ERA: Expert Retrieval and Assembly for Early Action Prediction

Lin Geng Foo[1⋆], Tianjiao Li[1⋆], Hossein Rahmani[2], Qiuhong Ke[3], and Jun Liu[1⋆⋆]

[1] ISTD Pillar, Singapore University of Technology and Design
{lingeng_foo,tianjiao_li}@mymail.sutd.edu.sg, jun_liu@sutd.edu.sg
[2] School of Computing and Communications, Lancaster University
h.rahmani@lancaster.ac.uk
[3] Department of Data Science & AI, Monash University
qiuhong.ke@monash.edu

Abstract. Early action prediction aims to successfully predict the class label of an action before it is completely performed. This is a challenging task because the beginning stages of different actions can be very similar, with only minor subtle differences for discrimination. In this paper, we propose a novel Expert Retrieval and Assembly (ERA) module that retrieves and assembles a set of experts most specialized at using discriminative subtle differences, to distinguish an input sample from other highly similar samples. To encourage our model to effectively use subtle differences for early action prediction, we push experts to discriminate exclusively between samples that are highly similar, forcing these experts to learn to use subtle differences that exist between those samples. Additionally, we design an effective Expert Learning Rate Optimization method that balances the experts' optimization and leads to better performance. We evaluate our ERA module on four public action datasets and achieve state-of-the-art performance.

Keywords: Early action prediction, dynamic networks, expert retrieval.

1 Introduction

The goal of early action prediction is to infer an action category at the early temporal stage, i.e., before the action is fully observed. This task is relevant to many practical applications, such as human-robot interaction [40, 17, 28], security surveillance [6, 23, 7] and self-driving vehicles [11, 36, 1] since a timely response is crucial in these scenarios. For example, for enhanced safety of self-driving vehicles, it is crucial that the actions of pedestrians can be predicted before they are fully completed, so that the vehicle can react promptly. Such utility of early action prediction has not gone unnoticed, and it has received a lot of research attention recently [30, 50, 15, 54, 55].

⋆ equal contribution
⋆⋆ corresponding author

Previous works [30, 55] show that one of the major challenges in early action prediction lies in the subtlety of the differences between some "hard" samples at the very beginning temporal stages, since only limited initial observations of the action sequences are seen and some important discriminative information in the middle or later parts of the sequences is not observed, greatly increasing the difficulty of making correct predictions. For instance, as shown in Fig. 1, though the human postures and motions in the full sequences of the actions "slapping" and "shaking hands" are quite different, their early parts are quite similar, with only subtle differences between them.

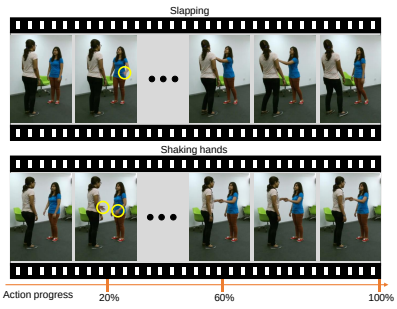

Fig. 1. Illustration of two actions at the early stage, taken from NTU RGB+D 120 [32]. Only subtle differences (highlighted in circles) exist for discrimination between the actions "Slapping" and "Shaking hands" at the early temporal stages (e.g., 20%). Best viewed in colour.

To tackle early action prediction, various types of deep networks have been proposed [30, 54, 55], but they still do not possess very good discrimination capabilities using subtle cues. In particular, deep networks prefer to learn to discriminate between the easier samples with major discriminative cues instead of the harder ones [20]. This can happen when we train the entire neural network by updating all its parameters using all samples – the gradients update all the parameters to contribute towards correctly classifying all these samples, which thus can lead to the network learning more *general patterns* that apply to more samples, as opposed to learning *specific subtle cues* to discriminate subtle differences that may only apply to a small subset of the data [12]. The performance drop from such sub-optimal training behaviour can be further exacerbated on the very challenging action prediction task, where there can be a lack of major discriminative cues at the early stages among different actions, and the importance of utilizing subtle differences is increased. Although recent work [30] on early action prediction has attempted to improve the discriminative ability on subtle cues through mining hard training samples, they train the parameters of the entire network using all samples, still leaving the network prone to sub-optimal performance with respect to subtle differences.

In this work, to improve the performance of deep networks on early action prediction, we propose an Expert Retrieval and Assembly (ERA) module that contains *non-experts* and *experts*. Unlike *non-experts* that contain parameters which are shared across all samples and capture general patterns that exist in many samples, *experts* are only trained on a subset of the data (according to their *keys*) and contain parameters that focus on encoding subtle differences to distinguish between highly similar samples. During the forward pass, a retrieval mechanism retrieves the most suitable experts, which are then assembled together with the non-experts to form a combination that is able to discriminate samples using an effective mix of general patterns and subtle differences. This re-

trieval mechanism is designed such that experts are retrieved by samples that are very similar, and thus, during training, the losses push the experts to learn *specialized discriminative subtle cues to distinguish exclusively between these similar samples*, encouraging the acquiring of expertise in exploiting relevant subtle cues. The proposed ERA module is *flexible*, and can be a plug-and-play replacement for traditional convolutional layers.

We design the ERA module with a set of experts to learn different subtle cues that exist across different actions. However, it is non-trivial to balance the training among different experts in the ERA module, especially when the experts might be selected by vastly different numbers of samples. For instance, as some subtle cues may be more common, a few experts are selected more often and might be better trained. Such unbalanced training may limit the overall performance of our ERA module. A possible solution could be for each expert to have its own individual learning rate, and we adjust these learning rates such that the experts that require more training will have correspondingly higher learning rates. However, considering the numerous experts in the ERA module, coupled with the envisioned scenario where ERA modules replace multiple convolutional layers in a network, the number of hyperparameters is too large for manual tuning to be practical. Thus, we design an Expert Learning Rate Optimization (ELRO) method that balances the training of experts within the ERA module, improving the overall performance.

In summary, our main contributions include: **(1)** We propose a novel ERA module that effectively utilizes subtle discriminative differences between similar actions through retrieval and assembly of the most suitable experts for action prediction. Our ERA module is a flexible plug-and-play module that can replace the traditional convolutional layer. **(2)** To balance the training among experts and further improve performance of the ERA module on early action prediction, we design an effective ELRO method. **(3)** We obtain state-of-the-art-performance on early action prediction on four widely used datasets by replacing convolutional layers of the baseline architectures with our ERA modules.

