

TEAM-Net: Multi-modal Learning for Video Action Recognition with Partial Decoding

Zhengwei Wang¹
villa.wang.zhengwei@gmail.com

Qi She*²
sheqi1991@gmail.com

Aljosa Smolic¹
smolica@tcd.ie

¹ V-SENSE
School of Computer Science and
Statistics, Trinity College Dublin
Ireland

² ByteDance
China

Abstract

Most of existing video action recognition models ingest raw RGB frames. However, the raw video stream requires enormous storage and contains significant temporal redundancy. Video compression (e.g., H.264, MPEG-4) reduces superfluous information by representing the raw video stream using the concept of Group of Pictures (GOP). Each GOP is composed of the first I-frame (aka RGB image) followed by a number of P-frames, represented by motion vectors and residuals, which can be regarded and used as pre-extracted features. In this work, we 1) introduce sampling the input for the network from partially decoded videos based on the GOP-level, and 2) propose a plug-and-play multi-modal Learning Module (TEAM) for training the network using information from I-frames and P-frames in an end-to-end manner. We demonstrate the superior performance of TEAM-Net compared to the baseline using RGB only. TEAM-Net also achieves the state-of-the-art performance in the area of video action recognition with partial decoding. Code is provided at <https://github.com/villawang/TEAM-Net>.

1 Introduction

Video understanding has drawn an increasing amount of attention, since video data accounts for 82% of all Internet traffic by 2022 [4]. For instance, millions of videos are uploaded to TikTok, Douyin, and Xigua Video to be processed everyday, wherein understanding video content acts a pivotal part. One of the most important tasks in video understanding is to understand human actions, which has many real-world applications, such as Virtual Reality/Augmented Reality (VR/AR) and human computer interaction. Video data contains a rich source of visual content including appearance information in each individual frame and motion information across consecutive frames, which poses challenges for effectively extracting information from video data.

Traditional action recognition approaches [2, 5, 19, 20, 38] investigated better modeling video data either by designing new architectures [5] or embedded modules [20]. These architectures and embedded modules are designed for highlighting the spatio-temporal perspective contained in videos. Most of these approaches are directly designed for images to parse a video frame by frame. However, video data contains significant redundancy both on

the spatial and the temporal dimension. For instance, an 1h of 720p video can be compressed from 222GB raw to 1GB [42], which indicates redundant information contained in videos.

Performing action recognition in the *partially compressed domain* becomes a recent interest [1, 11, 18, 28, 42]. The difference between performing in the raw domain (full decoding) and partially compressed domain (partial decoding) is illustrated in Figure 1. In terms of partial decoding on the right, video data is parsed by a stream of Group of Pictures (GOPs). Each GOP starts with an intra-frame (I-frame, aka an independently encoded RGB image) followed by n P-frames ($n = 11$ in this work). Regarding full decoding, RGB images (highlighted in red on the left top) can be reconstructed from the first I-frame and following consecutive P-frames in the corresponding GOP shown on the bottom left [42]. In other words, highlighted RGB images in red *only* depend on the first I-frame and a number of P-frames in the current GOP. Traditional action recognition approaches randomly sample from these RGB images. However, it would make more sense to sample images from different GOPs and to perform on the GOP-level. Further, on the GOP-level, we are able to take benefits from additional information provided by P-frames i.e., two extra modalities Motion Vectors (MVs) and residuals. These two modalities represent motion and texture/color change information between the current decoded RGB image and its reference I-frame, which are very useful for action reasoning. This information can be regarded and used as pre-extracted features to improve the network performance. It should be noticed that previous works [1, 11, 18, 28, 42] term this type of action recognition as ‘*compressed video action recognition*’. We argue the terminology ‘*compressed*’ is not fully correct because original compressed video formats such as MPEG-4 and H.264 stored in the Video Coding Layer (VCL) do not contain decoded frames i.e., they are compressed bit streams [40]. All previous works used decoded I-frames, and computed representative images from decoded MVs and residuals for training the network but not the compressed bit stream data directly. Unlike traditional action recognition that decodes every RGB image, this operation decodes the first I-frame and some information from P-frames in each GOP. So we name such a domain for processing video data as ‘*partially compressed domain*’, which refers to ‘*compressed domain*’ in previous works.

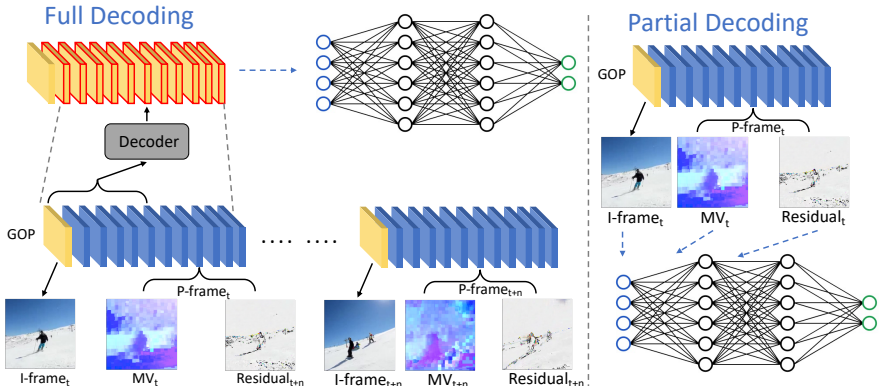


Figure 1: Comparison between traditional video action recognition (left) and partially decoded video action recognition (right). Left: Decoded RGB images are fed to the network. RGB images highlighted in red are reconstructed from the first I-frame and consecutive P-frames [42] in the corresponding GOP. Right: Without decoding every RGB image in a GOP, the partial decoding uses the first decoded I-frame and partial information from one P-frame (including MV and residual) in a GOP as an input.

