, we report one-shot performance for classifier-based and matching methods using both frame-based (ResNet, blue points) and clip-based (R(2+1)D, orange points) features. We clearly see that using a spatio-temporal backbone significantly boosts accuracy by more than 10% on all datasets. Even for the Kinetics-100 and the UCF-101 datasets that are known to be

Method	SS-v2		Kinetics-100		UCF-101	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Parametric classifier						
TSL [18]	60.6	79.9	93.6	98.0	97.1	99.4
Non-temporal matching						
Mean	65.8	79.1	95.5	98.1	97.6	98.9
Max	65.0	79.0	95.3	98.3	**97.9**	98.9
Chamfer++†	67.0	80.8	**96.2**	**98.4**	97.8	99.2
Chamfer++	**67.8**	81.6	96.1	98.3	97.7	99.3
Temporal matching						
Diagonal	66.7	80.1	95.3	98.1	97.6	99.0
Linear	66.6	80.1	95.5	98.1	97.6	98.9
OTAM [2]	67.1	80.2	95.9	**98.4**	97.8	99.0
TRX-{2,3} [10]	65.5	**81.8**	93.4	97.5	96.6	**99.5**
ViSiL [7]	67.7	81.3	95.9	98.2	97.8	99.0

Table 1. **Few-shot action recognition results**. All methods are trained and evaluated by us and use the same R(2+1)D [14] backbone. TSL [18] learns a classifier at each episode during testing. All the rest are pairwise matching-based methods and are split into two categories: non-temporal, *i.e*. invariant to the temporal order of features, and temporal. † denotes Chamfer++ matching using *all* tuples. Best (second-best) results are presented in **bold** (underlined).

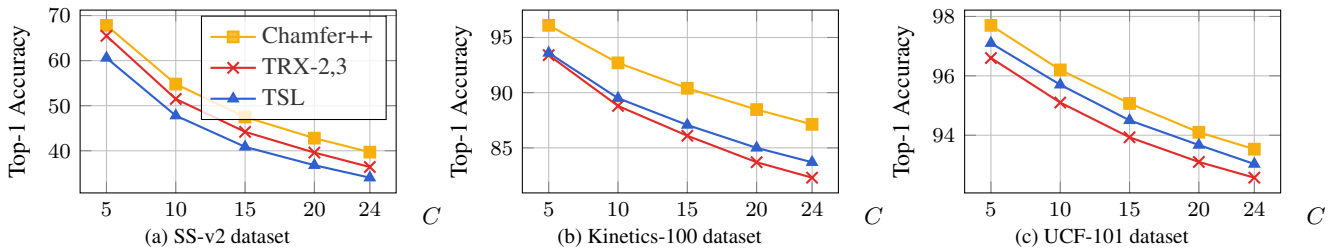


Figure 2. **Impact of the number of classes per episode** (C) on three datasets.

