Multi-View Action Recognition using Contrastive Learning

Ketul Shah1  Anshul Shah1  Chun Pong Lau1  Celso M. de Melo2  Rama Chellappa1
1Johns Hopkins University  2DEVCOM Army Research Laboratory

{ksahah33, ashah95, clau13, rchella4}@jhu.edu celso.miguel.de.melo@gmail.com

Abstract

In this work, we present a method for RGB-based action recognition using multi-view videos. We present a supervised contrastive learning framework to learn a feature embedding robust to changes in viewpoint, by effectively leveraging multi-view data. We use an improved supervised contrastive loss and augment the positives with those coming from synchronized viewpoints. We also propose a new approach to use classifier probabilities to guide the selection of hard negatives in the contrastive loss, to learn a more discriminative representation. Negative samples from confusing classes based on posterior are weighted higher. We also show that our method leads to better domain generalization compared to the standard supervised training based on synthetic multi-view data. Extensive experiments on real (NTU-60, NTU-120, NUMA) and synthetic (RoCoG) data demonstrate the effectiveness of our approach.

1. Introduction

Action recognition from videos is an active area of research [55, 70] in computer vision. A lot of work has focused on making better use of spatio-temporal information [54, 19, 4], developing more efficient architectures [18, 36, 62], etc. The hope is to learn models which are robust to novel viewpoints at test time. Much of this recent progress in action recognition is based on datasets where viewpoint information is not explicitly available. Our paper focuses on multi-view action recognition in scenarios where synchronized multi-view videos are available during training. This scenario arises in many practical applications in security, road safety, robotics, sports, etc.

Multi-view recognition has been extensively studied, the availability of large scale datasets [50, 37] driving recent progress of deep learning-based methods. Approaches based on RGB [57, 11], depth [26], infrared [13], skeleton [39, 8] modalities have been proposed. Recent advancements in multi-view action recognition are heavily focused on skeleton-based action recognition. 2D [71] and 3D [20, 8] skeleton-based methods have achieved state-of-the-art performance on benchmark datasets.

Methods relying on ground truth 2D/3D human pose information often assume access to datasets annotated with pose information, which is not necessarily available in data collected for real life applications. Annotating videos for ground truth pose is very expensive and dataset collectors often rely on specialized hardware (like Kinect) which exploit depth sensors to obtain accurate skeletons. An alternative could be to use estimated pose, but human pose estimation from RGB videos can be challenging, especially in scenarios where activities include human-object interactions, complex scenes with multiple people and heavy occlusion. Moreover, high quality pose estimation methods are slow, as most of these methods rely on object detection as an intermediate step. In most practical settings, multi-view setups have only RGB cameras.

In this paper, we focus on RGB-based multi-view action recognition. Fig. 1 presents an overview of our approach.
We use contrastive loss to achieve viewpoint robustness, leveraging videos captured simultaneously from multiple viewpoints. Specifically, we propose an improved version of supervised contrastive loss, and incorporate viewpoint synchronized videos as additional positives.

The quality and number of negatives is known to be important for contrastive learning [24, 49]. It has been shown [49, 47] that use of hard negatives for contrastive loss improves the quality of learned representations. Motivated by these observations in the self-supervised learning setting, we propose a novel approach for using hard negatives with contrastive loss in supervised settings. Our approach enables the classifier to guide the selection of negatives. Intuitively, the classifier gives a high score for confusing classes. This information lets us upweight negatives belonging to those classes for contrastive learning, effectively learning better features. We term the resulting hardness-aware contrastive loss ViewCon.

Extensive experiments on the NTU-RGB+D [50, 37] and NUMA [60] datasets demonstrate that our method can learn discriminative viewpoint-invariant representations, which can be used for transferring to small datasets showing generalization capability. We achieve state-of-the-art results compared to previous RGB-based approaches on these datasets. We also analyze the various design choices for our loss function. We note that it can be difficult to acquire synchronized multi-view videos for arbitrary real world scenarios/applications and there are associated privacy concerns. Synthetic data provides a natural solution to these problems, where it is easy to generate large-scale synchronized multi-view data without any privacy issues. We show that our approach also gives consistent improvements on a synthetic multi-view dataset [14] and that the resulting features generalize better to the real test data.

