Selective Volume Mixup for Video Action Recognition

Yi Tan, Zhaofan Qiu, Yanbin Hao, *Member, IEEE,* Ting Yao, *Member, IEEE,* Xiangnan He, *Senior Member, IEEE,* and Tao Mei, *Fellow, IEEE*

***Abstract*—The recent advances in Convolutional Neural Networks (CNNs) and Vision Transformers have convincingly demonstrated high learning capability for video action recognition on large datasets. Nevertheless, deep models often suffer from the overfitting effect on small-scale datasets with a limited number of training videos. A common solution is to exploit the existing image augmentation strategies for each frame individually including Mixup, Cutmix, and RandAugment, which are not particularly optimized for video data. In this paper, we propose a novel video augmentation strategy named Selective Volume Mixup (SV-Mix) to improve the generalization ability of deep models with limited training videos. SV-Mix devises a learnable selective module to choose the most informative volumes from two videos and mixes the volumes up to achieve a new training video. Technically, we propose two new modules, i.e., a spatial selective module to select the local patches for each spatial position, and a temporal selective module to mix the entire frames for each timestamp and maintain the spatial pattern. At each time, we randomly choose one of the two modules to expand the diversity of training samples. The selective modules are jointly optimized with the video action recognition framework to find the optimal augmentation strategy. We empirically demonstrate the merits of the SV-Mix augmentation on a wide range of video action recognition benchmarks and consistently boot the performances of both CNN-based and transformer-based models.**

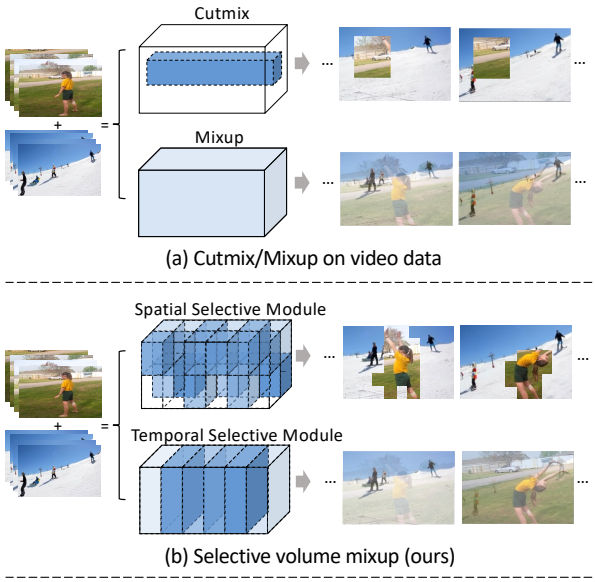
***Index Terms*—video action recognition, neural networks, data augmentation.**

I. Introduction

DEEP models, including CNN and transformer-based architectures, have successfully proven highly effective for understanding multimedia content on large-scale datasets. To date in the literature, there are various large models that push the limits of multimedia analysis systems, e.g., Vision Transformer [1], Swin Transformer [2], ConvNeXt [3] for image classification, MViT [4], Video Swin [5], Uniformer [6] for video analysis, CLIP [7], GLIP [8] for cross-modality understanding. Nevertheless, the impressive performances of these models highly rely on large-scale datasets and are easily affected by the overfitting effect on the tasks with insufficient training data. Such an issue becomes even worse particularly for video action recognition due to the difficulty of achieving large amounts of video data and expensive efforts for labeling.

Y. Tan, Y. Hao and X. He are with the School of Information Science and Technology, University of Science and Technology of China, Anhui, 230009, China. E-mail: ty133@mail.ustc.edu.cn, haoyanbin@hotmail.com, xiangnanhe@gmail.com.

Z. Qiu, T. Yao and T. Mei are with HiDream.ai, Beijing, 100000, China. E-mail: zhaofanqiu@gmail.com ,tingyao.ustc@gmail.com, tmei@live.com.



Method	Ucf101		Sth-Sth V1	
	Acc. (%)	ΔAcc. (%)	Acc. (%)	ΔAcc. (%)
TSM	85.2	-	45.5	-
+Cutmix	86.9	+1.7	45.7	+0.2
+Mixup	84.7	-0.5	44.6	-0.9
+SV-Mix	**88.4**	**+3.2**	**47.2**	**+1.7**

(c) Performance comparsion

Fig. 1. The intuition of (a) the typical Cutmix [9] and Mixup [10] augmentations on video data, and (b) our Selective Volume Mixup (SV-Mix). The typical methods randomly combine regions or entire frames from two videos and may lose crucial information. In contrast, our SV-Mix contains learnable selective modules to adaptively select valuable volumes. A tapas performance comparison between Cutmix/Mixup and our SV-Mix is also shown in (c).

To alleviate this issue, a general practice is to exploit network regularization and data augmentation to preserve the effectiveness of large models with limited training data. Among these strategies, the network regularization methods, including dropout [11], drop path [12], and weight decay [13], are general network training schemes across different tasks and can be directly utilized for video models. However, the policy of data augmentation should be specially designed for different data formats since it is highly related to the intrinsic properties of the input modality. The current standard data augmentation strategy for video data (e.g., in [4], [5], [14]) is to simply perform the existing image augmentation to each frame individually, as illustrated in Figure 1(a). This solution is straightforward but ignores the properties of video data, e.g.,

the temporal correlation across frames, and thus weakens the effectiveness of data augmentation. Moreover, these strategies are all manually devised and not learnable for different architectures/datasets, which requires more significant engineering effort of human experts to tune the hyper-parameters given a new architecture/dataset.

In this work, we aim at investigating a learnable data augmentation mechanism to facilitate the data efficiency for video action recognition. We start from the basic idea of the popular image augmentations, i.e., Mixup [10] and Cutmix [9], that combines the content of two videos to obtain a new training video. Mixup blends the two training images by weighted summation, and Cutmix randomly exchanges a local region of two samples. These two manually designed strategies consistently show good performances in image models but are not acclimatized to the video domain. We speculate that the difficulty of video Mixup/Cutmix mainly originates from two aspects: 1) Video is an information-intensive media, and the labeled actions are related to objects, interaction, scene, etc. Randomly exchanging a local region (like in Cutmix) may lose some crucial information. 2) Video usually contains several background frames that do not contain the labeled action. Blending the background frame to the other video (like in Mixup) may dilute the useful cues of the original video. That motivates us to devise a selective module to preserve the informative volumes in the mixing process.

To this end, we present a new Selective Volume Mixup (SV-Mix), as shown in Figure 1(b). Specifically, SV-Mix contains two modules, i.e., a spatial selective module and a temporal selective module. The former builds cross attention between the local patches with the same timestamp from two videos, respectively, to determine the preserved patches in each spatial position. The latter utilizes a similar attention mechanism but on the frame level to blend the most informative frames while keeping the spatial structure. The two selective modules are complementary to each other that select the mixed volumes on the patch level and frame level, respectively. Hence, we stochastically choose one of the two modules at each time to expand the divergence of the augmented samples. SV-Mix is jointly optimized with the action recognition framework in an end-to-end manner. Moreover, to avoid the bad influence between the augmentation network and the action recognition network before convergence, we devise a disentangled training pipeline, which exploits a slow-moving average of action recognition parameters instead of the training one to guide the optimization of SV-Mix. As a result, the gradients of both components are disentangled and lead to the convergence of both optimal data augmentation and optimal action recognition framework.

To the best of our knowledge, our work is the first to devise a learnable data augmentation strategy for video data. The design also leads to the elegant view of how to adaptively mix two videos while maintaining valuable information. We uniquely formulate the problem as cross attention between the volumes of two videos and devise two selection modules by mixing along the spatial dimension and temporal dimension, respectively. Extensive experiments on five datasets demonstrate the effectiveness of our proposal, and with different action recognition frameworks including both CNN-based and transformer-based methods, our SV-Mix consistently improves the performances over other augmentation strategies.

II. Related work

We briefly group the related works into two categories: video action recognition and data augmentation strategies.

Video action recognition. With the prevalence of deep learning in multimedia analysis, the dominant paradigm in modern video action recognition is deep neural networks. The research of deep models for video action recognition has proceeded along three dimensions: 2D CNNs, 3D CNNs and video transformers. 2D CNNs [15]–[19] often treat a video as a sequence of frames or optical flow images, and directly extend the 2D CNNs from the image domain for frame-level recognition. For instance, the famous two-stream networks proposed in [15] apply two 2D CNNs separately on visual frames and stacked optical flow images. Later, Wang *et al.* [20] propose Temporal Segment Networks, which divide input video into several segments and sample one frame/optical flow image from each segment as the input of two-stream networks. The two-stream architecture is further extended by advanced fusion strategies [16], [17], and feature encoding mechanism [18], [19].