Previous works [1, 11, 18, 28, 42] investigated adding two extra modalities MVs and residuals in the partially compressed domain to improve the model performance. Most of current approaches [11, 28, 42] require training separated backbones for I-frames, MVs and residuals respectively, in which fusion only comes at the inference stage. This operation causes expensive computation for training models especially for a large-scale dataset. The fusion capability is also constrained by omitting mid-level fusion during training. In this work, we propose a plug-and-play multi-modal LEARNING Module (TEAM) to help train the network with three modalities by ingesting partially decoded videos in an end-to-end manner. We study the TEAM by adopting the widely adopted Temporal Segment Network (TSN) framework [35] in the partially compressed domain. It is worth nothing that our TEAM can be embedded to other advanced action recognition frameworks such as TSM [20]. To this end, we find the network equipped with our TEAM (TEAM-Net) outperforms the TSN in the raw domain i.e., using RGB only, and CoViAR in the partially compressed domain. Figure 2 shows the visualization for well-trained TSN, CoViAR and TEAM-Net. It can be seen that TEAM-Net successfully makes use of MV and residual modalities for better action reasoning compared to the other two baselines. In summary, our contributions are three-fold:

- We propose a novel plug-and-play module for fusing three modalities, which enables training the network with I-frames, MVs and residuals in an end-to-end manner for video action recognition with partial decoding.
- Unlike traditional action recognition that samples inputs on the frame-level, we introduce sampling inputs on the GOP-level for training the network, which can be understood as an extension of TSN in the partially compressed domain.
- Extensive experiments have been conducted to show that our proposed TEAM-Net outperforms the traditional action recognition baseline TSN and is superior to current state-of-the-art in video action recognition with partial decoding.

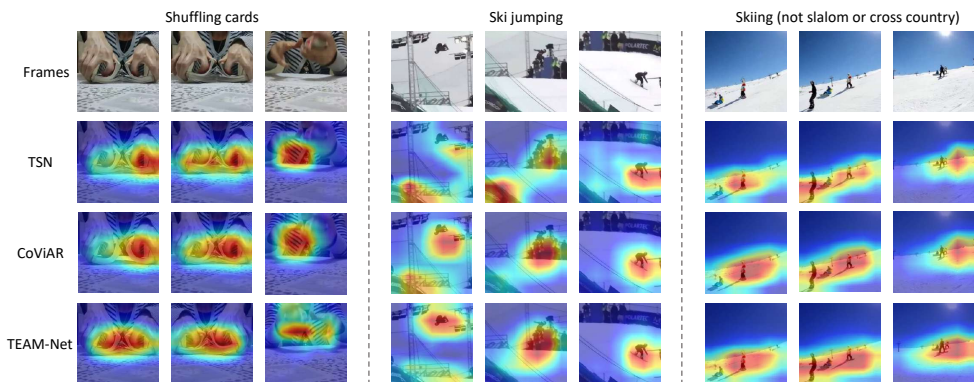


Figure 2: Visualization for significant features extracted by TSN, CoViAR and TEAM-Net. Features extracted by each method are visualized by using CAM [46]. It should be noted that our TEAM-Net does not contain any temporal modeling module for capturing motion information in videos, but uses the pre-extracted motion features from the MVs.

2 Related Works

Traditional Video Action Recognition. Recent video action recognition approaches normally operate in the raw video domain i.e., RGB images decoded from videos are fed to the network [5, 33, 35]. Current approaches can be mainly divided into two categories 1) 3D

