Action Knowledge Transfer for Action Prediction with Partial Videos

Yijun Cai,[1] Haoxin Li,[1] Jian-Fang Hu,[2] Wei-Shi Zheng[2,3]*
[1]School of Electronics and Information Technology, Sun Yat-sen University, China
[2]School of Data and Computer Science, Sun Yat-sen University, China
[3]The Key Laboratory of Machine Intelligence and Advanced Computing (Sun Yat-sen University), Ministry of Education
caiyj6@mail2.sysu.edu.cn, lihaoxin05@gmail.com, hujf5@mail.sysu.du.cn, wszheng@ieee.org

Abstract

Predicting action class from partially observed videos, which is known as action prediction, is an important task in computer vision field with many applications. The challenge for action prediction mainly lies in the lack of discriminative action information for the partially observed videos. To tackle this challenge, in this work, we propose to transfer action knowledge learned from fully observed videos for improving the prediction of partially observed videos. Specifically, we develop a two-stage learning framework for action knowledge transfer. At the first stage, we learn feature embeddings and discriminative action classifier from full videos. The knowledge in the learned embeddings and classifier is then transferred to the partial videos at the second stage. Our experiments on the UCF-101 and HMDB-51 datasets show that the proposed action knowledge transfer method can significantly improve the performance of action prediction, especially for the actions with small observation ratios (e.g., 10%). We also experimentally illustrate that our method outperforms all the state-of-the-art action prediction systems.

Action prediction is an important computer vision problem with many real-world applications. For example, in the traffic system, it is greatly expected that accidents can be predicted at earlier stages. Also, the computational resource can be saved if actions can be recognized from partial observations without processing the whole videos. Compared with the rapid progress in video action recognition (Simonyan and Zisserman 2014; Wang et al. 2016; Carreira and Zisserman 2017), the advance in action prediction is still unsatisfying, especially when the actions are observed at very early stage (e.g., 10%). In principal, the existing action recognition systems can be directly used for action prediction by treating partial videos as full videos. However, these models typically perform poorly for action prediction, since they are not specifically developed for mining action information from partial videos.

The main challenge for action prediction is that partially observed videos often contain incomplete action executions, and thus have less action information than the fully observed ones. These partially observed videos are more likely to be confused between different action classes. Take Figure 1 for example. From the partially observed video in Figure 1, it is not clear whether the boy was kicking a ball on the grass or doing somersault. To improve the performance for action prediction, previous works mainly focus on improving the discriminative power of partial videos by developing max margin learning (Kong and Fu 2015) or soft regression (Hu et al. 2016) frameworks. Considering the superior performance of existing action recognition models for recognizing actions from full videos, it is more attractive to transfer the knowledge contained in full videos to the partial videos. This idea was firstly explored in (Kong, Tao, and Fu 2017) and (Qin et al. 2017), where knowledge transfer was achieved by reconstructing the visual features of the full videos from the features of the partial videos. To obtain better prediction results, they also incorporated the action class information as additional constraints into their learning frameworks. This, however, could result in a sub-optimal solution both for feature reconstruction and action class information encoding, since their objectives for optimization are quite different. Furthermore, in their works, the action label for the full video is directly used to represent partial videos, which may introduce noises since partial videos often contain incomplete action executions.

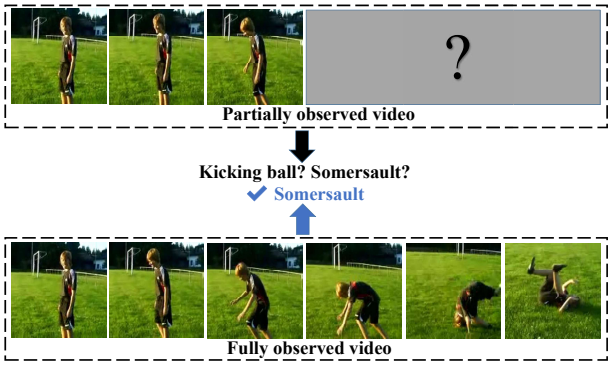


Figure 1: Actions from different classes may be very similar at the beginning stage. From the partially observed video, it is not clear whether the boy was kicking a ball on the grass or doing somersault. From the fully observed video however, it is clear that the action was somersault. This motivates us to transfer the knowledge from full videos to partial videos.

*Corresponding author

To solve the aforementioned problems, we propose a novel knowledge transfer method for action prediction. Our method intends to learn rich action knowledge from full videos, and then transfer the knowledge to partial videos for the prediction of partially observed actions. This is achieved by developing a two-stage learning framework. At the first stage, we learn a set of feature embeddings and a discriminative classifier from full videos. The knowledge in the learned embeddings and classifier is then transferred to the partial videos at the second stage, improving action prediction with partial videos. To encode rich information about action class in the learned embeddings for full videos, we enforce the distances between embeddings of different action classes to be larger than a margin, by employing the Additive Margin (AM) Softmax (Wang et al. 2018). Our discriminative classifier is learned only from the full videos, and thus has the advantage of avoiding noises introduced by partial videos, as partial videos often contain incomplete action executions. We have experimentally demonstrated that the prediction performance can be significantly improved by our proposed knowledge transfer framework.

