# Long-Short Term Cross-Transformer in Compressed Domain for Few-Shot Video Classification

**Wenyang Luo**[1,2 *], **Yufan Liu**[1,2 *], **Bing Li**[1,4 †], **Weiming Hu**[1,2,3], **Yanan Miao**[5] and **Yangxi Li**[5]
[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences
[3]CAS Center for Excellence in Brain Science and Intelligence Technology
[4]PeopleAI, Inc.
[5]National Computer Network Emergency Response Technical Team/Coordination Center of China
{luowenyang2020,yufan.liu}@ia.ac.cn, {bli,wmhu}@nlpr.ia.ac.cn, miaoyn@cert.org.cn, liyangxi@outlook.com

## Abstract

Compared with image few-shot learning, most of the existing few-shot video classification methods perform worse on feature matching, because they fail to sufficiently exploit the temporal information and relation. Specifically, frames are usually evenly sampled, which may miss important frames. On the other hand, the heuristic model simply encodes the equally treated frames in sequence, which results in the lack of both long-term and short-term temporal modeling and interaction. To alleviate these limitations, we take advantage of the compressed domain knowledge and propose a long-short term Cross-Transformer (LSTC) for few-shot video classification. For short terms, the motion vector (MV) contains temporal cues and reflects the importance of each frame. For long terms, a video can be natively divided into a sequence of GOPs (Group Of Picture). Using this compressed domain knowledge helps to obtain a more accurate spatial-temporal feature space. Consequently, we design the long-short term selection module, short-term module, and long-term module to comprise the LSTC. Long-short term selection is performed to select informative compressed domain data. Long/short-term modules are utilized to sufficiently exploit the temporal information so that the query and support can be well-matched by cross-attention. Experimental results show the superiority of our method on various datasets.

## 1 Introduction

Few-shot learning (FSL), a fundamental problem in machine learning, aims to learn information about categories from a few training samples. This topic become increasingly popular because in practice collecting a large amount of labeled data is often difficult. Recently, FSL has reached a milestone on image classification [Doersch *et al.*, 2020; Sung *et al.*, 2018]. For video few-shot classification, there are still challenges due to the complicated temporal structure.

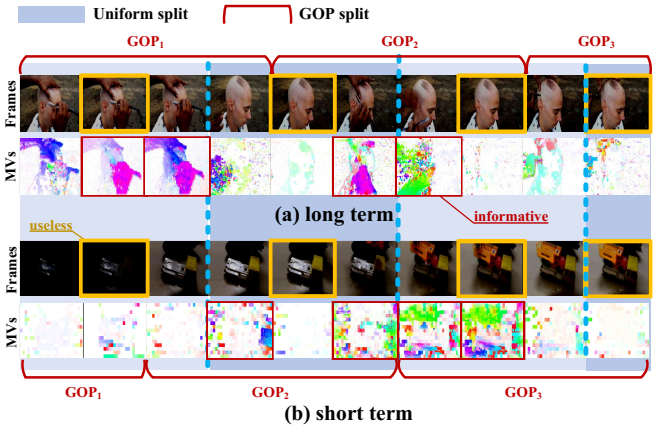


Figure 1: Video examples with long-term temporal information (a)-*ShavingHead* and short-term temporal information (b)-*PuttingSomethingOntoSomething*. The frames with yellow border are selected by traditional selection strategy, while the frames with red border are selected by our long-short term selection strategy.

Few-shot video classification [Dwivedi *et al.*, 2019; Cao *et al.*, 2020; Zhang *et al.*, 2020; Perrett *et al.*, 2021] has been tried recently, but most existing methods fail to sufficiently capture the temporal cues. On the one hand, frames are usually evenly sampled, and each frame is treated as equally important. In actuality, the information intensity of video is typically not uniformly distributed. As a result, some critical information may be missed. On the other hand, the heuristic model simply encodes the equally treated frames in sequence, which results in the lack of both long-term and short-term temporal modeling and interaction. For example, in Figure 1, the evenly sampling strategy often extracts some useless information (*e.g.*, the single head in (a) or the background in (b)). Besides, some class information reflects in the long term while others reflect in the short term. Thus, the long-short-term temporal modeling is extremely necessary.

To alleviate these problems above, we explore coarse and fine-grained temporal information and more accurate temporal relation, to better match the query and support. On the

*Equal contribution.
†Corresponding author.

one hand, more effective temporal data is used. We find that the compressed domain natively contains useful information. For example, the motion vectors (MVs) show the short-term temporal and motion information of each frame. The MV intensity can also indicate the importance of the current frame. For long term, the video is split into different groups of pictures (GOPs) according to the content, and the frames with the same scene and content are usually included in one GOP. Besides the temporal cues, the intra-frame (I-frame) of each GOP also provides the appearance information. More importantly, the compressed domain data can be acquired at a low cost. We only need to entropy-decode the video bitstream to access the compressed domain data, rather than fully decode the RGB frames.