Our main contributions are summarized below:

- We propose a method for RGB-based action recognition, using contrastive learning to leverage multi-view training videos for achieving robustness to viewpoints.
- We propose a novel way to sample hard negatives for contrastive loss in the supervised setting by re-weighting the negatives using class probabilities. To the best of our knowledge, the use of hard negatives with supervised contrastive learning has not been explored before.
- Our method achieves state-of-the-art results on the popular NTU-60, NTU-120 and NUMA datasets. We also show the effectiveness of our method when using synthetic training data, and in transfer learning setting, validating our method in diverse scenarios.

2. Related Work

Action Recognition. There has been significant progress in action recognition, from traditional methods [55], to the latest advancements using deep learning [70]. Two-stream networks [54] was one of the first works to show effectiveness of using two-stream CNN, for integrating appearance and flow features. 13D [4] introduced the idea of inflating 2D CNN weights learnt from image datasets for effectively training 3D CNNs. S3D [64] replaces spatio-temporal 3D CNN layers by factored spatial and temporal 3D convolutions, and also incorporates feature gating in their architecture. SlowFast networks [19] introduces two pathways, a low frame-rate slow pathway for encoding spatial semantics and a high frame-rate fast pathway to capture motion dynamics. [18, 36, 62] aim to increase efficiency, and transformer-based architectures have recently been proposed [65] for action recognition.

While these methods show impressive performance, they are prone to learn shortcuts and capture biases such as scene, objects, context, viewpoints, etc [61, 35] and hence may not generalize well. [32, 35] provides ways of quantifying these biases for video models. [9] mitigates scene bias by using an adversarial loss for scenes based on gradient reversal, along with a human mask confusion loss. [66, 59] suppresses scene information by performing video transformations in the self-supervised learning setting. All the methods discussed above mainly deal with single-view videos, and do not consider multi-view action recognition.


Skeleton-based methods [8, 20, 39, 31, 52, 53, 34, 3, 38, 67] have received a lot of attention due to the availability of accurate 3D skeleton ground truth in the benchmark datasets (all in indoor settings) as they are collected using Kinect. [20] proposes a geometry-aware deep neural network for processing skeleton data, using rigid and non-rigid transformations. [39] uses a disentangled multi-scale aggregation scheme, processed with a unified spatial-temporal graph convolution network. [53] proposes an efficient skeleton-based method by adaptively controlling the number of input joints and the model size based on input. Approaches combining skeleton and appearance modalities [12, 10, 1, 40] have also been studied in literature.

Most of the current state-of-the-art methods use ground
truth 3D skeleton information for training and testing. It is difficult to estimate accurate 3D skeleton for videos in-the-wild without access to depth information. Hence, we focus on RGB-based multi-view action recognition.

**Contrastive Learning.** Contrastive learning has recently become a popular paradigm for learning self-supervised representations, and has enjoyed wide success. [6, 24] have shown superior performance compared to supervised learning methods for image classification. The main idea of contrastive learning is based on the classical noise-contrastive estimation (NCE) [22], with recent methods adapting it to effectively solve the instance classification task. The goal is to learn a discriminative feature space by classifying positives (which are typically defined as data-augmented versions of input) from negatives. In the image domain, [6] studies the effect of data-augmentations for defining positives and introduces the use of a non-linear projection head for learning better representations. [24] enables the use of more negatives by maintaining a queue. The idea has been extended to video domain [44, 17], where [44] uses temporally distant clips from a given video with spatial augmentations as positives. Negatives are chosen from other videos. [17] additionally incorporates temporally shuffled negatives from the same video.

There has been work to adapt contrastive loss for supervised learning [29, 48, 5]. [29] extends the contrastive loss to leverage label information, and suggests ways for incorporating multiple positives in the contrastive loss. Supervised contrastive loss has been applied for tasks such as domain adaptation [48], continual learning [5], etc.

