IntegralAction: Pose-driven Feature Integration for Robust Human Action Recognition in Videos

Gyeongsik Moon∗1 Heeseung Kwon∗2 Kyoung Mu Lee1 Minsu Cho2

1SNU ECE & ASRI
{mks0601, kyoungmu}@snu.ac.kr
2POSTECH CSE & AIGS
{aruno, mscho}@postech.ac.kr

Abstract

Most current action recognition methods heavily rely on appearance information by taking an RGB sequence of entire image regions as input. While being effective in exploiting contextual information around humans, e.g., human appearance and scene category, they are easily fooled by out-of-context action videos where the contexts do not exactly match with target actions. In contrast, pose-based methods, which takes a sequence of human skeletons only as input, suffer from inaccurate pose estimation or ambiguity of human pose per se. Integrating these two approaches has turned out to be non-trivial; training a model with both appearance and pose ends up with a strong bias towards appearance and does not generalize well to unseen videos. To address this problem, we propose to learn pose-driven feature integration that dynamically combines appearance and pose streams by observing pose features on the fly. The main idea is to let the pose stream decide how much and which appearance information is used in integration based on whether the given pose information is reliable or not. We show that the proposed IntegralAction achieves highly robust performance across in-context and out-of-context action video datasets.

1. Introduction

Human action recognition in videos aims at classifying an input video of human action into one of pre-defined target classes [4, 6, 11, 12, 22, 25, 33, 43, 45]. Following the success of convolutional neural networks (CNNs) on image classification [15, 34], video action recognition has made remarkable progress by developing deep neural models that process RGB image frames via spatio-temporal convolution [4, 11, 22, 43] or two-stream convolution with scene optical flow [12, 33]. These appearance-based methods, however, learn to exploit contextual information (e.g., scene class, dominant objects, or background motion), rather than human action performed in the video [23, 24]. This is a critical issue in terms of robustness since they are all vulnerable to the attack of out-of-context actions, e.g., mimes [40]. While recent pose-based methods [6, 22, 44, 45], which replace the RGB input with human skeletons, have a potential to resolve this issue as a reasonable alternative, the situation is very difficult in the wild. For most of real-world videos in the standard benchmark datasets, human poses are not easy to detect, only partially visible (mostly, close-up

* equal contribution

1Seoul National University, Seoul, Korea
2Pohang University of Science and Technology, Pohang, Korea

Figure 1: Our robust integration of appearance-based and pose-based approaches. For both cases of (a) in-context action and (b) out-of-context action videos, it predicts correct actions by leveraging more reliable information. Representative frames and their class activation maps are visualized with predicted action classes. Blank images in the input pose sequence indicates a pose estimation failure. See text.
faces and hands), or completely absent (first-person view without any persons). Even with successful detection, human poses are often ambiguous. As shown in the example in Fig. 1(a), while the appearance-based approach successfully predicts “eating ice cream” by exploiting the appearance of ice creams, the pose-based approach often fails due to the ambiguity of the pose without the appearance context. In contrast, when successfully detected, the pose-based approach has robustness against contextual bias. As shown in the example of Fig. 1(b), while the appearance approach is misled by the appearance of barbels, the pose-based approach correctly predicts “hurdling”. To tackle the issue, several methods [6, 9, 27, 39, 44, 45, 47] attempt to integrate the two approaches; they either aggregates the two streams by averaging predicted action scores [6, 9, 27, 39, 44, 45] or fusing features from the two streams by fully-connected layers [27, 47]. Integrating appearance and pose, however, has turned out to be non-trivial. Recent research [40] shows that their models all inherit a strong bias towards contextual information present in the datasets [23, 24] and thus are easily fooled by out-of-context action videos.

To address the problem, we propose an effective and robust integration model, dubbed IntegralAction, that dynamically combines appearance and pose features in a pose-driven manner. The main idea is to let the pose stream decide how much and which appearance information is used in integration based on whether the given pose information is reliable or not. It thus encourages the system to filter out unnecessary contextual information and focus on human motion information when the pose information is sufficient for action recognition. In experimental evaluation, we demonstrate that the proposed pose-driven feature integration greatly improves action recognition on out-of-context action videos, Mimetics [40] dataset, while preserving good performance on in-context action videos, Kinetics [19] and NTU-RGBD [31] datasets.

2. Related work

**Appearance-based human action recognition.** The appearance-based human action recognition methods rely on raw RGB video frames for prediction. There exist different approaches to it using Recurrent Neural Networks (RNNs) [8], CNNs [12, 33], and 3D CNNs [4, 35, 36, 33]. Simonyan et al. [33] introduce two-stream networks with RGB and optical flow streams, which process RGB frames together with corresponding optical flow frames; each stream is trained separately, and softmax outputs from the two streams are then averaged to obtain a final prediction. Feichtenhofer et al. [12] propose similar two-stream networks that combine the two-stream features using convolution layers. Carreira et al. [4] introduce the inflated 3D CNN (I3D) that expands ImageNet [30] pre-trained kernels of 2D CNN to 3D. They verify 3D CNNs with large-scale in-context action dataset such as Kinetics [19] become strong baselines in action recognition. Du et al. [36] and Xie et al. [43] factorize 3D convolutions into 2D and 1D convolutions for efficient prediction. Lin et al. [25] propose the temporal shifting module (TSM) that enables them to learn spatio-temporal features using 2D convolutions only.

**Pose-based human action recognition.** The pose-based human action recognition methods use human skeleton or pose videos for prediction. The pose video can be obtained from depth sensors [20, 26] or off-the-shelf 2D multi-person pose estimation models [3, 14]. The earlier work of Jhuang et al. [18] show that human pose features can improve human action recognition in videos. Du et al. [10] introduce hierarchical RNNs that use separated models for decomposed human skeleton parts according to a skeleton hierarchy. Zhu et al. [46] propose RNNs with long short-term memory (LSTM) for modeling long-term temporal dependencies. Cheron et al. [5] construct two-stream networks that process RGB frames and optical flows of human parts. Choutas et al. [6] introduce a concept of pose motion that represents the temporal movement of human pose in a spatial domain. Yan et al. [44] propose a unified 3D CNN model that effectively encodes multiple pose modalities. Du et al. [9] and Rohit et al. [13] use estimated human poses for attention mechanisms in their model by converting human keypoints to 2-dimensional heatmaps. Recently, Graph Convolution Networks (GCNs) have been used for skeleton-based action recognition [22, 32, 45]. Yan et al. [45] construct a spatio-temporal human pose graph and process it by GCN. Shi et al. [32] introduce a GCN-based action recognition system that learns graph topology. Li et al. [22] propose an encoder-decoder structure to capture action-specific latent high-order dependencies in their GCN-based model.