CNN-based approaches (clip level) [5, 7, 33], and 2) 2D CNN with efficient temporal module design (frame level) [13, 19, 20, 38]. Different from image data, videos contain both spatial information in each individual frame and temporal information in the sequence i.e., movements. 3D CNNs utilize 3D convolutions to characterize spatio-temporal information for the clip input directly (a clip contains a number of frames). I3D [2] inflated the ImageNet pre-trained 2D convolution to a 3D convolution. SlowFast networks [5] were proposed to model inconsistent action tempos e.g., ‘*running vs walking*’. It comprised a slow path for modeling slow actions and a fast path for modeling fast actions respectively. While 3D CNN-based approaches have achieved exciting performance on several benchmark datasets, these models contain massive parameters and require expensive computation. Various problems may arise, such as overfitting [7], difficulty in converging [34] and slow inference [47]. Even though recent works [26, 34] have demonstrated that the 3D convolution can be factorized to lessen computations to some extent, the computation of 3D CNN-based models is still much more of a burden compared to 2D CNN-based models. Different from optimizing 3D CNNs on clips directly, 2D CNNs deploy 2D convolutions in a frame-level manner. TSN [35] applied 2D CNNs to video action recognition and introduced the concept of ‘*segment*’ to sample frames from videos i.e., extract short snippets over a long video sequence with a uniform sparse sampling scheme. Direct use of 2D CNNs is difficult to fully capture the temporal information of video. Current mainstream methods focus on designing an efficient module that can be embedded into 2D CNNs for temporal reasoning. TSM [20] introduced a shift operation for a part of channels along the temporal dimension. TEA [19] proposed a motion excitation module for short range temporal modeling and a multiple temporal aggregation module for long range temporal modeling. These two modules are connected sequentially. ACTION-Net [38] proposed a multipath excitation module for spatio-temporal, channel and motion modeling. These proposed modules can be inserted into 2D CNNs, in which temporal modeling is introduced to the network.

Action Recognition from Partially Decoded Video. Apart from manually designing methods for extracting inherent information in videos, video data itself contains useful information that previous approaches aim to model, such as motion information. Two-stream [2] approaches have demonstrated that augmenting input data by adding optical flow [12] is able to significantly enhance the performance for video action recognition. However, computation of optical flow is very expensive, which is not feasible for those applications requiring low latency in real life. On the other hand, exploiting information computed for encoding of video [29] recently comes into focus for video action recognition [11, 18, 28, 42, 44, 45]. Previous works [11, 18, 28, 42] tackled video action recognition using I-frames, MVs and residuals. Works [44, 45] extracted MVs from videos to replace the optical flow. Wu *et al.* [42] proposed the CoViAR model that contained three 2D CNNs for ingesting three modalities respectively. It is worth nothing that these three 2D CNNs were trained independently, in which fusion of three modalities was only performed at the inference phase. Huo *et al.* [11] proposed an Aligned Temporal Trilinear Pooling (ATTP) module for better fusing three modalities at the inference stage. Due to the low resolution of MVs, the DMC-Net [28] applied Generative Adversarial Networks (GANs) [6, 39] for generating motion cues by adopting the information of MVs, residuals and optical flow. Inspired by SlowFast networks [5], Li *et al.* [18] proposed an end-to-end Slow-I-Fast-P (SIFF) framework that treated I-frames as the slow pathway and P-frames as the fast pathway. Battash *et al.* [1] proposed an end-to-end framework that ingests a clip input as one GOP, in which a teacher-student distillation strategy was utilized for improving the model performance. IMRNet [43] employed bidirectional dynamic connections between I-frame & MV pathways and between

I-frame & residual pathways.

Multi-modal Action Recognition. Previous works have studied learning of multiple streams present in video [15, 22, 24, 25]. Most of these works deployed the audio stream as complementary information to enhance the performance. Our work makes use of inherent visual information, which exists in partially decoded video, and can be integrated to frameworks mentioned above. In this work, we tackle the multi-modal learning side that improves the network performance on action recognition by adopting multi-modal data in partially decoded videos. Previous works [1, 18] investigated the feasibility of training models in the partially compressed domain in an end-to-end manner. [18] is only suitable for a SlowFast-like architecture while the upper bound performance of [1] the teacher model trained in the raw domain. Luvizon *et al.* [23] proposed a multi-task framework jointly trained by pose estimation and action recognition tasks. PoseMap [21] employed a two-stream network to process spatio-temporal pose heatmaps and inaccurate poses separately. Wang *et al.* [37] presented a two-stream network for processing RGB and depth modalities in videos. A bi-linear pooling block was investigated for RGB-D video action recognition in [9]. Wang *et al.* [36] proposed gradient blending for avoiding overfitting in multi-modal networks.

Attention Mechanisms. Attention mechanisms [17] have been widely adopted in computer vision tasks. Hu *et al.* [10] proposed a SE block for explicitly modeling channel interdependencies using the attention mechanism for the channel dimension. CBAM [41] was proposed beyond the SE block by adding an extra attention map along the spatial dimension. Joze *et al.* [14] investigated attention mechanisms for fusing multi-modal features from the channel perspective and proposed a MMTM module beyond the SE. Inspired by previous works [10, 14, 41], we propose a mulTi-modal lEarning Module (TEAM) for fusing I-frames, MVs and residuals with respect to the channel dimension and the spatial dimension at different levels inside an end-to-end model.

3 Design of TEAM

We first clarify notations to be used in this section: N (batch size), T (temporal length), C (channel), H (height), W (width) and r (reduction ratio). Given input feature maps to TEAM: $\mathbf{F}_1 \in \mathbb{R}^{NT \times C_1 \times H \times W}$, $\mathbf{F}_2 \in \mathbb{R}^{NT \times C_2 \times H \times W}$ and $\mathbf{F}_3 \in \mathbb{R}^{NT \times C_3 \times H \times W}$, for three modalities I-frames, MVs and residuals respectively (the spatial size for three modalities is equal but the channel number is different due to different backbones that are used for different modalities), we aim to optimize the entire model jointly using three modalities. To achieve this, we construct joint feature representations along the channel and the spatial dimension to be processed by a channel fusion module and a spatial fusion module respectively.