The main contribution of this paper is two-fold. Firstly, we propose a novel knowledge transfer framework to boost the performance of action prediction with partial videos, by transferring knowledge from feature embeddings and discriminative classifier of full videos. Secondly, our method shows remarkable improvement for action prediction over several baselines, and outperforms all the state-of-the-art action prediction systems.

Related Works

Action recognition is a widely studied task in computer vision, which takes a fully observed video as input and output the action class depicted in the video. Traditional methods mainly focused on developing hand-crafted features for capturing action appearance and motion, among which dense trajectory (Wang et al. 2011) and its improvement (Wang and Schmid 2013) showed impressive results. Recent studies showed that deep learning methods such as convolutional neural networks (CNNs) can obtain good results for learning spatiotemporal features. Two-stream architecture (Simonyan and Zisserman 2014) learned spatial and temporal information separately from RGB frame and stacked optical flow streams. Features from the two streams were further enhanced by fusion operations (Feichtenhofer, Pinz, and Zisserman 2016; Feichtenhofer, Pinz, and Wildes 2017) or long-term temporal modelling (Wang et al. 2016). In (Tran et al. 2015), spatiotemporal features were learned jointly by 3D convolutions. Carreira *et al.* (Carreira and Zisserman 2017) inflated existing 2D CNNs with pre-trained weights into 3D ones. Deeper architectures (Hara, Kataoka, and Satoh 2018) and decompositions (Qiu, Yao, and Mei 2017; Tran et al. 2018) were also studied for 3D convolutions. These methods mainly learned features for the videos with full action executions.

Action prediction aims to predict actions from partially observed videos, before end of the action executions. Ryoo *et al.* (Ryoo 2012) proposed to use integral and dynamic bag-of-words for action prediction. In (Kong and Fu 2015), a max margin learning framework was presented to learn discriminative features for prediction. Monotonic constraints were also utilize for early action detection (Ma, Sigal, and Sclaroff 2016). Instead of adding constraints to the score function, Hu *et al.* (Hu et al. 2016) proposed to learn a set of soft labels for annotating partial action sequences. Lan *et al.* (Lan, Chen, and Savarese 2014) developed hierarchical representations at multiple granularities to predict human action before it starts. Vondrick *et al.* (Vondrick, Pirsiavash, and Torralba 2016) proposed to predict the feature of future frames to learn better representations for action recognition. These approaches do not seek to make use of the action knowledge learned from full sequences for prediction. In (Kong, Tao, and Fu 2017; Qin et al. 2017), knowledge in full videos was transferred to partial videos by constructing a linear projection from the visual features of partial video to those of full videos. We also enhance the discriminative power of partial videos by transferring knowledge from full videos to partial videos. Different from existing approaches, we propose to mine rich action knowledge from full videos. This is achieved by learning feature embeddings for full videos in a discriminative way. We then transfer the knowledge from the learned embeddings and discriminative function to partial videos. Metric learning is also exploited for action prediction (Lai et al. 2018; Kong et al. 2018). Despite the impressive performance, these models relied on nearest neighbor matching during inference, which is computationally expensive. Our method is more computationally efficient both for training and inference, and thus is more feasible for practical applications.

Knowledge distillation is also related to our work. In (Hinton, Vinyals, and Dean 2015; Huang and Wang 2017; Yim et al. 2017), the knowledge contained in a large network was distilled and transferred to a small network, by enforcing the outputs or intermediate activations of the small network to match those of the large network. We employed a similar idea of knowledge transfer. Different from knowledge distillation, our goal is to improve the discriminative power of partially observed videos, and we achieve this goal by transferring knowledge from the embedding and classifier knowledge learned from full videos.

Our Approach

Given an action video (may be partially observed), our goal is to predict the action class depicted in the video. Following (Kong and Fu 2015; Kong, Tao, and Fu 2017; Kong et al. 2018; Hu et al. 2018), we divide a video $\mathbf{x}$ containing T frames uniformly into K sub-segments (K=10 in our case). Each segment contains $\frac{K}{T}$ frames, and the k-th segment ranges from the $[(k-1)\cdot\frac{T}{K}+1]$-th frame to the $(\frac{kT}{K})$-th frame. A partial video $\mathbf{x}^{(k)}$ is generated from the full video $\mathbf{x}$ by taking the beginning k segments, and the corresponding *progress level* and *observation ratio* are defined as k and $\frac{k}{K}$, respectively. Indeed, for the fully observed actions (i.e., the observation ratio r is 1), any existing action recognition model (e.g., 3D CNN (Hara, Kataoka, and Satoh 2018)) can be employed for prediction. However, for the prediction of videos with partial action executions (e.g., the