**Hard Negatives for Contrastive Learning.** The quality of representations learned using contrastive loss is dependent on the amount and quality of negative samples used. Recently, [28, 49, 47] have proposed ways to choose and include hard negatives in the contrastive loss for self-supervised setting. [28] uses mixup to generate hard negatives by combining highest similarity samples in the queue with each other, and with the anchor. [49] selects hard negatives as a sparse set of support vectors and contrastiveness is enforced by maximizing the margin between positives and negatives. [47] proposes a method to select hard negatives based on the similarity of negatives to the anchor.

We propose to leverage hard negatives for the supervised contrastive learning setting, which to the best of our knowledge has not been explored.

### 3. Method

In this section, we first discuss the problem setup before presenting our approach dubbed ViewCon.

**Problem Formulation.** Let us denote the dataset $\mathcal{D}$ as $\{(x_1^i, x_2^i, ..., x_{V}^i), y_i\}_{i=1}^{N}$, where each activity instance $i$ consists of synchronised videos $(x_1^i, x_2^i, ..., x_{V}^i)$ captured from $V$ viewpoints, with class label $y_i \in \{1, ..., C\}$. $C$ denotes the total number of classes. The dataset consists of a total of $N \times V$ videos, with $N$ activity instances, each captured from $V$ viewpoints.

The goal is to learn a function $f_\theta$ which maps a video clip to its representation, such that the representation is robust to changes in viewpoint, while also being discriminative of action classes. We use contrastive learning to guide feature embeddings of different viewpoints of the same activity instance nearby in the feature space, close to same class features. Contrastive learning methods usually devise different ways of defining positives (data augmentations, sampling at different frame rate, etc), such that the semantic content in the data is preserved. [6, 24] use scaling, color jittering, blurring, etc as augmentations, and [44, 23] use optical flow, varying frame rate clips, etc as positives to learn features robust to these changes. In our work, we seek view invariance in addition to robustness from various data augmentations. To do so, we use features of different viewpoints of the same activity instance as positives, to pull them closer. We employ an improved version of the supervised contrastive loss to realize this. We also propose a novel method to make effective use of hard negative samples in the contrastive loss, by leveraging classifier probabilities. Finally, we discuss different practical considerations involved in using these successfully. The overall pipeline is shown in Fig. 2. Next, we describe each part of our method in detail.

**View Contrastive Learning** Contrastive learning-based methods have shown great success in self-supervised representation learning [6, 24, 25, 7, 23], as well as supervised representation learning [29]. Our method is based on the MoCo v2 [7] framework, which we briefly describe next.

Given an input video clip, an anchor (query) and a positive (key) sample is generated. The anchor sample is passed through an encoder $f_\theta$ to obtain anchor features, and a momentum updated version of the encoder is used to obtain...

![Figure 2: Overall pipeline. We first extract features for the anchor $x_a$ and positives $\{x_p\}$. The classifier scores of the anchor are used to generate weights for negatives $N$ using the function $W(\cdot)$. The weights along with the projected features (obtained using $g_\phi$) for the anchor, positives and negatives are used to compute the ViewCon loss.](image-url)
positive features. A projection head $g$ is used to project these features to a lower dimensional space, where the contrastive loss is computed. A queue stores positive features from previous batches which are used as negatives in the contrastive loss. For this task of instance discrimination, the InfoNCE loss [6] is used for training encoder and projection head.

[29] studies ways to extend the InfoNCE loss to supervised setting where we have multiple positives, and no false negatives. The supervised contrastive loss proposed by [29] is given in Eq. 1.

$$L_{\text{SupCon}} = \frac{1}{|P_i|} \sum_{p \in P_i} L(i, p)$$  \hspace{1cm} (1)

$$L(i, p) = -\log \frac{\exp (z_i \cdot z_p / \tau)}{\sum_{p \in P_i} \exp (z_i \cdot z_p / \tau) + \sum_{n \in N_i} \exp (z_i \cdot z_n / \tau)}$$  \hspace{1cm} (2)

Here, for an anchor video clip $x_i$ belonging to class $y_i$, the positive set $P_i$ is made of data-augmented clips from other videos with the same label, in addition to augmented anchor clip. $N_i$ is the set of negatives, which contains clips from other classes. $z_i \cdot z_j$ denotes the cosine similarity between normalized features of clips $x_i$ and $x_j$. $\tau$ is the temperature parameter.