The above 2D CNNs proceed each frame individually at early layers, and the pixel-level temporal evolution across consecutive frames is seldom explored. To alleviate this issue, 3D CNNs [14], [21]–[32] are devised to directly learn spatio-temporal correlation from video clips via 3D convolution. A prototype of 3D CNNs is introduced in [21] by replacing 2D convolution in 2D CNNs with 3D convolution. A widely adopted 3D CNN, called C3D [22], is devised by expanding VGG-style 2D CNN to 3D manner with both 3D convolutions and 3D poolings. To reduce the expensive computations and the model size of 3D CNNs, the fully 3D convolution is decomposed into a spatial convolution plus a temporal convolution [23]–[25] or a depth-wise convolution plus a point-wise convolution [26], [27]. Another scheme to improve 3D CNNs is to expand the temporal receptive field. Varol *et al.* present LTC architecture [28] that increases the length of input clips while reducing the resolution of the input frame. Furthermore, 3D convolution on different time scales [14], [29], [30] and holistic view of video [31], [32] are also proven to be effective on long-term modeling.

More recently, video transformers [5], [6], [33]–[39] become formidable competitors to 3D CNNs. The early works for video transformers, i.e., TimesSformer [33] and ViViT [34] study the basic designs of video transformer including tubelet embedding and attention decomposition. MViT [35] and Video Swin [5] follow the philosophy of CNNs, where the channel dimension increases while the spatial resolution shrinks with the layer going deeper, to reduce the computational cost. More fine-grained designs are proposed recently to improve video transformers, including Multiview Transformer [36], cross-frame attention [37], recurrent attention [38], trajectory attention [39], and combining 3D convolutions [6].

Data augmentation strategies. The data augmentation strategy, as an important facility to alleviate the overfitting

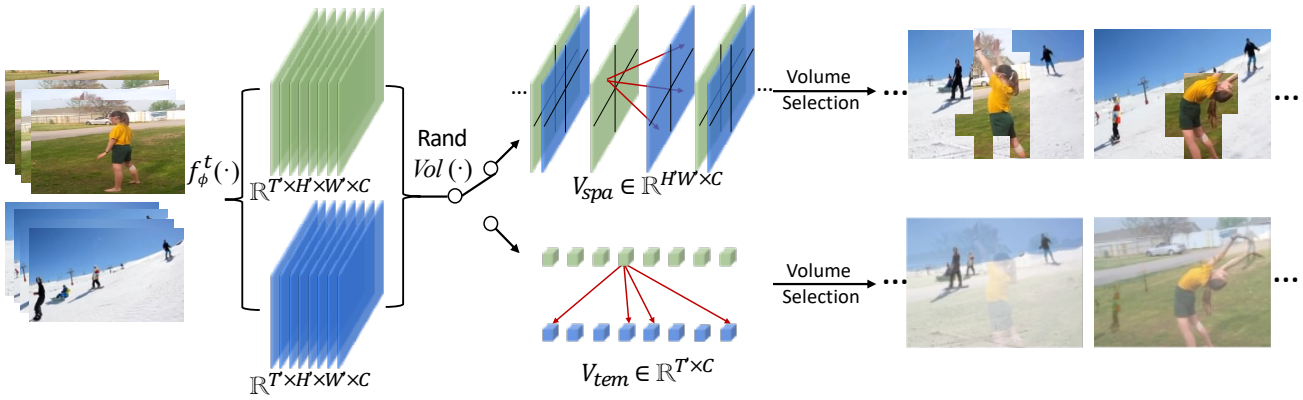


Fig. 2. The overview of our proposed Selective Volume Mixup (SV-Mix) data augmentation. Given two training videos, we first extract the volume-level feature map of each video by an encoder $f_\phi^t(\cdot)$. Next, we randomly choose to select the target volumes either on patch level (spatial selective module) or on the frame level (temporal selective module). The selected volumes are then combined together to achieve an augmented video as the input of the subsequent action recognition framework.

effect, has attracted intensive research interests in recent years. Take the augmentation strategies in the image domain as an example, random erasing [40] randomly choose a local region of an image and erase the content inside the bounding box. Autoaugment [41] formulates the augmentation process as the combination of sequential augmentation operations and proposes a search-based algorithm to tune the strength of each operation. Similarly, RandAugment [42] is also based on a group of augmentation operations but replaces the search process in auto augmentation by randomly choosing operations. Unlike the mentioned augmentations that are utilized on each sample individually, Mixup [10] proposes to blend two samples by random weights and perform multi-label classification on the mixed image. Similarly, Cutmix [9] devises a strategy to exchange the pixels in a local region of two images to be mixed. Moreover, the such idea of mixing two images is improved by the advanced techniques including TransMix [43] and Automix [44].

In summary, our work aims to devise a learnable data augmentation strategy for video action recognition. The most closely related works are VideoMix [45] and DynaAugment [46]. They remould the existing Cutmix and RandAugment to the video domain, respectively. Ours is different in that the proposed SV-Mix contains two learnable selective modules to determine the mixed video, which are jointly optimized with the action recognition framework. Moreover, the selective modules can be optimized adaptively during training when using different action recognition models or on different datasets.

III. METHODOLOGY

In this section, we deliberate our proposed Selective Volume Mixup (SV-Mix) for video action recognition. First, we briefly summarize the preliminaries of video model training using mixed video samples. Then, we detail the architecture of volume selection, and show how to utilize this architecture to construct spatial selective module and temporal selective module. Finally, a novel disentangled training pipeline is proposed to jointly optimize the volume selection modules and the action recognition framework. Figure 2 illustrates the overview of our proposed SV-Mix.

A. Preliminaries

Given a video sample $x \in \mathbb{R}^{T\times H\times W\times 3}$ which contains T frames with size $H \times W$ and 3 channels, the goal of the video action recognition model is to inference its one-hot class label $y \in \mathbf{Y} = \{0,1\}^K$, where K denotes the number of categories. In the action recognition pipeline with mixed video samples, the input video data and the corresponding label are rearranged as the linear interpolation of two or more videos, and in following statement, we focus on video mixing using two samples for conciseness. Particularly, given two video-label pairs (x_i, y_i) and (x_j, y_j), the mixed process of samples and labels can be represented as:

$$\widetilde{x} = \mathbf{M} \odot x_i + (1 - \mathbf{M}) \odot x_j, \tag{1}$$

$$\widetilde{y} = \lambda y_i + (1 - \lambda) y_j, \tag{2}$$

where $\odot$ denote hadamard product and $\mathbf{M} \in [0,1]^{T\times H\times W}$ is the mixing weights. Each element $\mathbf{M}_{t,w,h}$ represents the mix proportion of a specific pixel and λ is the label proportion calculated from $\mathbf{M}$. The goal of video model training is to find a group of parameters ϕ of deep neural network f_ϕ which minimize the following loss function:

$$\hat{\phi} = \arg\min_{\phi} \mathcal{L}(\phi), \quad \mathcal{L}(\phi) = \mathcal{L}_{sce}(f_\phi(\widetilde{x}), \widetilde{y}), \tag{3}$$

where $\mathcal{L}_{sce}$ denotes the soft-targat cross entropy loss.

B. Selective Volume Mixup

In the traditional mixing process, the mixing weights $\mathbf{M}$ are usually randomly sampled from a manually designed principle, e.g., the frame-level random weights in Mixup, and the random rectangle with value one in Cutmix. These strategies are not learnable and ignore the content of input videos. In contrast, our goal is to parameterize the generation process of mixing weights $\mathbf{M}$. In the other words, we try to devise another neural network to predict the probability of each volume in the mixed video coming from x_i.

