Revisiting the Spatial and Temporal Modeling for Few-shot Action Recognition

Jiazheng Xing, Mengmeng Wang*, Boyu Mu ,Yong Liu†

Zhejiang University, Hangzhou, China
{jiazhengxing,mengmengwang, muboyu}@zju.edu.cn
yongliu@iipc.zju.edu.cn

Abstract

Spatial and temporal modeling is one of the most core aspects of few-shot action recognition. Most previous works mainly focus on long-term temporal relation modeling based on high-level spatial representations, without considering the crucial low-level spatial features and short-term temporal relations. Actually, the former feature could bring rich local semantic information, and the latter feature could represent motion characteristics of adjacent frames, respectively. In this paper, we propose SloshNet, a new framework that revisits the spatial and temporal modeling for few-shot action recognition in a finer manner. First, to exploit the low-level spatial features, we design a feature fusion architecture search module to automatically search for the best combination of the low-level and high-level spatial features. Next, inspired by the recent transformer, we introduce a long-term temporal modeling module to model the global temporal relations based on the extracted spatial appearance features. Meanwhile, we design another short-term temporal modeling module to encode the motion characteristics between adjacent frame representations. After that, the final predictions can be obtained by feeding the embedded rich spatial-temporal features to a common frame-level class prototype matcher. We extensively validate the proposed SloshNet on four few-shot action recognition datasets, including Something-Something V2, Kinetics, UCF101, and HMDB51. It achieves favorable results against state-of-the-art methods in all datasets.

Introduction

With the development of deep learning, a large amount of excellent work has emerged in the field of action recognition (Li et al. 2022a; Liu et al. 2022b; Wang et al. 2022; Feichtenhofer et al. 2019; Wang et al. 2018). Most studies use large amounts of labeled data to perform video understanding or classification tasks to learn video representations. Such approaches are unsatisfactory in industrial applications because of the massive time-consuming and labor-consuming data annotation. On the contrary, the core assumption of few-shot learning is using only a handful of labeled training samples from numerous similar tasks as a surrogate for large quantities of labeled training samples.

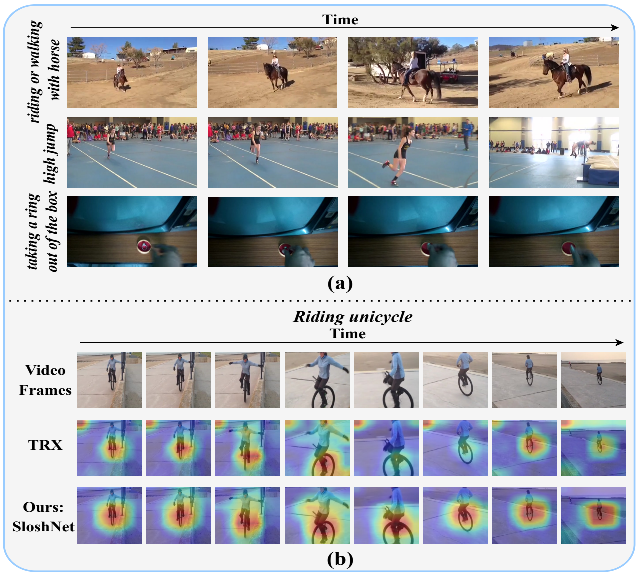

Figure 1: (a): Some examples in few-shot action recognition (b): Visualization of the attention map from the recent work TRX (Perrett et al. 2021) and our proposed SloshNet.

Therefore, the attention on few-shot learning methods is increasing daily. The task of few-shot action recognition aims to classify an unlabeled query video into one of the action categories in the support set (usually five categories) with limited samples per action class.

Inspired by few-shot image recognition (Finn, Abbeel, and Levine 2017; Doersch, Gupta, and Zisserman 2020; Elsken et al. 2020; Ma et al. 2020), existing few-shot video action recognition methods mainly focus on comparing the similarity of different videos in the feature space for recognition. However, videos have an extra temporal dimension compared to images, so that it is insufficient to represent the whole video as a single feature vector. Therefore, the spatial-temporal feature modeling becomes one of the core problems of few-shot action recognition. Specially, spatial feature aims to express spatial semantic information for every single frame. In some cases, a video could be recognized with only a single frame like the example of the first row in Fig. 1(a). Current approaches (Bishay, Zoumpourlis,

[*] Co-corresponding author.

[†] Corresponding author.

and Patras 2019; Kumar and Narang 2021; Li et al. 2022b) usually extract the spatial features through a TSN (Wang et al. 2016) model. However, they usually consider the high-level spatial features as default but ignore the evenly crucial low-level spatial features, which contain rich texture information. Fusing the low-level spatial features with the high-level ones could compensate and even highlight the low-level semantic features. For the temporal features, we classify them into two categories, long-term and short-term temporal features. Long-term temporal features present the relationship between spatial appearance features of different timestamps, which has also been a hot topic in previous works. For instance, the action of "high jump" in Fig. 1(a) is easily mistaken for "running" if the feature of jumping into the mat in the last frame is not integrated into all the previous features. Existing methods (Zhu and Yang 2018; Cao et al. 2020; Perrett et al. 2021) model the long-term temporal features mainly through hand-designed temporal alignment algorithms during the class prototype construction process, which aims to obtain better global features for comparison. On the other hand, short-term temporal features represent the motion characteristics of adjacent frames, i.e., focus on the local temporal relation modeling. For example, in Fig. 1(a), without the short-term temporal information, it is hard to classify whether the action is "taking the ring out of the box" or "putting the ring in the box". Nevertheless, we have observed that the short-term temporal modeling remains unexplored for the few-shot action recognition task.