We improve the loss in Eq. 2 as discussed below and use the updated loss in our method. Note that Eq. 2 can be viewed as performing $|P_i| + |N_i|$ way classification, for classifying the positive sample in the numerator, from all the samples included in the denominator. We argue this is not ideal, since it tries to discriminate one positive from other positives, and hence we propose to remove other positives from the denominator (as in Eq. 4). This effectively leads to discrimination of the current positive only from other negatives.

Moreover, our problem formulation allows us to use synchronized viewpoints as positives, which helps us achieve robustness to viewpoints. More specifically, given an anchor video clip $x_i$ from class $y_i$, the positive set $P_i$ is constructed with three types of positives: 1) augmented clip from the same video, 2) clips from videos of other viewpoints of this instance, and 3) clips from other instances of the same class. The negative set $N_i$ consists of clips from videos belonging to other classes. Intuitively, construction of our positive and negative set enforces features from different viewpoints of the same activity instance to be pulled together while being pushed away from features of other activity classes. Next, we describe our approach for sampling hard negatives.

**Hard Negative Sample Re-weighting.** Hard negative sampling for self-supervised contrastive learning [49, 47, 28] has been proven effective in learning better representations.

In our supervised contrastive setting, we propose to leverage the classifier probabilities for selecting and re-weighting hard negative samples in the contrastive loss.

Specifically, given an anchor clip $x_i$, its features $(h = f_\theta(x))$ are passed through the classifier $c_\zeta$ to obtain a probability distribution $p(y|x)$ over all classes, where $p(y|x) \in \mathbb{R}^C$. The probability of classifying input clip $x_i$ to class $y_j$ is given by $p(y_j|x)$. This is used to determine the hardness of class $y_j$ for anchor $x$. To see this, note that classes more similar to anchor class are harder to discriminate (and receive higher probability) than classes which are very distinct from the anchor class (which receive low probability). For example, *typing on keyboard* is much more similar to *writing on paper* than *clapping*. As shown in Fig. 3, a negative sample $x_n$ belonging to class $y_n$ in the contrastive loss is re-weighted using $w_n$, which is proportional to $p(y_n|x)$ (the probability of anchor clip being classified to class $y_n$).

The updated hardness-aware view contrastive loss (dubbed ViewCon) is given in Eq. 3 below:

$$L_{\text{ViewCon}} = \frac{1}{|P_i|} \sum_{p \in P_i} L_1(i, p)$$  \hspace{1cm} (3)

$$L_1(i, p) = -\log \frac{\exp (z_i \cdot z_p / \tau)}{\sum_{p \in P_i} \exp (z_i \cdot z_p / \tau) + \sum_{n \in N_i} w_n \exp (z_i \cdot z_n / \tau)}$$  \hspace{1cm} (4)

Here, $w_n \propto p(y_n|x_i)$. Negative classes similar to anchor $x_i$ class will have higher $p(y_n|x_i)$ and will be weighted higher.
We now explain how we obtain the final weights \( \{w_n\}_{n \in N} \) for an anchor \( x_i \), given the classifier probabilities \( p(y|x_i) \) and the labels of negatives \( N \). First, we assign the weight \( w_n \) for each negative sample \( x_n \) from class \( y_n \) to \( p(y_n|x_i) \). Note that these weights lie in \([0,1]\), and hence only allow decreasing the effective similarity (pushing farther in feature space). To see this, recollect that term \( s_{i,n} = z_i \cdot z_n \) calculates the similarity between L2-normalized features of the anchor \( z_i \) and those of a negative sample \( z_n \). Our approach weights this exponential similarity by \( w_n \) as in Eq. 4. This reweighting can be seen as effectively modifying the original similarity with \( s_{i,n}^{new} = \tau \log w_n + s_{i,n} \). It can be seen that weights less than one reduces effective similarity. Next, we normalize the weights by the average weight, i.e. \( w_n \leftarrow w_n / \text{mean}(w_n) \), now allowing the weights to take values greater than one. We clamp the minimum value of weights \( w_n \) to 1, thus only allowing increasing effective similarity.

