# Cross-stream Selective Networks for Action Recognition

Bowen Pan[1⋆] Jiankai Sun[1,2⋆] Wuwei Lin[1⋆] Limin Wang[3] Weiyao Lin[1†]
[1]Shanghai Jiao Tong University [2]SenseTime Research
[3]State Key Laboratory for Novel Software Technology, Nanjing University, China
{googletornado, linwuwei13, wylin}@sjtu.edu.cn
sunjiankai@sensetime.com lmwang.nju@gmail.com

## Abstract

*Combining multiple information streams has shown obvious improvements in video action recognition. Most existing works handle each stream independently or perform a simple combination on temporally simultaneous samples in multi-streams, which fails to make full use of the stream-wise complementary property due to the negligence of the temporal pattern gaps among streams. In this paper, we propose a cross-stream selective network (CSN) to properly integrate and evaluate information in multi-streams. The proposed CSN first introduces a local selective-sampling module (LSM), which can find asynchronous correspondences among streams and construct high-correlated sample groups across multiple information streams. This LSM can effectively deal with the temporal dis-alignment among different streams, leading to a better integration of cross-stream information. We further introduce a global adaptive-weighting module (GAM). It adaptively evaluates the importance weights for each cross-stream sample group and selects temporally more important ones in action recognition. With the integration of cross-stream information, our GAM can obtain more reasonable importance than the existing single-stream weighting schemes. Extensive experiments on benchmark datasets of UCF101 and HMDB51 demonstrate the effectiveness of our approach over previous state-of-the-art methods.*

## 1. Introduction

Video action recognition has attracted much attention due to its importance in many applications. Combining multiple information streams (e.g., RGB frames, optical flow, RGB differences) with Convolutional Neural Networks (CNNs) has shown superior performances and becomes a popular framework in action recognition [11, 5, 18, 24, 9].

These information streams capture different aspects of an input video and exhibit different temporal variation patterns (cf. Fig. 1). Therefore, one major challenge in action recognition is how to properly evaluate and integrate information in multi streams, such that the complementary information among streams can be fully utilized. However, this issue is not fully studied. Most existing works combine multi-streams in a relatively simple way, which either handle each stream independently or perform a simple combination on temporally simultaneous samples in multi streams [18, 11]. They have limitations in making full use of the stream-wise complementary property due to the negligence of the temporal pattern gaps among streams.

In this paper, we argue that a proper multi-stream information integration method should be able to boost action recognition in two aspects:

(1) *Dealing with temporal dis-alignment among streams*. For example, in Fig. 1, the most discriminative sample for action *long jump* is around time $t_1$ in RGB stream when the person is jumping in the air (most indicative of long jump in appearance aspect), while the most discriminative one in optical flow stream is around time $t_2$ when the person is just jumping out (with most indicative 'long jump' motion pattern). This temporal dis-alignment would affect the action recognition accuracy. Therefore, it is expected to establish a correspondence between two streams (cf. Fig. 2b), which is able to align RGB and optical flow locally and adaptively, thus reducing the stream-wise asynchrony and selecting more discriminative sample groups (i.e. groups of RGB and flow).

(2) *Obtaining more reasonable global importance weights*. As samples in each single stream have different discriminative capability in recognizing actions, it is non-trivial to evaluate the importance of each sample such that more discriminative samples can have larger weights when determining the action label. Most existing works perform importance evaluation for each stream independently [17],

⋆ Authors contributed equally.
† Corresponding author.

which may create some improper weights due to the limitation of information sources. For example, for action *long jump* in Fig. 2a, when only considering a single RGB stream, the single-stream-based weighting method may improperly assign higher weights on early samples when the athlete is running, rather than more discriminative samples when the athlete is in the air. Comparatively, if we jointly consider multi-streams during importance evaluation, we are able to have more information sources and obtain more reasonable importance weighting results (cf. Fig. 2b).

To meet the above requirements, this paper proposes a cross-stream selective network (CSN) to properly integrate and evaluate multi-stream information. The proposed CSN first introduces a local selective sampling module (LSM), which can find asynchronous correspondences among streams and construct high-correlated sample groups across multiple information streams. This LSM can effectively deal with the temporal dis-alignment problem among streams, leading to a better integration of cross-stream information. We further introduce a global adaptive weighting module (GAM). It automatically evaluates the importance weights for each cross-stream sample group and highlights more important ones in action recognition. Based on the integration of cross-stream information, our GAM can obtain more reasonable importance weights than the existing single-stream weighting schemes. Extensive experiments on benchmark datasets demonstrates the effectiveness of our approach.