The critical insight of our work is to provide powerful spatial-temporal features, making it possible to realize effective recognition with a common frame-level class prototype matcher for few-shot action recognition. To this end, we propose a novel method for few-shot action recognition, dubbed **SloshNet**, a short for **S**patial, **lo**ng-term temporal and **sh**ort-term temporal features integrated **Net**work. Specifically, to exploit the low-level spatial features, we first design a feature fusion architecture search module (FFAS) to automatically search for the best fusion structure of the low-level and high-level spatial features in different scenarios. Low-level features focus more on texture and structural information, while high-level features focus more on the semantic information, and their combination can enhance the representation of spatial features. Furthermore, based on the extracted spatial appearance features, we introduce a long-term temporal modeling module (LTMM) to model the global temporal relations. Meanwhile, we design another short-term temporal modeling module (STMM) to encode the motion characteristics between the adjacent frame representations and explore the optimal integration of long-term and short-term temporal features. For class prototype matcher, we follow a frame-level method TRX (Perrett et al. 2021), using an attention mechanism to match each query sub-sequence with all sub-sequences in the support set and aggregates this evidence. Fig. 1(b) shows the learned attentions of our SloshNet with TRX, where the attention learned by our method is highly concentrated and more correlated with the action subject, demonstrating the effectiveness of the spatial-temporal modeling of our SloshNet. The main contributions of our work can be summarized as follows:

- We propose a simple and effective network named SloshNet for few-shot action recognition, which integrates spatial, long-term temporal and short-term temporal features.
- We design a feature fusion architecture search module (FFAS) to automatically search for the best combination of the low-level and high-level spatial features.
- We introduce a long-term temporal modeling module (LTMM) and design a short-term temporal modeling module (STMM) based on the attention mechanism to encode complementary global and local temporal representations.
- The extensive experiments on four widely-used datasets (Something-Something V2, SSV2 (Goyal et al. 2017), Kinetics (Carreira and Zisserman 2017), UCF101 (Soomro, Zamir, and Shah 2012), and HMDB51 (Kuehne et al. 2011)) demonstrate the effectiveness of our methods.

Related Works

Few-shot Image Classification

The core problem of few-shot image classification is to obtain satisfactory prediction results based on a handful of training samples. Unlike the standard training methods in deep learning, few-shot image classification uses the episodic training paradigm, making a handful of labeled training samples from numerous similar tasks as a surrogate for many labeled training samples. Existing mainstream methods of few-shot classification can mainly be classified as adaptation-based and metric-based. The adaptation-based approaches aim to find a network initialization that can be fine-tuned for unknown tasks using few data, called *gradient by gradient*. The evidence of adaptation-based approaches can be clearly seen in the cases of MAML (Finn, Abbeel, and Levine 2017) and Reptile (Nichol and Schulman 2018). The metric-based approaches aim to find a fixed feature representation in which the target task can be embedded and classified. The effectiveness of this kind of approach has been exemplified in Prototypical Networks (Snell, Swersky, and Zemel 2017) and Matching Networks (Vinyals et al. 2016). In addition, CrossTransformer (Doersch, Gupta, and Zisserman 2020) aligns the query and support set based on co-occurrences of image patches that combine metric-based features with task-specific adaptations.

Few-shot Video Action Recognition

Inspired by few-shot image classification, MetaUVFS (Patravali et al. 2021) apply the adaptation-based method and design an action-appearance aligned meta-adaptation module to model spatial-temporal relations of actions over unsupervised hard-mined episodes. However, the adaptation-based method requires high computational resources and long experimental time, so it is less commonly used in few-shot action recognition compared to the metric-based method. In this field, scholars have developed different subdivisional concerns about the metric-based approach. Some of metric-based approaches (Zhu and Yang 2018; Cao et al.

2020; Li et al. 2022b) focus on hand-designed temporal alignment algorithms during the class prototype construction process. Simultaneously, TRPN (Wang et al. 2021) focuses on combining visual and semantic features to increase the uniqueness between similar action classes, and another work (Liu et al. 2022a) focuses on frame sampling strategies to avoid omitting critical action information in temporal and spatial dimensions. Moreover, unlike the above methods of prototype matching at the video level, some methods (Perrett et al. 2021; Thatipelli et al. 2022) inspired by CrossTransformer match each query sub-sequence with all sub-sequences in the support set, which can match actions at different speeds and temporal shifts. Our method focuses on modeling spatial-temporal relations based on a handful of labeled data. We can obtain good predictions by feeding rich spatial-temporal features to a common frame-level class prototype matcher like TRX (Perrett et al. 2021).

Method

Fig. 2 illustrates our overall few-shot action recognition framework. The query video Q and the class support set videos S^k passed through the feature extractor, and store the output features of each layer in a feature bank. The features from the feature bank are input into the feature fusion architecture search module (FFAS) to obtain the spatial fusion feature $\mathbf{F}_{SP}^{Q}$, $\mathbf{F}_{SP}^{S^k}$. Next, we do the weighted summation of the fused feature and the original last layer feature from the feature bank with a learnable parameter γ to obtain the enhanced spatial feature. Followed previous works (Yang et al. 2020; Zhu et al. 2021b; Thatipelli et al. 2022), we model the temporal relation after the acquisition of spatial features to obtain better spatial-temporal integration features. Therefore, the obtained spatial features will be passed through a long-term temporal modeling module (LTMM) and a short-term temporal modeling module (STMM) to model long-term and short-term temporal characteristics $\mathbf{F}_{LT}^{Q}, \mathbf{F}_{LT}^{S^k}, \mathbf{F}_{ST}^{Q}, \mathbf{F}_{ST}^{S^k}$ in parallel. Then do the fusion with another learnable parameter, which adaptively fuses the two kinds of temporal features. Finally, the class prediction $\widehat{y}_Q$ of the query video Q and loss $\mathcal{L}$ are obtained by a frame-level prototype matcher. Details are shown in the subsequent subsections.