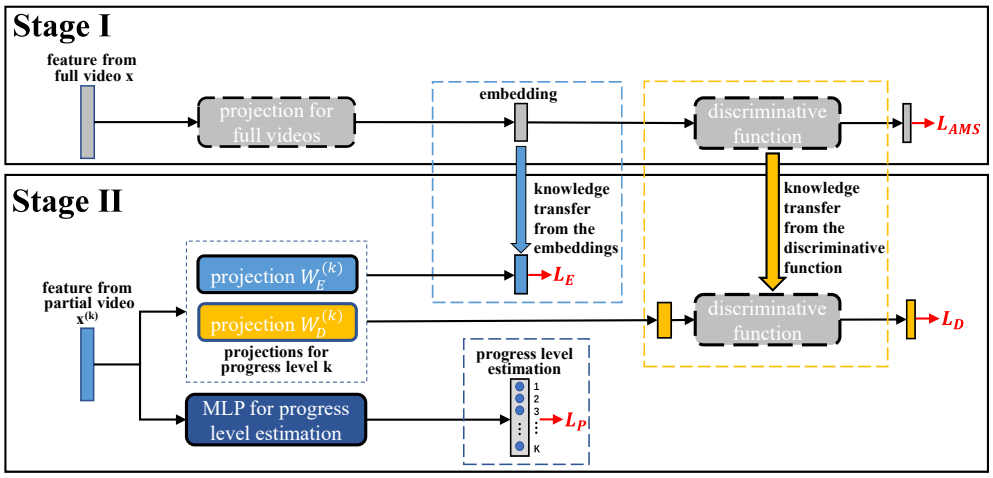


Figure 2: Learning architecture for our proposed two-stage knowledge transfer scheme. Blocks with dashed lines are learned at the first stage and fixed at the second stage. Blocks with solid lines are learned at the second stage, where light blue blocks correspond to the embedding knowledge transfer, and orange blocks correspond to discriminative classifier knowledge transfer. Red arrows are for loss computation. Best viewed in color.

observation ratio $r = 0.1$), these recognition models often perform poorly as partially observed videos do not contain enough action information for recognition. Here, we aim at boosting the performance of existing recognition model for predicting the actions at early stages, without sacrificing the recognition accuracy for full videos.

Our method first learns a set of feature embeddings and a discriminative action classifier from full videos, and then use the knowledge gained from the full videos (the learned embeddings and discriminative function) to guide our feature learning for the partial videos. The overall learning architecture is presented in Figure 2. As shown, our learning framework consists of two stages. At the first stage, we learn feature embeddings and a discriminative classifier from the visual features of full videos. At the second stage, we learn a set of projections to map the visual features of partial videos into the embedding space. During the projection learning, the embeddings and classifier learned from full videos are fixed and used to transfer the action knowledge gained from full video to partial videos.

Learning Action Knowledge from Full Videos

Firstly, we would like to learn a set of feature embeddings and an action classifier from the full videos, which will be used to guide our feature learning for the partial videos. To encode more information about action class, the feature embeddings are learned so that they have large inter-class distances and small intra-class distances. Here, instead of using the *push-and-pull* learning strategy popularly used for metric learning, we opt to learn the embeddings under a discriminative learning framework, so that both the learned embeddings and discriminative function can capture rich action information, which will be transferred to partial videos.

Given a set of full videos $\{\mathbf{x}_i\}$ with corresponding features $\{\mathbf{f}_i\}$ and labels $\{y_i\}$, we intend to learn an embedding function G to project the original feature onto an embedding space, and a discriminative classifier D to project the embedding to the label space:

$$\mathbf{e}_i = G(\mathbf{f}_i), \tag{1}$$

$$\mathbf{p}_i = D(\mathbf{e}_i). \tag{2}$$

Here, we define the linear discriminative function as $D(\mathbf{e}) = \mathbf{We}$, where $\mathbf{W} \in R^{p \times C}$ is the weight encoding the action class information, which would be learned in the training phase. C is the number of action classes and p is the dimension of the embeddings. To encourage large distances between embeddings from different classes, we employed the AM Softmax (Wang et al. 2018) to constrain our feature embedding and classifier learning. In specific, we would minimize the following cross-entropy loss:

$$L_{AMS} = -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{s\cdot(cos\theta_i^{y_i}-m)}}{e^{s\cdot(cos\theta_i^{y_i}-m)}+\sum_{j=1,j\neq y_i}^{c} e^{s\cdot cos\theta_i^j}} \tag{3}$$

$$= -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{s\cdot(\mathbf{w}_{y_i}^T\mathbf{e}_i-m)}}{e^{s\cdot(\mathbf{w}_{y_i}^T\mathbf{e}_i-m)}+\sum_{j=1,j\neq y_i}^{c} e^{s\cdot\mathbf{w}_j^T\mathbf{e}_i}}, \tag{4}$$

where $\mathbf{w}_j^T$ is the j-th row of $\mathbf{W}$. Here, both $\mathbf{e}_i$ and $\mathbf{w}_j^T$ are L2-normalized. Hence, the element p_i^j can be considered as the cosine distance between $\mathbf{e}_i$ and $\mathbf{w}_j$: $p_i^j = \mathbf{w}_j^T\mathbf{e}_i = \cos\theta_i^j$. A margin m is added to the cross-entropy loss to explicitly constrain the inter-class distances.

