Spatio-Temporal Collaborative Module for Efficient Action Recognition

Yanbin Hao, *Member, IEEE,* Shuo Wang, *Member, IEEE,* Yi Tan, Xiangnan He, *Member, IEEE,* Zhenguang Liu, and Meng Wang, *Fellow, IEEE*

***Abstract*—Efficient action recognition aims to classify a video clip into a specific action category with a low computational cost. It is challenging since the integrated spatial-temporal calculation (*e.g.*, 3D convolution) introduces intensive operations and increases complexity. This paper explores the feasibility of the integration of channel splitting and filter decoupling for efficient architecture design and feature refinement by proposing a novel spatio-temporal collaborative (STC) module. STC splits the video feature channels into two groups and separately learns spatio-temporal representations in parallel with decoupled convolutional operators. Particularly, STC consists of two computation-efficient blocks, *i.e.*, $\mathbf{S}_\mathrm{T}$ and $\mathbf{T}_\mathrm{S}$, where they extract either spatial ($\boldsymbol{S}.$) or temporal ($\boldsymbol{T}.$) features and further refine their features with either temporal ($\cdot_T$) or spatial ($\cdot_S$) contexts globally. The spatial/temporal context refers to information dynamics aggregated from temporal/spatial axis. To thoroughly examine our method's performance in video action recognition tasks, we conduct extensive experiments using five video benchmark datasets requiring temporal reasoning. Experimental results show that the proposed STC networks achieve a competitive trade-off between model efficiency and effectiveness.**

***Index Terms*—Efficient action recognition, deep video neural network, channel split, feature contextualization.**

I. Introduction

The advances in data capturing, storage, and communication devices have produced vast amounts of video data in security, defense, consumer and enterprise communities. Efficient action recognition techniques that automatically and accurately extract and model actions/activities from videos are highly desired by various applications, such as video surveillance, large-scale retrieval, robotics, etc. Current deep neural network technologies significantly boost the development of video action recognition, e.g., video CNNs [1]–[4].

The key problems of building efficient video networks lie in how to significantly reduce the model complexity while maintain or even improve the recognition performance. Before reviewing the efficient network architectures, we first give an in-depth look at the video actions. Actions in videos are syntheses of object entities and their interactions taking place in specific surrounding environments. The interactions can be short-term or long-term. Here, we show some example clips in Fig. 1 sampled from the widely used fine-grained video classification datasets. To recognise the action of "*cut bell pepper*", both the objects and their short-term interactions are needed to be modeled. In contrast, for the other two long-term actions, further modeling of long-range dependencies is critical. This indicates that jointly modeling local spatio-temporal patterns and long-range dependencies are essential for a video network, especially when processing long-term actions.

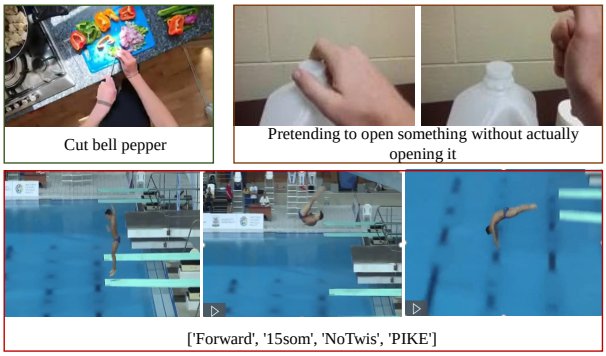


Fig. 1: Sampled clips from EPIC-KITCHENS (top left), Something-Something V1 (top right), and Diving48 (bottom) datasets. The "*cut bell pepper*" is a short-term action, where the blended interactions among hand, knife and bell pepper are key clues. The other two are long-term actions, which require long-term temporal reasoning.

Current efficient video network models mainly focus on designing the lightweight spatio-temporal unit to take the place of the heavy computational 3D convolution. The general low-computation regime relies on kernel decomposition, *i.e.*, 2D spatial convolution plus 1D temporal convolution [5]–[7]. Another line of work uses the parameter-free operations to replace the convolutional operations, e.g., temporal shift [8], spatio-temporal shift [9], and the learnable correlation operator [10]. Moreover, feature channel splitting can also significantly reduce the model parameters [11]. However, these one-size-fits-all spatio-temporal units cannot discriminatively captures diverse video actions. For more powerful video representation learning, there are also some works that pay attention to feature refinement with global contexts, such as non-local neural network [12], temporal excitation and aggregation (TEA) [13], and temporal adaptive module (TAM) [14]. Although these methods improve the performance of their backbones significantly, they inherently incur more extra computation burden.

Y. Hao, S. Wang (corresponding author), Y. Tan and X. He are with the CCCD Key Lab of Ministry of Culture and Tourism, School of Data Science, School of Information Science and Technology, University of Science and Technology of China, Anhui, 230026, China. E-mail: haoyanbin@hotmail.com, shuowang.hfut@gmail.com, ty133@mail.ustc.edu.cn, xiangnanhe@gmail.com.

Z. Liu is with School of Cyber Science and Technology, Zhejiang University, Zhejiang, 310058, China. E-mail: liuzhenguang2008@gmail.com.

M. Wang is with School of Computer Science and Information Engineering, School of Artificial Intelligence, Hefei University of Technology, Anhui, 230009, China. E-mail: eric.mengwang@gmail.com.

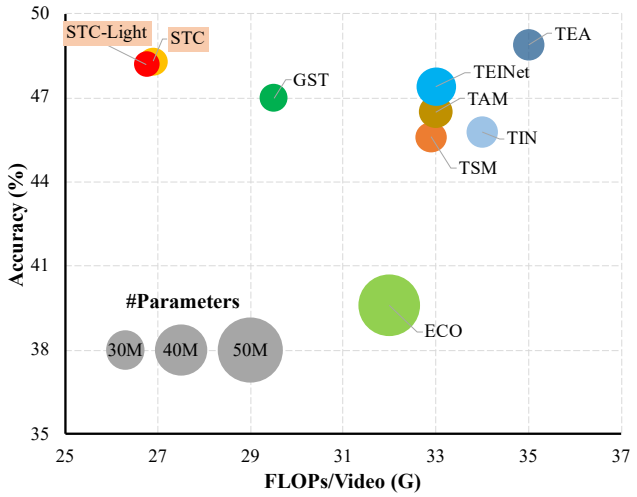


Fig. 2: Performance and complexity comparison on the action classification dataset Something-Something V1 [15]. Our STC and its lightweight version STC-Light show better trade-off between accuracy and efficiency, when compared with previous methods such as ECO [16], TIN [17], TEINet [18], TSM [4], GST [11], TAM [14] and TEA [13]. Except ECO adopts ResNet-18 as backbone, all the other methods use ResNet-50 as backbone.

In this work, we propose a new efficient spatio-temporal paradigm, *i.e.*, spatio-temporal collaborative (STC) module, which can jointly model the local spatio-temporal feature pattern and the global long-range dependency with a much lower computational load. STC combines the channel splitting (or group convolution) and temporal shift to achieve efficient modeling and meanwhile refines features with global spatial/temporal contexts aggregated from the other axis. Specifically, the proposed STC module consists of two spatio-temporal operation blocks, named S_T and T_S, where S_T firstly extracts ***spatial*** feature from a channel group and then refine it with ***temporal*** contexts, while in contrast, T_S firstly extracts ***temporal*** feature from the other channel group and then refine it with ***spatial*** context. S_T and T_S process their feature channel groups in parallel. The two resulted features are finally concatenated along the channel dimension.

In particular, S_T uses the "temporal shift + 2D convolution" to achieve spatio-temporal modeling. As temporal shift operation is computation-free, this combination is much more efficient than the 3D convolution used in GST [11]. Moreover, we introduce a feature refinement block that imposes global spatial context to influence each channel element along the temporal dimension. While in T_S, we use a 1D convolution to model temporal information from the other channel group and build a mirror refinement block to explore the impact of global temporal context on spatial pixels. Since spatial and temporal contents work together for video representation learning in S_T and T_S, we thus name our proposed module as spatio-temporal collaborative (STC) module. Compared to GST, our STC emphasizes more temporal modeling and also exploits the global contextual information for feature refinement, making it more capable of capturing long-term actions as demonstrated in the experiments. In the implementation, we provide two variants of STC, *i.e.*, the standard STC and lightweight STC. We evaluate their efficiency and effectiveness in various fine-grained action classification benchmarks. Fig. 2 shows that both STC variants achieve better accuracy-cost trade-off than the other ResNet-based video neural network models.

The contributions of our method are threefold:

- **Two lightweight and powerful spatio-temporal units**. The proposed S_T and T_S blocks can achieve both neighboring local spatio-temporal and long-range dependency modeling with low computational costs.
- **Efficient action recognition model**. We construct two variants of STC models for efficient action recognition through separately processing two paralleled feature groups with S_T and T_S. The STC/STC-Light model has only 0.92/0.84× parameters and 0.82/0.81× FLOPs of the standard 2D ResNet.
- **Competitive trade-off between model efficiency and effectiveness**. We verify our STC models on five video datasets that require temporal modeling. Experiments demonstrate that STC variants can achieve a competitive trade-off between efficiency (parameters/FLOPs) and effectiveness (action recognition accuracy).