### 1.1. Related Work

**Video Action Recognition** Largely driven by image recognition methods with Convolutional Networks (CNN), video action recognition research has benefited greatly from advancements in deep CNN-based representations [16, 23]. Existing popular benchmarks like UCF101 [12] or HMDB51 [8] are used for action recognition in trimmed clips. There are many previous work [10, 6, 14, 7] for this task. [24] present the Structured Segment Network (SSN) which models the temporal structure of each action instance. [17] propose UntrimmedNet which couples the classification module and the selection module to learn the action models and reasons about the temporal duration of action instances. [25] introduce the Temporal Relation Network (TRN) module to learn and reason about temporal dependencies between video frames at multiple time scales. CNN-based action recognition methods have broadly followed two main paradigms: multi-stream method and 3D CNNs. ST-GCN [21] automatically learns both the spatial and temporal patterns for video action recognition. Some works [1, 19] also try to use pre-trained features on large dataset like Kinetics or a more complex backbone network such as resnet. Tran *et al*. [15] show that factorizing the 3D convolutional filters into separate spatial and temporal components yields significantly gains in accuracy.

**Stream-wise Fusion** Since the traditional multi-stream methods handle each stream independently, some methods further consider the fusion of multi-stream information to boost action recognition. For example, simultaneous fusion [5], sequence and video level fusion [20]. Since these methods do not consider the temporal pattern gap among different streams, they still have low efficiency in utilizing stream-wise information. Recently, [9] introduce an asynchronous fusion network to fuse information at different time points. However, our approach differs from this work in two aspects: (1) The method in [9] simply fuses a sample in one stream with all temporally neighboring ones in another stream while not differentiating their correspondence relationship. Comparatively, our approach aims to find a precise correspondence among cross-stream samples, thus can handle the temporal pattern gap among streams in a more effective way. (2) The method in [9] does not differentiate the importance of different samples. Comparatively, our approach also introduce a global adaptive-weighting module to evaluate the importance of cross-stream sample groups.

## 2. Cross-stream Selective Networks

### 2.1. Architecture

The architecture of the proposed cross-stream selective network (CSN) is shown in Fig. 3. Assuming that we have two information streams for an input video $V$, we first clip it into $N$ segments with equal duration. For each segment, we randomly sample an RGB image, thus obtaining an RGB image sequence $\{I_{t_1}, I_{t_2}, \cdots, I_{t_N}\}$ for the entire video. We then input this RGB sequence into the Local Selective-sampling Module (LSM) (detailed in Sec. 2.2), which will output a correspondence strength sequence $\{\mathbf{R}_{t_1}, \mathbf{R}_{t_2}, \cdots, \mathbf{R}_{t_N}\}$, where $\mathbf{R}_{t_i} = \{r_{t_i,t_i-\tau}, \cdots, r_{t_i,t_i+\tau}\}$ represents the correspondence strengths between an RGB image $I_{t_i}$ at time $t_i$ and optical flow stacks $\{O_{t_i-\tau}, \cdots, O_{t_i+\tau}\}$ within a temporal interval centered at $t_i$ (cf. Fig. 4a). Then, we select $M$ most correlated optical flow stacks $\{O_{m_i^1}, O_{m_i^2}, \cdots, O_{m_i^M}\}$ for each RGB image $I_{t_i}$ according to $\mathbf{R}_i$ and construct a set of high-correlated sample groups: $G_1 = \{I_1, O_{m_1^1}, \cdots, O_{m_1^M}\}, \cdots, G_N = \{I_N, O_{m_N^1}, \cdots, O_{m_N^M}\}$. These temporally-aligned groups of RGB images and optical flow stacks are the selected cross-stream samples which will be used for recognizing action.