2 Related Work

Early Action Prediction refers to the task where only the front parts of each sequence are observed by the model. The loss of important discriminative information leads to a challenging scenario where subtle cues need to be properly utilized for successful discrimination. Different approaches [30, 50, 15, 54, 22, 24, 55, 10, 25, 27, 62, 33, 26, 21, 41, 39, 2, 52] have been proposed to address the early action prediction problem. Li *et al.* [30] focused on the 3D early activity prediction task by mining hard instances and training the model to discriminate between them. Ke *et al.* [22] proposed a Latent Global Network to learn how to transfer knowledge from full-length sequences to partially-observed sequences in an adversarial manner. Weng *et al.* [55] introduced a policy-based reinforcement learning mechanism to generate binary masks to preclude the negative category information leading to improved recognition accuracy. Wang *et al.* [54] proposed

a teacher-student network architecture to distill the global information contained within the full action video from the teacher network to the student network.

In this work, unlike the above-mentioned methods, we explore the usage of a *dynamic model* that pushes expert parameters to effectively encode subtle differences. We propose a novel ERA module, which learns to discriminate among similar samples using subtle cues by retrieving and assembling relevant experts for each sample.

Action Recognition is the task where a model predicts the classes of actions based on their full action sequences. Input data can come from different modalities, such as RGB data [31, 8, 60, 63, 64, 65] and skeletal data [44, 45, 4, 5, 48, 67, 34, 46, 38]. Here, we focus on early action prediction, which is important yet more challenging since the early segments of different actions can be highly similar [30, 54, 55].

Dynamic Networks refer to neural networks that adapt their parameters or structures according to the input. A variety of different methods have been explored, including dynamic depth [53, 51, 59], dynamic widths [37, 43], weight generation [3, 66], dynamic routing [58, 29] and spatially dynamic [61] methods. In general, dynamic networks can be employed for their improved computational efficiency and representation power.

As our ERA module retrieves a different set of experts for each input sample, it can be considered a type of dynamic module. To the best of our knowledge, our ERA module is the first work that dynamically assigns experts to handle *subsets of similar samples during training*, pushing them to *gain expertise in exploiting subtle differences*. This relies on our novel retrieval mechanism involving key-query matching that retrieves experts to handle similar samples, which is different from existing mechanisms [3, 66, 37, 43]. Moreover, we explore a novel ELRO method to further improve performance of our dynamic ERA module.

3 Method

Subtle differences among highly similar samples are difficult to be well-learned by deep neural networks that share all network parameters across all samples. When tackling the challenging early action prediction task, the importance of exploiting subtle cues is increased, as there can be a lack of major discriminative cues at the early stages of actions, which exacerbates the performance drop from the sub-optimal performance of deep networks using subtle cues.

Motivated by this, we design a novel ERA module with an expert-retrieval mechanism to better exploit subtle cues. The expert-retrieval mechanism retrieves experts (from the Expert Banks) with relevant expertise for each input sample, and assembles them with non-experts. By matching experts with input samples that are highly similar to each other during training, this mechanism allows experts to ignore distant samples, while pushing them to focus on distinguishing between highly similar samples by specializing in subtle differences.

Due to the uneven distribution of samples across different experts, there might be uneven training among experts, which limits performance of our ERA

module. To mitigate this issue, an effective ELRO method is implemented during the training of the experts, which tunes their individual learning rates, resulting in a more effective training of experts and improved performance.

Below, we first describe the early action prediction task. Then, we introduce the ERA module and explain in detail how the expert-retrieval mechanism can encourage experts to specialize during training. Lastly, we describe our ELRO method.

3.1 Problem Formulation

A full-length action sequence can be represented as a set $S = \{s_t\}_{t=1}^{T}$ containing T frames, where s_t denotes the frame at the t-th time step. Following previous works [15, 22, 30], S is divided into N independent segments, with each segment containing $\frac{T}{N}$ frames. A partial sequence consists of a set of frames $P = \{s_t\}_{t=1}^{\tau}$, with τ being the last frame in any one of the N segments, i.e., $\tau = i\frac{T}{N}, i = \{1, 2, ..., N\}$. The task of early action prediction is to predict the class $c \in \{1, 2, ..., C\}$ of the activity that the partial sequence P belongs to, and different observation ratios $\frac{\tau}{T}$ of P are tested.

3.2 ERA Module

As shown in Fig. 2, our ERA module consists of *candidate experts* contained within multiple Expert Banks and a *non-expert block*. Considering that convolutional architectures have been shown to be effective for the early action prediction task [30, 54], the experts are implemented as convolutional kernels. For ease of notation, we describe our method in a 2D convolutional kernel setting, even though it can be generalized to 1D, 3D or graph convolutions as well. This ability to generalize to other types of convolutions is important, as existing early action prediction architectures often use various types of convolutions, such as 3D convolutions [45] or graph + 2D convolutions [13].

Let an input be $X \in \mathbb{R}^{N_{in} \times N_h \times N_w}$, where N_{in}, N_h and N_w represent the channel, height and width dimensions of the input feature map. Note that here we omit the batch dimension for simplicity. Assume that, in the backbone model, input X is processed by a convolutional filter $W_{conv} \in \mathbb{R}^{N_{out} \times N_{in} \times b_h \times b_w}$, where N_{out} represents the number of output channels, and b_h and b_w represent the height and width of the convolutional kernel. We aim to replace W_{conv} with our ERA module, for better performance on early action prediction.

Specifically, we design our ERA module to also ultimately produce weights W_{ERA} of the same shape ($N_{out} \times N_{in} \times b_h \times b_w$) as W_{conv}, which can be seen as N_{out} kernels (each of shape $N_{in} \times b_h \times b_w$) that respectively produce each of the N_{out} output channels. More specifically, in our ERA Module, we split the N_{out} channels (and therefore also kernels) into two parts: d expert channels and $N_{out} - d$ non-expert channels, where d is a hyperparameter. To allow our d expert channels to specialize in subtle cues, we would like *each expert to be trained on only a subset of the data*, thus we introduce d Expert Banks containing M candidate experts each, and retrieve only one expert from each Expert Bank

per sample, such that the other $M-1$ candidate experts in the bank are unused for this sample. The $N_{out}-d$ non-expert kernels (collectively defined in a non-expert block $W_{nonexpert} \in \mathbb{R}^{(N_{out}-d)\times N_{in}\times b_h \times b_w}$) are shared over all samples, and thus tend to learn general patterns. We utilize a combination of both non-expert and expert kernels, because the usage of non-expert kernels to capture general patterns, is complementary with our experts that specialize at capturing subtle cues for discriminating between similar samples, and their combination leads to improvements on early action prediction.

1) Expert Banks To facilitate our intention to let each expert be trained on only a subset of samples, we define d Expert Banks, each containing M candidate experts, as shown in Fig. 2. The M candidate experts in the p-th Expert Bank are all potential candidates that can be retrieved for the corresponding p-th expert channel.