II. Related Work

We briefly review the deep neural networks for video action recognition and organize them according to their temporal modeling strategies and model complexities. Then, we review the related attention mechanisms proposed for video feature refinement.

Classical Deep Video Architectures. Deep convolutional neural networks (CNNs) have pride of place in computer vision. The earliest video networks [19]–[23] make efforts to temporally aggregate spatial features extracted by 2D CNNs (e.g., VGG [24], ResNet [25]) among frames. The temporal aggregation mainly includes frame feature fusion [19], [21], [22] and frame score fusion [20]. For example, video CNN [19] proposes to fuse the frame features at the early or/and late layers, dynamic image networks [21] use rank pooling machine to encode the temporal evolution across frames, while temporal relation networks (TRN) [22] introduce extra multilayer perceptrons to model temporal relations. In contrast, temporal segment network (TSN) [20] averagely pools the predicted frame-level scores to obtain the video score. Although these video networks incur little overhead to model complexity, having similar model complexities with the 2D CNN backbones, they are usually failed to recognise complex dynamics.

Instead of simply adopting the pooling operation on the frames, there are also some works that use more advanced feature fusion techniques to organize the frame-level features. For example, ActionS-ST-VLAD [26] uses the vector-based encoding method-vector of locally aggregated descriptors (VLAD) to learn video-level representation. The work [27] extracts various CNN latent concept descriptors and applies video pooling to them to obtain video representation. Attention clusters [28] adopt the attention mechanism to integrate the video local features. The works [29], [30] input the features

extracted from 2D CNN into the LSTM to model the temporal relations among those continuous video frames. In the work [31], the authors present an energy optimization method for dynamic texture extraction and recognition in videos. Apart from the RGB features, TVNet [32] proposes to learn optical-flow-like features from videos by unfolding the iterations of the TV-L1 [33] method to customized neural layers.

Research preferences are then shifted to constructing spatio-temporal units, which can jointly model neighboring local spatio-temporal patterns. The most representative works are 3D CNNs, such as C3D [1], I3D [2], ECO [16] and V4D [34]. C3D directly replaces each 2D convolution of a 2D network with a 3D convolution. I3D shares the same network architecture with C3D but inflates an ImageNet [2] pretrained 2D model to initialize network by weight copying. ECO [16] tops a 3D net on the 2D net to achieve temporal modeling. While, V4D [34] proposes to use 4D convolution to model clip-level relations. There are also other attempts to enable rich video content extraction. For example, the work [35] applies the deep manifold learning to a convolutional layer to learn more discriminative features for action recognition. The pairwise two-stream ConvNets (PTC) [36] adaptively combine the RGB and flow feature to learn domain-invariant features for cross-domain action recognition. Since the use of heavy computational operators, these video network models have huge number of parameters and are learning inefficient.

Efficient Deep Video Architectures. To tackle the problem of high computational cost of 3D CNNs, academic efforts have been made to design efficient deep architectures and improve network flexibility. Example works include P3D [5], R(2+1)D [37], S3D [38], SlowFast [39], X3D [40], CoST [41], GST [11], bLVNet [42], TSM [8], GSM [43] and RubikShift [9]. P3D, R(2+1)D, S3D and SlowFast propose to decompose the 3D convolution to the combination of 2D spatial convolution and 1D temporal convolution, while CoST performs 2D convolution along three orthogonal views of video data to achieve the similar function of 3D convolution. SlowFast further builds slow and fast pathways to explore the resolution trade-off across axes. X3D reduces the model parameters by progressively expanding a 2D network across several axes. GST decomposes the feature channels into spatial and temporal groups in parallel. bLVNet operates on both low-resolution and high-resolution frames with the use of depthwise temporal aggregation. Different to those decomposition regimes, another direction is to propose computation-free operators. For example, TSM replaces the 1D temporal convolution with the shift operation along time axis, sharing the same number of parameters and FLOPs with C2D. GSM extends TSM with learnable shift gates. Moreover, RubikShift even replaces all convolutional filters with lightweight spatial/temporal shift operations. The proposed STC adopts the channel splitting strategy used by GST but replace the spatial-only (2D convolution) and temporal-only (1D convolution) operators with two integrated spatio-temporal blocks.

Attentions for Video Feature Refinement. Attention mechanisms show promising performance on modeling long-range dependencies. In video action recognition, the long-range dependencies can be reflected in space and time axes. Existing works make efforts towards the exploration and utilization of global perspective contexts for feature refinement. For example, the non-local network [12] recomputes each of local feature points in the 3D feature map as a weighted sum of feature responses of its all spatio-temporal neighbors. S3D-G [7] refine the learnt feature of S3D by using the global spatio-temporal contexts squeezed along the channel dimension in a gating manner. Temporal excitation and aggregation (TEA) [13] extends the squeeze-and-excitation network (SE-Net) [44] proposed for image processing to enhance models with aggregated temporal context. While, TAM [14] proposes to refine the layer-wise feature with global temporal context adaptively. The work [45] proposes two types of attention mechanism called statistic-based attention (SA) and learning-based attention (LA) to attach higher importance to the crucial elements in each video frame. STA-CNN [46] incorporates a temporal attention mechanism and a spatial attention mechanism into a unified convolutional network to recognize actions in videos. Compared to prior works, STC takes both spatial and temporal axial contexts into account and achieves paralleled feature refinement with channel group operation.

III. Spatial-Temporal Collaborative Module

Spatio-temporal collaborative module replaces the inefficient 3D $3 \times 3 \times 3$ convolution (Fig. 3(a)) in a residual layer with two efficient spatio-temporal operational blocks, i.e., $\mathrm{S_T}$ and $\mathrm{T_S}$, for video representation learning. As shown in Fig. 3(d), STC firstly splits the input channels into two groups and then separately processes them by $\mathrm{S_T}$ and $\mathrm{T_S}$ in parallel. Different to the operations used in P3D/GST that only processes one specific aspect information (spatial-only or temporal-only) in one channel group, both the two $\mathrm{S_T}$ and $\mathrm{T_S}$ blocks in STC can achieve sptatio-temporal information encoding. Benefiting from the use of channel splitting, temporal shift and global pooling operations, the proposed STC models are much more efficient and have even fewer parameters than a standard 2D network counterpart (i.e., ResNet-50). In the following sections, we elaborate the details of STC module, including the designing and computational analysis of $\mathrm{S_T}$ and $\mathrm{T_S}$ and the final video classification network architecture.

A. *Collaborative Blocks* $\mathrm{S_T}$ *and* $\mathrm{T_S}$

Channel splitting, aka channel decomposition, is an efficient way to reduce model complexity in both image and video neural networks, and separately modeling appearance and motion cues on the two split channel groups has been demonstrated to be effective for video action recognition. However, we argue that the spatial/temporal-only feature can be further refined by exploring and utilizing the contextual information from the other perspective. Our STC is inspired by GST and TSM but replaces the 2D spatial-convolution with a newly designed $\mathrm{S_T}$ operational block and the 3D spatio-temporal convolution with the other $\mathrm{T_S}$ operational block. Formally, lets denote the input feature map of our STC module as $\mathbf{X} \in \mathbb{R}^{T \times H \times W \times C}$, where T, H, W, and C are the size of the temporal, height, width, and channel, respectively. Different to the work of GST (Fig. 3(c)) that splits the feature channels equally to two groups, we

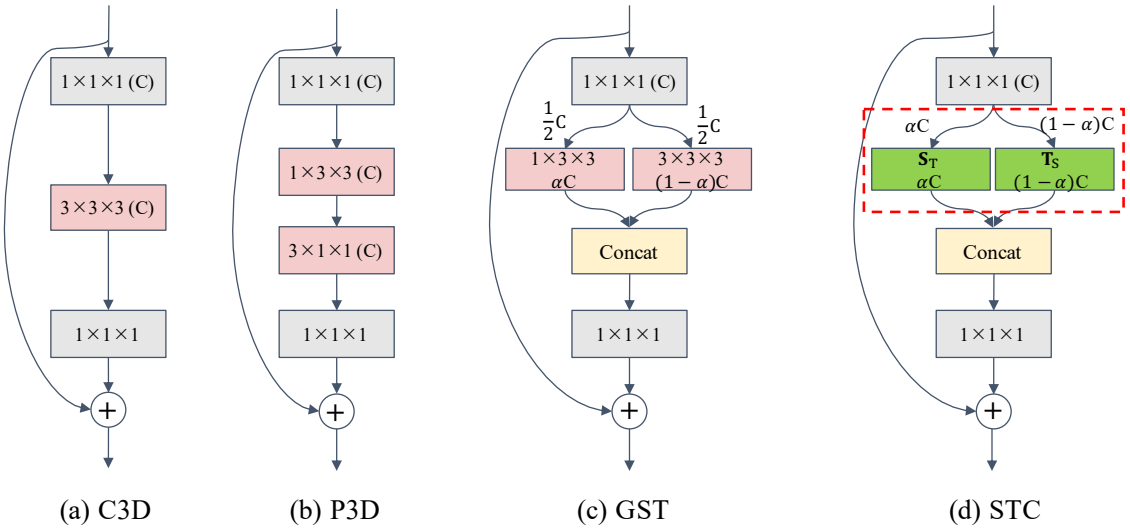


Fig. 3: Architecture comparison between the existing networks and our STC module. (a) shows the 3D convolution network (C3D). (b) shows the P3D block, which decouples the spatial and temporal filters. (c) shows a GST module, which decomposes the feature channels into spatial and temporal groups in parallel. (d) shows the proposed STC module.