Problem Formulation

The few-shot action recognition is considered an N-way, K-shot task. It assigns an unlabeled query video to one of the N classes in the support set, each containing K-labeled videos that were not seen during the training process. We follow an episode training paradigm in line with most previous works (Zhu and Yang 2018; Cao et al. 2020; Perrett et al. 2021), where episodes are randomly drawn from an extensive data collection, and each episode is seen as a task. In each task, we let $Q = \{q_1, q_2, \cdots, q_l\}$ denote a query video randomly sampled l frames, and $S_m^k = \{s_{m1}^k, s_{m2}^k, \cdots, s_{ml}^k\}$ represents the m^{th} video in class $k \epsilon K$ randomly sampled l frames.

FFAS: Feature Fusion Architecture Search Module

The low-level features extracted in the earlier layers of the feature extractor focus more on the structure and texture information, while the high-level features extracted in the last layers focus more on the semantic information. The fusion of them helps improve the spatial representations. Inspired by (Liu, Simonyan, and Yang 2018; Ghiasi, Lin, and Le 2019), we design a feature fusion architecture search module (FFAS). Our goal is to fuse features from different layers output by the feature extractor with an auto-search fusion module, which enables us to find the best combination of the low-level and high-level spatial characteristics in different scenarios. Specifically, we give the features of each layer (total L layers) $\mathcal{F} = \{\mathbf{F}_1, \cdots, \mathbf{F}_i, \cdots, \mathbf{F}_L\}$ ($\mathbf{F}_i \in \mathbb{R}^{NT \times C_i \times H_i \times W_i}$) where N, T, C, H, W are the batch size, time, spatial, height, and width, respectively. To facilitate the subsequent fusion process, we align each layer feature's spatial and channel dimension to the last layer feature, i.e.: $\mathbf{F}_i \in \mathbb{R}^{NT\times\ C_i\times\ H_i\times\ W_i} \to \mathbb{R}^{NT\times\ C\times\ H\times\ W}$ as follows:

$$\mathbf{F}_i = Module_{align}\left(\mathbf{F}_i\right) \tag{1}$$

where $Module_{align}$ here is a 3 × 3 convolution layer. After feature alignment, each layer feature will be updated with the fusion of all previous layers' features as:

$$\mathbf{F}_j = \sum_{i<j} \bar{o}_{i,j}(\mathbf{F}_i, \mathbf{F}_j) \tag{2}$$

where $\bar{o}_{i,j}\left(\mathbf{F}_i, \mathbf{F}_j\right)$ is the weighted summation result of the features of layer i and j after passing all optional fusion operations. We let the set of these fusion options be indicated as $\mathcal{O}$. In our work, we provide three parameter-free fusion options Sum, GP_{low}, and GP_{high}, as shown in Fig. 3(a). To make the search space continuous, we assign a weight α to each operation and perform a $softmax$. This search task can be simplified to learning weights α, and $o_{i,j}\left(\mathbf{F}_i, \mathbf{F}_j\right)$ can be calculated as:

$$\bar{o}_{i,j}\left(\mathbf{F}_i, \mathbf{F}_j\right) = \sum_{o \in \mathcal{O}} \frac{exp\left(\alpha_{i,j}{}^{o}\right)}{\sum_{o' \in \mathcal{O}} exp\left(\alpha_{i,j}{}^{o'}\right)} o_{i,j}\left(\mathbf{F}_i, \mathbf{F}_j\right) \tag{3}$$

The feature fusion search process is shown in Fig. 3(b), and the weights of each fusion operation are initialized equally. Moreover, the output of the module is the fusion of the last layer feature with the updated features of all the previous layers, denoted as $\mathbf{F}_{SP} \in \mathbb{R}^{NT \times C \times H \times W}$. Finally, we do the weighted summation of the fused feature $\mathbf{F}_{SP}$ and the final feature $\mathbf{F}_L$ output by the feature extractor with a learnable parameter $\gamma \epsilon \left[0, 1\right]$, given by

$$\mathbf{F}_{SP} = \left(1 - \gamma\right) \mathbf{F}_{SP} + \gamma \mathbf{F}_L \tag{4}$$

LTMM: Long-term Temporal Modeling Module

In few-shot action recognition, many objects move over time, so many actions could be classified according to their global temporal contextual information. We employ a long-term temporal modeling module (LTMM) to model the global temporal relations based on the extracted spatial appearance features.