We define the i-th expert in the p-th Expert Bank as E_i^p, where E_i^p contains convolutional kernel weights m_i^p and a key k_i^p. The key k_i^p is used for matching with the most suitable samples, and represents the *area of expertise* of this expert, as it determines the samples that the expert will be retrieved for. Meanwhile, the expert kernel m_i^p acts as a *specialized mechanism* to process the discriminative subtle cues on the input features that match the key k_i^p. The expert key k_i^p and kernel m_i^p are model parameters that are trained in an end-to-end manner. For each expert E_i^p, $m_i^p \in \mathbb{R}^{N_{in}\times b_h \times b_w}$ and $k_i^p \in \mathbb{R}^K$, where K represents the dimensionality of the key, and $K << N_{in} \times N_h \times N_w$ for efficiency.

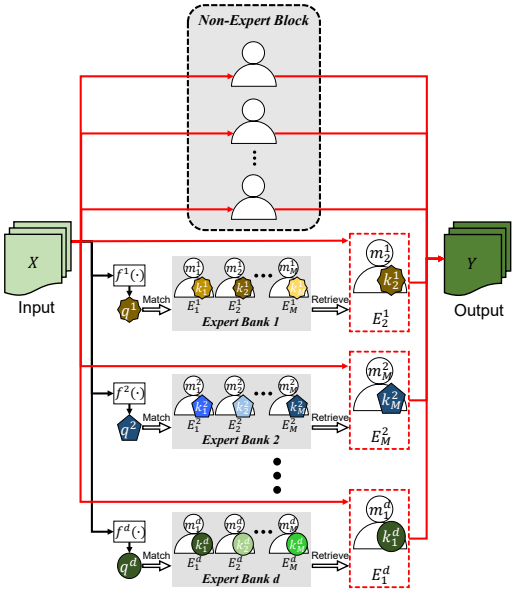

Fig. 2. Schema of our ERA module. Our ERA module contains N_{out} channels in total: $N_{out}-d$ non-expert channels (Top) and d expert channels (Bottom). Each expert channel retrieves its expert from a corresponding Expert Bank that contains M candidate experts, where each expert E_i^p consists of parameters m_i^p and a key k_i^p. The two important steps (i.e., retrieval and assembly) are indicated with red arrows. In the *retrieval* step, an expert will be retrieved from each Expert Bank through a key-query matching mechanism, such that d experts are retrieved across d expert channels. In the *assembly* step, the d retrieved experts are assembled and combined with the $N_{out}-d$ non-expert kernels to produce the output Y (with N_{out} channels).

2) Expert Retrieval We now show how we can retrieve the most suitable expert from each Expert Bank for input X, taking the p-th Expert Bank as an example. As shown in Fig. 2, we

first extract a compact and meaningful representation from the input feature map X. The query mapping function f^p maps the input feature map X to a lower-dimensional query $q^p \in \mathbb{R}^K$ as follows:

$$q^p = f^p(X). \tag{1}$$

This step transforms the feature map X into a query (vector) q^p of the dimensionality K.

Next, conditioned on q^p, we retrieve the most suitable expert from the p-th Expert Bank for discriminating subtle cues within the feature map X. Recall that each expert E_i^p holds a key k_i^p which represents its area of expertise, while q^p is the representation of the feature map X. The degree of suitability of the expert E_i^p on the feature map X can thus be obtained by calculating the matching score between k_i^p and q^p. We calculate the matching score s_i^p for each expert E_i^p using dot product between the query q^p and the key k_i^p as:

$$s_i^p = q^{p\top} k_i^p, \quad i = \{1, 2, ..., M\}. \tag{2}$$

$$I^p = Argmax_i(\{s_i^p\}_{i=1}^M), \tag{3}$$

where $Argmax_i$ returns the index i belonging to the largest element in the set $\{s_i^p\}_{i=1}^M$, and the returned index I^p represents the index of the retrieved expert. We take the highest matching score ($s_{I^p}^p$) within the set, as it will come from the key $k_{I^p}^p$ representing an area of expertise that matches the query q^p the best. Thus, the corresponding expert $E_{I^p}^p$ is the most suitable expert to be applied to the feature map X, and is retrieved from the p-th Expert Bank in this step.

It is worth mentioning that, using this key-query mechanism, the input feature maps that are highly similar (i.e., with similar q values) will tend to have high matching scores with the same key and retrieve the same expert. Crucially, this leads to the experts having to discriminate between highly similar input samples, pushing each expert to specialize in exploiting subtle cues for distinguishing between those similar samples to tackle early action prediction.

Above, we only show the operations on the p-th Expert Bank, but the same process is conducted for all d banks to retrieve d experts, which is shown in Fig. 2. Notably, this process (Eq. 1, 2, 3) across d Expert Banks can be done in *parallel*, so it is *efficient*.

3) Expert Assembly We assemble the retrieved expert kernels from all d Expert Banks (i.e., $\{m_{I^p}^p\}_{p=1}^d$) to form an expert block W_{expert} as follows:

$$W_{expert} = Concat(\{m_{I^p}^p\}_{p=1}^d), \tag{4}$$

where $W_{expert} \in \mathbb{R}^{d \times N_{in} \times b_h \times b_w}$ is composed of the parameters of the d experts that have the highest matching scores in the d banks, and $Concat$ denotes concatenation along the channel dimension. These retrieved experts will be specialized in capturing multiple subtle cues in X, that distinguish between the true class of X and other similar classes for tackling early action prediction.

Finally, to form the full convolutional block W_{ERA}, we further assemble the non-expert block $W_{nonexpert}$ and the expert block W_{expert} (shown in Fig. 2) as:

$$W_{ERA} = Concat(W_{nonexpert}, W_{expert}). \tag{5}$$

The final assembled block W_{ERA} will be applied to the feature map in a similar manner to a traditional convolutional kernel W_{conv}. Notably, as W_{ERA} can directly replace W_{conv}, our ERA module is a plug-and-play module that can replace the basic convolutional layer.

To apply the ERA module to other types of convolutions that are used in early action prediction architectures, only minor changes need to be made. For 1D and 3D convolutions, we change the shape of m_i^p and $W_{nonexpert}$ according to the corresponding 1D or 3D kernel. As for graph convolutions, since many graph convolutions (as used in [45]) are implemented based on traditional convolutions with additional parameters and steps to account for adjacency information, thus we can also implement our method by replacing the contained convolutional kernel with our ERA module in these scenarios.

4) Analysis of specialization of experts Next, we analyze how our ERA module allows experts to specialize in subtle cues through the expert-retrieval mechanism during training. This justification is rather important, as it explains why the retrieval of the most suitable d experts leads to better discrimination of subtle differences and tackles the sub-optimal training behaviour of the deep neural networks. We approach this by analyzing the differences between the gradients that update experts and non-experts *during backpropagation*.

