applied sciences

MDPI

Article

Reconstructed Prototype Network Combined with CDC-TAGCN for Few-Shot Action Recognition

Aihua Wu and Songyu Ding *

School of Information Engineering, Shanghai Maritime University, Shanghai 201306, China; ahwu@shmtu.edu.cn
* Correspondence: 202130310067@stu.shmtu.edu.cn

Abstract: Research on few-shot action recognition has received widespread attention recently. However, there are some blind spots in the current research: (1) The prevailing practice in many models is to assign uniform weights to all samples; nevertheless, such an approach may yield detrimental consequences for the model in the presence of high-noise samples. (2) Samples with similar features but different classes make it difficult for the model to be distinguished. (3) Skeleton data harbors rich temporal features, but most encoders face challenges in effectively extracting them. In response to these challenges, this study introduces a reconstructed prototype network (RC-PN) based on a prototype network framework and a novel spatiotemporal encoder. The RC-PN comprises two enhanced modules: Sample coefficient reconstruction (SCR) and a reconstruction loss function (L_{RC}). SCR leverages cosine similarity between samples to reassign sample weights, thereby generating prototypes robust to noise interference and more adept at conveying conceptual essence. Simultaneously, the introduction of L_{RC} enhances the feature similarity among samples of the same class while increasing feature distinctiveness between different classes. In the encoder aspect, this study introduces a novel spatiotemporal convolutional encoder called CDC-TAGCN. The temporal convolution operator is redefined in CDC-TAGCN. The vanilla temporal convolution operator can only capture the surface-level characteristics of action samples. Drawing inspiration from differential convolution (CDC), this research enhances TCN to CDC-TGCN. CDC-TGCN allows for the fusion of discrepant features from action samples into the features extracted by the vanilla convolutional operator. Extensive feasibility and ablation experiments are performed on the skeleton action dataset NTU-RGB + D 120 and Kinetics and compared with recent research.

Keywords: few-shot; skeleton-based action recognition; prototype network; CDC-TAGCN

check for updates

Citation: Wu, A.; Ding, S. Reconstructed Prototype Network Combined with CDC-TAGCN for Few-Shot Action Recognition. *Appl. Sci.* **2023**, *13*, 11199. https://doi.org/10.3390/app132011199

Academic Editors: Antonio Fernández-Caballero and Douglas O'Shaughnessy

Received: 8 September 2023
Revised: 1 October 2023
Accepted: 10 October 2023
Published: 12 October 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the continuous advancement in deep learning and the growing richness of relevant datasets, the domain of action recognition has experienced significant progress. However, datasets in specific fields, such as security, sports, and medicine, are exceedingly scarce, and the associated costs of curating and annotating these datasets are prohibitively high. To tackle the scarcity of data, few-shot action learning [1] has come into existence. In action recognition, this is referred to as few-shot action recognition.

The task of few-shot action recognition is to train a model using a limited number of labeled samples and achieve accurate classification when confronted with novel classes that were not part of the training dataset. Currently, the state-of-the-art approach [2] combines prototype networks with deep learning frameworks, harnessing the powerful generalization capabilities of deep learning to enable models to address this challenge effectively.

The models related to few-shot action recognition primarily consist of the following modules: (a) An encoder module: it maps input data to a lower-dimensional feature space while maximizing the preservation of the information contained in the input data. (b) A prototype representation module: it generates prototypes for each class with support

datasets. These prototypes can be viewed as cluster centers. (c) A distance measurement module: unlike traditional classification models, models for few-shot learning require distance measurement to gauge the proximity between a query sample and prototypes. The closer the distance, the more similar they are. (d) A loss function module: it calculates the deviation in each feedforward and updates the parameters to bring the model closer to an ideal state. This paper collectively terms modules b, c, and d as the prototype network module, while treating the encoder as a standalone primary module.

Existing prototype networks have some limitations. In most cases, researchers generate prototypes for each class by computing the mean of the corresponding support sample features. However, this approach overlooks the potential disparities in quality among the samples. In data acquisition, each sample inevitably incurs some degree of noise interference. The averaging operation treats all samples as having equal weight coefficients. Constructing prototypes using highly noisy samples alongside other samples could lead to the deviation of prototypes from their ideal feature representation. This study proposes an improved prototype network, the reconstructed prototype network (RC-PN), which employs the cosine similarity coefficient reconstruction (SCR) approach to reassign the weights of samples so that prototypes can better encapsulate class characteristics. RC-PN not only alters the prototype's representation but also considers the problem of misclassification of query samples due to minimal differences in the characteristics of some action classes. Considering the above issue, this study proposes a loss function L_{RC}. L_{RC} amplifies the differences between prototypes of different classes with minimal differences while reducing the dissimilarity between samples within the same class. Lastly, sample labels are determined by nearest-neighbor calculations, thus requiring a distance metric. Since action samples are represented as time sequences, this study follows the approach used in DASTM [2], employing Soft-DTW [3] distance. Soft-DTW approximates the discrete values in the DTW formula with continuous values, allowing distance metrics to be trained and yield smoother results.

In terms of the encoder, its role is to transform sample features from high-dimensional to low-dimensional while preserving semantic information as much as possible. The action sequences encompass spatial attributes in the form of coordinate information and temporal attributes in the form of frame information. Traditional encoders (such as RNNs [4], CNNs [5,6], and LSTMs [7]) employ single-step convolutions, resulting in limitations in effectively fusing spatial and temporal features. Thus, this study selects the spatiotemporal encoder comprising spatial and temporal convolution processes. Regarding time convolution, previous research often employed simple 2D-CNN convolutions. However, vanilla convolution can only extract surface-level temporal features from the sequences. This study introduces the spatiotemporal encoder CDC-TAGCN, which leverages the advantages of CDC [8] principles, redefining the time convolution operator, known as CDC-TCN. CDC-TCN can compute the discrepancies between each feature within a convolution window and the window's mean value while performing vanilla convolution operations. These discrepant values capture the variations in the current convolution window's features. The method of mean-based computation enhances the model's ability to resist interference from temporal noise.