We achieve this goal by answering two core design questions: 1) How to model the relationship between volumes from two input videos to obtain a most informative mixed

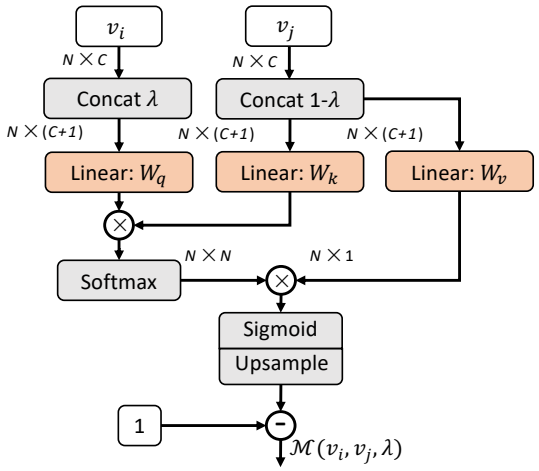


Fig. 3. A diagram of the volume selection module in our SV-Mix. Given the volumes from two videos $v_i, v_j \in \mathbb{R}^{N\times C}$ where N and C are the number of volumes and channels, respectively, the attention weights across two videos are calculated through three trainable linear mappings W_q, W_k, W_v. Here λ denotes the label proportion of the first video.

video; 2) With the additional temporal dimension, how to avoid the intensive full spatio-temporal dependency modeling and further leverage this property of video data for a modality-specific sample mixing; For the first question, we propose an attention-based block, which treats each volume in x_i as a query and the volumes in x_j as keys and values to evaluate if this volume should be maintained in the mixed video. For the second one, we propose to decompose the full spatio-temporal relation, and at each time, only calculate the attention along the spatial dimension or temporal dimension probabilistically. Toward these, we first reformulate the generation of the mixed sample as:

$$\begin{aligned}\mathcal{G}_\theta(x_i, x_j, \lambda) = \mathcal{M}_\theta(Vol(x_i), Vol(x_j), \lambda) \odot x_i + \\ (1 - \mathcal{M}_\theta(Vol(x_i), Vol(x_j), \lambda)) \odot x_j,\end{aligned} \tag{4}$$

in which θ parameterised volume selection function $\mathcal{M}_\theta(\cdot)$ and the volume partitioning function $Vol(\cdot)$ implements the attention decomposition. We then detail these two components one by one.

Volume selection. Given video volume $v_i = Vol(x_i)$ and $v_j = Vol(x_j) \in \mathbb{R}^{N\times C}$, where N and C represent the number of volumes and channels under a specific $Vol(\cdot)$ function, we devise a cross-attention mechanism to model the relations between v_i and v_j, as shown in Figure 3. The attention response naturally serves as an importance measurement of each element, and thus we can easily transfer it to volume selection. Technically, the similarity between volumes is formulated as:

$$S(v_i, v_j, \lambda) = Softmax\left(\frac{(W_q v_{i,\lambda})^T \otimes (W_k v_{j,1-\lambda})}{\sqrt{d_k}}\right), \tag{5}$$

where W_q and W_k are learnable matrices for projecting volumes into queries and keys respectively. d_k is the dimension of queries and keys, and $v_{i,\lambda}$ and $v_{j,1-\lambda}$ denote λ embeded volume representations achieved by concatenating λ or $1-\lambda$ on the channel dimension. By gathering the similarities between volumes from different videos, the attention response is calculated as the summation of values using the similarities as weights:

$$\mathcal{M}(v_i, v_j, \lambda) = 1 - Upsample\left(\delta\left(S(v_i, v_j, \lambda) \otimes W_v v_{j,1-\lambda}\right)\right), \tag{6}$$

where $W_v \in \mathbb{R}^{1\times C}$ is projection matrix for values. $\delta(\cdot)$ is the sigmoid operation to normalize the responds into $[0, 1]$ as a proportion value. $Upsample(\cdot)$ infers the full spatio-temporal mixing weights from the N sampled volumes. Please note that, here we utilize the inverse of attention responses as the mixing weights, since the higher attention responses usually indicate higher similarity with the other video but we prefer to preserve the most distinctive volumes.

Volume partition. Taking inspiration from the concept of spatio-temporal decomposition [23], [25], we employ a divide-and-conquer approach to partition the video into volumes from spatial or temporal perspective. This allows us to perform volume selection along a single dimension, thereby improving modeling efficiency and more importantly, enhancing the distribute diversity of mixed video samples. Notably, since the attention computation is shape agnostic for input data, individual spatial selection and temporal selection share the same parameters but partitioning the volumes along different dimensions.

Given the video feature $Z \in \mathbb{R}^{B\times T'\times H'\times W'\times C}$ encoded by the backbone $f^t_\phi(\cdot)$, we reshape Z and achieve volumes as $V^{spa} \in \mathbb{R}^{BT'\times H'W'\times C}$ for spatial selection. By moving the temporal dimension to the batch dimension, the attention is calculated along the spatial dimension for each timestamp individually.

Similarly, for the temporal selective module, we shrink the spatial dimension to gather volumes $V^{tem} \in \mathbb{R}^{B\times T'\times C}$

$$V_{tem} = \frac{1}{H' \times W'} \sum_{i}^{H'} \sum_{j}^{W'} Z_{i,j}. \tag{7}$$

With V^{tem} as the input, the temporal selective module captures the relationship between frames instead of patches, and then assigns the same weight for all patches in the identical frame. Through temporal selection, we assign relative importance for each frame in the mixed video while maintaining the spatial pattern.

A common strategy to ensemble the spatial selective module and temporal selective module is to simply average the mixing weight from the two modules as

$$\mathcal{M}_{en} = \frac{1}{2}\left(\mathcal{M}_\theta(v_i^{tem}, v_j^{tem}, \lambda) + \mathcal{M}_\theta(v_i^{spa}, v_j^{spa}, \lambda)\right). \tag{8}$$

However, this strategy produces mixed training samples with only a single style, which limits the diversity [47] of data augmentation in model training. Under this consideration, we propose to probabilistically ensemble spatial selective module and temporal selective module by randomly choosing one of the two modules at each time:

$$\mathcal{M}_{en} = \begin{cases}\mathcal{M}_\theta(v_i^{tem}, v_j^{tem}, \lambda), & \mu \le P \\ \mathcal{M}_\theta(v_i^{spa}, v_j^{spa}, \lambda), & \mu > P\end{cases} \tag{9}$$

where $\mu \sim U(0, 1)$ is sampled from a uniform sampling, and P is the switch probability between two modules. Here,

we simply set $P = 0.5$ to demonstrate the effectiveness of prebabilistically ensemble.

Disentangled training pipeline. By parameterising the sample mixing process, the forward propagation of the action recognition model changes from $f_\phi(x)$ to $f_\phi(\mathcal{G}_\theta(v_i, v_j, \lambda))$. An intuitive way to jointly optimize the selective modules θ and action recognition framework ϕ is to directly conduct end-to-end learning (we refer it to entangled training in the following parts) under the supervision of mixed label $\widetilde{y} = \lambda y_i + (1-\lambda y_j)$. However, a gradient entanglement occurs in the optimization process of volume selective module θ:

$$\frac{\delta \mathcal{L}_{sce}}{\delta \theta} \propto \frac{\delta f_\phi(\mathcal{G}_\theta(v_i, v_j, \lambda))}{\delta \mathcal{G}_\theta(v_i, v_j, \lambda)} \cdot \frac{\delta \mathcal{G}_\theta(v_i, v_j, \lambda)}{\delta \theta} \tag{10}$$

During training, both the parameters ϕ and θ undergo rapid changes and then the gradient of θ may be corrupted by $\frac{\delta f_\phi(\mathcal{G}_\theta(v_i, v_j, \lambda))}{\delta \mathcal{G}_\theta(v_i, v_j, \lambda)}$, which may lead to sub-optimal state for $\mathcal{M}_\theta$ and $\mathcal{G}_\theta$, and in turn results in sub-optimal classification performance for f_ϕ. To solve this problem, inspired by the disentanglement trick in self-supervised learning, e.g. BYOL [48], which reveals the stability of the slow-moving average (momentum update) of current training model and further utilizes it as the teacher network to guide the optimization of student network. We follow this basic idea and disentangle the optimization process of volume selective module and video action recognizer. In the other words, we wish each module concentrates on its own task, i.e., action recognition and mixed video generation, respectively. Particularly, we refer to the slow-moving average of current action recognizer as the teacher f_ϕ^t to guide the training of SV-Mix modules. Thus, the optimization process in one iteration is as demonstrated in Figure 4 (a): 1) frozen f_ϕ^t encodes two video samples x_i and x_j into semantic space z_i and z_j; 2) randomly choose a volume partition strategy, e.g. V^{spa} or V^{tem}, and generate two mixed video sample $\widetilde{x}_s$ and $\widetilde{x}_t$ for f_ϕ^s and f_ϕ^t, respectively, to predict action categories; 3) update ϕ_s and θ; 4) update ϕ_t via exponential moving average (EMA): $\phi_t \leftarrow m\phi_t + (1-m)\phi_s$, where $m \in [0, 1)$ is the momentum coefficient.