Usually, the aggregated gradients $\bar{g}$ of a loss $\mathcal{L}$ w.r.t a model parameter w is computed by averaging over the entire batch with batch size B:

$$\bar{g} = \frac{1}{B} \sum_{j=1}^{B} \frac{\partial \mathcal{L}_j}{\partial w}. \tag{6}$$

The non-expert parameters are updated using $\bar{g}$ in Eq. 6, which trains the non-expert parameters to contribute towards classifying all samples, resulting in the learning of general patterns that apply to more samples, as opposed to subtle differences that occur only in a small subset of the data. If all parameters in a network are non-experts, this results in the network having sub-optimal performance with respect to subtle cues [20] and leads to worse performance on early action prediction.

In contrast, in our ERA module, not all experts are selected by each sample, as each expert is only retrieved for its most suitable samples. When backpropagating using the loss $\mathcal{L}$ on experts, the aggregated gradient $\bar{g}_i^p$ for the expert kernel weights m_i^p thus becomes:

$$\bar{g}_i^p = \frac{1}{|\mathcal{N}(k_i^p)|} \sum_{j \in \mathcal{N}(k_i^p)} \frac{\partial \mathcal{L}_j}{\partial m_i^p}, \tag{7}$$

where $\mathcal{N}(k_i^p)$ denotes the set of samples in the batch that select expert E_i^p (with key k_i^p and kernel m_i^p), i.e., $\mathcal{N}(k_i^p) = \{j \text{ s.t } I_j^p = i\}_{j=1}^{B}$, where I_j^p refers to the

index of the selected expert in the p-th Expert Bank (I^p) for the j-th sample in the batch. The samples in $\mathcal{N}(k_i^p)$ are likely to be very similar, with only some subtle differences, due to their close proximity to k_i^p in the feature space.

If we train the expert E_i^p using the gradient $\bar{g}_i^p$ as in Eq. 7, the expert is only updated using samples that are closer to this expert's area of expertise, i.e., samples which are in $\mathcal{N}(k_i^p)$. Thus, as compared to $\bar{g}$, much more emphasis is placed on learning to distinguish between these similar samples in $\mathcal{N}(k_i^p)$ only, which *pushes the expert to learn to exploit subtle differences in these samples*, as opposed to general patterns that generally hold across all data.

3.3 Expert Learning Rate Optimization

Experts in our ERA modules, together with all other network parameters, are end-to-end trainable using backpropagation. However, due to the uneven distribution of samples across experts, some experts might be selected by more samples and be better trained than others, possibly causing imbalanced training that limits the performance of our ERA module. To mitigate this effect, we design an Expert Learning Rate Optimization (ELRO) method that optimizes the training among experts, leading to improved early action prediction accuracy. For ease of notation, in this section, we only use one ERA module, although this method can also work for multiple ERA modules. We introduce a set of expert learning rates $\beta = \{\beta_i^p\}_{i\in\{1..M\},p\in\{1..d\}}$ as additional parameters, where each element β_i^p is a scalar that balances the training of a corresponding expert E_i^p during backpropagation. Instead of using manual tuning to adjust the large set of β, we update β using a *meta-learning approach* in an end-to-end manner.

The core idea of meta-learning [9, 47] is about "learning-to-learn", which in our case is *learning to optimize the learning rates* β for *improved training of experts*. This meta-optimization of β is conducted over two steps. Firstly, we simulate training on a training set while using the current β_i^p values to balance updates for each expert E_i^p respectively, to obtain a virtually updated *interim model*. Next, we evaluate the performance of this interim model on a validation set, and the gradients of these validation losses will *provide feedback on how we can adjust* β *to a more optimized* β' (which improves training of experts, and results in better performance on unseen validation samples). Finally, we then use the meta-optimized β' values for balancing expert updates during actual model training, which yields improvements in performance.

An illustration of our proposed ELRO method is shown in Fig 3. Specifically, in each iteration, we draw two batches of training data that are non-overlapping, which we call training samples $\mathcal{D}_{train}$ and validation samples $\mathcal{D}_{val}$. Then, the following three steps are employed to update the model parameters.

(1) Virtual Training. We simulate the training on $\mathcal{D}_{train}$ by virtually updating all the model parameters other than β as follows:

$$\hat{w} = w - \alpha\nabla_w\mathcal{L}(w, \mathcal{E}; \mathcal{D}_{train}), \tag{8}$$

$$\hat{E}_i^p = E_i^p - \beta_i^p\nabla_{E_i^p}\mathcal{L}(w, \mathcal{E}; \mathcal{D}_{train}), i = \{1, .., M\}, p = \{1, .., d\}, \tag{9}$$

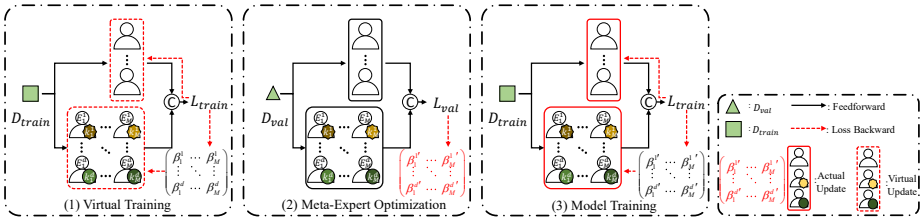

Fig. 3. Illustration of the Expert Learning Rate Optimization method. Two independent batches are sampled: training samples $\mathcal{D}_{train}$ and validation samples $\mathcal{D}_{val}$. Forward propagation paths are in black, while backpropagation paths are in red. The entire method consists of three phases: (1) Virtual Training, (2) Meta-Expert Optimization and (3) Model Training. At the Virtual Training step, all non-β model parameters are virtually updated using $\mathcal{D}_{train}$. At the Meta-Expert Optimization step, expert learning rate parameters β are dynamically updated using the gradients from validation loss $\mathcal{L}_{val}$. At the Model Training step, all non-β parameters are updated using $\mathcal{D}_{train}$ with updated β'. Best viewed in color.

where w represents the non-expert parameters, $\mathcal{E}$ represents the set of experts $\{E_i^p\}_{i\in\{1..M\},p\in\{1..d\}}$, α is the learning rate (which is a fixed hyperparameter), and $\mathcal{L}$ refers to the supervised loss for the early action prediction task. Note that here the update of each expert E_i^p (which includes key k_i^p and kernel weights m_i^p) is scaled by β_i^p.