Channel Fusion. Figure 3(a) illustrates the channel fusion module. As we only care about channel information at this stage, we average each modality feature on the spatial dimension

$$\mathbf{F}_{c,i} = \frac{1}{H \times W} \sum_{m=1}^H \sum_{n=1}^W \mathbf{F}_i[:, :, m, n], \quad (1)$$

where $\{c, i\}$ indicates the channel fusion (to be distinguished from the spatial fusion) and modality type i.e., $i = 1$ for I-frames, $i = 2$ for MVs, and $i = 3$ for residuals, and \mathbf{F}_i is the input for the corresponding modality. We then construct a joint channel representation $\mathbf{z}_c \in \mathbb{R}^{NT \times C_z}$ ($C_z = C_1 + C_2 + C_3$) by concatenating three feature maps along the channel dimension. We follow the same squeeze strategy as in [10] to reduce the channel dimension for the joint channel representation as

$$\mathbf{z}_l = \text{ReLU}(\mathbf{z}_c \mathbf{W}_c), \quad (2)$$

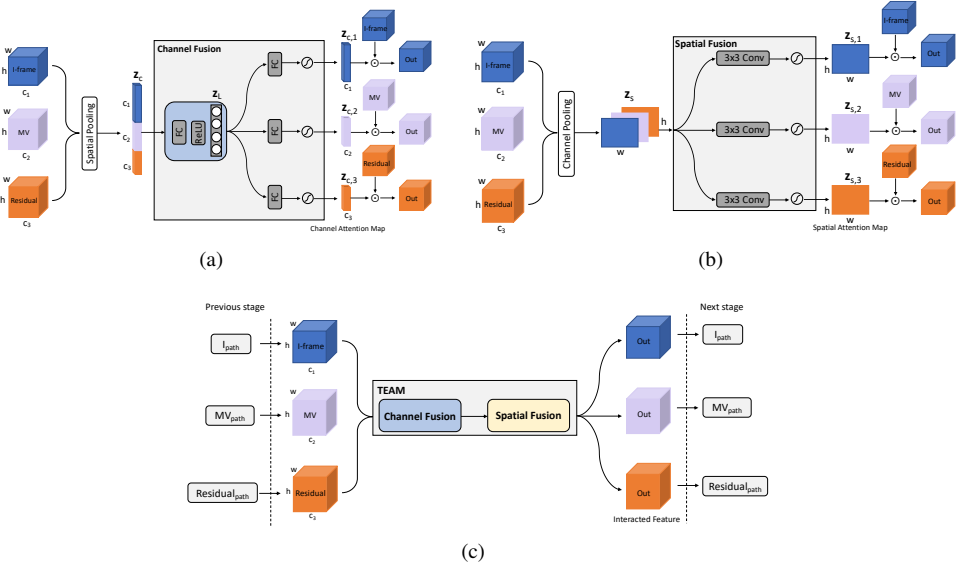


Figure 3: Schematics of channel fusion, spatial fusion and TEAM module. The TEAM module in (c) involves two-stage fusion of (a) the channel fusion module and (b) the spatial fusion module.

where $\mathbf{z}_l \in \mathbb{R}^{NT \times C_z/r}$ and $\mathbf{W}_c \in \mathbb{R}^{C_z \times C_z/r}$ is the FC weight. The squeezed representation \mathbf{z}_l is then expanded by three FC layers followed by a Sigmoid to generate three attention maps for each modality as follows

$$\mathbf{z}_{c,i} = \sigma(\mathbf{z}_l \mathbf{W}_{c,i}), \quad (3)$$

where $\mathbf{z}_{c,i} \in \mathbb{R}^{NT \times C_i}$ is the attention map and $\mathbf{W}_{c,i} \in \mathbb{R}^{C_z/r \times C_i}$ is the FC weight for the corresponding modality. σ is the Sigmoid function. The final output for each modality can be computed as

$$\mathbf{F}_{c,i}^{out} = \mathbf{F}_i \odot \mathbf{z}_{c,i}, \quad (4)$$

where \odot is the element-wise multiplication, \mathbf{F}_i is the same input as mentioned in eq (1), fed to the channel fusion module corresponding to each modality, and $\mathbf{F}_{c,i}^{out} \in \mathbb{R}^{NT \times C_i \times H \times W}$ has the same dimension as the input $\mathbf{F}_i \in \mathbb{R}^{NT \times C_i \times H \times W}$.