When recognizing actions, we input the selected cross-stream sample groups (i.e., RGB images and optical flow stacks in the high-correlated sample groups) into two-stream CNNs to calculate deep features and predict action

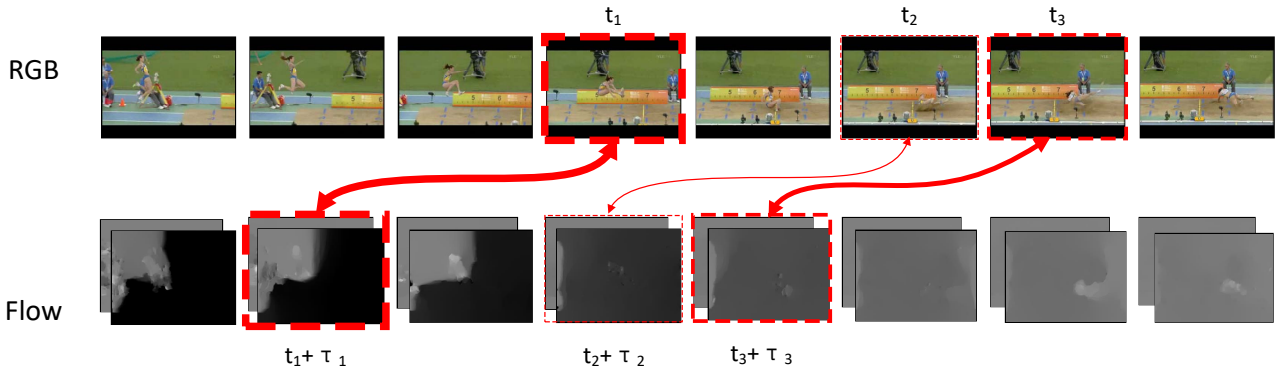
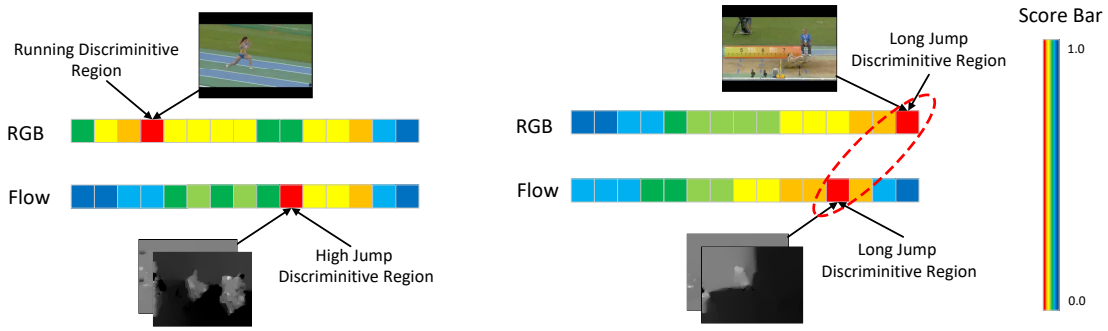



Figure 1: Illustration of a stream-wise temporal dis-alignment for a long jump action. Our CSN finds correspondences among samples **locally** and **adaptively** in the two streams. In addition, our CSN also learns a **global** importance weight based on the detected discriminative sample groups, where thicker line represents larger weights.




(a) Single-stream importance weighting

(b) Cross-stream importance weighting

Figure 2: Comparison of weighting scheme based on samples from each single stream and sample groups from two streams for a *long jump* action instance. Red indicates larger importance weights.

labels of the input video. At the same time, a global adaptive weighting module (GAM) is applied to evaluate importance weights of the selected sample groups (detailed in Sec. 2.3). These importance weights are combined (by weighted pooling) with the deep features of the selected cross-stream samples to yield the final recognition results.

Note that in order to make the step of *selecting most correlated optical flow stack* differentiable in training, we also introduce a Softmax Normalize Warping Unit (SNW) in our CSN framework. The SNW combines each optical flow stack sequence $\{O_{m_i^1}, O_{m_i^2}, \cdots, O_{m_i^M}\}$ with correspondence strength $\mathbf{R}_{t_i}$ and outputs a tensor $T_{t_i}$ which has same size with $O_{m_i^j}$ (detailed in Sec. 2.2).