For action recognition datasets, they fall into two categories: video datasets and skeleton datasets. Skeleton datasets have the following advantages compared to video datasets: (1) Their data volume is smaller than the video dataset but contains rich and critical semantic information. (2) They remain unaffected by background and lighting conditions. This study chose NTU-RGB + D 120 [9] and Kinetics [10] as datasets; they are currently the most commonly used and challenging data sets.

The contributions of this paper can be summarized as follows:

(1) The SRC module of RC-PN leverages cosine similarity between samples to reconstruct the weight coefficients of samples, reducing the interference of high-noise samples.

(2) A novel loss is proposed to adjust the distances between different classes and within the same class, which solves the misclassification problem arising from minimal distributional differences.
(3) Introducing a novel spatiotemporal encoder called CDC-TAGCN, in which the time convolution operator, CDC-TCN, retains the vanilla convolutional feature information and extracts discrepant features during convolution.

2. Related Work

In this section, relevant work is presented from two main aspects: few-shot action recognition and spatiotemporal encoders.

Few-Shot Action Recognition: Few-shot learning is an application of meta-learning [11] in supervised learning, aiming to enable models to learn how to learn. It has been applied in various domains, such as face recognition [12], fault detection [13,14], and object recognition [15,16]. Zhu et al. [17] were among the early adopters of the few-shot concept in action recognition. They introduced CMN, which is based on memory networks. CMN uses multiple vectors to represent videos and presents a series of hidden saliency descriptors as constituent keys in storage. Although a layered structure improves efficiency, memory networks [17,18] consume significant storage space. As research progressed, relation networks [19], matching networks [20–22], and prototype networks [2,21,23] were gradually introduced. Jiang et al. [19] used a hybrid relation module to capture relationships between different episodic tasks and utilized the mean hausdorff metric to measure the distance between query and support samples. In one study by Careaga [21], an LSTM with a matching network framework was employed. This framework effectively extracted features from video sequences, and with the help of the matching network, query samples could find the nearest corresponding support samples. Relation networks and matching networks need to calculate all sample relationships, so the computational cost is very high. In Careaga et al.'s research, metric learning was divided into matching networks and prototype networks. The experiments demonstrate that prototype networks are better at focusing on class-specific features, and computational cost is significantly reduced. TAEN [23] feeds trajectory features into a framework that combines RepMet with prototype networks, and experiments demonstrate the strong generalization capabilities of prototype networks. The aforementioned studies are all based on video data, while recent research combines few-shot action recognition with skeleton data. In the latest DASTM study, spatial-temporal encoding was combined with prototype networks, and the more challenging Kinect-based and NTU-based datasets were introduced.

Spatiotemporal Encoder: The action dataset contains information in two dimensions: the topological relationships of joints in space and the temporal changes in joints. Early handcrafted features have been proven to lack generality and are only suitable for specific datasets [24]. Existing research has employed CNNs [5,6], RNNs [4], and LSTMs [7,21]. However, it is difficult for those single-step convolutional encoders to fuse temporal and spatial features. Franco first introduced the concept of graph neural networks (GNNs) [25]. Inspired by GNNs, Yan et al. [26] proposed spatiotemporal graph convolutional networks (ST-GCN) and applied them to skeleton-based action recognition, gaining significant attention. ST-GCN employs a two-step convolution: GCN to extract spatial features and TCN to extract temporal features, allowing it to handle non-Euclidean skeletal sequences. Subsequent work has built upon this foundation. Shi et al. [26] argued that different joints exhibit varying changes in various actions, and their topological relationships should not be static. They used two convolution kernels of different sizes to calculate the similarity between joints, forming an additional matrix representing the topological relationships. Chen et al. [27] believed that different channels might have different topological relationships and aimed to learn appropriate matrices while minimizing computational complexity automatically. Most research efforts have focused on improving the GCN module. However, this study points out limitations in feature extraction within the TCN module. The temporal convolution kernel of PYSKL [28] captures features in a multi-branch structure

but only captures surface-level features. The paper refers to the concept of CDC [8], fusing discrepant features into the original features and enhancing the model's ability to recognize action classes.

3. Method

Skeleton Sequence: A frame of a skeleton sequence can be defined as $G = \{X, A\}$, where $X \in R^{n \times h}$ is a matrix representing the sample feature, n is the number of joints, and h is the dimensionality of joints. $A \in R^{n \times n}$ is the topological matrix. Based on the above definition, a skeleton sequence with m frames can be described as $\Omega = \{G_1, G_2, \ldots, G_m\}$.

Few-shot Learning: In few-shot learning, the original dataset is divided into three subsets with non-overlapping classes: D_{train}, D_{val}, and D_{test}. Models for few-shot learning are usually trained, validated, and tested in an 'episodic' manner. Illustrating with D_{train}, the model randomly selects N classes from D_{train} and selects K examples from each class to create a support set, $S_{train}(N \times K)$. Then, L examples are chosen from the remaining samples of the same N classes to form a query set, $Q_{train}(N \times L)$. Due to the inherent similarity between samples from the same classes in the dataset, S_{train} can be used to achieve the task of recognizing query samples. Validation and testing share the same pattern.

$$S_{train} = \{(\Omega_i, y_i) | y_i \in C_{label}\}_i^{N\,K} \tag{1}$$

$$Q_{train} = \{(\Omega_i, y_i) | y_i \in C_{label}\}_i^{N\,L} \tag{2}$$

Ω and y_i represent sample features and sample labels, respectively. C_{label} is the set of all class labels in the dataset. The superscripts K and L represent the numbers of smaples. $S_{train} \cap Q_{train} = \varnothing$. The classes in D_{val} and D_{test} are different from those in D_{train}. D_{val} and D_{test} are used for validation and test without updating the model parameters.

3.1. Prototype Network (PN) and Reconstructed Prototype Network (RC-PN)

Prototype Network (PN): The prototype network generates a corresponding prototype C_k for each class in the support set. The generation process of C_k is shown in Formula (3).

$$C_k = \frac{1}{K} \sum_{(\Omega_i^S, y_i^S) \in S_k} f_\Phi\left(\Omega_i^S\right) \tag{3}$$

In the training phase, S_k is a subset of S_{train}. Ω_i^S denotes the $i-th$ input of the sample in support set S_k, f_Φ represents the encoder, and $f_\Phi\left(\Omega_i^S\right)$ represents the encoded features.