On the other hand, a more accurate spatial-temporal feature space and temporal relation are constructed. Taking advantage of compressed domain knowledge, we propose a Long-short Term Cross-Transformer (LSTC) for few-shot video classification. Firstly, the informative data is adaptively selected from the multi-modal compressed domain data with long-short term selection. Then for the short-term temporal module, a multi-modal integration network is designed, in which the I-frames and MVs sequences from the same GOP interact with each other. Finally, the long-term temporal module performs self-attention for long-term modeling and then computes the cross-attention between tuples of query embeddings and those of support for more sufficient matching. The prototype for each support class is generated. The label is predicted by the distance between the query and each prototype.

Our main contributions are summarized as follows:

- We proposed a novel framework called long-short term cross-transformer (LSTC) for few-shot video classification, which can deeply exploit the long-short term video information and well match the query-support pair.
- We take advantage of the compressed domain knowledge to obtain effective spatial-temporal information at a low cost and adaptively select the informative data to be processed.
- Experimental results show that our method is effective and outperforms the state-of-the-art (SOTA) on both large-scale and small-scale datasets, including SSV2 [Goyal *et al.*, 2017], Kinetics [Carreira and Zisserman, 2017], UCF [Soomro *et al.*, 2012] and HMDB [Kuehne *et al.*, 2011].

## 2 Related Work

**Few Shot Video Classification.** Few-shot learning (FSL) addresses the challenging problem of learning from a few labeled examples. Videos reside in a higher-dimensional space than images, which increases the difficulty to learn a strong classifier with limited samples. Compound Memory Network (CMN) [Zhu and Yang, 2018] constructs a two-layer compound memory structure to store video features for matching. Temporal Attentive Relation Network (TARN) [Bishay *et al.*, 2019] performs segment-wise alignment before matching support and query videos with relation network [Sung *et al.*, 2018]. EOSVR [Fu *et al.*, 2019] proposes embodied one-shot video recognition with synthetic data. OTAM [Cao *et al.*, 2020] aligns videos with differentiable dynamic programming. Temporal Cross-Transformers (TRX) [Perrett *et al.*, 2021] adapt Cross-Transformers [Doersch *et al.*, 2020] to constructs query-specific prototypes from tuples of frames. The problem of video embedding is discussed in [Zhu *et al.*, 2021]. The previous methods attempt to capture the temporal structure of videos but achieve limited success. A critical problem is that their input is RGB frames, where temporal clues must be inferred indirectly and often implicitly. AMeFu-Net[Fu *et al.*, 2020] exploits depth as additional input. In contrast, our method considers compressed domain data that contains direct temporal information, exploiting global and local temporal structure with a long-short term model that better suits the new input modality.

**Compressed Video Classification.** Compressed domain data provides simple and fast temporal information, which can improve the performance of video classification methods with limited overhead. The acquirement of compressed domain information is nearly cost-free compared with conventional methods that fully decode the input videos. Moreover, MVs contain the movement at the block level so they can serve as coarse motion estimation. CoViAR [Wu *et al.*, 2018] pioneers compressed video classification, replacing optical flow in [Wang *et al.*, 2016] with MVs and residuals. DMC-Net [Shou *et al.*, 2019] refines MVs with the supervision of optical flow. Slow-I-Fast-P [Li *et al.*, 2020] establishes pseudo optical flow from MVs and residuals for the fast path of SlowFast Network [Feichtenhofer *et al.*, 2019]. These methods reveal the potency and efficiency of compressed domain features for video classification. Nonetheless, they treat compressed domain features as insertion into conventional architecture instead of developing a new befitting compressed video classification framework.

## 3 The Proposed Method

We propose a novel Long-Short Term Cross-Transformer (LSTC) for few-shot video classification. It takes advantage of the compressed domain knowledge and performs accurate query-support matching with a long-short term structure. Here, we introduce the proposed method, including the framework, formulation, and technical details.

### 3.1 Framework and Formulation

The overall framework is summarized in Figure 2. We first extract the compressed domain data by partially decoding the video bitstream at a low cost. Specifically, the I-frames and the motion vectors (MVs) from different Groups of Pictures (GOPs) are obtained. We denote the frames in the $t$-th GOP as $\{\mathbf{G}_{t,l}\}_{l=0}^{L_t}$, where $\mathbf{G}_{t,0}$ is the I-frame. The MVs are denoted as $\mathbf{M}_{t,l}$. The I-frames are essentially RGB images, containing the appearance information. The MVs denote the motion from the source positions in the previous frame to the destination positions in the current frame, containing the temporal information. The I-frames and MVs are extracted from the video stream by entropy decoding.