**Integrating appearance and pose.** Several methods have attempted to improve performance in action recognition by integrating the two approaches [6, 9, 13, 27, 39, 44, 45, 47]. Most of the methods [6, 9, 27, 39, 44, 45] use to simply add predicted action scores from the two models in their testing stage. Rohit et al. [13] propose to use a bilinear pooling between appearance and pose features for their action classifier. Zolfaghari et al. [47] develop a multi-stream 3D CNN for processing multi-frame human poses, optical flows, and RGBs together. Luvizon et al. [27] propose a multi-task system that simultaneously predicts 2D/3D human pose and action class. Compared to our method, all these integration methods have serious limitations in robustness to in-context or out-of-context actions. Their models heavily rely on the appearance stream without any gating or regularization so that they are not able to filter out a strong contextual bias from out-of-context action videos. Furthermore, the post-processing integration of [6, 9, 27, 39, 44, 45] is easily affected by inaccurate action prediction from the pose stream. Our dynamic integration method of IntegralAction effectively
A simple gating block for language modeling. Hu et al. propose information with useful contextual appearance. Gating for neural networks. Gating mechanisms control the information flow of neural networks through a multiplicative interaction and are being widely used for a variety of tasks [7, 16, 28, 43]. For example, Dauphin et al. [7] use a simple gating block for language modeling. Hu et al. [16] introduce a squeeze-and-excitation block for image classification. For video classification, Miech et al. [28] and Xie et al. [43] propose gating modules to reassign the distribution of feature channels. In this paper, we propose a new gating mechanism, which controls the information flow of appearance and human pose information through dynamic integration for recognizing human actions.

Our contributions can be summarized as follows.

- We present an action recognition model, IntegralAction, that dynamically integrates appearance and pose information in a pose-driven way.
- We show that our pose-driven feature integration significantly improves the action recognition accuracy on out-of-context action videos while preserving the accuracy on in-context action videos.
- Our IntegralAction achieves highly robust performance across in-context and out-of-context action video datasets. Especially, it significantly outperforms the recent state-of-the-art methods on out-of-context action video dataset.

3. IntegralAction model

Given a video clip, the proposed model, IntegralAction, takes as input both its appearance sequence $A$ and the corresponding pose estimation sequence $P$, and then predicts action probabilities of the video over target classes $C$. The appearance sequence is composed of RGB frames, while the pose sequence is of human pose frames. As illustrated in Fig. 2, IntegralAction processes the two sequences, $A$ and $P$, via appearance and pose streams; these two streams transform $A$ and $P$ into spatio-temporal appearance and pose features, respectively. The proposed pose-driven feature integrator then combines the two features into an integrated action feature. Finally, the action feature is fed to a classifier, which outputs action probabilities over target classes $C$.

3.1. Appearance stream

For an input to the appearance stream, we sample $T$ frames from the given video clip and construct a sequence $A \in \mathbb{R}^{T \times H_A \times W_A}$ where $T$, $H_A$, and $W_A$ denote the numbers of frames, height, and width of the sequence, respectively. The frames are sampled by randomly choosing the first frame within the video clip and then collecting subsequent frames at $\tau$ frame intervals $[4, 43]$. The frames are resized and center-cropped into the size of $H_A \times W_A$. Given the sequence of RGB frames $A$, the appearance stream transforms it into an appearance feature $F_A \in \mathbb{R}^{T \times C}$.

For an architecture of the stream, we employ ResNet [15] and add the temporal shift module (TSM) [25] for each residual block of the ResNet for efficient and effective spatio-temporal feature extraction following Lin et al. [25]. The TSM enables to obtain the effect of 3D convolutions using efficient 2D convolutions by shifting a part of input feature channels along the temporal axis before the convolution operation. Following the setting in [25], we shift 1/8 of the input feature channels forward and another 1/8 of the channels backward in the TSM. The final appearance feature $F_A$ is obtained by performing a global average pool on the output of the ResNet.

3.2. Pose stream

For an input to the pose stream, we sample $T$ frames from the video clip using the same sampling scheme used in the appearance stream, and extract human body keypoints from the frames using off-the-shelf 2D multi-person pose estimation methods [3, 42]. The keypoints for each frame are then translated into $K$ keypoint heatmaps [2, 42] and $B$ part affinity fields [4], which will be described below. Combining the heatmaps and fields along channels, we construct a sequence $P \in \mathbb{R}^{T \times (K+B) \times H_P \times W_P}$ where $H_P$ and $W_P$ represent the height and width of the sequence, respectively. Given the sequence of human pose frames $P$, the pose stream transforms it into a pose features $F_P \in \mathbb{R}^{T \times C}$.

For an architecture of the stream, we employ the same ResNet model used in the appearance stream but with some modification for memory efficiency. Specifically, we skip the first two-strided convolutional layer and the next max-pooling layer of the ResNet; this allows us to set $H_P$ and $W_P$ to one-quarter of $H_A$ and $W_A$, respectively while preserving the spatial size of the output to be the same as that from the appearance stream. As the channel dimension of $P$, $K + B$, is relatively large, we reduce the spatial size of $P$ for efficient feature extraction. To adapt the channel dimension
of P to the input channel dimension of the first block of the ResNet, we put a front-end convolutional block that consists of a 3-by-3 convolutional layer, batch normalization [17], and a ReLU function.

**Keypoint heatmap.** Each keypoint is translated into a heatmap $H_k$ using a Gaussian blob [2,42]: $H_k(x, y) = \exp \left( \frac{-(x-x_k)^2+(y-y_k)^2}{2\sigma^2} \right)$, where $x_k$ and $y_k$ are the coordinates of $k$th keypoint and $\sigma$ is set to 0.5. When multiple persons are detected in the frame, we sort them by the pose estimation score and select top-5 persons. If the score is below 0.1, the keypoint heatmap of the person becomes zero heat map. Then, we accumulate heatmaps of each keypoint from the persons and clamp the value to a maximum of 1.0.