In addition, we introduce an extra tributary loss for leading the mixing weight $\mathcal{M}_\theta$ to match the predefined λ:

$$\mathcal{L}_\mathcal{M} = \|\lambda - \frac{1}{T \times H \times W} \sum_{i,j,k} \mathcal{M}_\theta^{i,j,k}\| \tag{11}$$

We then scale $\mathcal{L}_\mathcal{M}$ by a coefficient ω and add it to the joint loss as shown in Figure 4.

IV. Experiments

In this section, we empirically evaluate the performance of the SV-Mix on various video recognition datasets, video recognition models and different settings to answer the following research questions:

- **RQ1**: How does SV-Mix perform on different video datasets when adopted on various recognition models?
- **RQ2**: How does SV-Mix compare with other data augmentation methods?
- **RQ3**: How do different settings affect SV-Mix?

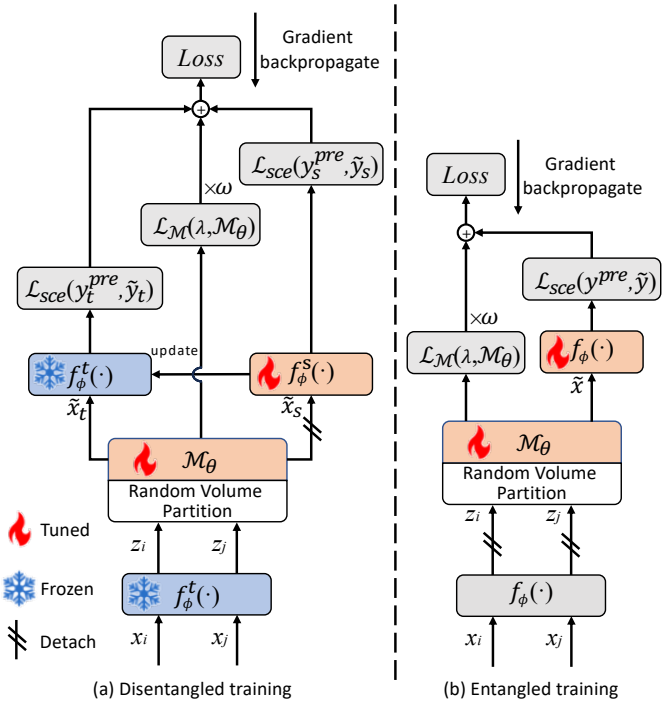


Fig. 4. The proposed disentangled training pipeline for jointly optimizing SV-Mix and the action recognition networks f_ϕ^s. In this pipeline, the gradient of SV-Mix is provided by a momentum-updated version of action recognition network f_ϕ^t. Therefore, the gradients of SV-Mix and f_ϕ^s are disentangled, which stabilizes the training process.

A. Datasets

Something-Something. Something-Something dataset consists of 174 fine-grained action categories that depict humans performing everyday actions with common objects. Recognizing actions in the Something-Something dataset heavily relies on identifying key regions and frames. Classical random linear interpolation methods, such as Mixup [10] or Cutmix [9], can weaken or delete these key patterns. The dataset has two versions, V1 [49] and V2 [50], with 110k and 220k videos respectively. We report the performances on the validation set as the annotations for test set are not released.

Mini-Kinetics. Kinetics-400 [51] is a widely used action recognition benchmark. It contains 240k training samples and 20k validation samples in 400 human action classes. Kinetics dataset mainly focus on static spatial appearance pattern. We create a mini version of Kinetics-400 dataset following [52] which accounts for half of the full Kinetics400 through randomly selecting half of the categories of Kinetics-400.

Diving48. Diving48 [53] is a fine-grained video dataset comprising approximately 18,000 trimmed video clips of 48 unambiguous dive sequences in competitive diving. Due to the diverse sub-poses distributed throughout the timeline in the dive sequences, it is crucial to capture the local sub-poses over time. The dataset provider has manually cleaned dive annotations, removing poorly segmented videos. We conduct experiments using the official train/validation split V2 on the updated version of the dataset.

EGTEA Gaze+. EGTEA Gaze+ [54] offers approximately 10,000 samples of 106 non-scripted daily activities that occur in a kitchen and provides researchers with a first-person per-

TABLE I
PERFORMANCE COMPARISONS WITH DIFFERENT ACTION RECOGNITION FRAMEWORKS ON STH-STH V1&V2 AND MINI-KINETICS DATASETS.

Method	Sth-Sth V1		Sth-Sth V2		Mini-Kinetics	
	Acc 1.(%)	ΔAcc 1.(%)	Acc 1.(%)	ΔAcc 1.(%)	Acc 1.(%)	ΔAcc 1.(%)
TSM	45.5	+1.7	59.3	+1.0	75.9	+0.7
TSM+SV-Mix	**47.2**		**60.3**		**76.6**	
R(2+1)D	45.9	+0.8	58.9	+1.4	75.5	+0.6
R(2+1)D+SV-Mix	**46.7**		**60.3**		**76.1**	
MViTv2	57.0	+0.9	67.4	+1.2	79.3	+0.2
MViTv2+SV-Mix	**57.9**		**68.6**		**79.5**	

TABLE II
PERFORMANCE COMPARISONS ON RELATIVELY SMALL-SCALE DATASETS INCLUDING UCF101, DIVING48 AND EGTEA GAZE+.

Method	UCF101		Diving48		EGTEA GAZE+	
	Acc 1.(%)	Δ Acc 1.(%)	Acc 1.(%)	ΔAcc 1.(%)	Acc 1.(%)	ΔAcc 1.(%)
TSM	85.2	+3.2	77.6	+2.6	63.5	+2.0
TSM+SV-Mix	**88.4**		**80.2**		**65.5**	
MViTv2	90.0	+2.2	80.7	+3.1	66.5	+1.3
MViTv2+SV-Mix	**92.2**		**83.8**		**67.8**	

spective. Unlike Something-Something and Diving48 datasets, EGTEA Gaze+ is filmed from a first-person point of view.

UCF101. The UCF101 [55] dataset comprises 13,320 videos with 101 classes in the wild. Due to its limited size, video models trained on UCF101 may lean towards overfitting. Therefore, UCF101 is suitable for measuring the effectiveness of video data augmentations.

B. Implementation Details

Baseline model. We evaluate the effectiveness of SV-Mix on both CNN and tranformer models. Among CNN models, we choose one 2D CNN based model, i.e. TSM [56] and one decomposed 3D CNN based model, i.e. R(2+1)D [25] because the core modules, i.e., temporal shift and 1D temporal convolstion are widely adopted as key components to construct other advanced CNN video models. Besides CNN video models, we also demonstrate the merits of SV-Mix on sophisticated transformer model, e.g., MViTv2 [4]. For efficiency consideration, we adopt ResNet-50 [57] as the backbone to construct TSM and R(2+1)D, as for MViTv2, we choose MViTv2-S as the baseline model.

Training. Following the common setting [20], we uniformly sample 8 or 16 frames from input videos for all datasets. As for resolution, we resize the short-side of frames to 256 maintaining the aspect ratio and then crop a 224×224 patch out of resized frames. Data augmentations used by baseline model such as random scaling before cropping and random horizontal flipping (for TSM and R(2+1)D) and RandAugment [42] (for MViTv2) are also adopted, unless otherwise statement. For CNN models, we train the network via SGD optimizer and we set the learning rate (lr) as $0.01 \times \frac{batchsize}{32}$. The total training epoch is set as 50 for all datasets. At epoch 20, 40, we decay lr by multiplying 0.1. The dropout ratio is set as 0.5. The backbone ResNet is pre-trained on ImageNet. As for transformer models, we adopt AdamW [58] instead of SGD and set lr as $2e{-}4 \times \frac{batchsize}{32}$. We train transformer models for 60 epochs for Something-Something datasets and 50 epochs for other datasets. Cosine learning rate schedule is adopted with 5 warmup epochs.

