Correlative Channel-Aware Fusion for Multi-View Time Series Classification

Yue Bai,[1] **Lichen Wang,** [1] **Zhiqiang Tao,** [2] **Sheng Li,** [3] **Yun Fu** [1]
[1] Department of Electrical and Computer Engineering, Northeastern University, Boston, USA
[2] Department of Computer Science and Engineering, Santa Clara University, Santa Clara, USA
[3] Department of Computer Science, University of Georgia, Athens, USA
bai.yue@northeastern.edu, wanglichenxj@gmail.com, ztao@scu.edu, sheng.li@uga.edu, yunfu@ece.neu.edu

Abstract

Multi-view time series classification (MVTSC) aims to improve the performance by fusing the distinctive temporal information from multiple views. Existing methods for MVTSC mainly aim to fuse multi-view information at an early stage, *e.g.*, by extracting a common feature subspace among multiple views. However, these approaches may not fully explore the unique temporal patterns of each view in complicated time series. Additionally, the label correlations of multiple views, which are critical to boosting, are usually under-explored for the MVTSC problem. To address the aforementioned issues, we propose a Correlative Channel-Aware Fusion (C^2AF) network. First, C^2AF extracts comprehensive and robust temporal patterns by a two-stream structured encoder for each view, and derives the intra-view/inter-view label correlations with a concise correlation matrix. Second, a channel-aware learnable fusion mechanism is implemented through CNN to further explore the global correlative patterns. Our C^2AF is an end-to-end framework for MVTSC. Extensive experimental results on three real-world datasets demonstrate the superiority of our C^2AF over the state-of-the-art methods. A detailed ablation study is also provided to illustrate the indispensability of each model component.

Introduction

Time series classification (TSC) is becoming a popular research topic recently, which provides more comprehensive information for the changing world. Many algorithms are proposed for modeling time series data in different application domains, *e.g.*, transportation (Yao et al. 2018), healthcare (Harutyunyan et al. 2017), and human action (Wang, Ding, and Fu 2019, 2018). However, compared with static data such as images, the complicated dynamic patterns contained in time series make TSC a challenging problem. Fortunately, owing to the advanced sensing techniques, objects or events can be observed through multiple modalities, which brings in multi-view time series data to improve the classification performance. For example, RGB, depth, and skeleton are three common modalities for human action recognition. They provide more comprehensive information to depict human actions than each single view. For another example, several types of human body signals are recorded as different modalities in health-care applications, such as magnetic resonance imaging (MRI) and electrocardiograph (ECG). These multi-view signals could monitor different physical states simultaneously. Generally, multi-view time series provide view-specific information from different angles and facilitate with each other for higher learning performance over an individual view.

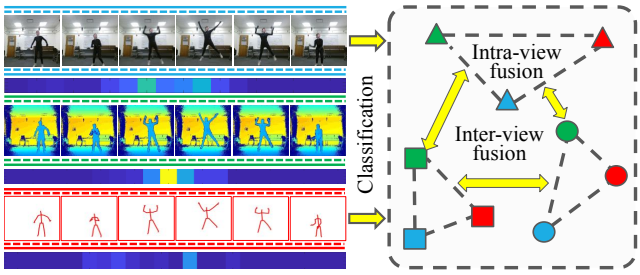

Figure 1: Multi-view temporal data has distinctive patterns in each view such as the attention scores. Intra-view and inter-view label correlations are crucial to improve multi-view performance.

Multi-view learning (MVL) has drawn significant attention, since utilizing complementary information from different views has great potential to boost the final learning performance. MVL is successfully applied in many applications (Xu, Tao, and Xu 2013; Nie et al. 2016; Nie, Cai, and Li 2017; Tao et al. 2019; Zhang et al. 2019). Previous algorithms could be roughly divided into three groups (Xu, Tao, and Xu 2013): (1) co-training; (2) multiple kernel learning; and (3) subspace learning. Specifically, the co-training methods integrate multi-view data via maximizing the common mutual information of different views; the multiple kernel learning methods design specific learning kernels for each view and then combine them together; and the subspace learning methods seek for the common latent subspace shared by multiple views. Although these methods have achieved promising results, it is not straightforward to directly employ them for TSC due to the dynamic temporal patterns in time series.

Existing TSC methods focusing on single-view time series have been widely explored under two cases: univari-

Code will be released at https://github.com/yueb17/C2AF

This research is supported by the U.S. Army Research Office Award W911NF-17-1-0367.

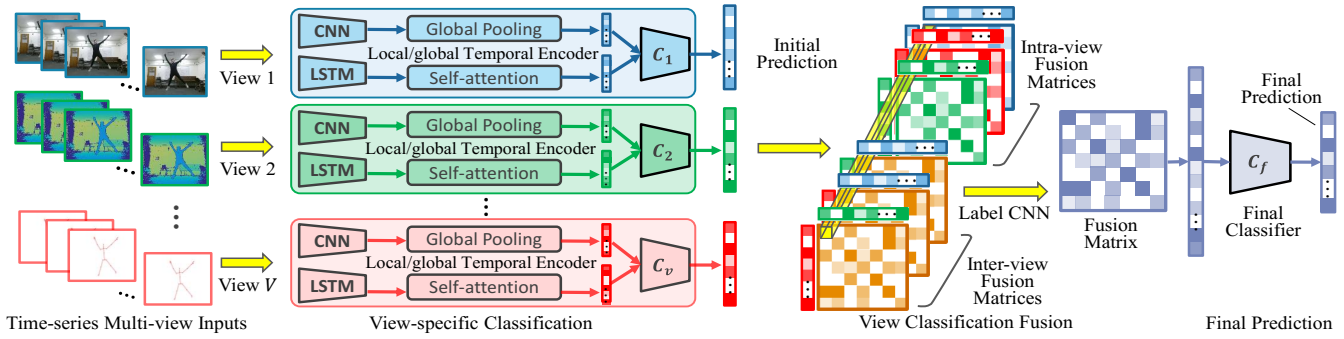

Figure 2: Multi-view temporal data are set as input simultaneously to train the end-to-end C^{2}AF network. A two-stream encoder extracts view-specific temporal patterns. Intra-view/inter-view label correlations are captured by correlation matrices. The channel-aware learnable fusion integrates and fully utilizes multi-view label correlations for performance improvement.