**Part affinity field (PAF).** The PAF [3] is a vector field between human keypoint locations in the image space. For its construction, we define a bone as a pair of parent and child nodes in the human skeleton graph. To translate the $b$th bone of a detected human into a PAF $L_b$, we assign the 2D orientation vector of the corresponding keypoint pair to all positions $(x, y)$ on the line formed by the keypoint pair: $L_b(x, y) = v_b$ where $v_b \in \mathbb{R}^2$ is the 2D unit vector in the direction of the $b$th bone [3]. Given a pose estimator (e.g., [2]), we sort its pose estimation results by the scores and select top-5 persons to make PAFs. When a position is assigned bones of multiple persons, it obtains the average of all the corresponding vectors. PAFs implicitly provide information about which person each keypoint belongs to, e.g., greedy parsing of PAFs would be able to reconstruct the keypoint connections (i.e., bones) of a person. PAFs are thus useful in the presence of multiple persons and complementary to the keypoint heatmaps.

**3.3. Pose-driven feature integration**

To combine the appearance and pose features, $F_A$ and $F_p$, into an adaptive and robust action feature $F$, we propose to learn pose-driven feature integration. As illustrated in Fig. 3, it proceeds in three steps: (1) feature alignment, (2) pose-driven gating, and (3) aggregation.

**Feature alignment.** The appearance and pose features, $F_A$ and $F_p$, are transformed into $F_A'$ and $F_p'$ so that they are semantically aligned (i.e., channel-wise aligned) to each other with the same channel dimension. This is done by two temporal convolution blocks, TCB$A$ and TCB$P$, each of which consists of convolution with kernel size 1, layer normalization [1], and ReLU activation function; the two temporal convolution blocks are separately applied to $F_A$ and $F_p$, respectively, and produce outputs with the same channel dimension as inputs. This semantic alignment along channels is necessary for the two features, $F_A'$ and $F_p'$, to be integrated by addition after gating.

**Pose-driven gating.** The pose-driven gating takes as input the pose feature $F_p$, predicts a channel-wise gating matrix $G \in \mathbb{R}^{T \times C}$, and imposes it on semantically aligned features $F_A'$ and $F_p'$. The gating matrix $G$, where each element lies in $[0, 1]$, is used to perform gating on $F_A'$ and $F_p'$ in opposite directions: $G \odot F_A' + (1 - G) \odot F_p'$, where $\odot$ denotes element-wise multiplication.

**Aggregation.** Finally, the aggregation simply combines the gated features of appearance and pose by element-wise addition, which produces an integrated action feature $F \in \mathbb{R}^{T \times C}$ as follows:

$$F = G \odot F_A' + (1 - G) \odot F_p'. \quad (1)$$

The main idea behind the pose-driven feature integration is to let the pose feature decide how much and which appearance information is used in integration based on whether the given pose information is reliable or not. To this end, we enforce it to learn a priority for pose by training the gating module with a regularizer:

$$L_{\text{gate}} = -\log(1 - G), \quad (2)$$

which encourages the gate to be open to the appearance feature only when the cost needs to be taken based on the given pose feature. This prevents the integration from being biased towards strong contextual features from the appearance stream. If the pose feature provides sufficient information for action recognition, the proposed integration will focus on it without the risk of being misled by contexts from the appearance feature. Otherwise, it will utilize more contextual information from the appearance feature.

**3.4. Action classification**

The final classifier uses the integrated action feature $F$ to predict probabilities over target action classes. The classification is performed for each frame of $F$ by a fully connected layer with softmax output. As in [25,38], we
average the probabilities of the frames to obtain the final action probabilities of the video.

We train our model, IntegralAction, by minimizing the loss function:

$$ L = L_{\text{cls}} + \lambda L_{\text{gate}}. $$

where $L_{\text{cls}}$ represents the standard cross-entropy loss and $L_{\text{gate}}$ is the gate regularizer described above. $\lambda$ is a balancing factor between them.

The training procedure proceeds as follows. We first train an appearance-only model and a pose-only model, which use $F_A$ and $F_P$ for classification, respectively, by minimizing $L_{\text{cls}}$ only. These pre-trained appearance-only and pose-only models are used to initialize the IntegralAction except for the parts of feature integration and action classification, which are randomly initialized. The feature integration and classification parts are then trained by minimizing the entire loss $L$ while the pre-trained parts of the network are frozen.

3.5. Implementation details

PyTorch [29] is used for implementation. The ResNet is initialized with the publicly released weight pre-trained on the ImageNet dataset [30], and the weights of the remaining part are initialized by Gaussian distribution with $\sigma = 0.001$. The weights are updated by the SGD with a mini-batch size of 32. $C = 512$ is used for all experiments. We set the size of the input RGB sequence as $224 \times 224$, and that of the input pose sequence as $56 \times 56$. When training the appearance and pose stream on Kinetics [19], the initial learning rate is set to $10^{-2}$ and reduced by a factor of 10 at 30th and 60th epoch until 70th epoch. For Kinetics50 [19] and NTU-RGBD [31], the initial learning rate is set to $10^{-2}$ and reduced by a factor of 10 at 20th and 30th epoch until 40th epoch. Then, for the final training on all datasets, the initial learning rate is set to $10^{-3}$ and reduced by a factor of 10 at 10th and 15th epoch until 20th epoch. We use four NVIDIA RTX 2080Ti GPUs for training our IntegralAction. In testing time, we sampled 10 clips per video and averaged the prediction following the work of [25].

4. Experiments

4.1. Datasets and evaluation metrics

Kinetics. Kinetics [19] is a large-scale human action dataset mostly with in-context action videos. The dataset consists of 240K training and 20K validation videos with 400 action classes. Following the work of [40], we also use a subset of Kinetics, Kinetics50, which contains 33K training and 2K validation videos with 50 action classes; these 50 action classes are strongly related with human body motions rather than background scene and objects. For human pose estimation on Kinetics, we use the results released by Yan et al. [45], which are obtained using OpenPose [2].

Mimetics. Mimetics [40] is a human action dataset mostly with out-of-context action videos. It consists of 713 videos and 50 human action classes, which are the same as those of Kinetics50. Due to its small scale, it is only used to evaluate models trained on Kinetics50 or Kinetics following Weinzaepfel et al. [40]. For human pose estimation on Mimetics, we use ResNet50-based Mask R-CNN [14, 41].