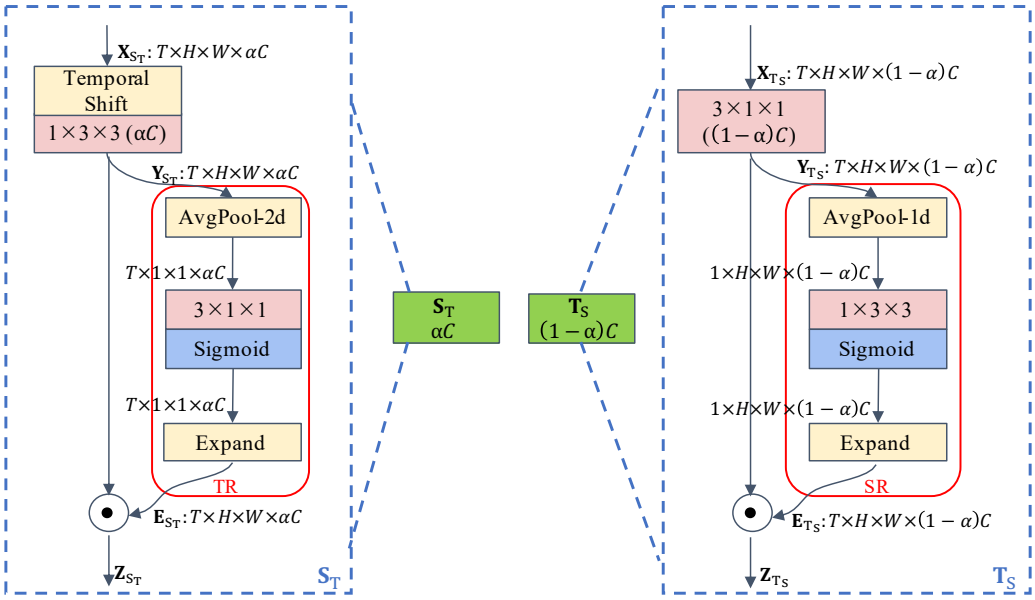


Fig. 4: The details of the spatial-temporal collaborative blocks in our standard STC module (S_T and T_S).

introduce a hyper-parameter α to control the splitting ratio, as shown in Fig. 3(d). Below, we illustrate S_T and T_S blocks, as well as their light-weight versions, in detail.

Spatial-temporal Modeling with Temporal Refinement, S_T Block. As shown in the left part of Fig. 4, given the input feature $\mathbf{X}_{S_T} \in \mathbb{R}^{T\times H\times W\times \alpha C}$, we use a 2D convolution with $1\times3\times3$ kernel to capture the appearance information from these spatial channels. As temporal information is much more useful in temporal reasoning, we further introduce the temporal shift with moving 1/8 channels as in TSM before the 2D convolution to model the temporal interaction in channel dimension. Then, we build a temporal refinement (TR) block to refine the learnt feature by considering the global spatial contextual information. Specifically, we first shrink the learnt feature along the spatial dimension using 2D average pooling, maintaining temporal $T \times 1 \times 1 \times \alpha C$ statistics, then adopt the 1D convolution with kernel $3 \times 1 \times 1$, padding 1 and stride 1, to mix the global spatial contextual information within a small temporal receptive field, and finally we use the sigmoid function and expand operation to calculate an element-wise weight tensor with values in the range of (0.0, 1.0). Suppose that the learnt feature by 2D convolution is $\mathbf{Y}_{S_T} \in \mathbb{R}^{T\times H\times W\times \alpha C}$ and the element-wise weight tensor is $\mathbf{E}_{S_T} \in \mathbb{R}^{T\times H\times W\times \alpha C}$. The final output feature $\mathbf{Z}_{S_T} \in \mathbb{R}^{T\times H\times W\times \alpha C}$ can thus be computed by

$$\mathbf{Z}_{S_T} = \mathbf{E}_{S_T} \odot \mathbf{Y}_{S_T}, \tag{1}$$

where $\odot$ denotes the Hadamard product.

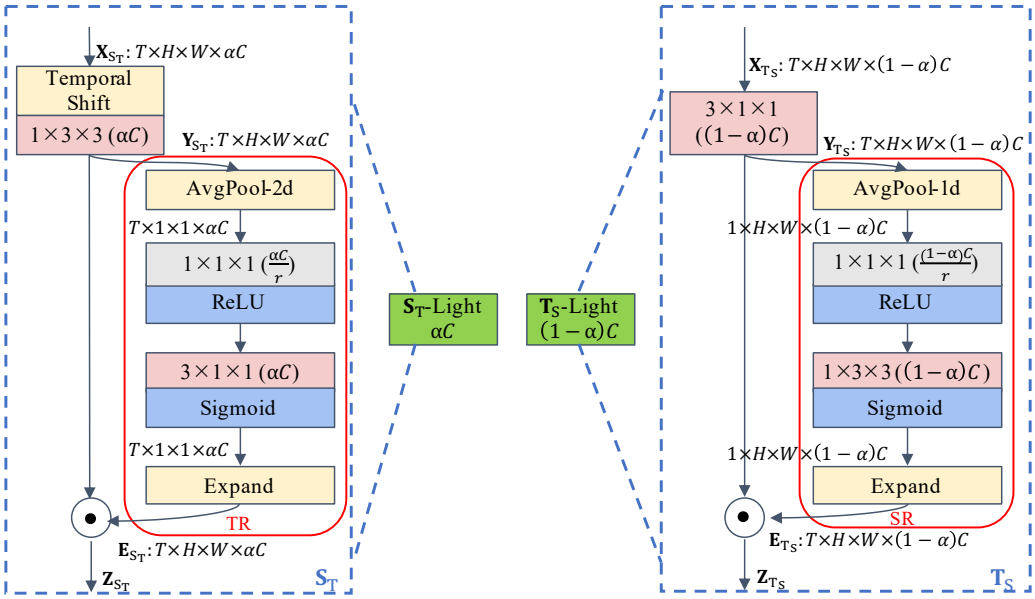


Fig. 5: The details of the lightweight STC module (S_T-Light and T_S-Light).

Temporal Modeling with Spatial Refinement, T_S Block. As shown in the right part of Fig. 4, the calculations of T_S are similar mirroring operations to S_T. In particular, given the input feature $\mathbf{X}_{T_S} \in \mathbb{R}^{T\times H\times W\times(1-\alpha)C}$, we use a 1D $3\times1\times1$ convolution to capture the temporal information across frames, resulting in a $T\times H\times W\times(1-\alpha)C$ output feature $\mathbf{Y}_{T_S}$. Afterwards, similar to the operation in S_T, we build a spatial refinement (SR) block to utilize the global temporal information to refine the learnt feature $\mathbf{Y}_{T_S}$. Specifically, we averagely pool $\mathbf{Y}_{T_S}$ along the temporal dimension, representing the temporal context as a $1\times H\times W\times C$ global temporal matrix. Then, a 2D $1\times3\times3$ convolution (padding 1 and stride 1) is used to compute the impact of the aggregated temporal context to each spatial location. Finally, the sigmoid function and expand operation are used to obtain the element-wise weights. Formally, the final output feature is computed as follows:

$$\mathbf{Z}_{T_S} = \mathbf{E}_{T_S} \odot \mathbf{Y}_{T_S}. \tag{2}$$

The features $\mathbf{Z}_{S_T}$ and $\mathbf{Z}_{T_S}$ calculated by S_T and T_S respectively are further concatenated together in channel dimension as

$$\mathbf{Z} = \mathrm{Concat}(\mathbf{Z}_{S_T}, \mathbf{Z}_{T_S}), \tag{3}$$

where $\mathbf{Z} \in \mathbb{R}^{T\times H\times W\times C}$ is the ultimate learnt feature map of STC.