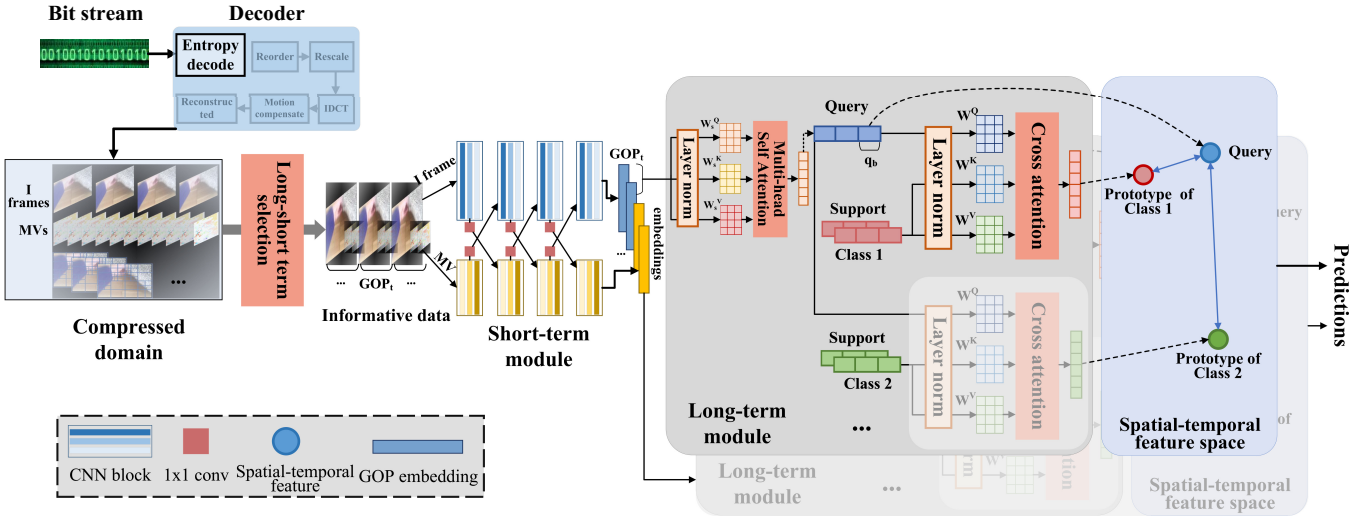


Figure 2: Overall framework of the proposed method, which contains compressed domain information extraction, long-short term selection and long-short term modules.

Secondly, we design a long-short term selection module to extract the informative I-frames and MVs from long-term GOP-level and short-term frame-level. After that, a short-term module is embedded to exploit the short-term temporal information and fuse the multiple modals of I-frames and MVs at GOP level. Fed with the embeddings from the short-term module, a cross-transformer is constructed to build the long-term temporal information and to explore the relationship between the support and the query.

In this paper, we consider the C-way K-shot few-shot video classification problem, in which an episode [Vinyals *et al.*, 2016] consists of $C$ classes with $K$ support videos for each class. The target of this problem is to classify the query video into one of the classes $c \in \{1, 2, \cdots, C\}$.

## 3.2 Long-short Term Selection

The input frame selection is an important problem for video classification. Usually frames are randomly sampled either evenly or successively with a fixed stride [Wang *et al.*, 2016; Cao *et al.*, 2020; Perrett *et al.*, 2021]. This selection strategy may miss some keyframes and introduce some irrelevant background frames. In our method, we utilize the compressed domain to conduct long-short term information selection.

There are useful cues to distinguish the informative frames in the compressed domain. In the long term, the GOPs with different lengths have already divided the contents into several parts. In the short term, the MVs reflect the importance of the current action. Hence, we define a metric to evaluate the importance of the $l$-th frame in the $t$-th GOP:

$$I(\mathbf{M}_{t,l}) = \left( \frac{1}{|\mathbf{G}_{t,l}|} \sum_{(x,y)\in\mathbf{G}_{t,l}} ||\mathbf{M}_{t,l}(x,y)||_1 \right)^{\alpha}, \quad (1)$$

where $|\cdot|$ is the cardinality of a set, $||\cdot||_1$ is the $L_1$ norm, $(x, y)$ is spatial location, and $\alpha \geq 0$. Based on this metric, we obtain the importance score of the $t$-th entire GOP:

$$I(\mathrm{GOP}_t) = \frac{1}{L_t} \sum_{l=1}^{L_t} I(\mathbf{M}_{t,l}), \quad (2)$$

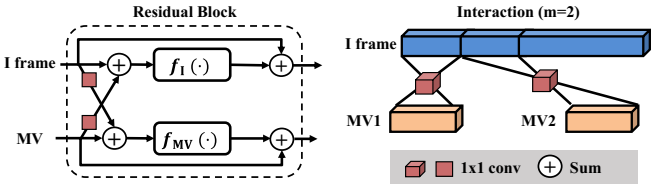


Figure 3: Short-term module (STM) performs short-term modeling in a GOP. Left illustrates a modified residual block, which replaces the first block at each stage. Right is the lateral connections in the modified residual block, implemented by grouped conv1 $\times$ 1.