NTU-RGBD. NTU-RGBD [31] contains 56K RGBD video clips captured from a controlled lab environment. 40 subjects perform 60 actions, and most of the videos correspond to in-context actions. We use the standard cross-subject split for training and testing and exploit 2D human pose annotations provided by [31].

Following the previous results reported on Kinetics, Mimetics [40], and NTU-RGBD [22, 32, 40, 45], both top-1 and top-5 accuracies are used as evaluation metrics for Kinetics and Mimetics while top-1 accuracy is used as an evaluation metric for NTU-RGBD.

4.2. Results on the benchmark datasets

In this experiment, we use ResNet-50 for the appearance model and ResNet-18 for the pose model. We set $T = 8$ and $\tau = 8$ for the appearance model, and $T = 32$ and $\tau = 2$ for the pose model. The same settings are used for appearance and pose streams of our integrated model. We apply $4 \times 1 \times 1$ average pooling along the temporal axis for $F_P$ to set the same feature dimension as that of $F_A$ before the feature integration.

Comparison to the baselines. First of all, we compare single-stream models with the proposed integrated model on different benchmark datasets. Since the balancing factor $\lambda$ controls the degree of pose priority in dynamic gating, we use two versions of IntegralAction: weak priority ($\lambda = 1.5$) and strong priority ($\lambda = 5.0$). The results are summarized in Table [1]. The appearance-only model shows strong performance on the in-context action datasets (Kinetics and NTU-RGBD) but drastically fails on the out-of-context action dataset (Mimetics). The pose-only model shows the opposite; it achieves strong performance on Mimetics while significantly underperforming others on Kinetics and NTU-RGBD. The proposed method, IntegralAction, presents robust performance on all the benchmark datasets. It achieves good performance on Mimetics while being comparable to appearance-only model on Kinetics. In NTU-RGBD, our method obtains an additional gain by integrating appearance and pose information. The last column of Table [1] represents the average rank over all the three datasets, which shows that the proposed method performs the best on average.

Comparison to the state-of-the-art methods. We compare our results with recent state-of-the-art results on Kinetics and Mimetics in Table [2]. Each section of the table contains the results of appearance-based methods [4, 25, 36, 40], pose-based methods [21, 31, 40, 45], and the proposed
Table 1: Top-1 & top-5 accuracy comparison with baselines on Kinetics, Mimetics, and NTU-RGBD.

| methods                      | Kinetics | Mimetics | NTU-RGBD | avg rank |
|------------------------------|----------|----------|----------|----------|
|                              | top-1    | top-5    | top-1    | top-5    | rank    |
| appearance-only model        | 73.5     | 91.0     | 6.3      | 16.7     | 2.7     |
| pose-only model              | 30.2     | 51.3     | 15.2     | 33.1     | 3.2     |
| IntegralAction (ours, $\lambda = 1.5$) | 73.3     | 90.8     | 12.8     | 26.0     | 1.8     |
| IntegralAction (ours, $\lambda = 5.0$) | 65.0     | 85.9     | 15.3     | 31.5     | 2.2     |

Table 2: Top-1 & top-5 accuracy on Kinetics and Mimetics.

| methods                      | Kinetics | Mimetics |          |          |
|------------------------------|----------|----------|----------|----------|
|                              | top-1    | top-5    | top-1    | top-5    |
| Appearance-based methods     |          |          |          |          |
| I3D (RGB) [4]                | 67.1     | 89.3     | -        | -        |
| R(2+1)D (RGB) [36]           | 72.0     | 90.0     | -        | -        |
| TSM [25]                     | 73.5     | 91.0     | 6.3      | 16.7     |
| 3D ResNeXt-101 (RGB) [40]    | 74.5     | -        | 8.6      | 20.1     |
| 3D ResNeXt-101 two-stream [40] | -     | -        | 10.5    | 26.9     |
| Pose-based methods           |          |          |          |          |
| Deep LSTM [31]               | 16.4     | 35.3     | -        | -        |
| TCN [21]                     | 20.3     | 40.0     | -        | -        |
| ST-GCN [45]                  | 30.7     | 52.8     | 12.6     | 27.4     |
| SIP-Net [40]                 | 32.8     | -        | 14.2     | 32.0     |
| IntegralAction (ours, $\lambda = 1.5$) | 73.3     | 90.8     | 12.8     | 26.0     |
| IntegralAction (ours, $\lambda = 5.0$) | 65.0     | 85.9     | 15.3     | 31.5     |

Table 3: Top-1 accuracy on NTU-RGBD.

| methods                      | NTU-RGBD |          |          |
|------------------------------|----------|----------|----------|
| Lie Group [37]               | 50.1     |          |          |
| Du et al. [10]               | 59.1     |          |          |
| Deep LSTM [31]               | 60.7     |          |          |
| SIP-Net [40]                 | 64.8     |          |          |
| Zolfaghari et al. [47]       | 67.8     |          |          |
| TCN [21]                     | 74.3     |          |          |
| ST-GCN [45]                  | 81.5     |          |          |
| AS-GCN [22]                  | 86.8     |          |          |
| 2S-AGCN [32]                 | 88.5     |          |          |
| IntegralAction (ours, $\lambda = 1.5$) | 91.7     |          |          |
| IntegralAction (ours, $\lambda = 5.0$) | 91.0     |          |          |

As expected, appearance-based methods are better than pose-based methods on Kinetics, while pose-based methods are better than appearance-based methods on Mimetics. The proposed method is competitive with the appearance-based methods [4, 25, 36, 40], and outperforms all the other pose-based methods [21, 31, 40, 45] by a large margin in Kinetics. At the same time, our method achieves state-of-the-art performance in Mimetics. Table 2 summarizes the comparative results on NTU-RGBD. The upper part of the table contains the results of previous pose-based methods including a hand-crafted feature method [37], RNN methods [10, 31], CNN methods [21, 40, 47], and GCN methods [22, 32, 45]. The proposed method outperforms all the other methods, outperforming the state of the arts by the margin of 3.2% points at top-1 accuracy.