ate (Cuaresma et al. 2004) and multivariate (Zheng et al. 2014; Hüsken and Stagge 2003). On the one hand, the univariate TSC mainly studies the distance measurement between two time series such as (Marteau and Gibet 2014). On the other hand, many research attempts are also made for handling the multivariate time series. To name a few, Bankó and Abonyi (2012) revised the dynamic temporal wrapping (DTW) method, and Cui, Chen, and Chen (2016) utilized the CNNs to model time series. Nevertheless, only a few methods are proposed for solving multi-view and multivariate TSC. For instance, Li, Li, and Fu (2016) proposed a discriminative bilinear projection framework to build a shared subspace for multi-view temporal data. Zadeh et al. (2018) designed a fusion strategy based on LSTM networks. Yuan et al. (2018) proposed an attention mechanism to model multi-view time series. It is worth noting that, all these methods adopt an early fusion strategy, *e.g.*, integrating multi-view information by learning a common feature subspace, which may not fully explore the view-specific distinctive patterns and ignore the multi-view label correlations.

To handle the above issues, we propose a Correlative Channel-Aware Fusion (C^{2}AF) network for the multi-view time series classification (MVTSC) task. Our C^{2}AF jointly leverages the view-specific distinctive temporal patterns existing in feature spaces and the multi-view correlations in label spaces (see Figure 1), to boost the classification performance. Specifically, our model first applies a two-stream temporal encoder to extract robust temporal features, followed by a classifier for each view. By this means, the raw label information is first obtained. After that, the multi-view label correlations are captured by a concise correlation matrix. Finally, a channel-aware learnable fusion mechanism is designed to globally integrate the label correlations and tune the entire network. The main contributions of our paper are summarized as below.

- We propose an end-to-end MVTSC network, namely C^{2}AF, to jointly capture view-specific temporal patterns by two-stream encoders and automatically fuse the multi-view label correlations.
- We design a channel-aware learnable fusion mechanism, which provides an effective late fusion strategy for the MVTSC problem and adopts a concise implementation via convolutional neural networks.
- We conduct substantial experiments on three real-world datasets to show the effectiveness of our C^{2}AF, and provide detailed ablation studies to demonstrate the indispensability of each model component.

Related Work

Time Series Classification

Time series data are collected and analyzed in several domains (Xing, Pei, and Keogh 2010; Cao et al. 2017; Jin and Dong 2016). Generally, the methods for time series classification (TSC) task can be categorized into three groups: (1) feature based classification; (2) sequence distance based classification; and (3) model based classification. Feature based algorithms such as (Kadous and Sammut 2005; Ye and Keogh 2009) extract a feature vector from time series and then apply traditional methods, *e.g.*, K-Nearest neighbor (KNN) (Fukunaga and Narendra 1975) and support vector machine (SVM) (Cortes and Vapnik 1995), to make classification. Further, deep neural network has great capacity to fit non-linear mapping and extract complicated temporal features for classification (Karim et al. 2019). Reservoir computing (Bianchi et al. 2018) is proposed based on recurrent neural networks to learn the representations for multivariate TSC. Distance based methods aim to design distance functions to measure the similarity of a pair of time series. After obtaining a reasonable distance metric, we apply conventional algorithms to further make classification. For example, DTW (Xi et al. 2006) is a typical distance based algorithm which is eligible for time series with different lengths. Other distance based models are also proposed for TSC such as (Wei and Keogh 2006; Dorle et al. 2020; Ratanamahatana and Keogh 2004; Keogh and Kasetty 2003). Model based methods assume that all time series belonging to each class are generated by a potential generative model. During the training stage, the corresponding parameters of the potential model are learned and the test samples are classified based on the likelihood. To name a few, hidden markov model (HMM) (Rabiner 1989) is widely used in TSC for speech recognition. Naive bayes sequence classifier (Rish

et al. 2001) is another typical model based method which observes the feature independent assumption. In our work, we focus on multi-view time series classification (MVTSC) which is not fully explored by above methods.

Multi-View Learning

Multi-view learning (MVL) attracts more attention in recent decades. The distinctive patterns extracted from different views mutually support with each other to benefit final performance. MVL is widely used in several tasks such as object classification (Qi et al. 2016), clustering (Bickel and Scheffer 2004; Zhang et al. 2020; Wang et al. 2018), semi-supervised learning (Hou et al. 2010), action recognition (Cai et al. 2014), and face recognition (Li et al. 2002). Fusing information from multiple views is an effective way to leverage mutual-support patterns for performance improvement in MVL (Swoger et al. 2007; Bruno and Marchand-Maillet 2009). Fusion strategies can be categorized into three groups (Atrey et al. 2010): (1) feature fusion; (2) decision fusion; and (3) hybrid fusion. Feature fusion (early fusion) (Wang et al. 2017; Louis-Philippe, Rada, and Payal 2011) focuses on merge distinctive information from different views in feature space. Decision fusion (late fusion) (Wörtwein and Scherer 2017; Tao et al. 2017, 2020; Zhou et al. 2012) aims to fuse the multiple decisions in label space. Hybrid fusion is a combination strategy of early fusion and late fusion. However, most fusion strategies leverage the multi-view information directly instead of through a learnable way, which may not fully explore the complementary patterns of multiple views. Further, most of them are not designed for temporal data. Deploying them on time series directly will ignore temporal dynamic patterns. In our work, we propose a novel Correlative Channel-Aware Fusion (C^2AF) network for MVTSC. Our proposed C^2AF extracts robust temporal representations and fully explores the multi-view latent correlations through a learnable fusion strategy.

Methodology

Preliminary

Let $\mathcal{X} = \{X^v\}_{v=1}^V$ be the multi-view time series data, where $X^v \in \mathbb{R}^{T\times D^v}$ refers to the v-th view feature matrix. For $\forall v$, T and D^v represent the time series length and feature dimensions, respectively. Let $Y \in \mathbb{R}^K$ be the corresponding label, where K denotes the number of classes. All the views in $\mathcal{X}$ share the same label Y. In this study, we focus on multi-view time series classification (MVTSC) by leveraging multi-view complementary information through a Correlative Channel-Aware Fusion (C^2AF) network. Our C^2AF consists of two parts, global-local temporal encoder and channel-aware learnable fusion.