Samples from the same action class exhibit a high degree of similarity in their features. The prototype network employs the K-nearest neighbors approach to classify the samples in the query set with the following Formula (4):

$$P_\Phi\left(y = k \,\middle|\, \Omega_i^Q\right) = \frac{exp\left(-d\left(f_\Phi\left(\Omega_i^Q\right)\right), C_k\right)}{\sum_{k'} exp\left(-d\left(f_\Phi\left(\Omega_i^Q\right)\right), C_{k'}\right)} \tag{4}$$

Formula (4) utilizes the Softmax function to convert distances into probabilities. Here, $exp(\cdot)$ represents the exponential function. The function $d(\cdot)$ is the distance between query samples and prototypes. The negative sign is used to invert the magnitude relationship of values so that the smaller the distance, the greater the similarity. The distance can be Euclidean distance, Mahalanobis distance, and so on. However, these distance metrics may not be optimal for time series calculations. The DTW algorithm can better measure the distance between time series.

$$\Gamma(i,j) = E(i,j) + min\{\Gamma(i-1, j-1), \Gamma(i-1, j), \Gamma(i, j-1)\} \tag{5}$$

$E(i,j)$ represents the Frobenius norm of certain two moments in two sequences Ω_i and Ω_j. When $\Gamma(i,j) = \Gamma(m,m)$, it indicates the negative cumulative DTW distance between two sequences Ω_i and Ω_j, each consisting of m moments. In practice, in real deployment, we use Soft-DTW, which allows the $min\{\Gamma(i-1,j-1), \Gamma(i-1,j), \Gamma(i,j-1)\}$ to transition from being discrete to differentiable, thus resulting in a smoother outcome. For more details, please refer to the paper [3]. Figure 1 is a schematic diagram of the prototype network.

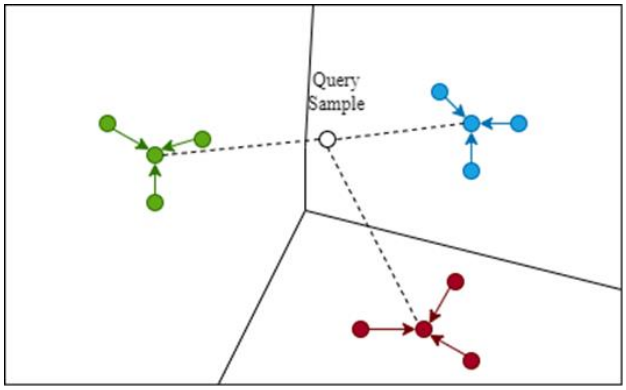

Figure 1. Different colors represent different classes, and the dotted line represents the distance between the query sample and prototypes. According to Equation (4), the closer to a prototype, the higher the probability of belonging to that class.

Formula (6) is the loss function of the prototype network:

$$L_{orgin} = -\frac{1}{L}\sum_{i}^{L} \log P_{\Phi}\left(\hat{y}_i = y_i \middle| \Omega_i^Q\right) \tag{6}$$

L represents the number of query samples, while $\hat{y}_i$ and y_i are the predicted and true labels for the query sample Ω_i^Q. When the $\hat{y}_i$ matches perfectly to y_i, the independent variable of the $\log(\cdot)$ function is 1, resulting in a total loss of 0.

Reconstructed Prototype Network (RC-PN): This section will introduce our reconstructed prototype network (RC-PN). RC-PN introduces SCR (sample coefficient reconstruction) and L_{RC} (reconstruction loss of distance). SCR makes the model more robust to noise interference. Additionally, L_{RC} enlarges the distances between class prototypes, leading to a more dispersed distribution of class prototypes and smaller differences between samples of the same class.

Existing prototypes are generated by taking the mean of all samples within a class. However, there are often high-noise samples within datasets, and generating prototypes by a mean-based approach can be easily influenced by noise, leading to destructive results. In SCR, the generation of prototypes abandons the mean-based approach and attempts to create new prototypes using the cosine similarity between query samples and support set samples. The cosine similarity between query and support samples is denoted as Sim and is calculated using the following Formula (7):

$$Sim\left(f_{\Phi}\left(\Omega_i^S\right), f_{\Phi}\left(\Omega_i^Q\right)\right) = \frac{f_{\Phi}\left(\Omega_i^S\right) \cdot f_{\Phi}\left(\Omega_i^Q\right)}{\left|\left|f_{\Phi}\left(\Omega_i^S\right)\right|\right| \cdot \left|\left|f_{\Phi}\left(\Omega_i^Q\right)\right|\right|} \tag{7}$$

Larger *Sim* values imply greater similarity between two features. With a larger *Sim* value, the corresponding weight α_i of the support sample should also be greater, α_i calculated as follows:

$$\alpha_i = \frac{exp\left(Sim\left(f_\Phi(\Omega_i^S), f_\Phi\left(\Omega_i^Q\right)\right)\right)}{\sum_{\Omega_j^S \in S_k} exp\left(Sim\left(f_\Phi\left(\Omega_j^S\right), f_\Phi\left(\Omega_i^Q\right)\right)\right)} \tag{8}$$

Figure 2 is a schematic diagram of the new prototypes generated by SCR. Further scrutinizing the Formula (8), α_i varies with variations in the query sample Ω_i^Q. In other words, the prototypes are adaptively generated based on the query sample. The new prototype generation formula becomes:

$$C_k = \sum_{\Omega_j^S \in S_k} \alpha_i f_\Phi\left(\Omega_i^S\right) \tag{9}$$

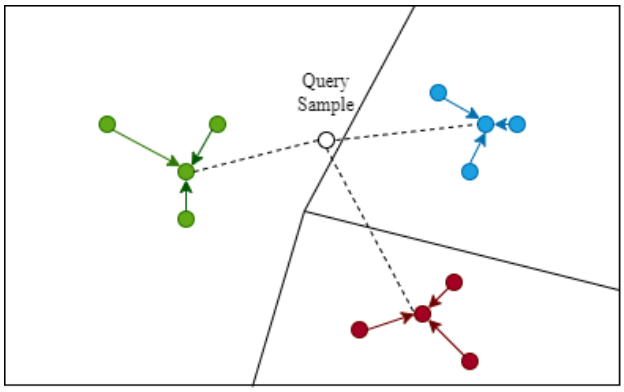

Figure 2. In contrast to Figure 1, this figure illustrates how SCR enables query samples that were originally misclassified into the blue class to be correctly classified into the green class.