According to this importance score, in the long term, we extract the informative GOPs with probabilities of $\{P_t^{\mathrm{sel}} \propto I(\mathrm{GOP}_t)\}_{t=1}^{T}$. Likewise, in the short term, we extract the informative MVs with probabilities of $\{P_l^{\mathrm{sel}} \propto I(\mathrm{M}_{t,l})\}_{l=1}^{L_t}$. We sample them with a probability rather than select the fixed top-n frames because the fuzzification in the proposed selection scheme can tolerate some noise in these frames. This alleviates the influence of some outliers and noise on the performance. $g$ GOPs are sampled for each video, and the I-frame and $m$ additional frames (P-frames) are selected for each sampled GOP. We then accumulate the MVs and conduct an alignment between the accumulated MVs and the I-frames. The detailed process is described in the Supplementary Materials. The selected I-frames and aligned accumulated MVs are used as input to the Long-short Term Cross-Transformer.

## 3.3 Long-short Term Cross-Transformer

### Short-Term Module

We conduct short-term temporal interaction in each GOP using the designed short-term module (STM). The embeddings generated by STM are enhanced with local temporal relation, providing better appearance and motion modeling than features from RGB frames only. Compressed domain natively decouples the original video into appearance part and motion part (*i.e.*, I-frames and MVs). Hence, we construct a two-branch network, one for appearance modeling (I-branch, *i.e.*, $f_{\mathrm{I}}(\cdot)$), and the other for motion modeling (MV-branch, *i.e.*,

$f_{\mathrm{MV}}(\cdot)$). Each branch employs a staged convolutional neural network (CNN) as the backbone and extracts features for each frame. Between these two branches, they interact with each other at each stage $r$ (*i.e.*, $r = 1, 2, ..., R$). The interaction is performed between the I-frame feature maps and the MV feature maps from the same GOP.

In our experiments, we use ResNet [He *et al.*, 2016] as the backbone. The appearance-motion interaction is performed at the first residual blocks of stages Conv2-Conv5. As depicted in Figure 3, the feature maps of each branch are encoded with conv$1 \times 1$ and added to the residual path of the other branch. Only a proportion $p \in [0, 1]$ of I feature channels interact with MV features so that STM can preserve the original appearance in the other $(1 - p)$ I-branch channels. All MV features participate in the interaction. The interacting I feature channels are evenly divided into $m$ groups, each of which exchanges information with the MV features of a P-frame by lateral connections. STM is applied independently for each GOP. Finally, for each video, the output features of all the I-frames at I-branch are stacked as $\mathbf{Z}_I \in \mathbb{R}^{g \times d_I}$, and the output features of all the P-frames at MV-branch are stacked as $\mathbf{Z}_{MV} \in \mathbb{R}^{gm \times d_{MV}}$, where $d_I$ and $d_{MV}$ are the output dimensions of I-branch and MV-branch, respectively.

**Long-Term Module**

After obtaining the embeddings, we match the query and support videos with long-term module (LTM). LTM consists of two Cross-Transformers, one for the appearance embeddings from the I-branch and one for the motion embeddings from the MV-branch. For each Cross-Transformer, a self-attention layer is first adopted to the embeddings $\mathbf{Z} \in \{\mathbf{Z}_I, \mathbf{Z}_{MV}\}$ of each video. Given query/key/value matrices $\mathbf{W}_S^Q, \mathbf{W}_S^K, \mathbf{W}_S^V \in \mathbb{R}^{d \times d}$, where $d \in \{d_I, d_{MV}\}$ is the column dimension of $\mathbf{Z}$, the self-attention is calculated as:

$$\mathbf{H} = \mathbf{Z} + softmax\left(\frac{\mathbf{Z}\mathbf{W}_S^Q\left(\mathbf{Z}\mathbf{W}_S^K\right)^T}{\sqrt{d}}\right)\mathbf{Z}\mathbf{W}_S^V, \quad (3)$$

Position encoding [Vaswani *et al.*, 2017] is applied on both $\mathbf{Z}$ and $\mathbf{H}$. After self-attention, we construct embeddings for tuples of length $n$ from each video's embeddings $\mathbf{H}$, allowing for fine-grained matching. For the query video, the tuple embedding $\mathbf{q_b}$ is constructed as following:

$$\begin{aligned} &\mathbf{q_b} = [\mathbf{h}_{j_1} \oplus \mathbf{h}_{j_2} \oplus, ..., \oplus \mathbf{h}_{j_n}] \in \mathbb{R}^{nd}, \\ &s.t. \quad \mathbf{b} = \{j_1, ..., j_n\} \end{aligned} \quad (4)$$

in which $\oplus$ denotes vector concatenation, $\{\mathbf{h}_{j_1}, \cdots, \mathbf{h}_{j_n}\}$ are $n$ different row vectors from $\mathbf{H}$. Similarly, the tuple embeddings of the support videos are obtained in the same way. Then for each class $c$, we stack all possible tuple embeddings from every shot as row vectors to obtain the class representation $\mathbf{S}^c$. Given query matrix $\mathbf{W}^Q \in \mathbb{R}^{nd \times d_k}$, key matrix $\mathbf{W}^K \in \mathbb{R}^{nd \times d_k}$ and value matrix $\mathbf{W}^V \in \mathbb{R}^{nd \times d_v}$, where $d_k, d_v$ are hidden dimensions, the support prototype of class $c$ is obtained by cross-attention:

$$\mathbf{u}_{\mathbf{b},c} = softmax\left(\frac{\mathbf{q_b}\mathbf{W}^Q\left(\mathbf{S}^c\mathbf{W}^K\right)^T}{\sqrt{d_k}}\right)\mathbf{S}^c\mathbf{W}^V, \quad (5)$$