4.3. Comparison with other integration methods

We compare top-1 and top-5 accuracy from appearance-only model, pose-only model, and different feature integration methods including feature fusion [27, 47] and score average in testing stage [6, 9, 39, 44, 45] in Table 4. To implement the feature fusion, we pass $F_P$ and $F_A$ to an additional fully-connected layer, of which output is used for the action prediction. We additionally train variants of our IntegralAction by disabling the gating or gating without $L_{\text{gate}}$. Among all the integration methods, our method achieves the best accuracy on Mimetics and marginally lower accuracy than the best performing one on Kinetics50. The relative performance difference between other best performing integration method and ours on Mimetics is 23%, while that on Kinetics50 is 3%. For all models, the network architectures are based on that of our IntegralAction. The appearance and pose streams are based on ResNet-18, and we set $T = \tau = 8$. All models are trained on Kinetics50 and then tested on Kinetics50 and Mimetics.

Comparison with the feature fusion. The feature fusion method [27, 47] suffers from low accuracy on Mimetics because their networks are trained without any gating or regularization; therefore theirs cannot filter out a strong con-
Table 4: Top-1 and top-5 accuracy comparison between various appearance and pose stream integration methods on Kinetics50 and Mimetics. The numbers in parentheses represent relative difference between other best performing integration method and ours.

| methods                     | Kinetics50 | Mimetics |
|-----------------------------|------------|----------|
|                             | top-1      | top-5    | top-1      | top-5    |
| appearance-only model       | 72.8       | 91.7     | 11.2       | 31.7     |
| pose-only model             | 45.6       | 72.9     | 26.0       | 52.2     |
| feature fusion [27, 47]     | 73.8       | 93.4     | 19.5       | 42.3     |
| score average [6, 44]       | 73.9       | 91.4     | 21.6       | 48.7     |
| ours without gating         | 74.2       | 93.2     | 21.3       | 45.5     |
| ours without $L_{gate}$     | 73.2       | 92.1     | 15.9       | 37.2     |
| IntegralAction (ours)       | $72.2$ ($\downarrow 3\%$) | $92.3$ ($\downarrow 1\%$) | $26.5$ ($\uparrow 23\%$) | $50.5$ ($\uparrow 6\%$) |

Table 5: Class-wise average gating $G$ on Kinetics50.

| action classes  | average $G$ |
|-----------------|-------------|
| writing         | 0.33        |
| flying kite     | 0.32        |
| tying tie       | 0.31        |
| driving car     | 0.28        |
| hitting baseball| 0.18        |
| skipping rope   | 0.17        |
| deadlifting     | 0.16        |
| clean and jerk  | 0.15        |

textual bias from out-of-context action videos. In contrast, our IntegralAction is not dominated by the biased contextual information by introducing the regularizer $L_{gate}$, which results in significant performance improvement on Mimetics as shown in 6th and the last row of the Table 4. In order to further validate this, we visualize in Figure 4 the distributions of the gating $G$ averaged over the channel dimension from models, which are trained with and without $L_{gate}$, respectively, on Kinetics50 and Mimetics. For better visualization, we fit Gaussians to the original distributions. As Figure 4 shows, the regularizer translates the mean of each distribution to a lower value, which enforces our system to prefer pose features over the appearance features. Our IntegralAction thus utilizes the pose feature when the input pose sequence provides sufficient information for action recognition without being dominated by the appearance feature.

**Comparison with the score average.** The score averaging methods [6, 9, 39, 44, 45] average the predicted action probability from the appearance-only and pose-only models in testing stage. This makes their system suffer from low accuracy on Mimetics because strongly biased action prediction from the appearance-based model is not filtered out when averaging. In contrast, our IntegralAction dynamically integrates the appearance features and pose features. Figure 4 shows that average distributions of the gating $G$ from Kinetics50 and Mimetics have very different variances, which indicates that the estimated gating matrix varies according to the input sequence. As some videos of Kinetics50 contain human-central videos while others contain context-central videos, the importance of the input pose sequence for action recognition significantly varies, which makes the gating values diverge. In contrast, most of the videos of Mimetics contain human-central videos, which results in gating values with less variance. We also show the average values of gating matrix $G$ for different action classes in Table 5. It shows that the gating value tends to be lower when the action class is highly related to human motion and becomes higher when the class is related to the context such as background and objects. This dynamic feature integration is essential for filtering out the biased action prediction from the appearance feature.

**Performance on Kinetics50.** Although our IntegralAction outperforms other integration methods on Mimetics by a large margin, it decreases the top-1 accuracy of the appearance-only model on Kinetics50 slightly while increasing the top-5 accuracy. We believe this is because our regularizer is applied to all frames equally, which can enforce our model to prefer the pose feature even when the appearance feature plays a critical role in the frames for the action recognition. While we have tried to utilize the action recognition results on the training frames to supervise the gating matrix differently for each frame, it did not improve the action recognition accuracy in a meaningful margin. We leave designing a frame (or video)-adaptive regularizer as future work.

4.4. Ablation study on IntegralAction

For this ablation study, we use ResNet-18 and set $T = \tau = 8$ for both appearance and pose streams. The model is trained on Kinetics50 and then tested on Kinetics50 and Mimetics.

**Pose-driven vs. appearance-driven feature integration.** To demonstrate the effectiveness of the pose-driven feature integration, we compare the top-1 and top-5 accuracy be-
Table 6: Top-1 & top-5 accuracy comparison on Kinetics50 and Mimetics between different models that estimate gating from appearance feature, pose feature, and both of them, respectively. The numbers in parentheses represent the rate of improvement from the other best performer.

| methods          | Kinetics50 | Mimetics |
|------------------|------------|----------|
|                  | top-1  | top-5  | top-1  | top-5  |
| appearance-driven| 73.8   | 93.0   | 23.2   | 48.2   |
| pose-driven (ours)| 72.2 (↓2%) | 92.3 (↓1%) | 26.5 (↑12%) | 50.5 (↑5%) |
| both-driven       | 73.3   | 92.4   | 23.7   | 46.5   |

Table 7: Top-1 & top-5 accuracy of models trained with different $\lambda$ values on Kinetics50 and Mimetics.

| $\lambda$ value | Kinetics50 | Mimetics |
|-----------------|------------|----------|
|                 | top-1  | top-5  | top-1  | top-5  |
| 0.0             | 73.2   | 92.1   | 15.9   | 37.2   |
| 1.0             | 72.8   | 92.6   | 25.7   | 47.8   |
| 1.5 (ours)      | 72.2   | 92.3   | 26.5   | 50.5   |
| 5.0             | 67.8   | 89.7   | 27.2   | 48.2   |