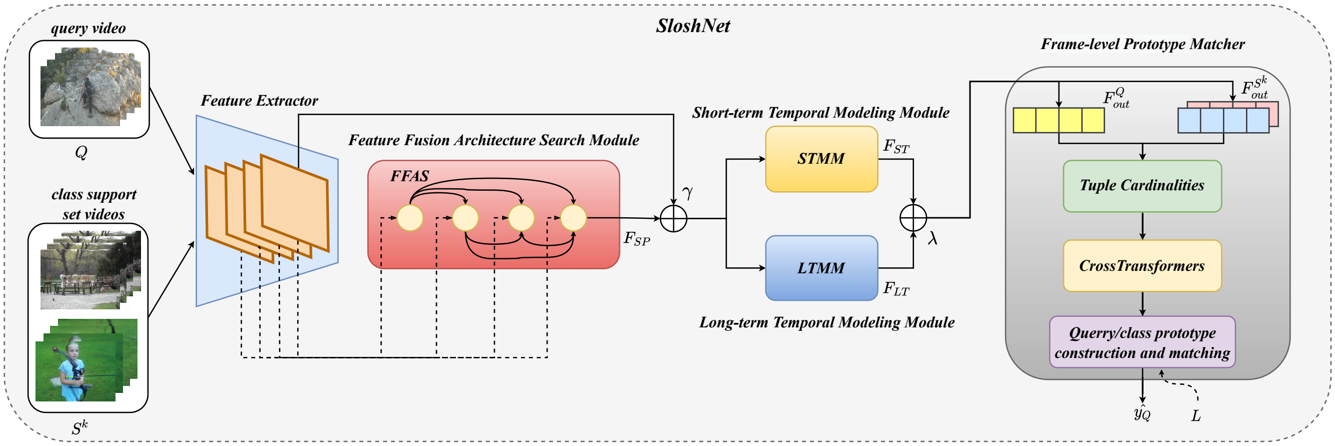

Figure 2: Overview of **SloshNet**. The spatial fusion feature $\mathbf{F}_{SP}$ is obtained by the future fusion architecture search module (FFAS). The long-term temporal feature $\mathbf{F}_{LT}$ is obtained by the long-term temporal modeling module (LTMM). The short-term temporal feature $\mathbf{F}_{ST}$ is obtained by the short-term temporal modeling module (STMM). The $\widehat{y}_Q$ is the class prediction of the query video and the loss $\mathcal{L}$ is a standard cross-entropy loss. $\oplus$ indicates element-wise weighted summation with a learnable parameter.

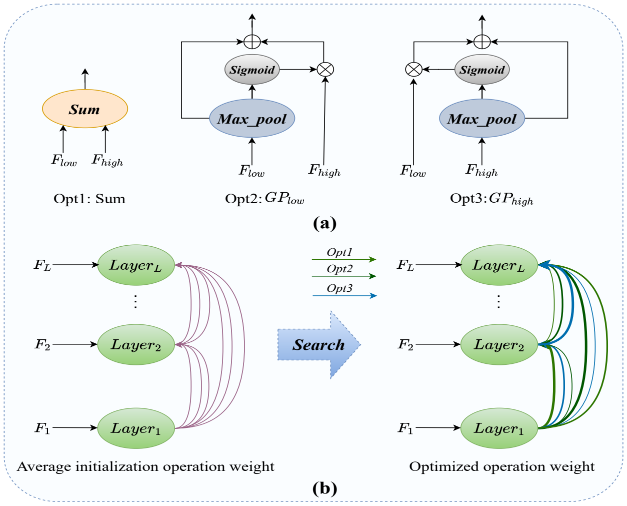

Figure 3: Three feature fusion options, including the sum option and two types of global pooling options, are shown in (a), in which Max_pool devotes max pooling. (b) shows the feature fusion search process. $\oplus$ indicates element-wise summation and $\otimes$ indicates element-wise product.

We present the structure of LTMM in Fig. 4. Given a video feature map after spatial enhancement indicated as $\mathbf{F}_{SP} \in \mathbb{R}^{N \times T \times C \times H \times W}$, it will be reshaped to a sequence shown as $\mathbf{F}_{sq} \in \mathbb{R}^{NHW \times T \times C}$. Then, we let $\mathbf{F}_{sq}$ do self-attention in the temporal dimension:

$$\mathbf{F}_{sq} = \mathbf{F}_{sq} + Module_{att}\left(\mathbf{F}_{sq}\right) \tag{5}$$

where $Module_{att}$ is a L_t layer multi-head attention. The obtained features are then pointed-wise refined by a residual feed-forward network to obtain long-term temporal features $\mathbf{F}_{LT} \in \mathbb{R}^{NHW \times T \times C}$, given by:

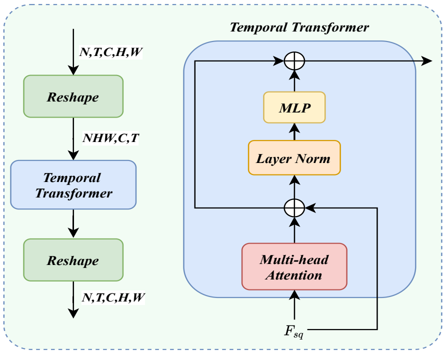

Figure 4: The architecture of the long-term temporal modeling module (LTMM). $\oplus$ denotes element-wise summation.

$$\mathbf{F}_{LT} = \mathbf{F}_{sq} + \varphi(LN(\mathbf{F}_{sq})) \tag{6}$$

where LN denotes the layer normalization and φ denotes the multi-layer perceptron. Next, $\mathbf{F}_{LT}$ will be reshaped to the original input shape (i.e. [N,T,C,H,W]).

STMM: Short-term Temporal Modeling Module

The classification of many action categories requires short-term temporal information, representing the motion characteristics of adjacent frames, and is beneficial for recognizing many temporal-related actions. Therefore, we propose a novel short-term temporal modeling module (STMM) to encode the motion information between adjacent frame representations in the feature level.

Given a video feature map after spatial enhancement $\mathbf{F}_{SP} \in \mathbb{R}^{N \times T \times H \times W \times C}$, we obtain query-key-value triplets using learnable weights $W_1, W_2, W_3 \in \mathbb{R}^{D \times D}$,

$$\mathbf{F}^q = \mathbf{F}_{SP}\mathbf{W}_1, \quad \mathbf{F}^k = \mathbf{F}_{SP}\mathbf{W}_2, \quad \mathbf{F}^v = \mathbf{F}_{SP}\mathbf{W}_3 \tag{7}$$