Subsequently, the distance between the query and the $c$-th class support is computed:

$$\delta_{\mathbf{b},c} = ||\mathbf{q_b}\mathbf{W}^V - \mathbf{u}_{\mathbf{b},c}||_2^2, \quad (6)$$

We extract all the possible tuples of the query video to match the support videos. The average distance is calculated:

$$\bar{\delta}_c = \frac{1}{|\mathbf{b}|}\sum_{\mathbf{b}} \delta_{\mathbf{b},c}, \quad (7)$$

The predicted result is the class with the smallest distance between the support and the query.

### 3.4 Optimization

To train and optimize the proposed Long-short Term Cross-Transformer, we minimize the distance between the query video and the matched support video. In the loss function, negative distances are regraded as *logits* to compute the cross-entropy loss:

$$Loss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{C} h_{c,i} \log\left(\frac{\exp\left(-\bar{\delta}_{c,i}\right)}{\sum_{c'=1}^{C}\exp\left(-\bar{\delta}_{c',i}\right)}\right). \quad (8)$$

where $h_{c,i}$ denotes the label of the $i$-th training sample.

## 4 Experiments

### 4.1 Settings

**Datasets.** Our evaluations are conducted on four datasets, including Kinetics [Carreira and Zisserman, 2017], Something-Something V2 (SSV2) [Goyal *et al.*, 2017], UCF [Soomro *et al.*, 2012] and HMDB [Kuehne *et al.*, 2011]. The first few shot video classification dataset is constructed from Kinetics by CMN [Zhu and Yang, 2018]. In their setup 100 classes are sampled from the total 400 classes. Selected classes are then split into train/val/test sets of 64/12/24 classes without overlapping, and 100 videos are sampled for each class. On SSV2 we evaluate our method with the split proposed by [Cao *et al.*, 2020] which contains 64/12/24 classes for train/val/test with approximately 1000 videos for each class. Following [Zhang *et al.*, 2020], we use 70/10/21 classes as train/val/test set for UCF and 31/10/10 for HMDB, respectively. All reported results are measured over 10,000 randomly sampled episodes from testing sets.

**Implementation.** MPEG-4 encoded videos are used as input. We sample $g = 4$ GOPs from each video and extract the corresponding I-frames and the MVs of $m = 2$ P-frames. When the video has no enough GOPs or P-frames, the GOPs and P-frames may be sampled several times. The horizontal and vertical components of MVs are rescaled respectively by the width and height of the video. For the short-term module, we use ResNet-50 [He *et al.*, 2016] as the backbone for I-branch and ResNet-18 for MV-branch, both initialized with ImageNet [Deng *et al.*, 2009] pre-trained weights. We set $\alpha = 0.4$ in long-short term selection. For the LSTC, we set $p = 0.5$, the latent dimensions are set to $d_k = d_v = 1024$ and $d_k = d_v = 512$ for I and MV, respectively. The two Cross-Transformers in the long-term module are implemented with $n = 2$ and $n = 3$, respectively. Their predictions are merged

| Method | SSV2 | | HMDB | | UCF | | Kinetics | |
|---|---|---|---|---|---|---|---|---|
| | 1-shot | 5-shot | 1-shot | 5-shot | 1-shot | 5-shot | 1-shot | 5-shot |
| MatchNet† [Vinyals *et al.*, 2016] | - | - | 56.2 | 66.9 | 78.0 | 87.9 | 67.8 | 77.1 |
| MAML [Finn *et al.*, 2017] | - | - | - | - | - | - | 54.2 | 75.3 |
| CMN† [Zhu and Yang, 2018] | - | - | 45.6 | 62.6 | 71.8 | 88.7 | 60.5 | 78.9 |
| TARN [Bishay *et al.*, 2019] | - | - | - | - | - | - | 64.8 | 78.5 |
| ARN [Zhang *et al.*, 2020] | - | - | 45.5 | 60.6 | 66.3 | 83.1 | 63.7 | 82.4 |
| OTAM† [Cao *et al.*, 2020] | 42.8 | 52.3 | 55.0 | 65.7 | 80.1 | 89.8 | 73.0 | 85.8 |
| TRX [Perrett *et al.*, 2021] | - | 64.6 | - | 75.6 | - | 96.1 | - | 85.9 |
| **Ours** | **46.7** | **66.7** | **60.9** | **76.8** | **85.7** | **96.5** | **73.4** | **86.5** |

Table 1: **Comparison with state-of-the-art models**. Table entries are top-1 accuracy (%) on each dataset, with 1-shot and 5-shot setting. The performance margin is larger on SSV2, but lesser on Kinetics and UCF which rely more on appearance and scene inference. Methods with † are evaluated with our implementations when reported results are not available in the literature.

by averaging. During training, the I-frames and MVs are randomly cropped to 224×224. During testing, they are resized to height 256 before center cropping. Following [Wu *et al.*, 2018], we augment I-frames with random color jittering.