In the course of our research, we observed the presence of similar samples belonging to different classes. Logically, similar samples cause their respective feature matrices to be very close in value, resulting in their scattering within an overlapping space. This study incorporated a reconstruction distance loss, denoted as L_{RC}. Unlike triplets loss [29], L_{RC} does not require label involvement and is computationally simpler. L_{RC} introduces a reward-penalty mechanism for distances into the original loss function, making the features of prototypes' intra-class samples more similar, bringing them closer in the same space. Additionally, it pushes features of samples from different classes farther apart, enhancing the overall dispersion of samples in the feature space. The L_{RC} loss formula is as follows:

$$L_{RC} = \begin{cases} d(F_b, C_k) & k = b \\ max(0, \Delta - d(F_b, C_k)) & k \neq b \end{cases} \tag{10}$$

$F_b = f_\Phi\left(\Omega_i^S\right)$, where the subscript b indicates that the label of a sample belongs to class b, and C_k represents the prototype for class k. When $k = b$, it indicates that the support sample and the class represented by C_k are of the same class. In this case, adding distance between the support sample and the prototype C_k to the loss function aims to gradually bring the support samples closer to the prototype. When $k \neq b$, it means that the support sample and C_k do not belong to the same class. In this scenario, Δ represents the allowed distance between F_b and C_k. If the distance does not exceed Δ, it is recorded as 0; otherwise, it is taken as $\Delta - d(F_c, C_k)$. This ensures that the support samples move farther from other prototypes during iterations, making prototypes more dispersed. Figure 3 is a schematic diagram of L_{RC}.

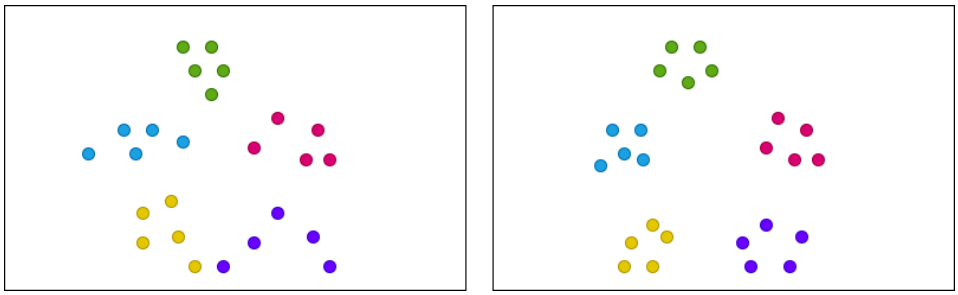

Figure 3. The visualization illustrates the effect of L_{RC}, as observed from the results on the right. L_{RC} enhances the dispersion between classes while bringing samples of the same class closer together.

The final loss function Formula (11) is as follows, λ controls the contribution:

$$L_{total} = -\frac{1}{|Q|}\sum_{i}^{|Q|} \log P_{\Phi}\left(\hat{y}_i = y_i \middle| \Omega_i^Q\right) + \lambda L_{RC} \tag{11}$$

Figure 4 summarizes the process of the prototype network part.

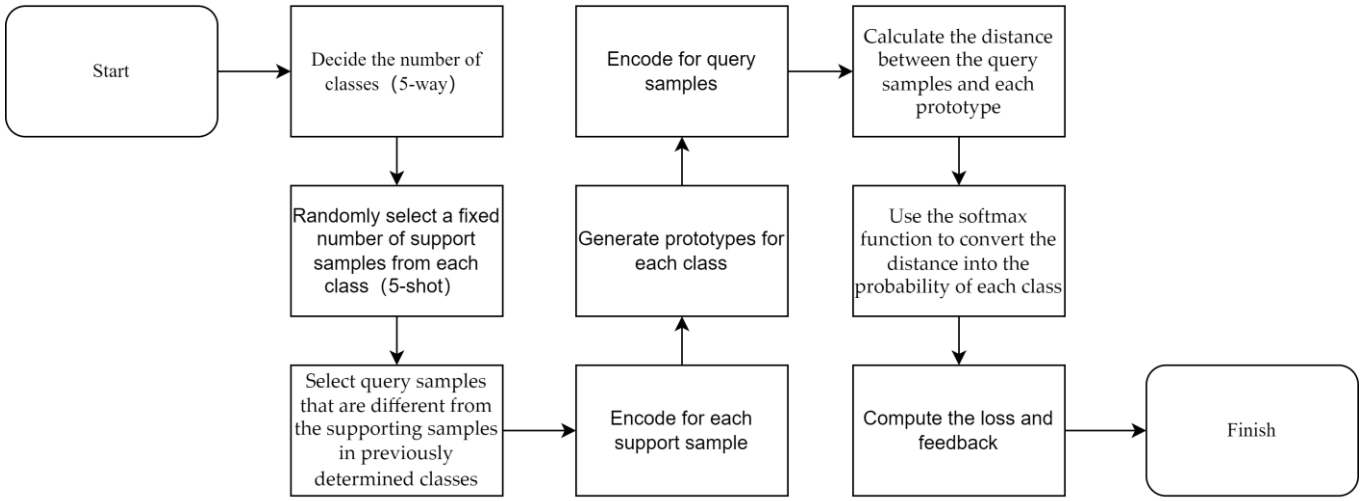

Figure 4. This figure takes a 5-way 5-shot task as an example to show the process of the prototype network in the training phase. The content of the encoder will be introduced in detail in the next section.

3.2. Spatiotemporal Encoder and CDC-TAGCN

This section will focus exclusively on the spatiotemporal encoder of this study. The study introduces DCD-TAGCN to enhance the encoder's ability to capture richer features from spatiotemporal sequences.