Table 8: Top-1 and top-5 accuracy comparison between models trained with various training strategies on Kinetics50 and Mimetics.

| pre-train | fix | Kinetics50 | Mimetics |
|-----------|-----|------------|----------|
|           |     | top-1  | top-5  | top-1  | top-5  |
| $\times$  | $\times$ | 72.2 | 93.6   | 17.8   | 43.1   |
| $\checkmark$ | $\times$ | 72.5 | 92.0   | 23.5   | 46.6   |
| $\checkmark$ | $\checkmark$ | 72.2 | 92.3   | 26.5   | 50.5   |

We combined them by a concatenation along the channel dimension. As the table shows, our pose-driven feature integration significantly outperforms appearance-driven and both-driven ones on Mimetics, while produces marginally lower accuracy on Kinetics50. The relative performance gap on Mimetics is 12%, while that on Kinetics50 is 2%. The reason for worse performance of the appearance-driven and both-driven ones on Mimetics is that the appearance feature provides strong contextual information and forces CGB, which estimates the gating matrix, to be dominated by the contextual information. In contrast, pose feature without contextual information can make CGB robust to the biased context, which leads to more robust performance on out-of-context action videos.

**Results with different training settings.** We provide results using different $\lambda$ values in Table 7. As the table shows, a larger $\lambda$ value regularizes the model to choose the feature from the pose stream stronger, which results in low performance on Kinetics50 and good performance on Mimetics. We found that $\lambda = 1.5$ achieves high accuracy on Mimetics while marginally decreases the accuracy on Kinetics50; therefore we set $\lambda = 1.5$ through the whole experiments.

In addition, in Table 8 we compare ours with two other training strategies. The first one is to train from scratch without pre-training (the first row), and the second one is to train with pre-training together with fine-tuning (the second row), meaning that the pre-trained streams are not fixed when training feature integration and classification part. As the table shows, starting from pre-trained appearance and pose streams greatly improves the accuracy on Mimetics. In addition, fixing the pre-trained two streams also increases the accuracy on Mimetics significantly.

We think our strategy, i.e., pre-training each stream separately and fixing them, can maintain their complementary characteristics better than training the whole system from scratch or fine-tuning them during the final training stage. This may be crucial because maintaining the complementary characteristics of each stream and utilizing them is the essence of our IntegralAction.

**5. Conclusion**

We propose IntegralAction, which dynamically integrates appearance and pose information in a pose-driven manner for robust human action recognition. The previous integration methods use a static aggregation of the two information or sequentially refine the pose information using the appearance information, which suffers from a strong bias of contextual information from out-of-context action videos. In contrast, our pose-driven feature integration can filter out the biased contextual information, thus can perform robust action recognition on both in-context and out-of-context action videos. The proposed IntegralAction achieves highly robust performance across in-context and out-of-context action video datasets.

**Acknowledgments**

This work was supported by the Next-Generation Information Computing Development Program (NRF-2017M3C4A7069369) and the Visual Turing Test project (IITP-2017-0-01780) funded by the Ministry of Science and ICT of Korea.
References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. arXiv preprint arXiv:1607.06450, 2016.

[2] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Openpose: realtime multi-person 2d pose estimation using part affinity fields. arXiv preprint arXiv:1812.08008, 2018.

[3] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In CVPR, 2017.

[4] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In CVPR, 2017.

[5] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-cnn: Pose-based cnn features for action recognition. In ICCV, 2015.

[6] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. Potion: Pose motion representation for action recognition. In CVPR, 2018.

[7] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In ICML, 2017.

[8] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In CVPR, 2015.

[9] Wenbin Du, Yali Wang, and Yu Qiao. Rpan: An end-to-end recurrent pose-attention network for action recognition in videos. In ICCV, 2017.

[10] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In CVPR, 2015.

[11] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In ICCV, 2019.

[12] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In CVPR, 2016.

[13] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In NeurIPS, 2017.

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn: Mask r-cnn. Mask r-cnn. In ICCV, 2017.

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In CVPR, 2016.

[16] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In CVPR, 2018.

[17] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML, 2015.

[18] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J. Black. Towards understanding action recognition. In ICCV, 2013.

[19] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. arXiv preprint arXiv:1705.06950, 2017.

[20] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. In CVPRW, 2017.

[21] Tae Soo Kim and Austin Reiter. Interpretable 3d human action analysis with temporal convolutional networks. In CVPRW, 2017.

[22] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In CVPR, 2019.

[23] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In ECCV, 2018.

[24] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In CVPR, 2019.

[25] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In ICCV, 2019.

[26] Jun Liu, Amir Shahroury, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In ECCV, 2016.

[27] Diogo C Luuivzon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In CVPR, 2018.

[28] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. arXiv preprint arXiv:1706.06905, 2017.

[29] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[30] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. IJCV, 2015.

[31] Amir Shahroury, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In CVPR, 2016.

[32] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Deep residual learning for image recognition. In CVPR, 2016.

[33] Min Gue Lee, Yifan Zhang, and Hanqing Lu. Deep residual learning for image recognition. In CVPR, 2016.

[34] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In CVPR, 2018.

[35] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. ICML, 2015.

[36] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J. Black. Towards understanding action recognition. In ICCV, 2013.

[37] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. arXiv preprint arXiv:1705.06950, 2017.

[38] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. In CVPRW, 2017.

[39] Tae Soo Kim and Austin Reiter. Interpretable 3d human action analysis with temporal convolutional networks. In CVPRW, 2017.

[40] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, and Qi Tian. Actional-structural graph convolutional networks for skeleton-based action recognition. In CVPR, 2019.

[41] Yingwei Li, Yi Li, and Nuno Vasconcelos. Resound: Towards action recognition without representation bias. In ECCV, 2018.

[42] Yi Li and Nuno Vasconcelos. Repair: Removing representation bias by dataset resampling. In CVPR, 2019.

[43] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In ICCV, 2019.

[44] Jun Liu, Amir Shahroury, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In ECCV, 2016.

[45] Diogo C Luvivzon, David Picard, and Hedi Tabia. 2d/3d pose estimation and action recognition using multitask deep learning. In CVPR, 2018.

[46] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. arXiv preprint arXiv:1706.06905, 2017.

[47] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

[48] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. IJCV, 2015.

[49] Amir Shahroury, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In CVPR, 2016.

[50] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In CVPR, 2019.

[51] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In NeurlIPS, 2014.