Spatial Fusion. The channel fusion ignores local spatial information, which will be addressed by the spatial fusion module as seen in Figure 3(b). We first globally average the output $\mathbf{F}_{c,i}^{out}$ of the channel fusion module over the channel dimension

$$\mathbf{F}_{s,i} = \frac{1}{C_i} \sum_{j=1}^{C_i} \mathbf{F}_{c,i}^{out}[:, j, :, :], \quad (5)$$

where $\mathbf{F}_{s,i} \in \mathbb{R}^{NT \times 1 \times H \times W}$. We then concatenate these three feature maps over the channel dimension to get a joint spatial representation $\mathbf{z}_s \in \mathbb{R}^{NT \times 3 \times H \times W}$. We generate three spatial attention maps for each modality using the joint representation \mathbf{z}_s

$$\mathbf{z}_{s,i} = \sigma(\mathbf{W}_{s,i} * \mathbf{z}_s), \quad (6)$$

where $\mathbf{z}_{s,i} \in \mathbb{R}^{NT \times 1 \times H \times W}$, $\mathbf{W}_{s,i}$ is a 3×3 2D convolution kernel for modality i , and $*$ is the convolution operation. Similar to the channel fusion, the final output of the spatial fusion can be computed as

$$\mathbf{F}_{s,i}^{out} = \mathbf{F}_{c,i}^{out} \odot \mathbf{z}_{s,i}, \quad (7)$$

where $\mathbf{F}_{s,i}^{out} \in \mathbb{R}^{NT \times C_i \times H \times W}$ has the same dimension as the input $\mathbf{F}_i \in \mathbb{R}^{NT \times C_i \times H \times W}$.

Instantiations. The TEAM module is composed of the channel and the spatial fusion in a sequential manner as seen in Figure 3(c). TEAM ingests three modalities and outputs interacted features for each modality. Our idea of TEAM is generic, and it can be instantiated with different backbones (e.g., [8, 27, 31]) and action recognition implementation specifics (e.g., [19, 20, 35, 38]). In this work, we demonstrate instantiation of our TEAM-Net on the ResNet architecture for action recognition. TEAM is inserted after Conv₁, Res₂, Res₃ and Res₄ respectively. Details of the instantiation can be referred to Figure 1 and Table 2 in *Supplementary Materials A*.

4 Experiments

4.1 Setups

Datasets. We evaluate the performance of the proposed TEAM-Net on three public action recognition datasets Kinetics-400 [2], UCF-101 [30] and HMDB-51 [16]. Kinetics-400 contains 400 human action categories and provides URL links for ~240k training videos and ~20k validation videos. We successfully collected 213,991 training videos and 17,575 validation videos since around 10% of URLs are no longer valid. We directly report the accuracy on the validation set for Kinetics-400. UCF-101 contains 101 classes with 13,320 videos. HMDB-51 includes 51 classes with 6,766 videos. For these two datasets, we follow previous works [11, 18, 42] to utilize three splits for training and evaluation. The average accuracy on three splits is reported. Following previous works [28, 42], we use MPEG-4 encoded videos, which have on average 11 P-frames for every I-frame in each GOP.

Training and Inference. We prepared partially decoded videos to conduct our experiments on video action recognition tasks by using the same encoding tool as CoViAR [42]. Given a partially decoded video input, we firstly divided it into T segments of equal duration. Then we randomly selected one GOP from each segment to obtain a clip input with T GOPs. The first frame in each GOP was chosen as an I-frame. The MV and residual were obtained from a randomly picked P-frame in each GOP. In other words, one partially decoded video clip input contains T I-frames, T MVs and T residuals to be fed to the network. The size of the shorter side of these frames was fixed to 256, and corner cropping and scale-jittering were utilized for data augmentation. Each cropped frame was finally resized to 224×224 for training the network. We utilized 2D ResNet-50 as a backbone for the I-frame modality, and 2D ResNet-18 as backbones for MV and residual modalities for both TEAM-Net and CoViAR baseline. For TSN, we used 2D ResNet-50 as the backbone. Models were trained on a NVIDIA DGX station with four Tesla V100 GPUs. We adopted SGD as the optimizer with a momentum of 0.9 and a weight decay of 5×10^{-4} . Batch size was set to $N = 80$. For Kinetics, network weights were initialized using ImageNet pretrained weights. We started an initial learning rate of 0.01 and reduced it by a factor of 10 at 30, 40, 45 epochs and stopped at 50 epochs. For UCF-101 and HMDB-51, the network was finetuned using a pretrained Kinetics model. We started an initial learning rate of 0.005 and reduced it by a factor of 10 at 15, 20, 25 and stopped at 30 epochs. Following previous works [42], we used 25 segments with a center-crop strategy for inference.

4.2 Improving Baselines

Action Recognition Performance. We compare TEAM-Net with two fundamental baselines with full decoding (TSN) and partial decoding (CoViAR) on three datasets in Table 1. It can be noticed that both TEAM-Net and CoViAR are able to improve the performance on UCF-101 and HMDB-51, which indicates that MVs and residuals contained in the partially decoded videos help the action reasoning. Interestingly, we observe there is a performance drop (0.9%) of CoViAR on Kinetics compared to TSN. On the contrary, our TEAM-Net is still able to improve the performance by 2.2% compared to TSN. This demonstrates that TEAM-Net is more robust on different datasets compared to CoViAR. Our findings here are consistent with previous research [35, 42] that proves consecutive frames are highly redundant in videos. Proper use of P-frames in partially decoded videos is able to boost the model performance [11, 18, 28, 42].