Global-Local Temporal Encoder

Dynamic and complicated temporal pattern is the key factor to tackle time series data. It usually provides discriminative characteristics to guarantee high quality classification. In our C^2AF approach, obtaining comprehensive and robust temporal representations for each view is indispensable, which provides reliable label information and benefits fusion process. We propose a global-local temporal encoder to fully explore the temporal context. It consists of a global-temporal encoder E_g and a local-temporal encoder E_l. We obtain view specific representations by

$$\begin{aligned} H^v &= q(H_g^v, H_l^v) \\ H_g^v &= E_g(X^v; \phi_g^v) \\ H_l^v &= E_l(X^v; \phi_l^v), \end{aligned} \tag{1}$$

where $H^v \in \mathbb{R}^{d^v}$ is the encoded representations for X^v, H_g^v/H_l^v represents the E_g/E_l output, q denotes a common fusion operation (we use concatenation operation in our work), and E_g, E_l are two networks with learnable parameters ϕ_g^v and ϕ_l^v, respectively. We update ϕ_g^v and ϕ_l^v by minimizing the following loss:

$$L^v = \sum_{i=1}^{N} \ell(Y_i, \hat{Y}_i^v), \tag{2}$$

where ℓ represents the cross-entropy loss and N is the number of samples. $\hat{Y}_i^v = C_v(H_i^v)$ is the prediction for the i-th sample. $C_v : \mathbb{R}^{d^v} \to \mathbb{R}^K$ is the v-th view specific classifier achieved by a linear mapping.

Global-Temporal Encoder Next, we will introduce E_g and E_l with more details. The E_g and E_l are deployed for each view. For convenience, we omit the subscript v in the rest of this section. We adopt recurrent neural networks (RNN) to parameterize our global-temporal encoder E_g, as RNN is well validated as an effective way to explore the temporal context for time-series. Particularly, we employ the LSTM (Hochreiter and Schmidhuber 1997) as the cell, which is given by

$$\begin{aligned} f_t &= \sigma_g(W_f x_t + U_f h_{t-1} + b_f), \\ i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i), \\ o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o), \\ c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c h_t + U_c h_{t-1} + b_c), \\ h_t &= o_t \circ \sigma_h(c_t), \end{aligned} \tag{3}$$

where x_t is t-th representation in sequence input X ($1 \le t \le T$). f_t, i_t, o_t, c_t, and h_t serve as forget gate, input gate, output gate, cell state, and hidden state at time t, respectively. c_{t-1} and h_{t-1} represent cell and hidden states at time $t-1$. σ_g, σ_c, σ_h are activation functions, and $\circ$ represents the element-wise product. In Eq.(3), W_*, U_* and b_* are all learnable weights, $\forall * \in \{f, i, o, c\}$.

To further enhance the global temporal representation, we leverage attention mechanism to integrate the hidden states sequence. By using attention, we explicitly learn the dynamic correlations cross different time points, and obtain the global temporal representation H_g by

$$H_g = \sum_{t=1}^{T} \omega_t h_t, \tag{4}$$

where $\omega = \{\omega_t\}$ is the learnable attention weights.

By using Eqs. (3-4), we formulate our E_g as LSTM with attention mechanism, and have $\phi_g = \{\{W_*, U_*, b_*\}, \omega\}$, $* \in \{f, i, o, c\}$.

Local-Temporal Encoder Different from the global-temporal encoder, we utilize convolutional neural networks (CNN) to formulate our local-temporal encoder E_l, as CNN works well on probing patterns from local-characterized data. Specifically, we apply a set of 1D convolutional filters to extract local patterns in X following the similar strategy in temporal convolutional networks (TCN) (Lea et al. 2016). Let M be the number of CNN layers and $F_m \in \mathbb{R}^{T_m \times D_m}$ be the output of the m-th layer ($1 \leq m \leq M$). T_m and D_m denote the corresponding temporal and feature dimensions, respectively. Given $F_0 = X$, we compute F_m by

$$F_m = \text{BN}_{\{\gamma_m, \beta_m\}}(\text{ReLU}(W_m * F_{m-1} + b_m)), \quad (5)$$

where $W_m \in \mathbb{R}^{D_m \times D_{m-1} \times \Delta T}$ is the weight of convolutional filter, $b_m \in \mathbb{R}^{D_m}$ is the bias. ΔT represents the size of temporal sliding window and $*$ represents the convolution operation. In Eq. (5), $\text{BN}_{\{\gamma_m, \beta_m\}}$ refers to the batch normalization block (Ioffe and Szegedy 2015) with learnable parameters γ_m and β_m. It is used to further improve the effectiveness and stability of E_l.

To avoid the over-fitting issue and diminish the number of parameters, a global average pooling layer (Lin, Chen, and Yan 2013) is deployed after each convolutional block. By using these methods, we efficiently extract local temporal information and obtain high-level representation H_l by

$$H_l = g(F_M), \quad (6)$$

where g is the global average pooling layer.

Through applying Eqs. (5-6), we concretize our E_l as CNN with batch normalization and global average pooling, and have $\phi_l = \{W_m, b_m, \gamma_m, \beta_m\}_{m=1}^{M}$.

Channel-Aware Learnable Fusion

Efficiently fusing mutual-support information from multi-view predicted labels $\hat{Y}^v$ ($1 \leq v \leq V$) is the central fact of performance improvement. In our model, we propose a channel-aware learnable fusion mechanism to sufficiently capture and utilize the label correlations. It takes advantage of intra-view and inter-view label correlations to achieve better multi-view learning results. Specifically, we construct a graph based correlation matrix to probe intra-view/inter-view label correlations and a CNN based fusion module to integrate global patterns. Next, we introduce the channel-aware learnable fusion with more details.