(2) Meta-Expert Optimization. In this step, we evaluate the performance of the virtually updated model (consisting of $\hat{w}$ and $\hat{\mathcal{E}}$) on $\mathcal{D}_{val}$. The gradients w.r.t each expert learning rate β_i^p provide feedback on how β_i^p should be tuned for the virtually updated model to generalize better to unseen samples, as follows:

$$\beta_i^{p\prime} = \beta_i^p - \alpha\nabla_{\beta_i^p}\mathcal{L}(\hat{w},\hat{\mathcal{E}};\mathcal{D}_{val}), i=\{1,..,M\}, p=\{1,..,d\}. \tag{10}$$

Only β is updated in this step, and other parameters ($\hat{w}$ and $\hat{\mathcal{E}}$) remain fixed. Note that $\nabla_{\beta_i^p}$ takes gradients with respect to β_i^p as used in Eq. 9. This means that, by tuning β_i^p in Eq. 10, the newly updated expert learning rate $\beta_i^{p\prime}$ can provide better training for expert E_i^p if Eq. 9 is performed again.

(3) Model Training. After we obtain the set of meta-optimized expert learning rates $\beta' = \{\beta_i^{p\prime}\}_{i\in\{1..M\},p\in\{1..d\}}$, we can perform actual model training by updating model parameters $\mathcal{E}$ and w on $\mathcal{D}_{train}$ as:

$$w' = w - \alpha\nabla_w\mathcal{L}(w,\mathcal{E};\mathcal{D}_{train}), \tag{11}$$

$$E_i^{p\prime} = E_i^p - \beta_i^{p\prime}\nabla_{E_i^p}\mathcal{L}(w,\mathcal{E};\mathcal{D}_{train}), i=\{1,..,M\}, p=\{1,..,d\}, \tag{12}$$

In this step, the meta-optimized β' *balances the training of experts such that performance on unseen samples is improved.* This concludes one iteration of ELRO, where we have obtained updated parameters w', $\mathcal{E}'$ and β'. An outline of this algorithm is shown in the Supplementary Material.

3.4 Loss function

We train our model using a cross-entropy loss $\mathcal{L}_{CE}$ on the early action prediction task. Furthermore, we find that applying an additional similarity loss $\mathcal{L}_s$ brings some improvements in practice, where the similarity loss $\mathcal{L}_s$ penalizes experts that get too close to each other, which encourages experts to be more diverse. Specifically, we implement $\mathcal{L}_s$ on all ERA modules within our network, using a negative pairwise mean-squared loss among expert kernels in each Expert Bank, i.e., using one ERA module as an example, $\mathcal{L}_s = \sum_{p=1}^{d} \sum_{i=1}^{M} \sum_{j \neq i} ||m_i^p - m_j^p||^2$. Overall, our loss $\mathcal{L}$ is then given by $\mathcal{L} = \mathcal{L}_{CE} - \gamma_s \mathcal{L}_s$, where γ_s is a hyperparameter that weights the relative significance of losses.

4 Experiments

To validate effectiveness of our ERA module for early action prediction, we conduct extensive experiments on both skeletal and RGB datasets. We experiment on the NTU RGB+D 60 (NTU60) [42], NTU RGB+D 120 (NTU120) [42] and SYSU [14] datasets for skeletal data, and the UCF-101 (UCF101) dataset [49] for RGB data.

4.1 Implementation Details

Network Architecture. Following the previous works [30, 54], we use 2s-AGCN [45] and 3D ResNeXt-101 [13] as the backbone networks for skeletal and RGB datasets, respectively. As mentioned above, since the ERA module serves as a plug-and-play replacement for the conventional convolutional module, we uniformly replace 25% of convolutional layers with our ERA module in the backbone networks. Network hyperparameters $N_{in}, N_{out}, b_h, b_w$ at each layer follow the original settings in the backbone networks. Also, in each ERA module, $d = 0.2N_{out}$, i.e. 80% of the convolutional kernels are non-expert kernels and the other 20% are expert kernels, and $M = 5$.

For the mapping function f^p in Eq. 1, we first conduct average pooling across the spatial and temporal dimensions of the feature map before a linear layer is used to downsample to the dimensionality K, where K is set to 64.

Training. We perform experiments on Nvidia RTX 3090 GPU. For skeletal datasets, NTU60, NTU120 and SYSU, we follow [45] and set the initial learning rate α as 0.1, which then gradually decays to 0.001. The batch size B is 64. For RGB dataset UCF101, we follow the same experimental settings as [54]. Network parameters θ and expert learning rates β are updated using $\mathcal{L}$ defined in Sec. 3.4. γ_s is set to 0.1. Each β_i^p is initialized to α, and constrained to non-negative values. To allow end-to-end training of the retrieved experts in Eqn. 3, Gumbel-Softmax [19] gradients are computed during backpropagation for the Argmax operation, with temperature τ set to 1.

Table 1. Performance comparison (%) of Early Action Prediction on NTU60 and SYSU. We follow the evaluation setting of [30, 55, 39] and [54] respectively. Even without ELRO, we can attain state-of-the-art performance. With ELRO, our method obtains further improvements.

Methods	Observation Ratios on NTU60						Observation Ratios on SYSU					
	20%	40%	60%	80%	100%	AUC	20%	40%	60%	80%	100%	AUC
Jain *et al.* [18]	7.07	18.98	44.55	63.84	71.09	37.38	31.61	53.37	68.71	73.96	75.53	57.23
Ke *et al.* [21]	8.34	26.97	56.78	75.13	80.43	45.63	26.76	52.86	72.32	79.40	80.71	58.89
Kong *et al.* [26]	-	-	-	-	-	-	51.75	58.83	67.17	73.83	74.67	61.33
Ma *et al.* [35]	-	-	-	-	-	-	57.08	71.25	75.42	77.50	76.67	67.85
Weng *et al.* [55]	35.56	54.63	67.08	72.91	75.53	57.51	-	-	-	-	-	-
Aliakbarian *et al.*[41]	27.41	59.26	72.43	78.10	79.09	59.98	56.11	71.01	78.39	80.31	78.50	69.12
Hu *et al.* [16]	-	-	-	-	-	-	56.67	75.42	80.42	82.50	79.58	71.25
Wang *et al.* [54]	35.85	58.45	73.86	80.06	82.01	60.97	63.33	75.00	81.67	86.25	87.92	74.31
Pang *et al.* [39]	33.30	56.94	74.50	80.51	81.54	61.07	-	-	-	-	-	-
Tran *et al.* [50]	24.60	57.70	76.90	85.70	88.10	62.80	-	-	-	-	-	-
Ke *et al.* [22]	32.12	63.82	77.02	82.45	83.19	64.22	58.81	74.21	82.18	84.42	83.14	72.55
HARD-Net [30]	42.39	72.24	82.99	86.75	87.54	70.56	-	-	-	-	-	-
Baseline	38.09	66.36	78.67	83.29	84.10	66.43	60.71	73.04	77.81	83.88	84.32	72.20
ERA-Net w/o ELRO	43.94	73.23	84.53	87.61	87.97	71.62	63.50	80.82	82.70	86.33	87.10	75.78
ERA-Net	**53.98**	**74.34**	**85.03**	**88.35**	**88.45**	**73.87**	**65.30**	**81.27**	**85.67**	**89.17**	**89.38**	**77.73**

4.2 Experiments on Early Action Prediction

NTU60 dataset [42] has been widely used for 3D action recognition and early action prediction. It is a large dataset that contains more than 56 thousand skeletal sequences from 60 activity classes. All human skeletons in the dataset contain 3D coordinates of 25 body joints. As noted in [30], this dataset is challenging for the 3D early action prediction task due to the presence of many classes with very similar starting sequences. We follow the evaluation protocol of [30].