Computational Efficiency. Current frameworks [19, 35, 38] usually sample frames sparsely from videos, that means there is no significant difference on the computation when the model ingests raw videos or partially decoded videos i.e., the number of sampled frames is same if excluding P-frames. However, the computation is a bit heavier when taking P-frames into account. Table 1 shows the inference speed of TEAM-Net and two baselines. TSN is about 1.5 times faster compared to TEAM-Net. TEAM-Net performs closely to CoViAR but saves the training cost significantly i.e., *only need to train one model instead of three*.

Table 1: TEAM-Net consistently outperforms baselines on three representative datasets. All methods use 8 frames for fair comparison.

Model	Kinetics-400	UCF-101	HMDB-51	Speed
TSN ¹	70.0	89.7	66.8	45 V/s
CoViAR ²	69.1 (↓0.9)	91.0 (↑1.3)	73.2 (↑6.4)	31 V/s
TEAM-Net	72.2 (↑2.2)	94.3 (↑4.6)	73.8 (↑7.0)	30 V/s

¹ Different from [35], TSN on UCF-101 and HMDB-51 are finetuned using a pretrained model on Kinetics-400 in this work, which give higher accuracies compared to [35].

² We re-implemented CoViAR using the official code in [42] by replacing ResNet-152 with ResNet-50 for the I-frame modality for fair comparison.

Table 2: Comparison with state-of-the-arts using partially decoded videos on Kinetics, UCF and HMDB. Our TEAM-Net is an end-to-end (ETE) network and optical flow (OF) free.

Method	OF	ETE	Backbone	Kinetics	UCF	HMDB
DMC-Net [28]	✓	✗	ResNet152 (I), ResNet18 (P)	-	92.3	71.8
ATTP [11]	✓	✗	EfficientNet (I), EfficientNet (P)	-	91.1	62.9
IMRNet [43]	✓	✓	3D-ResNet50 (I), 3D-ResNet50 (P)	-	95.1	72.2
CoViAR [42]	✗	✗	ResNet50 (I), ResNet18 (P)	69.1	91.0	73.2
SIFP [18]	✗	✓	SlowFast-ResNet50	-	94.0	72.3
MFCD-Net [1]	✗	✓	Multi-Fiber Network [3]	68.3	93.2	66.9
IMRNet [43]	✗	✓	3D-ResNet50 (I), 3D-ResNet50 (P)	-	92.6	67.8
TEAM-Net	✗	✓	ResNet50 (I), ResNet18 (P)	72.2	94.3	73.8

4.3 Comparison with State-of-the-Arts

We compare our approach with current state-of-the-arts in the area of action recognition using partially decoded videos on Kinetics-400, UCF-101 and HMDB-51 in Table 2. Un-

like recent works [11, 18, 43] that utilize advanced backbones such as 3D CNN [7], Slow-Fast Networks [5] and EfficientNet [32], it is worth nothing that our TEAM-Net utilizes 2D ResNet as backbones for three modalities. TEAM-Net learns the informatively fused features from I-frames and P-frames through the TEAM module. More advanced backbones are able to boost the performance for TEAM-Net. Our TEAM-Net, together with SIFP [18], MFCD-Net [1] and IMRNet [43], enjoys end-to-end training. It can be seen that our TEAM-Net outperforms current state-of-the-arts without optical flow and achieves competitive results even compared to those approaches with optical flow. Regarding HMDB-51, TEAM-Net outperforms all previous approaches with/without optical flow. In terms of UCF-101, TEAM-Net performs closely to IMRNet (3D CNN backbone) with optical flow. Only few methods are evaluated on Kinetics-400 since many approaches suffer from expensive training i.e., require training three separated backbones for each modality. Our TEAM-Net also performs well on Kinetics-400, which demonstrates good ability for generalization across datasets.

4.4 Ablation Study

Arrangements of Channel and Spatial Fusion. We investigate different arrangements of channel and spatial fusion from a network engineering side. Table 3 shows 5 possible arrangements for formulating a TEAM module. $TEAM_c$ and $TEAM_s$ indicate channel fusion only or spatial fusion only contained in the module. $TEAM_{c+s}$ refers to connecting channel and spatial module in a parallel way. $TEAM_{c \rightarrow s}$ indicates channel fusion followed by spatial fusion, vice versa for $TEAM_{s \rightarrow c}$. It can be seen that three possible combinations of channel and spatial fusion work better than channel fusion only and spatial fusion only on both UCF and HMDB in Table 3. The sequential arrangement, in which the channel fusion is followed by the spatial fusion $TEAM_{c \rightarrow s}$, performs the best compared to the other two.

Table 3: Arrangements of channel and spatial fusion. Performances are evaluated on UCF and HMDB using 8-frame inputs for training. Average accuracy for three splits is reported.

Dataset	$TEAM_c$	$TEAM_s$	$TEAM_{c+s}$	$TEAM_{s \rightarrow c}$	$TEAM_{c \rightarrow s}$
UCF-101	93.7	93.7	94.0	94.1	94.3
HMDB-51	73.0	72.9	73.1	73.2	73.8

Table 4: Performance of different embedded locations.