Inference. We sample 8 frames per video for CNN models and 16 frames for MViTv2. We utilize 224 × 224 central crop and 1 clip × 1 crop for testing CNN models except on UCF101, where we using 256 × 256 × 2 clip × 3 crop. We use 224 × 224 central crop to test MViTv2, test views is set as 3 crops × 1 clip on Something-Something, 2 clip × 3 crop on UCF101 and 1 clip × 1 crop for others

C. The Effectiveness of SV-Mix (RQ1)

Something-Something and Mini-Kinetics. We empirically evaluate the effectiveness of adopting SV-Mix on various video models using the Something-Something dataset in Table I. It can be observed that our proposed SV-Mix not only enhances the generalization ability of classical CNN models, i.e. TSM and R(2+1)D with obvious margins, but also significantly boosts the performance of the advanced MViTv2 model. Particularly, SV-Mix improves the performance of TSM by 1.63% on Something-Something V1 and 0.91% on Something-Something V2. As for R(2+1)D, equipping SV-Mix achieves smaller 0.80% gain on Something-Something V1, but it increases to 1.37% when applied to Something-Something V2. When adopted to advanced MViTv2 which achieves SOTA performance, our SV-Mix still improves MViTv2 by 0.82% on Something-Something V1 and 1.13% on Something-Something V2. It is worth noting that the improvements of SV-Mix on Something-Something V2 are larger than those on Something-Something V1 for R(2+1)D and MViTv2, despite the former having less of an overfitting problem due to its double size. On Mini-Kinetics dataset which relies more on spatial appearance patterns to be correctly recognized, SV-Mix illustrates consistent effectiveness on TSM, R(2+1)D and MViTv2. Specifically, SV-Mix enhances TSM and R(2+1)D to reach 76.6% and 76.1% on Mini-Kinetics which are 0.7% and 0.6% higher than the baselines. On MViTv2, SV-Mix demonstrates a minor increase in performance (+0.2%) compared with other baseline, we speculate that the reason for this phenomenon is due to the fact that Kinetics dataset primarily focuses on spatial features, while the MViTv2 as a powerful image backbone that utilizes advanced image pre-training

TABLE III
PERFORMANCE COMPARISONS BETWEEN SV-MIX AND OTHER DATA AUGMENTATION STRATEGIES ON UCF101 AND STH-STH V1 DATASETS.

Model w/ Aug	UCF101		Sth-Sth V1	
	Acc 1.(%)	ΔAcc 1.(%)	Acc 1.(%)	ΔAcc 1.(%)
TSM	85.2	-	45.5	-
+Mixup [10]	84.7	-0.5	44.6	-0.9
+Cutmix [9] (VideoMix [45])	86.9	+1.7	45.7	+0.2
+Cutmix&Mixup	87.0	+1.8	45.4	-0.1
+Cutout [59]	-	-	44.7	-0.8
+Augmix [60]	-	-	46.2	+0.7
+RandAug [42]	87.5	+2.3	-	-
+SV-Mix	**88.4**	**+3.2**	**47.2**	**+1.7**
+SV-Mix+RandAug	**89.6**	**+4.4**	-	-
MViTv2	90.0	-	57.2	-
+Cutmix&Mixup	91.6	+1.6	57.0	-0.2
+SV-Mix	**92.2**	**+2.2**	**57.9**	**+0.7**

techniques, limits the potential improvement in downstream tasks that emphasize spatial perception by data augmentation.

Other Datasets. We conducted further evaluations of SV-Mix on datasets with relatively small sizes, including UCF101, Diving48, and EGTEA Gaze+ in Table II. Compared to larger datasets such as Something-Something V1&V2, these datasets tend to suffer from more severe overfitting problems. As a result, the performance gains of SV-Mix are more significant on these datasets. Specifically, SV-Mix boosts the performance of TSM by 3.2%, 2.6% , 2.0% on UCF101, Diving48 and EGTEA GAZE+, respectively. On stronger MViTv2, the improvements are still significant: 2.2%, 3.1% , 1.3% on UCF101, Diving48 and EGTEA GAZE+, respectively.

D. Comparison with Other Augmentation (RQ2)

We compare our proposed SV-Mix with advanced data augmentation methods, as these methods are design for image augmentation, we simply expand them into video version by conducting the same augmentation in all frames, as most works do [4], [5], [61], [62]. We conduct comparison using TSM and MViTv2 on UCF101 and Something-Something V1. As shown in Table III, simply deleting (i.e. Cutout [59]) or exchanging (i.e. Cutmix/VideoMix [9], [45] [1]) a random spatial region bring little improvement or even decrease the recognition accuracy, we infer it may due to core motion area missing cause by these two methods. Mixup [10] which blurs the whole frame also brings no consistent enhancement on recognition accuracy. Augmix [60] which mixes several augmented views of one video sample and RandAugment [42] which randomly selects augmentations from a pre-defined augmentation set significantly boost the performance as they enrich the training diversity and maintain the core motion pattern. Compared with methods with sample mixing [9], [10] and Cutout [59], our SV-Mix perform much better consistently because of the motion pattern selection capability (spatial/temporal selective module) and better sample diversity (random volume partition).

[1]It is worth notice that VideoMix [45] explore the adaption of Cutmix [9] in video recognition and propose to cut the same regions for all frames in the video. Because this strategy simply inflates Cutmix along the temporal dimension, we refer to VideoMix as Cutmix in the following part of this paper as popular projects do [61], [62].

TABLE IV
ABLATION STUDY OF SPATIAL SELECTIVE MODULE AND TEMPORAL SELECTIVE MODULE IN SV-MIX ON STH-STH V1 DATASET.

Model w/ Mixups	Sth-Sth V1	
	Acc 1.(%)	ΔAcc 1.(%)
TSM	45.5	-
+Cutmix	45.7	+0.2
+Spa. Select	**47.0**	**+1.5**
+Mixup	44.6	-0.9
+Temp. Select	**46.6**	**+1.1**
+Mixup&Cutmix	45.4	-0.1
+Temp.&Spa. Select	**47.2**	**+1.7**

TABLE V
PERFORMANCE COMPARISON BETWEEN DISENTANGLED TRAINING AND ENTANGLED TRAINING ON STH-STH V1 DATASET.

Mix Module	En	DisEn	Top1(%)
TSM	—	—	45.5
+Spa. Select	✔		45.6
		✔	**47.0**
+Temp. Select	✔		45.2
		✔	**46.6**
+Temp&Spa. Select	✔		45.8
		✔	**47.2**

E. Ablation Study (RQ3)

In this subsection, we demonstrate ablation studies of SV-Mix to verify the merits of our design choose, including the effectiveness of spatial/temporal selective module, disentangled training and the robustness under hyperparameter changing.

Spatial/temporal selective module. For better understanding about how the components of SV-Mix influence the performance of action recognition, we ablate spatial selective module and temporal selective module in Table IV, as well

TABLE VI
EFFECTIVENESS OF λ EMBEDDING AND $\mathcal{L}_{\mathcal{M}}$.

Mix Module	λ Em.	$\mathcal{L}_{\mathcal{M}}$	Top1(%)
TSM	—	—	45.5
+SV-Mix	✘	✘	45.1
	✔	✘	46.9
	✘	✔	45.9
	✔	✔	**47.2**

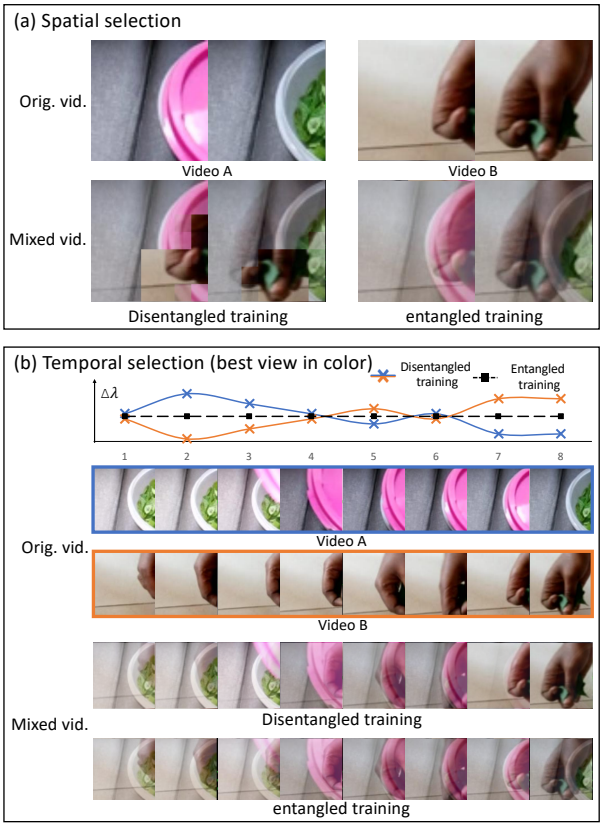


Fig. 5. Instance visualization of mixing two videos labeled as "*Pretending to close sth*" (Video A) and "*Tearing sth into two pieces*" (Video B). We compare mix videos generated by spatial selective module and temporal selective module under disentangled training and entangled training to verify the importance of training disentanglement. Both spatial and temporal selective modules fail to capture the informative spatial/temporal volumes and mix video samples in a uniform manner.