We also note that the weights resulting from classifier probabilities can be highly skewed if the classifier gives overconfident predictions. Our approach relies on the fact that there are multiple peaks in the probability distribution which helps reweight the negatives. Note that overconfident predictions would likely lead to a single dominant peak which will lead to very small weights. To correct this, we use label smoothing for regularization and it helps in getting better calibrated classifier predictions [41], resulting in more useful weights.

**Algorithm 1** ViewCon loss computation

1. Input: anchor \( x_a \), positives \( \{x_p\} \), negative features \( \mathcal{N} \), feature extractor \( f_\theta \), projection MLP \( g_\phi \), classifier \( c_\zeta \), temperature \( \tau \)
2. Output: loss, 
3. # extract features \( h \) and classifier output 
4. \( h_a, \{h_p\} = \text{extract_features}(x_a, \{x_p\}, f_\theta) \)
5. \( c_a = \text{classify}(h_a, c_\zeta) \)
6. # project and L2 normalize 
7. \( z_a, \{z_p\} = \text{projector}(h_a, \{h_p\}, g_\phi) \)
8. 
9. # query GT class for all negatives 
10. \( \{y_n\} := \text{get_class}(\mathcal{N}) \)
11. # index prob for GT class of each negative 
12. \( \mathcal{W} := \{\text{index}(c_a, y_n)\}_{n=1}^{\mathcal{N}} \)
13. # normalize and clamp minimum to 1 
14. \( \{w_n\} = \text{clamp} (\mathcal{W} \sum_n \frac{1}{w_n}, 1, \infty) \)
15. 
16. # calculate ViewCon loss (Eq. 3) 
17. \( \text{loss}_{\text{viewCon}} = \mathcal{L}_{\text{viewCon}}(z_a, \{z_p\}, \{z_n\}, \{w_n\}, \tau) \)

**Action Classifier.** In the supervised contrastive learning setting, it is a common practice [29] to pre-train using the contrastive loss and train the classifier on top in a separate stage while also fine-tuning the backbone. We instead train the classifier simultaneously, and propose to use the output probabilities of the classifier to guide sampling of hard negatives for contrastive loss as described above. The classifier is trained using the cross-entropy loss \( L_{CE} \). This loss is only used to train the classifier and the gradients are not backpropagated to the encoder. We use label smoothing for training the classifier in all our experiments as it reduces overconfident predictions and helps provide more calibrated classifier probabilities, as shown in [41].

### 4. Experiments

In this section, we present experimental results to show the effectiveness of the proposed approach.

#### 4.1. Datasets

**NTU-RGB+D 60.** NTU-60 [50] is a large-scale multi-view action recognition dataset containing 56880 videos from 60 action classes, captured from 40 subjects, using Kinect v2. Each activity instance is captured at the same time from three different viewpoints. We evaluate our method on the two standard benchmarks as provided in [50]: (1) Cross-Subject (xs)ub) and (2) Cross-View (xv)iew. For the cross-subject benchmark, the 40 subjects are split into two sets, for training and testing, with 20 subjects each. For the cross-view setting, videos from cameras 2 and 3 are used for training, and videos from camera 1 are used for testing.

**NTU-RGB+D 120.** NTU-120 [37] is the extended version of NTU-60 dataset, consisting of 114480 videos from 120 action categories. We evaluate on the two standard protocols as in [37]: (1) Cross-Subject (xs)ub) and (2) Cross-Setup (xset). The cross-subject setting splits the subjects into training and testing subjects, whereas the cross-setup setting divides the data into training and testing based on the setup ID.

**Northwestern-UCLA Multiview Action.** NUMA [60] is a smaller dataset consisting of 1493 videos from ten action classes. Each action is performed by ten actors and is captured from three viewpoints. The dataset provides RGB, depth, and skeleton modalities. We use this dataset for transfer learning setting and only use RGB frames for our experiments.