Minimizing L_{AMS} will enforce large cosine distance between embeddings of the videos from different classes. Since the embeddings are L2-normalized, this can also lead to smaller intra-class distances. By employing the AM Softmax, both the learned embeddings and classifier contain rich information about action class, which can benefit our knowledge transfer. A scaling factor s is applied for better conver-

gence, as suggested in (Wang et al. 2017). The parameters m and s are fixed during the model training.

Transferring Action Knowledge to Partial Videos

After learning embeddings and action classifier from full videos at the above stage, we then transfer the knowledge (the learned embeddings and classifier) to partial videos to improve the prediction performance. Considering that partial videos often contain only a part of action executions, learning action classifier from partial videos could introduce some noises. Therefore, here we mainly learn a set of projections to project the visual features of partial videos onto the embedding space with the action classifier fixed.

Similar to that for the full videos, visual features from partial videos are also projected onto the embedding space, so that the knowledge contained in full videos can be transferred to partial videos. Here, we use linear projection with weights $\mathbf{W}_E, \mathbf{W}_D \in R^{d\times p}$ for our projections, where d is the dimension of the features, and p the dimension of the embeddings. Given visual feature $\mathbf{f}_i^{(k)}$ for the partial video $\mathbf{x}_i^{(k)}$ and the embedding $\mathbf{e}_i$ of the corresponding full video $\mathbf{x}_i$, the projection parameterized by $\mathbf{W}_E$ is learned to minimize the squared Euclidean distance between the projected feature $\mathbf{e}_i^k$ and the embedding $\mathbf{e}_i$:

$$L_E = \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}||\mathbf{W}_E\mathbf{f}_i^{(k)} - \mathbf{e}_i||_2^2. \tag{5}$$

Note that linear projection is also employed in (Qin et al. 2017) and (Kong, Tao, and Fu 2017), with a similar goal of using knowledge contained in full videos to improve the prediction of partial videos. However, they proposed to reconstruct the visual features of full videos, which are lack of action class information. To obtain better learning results, they incorporated the action label information as additional constraints into their learning framework, which may result in a sub-optimal solution for both knowledge transfer and label information encoding. In comparison, we propose to first encode the action class information in the embeddings for full videos. The learned embeddings serve as a unified target for transferring both the knowledge in full videos and the action class information to partial videos. By transferring knowledge from a unified target, learning becomes much easier, and is more flexible to extend to deeper architectures.

In addition to the embeddings of full video, the learned discriminative classifier also contains some important action cues, which can be exploited for the partial videos. A linear projection with weight $\mathbf{W}_D$ is learned to exploit the knowledge in the discriminative classifier. More specifically, given the learned action classifier $\mathbf{W}$ and visual feature $\mathbf{f}_i^{(k)}$ for partial video $\mathbf{x}_i^{(k)}$, our discriminative knowledge transfer is achieved by minimizing the following cross-entropy loss:

$$L_D = -\frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K} log\frac{e^{\mathbf{w}_{y_i}^T\mathbf{W}_D\mathbf{f}_i^{(k)}}}{\sum_{j=1}^{c}e^{\mathbf{w}_j^T\mathbf{W}_D\mathbf{f}_i^{(k)}}}, \tag{6}$$

where $\mathbf{w}_{y_i}^T$ is the y_i-th row of $\mathbf{W}$, which is fixed during the learning for $\mathbf{W}_D$.

Note that features from videos of different progress level typically have different distributions. Sharing the projection weights across different progress levels may not be optimal. Here, we select to learn two projection weights $(\mathbf{W}_E^{(k)}, \mathbf{W}_D^{(k)})$ for each progress level k. In practice, the progress level of the partially observed video is not provided, and thus needs to be estimated. We formulate the progress level estimation as a classification problem, where a two-layer Multi-Layer Perceptron (MLP) is applied on the extracted features to obtain K scores corresponding to K progress levels. During training, the progress level is available, and a standard cross-entropy loss L_P is employed for the MLP learning. During inference, the MLP computes a estimation score for each progress level. The input partial video is fed into the projection modules for all progress levels, and the estimation scores are used as weights for combining the projection outputs of all progress levels.

Overall, the loss function for the second stage of our learning framework is as follows:

$$L = L_D + \beta L_E + \gamma L_P. \tag{7}$$

The weighting factors β and γ are hyper-parameters determined by cross-validation.

Model Training and Inference

Training. Our model training consists two stages. At the first stage, embeddings and the action classifier are learned from full videos with the loss defined in Eq. (3). At the second stage, projections $(\mathbf{W}_E^{(k)}, \mathbf{W}_D^{(k)})$ for each progress level k are learned with the loss in Eq. 7. In this step, the embeddings and action classifier are fixed and used to improve our feature learning for the partial videos.