[52] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. ICLR, 2014.

[53] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In ICCV, 2015.

[54] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In CVPR, 2018.

[55] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In CVPR, 2014.
[38] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In ECCV, 2016.

[39] Wei Wang, Jinjin Zhang, Chenyang Si, and Liang Wang. Pose-based two-stream relational networks for action recognition in videos. arXiv preprint arXiv:1805.08484, 2018.

[40] Philippe Weinzaepfel and Grégory Rogez. Mimetics: Towards understanding human actions out of context. arXiv preprint arXiv:1912.07249, 2019.

[41] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019.

[42] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. ECCV, 2018.

[43] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In ECCV, 2018.

[44] An Yan, Yali Wang, Zhifeng Li, and Yu Qiao. Pa3d: Pose-action 3d machine for video recognition. In CVPR, 2019.

[45] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In AAAI, 2018.

[46] Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. In AAAI, 2016.

[47] Mohammadreza Zolfaghari, Gabriel L Oliveira, Nima Sedaghat, and Thomas Brox. Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In ICCV, 2017.
Supplementary Material of “IntegralAction: Pose-driven Feature Integration for Robust Human Action Recognition in Videos”

In this supplementary material, we present more experimental results that could not be included in the main manuscript due to the lack of space.

1. Effect of the pose stream inputs

To analyze the effects of keypoint heatmaps and PAFs as inputs of the pose stream, we compare the top-1 and top-5 accuracy from pose-only models that take 1) the keypoint heatmaps, 2) the PAFs, and 3) both in Table 9. The table shows that taking both inputs achieves the best accuracy on Kinetics50 and Mimetics. The keypoint heatmaps provide the locations of each human body keypoint, which are useful in single person cases, but do not include sufficient information for differentiating each person in multi-person cases. On the other hand, the PAFs contain relationships between the keypoints from each person, which can provide information to differentiate each person in multi-person cases. We found that most of the videos in Mimetics contain a single person, which makes the heatmap-only model perform well on the action recognition. However, many videos in Kinetics contain multiple persons, and thus additional PAFs further improve the accuracy.

2. Deeper comparison with the score averaging

Most of the previous methods \cite{6,9,39,44,45} use to simply average predicted action scores from the appearance-based and pose-based action recognition models in their testing stage. We compared their accuracy with ours in Table 4 of the main manuscript, and we provide a deeper comparison between ours and theirs in Figure 5. We report top-1 accuracy on Kinetics50 and Mimetics of score averaging with various averaging weights. As the figure shows, the score averaging method that performs best on Kinetics50 achieves slightly better accuracy than ours. However, it suffers from a noticeable performance drop on Mimetics. The proposed IntegralAction achieves highly robust performance on both Kinetics50 and Mimetics datasets.

3. Appearance-only, pose-only, vs. IntegralAction

In this experiment, we analyze top-1 accuracy of each action class using the appearance-only, pose-only, and the proposed IntegralAction on Kinetics50, Mimetics, and NTU-RGBD in Table 10, 11, and 12 respectively. In addition, we compare them with the oracle selection that chooses the best prediction between the appearance-only and the pose-only. We also visualize confusion matrices from the appearance-only, pose-only, and our IntegralAction in Fig. 6. As the tables and figures show, our IntegralAction produces robust action recognition over action classes of the three datasets, while the appearance-only and pose-only fail on Mimetics and Kinetics50, respectively. The proposed IntegralAction achieves the best average accuracy on Mimetics and NTU-RGBD. In addition, it significantly outperforms the pose-only and achieves comparable average accuracy with the appearance-only on Kinetics50.

Figures 7, 8, 9, 10, and 11 show qualitative results from the appearance-only, pose-only, and the proposed IntegralAction. Interestingly, our IntegralAction often succeeds in recognizing correct actions even when both the appearance-only and pose-only fail and so does the oracle selection, as shown in Fig. 11. We found that this happens when the appearance-only model is fooled by focusing on the contextual information such as background scene and objects, and the pose-only model suffers from the context.
ambiguity because the input pose sequence can be mapped to multiple action classes. For example, the second example of Fig. 1 shows that the appearance-only model is fooled by drinking people, and the pose-only model suffers from the context ambiguity. As the input pose sequence does not contain finger keypoints, the pose-only model predicts the input pose is about playing volleyball based on the given body keypoints. The input pose sequence may need to contain richer geometric information of the human body for better performance, for example, finger keypoints and finally, a 3D mesh of the human. Also, improving the integration part to more effectively combine the context from the appearance stream and the human motion from the pose stream should also be studied.

4. Network architecture of IntegralAction

In this section, we provide the detailed network architectures used in our paper. Table 13 shows the network architecture we used in Section 4.2 of the main manuscript, while Table 14 shows the network architecture we used in Section 4.3 and 4.4 of the main manuscript.
Table 10: The top-1 accuracy for each action comparison between appearance-only, pose-only, our IntegralAction, and the oracle selection on Kinetics50.