Next, we reshape $\mathbf{F}^q$, $\mathbf{F}^k \in \mathbb{R}^{NTr \times C/r \times H \times W}$ to reduce the channels by a factor of r to ease the computing cost and leverage two channel-wise 3×3 convolution layers $\mathbf{K}^q$, $\mathbf{K}^k$ on $\mathbf{F}^q$ and $\mathbf{F}^k$, given by

$$\mathbf{F}^q = \sum_{i,j} \mathbf{K}^q_{c,i,j} \mathbf{F}^q_{c,h+i,w+j}, \quad \mathbf{F}^k = \sum_{i,j} \mathbf{K}^k_{c,i,j} \mathbf{F}^k_{c,h+i,w+j} \tag{8}$$

where c, h, w represent the channel and two spatial dimensions of the feature map. $\mathbf{K}^q_{c,i,j}$ and $\mathbf{K}^k_{c,i,j}$ indicate the c^{th} filter, with the subscripts $i, j \epsilon \{-1, 0, 1\}$ to show the spatial indices of the kernel. Next, we do the staggered subtraction over the temporal dimension between $\mathbf{F}^q$ and $\mathbf{F}^k$ to obtain the motion information in feature level, i.e. $\mathbf{F}^q_t$ and $\mathbf{F}^k_{t+1}$. Formally,

$$\mathbf{M} = Concat\left(\left(\mathbf{F}^q_1 - \mathbf{F}^k_2\right), \cdots, \left(\mathbf{F}^q_t - \mathbf{F}^k_{t+1}\right)\right) \tag{9}$$

where for the L frames video $(1 \leq t \leq L-1)$. The temporal dimension of the motion representation $\mathbf{M}$ is $T-1$, so we use zero to represent the motion information of the last time step to help $\mathbf{M}$ keep the temporal size compatible with the input feature maps. Then, the $\mathbf{M}$ is reshaped to the shape of original input features $\mathbf{M} \in \mathbb{R}^{N \times T \times H \times W \times C}$ to restore the number of channels to C. In the end, a feed-forward network (FFN) is applied on the motion attention $\mathbf{M}$ and the final output of STMM is obtained as:

$$\mathbf{F}_{ST} = Sigmoid\left(MLP\left(GELU\left(MLP\left(\mathbf{M}\right)\right)\right)\right)\mathbf{F}^v \tag{10}$$

The structure of STMM is shown in Fig. 5. Finally, we do the weighted summation of the short-term temporal features $\mathbf{F}_{ST}$ and the long-term temporal features $\mathbf{F}_{LT}$ with a learnable parameter $\lambda \epsilon [0, 1]$, given by

$$\mathbf{F}_{out} = (1 - \lambda)\mathbf{F}_{ST} + \lambda \mathbf{F}_{LT} \tag{11}$$

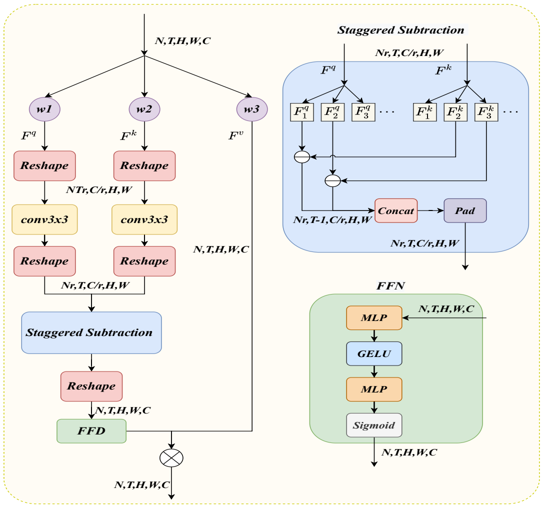

Figure 5: The architecture of the short-term temporal modeling module (STMM). $\ominus$ indicates element-wise subtraction, and $\otimes$ shows the element-wise product.

Class Prototype Matcher

Frame-level prototype construction and matching facilitate fine-grained classification of few-shot action recognition. In our work, we want to show that good predictions can be obtained by feeding rich spatial-temporal features to a common frame-level class prototype matcher. We followed the TRX (Perrett et al. 2021), a common frame-level prototype matcher, which matched each query sub-sequence with all sub-sequences in the support set to construct a query-specific class prototype.

Specifically, we first construct a frame-level feature representation of the video. $r_i \in \mathbb{R}^D$ denotes the i^{th} frame representation and a sequence representation between the i^{th} frame and the j^{th} frame $(\omega = 2)$ is shown as $(r_i, r_j) \in \mathbb{R}^{2D}$, where $1 \leq i \leq j \leq l$, and so on. For any tuple $t \in \Pi^\omega(\omega \epsilon \Omega)$, aggregate all possible sub-sequences in the support video $S^k_{mt} \in \mathbb{R}^{\omega D}$ of an action class to compute a query-specific class prototype, where the aggregation weights are based on the cross-attention of the query sub-sequence and the support class sub-sequence. Let the query embedding indicate as $\mathbf{q}_t \in \mathbb{R}^{D'}$ and the query-specific class prototype denote as $\mathbf{u}^k{}_t \in \mathbb{R}^{D'}$. Then, the distance between a query video Q and a class in support set $\mathbf{S}^k$ over multiple cardinalities Ω can be calculated as:

$$\mathbf{D}\left(Q, \mathbf{S}^k\right) = \sum_{\omega \epsilon \Omega} \frac{1}{|\Pi^\omega|} \sum_{t \in \Pi^\omega} ||\mathbf{q}_t - \mathbf{u}^k{}_t|| \tag{12}$$

The distance $\mathbf{D}(\cdot, \cdot)$ is minimized by a standard cross-entropy loss from the query video to its ground-truth class. During the inference, the query is assigned the class closest to the query with $\mathbf{D}$, i.e., $argmin(\mathbf{D})$.