Inference. Figure 3 shows our model architecture for inference. During inference, feature $\mathbf{f}^{(k)}$ for the partial video $\mathbf{x}^{(k)}$ is projected to the embedding space by $\mathbf{W}_E$ and $\mathbf{W}_D$, respectively. Estimated score $\alpha^{(p)}$ for each progress level p is also computed if the progress level is unknown. Then the prediction scores $\mathbf{p}^{(k)}$ are computed as follows:

$$\mathbf{e}^{(k)} = \sum_{p=1}^{K}\alpha^{(p)}(\mathbf{W}_E^{(p)}\mathbf{f}^{(k)} + \mathbf{W}_D^{(p)}\mathbf{f}^{(k)}), \tag{8}$$

$$\mathbf{p}^{(k)} = D(\mathbf{e}^{(k)}) = \mathbf{W}\mathbf{e}^{(k)}. \tag{9}$$

The predicted class $\tilde{y}^{(k)}$ is given by the element that has the largest prediction score:

$$\tilde{y}^{(k)} = \arg\max \mathbf{p}^{(k)}. \tag{10}$$

Experiment

Datasets

We test our method on two datasets: UCF-101 (Soomro, Zamir, and Shah 2012) and HMDB-51 (Kuehne et al. 2011). The UCF-101 dataset consists of 13,320 videos from 101 human action classes, which are mainly human-object interactions and sports, such as "Playing Guitar" and "Basketball Dunk". Following (Kong, Tao, and Fu 2017; Kong et

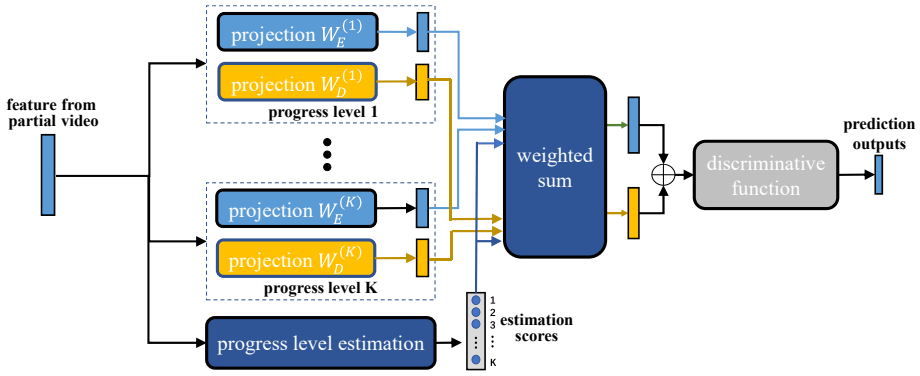


Figure 3: Model architecture for inference. Visual features extracted from the partial videos are fed into the projection modules for all progress levels. The outputs are then weighted combined using the scores for progress level estimation. The weighted sums are further added up and fed into the discriminative function for our final prediction outputs. Best viewed in color.

al. 2018), we use the first 15 groups of videos in UCF-101 split-1 for model training; the next 3 groups for model validation; and the remaining 7 groups for testing. The HMDB-51 dataset consists of 6766 videos from 51 action classes. Compared with UCF-101, HMDB-51 is more challenging with larger intra-class variance. For HMDB-51, We follow the standard evaluation protocol using three training/testing splits, and report the average accuracy over three splits.

Implementation Details

We use the 3D ResNext-101 (Hara, Kataoka, and Satoh 2018) trained on Kinetics (Kay et al. 2017) for feature extraction without finetuning. Visual features are extracted from sampled clips, which contain 16 video frames. Given a video with T frames, roughly $\frac{T}{4}$ clips are sampled for feature extraction. The extracted features from all the clips are averaged and normalized by L2 norm to form our video representation. The dimension of the extracted feature is 2048. We set the dimension of the feature embeddings as 1024. Stochastic gradient descent algorithm is employed for optimizing the model parameters, with a batch size of 64 and momentum rate of 0.9. We follow the suggestion in (Wang et al. 2018) and set the margin m and scaling factor s for the AM Softmax to 0.4 and 30, respectively.

Compared Baselines

We compare our method with two baselines with the same visual features. The first baseline uses the model trained at the first stage of our method for action prediction. We denote it as No-Transfer. Indeed, it is a traditional recognition model for recognizing actions from fully observed videos, without being specifically designed for action prediction. The second baseline has the same model architecture as our method, and the key difference lies in the learning strategy. For this baseline, the projection layers and classifier are learned jointly for both partial videos and full videos, with a single classification loss. The weight for the discriminative function is shared and learned from videos of all progress levels, thus enabling implicit knowledge transfer (Implicit-Transfer). In comparison, our method adopts a two-stage learning strategy. The discriminative function is learned from the full videos at the first stage, rather than learned jointly from the full videos and partial videos. Knowledge in the learned embeddings and discriminative function is then explicitly transferred to the partial videos, by learning the projections with the losses defined in Eq. (5) and Eq. (6). For both methods, we assume that the progress level is unknown during inference.