The whole network is trained using SGD optimizer [Bottou, 2010] with the learning rate of 0.025. Cross-entropy loss is calculated on a batch of 32 episodes. The training continues for 60,000 episodes on SSV2, and 10,000 episodes on other datasets. We use 4 NVIDIA RTX2060 GPUs for training.

## 4.2 Performance Comparison

To verify the effectiveness of the proposed method, 7 SOTAs are taken into account for comparison, including MatchNet [Vinyals *et al.*, 2016], MAML [Finn *et al.*, 2017], CMN [Zhu and Yang, 2018], TARN [Bishay *et al.*, 2019], OTAM [Cao *et al.*, 2020], ARN [Zhang *et al.*, 2020] and TRX [Perrett *et al.*, 2021]. The comparative results are summarized in Table 1. The proposed method achieves SOTA performance on all four benchmark datasets. For the 5-shot setting, the proposed method surpasses the previous SOTA by 2.1%, 1.2% respectively on SSV2 and HMDB. Note that the gains come with little overhead. Although MVs are introduced as an additional stream to I-frames, our elaborate design of short-term modeling module reduces the convolutional feature dimension by a factor of $m$, which greatly decreases the number of parameters of the Cross-Transformer. Furthermore, the acquirement of I-frames and MVs is more efficient than traditional fully decoding, because, in the extraction of compressed domain data, most of the computation-intensive decoding steps can be skipped. Compared with prior SOTA methods which typically take 8 RGB frames as input, the proposed method is fed with only 4 I-frames in a video, which contain much less appearance information than 8 RGB frames. Therefore, the major contribution to the SOTA performance of LSTC is the effective utilization of temporal information contained in MVs and the appearance information in I-frames.

## 4.3 Ablation Study

In this section, we conduct extensive experiments to further analyze and discuss the effectiveness of the proposed method. The results of ablation study on constituent parts of our method are summarized in Table 2. Baseline model consists of I-frames and MVs streams with traditional selection strategy, *i.e.*, GOPs and MVs are uniformly sampled.

| MVs | Selection | STM | LTM | HMDB | UCF |
|---|---|---|---|---|---|
| ✓ | | | | 69.2 | 89.6 |
| ✓ | ✓ | | | 69.9 | 90.4 |
| | | ✓ | ✓ | 73.5 | 94.1 |
| | ✓ | ✓ | ✓ | 73.7 | 94.0 |
| ✓ | ✓ | ✓ | | 74.2 | 94.4 |
| ✓ | ✓ | | ✓ | 74.9 | 95.0 |
| ✓ | | ✓ | ✓ | 75.3 | 95.5 |
| ✓ | ✓ | ✓ | ✓ | **76.8** | **96.5** |

Table 2: **Ablation study**. Selection means long-short term selection, STM means short-term module, LTM means long-term module. Results are reported on 5-way 5-shot setting.

The baseline model replaces the short-term module with plain ResNet backbones, without interaction between two streams. The long-term module is removed and the maximal cosine similarity between frames is used to measure the distance between two videos. In addition, we provide analysis on $g$, $m$, and few-shot settings, and compare compressed domain data with optical flows. See the Supplementary Materials for more quantitative and qualitative results.

**The Effect of Motion Vectors and Frame Selection**

We argue that densely sampled RGB frames are highly redundant for video classification. In the traditional setting [Perrett *et al.*, 2021; Cao *et al.*, 2020] 8 frames are drawn from the original video to construct a video representation. In our method, merely 4 I-frames containing RGB information are selected, and MVs provide additional motion information for accurate classification. As demonstrated in Table 2, utilizing I-frames and MVs already yields a few-shot video classifier with considerable accuracy. The performance is future improved with the introduction of long-shot term selection, which utilizes the imbalanced distribution of informative frames and the correlation between MVs and the significance of motion. Moreover, after removing the selection, a 1.5% drop and a 1.0% drop are observed on HMDB and UCF, respectively. This proves the effectiveness of the proposed long-short term selection procedure.

Although a few-shot classifier based on MVs and long-short term selection is a strong baseline, the performance of such a naive combination is far from satisfactory. The naive approach fails to operate on I-frames and MVs jointly and derive enough information. To achieve SOTA performance, we

| | $\mathbf{g=1}$ | $\mathbf{g=2}$ | $\mathbf{g=4}$ | $\mathbf{g=8}$ |
|---|---|---|---|---|
| $\mathbf{m=1}$ | 65.4 | 71.1 | 74.2 | 75.8 |
| $\mathbf{m=2}$ | 68.0 | 72.9 | 76.8 | 77.6 |
| $\mathbf{m=4}$ | 68.4 | 73.3 | 77.1 | 78.1 |
| $\mathbf{m=8}$ | 68.2 | 73.9 | 77.5 | 78.4 |

Table 3: **The effect of number of sampled GOPs and P-frames per GOP**. $g$ is the number of sampled GOP, $m$ is the number of additionally sampled frames (P-frames) per sampled GOP. Results are accuracy(%) on HMDB with 5-way 5-shot setting.

must design an elaborate method to fuse them effectively.