We first compare the proposed ERA-Net with the state-of-the-art approaches on NTU60. The results over different observation ratios are shown in Table 1. Our full method is employed in the **ERA-Net** setting. In **ERA-Net w/o ELRO**, we use ERA modules but do not implement β to train experts using the proposed ELRO algorithm, instead we train using a single backpropagation step that updates all model parameters at each iteration. We also provide the **Baseline** setting for comparison, where the backbone is used without ERA modules.

We report the prediction accuracy at each observation ratio. Furthermore, we use the Area Under Curve (AUC) metric in our experiments, following previous works [30, 55, 39]. The AUC measures the average precision over all observation ratios and broadly summarizes each model's performance into a single metric. On NTU60, we achieve more than a 3 point improvement against existing state-of-the art method [30], suggesting that the ERA module effectively increases the discriminative capabilities on the early action prediction task.

One crucial observation is that ERA-Net outperforms existing methods more significantly when the observation ratio is low. For example, when the observation ratio is 20%, ERA-Net improves over state-of-the-art [30] by more than 11%, which further demonstrates that the ERA module is especially effective in

Table 2. Performance comparison (%) of Early Action Prediction on NTU120 and UCF101. As no prior works report NTU120 early action prediction results, we compare our method to the baseline. For UCF101, we follow the evaluation setting of [54].

Methods	Observation Ratios on NTU120						Observation Ratios on UCF101					
	20%	40%	60%	80%	100%	AUC	10%	30%	50%	70%	90%	AUC
MSRNN [16]	-	-	-	-	-	-	68.01	88.71	89.25	89.92	90.23	80.89
Wu *et al.* [56]	-	-	-	-	-	-	80.24	84.55	86.28	87.53	88.24	80.57
Wu *et al.* [57]	-	-	-	-	-	-	82.36	88.97	91.32	92.41	93.02	84.66
Wang *et al.* [54]	-	-	-	-	-	-	83.32	88.92	90.85	91.28	91.31	89.64
Baseline	23.14	32.49	59.07	75.61	81.18	50.03	82.88	89.02	89.64	91.12	91.96	89.30
ERA-Net w/o ELRO	29.60	43.45	65.14	78.03	82.01	55.52	86.99	91.49	93.63	94.24	94.40	92.51
ERA-Net	**31.73**	**45.67**	**67.08**	**78.84**	**82.43**	**57.02**	**89.14**	**92.39**	**94.29**	**95.45**	**95.72**	**93.64**

picking up subtle cues to tackle hard samples (where samples are more similar at the earlier stages).

SYSU dataset [14] is also commonly used for 3D action recognition and early action prediction. The dataset contains 480 skeletal sequences belonging to 12 action classes performed by 40 subjects. The human skeletons in this dataset contain 3D coordinates of 20 joints. We follow evaluation protocol of [54]. Comparisons against state-of-the-art methods are displayed in Table 1, where ERA-Net outperforms the current state-of-the-art [54] by about 3 points.

NTU120 dataset [32] is an extension of NTU60. It is currently the largest RGB+D dataset for 3D action analysis with more than 114k skeletal sequences and contains 120 activity classes. This dataset is challenging for the early action prediction task, containing many classes that are hard to classify without observing the full sequences. Comparisons are displayed in Table 2, where ERA-Net outperforms the baseline by about 7 points on the AUC metric. We also observe very large improvements at lower observation ratios, demonstrating the efficacy of our method for early action prediction.

UCF101 dataset [49] is a popular dataset containing 13,320 video clips of 101 classes of human activities. It is a commonly used dataset for action prediction from RGB videos. Comparisons against state-of-the-art action prediction methods are shown in Table 2, where ERA-Net outperforms current state-of-the-art methods [57, 56, 54] by 4 or more AUC points, showing that ERA provides gains for early action prediction on RGB video datasets as well.

4.3 Ablation Study

Impact of number of experts. We evaluate the ratio of experts and non-experts in Table 3(a). As performance peaks at $20:80$, we set $d = 0.2N_{out}$, which allows for encoding of the most effective mix of general patterns and subtle cues within the layer.

Impact of size of Expert Banks (M). We evaluate the size of Expert Banks in Table 3(b). We find that the performance increases moderately when M is increased from 2 to 5, and remains stable when we further increase it. We argue

Table 3. Ablation studies conducted on NTU60. (a) Evaluation of ratios between number of experts and non-experts; (b) evaluation of size of Expert Banks M; (c) evaluation of the percentage (%) of convolutional layers replaced by ERA modules; (d) evaluation of the value of similarity loss weight γ_s; (e) evaluation of our dynamic retrieval mechanism against alternative static designs.

(a)		(b)		(c)		(d)		(e)	
Expert:Non-expert	AUC	M	AUC	% of ERA modules	AUC	γ_s	AUC	Method	AUC
0:100	66.43	1	66.43	0	66.43	0.05	72.92	Extra-Channel	67.55
20:80	73.87	2	71.55	25	73.87	0.1	73.87	Expert-Avg	68.02
60:40	71.22	5	73.87	50	73.79	0.2	73.85	ERA-Net	73.87
100:0	70.12	10	73.86	100	73.81	0.3	73.76		

that this is because the representation capacity by setting $M = 5$ is sufficient to capture the subtle cues present in the dataset.

Impact of number of ERA modules. We ablate the decision of replacing 25% of convolutional layers with ERA modules in Table 3(c). We find that, above 25%, the performance does not increase further. This suggests that, at 25%, there is already sufficient representation capacity to handle the encoding of subtle cues.

Impact of similarity loss weight (γ_s). We conduct ablation studies on the impact of γ_s in Table 3(d). $\gamma_s = 0.1$ performs the best. This because, when γ_s is set too low, the experts are not as diverse, and when it is set too high, the experts may lose focus on the main objective.