Lightweight S_T and T_S. Here, we also present the lightweight version of the proposed S_T and T_S blocks, which introduces fewer parameters without significant performance drop as demonstrated in the experiment. As shown in Fig. 5, before feeding into the gating weight computation unit, the lightweight S_T and T_S additionally use a linear projection to reduce the number of feature channels with a reduction ratio r in the refinement block. In implementation, we adopt a $1\times1\times1$ convolution to achieve the channel reduction and the relu function to get the new feature map. The two newly resulted feature refinement blocks are similar to the squeeze-and-excitation module [44] in model structure, i.e., the bottleneck structure, but differ in contextual information modeling. We denote the STC module with lightweight S_T and T_S blocks as STC-Light.

Our proposed STC modules take full advantages of the channel splitting and lightweight computational unit (*i.e.*, 1D and 2D convolution and temporal shift) in efficient video action modeling. The element-wise refinement blocks further exploit global contextual information (*i.e.*, the temporal context in S_T and the spatial context in T_S) to refine the learnt feature by attending on a specific feature aspect. In this case, the S_T and T_S, as well as their lightweight versions, are not only confined to a single local convolutional region. Moreover, the use of global average pooling in refinement blocks does not incur significant memory consumption.

B. Computational Analysis

TABLE I: Comparison of the number of parameters for different modules. For clarity, we use $C = 64$ (Stage-1 in ResNet-50) as an example to compute the detailed number of parameters. The hyperparameters are set to $\alpha = 3/4$ and $r = 4$, which are the final settings of our STC and STC-Light. For GST, α is set to $3/4$ following the setting in their paper.

Model	Params	Result ($C = 64$)	Percentage
C2D	$9\times C^2$	36,864	100.00%
C3D	$27\times C^2$	110,592	300.00%
P3D	$12\times C^2$	49,152	133.33%
GST	$9\times(3/2-\alpha)\times C^2$	27,648	75.00%
STC	$12\times[\alpha^2+(1-\alpha)^2]\times C^2$	30,720	83.33%
STC-Light	$[(9+4/r)\alpha^2+(3+10/r)(1-\alpha)^2]\times C^2$	24,448	66.32%

The channel split ratio α and the channel reduction ratio r are used to control and specify the complexity of S_T and T_S blocks, where r is only for the lightweight versions. For

the standard S_T and T_S, the total parameters are: $12\alpha^2C^2$ and $12(1-\alpha)^2C^2$. For the light-weight S_T and T_S, the total parameters are: $(9+4/r)\alpha^2C^2$ and $(3+10/r)\alpha^2C^2$. Thus, we can easily compute the number of parameters of STC modules. As the existing modules and the proposed STC modules only differ in the middle layer of the residual block, in Table I, we only list their middle layer parameters for comparison. From this table, we can find that the proposed STC and STC-Light models have even less number of parameters than the 2D C2D model when properly setting α and r. For example, when setting $\alpha = 3/4$, the number of parameters of STC (STC-Light) layer is only 83.33% (66.32%) of C2D's but enables multi-scale temporal modeling and multiple kinds of contextual information exploring.

C. Network Architecture

The basis network instantiation follows a ResNet structure. We replace each of the feature filtering layer in the ResNet, for example, the $3 \times 3 \times 3$ convolutional layer in C3D model, with the proposed efficient STC modules. To predict the probability distribution of action classes of the entire input video, we follow the strategy of TSN [20] that averagely pools the frame-level action scores among all the video frames. In terms of model architecture, our proposed STC model is similar to GST model. In the experiment, we show that the STC models can achieve better performance while require lower computational burden than GST.

IV. Experiments

A. Datasets and Evaluation Metrics

The proposed STC models focus on modeling actions that need temporal reasoning. Thus, we select five benchmarked video datasets that contain a broad range of complex dynamic actions such as human-object interactions, human poses and first-person vision actions, to evaluate our method.

Something-Something. Something-Something datasets have two versions, V1 [15] and v2 [47], and contain ∼110k (V1) and ∼220k (V2) video clips for 174 fine-grained action categories. The video clips show humans performing pre-defined basic actions with everyday objects, and thus require long-term temporal modeling to describe the interactions among human, object and surroundings. We report the performance on the validation set.

Diving48. Diving48 [48] is a fine-grained video dataset of competitive diving, consisting of ∼18k trimmed video clips of 48 unambiguous dive sequences. As dives differ in multiple stages, the action recognition requires modeling of long-term temporal dynamics. The holder recently updated the dataset by manually cleaning dive annotations and removing poorly segmented videos. We conduct experiments on the updated version using the new official train/validation split V2.

Egocentric Video Datasets. EGTEA Gaze+ [49] and EPIC-KITCHENS [50] offer first-person vision actions, covering a wide range of non-scripted daily activities and involving rich human-object interactions occurred in native environments. Specifically, EGTEA Gaze+ contains ∼10k instances of fine-grained actions for 106 activity classes. We use the three official train/validation splits for performance examination. While for EPIC-KITCHENS, we select the EPIC-KITCHENS-55 for use and report the verb and noun classification results following the train/validation splitting mechanism of [51]. The number of action instances in the training and validating sets are 23,191 and 5,281 respectively.

We evaluate the video classification performance with top-1/5 accuracy (%) and also report the number of parameters and FLOPs (floating point operations) to clearly compare the model complexity. Here, FLOPs measure the computational operations to run a single instance of a given model. Fewer FLOPs show that the model is more efficient. In the experiment, we use the python package Thop[1] to calculate the parameters and FLOPs.

B. Implementation Details

All STC variants are implemented in Pytorch and run on servers with 4×2080Ti or 3090 GPUs. We adopt ResNet-50 [25] pretrained on ImageNet [52] as the backbone. The parameters of newly added layers are randomly initialized.

Training and Inference. In the **training**, We use the uniform sampling method described in TSN [20] to obtain input frames. The uniform sampling method first separates the video frames into T equally sized groups along the time line and randomly select one frame from each group as a representative. The selected frames are further resized with the short-size as 240 for Something-Something datasets and as 256 for others and their original aspect ratios are kept. During training, a 224×224 patch is cropped out of the center of the frame, and then the center crop is randomly scaled within the range of $\{1, 0.875, 0.75, 0.66\}$ and randomly horizontal flipped for data augmentation. We train the network with a batch size 10 per GPU and optimize using SGD with an initial learning rate 0.01 for 50 epochs and decay it by 0.1 at epoch 30 and 40. The dropout is set to 0.5. In the **inference**, we report top-1 and top-5 accuracies and compute them on the evaluation set. The video frame selection here is also a uniform sampling but uses different selection strategies for different clip settings. Specifically, for 1-clip sampling only the middle frame in each of T frame groups is selected, and for 2-clip sampling both the first and the middle frames in each of T frame groups are selected. The two clip sampling strategies are proposed by TSN [20]. While for 10-clip sampling proposed by TEA [13], it randomly selects a frame from each group and repeats this operation 10 times to obtain 10 clips. It is worth noting that the time interval between any two neighboring sampling frames for the 1-clip and 2-clip sampling strategies is fixed, while it is a variable-sized one for the 10-clip sampling. The resolution of center crop is fixed to 224×224 for all experiments. Particularly, we sample one clip per video for the ablation study and multiple clips (2 or 10) per video for the final comparison with SOTAs following [8], [13], [20] on Something-Something datasets. While for the other datasets, we report the experimental results with 8 frames input and 1 clip per video. The number of sampled frames and clips will be specified in the tables.

[1]Pytorch-OpCounter: https://github.com/Lyken17/pytorch-OpCounter

C. Ablation Study

In this subsection, we present ablation studies on Something-Something V1 dataset to study the impacts of hyperparameters, including the channel splitting ratio α and the channel reduction ratio r, temporal shift and feature refinement.

TABLE II: Performance comparison with different α and r on the validation set of Something-Something V1. For all methods, we use ResNet-50 as the backbone and uniformly sample 8 frames per video. The results of C3D, P3D and GST are referenced from [11]. We fix the $\alpha = 3/4$ for STC-Light.

Method	α	Params	FLOPs/video	Top-1/Top-5 (%)
TSN	None	23.9M	32.9G	17.7/46.6
C3D	None	42.5M	62.5G	46.2/75.6
P3D	None	29.4M	37.8G	45.7/75.0
TSM	None	23.9M	32.9G	45.6/74.2
GST	3/4	21.0M	29.5G	47.0/76.1
GST-Large	3/4	29.6M	40.4G	47.7/76.4
STC (only T_S)	0	27.6M	32.9G	24.9/50.8
STC	1/4	22.0M	22.8G	41.9/71.3
STC	2/4	20.1M	23.5G	46.4/75.7
STC	3/4	22.0M	26.9G	**48.3/77.4**
STC (only S_T)	1	27.6M	32.9G	47.3/76.0
STC-Equal	3/4	20.1M	24.6G	47.0/76.3
$\alpha = 3/4$	r			
STC-Light	2	21.0M	26.80G	48.2/77.3
STC-Light	4	20.1M	26.76G	**48.2/77.4**
STC-Light	8	19.6M	26.75G	47.8/77.2
STC-Light	16	19.4M	26.75G	47.7/77.1

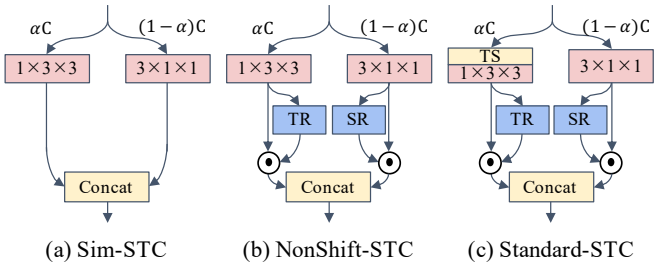


Fig. 6: STC variants w/ and w/o temporal shift and feature refinement.