Results

Table 1 presents detailed prediction performance at each observation ratio on the HMDB-51 dataset. As shown, without knowledge transfer (No-Transfer), the accuracy for action prediction at observation ratio $r = 0.1$ is 22.7% lower than that of full videos (observation ratio $r = 1.0$). Implicit-Transfer shows a 2.7% gain at observation ratio $r = 0.1$, by exploiting the partial videos during training and transferring knowledge from full videos to partial videos implicitly. Our method further improves the accuracy by 2.1% over Implicit-transfer, and 4.8% over No-transfer, demonstrating the effectiveness of our method to boost accuracy at early stages. The performance boost over the baselines becomes smaller as the observation ratio increases. This is expected, since the information contained in partial videos becomes richer with an increasing observation ratio. For full videos (observation ratio $r = 1.0$), the accuracy of our method is on par with No-Transfer, which is not surprising because we cannot gain extra information by transferring knowledge from a full video to itself. For action prediction, we focus more on improving the performance at early stages rather than late stages.

Table 2 shows the detailed prediction performance on the UCF-101 dataset. As can be seen, our method outperforms the baselines in term of action prediction at early stages. We can observe that our baseline without knowledge transfer already achieves an accuracy of 76.3% when only 10% of the videos are observed. The gap between the accuracies at observation ratio $r = 0.1$ and $r = 1.0$ is 14.8% for No-Transfer, which is much smaller than that on the HMDB-51 dataset (22.7%). This confirms that HMDB-51 is more chal-

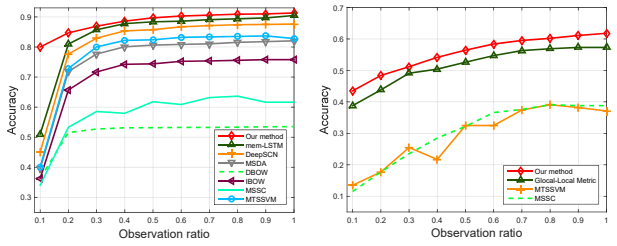
Table 1: Prediction results (%) on HMDB-51 dataset.

Methods	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	avg.
No-Transfer	38.7	44.7	47.5	52.5	55.2	57.0	58.7	60.0	60.6	61.4	53.7
Implicit-Transfer	41.4	46.7	49.9	52.3	55.2	56.9	58.3	60.1	61.1	61.5	54.3
Our method	**43.5**	**48.4**	**51.2**	**54.2**	**56.4**	**58.4**	**59.6**	**60.2**	**61.1**	**61.8**	**55.5**

Table 2: Prediction results (%) on UCF-101 dataset.

Methods	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0	avg.
No-Transfer	76.3	82.7	85.9	87.7	89.3	90.0	90.3	90.5	90.9	91.1	87.5
Implicit-Transfer	79.0	83.8	86.4	87.9	88.4	89.7	90.1	90.6	90.9	90.8	87.7
Our method	**80.0**	**84.7**	**86.9**	**88.6**	**89.7**	**90.3**	**90.6**	**90.9**	**91.0**	**91.3**	**88.4**

lenging than UCF-101. We also find that our method can achieve an improvement of 3.7% over No-Transfer and 1.0% over Implicit-Transfer at observation ratio $r = 0.1$. This demonstrates that the proposed knowledge transfer method can largely improve the prediction of partial videos over a strong baseline. However, the performance gap becomes smaller when observation ratio becomes larger, which is consistent with the results on HMDB-51 dataset.



(a) UCF-101 dataset (b) HMDB-51 dataset

Figure 4: Comparison with the state-of-the-art prediction systems on (a) UCF-101 dataset and (b) HMDB-51 dataset.

Table 3: Prediction accuracy (%) for different variants of our model on HMDB-51 split-1. Results are shown for partial videos at early stages with observation ratio $r \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$.

Methods	0.1	0.2	0.3	0.4	0.5
No-Transfer	40.7	46.9	51.4	54.9	57.7
Embedding-Only	46.0	50.2	53.3	55.3	58.2
Discriminative-Only	44.6	49.5	51.9	53.9	57.3
Ours w/o AM Softmax	44.8	49.9	52.4	55.7	57.6
Ours w/ shared weights	44.6	49.9	53.3	**56.7**	**58.8**
Our method	**46.8**	**51.0**	**53.5**	56.1	58.4

Comparison with the State-of-the-art Methods

We compare our method with IBOW and DBOW (Ryoo 2012), MSSC (Cao et al. 2013), MTSSVM (Kong and Fu 2015), MSDA (Chen et al. 2012), DeepSCN (Kong, Tao, and Fu 2017) and mem-LSTM (Kong et al. 2018) on the UCF-101 dataset. Figure 4(a) shows the detailed comparison results. As shown, our method outperforms all the state-of-the-arts by a large margin at early stages. We achieve an accuracy of 80.0% when only 10% of the videos are observed. We also observe that for the recognition of full action videos (i.e., observation ratio $r = 1.0$), the performance of our method is on par with the mem-LSTM model. However, mem-LSTM employed a complex network architecture (two-stream networks followed by Bi-LSTMs) for prediction, which is computationally expensive. In comparison, our method is much more efficient both for model training and inference. We also compare our method with MSSC, MTSSVM and Global-Local Metric Prediction (Lai et al. 2018) on HMDB-51 dataset. As shown in Figure 4(b), our method outperforms all of the compared methods.