Experiments

Experimental Setup

Network Architectures We use the ResNet-50 as the feature extractor with ImageNet pre-trained weights (Deng et al. 2009). In FFAS, we automatically search for the best combination of the four layers in ResNet-50 and the weights of the three optional operations are initialized equally in each layer. We use the 3×3 convolution layer as the $Module_{align}$ in FFAS and two layers multi-head attention as the $Module_{att}$ in LTMM. r in STMM is set to 16. The initial weight of the learnable parameter γ and λ is set to 0.9 and 0.5, respectively. In frame-level class prototype matcher, we set D' = 1152, $\Omega = \{1\}$ for spatial-related datasets, and $\Omega = \{1, 2\}$ for temporal-related dataset.

Training and Inference We uniformly sampled 8 frames (l=8) of a video as the input augmented with random horizontal flipping and 224×224 crops in training, while only a center crop in inference. For training, SSV2 were randomly sampled 100,000 training episodes with an initial learning rate of 10^{-4}, and the other datasets were randomly sampled 10,000 training episodes with an initial learning rate of 10^{-3}. Moreover, we used the SGD optimizer with the multi-step scheduler for our framework. For inference, we reported the average results over 10,000 tasks randomly selected from the test sets in all datasets.

Table 1: State-of-the-art comparison on the 5-way 5-shot spatial-related benchmarks of Kinetics, HMDB51, and UCF101.

Methods	Kinetics	HMDB	UCF
CMN (Zhu and Yang 2018)	78.9	-	-
ProtoNet (Snell, Swersky, and Zemel 2017)	64.5	54.2	78.7
TARN (Bishay, Zoumpourlis, and Patras 2019)	78.5	-	-
ARN (Zhang et al. 2020)	82.4	60.6	83.1
OTAM (Cao et al. 2020)	85.8	-	-
HF-AR (Kumar and Narang 2021)	-	62.2	86.4
TRX (Perrett et al. 2021)	85.9	75.6	96.1
TA2N (Li et al. 2022b)	85.9	75.6	96.1
STRM (Thatipelli et al. 2022)	86.5	77.3	96.9
SloshNet	**87.0**	**77.5**	**97.1**

Table 2: State-of-the-art comparison on the 5-way temporal-motion-related benchmark of SSV2. * refers to our re-implementation and () refers to the reported result.

Methods	SSV2	
	1-shot	5-shot
ProtoNet (Snell, Swersky, and Zemel 2017)	39.3	52.0
OTAM (Cao et al. 2020)	42.8	52.3
HF-AR (Kumar and Narang 2021)	43.1	55.1
PAL (Zhu et al. 2021a)	46.4	62.6
TRX (Perrett et al. 2021)	42.0	64.6
STRM (Thatipelli et al. 2022)	43.5	66.0* (68.1)
SloshNet	**46.5**	**68.3**

Results

Results on Spatial-Related Datasets For the spatial-related datasets, the recognition of actions depends mainly on the background information and a small part on the temporal information. So in the experiments of these datasets, we set $\Omega = \{1\}$, making it more focused on background information during the class prototype construction and matching. Also, since we have modeled the long-term and short-term temporal relations at the feature level, each frame feature has an intrinsic temporal representation. The state-of-the-art comparison for the 5-way 5-shot action recognition task of three spatial-related datasets, including Kinetics, HMDB51, and UCF101, was shown in Tab. 1. On all three datasets, we achieve the new state-of-the-art results. Taking the Kinetics as an instance, compared to our baseline TRX (Perrett et al. 2021), we bring a 1.1% performance improvement demonstrating the effectiveness of our spatial-temporal relation modeling. Meanwhile, compared to the similar method STRM (Thatipelli et al. 2022) that focused on spatial-temporal modeling but lacked short-term temporal modeling, our approach brings a 0.5% accuracy improvement showing the significance of characteristics between adjacent frame representations. Similarly, our SloshNet achieves the best performance in HMDB51 and UCF101.

Results on Temporal-Related Dataset For the temporal-related dataset SSV2, the key to action recognition is long-term and short-term temporal information. Therefore, we set $\Omega = \{1, 2\}$ to reinforce long-term and short-term temporal relation modeling both during feature-level and class prototype construction and matching process. The state-of-the-art comparison for the 5-way 1-shot and 5-way 5-shot action recognition tasks of the temporal-related dataset SSV2 was shown in Tab. 2. Compared to the best existing method STRM (Thatipelli et al. 2022) in SSV2, SloshNet has a large improvement of 3.0% on 5-way 1-shot task and 2.3% on 5-way 5-shot task.

Ablation Study

Impact of Proposed Modules To validate the contributions of each module in the SloshNet, we experiment on the 5-way 1-shot task of SSV2 ($\Omega = \{1, 2\}$) to ablate our proposed components in Tab. 3. The spatial representation enhancement module FFAS brings about a 0.8% accuracy improvement. LTMM and STMM combined as the temporal modeling module (TMM) bring a 1.6% gain. When combining FFAS and TMM, we can learn spatial and temporal features together and achieve the best accuracy, with the gain of 4.5% over the baseline. Meanwhile, we also provide the attention visualization of our SloshNet in Fig. 6, which gradually integrates the impact of our contribution. After integrating FFAS (third row), our framework enhances the feature spatial representation, which helps concentrate on relevant objects in a single video frame, e.g., the frames in (a) and (b) reduce attention on the background and extraneous objects. Furthermore, after integrating TMM (fourth row), including LTMM and STMM, our framework enhances the feature temporal relation, which lets our SloshNet highly correlate with the action subject, e.g., the fourth and eighth frame from the left in (a) has a better concentration on the snowman, and the frames in (b) has more detailed attention extended to the marker pen.

Table 3: The impact of proposed modules.