**Robot Control Gestures.** RoCoG [14] is a gesture recognition dataset for studying the usefulness of synthetic data. It consists of synthetic and real videos from seven gestures captured from multiple viewpoints. The real data includes videos from fourteen subjects, while synthetic data is rendered with varying parameters such as character, environment, camera angle, etc. We use the training and testing splits provided in [14].
4.2. Implementation details

We choose S3D [64] as our encoder \( f_\theta \) for all our experiments. A 2-layer MLP with ReLU non-linearity is used as the projection head \( g_\phi \), a common practice as in [6, 7]. The action classifier is a single linear layer with batch norm, whose inputs are L2-normalized. The encoder takes clips of 32 RGB frames as inputs with a skip rate of 2, which are sampled starting from a random time from the input video. We apply the following clip-consistent data augmentations: random crop, horizontal flip, gaussian blur, and color jitter. We use a queue size of 2048 to cache negative features, and use momentum of 0.999 for the momentum updated encoder and projection head. Label smoothing of 0.6 is used for the cross-entropy loss in all experiments. We use the Adam optimizer for all modules. For the encoder and projection head, we use a learning rate of \( 10^{-4} \) and weight decay of \( 10^{-5} \). For classifier, we use a learning rate of \( 10^{-3} \) and a weight decay of \( 10^{-3} \). We use a batch size of 32 and train our method for 100 epochs. For all the modules, the learning rate is halved after every thirty epochs. We implement our method using the PyTorch framework, and use 4 GPUs for training each experiment. At test time, we use ten crops from temporally overlapping clips spanning the duration of the video and average their class probabilities to produce the final prediction.

4.3. Comparisons to state-of-the-art

We evaluate our method on the cross-subject and cross-view benchmarks on the widely used NTU-60 [50] and NTU-120 [37] datasets. We compare our method to state-of-the-art approaches using RGB information for multi-view action recognition on these benchmarks. We also compare with other methods that use additional input modalities, such as skeleton pose and depth, along with RGB. For unsupervised methods, we compare with the reported end-to-end fine-tuned results using class labels. In our experiments, at train time, we use the average probabilities of anchor and its corresponding synchronized views to get weights for negatives. We note that the predictions for multiple viewpoints of a given instance are not combined (which can lead to further improvements) to be consistent with prior works.

**Results on NTU-60 [50] and NTU-120 [37].** In Tables 1 and 2, we show that our method consistently outperforms previous RGB-based methods on the cross-view and cross-subject benchmarks of both NTU-60 and NTU-120 datasets. On NTU-60, we show an improvement of 3.9% on the cross-view setting, demonstrating that features learned using our approach are more robust to viewpoint shifts. On the cross-subject benchmark, we improve upon the state-of-the-art by 1.7% showing the efficacy of our approach. On the larger NTU-120 dataset with more fine-grained activities, we observe improvements of 1.3% on the cross-setup benchmark and 1.1% on the cross-subject benchmark. We also report our 1-crop results in supplementary (Sec. 2) as some of the baselines ([57, 2, 21]) uses a single center crop at test time. Moreover, we also compare with methods using additional modality along with RGB, such as pose and depth. Significantly, in three of the four benchmarks, our RGB only method outperforms methods based on additional modalities. Similar to ViewCLR, we use AGCN [51] (joint stream only) to process the pose modality and perform late fusion of logits from both modalities. AGCN joint stream results in 94% on NTU-60 xview and 85.9% on NTU-60 xsub benchmarks. In Table 1, we see that our combined model leads to consistent improvements over single modality only. This shows that our RGB-based method extracts complementary information to pose-based AGCN.

![Table 1: Comparison with state-of-the-art on cross-view (xview) and cross-subject (xsub) benchmarks of NTU-60 dataset. The proposed approach outperforms SotA approaches trained on RGB on both benchmarks. Fusion with pose modality leads to consistent improvements.](image)