ST-GCN [26]: The spatiotemporal encoder is a multi-step convolutional network. Due to the unique topological structure of skeleton sequences, traditional CNNs and RNNs make it difficult to encode skeleton features effectively. Therefore, Yan et al. [26] introduced a spatiotemporal encoder that divided the convolution into two steps: CGN and TCN. In the GCN step, most research divides the joints into centrifugal, source, and centripetal joints based on biomechanics. At this point, the adjacency matrix A can be decomposed into the centrifugal matrix $A_1 = \{a^1{}_{ij}\} \in R^{N\times N}$, the source matrix $A_2 = \{a^2{}_{ij}\} \in R^{N\times N}$, and the centripetal matrix $A_3 = \{a^3{}_{ij}\} \in R^{N\times N}$.

After partitioning the adjacency matrix A using spatial strategies, the convolution Formula (12) for GCN is computed as follows:

$$f_{gcn} = \sum_{j}^{n} D^{-\frac{1}{2}} A_j D^{-\frac{1}{2}} \Omega W_j^g \tag{12}$$

D is the degree matrix of A, and W_j^g are the coefficients of the convolutional kernels in GCN, where the kernel size is 1×1. GCN attempts to use three 1×1 convolutions to find the relationships between centrifugal, source, and centripetal joints.

TCN is used to capture temporal features, and the computation process is as shown in Formula (12). TCN uses a $T \times 1$ convolutional kernel.

$$f_{out} = \sum_{t}^{T} f_{gcn} w^t \tag{13}$$

DCD-TAGCN: In this study, the encoder primarily focuses on improving TCN. For the GCN part, considering that A alone may not enable the model to capture rich skeletal topological relationships, this paper employs the spatial convolution approach of AGCN [26].

$$f_{gcn} = \sum_{j}^{n} D^{-\frac{1}{2}} (A_j + M_j + J_k) D^{-\frac{1}{2}} \Omega W_j^g \tag{14}$$

$A \in R^{n \times n}$ is a static matrix recording the original skeleton topological relationship. $M \in R^{n \times n}$ is a learnable matrix that is continuously updated with iteration. $J \in R^{n \times n}$ uses two Gaussian convolution kernels (1×1 conv) to find the similarity between the joints of each sample and then saves the values in J. Both M and J are matrices obtained through learning. The difference is that J varies with the characteristics of the sample, while M focuses on the characteristics of the overall dataset.

This study argues that for skeleton sequences, the TCN convolution operator may not fully capture the temporal characteristics of the skeleton. The convolution Formula (13) can be expressed in full as: $f_{out} = \sum_{p_n \in R} w^t(p_n) \cdot (f_{gcn}(p_0 + p_n))$. It indicates that the convolution kernel multiplies point-wise according to the position in the current convolution window. R represents the range of the convolution kernel, which is $T \times 1$ in size. p_n represents the coefficient position in the convolution window, p_0 is the current convolution central point, and $(p_0 + p_n)$ represents the position within the current convolution window. With each slide of the convolution window, it calculates the features within a specific time interval. However, there is additional information in the sequences that can be used to assist in recognizing action classes. Following the idea of the CDC [8], this paper calculates the discrepancy of the current convolution window. In detail, subtract the feature at position $(p_0 + p_n)$ from the feature position p_0 at the current convolution to obtain a discrepant feature between these two positions. In actual computation, the discrepant feature is calculated as the discrepant between the feature of each convolution point and the mean feature of the entire convolution window. This approach enhances the robustness of obtaining discrepant features over time. The convolution process can be expressed using Formula (15).

$$\begin{aligned} f_{out} &= \theta \sum_{p_n \in R} w^t(p_n) \cdot \left(f_{gcn}(p_0 + p_n) - Avg\left(\ldots, f_{gcn}\left(p_0^{t-1}\right), f_{gcn}(p_0), f_{gcn}\left(p_0^{t+1}\right), \ldots \right) \right) + (1-\theta) \sum_{p_n \in R} w^t(p_n) \cdot (f_{gcn}(p_0 + p_n)) \\ &= \sum_{p_n \in R} w^t(p_n) \cdot (f_{gcn}(p_0 + p_n)) + \theta \left(-Avg\left(\ldots, f_{gcn}\left(p_0^{t-1}\right), f_{gcn}(p_0), f_{gcn}\left(p_0^{t+1}\right), \ldots \right) \sum_{p_n \in R} w^t(p_n) \right) \end{aligned} \tag{15}$$

$\theta \left(-Avg\left(\ldots, f_{gcn}\left(p_0^{t-1}\right), f_{gcn}(p_0), f_{gcn}\left(p_0^{t+1}\right), \ldots \right) \sum_{p_n \in R} w^t(p_n) \right)$ represents the difference feature. p_0^{t+1} means at the next moment of p_0. $\sum_{p_n \in R} w^t(p_n) \cdot (f_{gcn}(p_0 + p_n))$ represents the vanilla convolutional feature.

The structure of the encoder is shown in Figure 5.

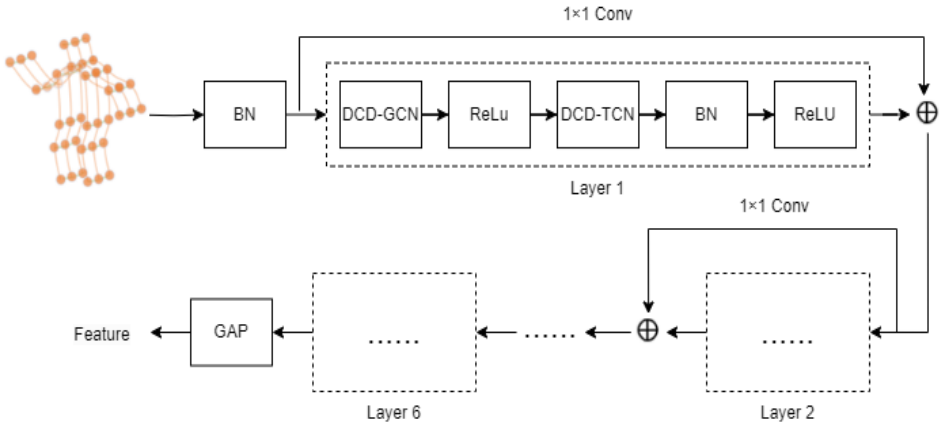

Figure 5. This is the framework diagram of the encoder, which consists of a total of 6 layers. The 1×1 Conv represents the residual connections used between layers to improve model fitting. The features output by the encoder are then input into RC-PN.