TABLE VII
PERFORMANCE COMPARISON OF DIFFERENT ENSEMBLE STRATEGIES AND PARAMETER SHARING SETTINGS OF SPATIAL/TEMPORAL SELECTIVE MODULES.

Mix Module	Prob. En.	Para. Share	Top1(%)
TSM	—	—	45.5
+SV-Mix	✘	✘	45.9
	✔	✘	46.2
	✘	✔	47.0
	✔	✔	**47.2**

as compare them with vanilla TSM model and their counterparts, i.e. CutMix [9] and Mixup [10]. As shown in Table IV, individual spatial selective module or temporal selective module already enhances the performance of TSM model (+1.52% and +1.10% respectively) which significantly outperform their non-parameter counterparts. Further ensembling spatial selective module and temporal selective module by random switching boosts the performance improvement to a higher level (+1.75%).

Disentangled training. We compare the proposed disentangled training pipeline with the intuitive entangled training pipeline using something-something V1 dataset. These two training strategy are conducted on spatial selective module, temporal selective module and the full SV-Mix. As shown

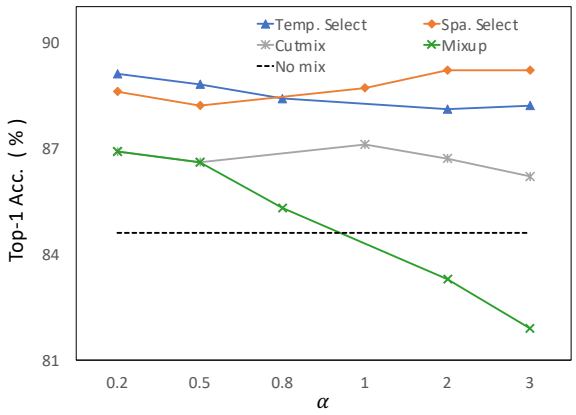


Fig. 6. The accuracy curve of SV-Mix with different α for the beta distribution $\lambda \sim Beta(\alpha, \alpha)$ on UCF101 dataset.

in Table V, disentangled training consistently outperforms intuitive entangled training under different mix module settings by substantial margins (+1.4%). To illustrate the contrast between the two training strategies more vividly, we provide comparison between mixed video samples generated by spatial/temporal selective module trained under these two training strategies. As shown in Figure V (a), trained by disentangled pipeline, spatial selective module keeps the fingertip actions areas that are highly correlated with the label *"Tearing sth into two pieces"* in video B for mixed sample generation. As a contrast spatial selective module fall into sub-optimal where videos are uniformly mixed across spatial regions. In temporal volume selection, similar phenomenon is observed. Specifically, as illustrated in the first row of Figure V (b), when trained by disentangled pipeline, temporal selective module is able to identify the first few frames in video A and the last few frames in video B as the informative frames which should be assigned with higher weights in the mixed video. When optimized by entangled training strategy, temporal selective module assigns similar weights for all frames and then fails to emphasize the informative frames in the mixed sample.

λ embedding & $\mathcal{L}_{\mathcal{M}}$. By embedding λ into the volume features through concatenation, SV-Mix is capable to control the mixing proportion of video samples. Additionally, $\mathcal{L}_{\mathcal{M}}$ provides explicitly guidance for SV-Mix to build correlation between λ and mixed training samples. The effectiveness of λ embedding and $\mathcal{L}_{\mathcal{M}}$ is illustrated in Table VI. In particular, in the absence of λ embedding and $\mathcal{L}_{\mathcal{M}}$, SV-Mix fails to improve the recognition performance (45.5% $\rightarrow$ 45.1%). This failure is attributed to the dissociation between the mixed samples and the predefined sample proportion λ, which results in the misalignment between the mixed samples and the mixed labels. With λ embedded in the input, SV-Mix boosts the TSM to reach 46.9% and the introduction $\mathcal{L}_{\mathcal{M}}$ further enhances the model to 47.2%.

Ensemble strategies and parameter sharing. We conduct performance comparison of different ensemble strategies and parameter sharing of spatial and temporal selective modules. As illustrated in Table VII, using shared parameters of spatial and temporal selective modules consistently outperforms unshared parameters settings. In addition, probabilistic ensemble (Eq 9) of spatial and temporal selective modules provide slight

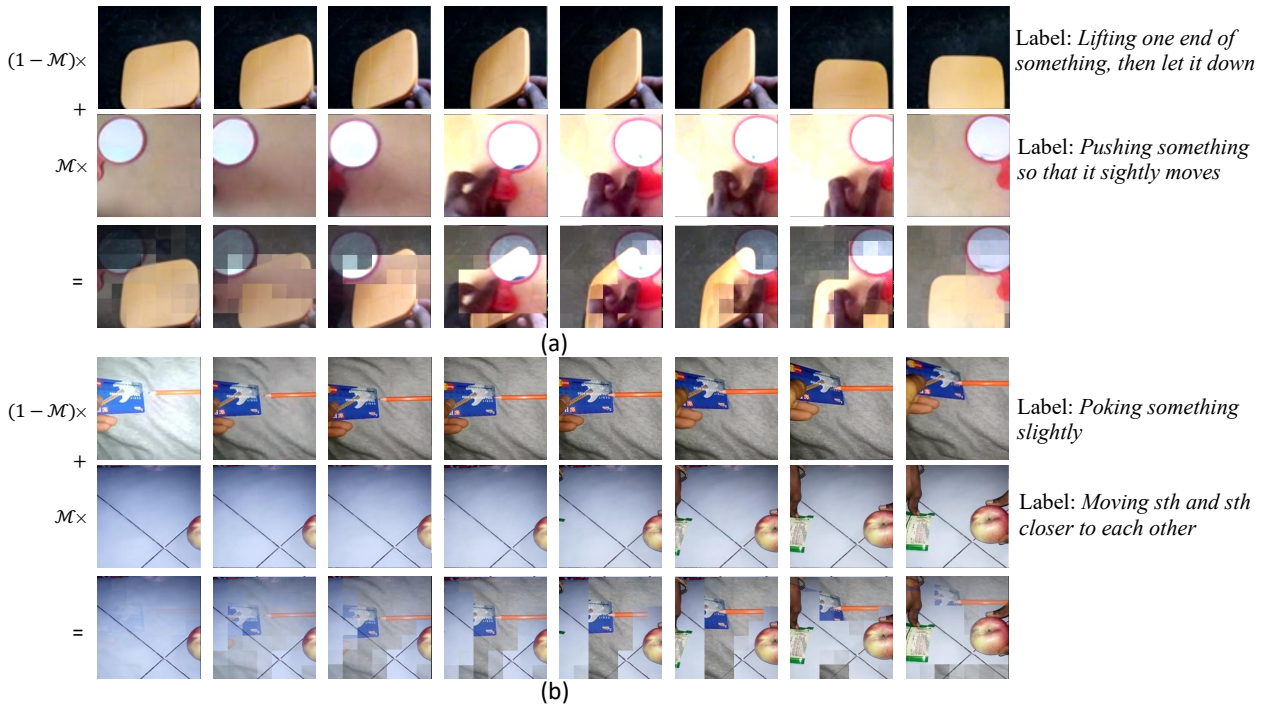


Fig. 7. Two examples of augmented video by spatial selective module on Sth-Sth V1 dataset. (a) Upper row: category *Lifting one end of something, then let it down*. Middle row: category *Pushing something so that it sightly moves*. Lower row: the mixed video sample with 35% top label plus 65% middle label. (b) Upper row: category *Poking something slightly*. Middle row: category *Moving something and something closer to each other*. Lower row: the mixed video sample with 32% top label plus 68% middle label.

improvements over average ensemble (Eq 8).