Label Correlation Matrix We adopt a graph based strategy to capture the intra-view and inter-view label correlations, respectively. The intra-view label correlation matrix for each view v is given by

$$G^{v,v} = \hat{Y}^v \cdot \hat{Y}^{v\top}, \quad (7)$$

where $G^{v,v} \in \mathbb{R}^{K \times K}$ is the correlation matrix derived by multiplying the predicted label $\hat{Y}^v \in \mathbb{R}^{K \times 1}$ and its transpose $\hat{Y}^{v\top} \in \mathbb{R}^{1 \times K}$ for $1 \leq v \leq V$. Each element in $G^{v,v}$ represents the intra-view pair-wise label correlations for view v. We integrate V intra-view label correlations by concatenating them together as follow:

$$r_{intra} = [G^{1,1}, G^{2,2}, ..., G^{V,V}], \quad (8)$$

Algorithm 1 The procedure of training C^{2}AF algorithm.

Input: batches of $\{\mathcal{X}, Y\}$, number of view V, number of training steps S

Output: prediction of each view $\hat{Y}^v$ and final result $\hat{Y}^f$

```
for each i ∈ [1, S] do
    for each v ∈ [1, V] do
        sample a batch data X^v from view v
        forward X^v into E_g and E_l
        compute H^v and Ŷ^v through Eq. (1) and C_v
        update φ_g^v, φ_l^v and C_v using Eq. (2)
    end for
    forward Ŷ^v, v ∈ 1, 2, ..., V into E_f
    compute Ŷ^f through Eq. (11) and C_f
    update φ_f and C_f using Eq. (13)
end for
return Ŷ^v and Ŷ^f
```

where $r_{intra} \in \mathbb{R}^{K \times K \times V}$ is the intra-view correlation tensor and $[\cdot]$ is the concatenation operation.

Similarly, the inter-view label correlation matrix for each pair of views is given by

$$G^{u,w} = \hat{Y}^u \cdot \hat{Y}^{w\top}, \quad (9)$$

where $G^{u,w} \in \mathbb{R}^{K \times K}$ is the correlation matrix derived by multiplying the predicted label $\hat{Y}^u \in \mathbb{R}^{K \times 1}$ from view u and the transpose of predicted label $\hat{Y}^w \in \mathbb{R}^{1 \times K}$ from view w for $\forall u, w \in V, u \neq w$. Each element in $G^{u,w}$ represents the inter-view pair-wise label correlations for view u and w. Considering all the possible combinations of view-pair, we integrate $\binom{V}{2}$ inter-view label correlations by concatenating them together as follow:

$$r_{inter} = [G^{1,2}, G^{1,3}, ..., G^{V-1,V}], \quad (10)$$

where $r_{inter} \in \mathbb{R}^{K \times K \times \binom{V}{2}}$ represents the inter-view correlation tensor.

By using Eqs. (7-8) and Eqs. (9-10), we extract the intra-view and inter-view label correlations as two multi-channel tensors r_{intra} and r_{inter}.

Channel-Aware Fusion Multi-view label correlations are extracted and represented by label correlation matrices. The informative patterns of label correlations are reserved in each element instead of a local area of these matrices, but still contained in the same place across different channels of r_{intra} and r_{inter}. Hence, we employ a CNN structure with 1×1 kernels as a channel-aware extractor to globally integrate cross-view correlative information. It is given by

$$r = E_f([r_{intra}, r_{inter}], \phi_f), \quad (11)$$

where $r \in \mathbb{R}^{K \times K \times N_k}$ is the fusion matrix. E_f is the CNN based fusion encoder parameterized by ϕ_f, with N_k kernels. We formulize the fusion encoder E_f by

$$r_{p,q}^{(o)} = f(b^{(o)} + \langle W^{(o)}, [r_{intra}, r_{inter}]_{p,q} \rangle), \quad (12)$$

Dataset	EV-Action				NTU RGB+D				UCI		
	RGB	Depth	Skeleton	Three-view	RGB	Depth	Skeleton	Three-view	View1	View2	Two-view
MFN	0.5752	0.3978	0.6603	0.4769	0.6830	0.7630	0.6854	0.8159	0.5563	0.7141	0.7260
RC classifier	0.5990	0.5790	0.7850	0.6130	0.7829	0.8013	0.6765	0.8270	0.7660	0.7700	0.8190
MLSTM-FCN	0.6814	0.6914	0.7613	0.7555	0.7760	0.7929	0.6778	0.8284	0.8754	0.9246	0.9208
Concat-LSTM	-	-	-	0.7325	-	-	-	0.8330	-	-	0.8290
Concat-CNN	-	-	-	0.6132	-	-	-	0.8295	-	-	0.8919
Label-Concat	0.7124	0.7134	0.7585	0.8206	0.7304	0.8113	0.7235	0.8402	0.8643	0.8535	0.9090
Label-Average	0.7285	0.7114	0.7505	0.8156	0.7214	0.8090	0.7402	0.8319	0.8699	0.8559	0.8728
Label-Max	0.7575	0.7044	0.7615	0.8026	0.7239	0.8006	0.7287	0.8221	0.8704	0.9052	0.9113
C^{2}AF (Ours)	0.7615	0.7284	0.7645	**0.8406**	0.7248	0.8034	0.7347	**0.8688**	0.8656	0.9027	**0.9314**

Table 1: Classification performance on three datasets

where $r^{(o)}_{p,q}$ is the (p, q) element of $r^{(o)} \in \mathbb{R}^{K\times K\times 1}$ which is the o-th component of multi-channel tensor r ($1 \le o \le N_k$). $W^{(o)} \in \mathbb{R}^{1\times 1\times (V+\binom{V}{2})}$ and $b^{(o)} \in \mathbb{R}^{1\times 1}$ are the learnable weights and bias of 1×1 filter. $[r_{intra}, r_{inter}]_{p,q}$ represents the (p, q) element of cross-view correlation tensor concatenated by r_{intra} and r_{inter}. f is the activation function.

Through Eqs. (11-12), we formulate our fusion encoder E_f, and have $\phi_f = \{W, b\}$. We update ϕ_f by minimizing the following loss:

$$L^f = \sum_{i=1}^{N} \ell(Y_i, \hat{Y}_i^f), \tag{13}$$

where $\hat{Y}_i^f = C_f(T_{flatten}(r_i))$ is the prediction for the i-th sample. $T_{flatten}$ is a flatten operation to transfer feature matrix r_i into a vector, and $C_f : \mathbb{R}^{D_f} \to \mathbb{R}^K$ is the final classifier achieved by a linear mapping with $D_f = K \times K \times N_k$. During the training, we alternatively optimize the set of loss L^v for each view and L^f for the final classifier. The training procedure of our C^{2}AF is summarized in Algorithm 1.