## 2.2. Local Selective-sampling Module

In this section, we describe the details of Local Selective-sampling Module. This module is devised to deal with the video modality temporal dis-alignment at the local scale. Given sampled RGB frames, we stack its adjacent RGB images to make up a local snippet $\{I_{t_i-1}, I_{t_i}, I_{t_i+1}\}$, which depicts both appearance and local motion information. To capture the temporal motion information from RGB frames, we input this local snippet to a Bi-directional LSTM. Compared with the naïve LSTMs, Bi-directional LSTMs can utilize both the forward and backward direction context information with two separate hidden layers. The specific formulation is defined as follows:

$$\begin{cases} h_t^f = tanh(W_x^f x_t + W_h^f h_{t-1}^f + b_h^f) \\ h_t^b = tanh(W_x^b x_t + W_h^b h_{t+1}^b + b_h^b) \\ \mathbf{y}_t = W_y^f h_{t-1}^f + W_y^b h_{t+1}^b + b_y \end{cases}, \quad (1)$$

where $h_t^f$ represents the forward hidden sequence, $h_t^b$ is the backward hidden sequence and $\mathbf{y}_t$ is the output sequence. Bidirectional-LSTM computes $h_t^f$, $h_t^b$, $\mathbf{y}_t$ by iterating the backward layer at time $t$.

Outputs of Bi-LSTM selector are three fixed-dimension vectors $\{\mathbf{y}_1, \mathbf{y}_2, \mathbf{y}_3\}$, and we choose $\mathbf{y}_2 \in \mathcal{R}^{2\tau}$ as the strength vector $\mathbf{R}_{t_i}$. We choose $y_2$ as the strength vector

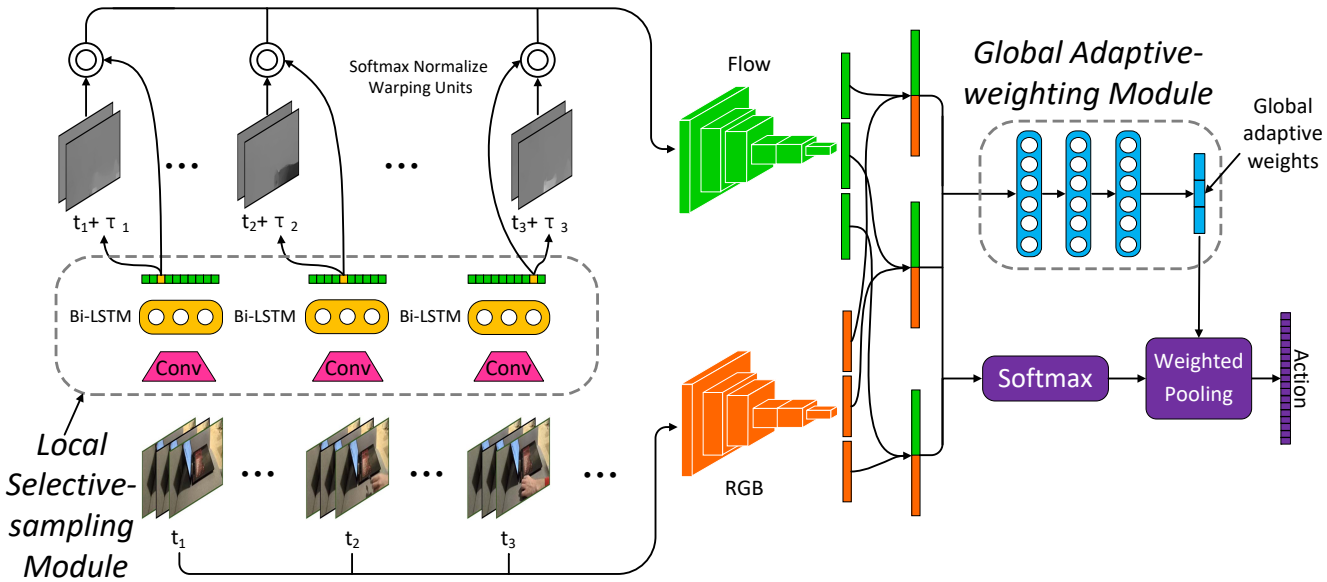



Figure 3: The entire architecture of Cross-stream Selective Networks.

since we use BiLSTM, intuitively $y_2$ can get the information from both directions. Traditionally, sampling optical flow images from a strength vector is a non-differentiable function with respect to the strength vector. In training, we choose 9 flow images for the flow CNN. Thus, we introduce a *Softmax Normalize Warping Unit* to deal with this issue. It takes an optical flow stack sequence and strength vector $\mathbf{R}_{t_i}$ as input and outputs a tensor $T_{t_i}$.