Dataset	Stage _{1}	Stage _{1,2}	Stage _{1,2,3}	Stage _{1,2,3,4}	Stage _{1,2,3,4,5}
UCF-101	93.6	93.2	93.3	94.3	91.7
HMDB-51	72.8	73.1	72.7	73.8	70.4

Location of TEAM. The default embedded locations of TEAM are after $Conv_1$, Res_2 , Res_3 and Res_4 respectively i.e., Stage_{1,2,3,4}, which is shown in Figure 1 in *Supplementary Materials A*. We find that including the last stage fusion (Res_5) deteriorates the fusion performance. We suppose this may be caused by the fact that the spatial resolution is too low at the last stage, which causes inaccurate spatial fusion and degrades the performance. The location Stage_{1,2,3,4} works the best, which indicates the efficacy of the mid-level fusion for these three modalities.

Understanding of TEAM. Previous experimental results show the promising performance of TEAM-Net. Here we provide insights behind the TEAM-Net and the effectiveness of the two fusion modules. Regarding the channel fusion, it takes global spatial information (see eq (1)) into account to fuse each modality i.e., it focuses on ‘*what*’ is meaningful given a frame. This indicates that the channel fusion has low spatial resolution, which results in

inaccurate local information reasoning. As seen on the 3rd row in Figure 4(a), there is a strong activation on the right for each frame detected by the channel fusion, while the main action is happening in the middle. The spatial fusion and TEAM characterize these actions happening in the local spatial space successfully, which suggests that spatial fusion focuses on ‘where’ is an informative part. However, it could be sensitive to unrelated movements in video. For instance, in the 2nd row in Figure 4(b), the spatial fusion falls into paying attention to the edge on several frames, which is probably caused by camera movement. Our proposed TEAM module takes ‘what’ and ‘where’ into account while fusing different modalities by incorporating two modules. Both recognition performance and visualization results confirm the effectiveness of the proposed approach. We provide more visualization results in *Supplementary Materials C*.

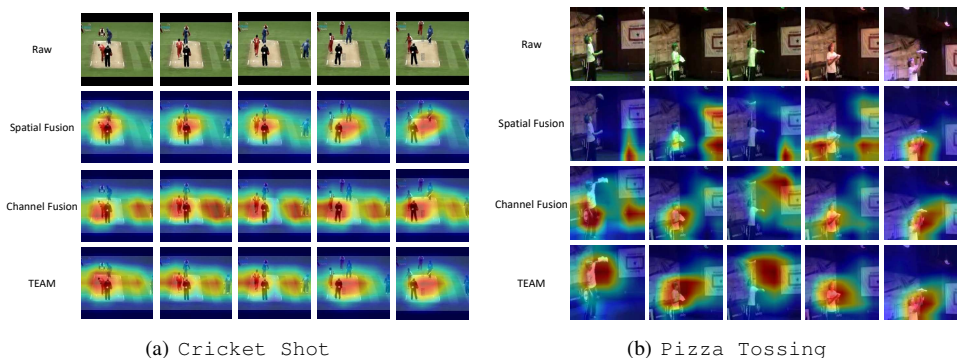


Figure 4: Class-specific visualization for channel fusion *only*, spatial fusion *only* and our proposed TEAM module.

5 Conclusion

In this work, we visit video action recognition in the partially compressed domain. We introduce a sampling strategy for inputs that comprises I-frames, MVs and residuals based on GOPs. We propose a novel TEAM module that is able to fuse three modalities effectively. The TEAM module comprises the channel fusion and the spatial fusion sequentially, and is embedded to the network in a plug-and-play manner as instantiations of TEAM-Net, which enables training an end-to-end network for three modalities. We demonstrate that TEAM-Net outperforms the baseline TSN in the raw domain and current state-of-the-arts in the partially compressed domain on all three public datasets presented in this work.

6 Acknowledgment

This work is financially supported by Science Foundation Ireland (SFI) under the Grant Number 15/RP/2776. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the NVIDIA DGX station used for this research. The authors appreciate informative and insightful comments provided by anonymous reviewers, which significantly improve the quality of this work.

References

- [1] Barak Battash, Haim Barad, Hanlin Tang, and Amit Bleiweiss. Mimic the raw domain: Accelerating action recognition in the compressed domain. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 684–685, 2020.
- [2] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6299–6308, 2017.
- [3] Yunpeng Chen, Yannis Kalantidis, Jianshu Li, Shuicheng Yan, and Jiashi Feng. Multi-fiber networks for video recognition. In *Proceedings of the european conference on computer vision (ECCV)*, pages 352–367, 2018.
- [4] V Cisco. Cisco visual networking index: Forecast and trends, 2017–2022. *White Paper*, 1:1, 2018.
- [5] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 6202–6211, 2019.
- [6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [7] Kensho Hara, Hirokatsu Kataoka, and Yutaka Satoh. Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6546–6555, 2018.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [9] Jian-Fang Hu, Wei-Shi Zheng, Jiahui Pan, Jianhuang Lai, and Jianguo Zhang. Deep bilinear learning for rgb-d action recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 335–351, 2018.
- [10] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
- [11] Yuqi Huo, Xiaoli Xu, Yao Lu, Yulei Niu, Mingyu Ding, Zhiwu Lu, Tao Xiang, and Ji-rong Wen. Lightweight action recognition in compressed videos. In *European Conference on Computer Vision*, pages 337–352. Springer, 2020.
- [12] Eddy Ilg, Nikolaus Mayer, Tonmoy Saikia, Margret Keuper, Alexey Dosovitskiy, and Thomas Brox. FlowNet 2.0: Evolution of optical flow estimation with deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2462–2470, 2017.