Distribution of λ. We further explore the influence of different distribution of λ by varying the α value which determines the distribution of λ as $\lambda \sim Beta(\alpha, \alpha)$. The results on UCF101 dataset are shown in Figure 6. The probability density of Beta distribution has a higher value around 0.5 when $\alpha > 1$, indicating that input samples have closer proportions in the mixed sample. As α decreases to lower than 1, the proportions of samples become more diverse. We evaluate the performance of spatial selective module and temporal selective module using TSM as backbone on $\alpha = \{0.2, 0.5, 1, 2, 3\}$ and $\alpha = \{0.2, 0.5, 0.8, 2, 3\}$ respectively. As shown in Figure 6, temporal selective module and spatial selective module demonstrate significant and robust performance improvement under different λ distribution, especially compared with mixup [10]. The performance of spatial selective module decreases when α goes lower than 1, which may be due to the fact that when α is lower than 1, the spatial selective module selects smaller spatial regions that may not contain complete motion regions, leading to semantic ambiguity in the training data-label pairs. In contrast, the performance of temporal selective module improves with smaller α. There may be two reason for this phenomenon, 1) temporal selective module select the whole frame instead of a small spatial region; 2) actions in UCF101 can be recognized using only a few frames. Notably, in other experiments, we fixed α as 0.8 and 1 for temporal and spatial selective module respectively, although adjusting α may lead to a slight improvement.

F. Analysis and Visualization

To demonstrate how SV-Mix works, we provide examples of mixed sample instances generated by the spatial selective module and temporal selective module. Figure 7 (a) shows the mixing process of two videos with label *"Lifting one end of something, then let it down"* (upper row) and *"Pushing something so that it sightly moves"* (middle row) respectively. Spatial selective module successfully selects patches that contain the interaction of *"hand"* and *"mirror"* in 4th $\sim$ 7th frames. Figure 7 (b) demonstrates mixing a video with label *"Moving something and something closer to each other"* (middle row) into a video labeled *"Poking something slightly"* (upper row), spatial selective module captures the region importance of both videos and maintains salient patches in the mixed video. It is worth noting that, unlike Cutmix, which sets the same proportion for all frames, our spatial selective module has a dynamic proportion for different frames, even though it does not contain any temporal module. For example, the proportion for the video in the middle row increases in 2rd $\sim$ 8th frames since the core *"pushing"* action is more significant in the later frames. This phenomenon demonstrates the merit of the spatial selective module, which not only captures the spatial saliency but also takes motion semantics into account for mixed video sample generation.

Unlike spatial selective module, our temporal selective module arranges whole frames and maintains the spatial pattern unchanged. We visualize the dynamic weights it assigns for different frames as well as the mixed video samples. Figure 8 (a) demonstrates the mixing process of a video labeled

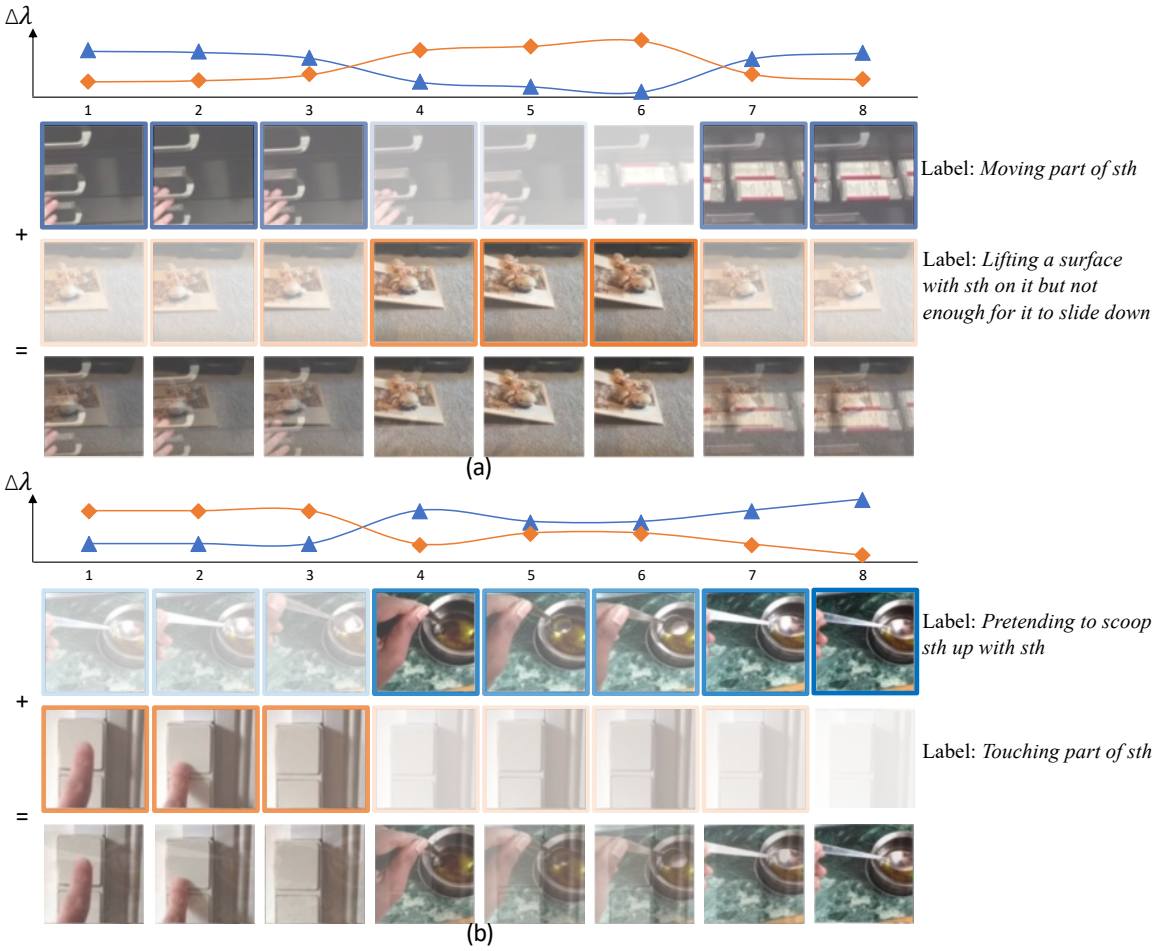


Fig. 8. Two examples of augmented video by temporal selective module on Sth-Sth V1 dataset. Upper row: the predicted attention weights of the frames from two videos. Middle two rows: the frames with high attention weights of the two input videos. Lower row: the mixed video sample.

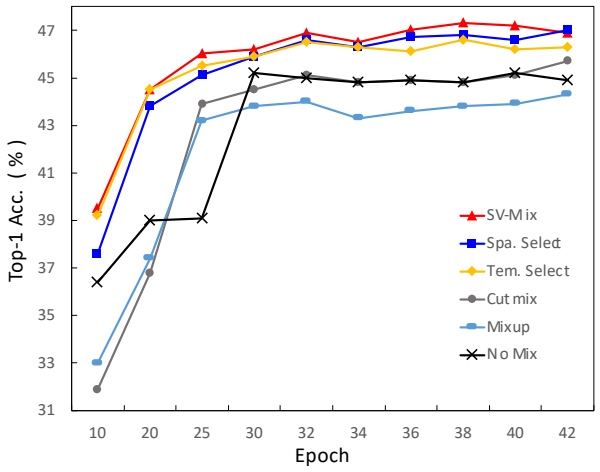


Fig. 9. Accuracy-epoch curves with different data augmentation strategies on Sth-Sth V1 dataset. The model with SV-Mix shows the faster convergence speed and better performance.

"Moving part of something" (upper row) and a video labeled *"Lifting a surface with something on it but not enough for it to slide down"* (middle row). Our temporal selective module selects the 1st $\sim$ 3rd frames and 7th $\sim$ 8th frames from the *"Moving part of something"* video because these frames show the start of the action (hand on the handle, pretend to push) and the end of the action (opened drawer), respectively. For the *"Lifting a surface with something on it but not enough for it to slide down"* video, our temporal selective module correctly selects the frames that capture the process of the surface being lifted. While in Figure 8 (b), a video labeled *"Touching part of something"* (middle row) is mixed into a video with label *"Pretending to scoop something up with something"* (upper row), our temporal selective module assigns high weights for the first 3 frames of video in middle row and low weights for the rest of this video because the motion information is concentrated in the first 3 frames.