**Softmax Normalize Warping Units.** Suppose $\mathbf{R}_{t_i} \in \mathcal{R}^{2\tau}$ is the strength vector of local RGB sequence. During the training phase, SNW units first sample $M$ optical flow stacks around $t_i$, which corresponds to the top $M$ highest score on $\mathbf{R}_{t_i}$, that is $\{O_{m_i^1}, O_{m_i^2}, \cdots, O_{m_i^M}\}$, where $m_i^j$ is the coordinate of the optical flow stack that corresponds to $j^{th}$ high score on the $\mathbf{R}_{t_i}$. We then input these $M$ high scores to a softmax layer to obtain the weight vector $\mathbf{S}_{t_i}$ and weighted pooling these $M$ optical flow stacks to get $T_{t_1}$,i.e.

$$T_{t_i} = \sum_{k=1}^{N} \frac{\exp(\mathbf{S}_{t_i}(k))}{\sum_{j=1}^{M} \exp(\mathbf{S}_{t_i}(j))} O_{m_i^k}. \tag{2}$$

$\mathbf{S}$ represents weight vector which is consist of $M$ high scores in $R_{t_i}$. We prove that SNW unit makes CSN differentiable during training. Suppose loss function is $\mathcal{L}$, the whole network is $\mathcal{N}$, thus,

$$\frac{\partial \mathcal{L}(I_{t_1}, I_{t_2}, \cdots, I_{t_N}|\mathcal{N})}{\partial W^l} = \sum_{i=1}^{N} \frac{\partial \mathcal{L}}{\partial T_{t_i}} \frac{\partial T_{t_i}}{\partial \mathbf{R}_{t_i}} \frac{\partial \mathbf{R}_{t_i}}{\partial W^l}. \tag{3}$$

According to Eq. 2, we can derive that:

$$\frac{\partial T_{t_i}}{\partial \mathbf{R}_{t_i}} = (B_j - \sum_{q=1}^{M} B_j B_q) O_{m_i^j} \tag{4}$$
$$if \quad j \in \{1, 2, \cdots, M\} \quad else \quad 0,$$

where,

$$B_j = \frac{\exp(\mathbf{S}_{t_i}(j))}{\sum_{q=1}^{M} \exp(\mathbf{S}_{t_i}(q))}. \tag{5}$$

During testing, we simply use the optical flow stack corresponding to the max value in $\mathbf{R}_{t_1}$, as:

$$T_{t_i} = O_{m_i^1}. \tag{6}$$

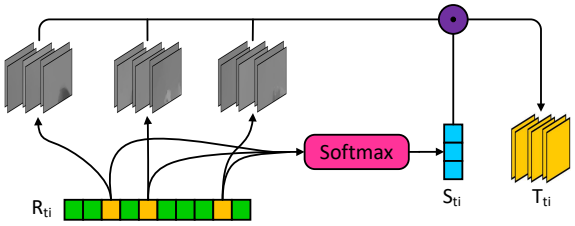
Only one frame is sampled while testing since sampling $N$ optical flow stacks tends to be slow. We will show that such a strategy produce similar results as $N$ sampling during inference.

### 2.3. Global Adaptive-weighting Module

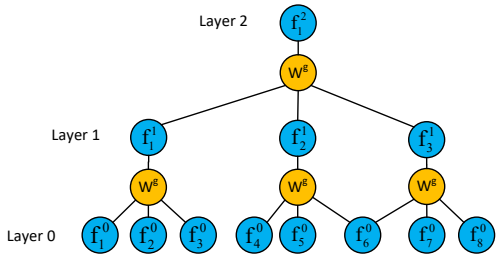
To measure the discriminative capacity of the selected sample groups $\{G_{t_1}, G_{t_2}, \cdots, G_{t_N}\}$, we introduce the Global Adaptive-weighting Module. $\mathbf{f}_{t_i}$ is the concatenation of features representing both RGB images and optical flow of a selected sample group $G_{t_i}$. We learn an attention weight with a linear transformation:

$$A^w = W^g[\mathbf{f}_{t_1}^T, \mathbf{f}_{t_2}^T, \cdots, \mathbf{f}_{t_N}^T]^T, \tag{7}$$

where $W^g \in \mathcal{R}^{N \times ND}$, $\mathbf{f}_{t_i} \in \mathcal{R}^D$ and $A^w \in \mathcal{R}^N$. $D$ is the dimension of $\mathbf{f}_{t_i}$. With this adaptive weight, we obtain the