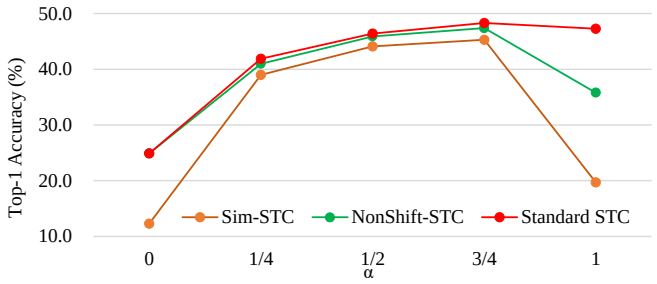


Fig. 7: Performance comparison of STC variants w/o temporal shift and feature refinement on Something-Something V1.

Settings of α and r. The channel splitting ratio α controls the proportion of input channels of S_T and T_S blocks. Larger value of α results in more channels being input to S_T branch for modeling spatio-temporal patterns. r is introduced for channel reduction in the refinement blocks specialized in STC-Light variants. Particularly, we use $\max(8, \alpha C/r)$ (S_T-Light) and $\max(8, (1-\alpha)C/r)$ (T_S-Light) to limit the channel size by a minimum of 8, which is similar to the operation in SK-Net [57]. Here, we examine α with values of $\{0, 1/4, 2/4, 3/4, 1\}$ and r with values of $\{2, 4, 8, 16\}$. For comparison clarity, we integrate the two performance results into one table, as shown in Table II. It can be found that when setting $\alpha = 3/4$, our STC model obtains best performance among these competing methods, which shows that the channel splitting mechanism significantly improves performance compared to the non-split C3D, P3D and TSM. Also, leaving more channels into spatio-temporal modeling, *i.e.*, a larger α, generally benefits the actions containing rich temporal object interactions. Compared to the GST-Large which uses the same channel splitting strategy with STC, STC not only achieves better performances (48.3%/77.4% vs. 47.7%/76.4% of GST-Large) but also requires much less computation costs (22.0M parameters and 26.9G FLOPs vs. 29.6M parameters and 40.4G FLOPs of GST-Large). STC-Equal follows the channel splitting strategy used by GST that splits the input channels into two equal groups and controls the output channels using α. We observe that STC-Equal is more efficient (20.1M parameters and 24.6G FLOPs) than GST (21.0M parameters and 29.5G FLOPs) while having the similar top-1/5 results. For the STC-Light variants, increasing the channel reduction ratio r does not result in significant performance degradation while can further reduce the number of parameters. By considering the trade-off between performance and parameters, we set $r = 4$ for STC-Light.

In terms of model complexity, the proposed efficient computational units S_T and T_S in STC greatly reduce model complexity (22.0M vs 42.5M of C3D, 26.9G vs 62.5G of C3D), and the STC-Light containing 20.1M parameters and 26.8G FLOPs is even superior to the much efficient GST model (21.0M parameters and 29.5G FLOPs) in both classification performance and model complexity.

Temporal shift and feature refinement. We next examine the functions of temporal shift and feature refinement blocks (*i.e.*, TR and SR) of STC by peeling them away one by one. Fig. 6 shows the resulted three STC variants, including the simplified STC (Sim-STC (a)), non-temporal-shift STC (NonShift-STC (b)) and standard STC (c). We report their top-1 accuracy comparison in Fig. 7 with the settings of $\alpha = 0, 1/4, 2/4, 3/4, 1$. We observe that the NonShift-STC consistently outperforms the Sim-STC, indicating the effectiveness of feature refinement.

TABLE III: Performance comparison w/wo temporal refinement (TR) and spatial refinement (SR) blocks on Something-Something V1 dataset. Note that temporal shift is used by S_T.

α	TR (S_T)	SR (T_S)	Top-1 (%)
0	✘	✘	12.3
	✘	✔	24.9
3/4	✘	✘	46.9
	✔	✘	48.1
	✘	✔	47.8
	✔	✔	48.3
1	✘	✘	45.6
	✔	✘	47.3

TABLE IV: Performance comparison of state-of-the-arts on Something V1 and V2 datasets.

Method	Backbone	#Pretrain	Frames×Crops×Clips	Params	FLOPs	V1		V2	
						Top-1	Top-5	Top-1	Top-5
TSN [20]	ResNet-50	ImageNet	8×1×1	23.9M	32.9G	19.7	46.6	30	60.5
ECO [16]	ResNet-18	Kinetics	8×1×1	47.5M	32G	39.6	—	—	—
ECO [16]			16×1×1	47.5M	64G	41.4	—	—	—
I3D [2]	3DResNet-50	ImageNet	32×1×2	28.0M	153.0G×1×2	41.6	72.2	—	—
NLI3D [12]				35.3M	168.0G×1×2	44.4	76	—	—
NLI3D+GCN [53]				62.2M	303.0G×1×2	46.1	76.8	—	—
TSM+TPN [54]	ResNet-50	ImageNet	8×1×1	24.3M	33.0G×1×1	49	—	62	—
TIN [17]	ResNet-50	Kinetics	8×1×1	24.3M	34.0G×1×1	45.8	75.1	—	—
TIN [17]	ResNet-50	Kinetics	16×1×1	24.3M	67.0G×1×1	47.0	76.5	60.1	86.4
TEINet [18]	ResNet-50	ImageNet	8×1×1	30.4M	33.0G×1×1	47.4	—	61.3	—
TEINet [18]			16×1×1	30.4M	66.0G×1×1	49.9	—	62.1	—
RubiksNet [9]	ResNet-50	ImageNet	8×1×2	—	—	46.4	74.5	61.7	87.3
TAM [14]	ResNet-50	ImageNet	8×1×1	25.6M	33.0G×1×1	46.5	75.8	60.5	86.2
TAM [14]			16×1×1	25.6M	66.0G×1×1	47.6	77.7	62.5	87.6
bLVNet-TAM [42]	bLResNet-50	ImageNet	8×1×2	25.0M	23.8G×1×2	46.4	76.6	59.1	86.0
GST [11]	ResNet-50	ImageNet	8×1×1	21.0M	29.5G×1×1	47.0	76.1	61.6	87.2
GST [11]			16×1×1	21.0M	59.0G×1×1	48.6	77.9	62.6	87.9
TSM [4]	ResNet-50	ImageNet	8×1×2	23.9M	32.9G×1×2	47.3	76.2	61.7	87.4
TSM [4]			16×1×2	23.9M	65.8G×1×2	48.4	78.1	63.1	88.2
SmallBig [55]	ResNet-50	ImageNet	8×3×2	—	57.0G×3×2	48.3	78.1	61.6	87.7
SmallBig [55]			16×3×2	—	114.0G×3×2	50.0	79.8	63.8	88.9
V4D [34]	V4DResNet-50	None	8×10×3	—	—	50.4	—	—	—
STM [56]	ResNet-50	ImageNet	8×3×10	24.0M	33.3G×3×10	49.2	79.3	62.3	88.8
STM [56]			16×3×10	24.0M	66.5G×3×10	50.7	80.4	64.2	89.8
TEA [13]	ResNet-50	ImageNet	8×1×1	24.5M	35.0G×1×1	48.9	78.1	—	—
TEA [13]			8×3×10	24.5M	35.0G×3×10	51.7	80.5	—	—
TEA [13]			16×3×10	24.5M	70.0G×3×10	52.3	81.9	—	—
STC	ResNet-50	ImageNet	8×1×1	22.0M	26.9G×1×1	48.3	77.4	61.2	87.1
			8×3×2	22.0M	26.9G×3×2	50.1	79.1	63.3	88.7
			8×3×10	22.0M	26.9G×3×10	51.1	79.7	63.4	88.6
			16×1×1	22.0M	53.6G×1×1	50.5	79.3	63.7	88.7
			16×3×2	22.0M	53.6G×3×2	51.7	80.2	65.1	90.0
			16×3×10	22.0M	53.6G×3×10	52.2	80.7	65.3	89.1
STC-Light	ResNet-50	ImageNet	8×1×1	20.1M	26.8G×1×1	48.2	77.4	61.4	87.2
			8×3×2	20.1M	26.8G×3×2	49.8	79.0	63.5	88.8
			8×3×10	20.1M	26.8G×3×10	50.2	79.1	63.9	88.9
			16×1×1	20.1M	53.5G×1×1	50.4	79.4	63.6	88.6
			16×3×2	20.1M	53.5G×3×2	51.4	80.3	65.1	89.8
			16×3×10	20.1M	53.5G×3×10	51.9	80.6	65.1	**91.0**
$\text{STC}_{\text{Ensemble}}$	ResNet-50	ImageNet	(8+16)×3×2	—	80.5G×3×2	53.3	81.8	66.4	90.5
			(8+16)×3×10	—	80.5G×3×10	**53.7**	**82.0**	**66.9**	**91.0**
$\text{STC-Light}_{\text{Ensemble}}$	ResNet-50	ImageNet	(8+16)×3×2	—	80.3G×3×2	53.2	81.8	66.4	90.4
			(8+16)×3×10	—	80.3G×3×10	53.4	81.9	66.8	90.3