Impact of dynamic retrieval mechanism. We evaluate our dynamic design by comparing our ERA module against other alternative static designs in Table 3(e). **Expert-Avg** averages the outputs of all experts within the Expert Bank (i.e. all experts are used for each input sample, without dynamic expert selection), while **Extra-Channel** adds extra channels to the traditional convolutional layer. *Notably, these alternative static designs use the same number of parameters as our ERA-Net.* We find that our dynamic retrieval mechanism provides significant improvement over these alternatives.

5 Conclusion

In this paper, we have proposed a novel plug-and-play ERA module for early action prediction. To encourage the experts to effectively use subtle differences for early action prediction, we push them to discriminate exclusively among similar samples. An Expert Learning Rate Optimization algorithm is further proposed to balance the training among numerous experts, which improves performance. Our method obtains state-of-the-art performance on four popular datasets.

Acknowledgement This work is supported by National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-100E-2020-065), Ministry of Education Tier 1 Grant and SUTD Startup Research Grant.

References

1. Chaabane, M., Trabelsi, A., Blanchard, N., Beveridge, R.: Looking ahead: Anticipating pedestrians crossing with future frames prediction. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2297–2306 (2020) 1
2. Chen, L., Lu, J., Song, Z., Zhou, J.: Recurrent semantic preserving generation for action prediction. IEEE Transactions on Circuits and Systems for Video Technology **31**(1), 231–245 (2020) 3
3. Chen, Y., Dai, X., Liu, M., Chen, D., Yuan, L., Liu, Z.: Dynamic convolution: Attention over convolution kernels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11030–11039 (2020) 4
4. 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. pp. 13359–13368 (2021) 4
5. Cheng, K., Zhang, Y., He, X., Chen, W., Cheng, J., Lu, H.: Skeleton-based action recognition with shift graph convolutional network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 183–192 (2020) 4
6. Emad, M., Ishack, M., Ahmed, M., Osama, M., Salah, M., Khoriba, G.: Early-anomaly prediction in surveillance cameras for security applications. In: 2021 International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC). pp. 124–128. IEEE (2021) 1
7. Fatima, I., Fahim, M., Lee, Y.K., Lee, S.: A unified framework for activity recognition-based behavior analysis and action prediction in smart homes. Sensors **13**(2), 2682–2699 (2013) 1
8. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6202–6211 (2019) 4
9. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning. pp. 1126–1135. PMLR (2017) 9
10. Gammulle, H., Denman, S., Sridharan, S., Fookes, C.: Predicting the future: A jointly learnt model for action anticipation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5562–5571 (2019) 3
11. Gujjar, P., Vaughan, R.: Classifying pedestrian actions in advance using predicted video of urban driving scenes. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 2097–2103. IEEE (2019) 1
12. Han, Y., Huang, G., Song, S., Yang, L., Wang, H., Wang, Y.: Dynamic neural networks: A survey. arXiv preprint arXiv:2102.04906 (2021) 2
13. Hara, K., Kataoka, H., Satoh, Y.: Can spatiotemporal 3d cnns retrace the history of 2d cnns and imagenet? In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 6546–6555 (2018) 5, 11
14. Hu, J.F., Zheng, W.S., Lai, J., Zhang, J.: Jointly learning heterogeneous features for rgb-d activity recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5344–5352 (2015) 11, 13
15. Hu, J.F., Zheng, W.S., Ma, L., Wang, G., Lai, J.: Real-time rgb-d activity prediction by soft regression. In: European Conference on Computer Vision. pp. 280–296. Springer (2016) 1, 3, 5

16. Hu, J.F., Zheng, W.S., Ma, L., Wang, G., Lai, J., Zhang, J.: Early action prediction by soft regression. IEEE transactions on pattern analysis and machine intelligence **41**(11), 2568–2583 (2018) 12, 13
17. Huang, C.M., Mutlu, B.: Anticipatory robot control for efficient human-robot collaboration. In: 2016 11th ACM/IEEE international conference on human-robot interaction (HRI). pp. 83–90. IEEE (2016) 1
18. Jain, A., Singh, A., Koppula, H.S., Soh, S., Saxena, A.: Recurrent neural networks for driver activity anticipation via sensory-fusion architecture. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). pp. 3118–3125. IEEE (2016) 12
19. Jang, E., Gu, S., Poole, B.: Categorical reparameterization with gumbel-softmax. arXiv preprint arXiv:1611.01144 (2016) 11
20. Johnson, J.M., Khoshgoftaar, T.M.: Survey on deep learning with class imbalance. Journal of Big Data **6**(1), 1–54 (2019) 2, 8
21. Ke, Q., Bennamoun, M., An, S., Sohel, F., Boussaid, F.: A new representation of skeleton sequences for 3d action recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3288–3297 (2017) 3, 12
22. Ke, Q., Bennamoun, M., Rahmani, H., An, S., Sohel, F., Boussaid, F.: Learning latent global network for skeleton-based action prediction. IEEE Transactions on Image Processing **29**, 959–970 (2019) 3, 5, 12
23. Kong, Y., Fu, Y.: Human action recognition and prediction: A survey. arXiv preprint arXiv:1806.11230 (2018) 1
24. Kong, Y., Gao, S., Sun, B., Fu, Y.: Action prediction from videos via memorizing hard-to-predict samples. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 32 (2018) 3
25. Kong, Y., Kit, D., Fu, Y.: A discriminative model with multiple temporal scales for action prediction. In: European conference on computer vision. pp. 596–611. Springer (2014) 3
26. Kong, Y., Tao, Z., Fu, Y.: Deep sequential context networks for action prediction. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1473–1481 (2017) 3, 12
27. Kong, Y., Tao, Z., Fu, Y.: Adversarial action prediction networks. IEEE transactions on pattern analysis and machine intelligence **42**(3), 539–553 (2018) 3
28. Koppula, H.S., Saxena, A.: Anticipating human activities using object affordances for reactive robotic response. IEEE transactions on pattern analysis and machine intelligence **38**(1), 14–29 (2015) 1
29. Li, H., Wu, Z., Shrivastava, A., Davis, L.S.: 2d or not 2d? adaptive 3d convolution selection for efficient video recognition. In: CVPR. pp. 6155–6164 (2021) 4
30. Li, T., Liu, J., Zhang, W., Duan, L.: Hard-net: Hardness-aware discrimination network for 3d early activity prediction. In: European Conference on Computer Vision. pp. 420–436. Springer (2020) 1, 2, 3, 4, 5, 11, 12
31. Lin, J., Gan, C., Han, S.: Tsm: Temporal shift module for efficient video understanding. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 7083–7093 (2019) 4
32. 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 transactions on pattern analysis and machine intelligence **42**(10), 2684–2701 (2019) 2, 13
33. Liu, J., Shahroudy, A., Wang, G., Duan, L.Y., Kot, A.C.: Skeleton-based online action prediction using scale selection network. IEEE transactions on pattern analysis and machine intelligence **42**(6), 1453–1467 (2019) 3