FFAS	LTMM	STMM	Acc
×	×	×	42.0
✓	×	×	42.8
×	✓	✓	43.6
✓	✓	✓	**46.5**

Impact of Options and Architecture Search Mechanism in FFAS There are three parameter-free feature fusion options in FFAS: Sum, GP_{low}, and GP_{high}. We conduct 5-way 1-shot task experiments on SSV2 ($\Omega = \{1, 2\}$) to explore the impact of the individual option, combination of options, and architecture search mechanism in FFAS shown in Tab. 4. Perform Sum, GP_{low}, GP_{high} options individually, which outperforms baseline by 0.6%, 0.3%, 0.9%, respectively. Above three options are performed simultaneously and summed with equal weights as output, bringing a 2.6% accuracy improvement. When using the architecture search

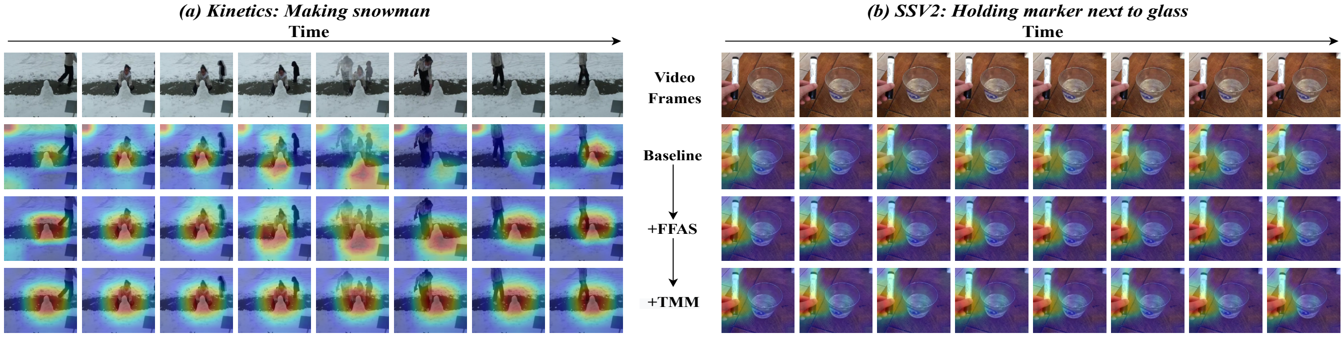
(a) Kinetics: Making snowman
Time
(b) SSV2: Holding marker next to glass
Time
Video Frames
Baseline
+FFAS
+TMM

Figure 6: Attention visualization of our SloshNet on two examples. From top to bottom, we gradually integrate the impact of our contribution, in which FFAS indicates feature fusion architecture search and TMM denotes the combination of short-term and long-term temporal relation modeling.

module under the three feature fusion options, we can yield the best result, bringing a 2.9% accuracy improvement.

Table 4: The impact of options and architecture search mechanism in FFAS.

Sum	GP_{low}	GP_{high}	Architecture Search	Acc
×	×	×	×	43.6
✓	×	×	×	44.2
×	✓	×	×	43.9
×	×	✓	×	44.5
✓	✓	✓	×	45.2
✓	✓	✓	✓	**46.5**

Impact of Temporal Modeling Integration We also discuss the impact of temporal modeling integration shown in Tab. 5. We conclude that neither short-term nor long-term temporal relation modeling can fully obtain the representation of temporal features. Moreover, compared to the concatenation and parallel connection summation, parallel connection weighted summation with a learnable parameter λ is the most effective way to integrate long-term and short-term temporal features, which brings a 3.7% accuracy improvement to the no any temporal modeling one.

Table 5: The impact of temporal modeling integration. "+" indicates concatenation, "//" indicates parallel connection summation, and $\oplus$ indicates parallel connection weighted summation with a learnable parameter λ.

Temporal Modeling Integration	Acc
no any temporal modeling	42.8
only STMM	43.3
only LTMM	43.7
LTMM + STMM	43.9
STMM + LTMM	44.8
STMM // LTMM	45.7
STMM $\oplus$ LTMM	**46.5**

Impact of Varying Cadinalities Tab. 6 shows the impact of varying cardinalities given temporal relation modeling during the class prototype construction and matching process. As part of our experiments, we perform 5-way 5-shot task on Kinetics and 5-way 1-shot task on SSV2. On both datasets, we then evaluate each cardinality of $\Omega \epsilon \{1, 2, 3\}$ independently and all their combinations. As for the spatial-related dateset Kinetics, we find $\Omega = \{1\}$ can achieve the best accuracy of 87.0%, which makes it more focused on background information during the class prototype construction and matching. Furthermore, for the temporal-related dataset SSV2, we set $\Omega = \{1, 2\}$ to reinforce temporal relation modeling during this process, which can get the best accuracy of 46.5%. In fact, for all datasets, high accuracy can be acquired at $\Omega = \{1\}$, which proves that we already have rich spatial-temporal representation at the feature level. Taken together, these results suggest that if strong enough spatial-temporal feature representations are extracted, the matching process could be simplified a lot.

Table 6: The impact of varying the cardinallities on Kinetics and SSV2. Comparing all values of Ω for our SloshNet.

Cadinalities (Ω)	$\{1\}$	$\{2\}$	$\{3\}$	$\{1,2\}$	$\{1,3\}$	$\{2,3\}$	$\{1,2,3\}$
Kinetics	**87.0**	86.7	86.5	86.7	86.6	86.5	86.8
SSV2	46.3	46.1	45.9	**46.5**	46.3	46.0	46.3

Conclusion

This paper presents a few-shot action recognition framework, SloshNet, which integrates spatial, long-term temporal, and short-term temporal features into a unified framework. A feature fusion architecture search module (FFAS) is proposed to automatically search for the best combination of the low-level and high-level spatial features to enhance feature spatial representation. A long-term temporal modeling module (LTMM) is introduced to model the global temporal relations based on the extracted spatial appearance features, and a short-term temporal modeling module (STMM) is proposed to encode the motion characteristics between adjacent frame representations. Comprehensive experiments demonstrate the effectiveness of every module and the whole framework.