<table>
<thead>
<tr>
<th>Method</th>
<th>Modality</th>
<th>NTU-60 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>xview</td>
<td>xsub</td>
</tr>
<tr>
<td>STA-Hands [1]</td>
<td>RGB</td>
<td>70.6</td>
</tr>
<tr>
<td>Separable STA [10]</td>
<td>RGB</td>
<td>75.3</td>
</tr>
<tr>
<td>VPN [12]</td>
<td>RGB</td>
<td>86.3</td>
</tr>
<tr>
<td>ViewCLR[11]+Pose</td>
<td>RGB+Pose</td>
<td>93.7</td>
</tr>
<tr>
<td>Zhang et al. [68]</td>
<td>RGB</td>
<td>94.1</td>
</tr>
<tr>
<td>DA-Net [58]</td>
<td>RGB</td>
<td>94.6</td>
</tr>
<tr>
<td>Vyas et al. [57]</td>
<td>RGB</td>
<td>94.3</td>
</tr>
<tr>
<td>Debnath et al. [16]</td>
<td>RGB</td>
<td>93.2</td>
</tr>
<tr>
<td>Glimpse Clouds [2]</td>
<td>RGB</td>
<td>93.7</td>
</tr>
<tr>
<td>Piergiovanni et al. [43]</td>
<td>RGB</td>
<td>94.1</td>
</tr>
<tr>
<td>ViewCLR [11]</td>
<td>RGB</td>
<td>98.0</td>
</tr>
<tr>
<td>Ours+Pose</td>
<td>RGB+Pose</td>
<td>98.9</td>
</tr>
</tbody>
</table>

4.4. Transfer Learning

To show the generalization capability of features learned using our method, we show results on the smaller NUMA [60] dataset containing ten action classes. We report results on the cross-view benchmark. In this protocol, videos from cameras 1 and 2 are used for training and videos from camera 3 are used for testing. We initialize with weights from NTU-60 pre-trained models and fine-tune on NUMA dataset for 300 epochs. Table 3 reports accuracy on the cross-view setting for NUMA dataset. Our method improves by 2.6% over previous approaches, showing generalizability of our approach.
Table 2: Comparison with state-of-the-art on cross-setup (xset) and cross-subject (xsub) benchmarks of NTU-120 dataset. † uses RGB, flow and depth while training.

<table>
<thead>
<tr>
<th>Method</th>
<th>Modality</th>
<th>NTU-120 (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>xset</td>
</tr>
<tr>
<td>Hu et al. [27]</td>
<td>RGB + Depth</td>
<td>44.9</td>
</tr>
<tr>
<td>Hu et al. [26]</td>
<td>RGB + Depth</td>
<td>54.7</td>
</tr>
<tr>
<td>DMCL [21] †</td>
<td>RGB</td>
<td>84.3</td>
</tr>
<tr>
<td>Liu et al. [37]</td>
<td>RGB</td>
<td>54.8</td>
</tr>
<tr>
<td>ViewCLR [11]</td>
<td>RGB</td>
<td>86.2</td>
</tr>
<tr>
<td>Ours</td>
<td>RGB</td>
<td>87.5</td>
</tr>
</tbody>
</table>

Table 3: Accuracy on the cross-view benchmark of NUMA dataset. We significantly outperform other RGB-based methods on this dataset in the transfer learning setting.

<table>
<thead>
<tr>
<th>Method</th>
<th>Modality</th>
<th>Accuracy (xview)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Li et al. [33]</td>
<td>RGB</td>
<td>62.5</td>
</tr>
<tr>
<td>Vyas et al. [57]</td>
<td>RGB</td>
<td>83.1</td>
</tr>
<tr>
<td>DA-Net [58]</td>
<td>RGB</td>
<td>86.5</td>
</tr>
<tr>
<td>ViewCLR [11]</td>
<td>RGB</td>
<td>89.1</td>
</tr>
<tr>
<td>Ours</td>
<td>RGB</td>
<td>91.7</td>
</tr>
</tbody>
</table>

4.5. Synthetic Data for Action Recognition

It can be difficult to collect and annotate activity videos, especially in a synchronized multi-view capture setting. In addition, collecting videos involving humans raises privacy concerns. We highlight that synthetic data [15] provides a practical alternative for creating custom multi-view action recognition datasets. Using a simulator, it is relatively easy to collect synchronized videos for each instance, with desired diversity and without any privacy issues. RoCoG [14] is a gesture recognition dataset consisting of videos from seven gestures. The dataset allows benchmarking on a realistic scenario where we have a large set of synthetic data collected from multiple viewpoints and a small set of real data for testing. For this dataset, we use the multi-view synthetic data for training and evaluate on real test data, and show that our method results in better domain generalization as opposed to standard cross-entropy training. We use the train/test split provided in [14].