Experiments

Experimental Setting

Datasets We utilize three real-world multi-view time series datasets to prove the model effectiveness.

- **EV-Action** (Wang et al. 2019) is a multi-view human action dataset. We choose RGB, depth, and skeleton views for our multi-view time series experiments. EV-Action contains 20 human common actions and 53 subjects performing each action 5 times, so that we have 5300 samples in total. We choose the first 40 subjects for training and the rest 13 subjects for test.
- **NTU RGB+D** (Shahroudy et al. 2016) is a large-scale dataset for multi-view action recognition. It includes 56000 action samples in 60 classes performed by 40 subjects. We choose the RGB, depth, and skeleton views for our experiments. We use the cross-subject benchmark provided by the original dataset paper, which contains 40320 samples for training and 16560 samples for test.
- **UCI Daily and Sports Activities** (Asuncion and Newman 2007) is a multivariate time series dataset, which includes the sensor data of 19 human actions. There are 45 sensors placed on subject's body. Each activity is performed by 8 subjects and has 480 samples. We follow the same mutli-view experimental setting from (Li, Li, and Fu 2016) in our model evaluation.

Baseline Methods Several comparison approaches including the state-of-the-art methods are deployed to demonstrate our model effectiveness.

- **MLSTM-FCN** (Karim et al. 2019) is a novel deep framework proposed to handle multivariate time series data, which achieves promising performances on extensive real-world time series datasets.
- **RC Classifier** (Bianchi et al. 2018) proposes a reservoir computing (RC) framework to encode multivariate time series data as a vectorial representation in an unsupervised fashion, which has a relatively low computational cost during handling temporal data.
- **MFN** (Zadeh et al. 2018) designs a memory fusion mechanism as an early fusion approach to tackle with multi-view time series.
- **Concat-LSTM/Concat-CNN** fuses multi-view time series using concatenation operation as input for LSTM and CNN. We use them as two early fusion baselines.
- **Label-Concat/Label-Average/Label-Max** fuses the predicted labels from multiple views using concatenation, average pooling, and max pooling, respectively. We utilize them as three late fusion baselines.

To adopt MLSTM-FCN and RC classifier for MVTSC, we concatenate multi-view time series along with the feature dimension as a multivariate time series for model input. MFN is designed for multi-view learning, we use it directly for model evaluation. We report the single-view and multi-view performances simultaneously for comparison except Concat-LSTM and Concat-CNN as they cannot provide single-view output.

Data Preprocessing We utilize the same strategy to preprocess multi-view data for EV-Action and NTU RGB+D as they both have RGB, depth, and skeleton views. Specifically, we align all the samples into the same 60 length with cutting and repeating strategies for longer and shorter samples. Next, we adopt TSN (Wang et al. 2016) to extract frame-level features for RGB view with pre-trained BNIncepction

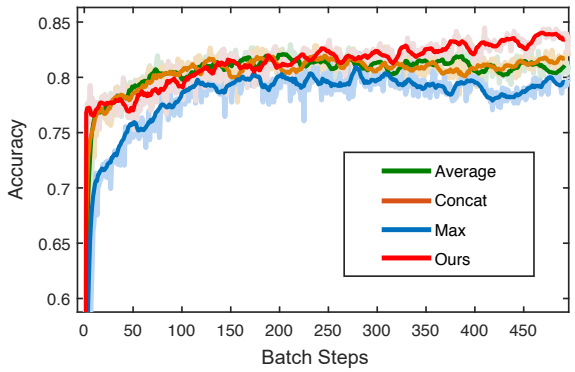

Figure 3: Comparisons between our model and late fusion baselines which prove that our channel-aware fusion is an effective and efficient fusion strategy. Shadow lines denote the exact performances per batch step, while the solid lines indicate the smoothed performances.

Setting	CCA	MvDA	MDBP	C^{2}AF (Ours)
Multi-view	25.90	67.24	78.96	93.14

Table 2: Comparison with traditional methods on UCI

backbone. The depth view is transferred into RGB format firstly using HHA algorithm (Gupta et al. 2014) and extract fetures using exactly the same strategy as RGB view. For skeleton view, we concatenate 3D coordinates of 25 joints at each time point as frame-level features. Specifically, in order to easily handle the large-scale skeleton data in NTU RGB+D dataset, we use VA-LSTM (Zhang et al. 2017) as the backbone to preprocess the 3D coordinates data. As a summary, for EV-Action and NTU RGB+D datasets, RGB, depth, and skeleton data are extracted as frame-level features with 60 temporal length and 1024, 1024, and 75 feature dimensions, respectively.

We follow the same data preprocessing procedure in (Li, Li, and Fu 2016) for UCI Daily Sports dataset. As a summary, the sensor data are set as View1 and View 2 with 125 temporal length, 27 and 18 feature dimensions, respectively.

Implementation

As shown in Figure 2, the frame-level features of each view are set as input of global-local temporal encoder simultaneously to obtain the view-specific representations. The outputs of global-temporal encoder and local-temporal encoder are concatenated as the input of view-specific classifier C_v. Each C_v is trained by optimizing its corresponding loss L_v. The predicted label from different views $\hat{Y}^v$ construct two sets of correlation matrices for capturing intra-view and inter-view label correlations. The cross-view correlative tensor is derived by stacking all the correlation matrices and fed into channel-aware learnable fusion module. Fused feature vector is set as input to train classifier C_f for final prediction through optimizing L^f. We set 128 as batch size. The Adam optimizer (Kingma and Ba 2014) is utilized for optimization and the learning rates are set as 0.0001 for all the view-specific and final classifiers synchronously. During the

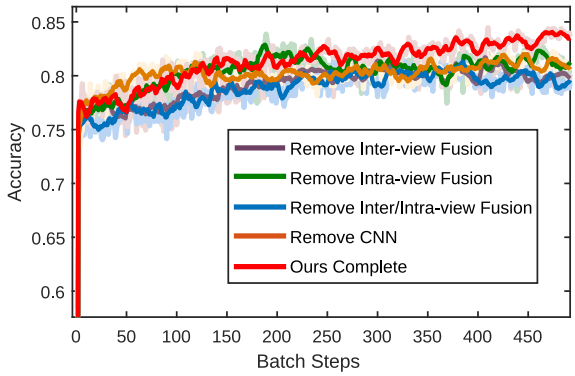

Figure 4: Ablation study on channel-aware learnable fusion. Shadow lines denote the exact performances per batch step, the solid lines indicate the smoothed performances.

training process, the classifiers of all views C_v are trained firstly to obtain the initial classification results which makes a concrete foundation for the learnable fusion module. Next, the final classifier C_f is trained based on the initial predicted labels. After that, C_v and C_f are trained alternatively during the whole training process and we report the single-view and final performances simultaneously. Our model is implemented using Tensorflow with GPU acceleration.