**The Effect of STM and LTM**

In our method, we propose short-term module (STM) and long-term module (LTM) to facilitate the fusion of appearance encoded by I-frames and motion encoded by MVs. In Table 2, results demonstrate the considerable improvement with the introduction of STM and LTM. With STM, the accuracy improves by 4.3% on HMDB and 4.0% on UCF compared with the baseline model with MV and long-short term selection only. LTM brings an improvement of 5.0% and 4.6%, respectively. These results prove that our STM and LTM design can bridge the gap between two modalities of appearance and motion, generating distinctive features from these two streams. Moreover, when MVs in the full method are replaced by all-zero input, *i.e.*, no MVs are provided, the performance dropped significantly by 2.9% and 2.5% on HMDB and UCF, respectively. The ability of STM and LTM to dig and utilize distinctive video features is considerably impaired without MVs.

**Varying the Number of Sampled GOPs and P-frames**

The numbers of sampled GOPs ($g$) and P-frames per GOP ($m$) are both hyperparameters in our method. We choose $g = 4$ and $m = 2$ in previous experiments. This setting allows us to make a fair comparison with previous methods which typically take 8 RGB frames as input. To demonstrate the scalability of our method, we summarize the performance on HMDB with different $g$ and $m$ in Table 3. All other settings are the same as section 4.1. Accuracy is generally improved with more GOPs and P-frames per GOP, but the computational cost also increases with more I-frames and MVs to process. Since the numbers of GOPs and P-frames in each GOP have upper bounds for videos in a certain dataset, sampling more GOPs or more P-frames per GOP than those existing in the video is pointless. A practical choice for $g$ and $m$ depends on the average length of videos, encoding, and computation ability.

**Performance with Different Few-Shot Settings**

Table 4 shows the impact of the few-shot setting, namely the number of ways ($C$) and shots ($K$), on performance. In general, our method performs better with more shots, *i.e.* more "hints", and fewer ways, *i.e.* fewer "possibilities". The same phenomenon is observed in previous works [Zhu and Yang, 2018; Bishay *et al.*, 2019; Perrett *et al.*, 2021] and is likely ubiquitous in few shot video classification. Nevertheless, such a problem may not be acute in the real world. In practice, usually more than a few samples could be retrieved for a well-defined category, albeit still considered "few-shot" as opposed to the traditional many-shot scenario which requires thousands of training samples.

| | **2-way** | **3-way** | **4-way** | **5-way** |
|---|---|---|---|---|
| **1-shot** | 78.4 | 68.5 | 62.1 | 60.9 |
| **2-shot** | 83.4 | 77.0 | 69.7 | 66.8 |
| **3-shot** | 88.0 | 82.1 | 75.0 | 71.2 |
| **4-shot** | 89.2 | 83.1 | 78.3 | 73.7 |
| **5-shot** | 90.9 | 84.5 | 82.2 | 76.8 |

Table 4: **The impact of few shot setting**. Rows vary in the number of ways, columns vary in the number of shots. Results are accuracy(%) on HMDB.

| **Method** | **Data Time** | **Infer. Time** | **Accuracy** |
|---|---|---|---|
| Optical Flow | 25.0 | 14.6 | 74.2 |
| **MVs** | **0.7** | 14.6 | **76.8** |

Table 5: **Comparison with optical flow**. Data Time is preprocessing time and Infer. Time is inference time, both measured in ms/frame. Accuracy(%) is reported on HMDB with 5-way 5-shot setting.

**Comparison with Optical Flow**

In Table 5 we replace MVs in our method with optical flows and compare the performance. The optical flows are extracted and processed following [Wang *et al.*, 2016]. All other parts of our method remain the same. Data time, inference time, and accuracy on HMDB are reported. As demonstrated by the results, extracting compressed domain data is by orders of magnitude faster than calculating optical flows, saving a significant amount of time and computing resources. On the other hand, the performance of the optical flow-based method is worse than the MV-based method. This may be caused by the fact that the optical flows lack the required GOP structure, which indicates that the compressed domain data is not a simple replacement of optical flows in our method.

## 5 Conclusion

This paper proposes a novel framework called Long-short Term Cross-Transformer (LSTC) for few-shot video classification. It takes advantage of compressed domain data to match the query and the support videos. In particular, LSTC consists of long-short term selection, short-term module, and long-term module. The long-short term selection adaptively selects and extracts the informative data by analyzing the implicit cues in the compressed domain. The short-term module integrates the multi-modal compressed domain data (*i.e.*, I-frames and MVs) and makes them interact with each other, to obtain fine-grained spatial-temporal features. Given these short-term embeddings, the long-term module computes the global temporal representations and cross-attention between the query and support. Experiments show the effectiveness of the proposed method on various datasets.

## Acknowledgments

This work was supported by the National Key Research and Development Program of China (Grant No.