(a) Softmax Normalize Warping unit (SNW unit)




(b) Weighting Pyramid

Figure 4: Figure **(a)** demonstrates the configuration of SNW unit. Figure **(b)** shows weighting pyramid process and the yellow block in figure **(b)** can be expressed in Eq. 7 and Eq. 8.

final representation and prediction as:

$$\mathbf{f}_c = \sum_{j=1}^{N} \frac{\exp(A^w(j))}{\sum_{i=1}^{N} \exp(A^w(i))} \mathbf{f}_{t_j}, \tag{8}$$

$$y = W^c \mathbf{f}_c, \tag{9}$$

here $W^c$ is the parameter of linear transformation of consensus feature $\mathbf{f}_c$ to final prediction results $y$.

## 3. Experimental Results

### 3.1. Evaluation on Cross-stream Selective Networks

In order to evaluate our model, we compare nine methods: (1) The standard two-stream baseline [11]. (2) Two-stream model equipped with asynchronous fusion network proposed by [9]. (3) Two-stream model equipped with our local selective-sampling module. (4) Temporal Segment Networks (TSN) baseline [18] with fixed interval during evaluation. (5) TSN with a single-stream selective mechanism (cf. Fig. 2a). (6) Our CSN with local selective-sampling module. (7) Our CSN with global adaptive-weighting module. (8) Our CSN with both LSM and GAM (using scaling factor 2). (9) Our CSN with both LSM and GAM (using scaling factor 4). Results on UCF101 and HMDB51 are shown in Tables 1-3.

**Evaluation of LSM.** We first validate the effectiveness of our local selective-sampling module. In Table 1, we compare our LSM and Asyn-Fusion Network. It can be seen

Table 1: Compare LSM with Asyn-Fusion Network [9] (Scaling factor=2).

| Methods | UCF101 | HMDB51 |
|---|---|---|
| Two-stream baseline | 86.9% | 58.0% |
| Asyn-fusion network | 91.0% | 60.9% |
| Two-stream + LSM | **92.1%** | **62.3%** |

Table 2: Ablation study of Global Adaptive-weighting Module.

| Methods | UCF101 | HMDB51 |
|---|---|---|
| TSN baseline (avg) | 94.0% | 68.5% |
| TSN + GAM | **94.2%** | **69.8%** |

Table 3: Compare LSM with Single-stream Selection method. (Scaling factor=2)

| Methods | UCF101 | HMDB51 |
|---|---|---|
| TSN baseline (avg) | 94.0% | 68.5% |
| TSN + Single-stream seletion | 93.8% | 69.5% |
| TSN + LSM | **94.5%** | **70.0%** |

Table 4: Comparison with other state-of-the-art methods.

| Methods | UCF101 | HMDB51 |
|---|---|---|
| C3D (3 nets) [14] | 85.2% | - |
| Two-stream model [11] | 88.0% | 59.4% |
| ST-VLMPF [2] | 93.6% | 69.5% |
| Lattice LSTM [13] | 93.6% | 66.2% |
| TSN (2 modalities) [18] | 94.0% | 68.5% |
| TSN (3 modalities) [18] | 94.2% | 69.4% |
| CO2FI+ASYN [9] | 94.3% | 69.0% |
| TVNet [3] | 94.5% | 71.0% |
| ST-ResNet [4] | 94.2% | 68.9% |
| Our approach | **94.6%** | **71.1%** |

that our LSM is well-designed for that it outperforms Aysn-Fusion Networks [9] which has a little common insights with us. And in Table 3, we prove the necessity of cross-stream selection. In some datasets, single-stream selection would even cause loss in performance. Since there is no existing method to do single-stream selection, we implement it by adopting a similar pipeline in [22]. More details of this implementation will be revealed in code we release once our paper is accepted. Also, some intermediate results show that LSM has really learned some temporal pattern difference between two streams (cf. Fig. 5).