When equipping the module with the temporal shift operation, the resulted standard STC shows some performance increase, especially for the non-shift STC with $\alpha = 1$. This can be explained by the fact that the temporal shift operation enables the non-shift STC to model local temporal interactions. When setting $\alpha = 3/4$, the effect of temporal shift in S_T block is not as significant when with $\alpha = 1$. This may be because that the temporal convolution of the other T_S block is already capable of capturing local temporal interactions.

We also examine STC with a single refinement block (TR or SR). Table III lists the performance comparison. Firstly, we test STC with a single path, e.g., S_T ($\alpha = 1$) and T_S ($\alpha = 0$). It can be found that both TR and SR can significantly improve their network's performance, e.g., 12.3%→24.9% for T_S w. SR and 45.6%→47.3% for S_T w. TR. Then, we test the standard STC (i.e., with α=3/4) and observe consistent performance improvements with TR, SR and both, e.g., 46.9%→48.1% (+1.2%) for TR, 46.9%→47.8% (+0.9%) for SR, and 46.9%→48.3% (+1.4%) for both. Particularly, the top-1 accuracy gain with TR (+1.2%) is higher than that with SR (+0.9%). This may be because that the Something-Something dataset requires strong temporal modeling, temporal context obtained by TR can provide richer information than SR for video activity recognition. While, the combination of TR and SR, i.e., the standard STC, obtains the highest performance of 48.3%, which gives evidence that both temporal and spatial contexts can contribute to action recognition.

D. Comparison with State-of-the-Arts

Something-Something V1&V2. We report the performance comparisons, including the number of parameters, FLOPs and top-1/top-5 classification accuracies, between our STC variants and SOTAs on Something-Something V1&V2 datasets in Table IV. Overall, our proposed STC variants achieve better or comparable performance while requiring the lowest model complexities among the competing methods. Given 8 frames as input, the STC model has only 22.0M parameters and 26.9G FLOPs, which is more efficient than the standard 2D ResNet-50 (TSN) (23.9M parameters and 32.9G FLOPs) by a large margin. The STC-Light model even outperforms the current

efficient GST model on both the classification accuracy (48.2% vs 47.0%) and the complexity (20.1M vs 21.0M in parameters and 26.8G vs 29.5G in FLOPs) when using the same settings.

More specifically, on Something-Something V1, STC achieves 52.2% top-1 accuracy with 16-frames×3-crops×10-clips, which is better than most of the SOTAs, except TEA that obtain the highest performance of 52.3%. However, considering the model size (parameters) and computation cost (FLOPs), STC (22.0M and 53.6G×3×10 FLOPs) is comparable to TEA (24.5M and 70.0G×3×10 FLOPs). On Something-Something V2, STC achieves the highest top-1 performance of 65.3%, and the lightweight version STC-Light (65.1% top-1 accuracy) also outperforms all the competing SOTAs. Since action categories of the Something-Something datasets lie more on long-range temporal dependency, models that have the inherent ability of long-range modeling, such as the proposed STC variants, TAM and TEA, consistently perform well. In addition, we give the ensemble result of {8,16} frames STC models following the fusion strategy in [4]. As shown in the last few rows of Table IV, $\text{STC}_{\text{Ensemble}}$ achieves a high performance of 53.7%/82.0% top-1/top5 accuracy on Something-Something V1 and 66.9%/91.0% top-1/top5 accuracy on Something-Something V2.

TABLE V: Performance comparison on the updated Diving48 dataset using the train/validation split V2.

Method	Backbone	#Frame	Top-1	Top-5
TSN (our impl.)	ResNet-50	8	72.4	96.8
C3D (our impl.)	3DResNet-50	8	73.4	96.0
GST (our impl.)	ResNet-50	8	74.2	94.5
TSM (our impl.)	ResNet-50	8	77.6	**97.7**
TIN (our impl.)	ResNet-50	8	73.1	96.3
TEA (our impl.)	ResNet-50	8	76.5	96.9
STC	ResNet-50	8	**77.9**	97.4
STC-Light	ResNet-50	8	77.8	**97.7**

Diving48. Table V shows the performance comparison on Diving48 [48] dataset. Since this new version of the dataset has been thoroughly cleaned, we re-run all the competing methods by ourselves, including TSN, C3D, GST, TSM, TIN and TEA, for a fair comparison. All the results are obtained with 8 sample frames as input. Interestingly, the simple 2D TSN model also achieves a relatively good result. We believe that this is because the continuous movement changes can be recognized by the simple average combination of subtle body poses at different dive stages. Further capturing the temporal cues in the subtle body pose can benefit the diving action recognition, which is demonstrated by the performance improvement (72.4% → 77.6%) obtained by TSM. Moreover, by further considering the long-range relations across subtle poses, our STC and STC-Light outperform all the other counterparts with 77.9% and 77.8% precision. It is worth noting that there is a huge difference between the reported results here and the ones in [48] (only 10%-30% for TSN and C3D). This is because that the used dataset version is an currently updated one, where the dive annotations have been manually cleaned and the poorly segmented videos were removed.

EGTEA Gaze+ and EPIC-KITCHENS. The egocentric actions in the two datasets contain various interactions between human and objects occurring in the daily environment. Those interactions generally continue for a short term. This observation prefers the action recognition methods that can model the spatial (e.g., objects) and the temporal (e.g., quick motion) cues at the same time.

TABLE VI: Performance comparison on EGTEA Gaze+ dataset using train/validation split 1/2/3. Except R34-2stream using ResNet-34 as backbone, all the other models adopt ResNet-50 as backbones.

Method	#Frame	Split1	Split2	Split3
I3D-2stream [49]	24	55.8	53.1	53.6
R34-2stream [58]	25	62.2	61.5	58.6
TSN (our impl.)	8	61.6	58.5	55.2
C3D (our impl.)	8	62.1	59.2	57.0
GST (our impl.)	8	63.3	61.2	59.2
TSM (our impl.)	8	63.5	62.8	59.5
TIN (our impl.)	8	61.6	61.6	57.1
TEA (our impl.)	8	**65.5**	**64.8**	**62.4**
STC	8	64.6	63.9	60.5
STC-Light	8	64.1	63.7	60.9

TABLE VII: Performance comparison on EPIC-KITCHENS-55 dataset. The results of all the methods are obtained using our train/validating split.

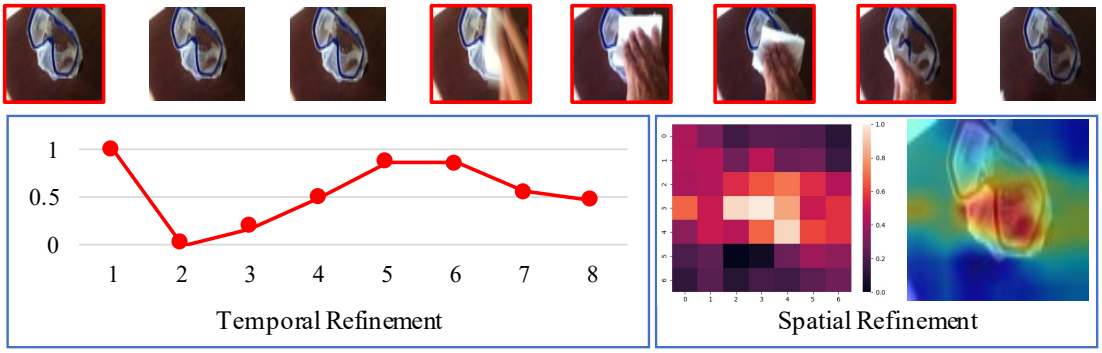
Method	Backbone	#Frame	Verb	Noun
TSN (our impl.)	ResNet-50	8	37.4	**23.1**
C3D (our impl.)	3DResNet-50	8	45.2	21.5
GST (our impl.)	ResNet-50	8	46.4	21.1
TSM (our impl.)	ResNet-50	8	48.2	22.9
TIN (our impl.)	ResNet-50	8	47.6	22.8
TEA (our impl.)	ResNet-50	8	**50.5**	21.7
STC	ResNet-50	8	48.7	22.8
STC-Light	ResNet-50	8	48.5	22.6

Table VI shows the performance comparison on the EGTEA Gaze+ using three official training/validation splits. It provides evidence for the above claim, that is, the sptaio-temporal modeling networks, *i.e.*, C3D, GST, TSM, TIN, TEA and STCs, obtain better performance than the space-only TSN. Also, the attention based methods TEA and STCs can further improve the performance, which show evidence that feature contextualization is important for action recognition.