2020AAA0106800), the Natural Science Foundation of China (Grant No.61902401, No. 62192785, No. 61972071, No. U1936204, No. 62122086, No. 62036011, No. 62192782, No. 61721004 and No. 61906052), the Beijing Natural Science Foundation No. M22005, the CAS Key Research Program of Frontier Sciences (Grant No. QYZDJ-SSW-JSC040). The work of Bing Li was also supported by the Youth Innovation Promotion Association, CAS.

## References

[Bishay *et al.*, 2019] Mina Bishay, Georgios Zoumpourlis, and Ioannis Patras. TARN: Temporal Attentive Relation Network for Few-Shot and Zero-Shot Action Recognition. In *BMVC*, 2019.

[Bottou, 2010] Léon Bottou. Large-Scale Machine Learning with Stochastic Gradient Descent. In *COMPSTAT*, 2010.

[Cao *et al.*, 2020] Kaidi Cao, Jingwei Ji, Zhangjie Cao, Chien-Yi Chang, and Juan Carlos Niebles. Few-Shot Video Classification via Temporal Alignment. In *CVPR*, 2020.

[Carreira and Zisserman, 2017] João Carreira and Andrew Zisserman. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In *CVPR*, 2017.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009.

[Doersch *et al.*, 2020] Carl Doersch, Ankush Gupta, and Andrew Zisserman. CrossTransformers: Spatially-Aware Few-Shot Transfer. In *NeurIPS*, 2020.

[Dwivedi *et al.*, 2019] Sai Kumar Dwivedi, Vikram Gupta, Rahul Mitra, Shuaib Ahmed, and Arjun Jain. ProtoGAN: Towards Few Shot Learning for Action Recognition. In *ICCV Workshop*, 2019.

[Feichtenhofer *et al.*, 2019] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast Networks for Video Recognition. In *ICCV*, 2019.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *ICML*, 2017.

[Fu *et al.*, 2019] Yuqian Fu, Chengrong Wang, Yanwei Fu, Yu-Xiong Wang, Cong Bai, Xiangyang Xue, and Yu-Gang Jiang. Embodied one-shot video recognition: Learning from actions of a virtual embodied agent. In *ACM Multimedia*, 2019.

[Fu *et al.*, 2020] Yuqian Fu, Li Zhang, Junke Wang, Yanwei Fu, and Yu-Gang Jiang. Depth guided adaptive meta-fusion network for few-shot video recognition. In *ACM Multimedia*, 2020.

[Goyal *et al.*, 2017] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fründ, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "Something Something" Video Database for Learning and Evaluating Visual Common Sense. In *ICCV*, 2017.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *CVPR*, 2016.

[Kuehne *et al.*, 2011] Hilde Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. HMDB: A large video database for human motion recognition. In *ICCV*, 2011.

[Li *et al.*, 2020] Jiapeng Li, Ping Wei, Yongchi Zhang, and Nanning Zheng. A Slow-I-Fast-P Architecture for Compressed Video Action Recognition. In *ACM Multimedia*, 2020.

[Perrett *et al.*, 2021] Toby Perrett, Alessandro Masullo, Tilo Burghardt, Majid Mirmehdi, and Dima Damen. Temporal-Relational CrossTransformers for Few-Shot Action Recognition. In *CVPR*, 2021.

[Shou *et al.*, 2019] Zheng Shou, Zhicheng Yan, Yannis Kalantidis, Laura Sevilla-Lara, Marcus Rohrbach, Xudong Lin, and Shih-Fu Chang. DMC-Net: Generating Discriminative Motion Cues for Fast Compressed Video Action Recognition. In *CVPR*, 2019.

[Soomro *et al.*, 2012] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. UCF101: A Dataset of 101 Human Actions Classes From Videos in The Wild. *ArXiv*, 2012.

[Sung *et al.*, 2018] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. Learning to Compare: Relation Network for Few-Shot Learning. In *CVPR*, 2018.

[Vaswani *et al.*, 2017] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is All you Need. In *NeurIPS*, 2017.

[Vinyals *et al.*, 2016] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching Networks for One Shot Learning. In *NeurIPS*, 2016.

[Wang *et al.*, 2016] 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 *ECCV*, 2016.

[Wu *et al.*, 2018] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R. Manmatha, Alex Smola, and Philipp Krähenbühl. Compressed Video Action Recognition. In *CVPR*, 2018.

[Zhang *et al.*, 2020] Hongguang Zhang, Li Zhang, Xiaojuan Qi, Hongdong Li, Philip H. S. Torr, and Piotr Koniusz. Few-Shot Action Recognition with Permutation-Invariant Attention. In *ECCV*, 2020.

[Zhu and Yang, 2018] Linchao Zhu and Yi Yang. Compound Memory Networks for Few-Shot Video Classification. In *ECCV*, 2018.

[Zhu *et al.*, 2021] Zhenxi Zhu, Limin Wang, Sheng Guo, and Gangshan Wu. A Closer Look at Few-Shot Video Classification: A New Baseline and Benchmark. In *BMVC*, 2021.