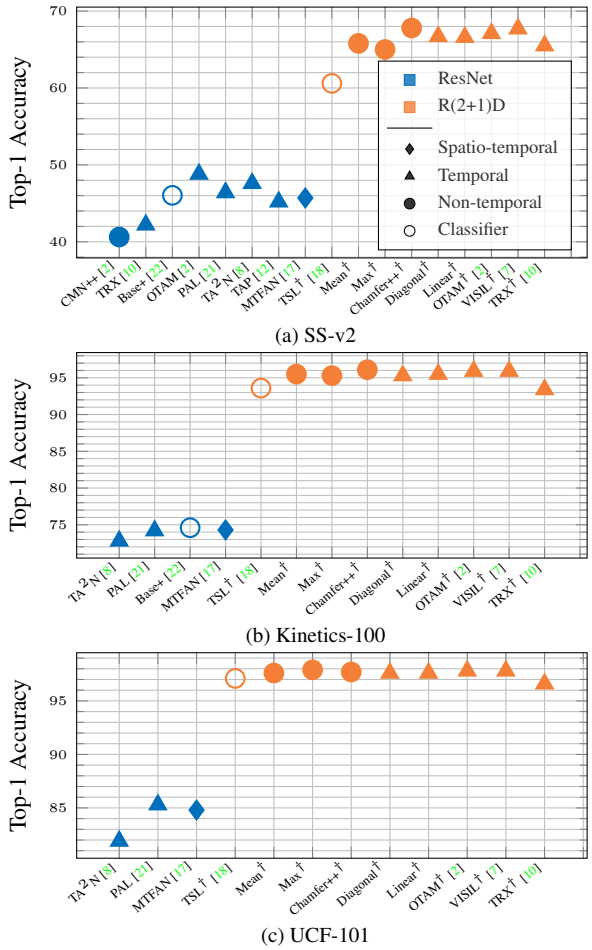


Figure 3. **One-shot performance** for different backbones, types of matching, or use of parametric classifiers. The different colors account for the different backbones. The different shapes account for the type of matching. † denotes methods reproduced in this study.

more biased towards spatial context [6], temporal dynamics remain a valuable cue for few-shot action recognition.

Pairwise matching or classifiers? Matching-based methods and classifiers are compared in Table 1 under a common framework, *i.e.* all methods share representations from an R(2+1)D backbone and are tested on the same set of episodes. For all setups and datasets, several matching approaches outperform TSL. In the 1-shot regime, *most* matching-based methods outperform TSL.

How useful is temporal matching? No significant difference in performance is observed between temporal and non-temporal matching approaches, as highlighted in Figure 3.

Method	Clip	SS-v2		Kinetics-100		UCF-101	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Classifier-based							
Baseline [22]		40.8	59.2	69.5	84.4	-	-
Baseline + [22]		46.0	61.1	74.6	86.6	-	-
TSL [18]	✓	59.1	80.1	92.5	97.8	94.8	-
TSL ‡	✓	60.6	79.9	93.6	98.0	97.1	**99.4**
Matching-based							
CMN++ † [20, 22]		40.6	51.9	65.9	82.7	-	-
ARN [19]	✓	-	-	63.7	82.4	66.3	83.1
OTAM [2]		48.8	52.3	73.0	85.8	-	-
PAL [21]		46.4	62.6	74.2	87.1	85.3	95.2
TRX-{2,3} [10]		42.0	64.6	63.6	85.9	-	96.1
STRM-{2} [13]		-	70.2	-	91.2	-	98.1
TA^{2}N [8]		47.6	61.0	72.8	85.8	81.9	95.1
MTFAN [17]		45.7	60.4	74.6	87.4	84.8	95.1
HyRSM [15]		54.3	69.0	73.7	86.1	-	-
TAP [12]		45.2	63.0	-	-	83.9	95.4
CPM [9]		59.6	-	81.0	-	79.0	-
Chamfer++ ‡	✓	**67.8**	**81.6**	**96.1**	**98.3**	**97.7**	99.3

Table 2. **Comparison with the state-of-the-art.** Unless otherwise stated, we report results as presented in the corresponding papers. † denotes results reproduced in [22] and ‡ denotes results generated by us. Best (second-best) results are presented in **bold** (underlined).

Varying the number of classes in an episode. Figure 2 shows the performance when extending the case of 5 classes to the maximum number of classes, $C = 24$, in the 1-shot regime. The proposed Chamfer++ method outperforms TRX and the classifier-based TSL method in all datasets.

Comparison to the state-of-the-art. In Table 2, we compare the performance of the proposed Chamfer++ with the corresponding numbers reported in many recent few-shot action recognition papers.

4. Conclusion

A number of recent few-shot learning papers are abandoning the meta-learning protocol for representation learning [3, 16, 18, 22] and show that adaptation from the best possible features leads to better performance. In the quest for rapid adaptation at test time, we also adopt this setup and show that, given strong visual representations, simple matching-based methods are really effective and able to beat both classifier-based and more complex matching-based approaches on many common benchmarks for few-shot action recognition. We further show that temporal information in the matching provides no particular benefit compared to the ability to learn or adapt from strong temporal features, and introduce an intuitive and parameter-free matching-based method.