**Evaluation of GAM.** Then, we turn to discuss our global adaptive-weighting module. Consensus the infor-

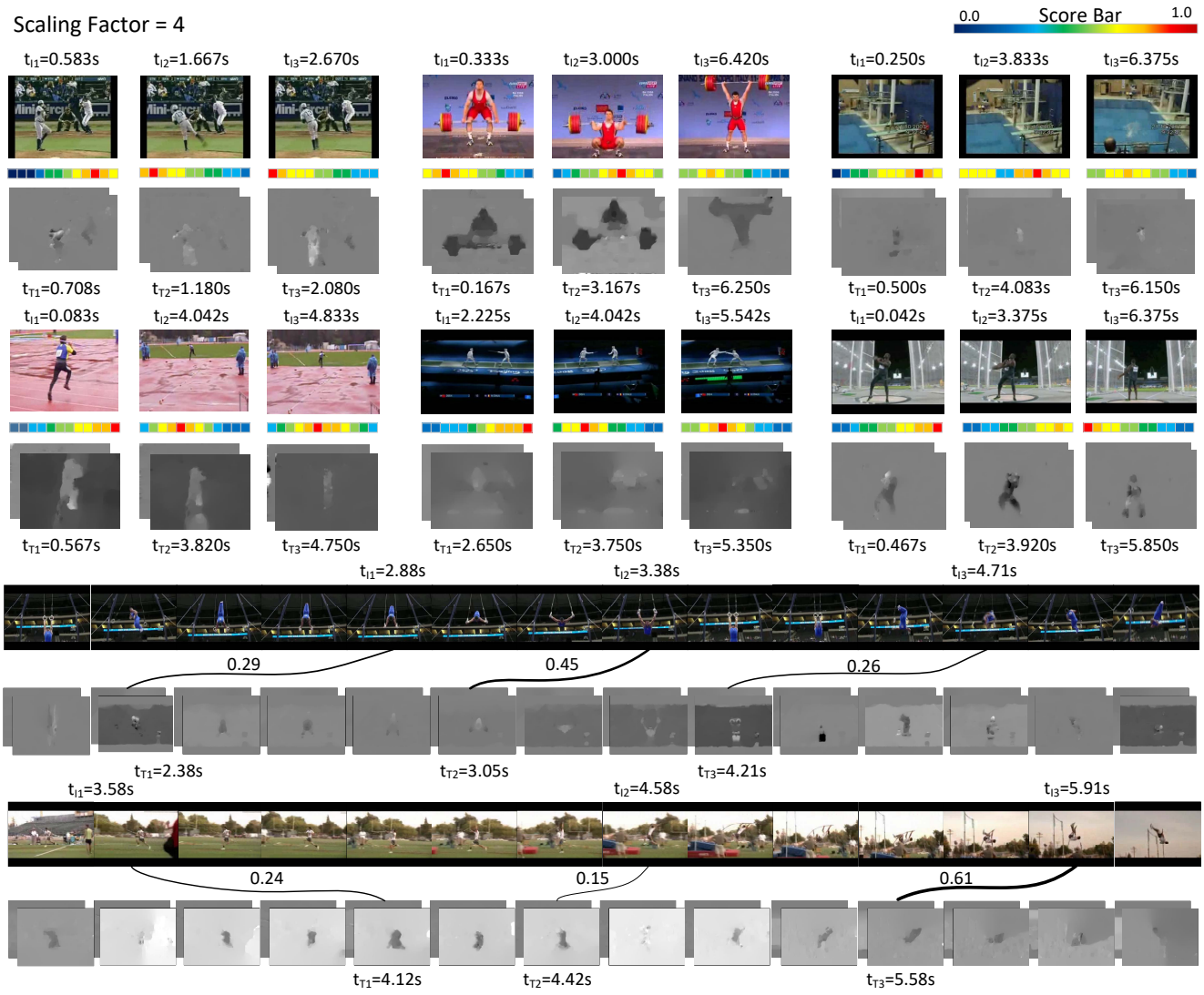



Figure 5: Visualization of of some results of Local Selective-sampling Module and Global Adaptive-weighting Module. Upper part: the intermediate results inside the LSM. It shows that for each RGB image, LSM is able to select out the most match optical flow stack. The heatbar here represents the intensity distribution of strength vector $\mathbf{R}_{t_i}$ introduced in Sec. 2.1. Lower part: the weight output by Global Adaptive-weighting Module. Thicker line indicates a larger weight.

mation from Table 2 and Table 3, we find that our GAM is able to generally improve the recognition accuracy and is easy to be extend to most existing networks. It enables temporal weighting for the importance weight of different RGB&Flow pairs and it is shown in Fig. 5 that different pairs contribute variantly to discriminate the video.