Performance Analysis

Classification performances for three datasets are shown in Table 1. For EV-Action dataset, the skeleton view is the most informative view achieving the best single-view performance. Other methods obtain comparable even better performances on single-view, however, our proposed model achieves the best multi-view performance. MFN cannot make early fusion efficiently to improve multi-view performance on EV-Action dataset which indicates the early fusion of MFN is not capable of handling high dimensional temporal data. However, our C^{2}AF will not suffer from this issue since we focus on extracting label correlations for multi-view fusion. RC classifier and MLSTM-FCN achieve competitive results on skeleton view but cannot effectively fuse multi-view information. The comparisons with three simple late fusion methods prove our learnable fusion is a more effective fusion strategy. We visualize the comparisons between late fusion baselines and our C^{2}AF in Figure 3, which shows the performance variations along with batch steps.

For NTU RGB+D dataset, the depth is the most informative view. All other approaches leverage the multi-view data to improve the final performance. However, our C^{2}AF fully explores the multi-view latent correlations and still achieves the best MVTSC performance.

For UCI dataset, View2 always obtains better results for single-view compared with View1. Other methods achieve competitive results for single-view but cannot outperform our fusion strategy. MFN improves the multi-view performance compared with single-view, however, it is still lower than our model. MLSTM-FCN obtains high performance for both single-view and multi-view, however, it cannot utilize multi-view data sufficiently for further improvement. Our

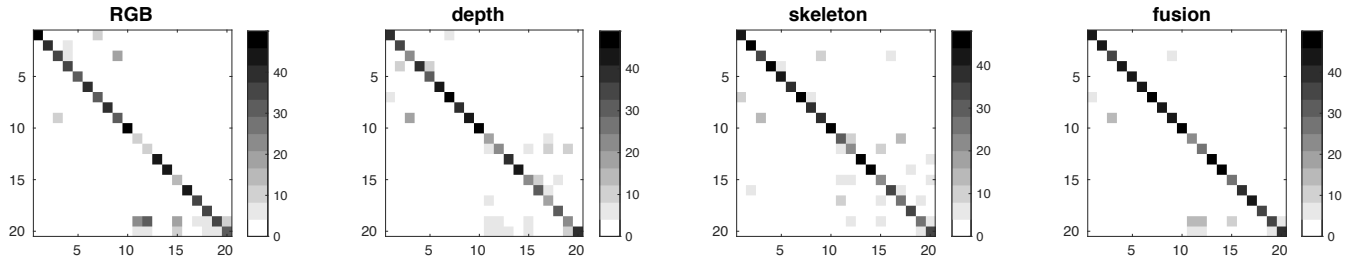

Figure 5: Confusion matrices for each single view and multi-view fusion on EV-Action dataset. The colorbar shows the color of corresponding prediction number. Being darker on the diagonal and being lighter off the diagonal indicate a better performance. The first 10 classes are performed by subjects themselves (*e.g.*, standing, walking, and jumping). The last 10 classes are performed with interactive objects (*e.g.*, moving table, reading book, and throwing ball).

Dataset	EV-Action			UCI	
View	RGB	Depth	Skeleton	View1	View2
Local Only	0.6263	0.6192	0.7735	0.8730	0.9001
Global Only	0.7104	0.7084	0.7665	0.7194	0.8292

Table 3: Temporal encoder ablations

proposed model achieves the best multi-view performance.

Moreover, several traditional approaches are proposed for multi-view learning, which focus on extracting the common features of multiple views. However, they cannot be efficiently applied on large-scale datasets (e.g., EV-Action and NTU RGB+D). To compare C^{2}AF with these approaches, we provide comparison experiments on UCI dataset as shown in Table 2. It includes three traditional methods: MDBP (Li, Li, and Fu 2016), MvDA (Kan et al. 2015), and CCA (Hotelling 1992). We find that leveraging on the great learning capacity of DNN, our C^{2}AF achieves the significant improvements compared with other approaches.

Ablation Study

We prove the necessity of each model component by conducting detailed ablation study. First, we use global and local temporal encoder individually to make view-specific classification on two datasets as shown in Table 3. Global encoder works better on EV-Action, while local encoder is better for UCI. Hence, our two-stream structure is indispensable to handle diverse time series data. It takes advantages of global and local encoders to obtain robust temporal representations.

We divide our learnable fusion module into several parts to make ablations. The whole fusion module can be separated as two parts, label correlative matrix and channel-aware fusion. Further, the label correlative matrix can be divided into intra-view and inter-view parts. The experimental results are shown in Table 4. **Intra-view Only/Inter-view Only** represents we only use intra-view/inter-view matrices. **Channel-aware Fusion Only** means we remove all the correlative matrices and concatenate predicted label vectors together as input to channel-aware fusion which proves the necessity of our whole correlative matrices. **Ours without Channel-aware fusion** indicates that we directly flatten all the correlation matrices into one feature vector as input to final classifier. The results illustrate each model component cannot obtain the best performance individually, while the complete C^{2}AF achieves the best accuracy. We visualize the performance curves of ablation in Figure 4, which shows the performance variations along with batch steps. To better understand how the learnable fusion process benefits the MVTSC, we show the classification confusion matrices for each single view and multi-view fusion on EV-Action in Figure 5. In EV-Action, classes can be divided into two groups (Wang et al. 2019): the first 10 actions are performed by subjects themselves, and the last 10 actions are performed interactively with other objects. RGB and depth views can accurately distinguish if the action is interactive, but easily make mistakes within each group. Skeleton view is not sensitive to the interactive objects so that it still makes mistakes cross these two groups. But its results are generally better than RGB and depth. We observe that our method takes full advantages of different views: it fuses the patterns from RGB and depth views to distinguish accurately if the action is interactive; it also benefits from the skeleton view to reduce the mistakes occurred within each group, so that our C^{2}AF achieves the most reasonable results.