Acknowledgements. This work was supported by Naver Labs Europe, by Junior Star GACR GM 21-28830M, and by student grant SGS23/173/OHK3/3T/13. The authors would like to sincerely thank Toby Perrett and Dima Damen for sharing their early code and supporting us, Diane Larlus for insightful conversations, feedback, and support, and Zakaria Laskar, Monish Keswani, and Assia Benbihi for their feedback.

References

[1] Mina Bishay, Georgios Zoumpourlis, and I. Patras. Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition. In *BMVC*, 2019. 1

[2] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-shot video classification via temporal alignment. In *CVPR*, 2020. 1, 2, 3, 4

[3] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang. A closer look at few-shot classification. In *ICLR*, 2019. 1, 4

[4] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *CVPR*, 2018. 2

[5] R. Goyal, S. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag, F. Hoppe, C. Thurau, I. Bax, and R. Memisevic. The "something something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 3

[6] De-An Huang, Vignesh Ramanathan, Dhruv Mahajan, Lorenzo Torresani, Manohar Paluri, Li Fei-Fei, and Juan Carlos Niebles. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In *CVPR*, 2018. 4

[7] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, Ioannis Patras, and Ioannis Kompatsiaris. Visil: Fine-grained spatio-temporal video similarity learning. In *ICCV*, 2019. 3, 4

[8] Shuyuan Li, Huabin Liu, Rui Qian, Yuxi Li, John See, Mengjuan Fei, Xiaoyuan Yu, and Weiyao Lin. Ta2n: Two-stage action alignment network for few-shot action recognition. In *AAAI*, 2022. 4

[9] Yifei Huang Lijin Yang and Yoichi Sato. Compound prototype matching for few-shot action recognition. In *ECCV*, 2022. 4

[10] Toby Perrett, Alessandro Masullo, Tilo Burghardt, Majid Mirmehdi, and Dima Damen. Temporal-relational crosstransformers for few-shot action recognition. In *CVPR*, 2021. 1, 2, 3, 4

[11] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012. 3

[12] Bing Su and Ji-Rong Wen. Temporal alignment prediction for supervised representation learning and few-shot sequence classification. In *ICLR*, 2022. 4

[13] Anirudh Thatipelli, Sanath Narayan, Salman Khan, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Bernard Ghanem. Spatio-temporal relation modeling for few-shot action recognition. In *CVPR*, 2022. 1, 2, 4

[14] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 1, 2, 3

[15] Xiang Wang, Shiwei Zhang, Zhiwu Qing, Mingqian Tang, Zhengrong Zuo, Changxin Gao, Rong Jin, and Nong Sang. Hybrid relation guided set matching for few-shot action recognition. In *CVPR*, 2022. 4

[16] Yan Wang, Wei-Lun Chao, Kilian Q Weinberger, and Laurens van der Maaten. Simpleshot: Revisiting nearest-neighbor classification for few-shot learning. *arXiv preprint arXiv:1911.04623*, 2019. 1, 4

[17] Jiamin Wu, Tianzhu Zhang, Zhe Zhang, Feng Wu, and Yongdong Zhang. Motion-modulated temporal fragment alignment network for few-shot action recognition. In *CVPR*, 2022. 4

[18] Y. Xian, B. Korbar, M. Douze, L. Torresani, B. Schiele, and Z. Akata. Generalized few-shot video classification with video retrieval and feature generation. *IEEE TPAMI*, 2021. 1, 2, 3, 4

[19] Hongguang Zhang, Li Zhang, Xiaojuan Qi, Hongdong Li, Philip H. S. Torr, and Piotr Koniusz. Few-shot action recognition with permutation-invariant attention. In *ECCV*, 2020. 1, 4

[20] Linchao Zhu and Yi Yang. Compound memory networks for few-shot video classification. In *ECCV*, 2018. 1, 3, 4

[21] Xiatian Zhu, Antoine Toisoul, Juan-Manuel Pérez-Rúa, Li Zhang, Brais Martinez, and Tao Xiang. Few-shot action recognition with prototype-centered attentive learning. In *BMVC*, 2021. 1, 2, 4

[22] Zhenxi Zhu, Limin Wang, Sheng Guo, and Gangshan Wu. A closer look at few-shot video classification: A new baseline and benchmark. In *BMVC*, 2021. 1, 2, 3, 4