We also evaluated our CSN on Something-something-v1 dataset (it is updated to v2 now) before the submission. Since there is no formal two-stream baseline on Something-something-v1 to be referred before the submission, we just produced it by ourselves. We used the TRN backbone. The top1 accuracy of TRN is 33.01%. And the two-stream version is 38.76%. Equipped with our CSN, the top1 accuracy can be obviously improved to 42.34%.

## 4. Conclusions

We have proposed a cross-stream selective network (CSN) for action recognition by leveraging the correlation and complementarity of different input streams. Our framework consists of three key ingredients: 1) a Local Selective-sampling Module, which can select most discriminative temporal frames aligned to spatial frames. 2) a Global Adaptive-weighting Module which learns to endow different weights for sample RGB&Flow groups. 3) Softmax Normalize Warping Units, which makes the index-to-

feature process to be differentiable. With this framework, we achieve significant performance gain over state-of-the-art methods on both UCF101 and HMDB51, which demonstrates the effectiveness of joint and selective modeling over two streams.

**Acknowledgement** This work is partially supported by Shanghai 'The Belt and Road' Young Scholar Exchange Grant (17510740100).

## References

[1] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4724–4733. IEEE, 2017. 2

[2] I. C. Duta, B. Ionescu, K. Aizawa, and N. Sebe. Spatio-temporal vector of locally max pooled features for action recognition in videos. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3205–3214, 2017. 5

[3] L. Fan, W. Huang, C. Gan, S. Ermon, B. Gong, and J. Huang. End-to-end learning of motion representation for video understanding. *arXiv preprint arXiv:1804.00413*, 2018. 5

[4] C. Feichtenhofer, A. Pinz, and R. P. Wildes. Spatiotemporal multiplier networks for video action recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7445–7454. IEEE, 2017. 5

[5] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. 2016. 1, 2

[6] K. Hara, H. Kataoka, and Y. Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA*, pages 18–22, 2018. 2

[7] D.-A. Huang, V. Ramanathan, D. Mahajan, L. Torresani, M. Paluri, L. Fei-Fei, and J. C. Niebles. What makes a video a video: Analyzing temporal information in video understanding models and datasets. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7366–7375, 2018. 2

[8] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2556–2563. IEEE, 2011. 2

[9] W. Lin, Y. Mi, J. Wu, K. Lu, and H. Xiong. Action recognition with coarse-to-fine deep feature integration and asynchronous fusion. In *AAAI Conf. Artificial Intelligence (AAAI)*, 2018. 1, 2, 5

[10] K. Ohnishi, M. Hidaka, and T. Harada. Improved dense trajectory with cross streams. *CoRR*, abs/1604.08826, 2016. 2

[11] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in neural information processing systems*, pages 568–576, 2014. 1, 5

[12] K. Soomro, A. R. Zamir, and M. Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 2

[13] L. Sun, K. Jia, K. Chen, D. Yeung, B. E. Shi, and S. Savarese. Lattice long short-term memory for human action recognition. In *IEEE International Conference on Computer Vision, ICCV 2017*, pages 2166–2175, 2017. 5

[14] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 2, 5

[15] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri. A closer look at spatiotemporal convolutions for action recognition. In *CVPR*, 2018. 2

[16] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4305–4314, 2015. 2

[17] L. Wang, Y. Xiong, D. Lin, and L. Van Gool. Untrimmednets for weakly supervised action recognition and detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4325–4334, 2017. 1, 2

[18] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision*, pages 20–36. Springer, 2016. 1, 5

[19] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. *CVPR*, 2018. 2

[20] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 461–470. ACM, 2015. 2

[21] S. Yan, Y. Xiong, and D. Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018. 2

[22] S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2678–2687, 2016. 5

[23] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang. Real-time action recognition with enhanced motion vector cnns. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2718–2726. IEEE, 2016. 2

[24] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin. Temporal action detection with structured segment networks. In *The IEEE International Conference on Computer Vision (ICCV)*, volume 8, 2017. 1, 2

[25] B. Zhou, A. Andonian, and A. Torralba. Temporal relational reasoning in videos. *arXiv preprint arXiv:1711.08496*, 2017. 2