34. Liu, Z., Zhang, H., Chen, Z., Wang, Z., Ouyang, W.: Disentangling and unifying graph convolutions for skeleton-based action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 143–152 (2020) 4
35. Ma, S., Sigal, L., Sclaroff, S.: Learning activity progression in lstms for activity detection and early detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1942–1950 (2016) 12
36. Mavrogiannis, A., Chandra, R., Manocha, D.: B-gap: Behavior-guided action prediction for autonomous navigation. arXiv preprint arXiv:2011.03748 (2020) 1
37. Mullapudi, R.T., Mark, W.R., Shazeer, N., Fatahalian, K.: Hydranets: Specialized dynamic architectures for efficient inference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8080–8089 (2018) 4
38. Nguyen, X.S.: Geomnet: A neural network based on riemannian geometries of spd matrix space and cholesky space for 3d skeleton-based interaction recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13379–13389 (2021) 4
39. Pang, G., Wang, X., Hu, J., Zhang, Q., Zheng, W.S.: Dbdnet: Learning bi-directional dynamics for early action prediction. In: IJCAI. pp. 897–903 (2019) 3, 12
40. Reily, B., Han, F., Parker, L.E., Zhang, H.: Skeleton-based bio-inspired human activity prediction for real-time human–robot interaction. Autonomous Robots **42**(6), 1281–1298 (2018) 1
41. Sadegh Aliakbarian, M., Sadat Saleh, F., Salzmann, M., Fernando, B., Petersson, L., Andersson, L.: Encouraging lstms to anticipate actions very early. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 280–289 (2017) 3, 12
42. Shahroudy, A., Liu, J., Ng, T.T., Wang, G.: Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1010–1019 (2016) 11, 12
43. Shazeer, N., Mirhoseini, A., Maziarz, K., Davis, A., Le, Q., Hinton, G., Dean, J.: Outrageously large neural networks: The sparsely-gated mixture-of-experts layer (2017) 4
44. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Skeleton-based action recognition with directed graph neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7912–7921 (2019) 4
45. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12026–12035 (2019) 4, 5, 8, 11
46. Shi, L., Zhang, Y., Cheng, J., Lu, H.: Adasgn: Adapting joint number and model size for efficient skeleton-based action recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13413–13422 (2021) 4
47. Shu, J., Xie, Q., Yi, L., Zhao, Q., Zhou, S., Xu, Z., Meng, D.: Meta-weight-net: Learning an explicit mapping for sample weighting (2019) 9
48. Song, Y.F., Zhang, Z., Shan, C., Wang, L.: Stronger, faster and more explainable: A graph convolutional baseline for skeleton-based action recognition. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 1625–1633 (2020) 4
49. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012) 11, 13

50. Tran, V., Balasubramanian, N., Hoai, M.: Progressive knowledge distillation for early action recognition. In: 2021 IEEE International Conference on Image Processing (ICIP). pp. 2583–2587. IEEE (2021) 1, 3, 12
51. Veit, A., Belongie, S.: Convolutional networks with adaptive inference graphs. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 3–18 (2018) 4
52. Wang, W., Chang, F., Liu, C., Li, G., Wang, B.: Ga-net: A guidance aware network for skeleton-based early activity recognition. IEEE Transactions on Multimedia (2021) 3
53. Wang, X., Yu, F., Dou, Z.Y., Darrell, T., Gonzalez, J.E.: Skipnet: Learning dynamic routing in convolutional networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 409–424 (2018) 4
54. Wang, X., Hu, J.F., Lai, J.H., Zhang, J., Zheng, W.S.: Progressive teacher-student learning for early action prediction. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3556–3565 (2019) 1, 2, 3, 4, 5, 11, 12, 13
55. Weng, J., Jiang, X., Zheng, W.L., Yuan, J.: Early action recognition with category exclusion using policy-based reinforcement learning. IEEE Transactions on Circuits and Systems for Video Technology **30**(12), 4626–4638 (2020) 1, 2, 3, 4, 12
56. Wu, X., Wang, R., Hou, J., Lin, H., Luo, J.: Spatial–temporal relation reasoning for action prediction in videos. International Journal of Computer Vision **129**(5), 1484–1505 (2021) 13
57. Wu, X., Zhao, J., Wang, R.: Anticipating future relations via graph growing for action prediction. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 35, pp. 2952–2960 (2021) 13
58. Wu, Z., Li, H., Zheng, Y., Xiong, C., Jiang, Y., Davis, L.S.: A coarse-to-fine framework for resource efficient video recognition. IJCV **129**(11), 2965–2977 (2021) 4
59. Wu, Z., Nagarajan, T., Kumar, A., Rennie, S., Davis, L.S., Grauman, K., Feris, R.: Blockdrop: Dynamic inference paths in residual networks. In: CVPR. pp. 8817–8826 (2018) 4
60. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: Proceedings of the European conference on computer vision (ECCV). pp. 305–321 (2018) 4
61. Xie, Z., Zhang, Z., Zhu, X., Huang, G., Lin, S.: Spatially adaptive inference with stochastic feature sampling and interpolation. In: European Conference on Computer Vision. pp. 531–548. Springer (2020) 4
62. Xu, W., Yu, J., Miao, Z., Wan, L., Ji, Q.: Prediction-cgan: Human action prediction with conditional generative adversarial networks. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 611–619 (2019) 3
63. Yan, R., Tang, J., Shu, X., Li, Z., Tian, Q.: Participation-contributed temporal dynamic model for group activity recognition. In: Proceedings of the 26th ACM international conference on Multimedia. pp. 1292–1300 (2018) 4
64. Yan, R., Xie, L., Tang, J., Shu, X., Tian, Q.: Higcin: Hierarchical graph-based cross inference network for group activity recognition. IEEE transactions on pattern analysis and machine intelligence (2020) 4
65. Yan, R., Xie, L., Tang, J., Shu, X., Tian, Q.: Social adaptive module for weakly-supervised group activity recognition. In: European Conference on Computer Vision. pp. 208–224. Springer (2020) 4
66. Yang, B., Bender, G., Le, Q.V., Ngiam, J.: Condconv: Conditionally parameterized convolutions for efficient inference (2019) 4

67. Ye, F., Pu, S., Zhong, Q., Li, C., Xie, D., Tang, H.: Dynamic gcn: Context-enriched topology learning for skeleton-based action recognition. In: Proceedings of the 28th ACM International Conference on Multimedia. pp. 55–63 (2020) 4