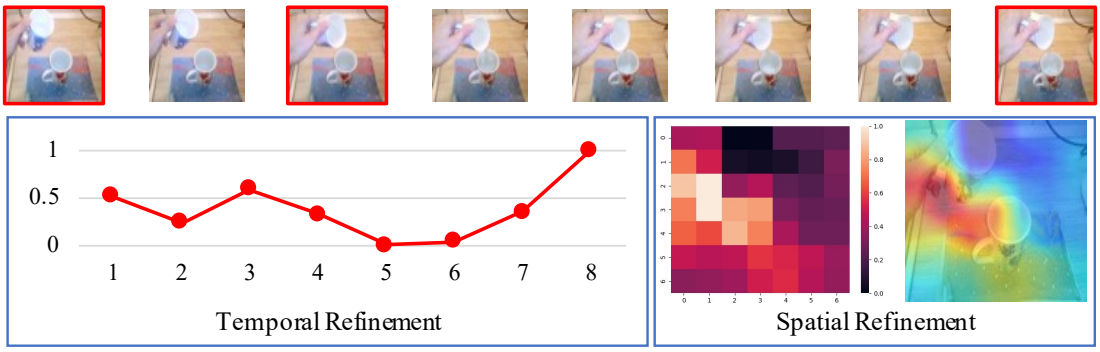
Different to the task on EGTEA Gaze+ dataset, models are required to separately recognize the motion ingredient (*i.e.*, verb) and object ingredient (*i.e.*, noun) of the action on the EPIC-KITCHENS dataset. As shown in Table VII, for verb recognition, all the temporal models outperform the 2D TSN, and our STC and STC-Light again obtain second better results. For noun recognition, the 2D TSN perform best among these methods, this may be because the objects need more spatial modeling rather than temporal modeling. Also, our STC variants obtain comparable results.

E. Visualization

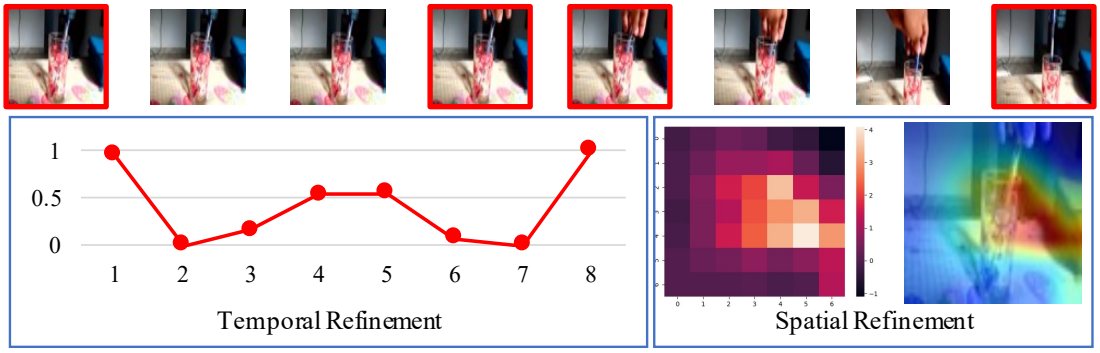
To analyze the effectiveness of the feature refinement blocks, we report the temporal refinement scores (S_T) and visualize the spatial refinement heatmap (T_S) with four examples of Something-Something V1 validation set in Fig. 8. For example in Fig. 8(a), given the clip with the label "Wiping something off of something", we first show the flattened frames in the top row. Then, we extract feature refinement scores of the last layer of STC model. Specifically, (1) the temporal refinement



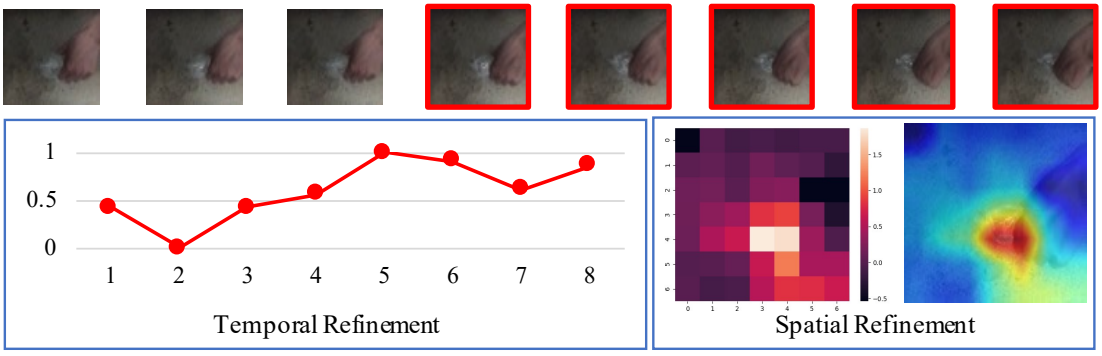
(a) Wiping something off of something



(b) Pouring something into something until it overflows



(c) Taking something from somewhere



(d) Spilling something onto something

Fig. 8: The visualization of the feature refinement results with 8 frames input on Something-Something V1 validation set.

(bottom left) provides the gating weights of all frames and marks the important moments (the frames in the red box). (2) the spatial refinement (bottom right) shows the heatmap and the responses of the frames. As expected, in the four examples the temporal attention and spatial attention success in highlight key timestamps and core regions, respectively.

V. Conclusion

In this paper, we have presented a new spatio-temporal collaborative (STC) module to tackle the problem of efficient action recognition, which consists of two lightweight collaborative computational units, *i.e.*, S_T and T_S, as well as their lightweight versions, particularly, exhibit effectiveness on both local spatio-temporal pattern and long-range dependency modeling. Through channel splitting and building upon the basis 2D ResNet structure, the resulted STC variants effectively decrease the computation cost. The extensive experiments have shown that our model achieves state-of-the-art results on five popular benchmarks and also demonstrate its robustness against both short- and long-term actions.

References

[1] 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*, 2015, pp. 4489–4497.

[2] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[3] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann, "Devnet: A deep event network for multimedia event detection and evidence recounting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2568–2577.

[4] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 7083–7093.

[5] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5533–5541.

[6] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[7] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 305–321.

[8] J. Lin, C. Gan, and S. Han, "Temporal shift module for efficient video understanding," *arXiv preprint arXiv:1811.08383*, 2018.

[9] L. Fan, S. Buch, G. Wang, R. Cao, Y. Zhu, J. C. Niebles, and L. Fei-Fei, "Rubiksnet: Learnable 3d-shift for efficient video action recognition," in *European Conference on Computer Vision*. Springer, 2020, pp. 505–521.

[10] H. Wang, D. Tran, L. Torresani, and M. Feiszli, "Video modeling with correlation networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 352–361.

[11] C. Luo and A. L. Yuille, "Grouped spatial-temporal aggregation for efficient action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5512–5521.

[12] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7794–7803.

[13] Y. Li, B. Ji, X. Shi, J. Zhang, B. Kang, and L. Wang, "Tea: Temporal excitation and aggregation for action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 909–918.

[14] Z. Liu, L. Wang, W. Wu, C. Qian, and T. Lu, "Tam: Temporal adaptive module for video recognition," *arXiv preprint arXiv:2005.06803*, 2020.

[15] R. Goyal, S. E. Kahou, V. Michalski, J. Materzynska, S. Westphal, H. Kim, V. Haenel, I. Fruend, P. Yianilos, M. Mueller-Freitag *et al.*, "The" something something" video database for learning and evaluating visual common sense." in *ICCV*, vol. 2, 2017, p. 8.

[16] M. Zolfaghari, K. Singh, and T. Brox, "Eco: Efficient convolutional network for online video understanding," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 695–712.

[17] H. Shao, S. Qian, and Y. Liu, "Temporal interlacing network," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 966–11 973.

[18] Z. Liu, D. Luo, Y. Wang, L. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and T. Lu, "Teinet: Towards an efficient architecture for video recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 11 669–11 676.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

[20] 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*. Springer, 2016, pp. 20–36.

[21] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould, "Dynamic image networks for action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3034–3042.

[22] B. Zhou, A. Andonian, A. Oliva, and A. Torralba, "Temporal relational reasoning in videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 803–818.