We conducted further exploration on the model's training state under different data augmentation methods. The results, as shown in Figure 9, indicate that simple Cutmix and Mixup methods slow down the model convergence (i.e., 10$\sim$20 epochs) and stabilize the training process (i.e., 20$\sim$25 epochs), but they don't bring significant performance improvements. In contrast, our SV-Mix and its components (i.e., spatial selective module and temporal selective module) exhibit no convergence issues and even accelerate the model training at an early stage. Moreover, our SV-Mix and its components outperform competing methods by wide margins.

V. Conclusion and Limitation

We have presented SV-Mix augmentation, which provides a learnable data augmentation strategy for video action recognition. Particularly, we formulate the learnable video mixing process as the attention mechanism across volumes from two videos. The volumes with the most distinctive content compared with another video are treated as informative volumes, which should be maintained in the mixed video. To materialize our idea, we devise spatial selective module and temporal selective module to seek the valuable volumes on patch level and frame level, respectively. By randomly choosing one of the two modules, SV-Mix can produce both spatially mixed video and temporally mixed video. The modules in SV-Mix are jointly optimized with the subsequent action recognition framework in a novelly designed disentangled manner. The results of SV-Mix on five action recognition datasets demonstrate a consistent improvements across different benchmarks. Furthermore, as shown in the experiments with different recognition frameworks, SV-Mix demonstrates good potential to benefit a large range of neural networks from 2D CNNs, 3D CNNs to video transformers.

This study investigates the effectiveness of volume selection in action recognition. However, there remains an outstanding issue regarding the efficiency of SV-Mix. The disentangled training pipeline of SV-Mix necessitates multiple forward propagations of the backbone model, which results in longer training time. Efficient volume selection based video data augmentation is left for our future research and improvement.

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2021.

[2] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *ICCV*, 2021.

[3] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," in *CVPR*, 2022.

[4] Y. Li, C.-Y. Wu, H. Fan, K. Mangalam, B. Xiong, J. Malik, and C. Feichtenhofer, "Mvitv2: Improved multiscale vision transformers for classification and detection," in *CVPR*, 2022.

[5] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *CVPR*, 2022.

[6] K. Li, Y. Wang, G. Peng, G. Song, Y. Liu, H. Li, and Y. Qiao, "Uniformer: Unified transformer for efficient spatial-temporal representation learning," in *ICLR*, 2022.

[7] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *ICML*, 2021.

[8] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang, L. Yuan, L. Zhang, J.-N. Hwang *et al.*, "Grounded language-image pre-training," in *CVPR*, 2022.

[9] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," in *ICCV*, 2019.

[10] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," in *ICLR*, 2018.

[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *JMLR*, vol. 15, no. 1, pp. 1929–1958, 2014.

[12] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *ECCV*, 2016.

[13] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[14] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *ICCV*, 2019.

[15] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NIPS*, 2014.

[16] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *CVPR*, 2016.

[17] W. Wang, D. Tran, and M. Feiszli, "What makes training multi-modal classification networks hard?" in *CVPR*, 2020.

[18] A. Diba, V. Sharma, and L. Van Gool, "Deep temporal linear encoding networks," in *CVPR*, 2017.

[19] Z. Qiu, T. Yao, and T. Mei, "Deep quantization: Encoding convolutional activations with deep generative model," in *CVPR*, 2017.

[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 *ECCV*, 2016.

[21] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE trans. on PAMI*, vol. 35, no. 1, pp. 221–231, 2012.

[22] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *ICCV*, 2015.

[23] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *ICCV*, 2017.

[24] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *ECCV*, 2018.

[25] 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.

[26] D. Tran, H. Wang, L. Torresani, and M. Feiszli, "Video classification with channel-separated convolutional networks," in *ICCV*, 2019.

[27] C. Feichtenhofer, "X3d: Expanding architectures for efficient video recognition," in *CVPR*, 2020.

[28] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. on PAMI*, vol. 40, no. 6, pp. 1510–1517, 2018.

[29] A. Diba, M. Fayyaz, V. Sharma, A. Hossein Karami, M. Mahdi Arzani, R. Yousefzadeh, and L. Van Gool, "Temporal 3d convnets using temporal transition layer," in *CVPR Workshops*, 2018.

[30] N. Hussein, E. Gavves, and A. W. Smeulders, "Timeception for complex action recognition," in *CVPR*, 2019.

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

[32] Z. Qiu, T. Yao, C.-W. Ngo, X. Tian, and T. Mei, "Learning spatio-temporal representation with local and global diffusion," in *CVPR*, 2019.

[33] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *ICML*, 2021.

[34] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *ICCV*, 2021.

[35] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *ICCV*, 2021.

[36] S. Yan, X. Xiong, A. Arnab, Z. Lu, M. Zhang, C. Sun, and C. Schmid, "Multiview transformers for video recognition," in *CVPR*, 2022.

[37] F. Long, Z. Qiu, Y. Pan, T. Yao, J. Luo, and T. Mei, "Stand-alone inter-frame attention in video models," in *CVPR*, 2022.

[38] J. Yang, X. Dong, L. Liu, C. Zhang, J. Shen, and D. Yu, "Recurring the transformer for video action recognition," in *CVPR*, 2022.

[39] M. Patrick, D. Campbell, Y. Asano, I. Misra, F. Metze, C. Feichtenhofer, A. Vedaldi, and J. F. Henriques, "Keeping your eye on the ball: Trajectory attention in video transformers," in *NeurIPS*, 2021.

[40] Z. Zhong, L. Zheng, G. Kang, S. Li, and Y. Yang, "Random erasing data augmentation," in *AAAI*, 2020.

[41] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *arXiv preprint arXiv:1805.09501*, 2018.

[42] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," in *CVPR workshops*, 2020.

[43] J.-N. Chen, S. Sun, J. He, P. H. Torr, A. Yuille, and S. Bai, "Transmix: Attend to mix for vision transformers," in *CVPR*, 2022.

[44] Z. Liu, S. Li, D. Wu, Z. Liu, Z. Chen, L. Wu, and S. Z. Li, "Automix: Unveiling the power of mixup for stronger classifiers," in *ECCV*, 2022.

[45] S. Yun, S. J. Oh, B. Heo, D. Han, and J. Kim, "Videomix: Rethinking data augmentation for video classification," *arXiv preprint arXiv:2012.03457*, 2020.

[46] T. Kim, J. Kim, M. Shim, S. Yun, M. Kang, D. Wee, and S. Lee, "Exploring temporally dynamic data augmentation for video recognition," *arXiv preprint arXiv:2206.15015*, 2022.

[47] R. Gontijo-Lopes, S. Smullin, E. D. Cubuk, and E. Dyer, "Tradeoffs in data augmentation: An empirical study," in *ICLR*, 2021.
[48] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. Richemond, E. Buchatskaya, C. Doersch, B. Avila Pires, Z. Guo, M. Gheshlaghi Azar *et al.*, "Bootstrap your own latent-a new approach to self-supervised learning," in *NeurIPS*, 2020.
[49] R. Goyal, S. Ebrahimi 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*, 2017.
[50] 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.
[51] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *CVPR*, 2017.
[52] C.-F. R. Chen, R. Panda, K. Ramakrishnan, R. Feris, J. Cohn, A. Oliva, and Q. Fan, "Deep analysis of cnn-based spatio-temporal representations for action recognition," in *CVPR*, 2021.
[53] Y. Li, Y. Li, and N. Vasconcelos, "Resound: Towards action recognition without representation bias," in *ECCV*, 2018.
[54] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *ECCV*, 2018.
[55] 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.
[56] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *ICCV*, 2019.
[57] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.
[58] I. Loshchilov and F. Hutter, "Fixing weight decay regularization in adam," *arXiv preprint arXiv:1711.05101*, 2017.
[59] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.
[60] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," in *ICLR*, 2020.
[61] H. Fan, Y. Li, B. Xiong, W.-Y. Lo, and C. Feichtenhofer, "Pyslowfast," https://github.com/facebookresearch/slowfast, 2020.
[62] M. Contributors, "Openmmlab's next generation video understanding toolbox and benchmark," https://github.com/open-mmlab/mmaction2, 2020.