4. Experiments

4.1. Dataset

NTU RGB + D 120 [9]: This is an action dataset established by the Rose Laboratory, containing 3D joint coordinates for human action recognition tasks. It comprises 114,480 samples, each with 25 body joints and 120 action classes. In the experiments conducted in this paper, the 120 action categories are divided into 80 for training, 20 for validation, and 20 for testing. For each class, 30 and 60 samples are randomly selected, denoted as two subsets, "NTU-S" and "NTU-T,", respectively.

Kinetics [10]: The Kinetics dataset originates from YouTube videos. This dataset comprises 260,232 videos across 400 different classes. For each video, the skeleton data is extracted using the OpenPose algorithm, retaining the coordinates of 18 body joints as the initial joint features. Each joint contains a 2D joint coordinate and a confidence score. This experiment utilizes only the first 120 action categories, with 100 samples per class.

The selection method for these two datasets is consistent with DASTM to ensure experimental rigor, and these two datasets are the most commonly used standard datasets in skeleton recognition.

4.2. Experimental Setup

All models were trained using the Adam optimizer with an initial learning rate of 0.001. During training, 500 episodes were randomly sampled, and 200 episodes for validation. During testing, 200 episodes were randomly sampled. Testing involved conducting ten tests to calculate the mean accuracy with standard deviation. Additionally, all experiments were built using PyTorch 1.11.0 and executed on a Nvidia (Santa Clara, CA, USA) GeForce RTX 4090 GPU with Ubuntu 20.04.

4.3. Result

This paper will use DASTM (ECCV 2022) (DASTM composed of PN and AGCN; compared to our approach, PN and TCN are improved versions) as the baseline. One thing worth noting is that while there are some researchers in the field of skeleton action recognition, the dataset standards and experimental metrics could not be quite consistent. DASTM, as a recent research advancement, has elevated the challenge by using 100 classes for training, with 20 classes designated for validation and testing. Regarding parameter settings, the original AGCN encoder had nine layers. However, in the case of small samples, it is believed that there is no need for so many parameters, as having too many parameters can lead to overfitting. Therefore, this study sets the number of layers for AGCN to six. Table 1 displays information about the layers in seven rows, but most encoders in the skeleton recognition field do not count the initial layer as part of the total layers. This study

uses a structure with two layers of 64 channels, two layers of 128 channels, and two layers of 256 channels. This structure can initially capture some low-level features, and then, as the layers deepen and the number of channels increases, the model starts capturing more complex features.

Table 1. Detailed parameters within the encoder.

Channels (Input)	Channels (Output)	Numbers/Size (CDC-GCN)	Number/Size (CDC-TCN)
3	64	3/1*1	1/9*1
64	64	3/1*1	1/9*1
64	64	3/1*1	1/9*1
64	128	3/1*1	1/9*1
128	128	3/1*1	1/9*1
128	256	3/1*1	1/9*1
256	256	3/1*1	1/9*1

Tables 2 and 3 display the results for the 5-way-1-shot task and 5-way-5-shot task. The results in the tables are presented as mean + variance after ten tests run. From the experimental results in Table 1, it can be observed that even with only one sample, this paper's method has achieved good performance, demonstrating the feasibility of this approach.

Table 2. 5-way-1-shot top-1 acc.

Methods	NTU-S	NTU-T	Kinetics
DASTM(PN + AGCN)	71.26% + 0.32%	73.86% + 0.22%	40.82% + 0.32%
RC-PN + AGCN	71.38% + 0.27%	74.37% + 0.11%	41.02% + 0.26%
PN + CDC-TAGCN	72.32% + 0.24%	74.47% + 0.20%	41.32% + 0.33%
RC-PN + CDC-TAGCN	73.69% + 0.35%	74.63% + 0.19%	41.47% + 0.50%

Table 3. 5-way-5-shot top-1 acc.

Methods	NTU-S	NTU-T	Kinetics
DASTM(PN + AGN)	83.20% + 0.25%	86.48% + 0.21%	50.81% + 0.13%
RC-PN + AGCN	83.72% + 0.19%	86.91% + 0.18%	50.96% + 0.16%
PN + CDC-TAGCN	83.92% + 0.24%	86.82% + 0.23%	50.90% + 0.18%
RC-PN + CDC-TAGCN	84.59% + 0.08%	86.97% + 0.27%	51.29% + 0.14%

To ensure fairness, the same seed and parameters were used under the same dataset for consistency. In the 5-way-1-shot task, where only one sample is available; sample coefficient reconstruction is not possible, so only L_{RC} is active in RC-PN. The 5-way-5-shot task provides a complete demonstration of this paper's method's performance. Tables 2 and 3 showcase the improvements brought by RC-PN and CDC-TACN, with both experiments using DASTM as the baseline.

Both NTU-S and NTU-T are based on NTU RGB + D 120. The difference is that the number of samples collected in each class in NTU-T is twice that of NTU-S. With fixed parameters, NTU-S and NTU-T have at most 30 and 60 class prototypes per class in the task 5-way-1-shot, and at most C_{30}^{5}, C_{60}^{5} in the 5-way-5-shot task. Under the baseline DASTM, in the 5-way-1-shot task, the average top-1 accuracy of the model in NTU-T is 73.86%, and that of NTU-S is 71.26%, with a difference of 2.6%. The difference in 5-way-5-shot task is 3.28%. The model of this study reduces the gap between the two types of tasks in the dataset to 0.94%, 2.38%, which is lower than the original 2.6%, 3.28%. This means that if the model can be continuously optimized, multi-sample performance can be achieved even with one sample. The Kinetics dataset has only 18 joints, so it is not as rich in semantic information as the NTU series dataset. In addition, it is a dataset generated based on video websites and OpenPose, resulting in a lot of noise, which is extremely challenging.