[23] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *European Conference on Computer Vision*. Springer, 2016, pp. 510–526.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] Z. Tu, H. Li, D. Zhang, J. Dauwels, B. Li, and J. Yuan, "Action-stage emphasized spatiotemporal vlad for video action recognition," *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2799–2812, 2019.

[27] Z. Xu, Y. Yang, and A. G. Hauptmann, "A discriminative cnn video representation for event detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1798–1807.

[28] X. Long, C. Gan, G. De Melo, J. Wu, X. Liu, and S. Wen, "Attention clusters: Purely attention based local feature integration for video classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7834–7843.

[29] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "Spatio-temporal attention-based lstm networks for 3d action recognition and detection," *IEEE Transactions on image processing*, vol. 27, no. 7, pp. 3459–3471, 2018.

[30] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4694–4702.

[31] H. Sakaino, "Spatio-temporal feature extraction/recognition in videos based on energy optimization," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3395–3407, 2019.

[32] L. Fan, W. Huang, C. Gan, S. Ermon, B. Gong, and J. Huang, "End-to-end learning of motion representation for video understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[33] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *Joint pattern recognition symposium*. Springer, 2007, pp. 214–223.

[34] S. Zhang, S. Guo, W. Huang, M. R. Scott, and L. Wang, "V4d: 4d convolutional neural networks for video-level representation learning," *arXiv preprint arXiv:2002.07442*, 2020.

[35] X. Chen, J. Weng, W. Lu, J. Xu, and J. Weng, "Deep manifold learning combined with convolutional neural networks for action recognition," *IEEE transactions on neural networks and learning systems*, vol. 29, no. 9, pp. 3938–3952, 2018.

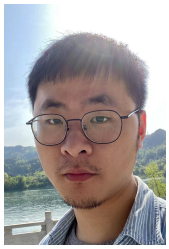
[36] Z. Gao, L. Guo, T. Ren, A.-A. Liu, Z.-Y. Cheng, and S. Chen, "Pairwise two-stream convnets for cross-domain action recognition with small data," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[37] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *The*

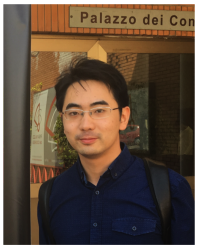
IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
[38] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1492–1500.
[39] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," *arXiv preprint arXiv:1812.03982*, 2018.
[40] C. Feichtenhofer, "X3d: Expanding architectures for efficient video recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 203–213.
[41] C. Li, Q. Zhong, D. Xie, and S. Pu, "Collaborative spatiotemporal feature learning for video action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
[42] Q. Fan, C.-F. Chen, H. Kuehne, M. Pistoia, and D. Cox, "More is less: Learning efficient video representations by big-little network and depthwise temporal aggregation," *arXiv preprint arXiv:1912.00869*, 2019.
[43] S. Sudhakaran, S. Escalera, and O. Lanz, "Gate-shift networks for video action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1102–1111.
[44] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.
[45] Z. Zheng, G. An, D. Wu, and Q. Ruan, "Global and local knowledge-aware attention network for action recognition," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 1, pp. 334–347, 2020.
[46] H. Yang, C. Yuan, L. Zhang, Y. Sun, W. Hu, and S. J. Maybank, "Sta-cnn: convolutional spatial-temporal attention learning for action recognition," *IEEE Transactions on Image Processing*, vol. 29, pp. 5783–5793, 2020.
[47] F. Mahdisoltani, G. Berger, W. Gharbieh, D. Fleet, and R. Memisevic, "On the effectiveness of task granularity for transfer learning," *arXiv preprint arXiv:1804.09235*, 2018.
[48] Y. Li, Y. Li, and N. Vasconcelos, "Resound: Towards action recognition without representation bias," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 513–528.
[49] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *The European Conference on Computer Vision (ECCV)*, September 2018.
[50] D. Damen, H. Doughty, G. Maria Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price *et al.*, "Scaling egocentric vision: The epic-kitchens dataset," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 720–736.
[51] F. Baradel, N. Neverova, C. Wolf, J. Mille, and G. Mori, "Object level visual reasoning in videos," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 105–121.
[52] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.
[53] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 399–417.
[54] C. Yang, Y. Xu, J. Shi, B. Dai, and B. Zhou, "Temporal pyramid network for action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 591–600.
[55] X. Li, Y. Wang, Z. Zhou, and Y. Qiao, "Smallbignet: Integrating core and contextual views for video classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1092–1101.
[56] B. Jiang, M. Wang, W. Gan, W. Wu, and J. Yan, "Stm: Spatiotemporal and motion encoding for action recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2000–2009.
[57] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 510–519.
[58] S. Sudhakaran and O. Lanz, "Attention is all we need: Nailing down object-centric attention for egocentric activity recognition," in *BMVC*, 2018.

Yanbin Hao is a research associate professor at the University of Science and Technology of China (USTC), China. He received the B.E. and Ph.D. degrees from the Hefei University of Technology, Hefei, China, in 2012 and 2017, respectively. During his Ph.D., he was also a Visiting student (2015-2017) in Department of Electrical Engineering and Electronics, University of Liverpool, UK. During 2018-2020, he joined the research group of Prof. Chong-Wah Ngo in the Department of Computer Science, City University of Hong Kong (CityU), as a Postdoc Fellow. His research interests mainly include machine learning and multimedia data analysis, such as large-scale multimedia indexing and retrieval, multimedia data embedding, and video understanding. Dr. Hao has served as the PC member for top-tier conferences such as ACMMM, CVPR, ICCV, IJCAI, and AAAI.

Shuo Wang received the B.E. and Ph.D. degrees from the Hefei University of Technology (HFUT), Hefei, China, in 2015 and 2020, respectively. He is currently a Postdoctoral researcher, School of Data Science, University of Science and Technology of China (USTC), China. His reseach interests mainly include machine learning and multimedia data analysis, such as large-scale multimedia indexing and retrieval, multimedia data embedding, and few shot learning.



Yi Tan is a ph.d. candidate at the University of Science and Technology of China, supervised by professor Xiangnan He. He got his bachelor's degree in USTC in 2018. His research interests include multi-media understanding and representation.



Xiangnan He is a professor at the University of Science and Technology of China (USTC), leading the USTC Lab for Data Science. His research interests span information retrieval, data mining, machine learning and multi-media analytics. He is in the Editorial Board of journals including TOIS and AI Open, and has over 100 publications that appeared in several top conferences such as SIGIR, WWW, KDD, and MM, and journals including TKDE and TOIS. He has received the Best Paper Award Honorable Mention in SIGIR 2021, 2016 and WWW 2018. Moreover, he has served as the PC chair of CCIS 2019, and the (senior) PC member for several top conferences including SIGIR, WWW, KDD, IJCAI etc.

Zhenguang liu is currently a research fellow in Zhejiang University. He had been a research fellow in National University of Singapore and A*STAR (Agency for Science, Technology and Research, Singapore) for several years. He respectively received his Ph.D. and B.E. degrees from Zhejiang University and Shandong University, China. His research interests include multimedia data analysis and smart contract security. Various parts of his work have been published in top-tier venues including CVPR, ICCV, TKDE, TIP, AAAI, ACM MM, INFOCOM, IJCAI. Dr. Liu has served as technical program committee member for top-tier conferences such as ACM MM, CVPR, AAAI, IJCAI, and ICCV, session chair of ICGIP, local chair of KSEM, and reviewer for top-tier journals IEEE TVCG, IEEE TPDS, IEEE TMM, ACM TOMM, etc.

Meng Wang is a professor at Hefei University of Technology, China. He received the B.E. degree and Ph.D. degree in the Special Class for the Gifted Young and the Department of Electronic Engineering and Information Science from the University of Science and Technology of China (USTC), Hefei, China, in 2003 and 2008, respectively.He worked as an associate researcher at Microsoft Research Asia and a senior research fellow at National University of Singapore. His current research interests include multimedia content analysis, computer vision, and pattern recognition. He has authored or co-authored over 200 book chapters, journal and conference papers. He holds over 30 US, Chinese, and international granted patents. He received paper prizes or awards from ACM MM 2009 (Best Paper Award), ACM MM 2010 (Best Paper Award), MMM 2010 (Best Paper Award), ICIMCS 2012 (Best Paper Award), ACM MM 2012 (Best Demo Award), ICDM 2014 (Best Student Paper Award), SIGIR 2015 (Best Paper Honorable Mention), IEEE TMM 2015 and 2016 (Prize Paper Award Honorable Mention), IEEE SMC 2017 (Best Transactions Paper Award), and ACM TOMM 2018 (Nicolas D. Georganas Best Paper Award). He is or has been an editorial board member of IEEE Trans. on Circuits and Systems for Video Technology, IEEE Trans. on Multimedia, IEEE Trans. on Knowledge and Data Engineering, IEEE Trans. on Neural Networks and Learning Systems, etc. He is the General Co-Chair of ICMR 2021, PCM 2018 and MMM 2013, and the Program Co-Chair of ICIMCS 2013.