We also compare our approach to a baseline that is trained on real data alone. This gives us the upper bound performance for models trained using synthetic data. Next we train the same backbone on synthetic data using the cross-entropy loss. This results in performance drop of 12.38%. Finally, we train using our proposed loss on synthetic data, and show that the domain gap reduces from 12.38% to 4.76%. The results of these experiments are presented in Table 4. The results show that features learned using our approach show much better domain generalization performance compared to standard loss function used for action recognition. These results, along with the transfer learning results show that our method can generalize well across different scenarios.

4.6. Ablation Studies

We perform extensive ablation experiments to study the effect of different contributions and design choices of our method. All ablation experiments are performed on the cross-subject benchmark of NTU-60 dataset. We use a smaller dataset (NTU-60-small) for training, which consists of half of the subjects from the original cross-subject training set chosen randomly, and perform testing on the full test set of NTU-60 cross-subject benchmark.

Effect of View Contrastive Loss. We compare with different standard loss functions in Table 5 to show the effect of incorporating viewpoint information in the loss. Specifically, we train the same backbone models using cross-entropy, SupCon [29] and our loss functions. We can see that our loss function, which makes use of multi-view information, leads to better performance as expected. For our loss function (Eq. 4), we modify the SupCon loss (Eq. 2) by removing other positive samples from the denominator.

Effect of Hard Negatives. We show the effectiveness of our approach of incorporating hard negatives in the contrastive loss in Table 6. We train our model without using the re-weighting of hard negatives, i.e. setting weight of all negatives to one, giving them equal importance. From the table, we can see that incorporating hard negatives improves the performance over treating all negatives equally. We also compare with HCL [47], a recent method which proposes...
Table 5: Effectiveness of the improved supervised contrastive loss. Compared to standard cross-entropy and SupCon, the loss function leads to higher accuracy. Ours-A uses all positives in denominator (as in SupCon) whereas Ours is the improved version. Models are trained on NTU-60-small training set.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (xsub)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cross-Entropy</td>
<td>78.44</td>
</tr>
<tr>
<td>SupCon</td>
<td>78.47</td>
</tr>
<tr>
<td>Ours-A</td>
<td>78.89</td>
</tr>
<tr>
<td>Ours</td>
<td>79.75</td>
</tr>
</tbody>
</table>

Table 6: Effectiveness of our hard negative selection method on NTU-60-small. Compared to HCL, our method leads to better hard negatives. The proposed method selects all negative samples from hard classes, which is better than choosing a fixed number of hard classes.

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy (xsub)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HCL</td>
<td>78.86</td>
</tr>
<tr>
<td>Ours w/o hard negatives</td>
<td>79.43</td>
</tr>
<tr>
<td>Ours w/ top-3</td>
<td>79.54</td>
</tr>
<tr>
<td>Ours</td>
<td>79.75</td>
</tr>
</tbody>
</table>

Table 7: Effect of holding out different viewpoints. We conduct three experiments, holding out one viewpoint for testing and train on the remaining two views.

<table>
<thead>
<tr>
<th></th>
<th>Front view</th>
<th>±45° view</th>
<th>±90° view</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>97.1</td>
<td>98.0</td>
<td>96.9</td>
</tr>
</tbody>
</table>

Figure 4: t-SNE visualizations. In this figure, we visualize the t-SNE plots on ten classes of the NTU-60 test set for two approaches: SupCon and ViewCon. We can clearly see that our approach of incorporating view invariance and use of hard negatives improves the learned representations.

Please refer to the supplementary material to find single-crop results and details on data augmentations and RoCoG experiments.

5. Conclusion

In this work, we present an approach for multi-view action recognition. We make use of an improved version of supervised contrastive learning with the set of positives augmented by synchronized views of clips in addition to augmentations from the video. We also propose a novel technique to reweigh hard negatives guided by the classifier, thus learning rich feature representations. We demonstrate the superiority of our approach through comparisons on multi-view data from NTU-60, NTU-120 and NUMA. In addition, experiments on synthetic data from RoCoG show the generalizability nature of our approach.

6. Acknowledgements

The authors would like to thank Aniket Roy for helpful discussions. This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-21-2-0211.
References