Acknowledgements

This work is partly supported by the following grant: Key R&D Program of Zhejiang (No.2022C03126).

References

Bishay, M.; Zoumpourlis, G.; and Patras, I. 2019. Tarn: Temporal attentive relation network for few-shot and zero-shot action recognition. *arXiv preprint arXiv:1907.09021*.

Cao, K.; Ji, J.; Cao, Z.; Chang, C.-Y.; and Niebles, J. C. 2020. Few-shot video classification via temporal alignment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10618–10627.

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

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Doersch, C.; Gupta, A.; and Zisserman, A. 2020. Crosstransformers: spatially-aware few-shot transfer. *Advances in Neural Information Processing Systems*, 33: 21981–21993.

Elsken, T.; Staffler, B.; Metzen, J. H.; and Hutter, F. 2020. Meta-learning of neural architectures for few-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12365–12375.

Feichtenhofer, C.; Fan, H.; Malik, J.; and He, K. 2019. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, 6202–6211.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, 1126–1135. PMLR.

Ghiasi, G.; Lin, T.-Y.; and Le, Q. V. 2019. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7036–7045.

Goyal, R.; Ebrahimi Kahou, S.; Michalski, V.; Materzynska, J.; Westphal, S.; Kim, H.; Haenel, V.; Fruend, I.; Yianilos, P.; Mueller-Freitag, M.; et al. 2017. The" something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE international conference on computer vision*, 5842–5850.

Kuehne, H.; Jhuang, H.; Garrote, E.; Poggio, T.; and Serre, T. 2011. HMDB: a large video database for human motion recognition. In *2011 International conference on computer vision*, 2556–2563. IEEE.

Kumar, N.; and Narang, S. 2021. Few shot activity recognition using variational inference. *arXiv preprint arXiv:2108.08990*.

Li, K.; Wang, Y.; Zhang, J.; Gao, P.; Song, G.; Liu, Y.; Li, H.; and Qiao, Y. 2022a. Uniformer: Unifying convolution and self-attention for visual recognition. *arXiv preprint arXiv:2201.09450*.

Li, S.; Liu, H.; Qian, R.; Li, Y.; See, J.; Fei, M.; Yu, X.; and Lin, W. 2022b. TA2N: Two-Stage Action Alignment Network for Few-Shot Action Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 1404–1411.

Liu, H.; Lv, W.; See, J.; and Lin, W. 2022a. Task-adaptive Spatial-Temporal Video Sampler for Few-shot Action Recognition. *arXiv e-prints*, arXiv–2207.

Liu, H.; Simonyan, K.; and Yang, Y. 2018. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055*.

Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; and Hu, H. 2022b. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3202–3211.

Ma, Y.; Bai, S.; An, S.; Liu, W.; Liu, A.; Zhen, X.; and Liu, X. 2020. Transductive Relation-Propagation Network for Few-shot Learning. In *IJCAI*, volume 20, 804–810.

Nichol, A.; and Schulman, J. 2018. Reptile: a scalable metalearning algorithm. *arXiv preprint arXiv:1803.02999*, 2(3): 4.

Patravali, J.; Mittal, G.; Yu, Y.; Li, F.; and Chen, M. 2021. Unsupervised Few-Shot Action Recognition via Action-Appearance Aligned Meta-Adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8484–8494.

Perrett, T.; Masullo, A.; Burghardt, T.; Mirmehdi, M.; and Damen, D. 2021. Temporal-relational crosstransformers for few-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 475–484.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Soomro, K.; Zamir, A. R.; and Shah, M. 2012. UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*.

Thatipelli, A.; Narayan, S.; Khan, S.; Anwer, R. M.; Khan, F. S.; and Ghanem, B. 2022. Spatio-temporal relation modeling for few-shot action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 19958–19967.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Gool, L. V. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *European conference on computer vision*, 20–36. Springer.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2018. Temporal segment networks for action recognition in videos. *IEEE transactions on pattern analysis and machine intelligence*, 41(11): 2740–2755.

Wang, M.; Xing, J.; Su, J.; Chen, J.; and Yong, L. 2022. Learning SpatioTemporal and Motion Features in a Unified 2D Network for Action Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Wang, X.; Ye, W.; Qi, Z.; Zhao, X.; Wang, G.; Shan, Y.; and Wang, H. 2021. Semantic-Guided Relation Propagation Network for Few-shot Action Recognition. In *Proceedings of the 29th ACM International Conference on Multimedia*, 816–825.

Yang, C.; Xu, Y.; Shi, J.; Dai, B.; and Zhou, B. 2020. Temporal pyramid network for action recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 591–600.

Zhang, H.; Zhang, L.; Qi, X.; Li, H.; Torr, P. H.; and Koniusz, P. 2020. Few-shot action recognition with permutation-invariant attention. In *European Conference on Computer Vision*, 525–542. Springer.

Zhu, L.; and Yang, Y. 2018. Compound memory networks for few-shot video classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 751–766.

Zhu, X.; Toisoul, A.; Perez-Rua, J.-M.; Zhang, L.; Martinez, B.; and Xiang, T. 2021a. Few-shot action recognition with prototype-centered attentive learning. *arXiv preprint arXiv:2101.08085*.

Zhu, Z.; Wang, L.; Guo, S.; and Wu, G. 2021b. A Closer Look at Few-Shot Video Classification: A New Baseline and Benchmark. *arXiv preprint arXiv:2110.12358*.