4.4. More Detail

In this section, (parameter settings, $\theta = 0.1$, $\lambda = 0.001$, $\Delta = 2$), a more detailed investigation of the proposed method will be conducted on the NTU-S dataset, with a primary focus on the 5-way-5-shot task. The main subjects of exploration will be the SCR and L_{RC} within RC-PN.

From the observations in Table 4, the performance improvement of SCR is greater than that of L_{RC}. The performance of SCR validates the previous hypothesis: for prototype networks, the contribution of a sample should be determined by its own quality rather than the average. Regarding L_{RC}, if L_{RC} is too large, it will completely disrupt the original space layout, and if L_{RC} is too small, the module may lose its meaning. The value of Δ here is slightly larger than the distance within the class. It is believed in this study that in cases where most samples are classified correctly, L_{RC} should play a role not as the decisive factor but for fine-tuning. Therefore, if L_{RC} occupies a significant portion of the loss in the model, it may cause a convergence problem and lead to more misclassifications. The Figure 6 shows the total loss and L_{RC} loss of the model on the NTU-S dataset.

Table 4. Ablation of RC-PN.

Methods	NTU-S
RC-PN + AGCN	83.72% + 0.19%
(RC-PN + AGCN)/L_{RC}	83.40% + 0.22%
(RC-PN + AGCN)/SCR	83.26% + 0.15%

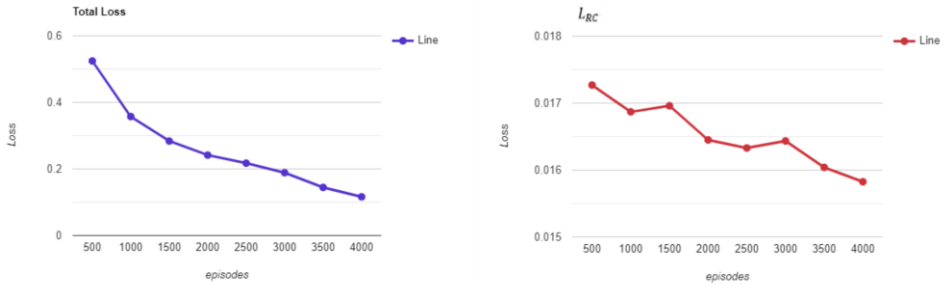

Figure 6. From the figure, it can be seen that after the model converges, L_{RC} approximately accounts for about one tenth of the total loss.

For the CDC-TACN module, θ controls the contribution of discrepant features. With other modules unchanged, this paper explores the trend of model performance with varying θ values, as shown in the Figure 7. From the table, it can be observed that when θ is set to 0, the temporal convolution operator degenerates into vanilla convolution. When θ values are set to 0.1 and 0.2, the model's performance surpasses that of the regular operator. As θ continues to increase, the concepts extracted by vanilla convolution become increasingly blurred, resulting in poor performance when θ is set to 0.4.

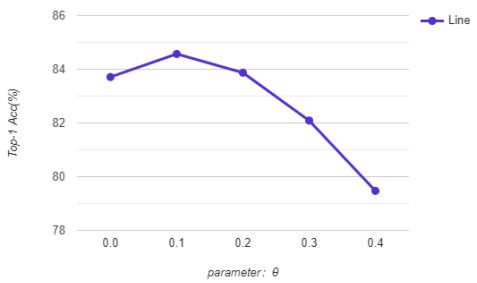

Figure 7. Model performance as a function of θ.

This study conducted experimental comparisons of three distances: Euclidean distance, Manhattan distance, and DTW. Experiments have proven that DTW takes advantage of its

dynamic alignment function under the 5-way-1-shot. As more samples are added, more noise is added to the calculation, and the complex calculation process of DTW makes its vulnerability begin to show. However, with a small number of samples, the performance of DTW is far better than others, which is also consistent with the situation of restricted samples, so DTW is the best choice for this study. The relevant experimental results are shown in Table 5.

Table 5. Experiment on distance.

Methods	5-way-1-shot	5-way-5-shot
Euclidean Distance	69.64% + 0.89%	84.42% + 0.29%
Manhattan Distance	72.24% + 0.67%	85.62% + 0.39%
DTW	73.69% + 0.35%	84.59% + 0.08%

The last experiment is about the number of layers. When the number of layers is three, there is one layer each for 64, 128, and 258. When the number of layers is 6, there are two layers each for 64, 128, and 258. When the number of layers is nine, there are three layers each for 64, 128, and 258. This study selected six layers under comprehensive consideration. The reasons for selecting six layers are based on two aspects: The one is whether in 5-way-1-shot or 5-way-5-shot, three layers, six layers, and nine layers are not linearly related to the accuracy, and there is no obvious improvement in stability; and even it may be that in some cases, the encoder with a three-layer structure is better than the latter two. Another consideration is that the increase in the number of layers brings about a substantial increase in the number of parameters. These unnecessary parameters bring iteration difficulties. The results are shown in Table 6.

Table 6. Experiment on layers.

Layers	5-way-1-shot	5-way-5-shot
3 layers	72.42% + 0.42%	83.84% + 0.12%
6 layers	73.69% + 0.35%	84.59% + 0.08%
9 layers	73.12% + 0.26%	83.62% + 0.12%

5. Conclusions

This article introduces a new prototype network, RC-PN, and a new spatiotemporal encoder, CDC-TAGCN. In RC-PN, SCR adjusts the weighting coefficients of samples using cosine similarity to avoid the interference of high-noise samples effectively. Moreover, the inclusion of L_{RC} increases the distance between prototypes of different classes, making samples within the same class more similar. The temporal convolution operator in CDC-TAGCN can provide discrepant features on top of regular convolution, greatly enriching the meaning of features.

However, there are still some limitations in the research, such as whether the vulnerability of DTW to noise can be improved or whether there is a better distance measurement method. This will be our future work.

There needs to be more research on few-shot action recognition with skeleton datasets. This is a brand-new, promising, and practically meaningful direction, and we hope to see more researchers joining.