Ablation Study

We provide more evaluation results on split-1 of HMDB-51 dataset. Specifically, we experiment with only transferring embedding knowledge from full videos (denoted by Embedding-Only), or only transferring discriminative classifier knowledge from full videos (Discriminative-Only). For Embedding-Only, only the projections $\mathbf{W}_E^{(k)}$ and the corresponding loss L_E in Eq. (5) are considered for knowledge transfer. For Discriminative-Only, only the classifier parameter $\mathbf{W}_D^{(k)}$ the loss L_D in Eq. (6) are used for knowledge transfer. We also experiment with other variants of our method, including the one without AM Softmax (denoted by Ours w/o AM Softmax), and the one with the same projection weights across different progress levels (Ours w/ shared weights). All the results are presented in Table 3, where the results for the prediction of actions at early stages (observation ratio $r \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$) are reported.

Evaluation for Embedding-Only and Discriminative-Only. Compared with No-Transfer, both Embedding-Only and Discriminative-Only improve the accuracy of action prediction at early stages significantly, demonstrating that both methods can enhance the discriminative power for partial videos at early stages. Embedding-Only shows better results than Discriminative-Only, because the embeddings of full videos are much more informative. By learning the embeddings in a discriminative way, both action class information and intra-class distribution are encoded in the learned embeddings. The learned discriminative function also contains information about action class, but it lacks of informa-

tion about the distribution within certain class. Our method achieves the best performance at early stages, by exploiting both the learned embeddings and discriminative function for knowledge transfer.

Effect of AM Softmax. Learning the embeddings for full videos without AM Softmax (Ours w/o AM Softmax) gives lower accuracies than our method. AM Softmax enforces large cosine distance between different classes by adding a margin in the loss function. Embeddings that have larger inter-class distances help to transfer more discriminative knowledge to partial videos. Although the desired embeddings can be learned in other ways, such as invoking the *pull-and-push* strategy, AM Softmax provides a more convenient way to incorporate this target into the discriminative learning process.

Effect of sharing weights. Sharing projection weights across all progress levels (Ours w/ shared weights) gives lower accuracies for action prediction at early stages. It is inherently challenging to learn a set of shared weights for different progress level, since the distribution of the extracted features varies a lot from small observation ratios to large observation ratios. The results show that learning a set of projection weights for each progress level can obtain a better prediction performance.

Effect of parameter β. In our knowledge transfer framework, we employ a parameter β to control the influence of the learned embedding knowledge (please refer to Eq. (7)). Figure 5 presents the mean accuracies for the prediction of videos over all the considered progress levels. As shown, our method is quite robust to the setting of β. A proper β gives a better result. Generally, too large β (e.g., larger than 20000) will result in an inferior performance.

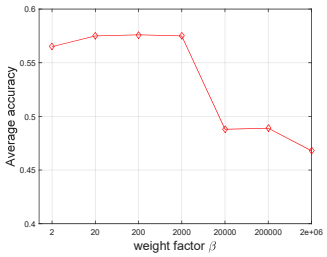


Figure 5: Average accuracy on HMDB-51 for different values of weighting factor β.

Table 4: Prediction accuracy (%) on HMDB-51 split-1 at observation ratio $r \in \{0.1, 0.2, 0.3, 0.4, 0.5\}$ with unknown/known progress level or uniform scores during inference.

Methods	0.1	0.2	0.3	0.4	0.5
Our method (Uniform)	44.4	49.8	53.7	56.0	58.1
Our method (unknown)	**46.8**	51.0	53.5	56.1	58.4
Our method (known)	46.5	**51.5**	**53.7**	**56.1**	**58.8**

Known progress level. Generally, we assume that the progress level is unknown during inference. We learn to predict a score for each progress level, which are then used to compute a weighted sum of the outputs for all progress levels. Here, we evaluate the influence of the progress level estimation by comparing with two variants. The first one assumes that the progress level is known. The second one use uniform scores for each progress level. Our results in Table 4 show that our method is quite robust to the progress level estimation. The prediction performance of our method drops slightly if we set the estimated scores to the uniform ones. We also observe that the performance would be slightly improved if we use the manually provided progress level instead of estimating it.

Conclusion

In this paper, we have proposed a novel knowledge transfer framework for improving the performance of action prediction with partial videos. We transferred the knowledge of feature embeddings and action classifier from full videos by a two-stage learning framework. At the first stage, we learn a set of feature embeddings and action classifier from the full videos. The learned embeddings and classifier knowledge are then used to improve the prediction of partial videos at the second stage. We experimentally show that the proposed knowledge transfer method can significantly improve the accuracy of action prediction with partial videos, especially for the actions of small observation ratios (e.g., less than 10%).