Settings	EV-Action	UCI
Intra-view Only	0.8146	0.9206
Inter-view Only	0.8036	0.9279
Channel-aware Fusion Only	0.8046	0.9095
Ours without Channel-aware fusion	0.8206	0.9256
C^{2}AF (Ours-complete)	**0.8406**	**0.9323**

Table 4: Channel-aware learnable fusion ablations

Conclusions

In this study, we propose a novel end-to-end Correlative Channel-Aware Fusion (C^{2}AF) network for multi-view time series classification (MVTSC) problem. A global-local temporal encoder is developed to extract robust temporal representations for each single-view, and a learnable fusion strategy is proposed to fully explore the multi-view label information and boost the final performance. Extensive experiments on three public datasets prove the effectiveness of our model. A detailed ablation study further validates the necessity of each component in the proposed C^{2}AF network.

References

Asuncion, A.; and Newman, D. 2007. UCI machine learning repository. http://archive.ics.uci.edu/ml.

Atrey, P. K.; Hossain, M. A.; El Saddik, A.; and Kankanhalli, M. S. 2010. Multimodal fusion for multimedia analysis: a survey. *Multimedia systems* 16(6): 345–379.

Bankó, Z.; and Abonyi, J. 2012. Correlation based dynamic time warping of multivariate time series. *Expert Systems with Applications* 39(17): 12814–12823.

Bianchi, F. M.; Scardapane, S.; Løkse, S.; and Jenssen, R. 2018. Reservoir computing approaches for representation and classification of multivariate time series. *arXiv preprint arXiv:1803.07870* .

Bickel, S.; and Scheffer, T. 2004. Multi-view clustering. In *Proc. ICDM*, volume 4, 19–26.

Bruno, E.; and Marchand-Maillet, S. 2009. Multiview clustering: a late fusion approach using latent models. In *Proc. SIGIR*, 736–737. ACM.

Cai, Z.; Wang, L.; Peng, X.; and Qiao, Y. 2014. Multi-view super vector for action recognition. In *Proc. CVPR*, 596–603.

Cao, B.; Zheng, L.; Zhang, C.; Yu, P. S.; Piscitello, A.; Zulueta, J.; Ajilore, O.; Ryan, K.; and Leow, A. D. 2017. Deepmood: modeling mobile phone typing dynamics for mood detection. In *Proc. ACM SIGKDD*, 747–755.

Cortes, C.; and Vapnik, V. 1995. Support-vector networks. *Machine learning* 20(3): 273–297.

Cuaresma, J. C.; Hlouskova, J.; Kossmeier, S.; and Obersteiner, M. 2004. Forecasting electricity spot-prices using linear univariate time-series models. *Applied Energy* 77(1): 87–106.

Cui, Z.; Chen, W.; and Chen, Y. 2016. Multi-scale convolutional neural networks for time series classification. *arXiv preprint arXiv:1603.06995* .

Dorle, A.; Li, F.; Song, W.; and Li, S. 2020. Learning Discriminative Virtual Sequences for Time Series Classification. In *Proc. CIKM*, 2001–2004.

Fukunaga, K.; and Narendra, P. M. 1975. A branch and bound algorithm for computing k-nearest neighbors. *IEEE transactions on computers* 100(7): 750–753.

Gupta, S.; Girshick, R.; Arbeláez, P.; and Malik, J. 2014. Learning rich features from RGB-D images for object detection and segmentation. In *Proc. ECCV*, 345–360. Springer.

Harutyunyan, H.; Khachatrian, H.; Kale, D. C.; Steeg, G. V.; and Galstyan, A. 2017. Multitask learning and benchmarking with clinical time series data. *arXiv preprint arXiv:1703.07771* .

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Hotelling, H. 1992. Relations between two sets of variates. In *Breakthroughs in statistics*, 162–190. Springer.

Hou, C.; Zhang, C.; Wu, Y.; and Nie, F. 2010. Multiple view semi-supervised dimensionality reduction. *Pattern Recognition* 43(3): 720–730.

Hüsken, M.; and Stagge, P. 2003. Recurrent neural networks for time series classification. *Neurocomputing* 50: 223–235.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167* .

Jin, L.-p.; and Dong, J. 2016. Ensemble deep learning for biomedical time series classification. *Computational intelligence and neuroscience* 2016.

Kadous, M. W.; and Sammut, C. 2005. Classification of multivariate time series and structured data using constructive induction. *Machine learning* 58(2): 179–216.

Kan, M.; Shan, S.; Zhang, H.; Lao, S.; and Chen, X. 2015. Multi-view discriminant analysis. *IEEE transactions on pattern analysis and machine intelligence* 38(1): 188–194.

Karim, F.; Majumdar, S.; Darabi, H.; and Harford, S. 2019. Multivariate lstm-fcns for time series classification. *Neural Networks* 116: 237–245.

Keogh, E.; and Kasetty, S. 2003. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and Knowledge Discovery* 7(4): 349–371.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Lea, C.; Vidal, R.; Reiter, A.; and Hager, G. D. 2016. Temporal convolutional networks: A unified approach to action segmentation. In *Proc. ECCV*, 47–54.

Li, S.; Li, Y.; and Fu, Y. 2016. Multi-view time series classification: A discriminative bilinear projection approach. In *Proc. CIKM*, 989–998.

Li, S. Z.; Zhu, L.; Zhang, Z.; Blake, A.; Zhang, H.; and Shum, H. 2002. Statistical learning of multi-view face detection. In *Proc. ECCV*, 67–81. Springer.

Lin, M.; Chen, Q.; and Yan, S. 2013. Network in network. *arXiv preprint arXiv:1312.4400* .