- [13] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. STM: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2000–2009, 2019.
- [14] Hamid Reza Vaezi Joze, Amirreza Shaban, Michael L Iuzzolino, and Kazuhito Koishida. Mmtm: multimodal transfer module for cnn fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13289–13299, 2020.
- [15] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5492–5501, 2019.
- [16] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 International conference on computer vision*, pages 2556–2563. IEEE, 2011.
- [17] Duo Li, Jie Hu, Changhu Wang, Xiangtai Li, Qi She, Lei Zhu, Tong Zhang, and Qifeng Chen. Involution: Inverting the inherence of convolution for visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12321–12330, 2021.
- [18] Jiapeng Li, Ping Wei, Yongchi Zhang, and Nanning Zheng. A slow-i-fast-p architecture for compressed video action recognition. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 2039–2047, 2020.
- [19] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. TEA: Temporal excitation and aggregation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 909–918, 2020.
- [20] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7083–7093, 2019.
- [21] Mengyuan Liu and Junsong Yuan. Recognizing human actions as the evolution of pose estimation maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1159–1168, 2018.
- [22] Yang Liu, Samuel Albanie, Arsha Nagrani, and Andrew Zisserman. Use what you have: Video retrieval using representations from collaborative experts. *arXiv preprint arXiv:1907.13487*, 2019.
- [23] Diogo C Luvizon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5137–5146, 2018.
- [24] Antoine Miech, Ivan Laptev, and Josef Sivic. Learning a text-video embedding from incomplete and heterogeneous data. *arXiv preprint arXiv:1804.02516*, 2018.
- [25] Arsha Nagrani, Samuel Albanie, and Andrew Zisserman. Seeing voices and hearing faces: Cross-modal biometric matching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8427–8436, 2018.

- [26] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *proceedings of the IEEE International Conference on Computer Vision*, pages 5533–5541, 2017.
- [27] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- [28] Zheng Shou, Xudong Lin, Yannis Kalantidis, Laura Sevilla-Lara, Marcus Rohrbach, Shih-Fu Chang, and Zhicheng Yan. Dmc-net: Generating discriminative motion cues for fast compressed video action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1268–1277, 2019.
- [29] Aljoscha Smolic, Michael Hoeyneck, and J-R Ohm. Low-complexity global motion estimation from p-frame motion vectors for mpeg-7 applications. In *Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101)*, volume 2, pages 271–274. IEEE, 2000.
- [30] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.
- [31] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, 2017.
- [32] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.
- [33] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.
- [34] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018.
- [35] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, pages 20–36. Springer, 2016.
- [36] Weiyao Wang, Du Tran, and Matt Feiszli. What makes training multi-modal classification networks hard? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12695–12705, 2020.
- [37] Zhengwei Wang, Qi She, Tejo Chalasani, and Aljosha Smolic. Catnet: Class incremental 3d convnets for lifelong egocentric gesture recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 230–231, 2020.

- [38] Zhengwei Wang, Qi She, and Aljosa Smolic. Action-net: Multipath excitation for action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021.
- [39] Zhengwei Wang, Qi She, and Tomas E Ward. Generative adversarial networks in computer vision: A survey and taxonomy. *ACM Computing Surveys (CSUR)*, 54(2):1–38, 2021.
- [40] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. *IEEE Transactions on circuits and systems for video technology*, 13(7):560–576, 2003.
- [41] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.
- [42] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R Manmatha, Alexander J Smola, and Philipp Krähenbühl. Compressed video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6026–6035, 2018.
- [43] Youngjae Yu, Sangho Lee, Gunhee Kim, and Yale Song. Self-supervised learning of compressed video representations. In *International Conference on Learning Representations*, 2021.
- [44] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with enhanced motion vector cnns. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2718–2726, 2016.
- [45] Bowen Zhang, Limin Wang, Zhe Wang, Yu Qiao, and Hanli Wang. Real-time action recognition with deeply transferred motion vector cnns. *IEEE Transactions on Image Processing*, 27(5):2326–2339, 2018.
- [46] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.
- [47] Yi Zhu, Xinyu Li, Chunhui Liu, Mohammadreza Zolfaghari, Yuanjun Xiong, Chongruo Wu, Zhi Zhang, Joseph Tighe, R Manmatha, and Mu Li. A comprehensive study of deep video action recognition. *arXiv preprint arXiv:2012.06567*, 2020.