Acknowledgments

This work was supported partially by the National Key Research and Development Program of China (2018YFB1004903), NSFC(61522115, 61661130157, 61472456, U1611461), Guangdong Province Science and Technology Innovation Leading Talents (2016TX03X157), and the Royal Society Newton Advanced Fellowship (NA150459). Jian-Fang Hu is partially supported by the NSFC (61702567), CCF-Tencent open research fund, and the Fundamental Research Funds for the Central Universities under Grant (18lgpy60).

References

Cao, Y.; Barrett, D.; Barbu, A.; Narayanaswamy, S.; Yu, H.; Michaux, A.; Lin, Y.; Dickinson, S.; Siskind, J. M.; and Wang, S. 2013. Recognize human activities from partially observed videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2658–2665.

Carreira, J., and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4724–4733.

Chen, M.; Xu, Z.; Weinberger, K.; and Sha, F. 2012. Marginalized denoising autoencoders for domain adaptation. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, 767–774.

Feichtenhofer, C.; Pinz, A.; and Wildes, R. P. 2017. Spatiotemporal multiplier networks for video action recogni-

tion. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7445–7454.

Feichtenhofer, C.; Pinz, A.; and Zisserman, A. 2016. Convolutional two-stream network fusion for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1933–1941.

Hara, K.; Kataoka, H.; and Satoh, Y. 2018. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Hinton, G.; Vinyals, O.; and Dean, J. 2015. Distilling the knowledge in a neural network. *Computer Science* 14(7):38–39.

Hu, J.-F.; Zheng, W.-S.; Ma, L.; Wang, G.; and Lai, J. 2016. Real-time rgb-d activity prediction by soft regression. In *European Conference on Computer Vision*, 280–296.

Hu, J.-F.; Zheng, W.-S.; Ma, L.; Wang, G.; Lai, J.-H.; and Zhang, J. 2018. Early action prediction by soft regression. *IEEE transactions on pattern analysis and machine intelligence*.

Huang, Z., and Wang, N. 2017. Like what you like: Knowledge distill via neuron selectivity transfer. *CoRR* abs/1707.01219.

Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; Suleyman, M.; and Zisserman, A. 2017. The kinetics human action video dataset. *CoRR*.

Kong, Y., and Fu, Y. 2015. Max-margin action prediction machine. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 38(9):1844–1858.

Kong, Y.; Gao, S.; Sun, B.; and Fu, Y. 2018. Action prediction from videos via memorizing hard-to-predict samples. In *AAAI Conference on Artificial Intelligence*.

Kong, Y.; Tao, Z.; and Fu, Y. 2017. Deep sequential context networks for action prediction. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3662–3670.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Lai, S.; Zheng, W.; Hu, J.; and Zhang, J. 2018. Global-local temporal saliency action prediction. *IEEE Transactions on Image Processing* 27(5):2272–2285.

Lan, T.; Chen, T. C.; and Savarese, S. 2014. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*, 689–704.

Ma, S.; Sigal, L.; and Sclaroff, S. 2016. Learning activity progression in lstms for activity detection and early detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1942–1950.

Qin, J.; Liu, L.; Shao, L.; Ni, B.; Chen, C.; Shen, F.; and Wang, Y. 2017. Binary coding for partial action analysis with limited observation ratios. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6728–6737.

Qiu, Z.; Yao, T.; and Mei, T. 2017. Learning spatio-temporal representation with pseudo-3d residual networks. In *IEEE International Conference on Computer Vision (ICCV)*.

Ryoo, M. S. 2012. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *IEEE International Conference on Computer Vision (ICCV)*, 1036–1043.

Simonyan, K., and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*.

Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; and Paluri, M. 2015. Learning spatiotemporal features with 3d convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, 4489–4497.

Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; and Paluri, M. 2018. A closer look at spatiotemporal convolutions for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Vondrick, C.; Pirsiavash, H.; and Torralba, A. 2016. Anticipating visual representations from unlabeled video. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 98–106.

Wang, H., and Schmid, C. 2013. Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision (ICCV)*, 3551–3558.

Wang, H.; Kläser, A.; Schmid, C.; and Liu, C. 2011. Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3169–3176.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2016. Temporal segment networks: Towards good practices for deep action recognition. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *European Conference on Computer Vision*, 20–36. Cham: Springer International Publishing.

Wang, F.; Xiang, X.; Cheng, J.; and Yuille, A. L. 2017. Normface: L2 hypersphere embedding for face verification. In *Proceedings of the 2017 ACM on Multimedia Conference*, MM '17, 1041–1049. New York, NY, USA: ACM.

Wang, F.; Cheng, J.; Liu, W.; and Liu, H. 2018. Additive margin softmax for face verification. *IEEE Signal Processing Letters* 25(7):926–930.

Yim, J.; Joo, D.; Bae, J.; and Kim, J. 2017. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 7130–7138.