Author Contributions: Methodology, A.W. and S.D.; Software, S.D.; Validation, A.W. and S.D. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, F.; Fergus; Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*; Nice, France, 13–16 October 2003, IEEE: New York, NY, USA, 2003; pp. 1134–1141.
2. Ma, N.; Zhang, H.; Li, X.; Zhou, S.; Zhang, Z.; Wen, J.; Li, H.; Gu, J.; Bu, J. Learning spatial-preserved skeleton representations for few-shot action recognition. In Proceedings of the European Conference on Computer Vision, Tel Aviv, Israel, 23–27 October 2022; Springer: Berlin/Heidelberg, Germany, 2022; pp. 174–191.
3. Cuturi, M.; Blondel, M. Soft-dtw: A differentiable loss function for time-series. In Proceedings of the International Conference on Machine Learning (PMLR), Sydney, Australia, 6–11 August 2017; pp. 894–903.
4. Du, Y.; Wang, W.; Wang, L. Hierarchical recurrent neural network for skeleton based action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1110–1118.
5. Byeon, Y.-H.; Kim, D.; Lee, J.; Kwak, K.-C. Body and hand–object ROI-based behavior recognition using deep learning. *Sensors* **2021**, *21*, 1838. [CrossRef] [PubMed]
6. Ren, J.; Reyes, N.; Barczak, A.; Scogings, C.; Liu, M. Towards 3D human action recognition using a distilled CNN model. In Proceedings of the 2018 IEEE 3rd International Conference on Signal and Image Processing (ICSIP), Shenzhen, China, 13–15 July 2018; IEEE: New York, NY, USA, 2018; pp. 7–12.
7. Liu, J.; Shahroudy, A.; Xu, D.; Kot, A.C.; Wang, G. Skeleton-based action recognition using spatio-temporal LSTM network with trust gates. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *40*, 3007–3021. [CrossRef] [PubMed]
8. Yu, Z.; Zhao, C.; Wang, Z.; Qin, Y.; Su, Z.; Li, X.; Zhou, F.; Zhao, G. Searching central difference convolutional networks for face anti-spoofing. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 5295–5305.
9. Liu, J.; Shahroudy, A.; Perez, M.; Wang, G.; Duan, L.-Y.; Kot, A.C. Ntu rgb+ d 120: A large-scale benchmark for 3d human activity understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 2684–2701. [CrossRef] [PubMed]
10. Kay, W.; Carreira, J.; Simonyan, K.; Zhang, B.; Hillier, C.; Vijayanarasimhan, S.; Viola, F.; Green, T.; Back, T.; Natsev, P.; et al. The kinetics human action video dataset. *arXiv* **2017**, arXiv:1705.06950.
11. Glass, G.V. Primary, secondary, and meta-analysis of research. *Educ. Res.* **1976**, *5*, 3–8. [CrossRef]
12. Meddad, M.; Moujahdi, C.; Mikram, M.; Rziza, M. Convolutional Siamese neural network for few-shot multi-view face identification. *Signal Image Video Process.* **2023**, *17*, 3135–3144. [CrossRef]
13. Zhang, K.; Wang, Q.; Wang, L.; Zhang, H.; Zhang, L.; Yao, J.; Yang, Y. Fault diagnosis method for sucker rod well with few shots based on meta-transfer learning. *J. Pet. Sci. Eng.* **2022**, *212*, 110295. [CrossRef]
14. Zou, F.; Sang, S.; Jiang, M.; Li, X.; Zhang, H. Few-shot pump anomaly detection via Diff-WRN-based model-agnostic meta-learning strategy. *Struct. Health Monit.* **2023**, *22*, 2674–2687. [CrossRef]
15. Lin, C.; Gao, F. An Extension of Prototypical Networks. In Proceedings of the 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 12–14 June 2020; IEEE: New York, NY, USA, 2020; Volume 1, pp. 421–425.
16. Xie, Z.; Duan, P.; Liu, W.; Kang, X.; Wei, X.; Li, S. Feature consistency-based prototype network for open-set hyperspectral image classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**. [CrossRef] [PubMed]
17. Zhu, L.; Yang, Y. Compound memory networks for few-shot video classification. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 751–766.
18. Zhang, L.; Chang, X.; Liu, J.; Luo, M.; Prakash, M.; Hauptmann, A.G. Few-shot activity recognition with cross-modal memory network. *Pattern Recognit.* **2020**, *108*, 107348. [CrossRef]
19. Jiang, L.; Yu, J.; Dang, Y.; Chen, P.; Huan, R. HiTIM: Hierarchical Task Information Mining for Few-Shot Action Recognition. *Appl. Sci.* **2023**, *13*, 5277. [CrossRef]
20. Guo, M.; Chou, E.; Huang, D.-A.; Song, S.; Yeung, S.; Li, F. Neural graph matching networks for fewshot 3d action recognition. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 653–669.
21. Careaga, C.; Hutchinson, B.; Hodas, N.; Phillips, L. Metric-based few-shot learning for video action recognition. *arXiv* **2019**, arXiv:1909.09602.
22. Xing, E.; Jordan, M.; Russell, S.J.; Ng, A. Distance metric learning with application to clustering with side-information. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2002; Volume 15.
23. Ben-Ari, R.; Nacson, M.S.; Azulai, O.; Barzelay, U.; Rotman, D. TAEN: Temporal aware embedding network for few-shot action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 2786–2794.
24. Wang, L.; Huynh, D.Q.; Koniusz, P. A comparative review of recent kinect-based action recognition algorithms. *IEEE Trans. Image Process.* **2019**, *29*, 15–28. [CrossRef] [PubMed]
25. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2008**, *20*, 61–80. [CrossRef] [PubMed]
26. Yan, S.; Xiong, Y.; Lin, D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.

27. Chen, Y.; Zhang, Z.; Yuan, C.; Li, B.; Deng, Y.; Hu, W. Channel-wise topology refinement graph convolution for skeleton-based action recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 13359–13368.
28. Duan, H.; Wang, J.; Chen, K.; Lin, D. PYSKL: Towards Good Practices for Skeleton Action Recognition. *arXiv* **2022**, arXiv:2205.09443.
29. Hermans, A.; Beyer, L.; Leibe, B. In defense of the triplet loss for person re-identification. *arXiv* **2017**, arXiv:1703.07737.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.