Louis-Philippe, M.; Rada, M.; and Payal, D. 2011. Towards multimodal sentiment analysis: Harvesting opinions from the web. *Proc. International conference on multi-modal interfaces* 169–176.

Marteau, P.-F.; and Gibet, S. 2014. On recursive edit distance kernels with application to time series classification. *IEEE TNNLS* 26(6): 1121–1133.

Nie, F.; Cai, G.; and Li, X. 2017. Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Proc. AAAI*.

Nie, F.; Li, J.; Li, X.; et al. 2016. Parameter-Free Auto-Weighted Multiple Graph Learning: A Framework for Multiview Clustering and Semi-Supervised Classification. In *Proc. IJCAI*, 1881–1887.

Qi, C. R.; Su, H.; Nießner, M.; Dai, A.; Yan, M.; and Guibas, L. J. 2016. Volumetric and multi-view cnns for object classification on 3d data. In *Proc. CVPR*, 5648–5656.

Rabiner, L. R. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* 77(2): 257–286.

Ratanamahatana, C. A.; and Keogh, E. 2004. Making time-series classification more accurate using learned constraints. In *Proc. ICDM*, 11–22.

Rish, I.; et al. 2001. An empirical study of the naive Bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, 41–46.

Shahroudy, A.; Liu, J.; Ng, T.-T.; and Wang, G. 2016. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proc. CVPR*, 1010–1019.

Swoger, J.; Verveer, P.; Greger, K.; Huisken, J.; and Stelzer, E. H. 2007. Multi-view image fusion improves resolution in three-dimensional microscopy. *Optics express* 15(13): 8029–8042.

Tao, Z.; Liu, H.; Fu, H.; and Fu, Y. 2019. Multi-View Saliency-Guided Clustering for Image Cosegmentation. *IEEE Transaction on Image Processing* 28(9): 4634–4645.

Tao, Z.; Liu, H.; Li, S.; Ding, Z.; and Fu, Y. 2017. From Ensemble Clustering to Multi-View Clustering. In *Proc. IJCAI*, 2843–2849.

Tao, Z.; Liu, H.; Li, S.; Ding, Z.; and Fu, Y. 2020. Marginalized Multiview Ensemble Clustering. *IEEE Transactions on Neural Networks and Learning Systems* 31(2): 600–611.

Wang, H.; Meghawat, A.; Morency, L.-P.; and Xing, E. P. 2017. Select-additive learning: Improving generalization in multimodal sentiment analysis. In *Proc. ICME*, 949–954. IEEE.

Wang, L.; Ding, Z.; and Fu, Y. 2018. Learning Transferable Subspace for Human Motion Segmentation. In *Proc. AAAI*.

Wang, L.; Ding, Z.; and Fu, Y. 2019. Low-Rank Transfer Human Motion Segmentation. *IEEE Transactions on Image Processing* 28(2): 1023–1034. doi:10.1109/TIP.2018.2870945.

Wang, L.; Sun, B.; Robinson, J.; Jing, T.; and Fu, Y. 2019. EV-Action: Electromyography-Vision Multi-Modal Action Dataset. *arXiv preprint arXiv:1904.12602* .

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2016. Temporal segment networks: Towards good practices for deep action recognition. In *Proc. ECCV*, 20–36. Springer.

Wang, Q.; Ding, Z.; Tao, Z.; Gao, Q.; and Fu, Y. 2018. Partial Multi-view Clustering via Consistent GAN. In *Proc. ICDM*, 1290–1295.

Wei, L.; and Keogh, E. 2006. Semi-supervised time series classification. In *Proc. ACM SIGKDD*, 748–753.

Wörtwein, T.; and Scherer, S. 2017. What really matters—An information gain analysis of questions and reactions in automated PTSD screenings. In *Proc. ACII*, 15–20. IEEE.

Xi, X.; Keogh, E.; Shelton, C.; Wei, L.; and Ratanamahatana, C. A. 2006. Fast time series classification using numerosity reduction. In *Proc. ICML*, 1033–1040. ACM.

Xing, Z.; Pei, J.; and Keogh, E. 2010. A brief survey on sequence classification. *ACM SIGKDD Explorations Newsletter* 12(1): 40–48.

Xu, C.; Tao, D.; and Xu, C. 2013. A survey on multi-view learning. *arXiv preprint arXiv:1304.5634* .

Yao, H.; Wu, F.; Ke, J.; Tang, X.; Jia, Y.; Lu, S.; Gong, P.; Ye, J.; and Li, Z. 2018. Deep multi-view spatial-temporal network for taxi demand prediction. In *Proc. AAAI*.

Ye, L.; and Keogh, E. 2009. Time series shapelets: a new primitive for data mining. In *Proc. SIGKDD*, 947–956.

Yuan, Y.; Xun, G.; Ma, F.; Wang, Y.; Du, N.; Jia, K.; Su, L.; and Zhang, A. 2018. Muvan: A multi-view attention network for multivariate temporal data. In *Proc. ICDM*, 717–726. IEEE.

Zadeh, A.; Liang, P. P.; Mazumder, N.; Poria, S.; Cambria, E.; and Morency, L.-P. 2018. Memory fusion network for multi-view sequential learning. In *Proc. AAAI*.

Zhang, C.; Fu, H.; Hu, Q.; Cao, X.; Xie, Y.; Tao, D.; and Xu, D. 2020. Generalized Latent Multi-View Subspace Clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42(1): 86–99.

Zhang, C.; Han, Z.; cui, y.; Fu, H.; Zhou, J. T.; and Hu, Q. 2019. CPM-Nets: Cross Partial Multi-View Networks. In *Advances in Neural Information Processing Systems*, volume 32, 559–569.

Zhang, P.; Lan, C.; Xing, J.; Zeng, W.; Xue, J.; and Zheng, N. 2017. View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In *Proc. ICCV*, 2117–2126.

Zheng, Y.; Liu, Q.; Chen, E.; Ge, Y.; and Zhao, J. L. 2014. Time series classification using multi-channels deep convolutional neural networks. In *Proc. International Conference on Web-Age Information Management*, 298–310. Springer.

Zhou, J. T.; Pan, S. J.; Mao, Q.; and Tsang, I. W. 2012. Multi-view positive and unlabeled learning. *Journal of Machine Learning Research* .