| classes                                      | appearance-only | pose-only | IntegralAction (ours) | oracle selection |
|----------------------------------------------|-----------------|-----------|------------------------|------------------|
| surfing water                                | 87.5            | 25.0      | 91.7                   | 87.5             |
| shooting goal (soccer)                       | 51.0            | 10.2      | 49.0                   | 51.0             |
| hitting baseball                             | 84.0            | 54.0      | 88.0                   | 88.0             |
| playing bass guitar                          | 86.0            | 44.0      | 78.0                   | 88.0             |
| reading book                                 | 66.0            | 38.0      | 60.0                   | 70.0             |
| juggling soccer ball                         | 58.0            | 66.0      | 70.0                   | 80.0             |
| dribbling basketball                         | 74.0            | 58.0      | 72.0                   | 82.0             |
| playing accordion                            | 91.8            | 75.5      | 89.5                   | 81.8             |
| catching or throwing baseball                | 39.6            | 10.4      | 41.7                   | 41.7             |
| archery                                      | 77.6            | 30.6      | 75.5                   | 79.6             |
| tying tie                                    | 86.0            | 44.0      | 86.0                   | 88.0             |
| skiing (not slalom or crosscountry)          | 95.9            | 63.3      | 95.9                   | 95.9             |
| brushing hair                                | 62.0            | 34.0      | 60.0                   | 68.0             |
| hurdling                                     | 92.0            | 66.0      | 90.0                   | 94.0             |
| playing violin                               | 76.0            | 60.0      | 74.0                   | 86.0             |
| playing volleyball                           | 79.2            | 45.8      | 77.1                   | 81.2             |
| deadlifting                                  | 87.8            | 87.8      | 91.8                   | 93.9             |
| skipping rope                                | 67.3            | 77.6      | 85.7                   | 85.7             |
| playing piano                                | 78.0            | 38.0      | 76.0                   | 80.0             |
| writing                                      | 72.0            | 22.0      | 72.0                   | 74.0             |
| climbing a rope                              | 78.0            | 82.0      | 78.0                   | 88.0             |
| dunking basketball                            | 56.2            | 41.7      | 60.4                   | 68.8             |
| playing basketball                            | 58.0            | 32.0      | 58.0                   | 66.0             |
| brushing teeth                               | 66.0            | 44.0      | 68.0                   | 72.0             |
| drinking                                     | 30.6            | 14.3      | 32.7                   | 36.7             |
| driving car                                  | 91.7            | 39.6      | 91.8                   | 91.7             |
| walking the dog                              | 93.9            | 65.3      | 91.8                   | 95.9             |
| playing saxophone                            | 80.0            | 54.0      | 78.0                   | 86.0             |
| playing trumpet                              | 83.7            | 57.1      | 83.7                   | 85.7             |
| bowling                                      | 93.9            | 38.8      | 87.8                   | 93.9             |
| punching person (boxing)                     | 79.2            | 60.4      | 75.0                   | 83.3             |
| cleaning windows                             | 76.0            | 16.0      | 76.0                   | 80.0             |
| clean and jerk                               | 91.8            | 87.8      | 91.8                   | 93.9             |
| eating cake                                  | 58.0            | 22.0      | 48.0                   | 64.0             |
| flying kite                                  | 90.0            | 54.0      | 90.0                   | 96.0             |
| opening bottle                               | 52.0            | 18.0      | 70.0                   | 58.0             |
| canoeing or kayaking                         | 94.0            | 38.0      | 90.0                   | 94.0             |
| reading newspaper                            | 54.0            | 8.0       | 38.0                   | 54.0             |
| skiing slalom                                | 86.0            | 76.0      | 86.0                   | 92.0             |
| playing guitar                               | 80.0            | 56.0      | 76.0                   | 84.0             |
| eating ice cream                             | 54.0            | 20.0      | 46.0                   | 66.0             |
| climbing ladder                              | 68.0            | 40.0      | 72.0                   | 74.0             |
| juggling balls                               | 81.6            | 79.6      | 85.7                   | 91.8             |
| shooting basketball                          | 30.6            | 16.3      | 22.4                   | 40.8             |
| catching or throwing frisbee                 | 56.0            | 8.0       | 48.0                   | 58.0             |
| sweeping floor                               | 72.0            | 42.0      | 72.0                   | 76.0             |
| playing tennis                               | 93.9            | 69.4      | 93.9                   | 98.0             |
| sword fighting                               | 38.8            | 32.7      | 40.8                   | 53.1             |
| smoking                                      | 55.1            | 36.7      | 53.1                   | 65.3             |
| golf driving                                  | 84.0            | 80.0      | 86.0                   | 86.0             |
| average                                      | 72.8            | 45.6      | 72.2                   | 78.2             |
Table 11: The top-1 accuracy for each action comparison between appearance-only, pose-only, our IntegralAction, and the oracle selection on Mimetics.

| classes                          | appearance-only | pose-only | IntegralAction (ours) | oracle selection |
|----------------------------------|-----------------|-----------|-----------------------|-----------------|
| surfing water                    | 0.0             | 0.0       | 12.5                  | 0.0             |
| shooting goal (soccer)           | 18.2            | 9.1       | 18.2                  | 27.3            |
| hitting baseball                 | 0.0             | 10.0      | 10.0                  | 10.0            |
| playing bass guitar              | 9.1             | 18.2      | 27.3                  | 27.3            |
| reading book                     | 11.1            | 11.1      | 0.0                   | 22.2            |
| juggling soccer ball             | 16.7            | 41.7      | 33.3                  | 41.7            |
| dribbling basketball             | 0.0             | 61.5      | 23.1                  | 61.5            |
| playing accordion                | 10.0            | 40.0      | 50.0                  | 50.0            |
| catching or throwing baseball    | 14.3            | 14.3      | 0.0                   | 28.6            |
| archery                           | 6.7             | 33.3      | 20.0                  | 33.3            |
| tying tie                        | 16.7            | 16.7      | 16.7                  | 16.7            |
| skiing (not slalom or crosscountry) | 12.5       | 12.5      | 12.5                  | 12.5            |
| brushing hair                    | 11.8            | 41.2      | 47.1                  | 41.2            |
| hurling                          | 11.1            | 55.6      | 22.2                  | 55.6            |
| playing violin                   | 11.8            | 17.6      | 23.5                  | 17.6            |
| playing volleyball               | 23.1            | 23.1      | 38.5                  | 38.5            |
| deadlifting                      | 11.1            | 100.0     | 100.0                 | 100.0           |
| skipping rope                    | 25.0            | 83.3      | 75.0                  | 83.3            |
| playing piano                    | 11.8            | 11.8      | 11.8                  | 17.6            |
| writing                          | 0.0             | 0.0       | 0.0                   | 0.0             |
| climbing a rope                  | 0.0             | 42.9      | 50.0                  | 42.9            |
| dunking basketball               | 0.0             | 33.3      | 33.3                  | 33.3            |
| playing basketball               | 23.1            | 23.1      | 30.8                  | 30.8            |
| brushing teeth                   | 35.7            | 35.7      | 42.9                  | 50.0            |
| drinking                         | 10.0            | 20.0      | 30.0                  | 25.0            |
| driving car                      | 0.0             | 0.0       | 0.0                   | 0.0             |
| walking the dog                  | 7.7             | 7.7       | 7.7                   | 7.7             |
| playing saxophone                | 0.0             | 7.7       | 15.4                  | 7.7             |
| playing trumpet                  | 0.0             | 50.0      | 50.0                  | 50.0            |
| bowling                          | 8.3             | 0.0       | 8.3                   | 8.3             |
| punching person (boxing)         | 9.1             | 36.4      | 45.5                  | 36.4            |
| cleaning windows                 | 6.7             | 0.0       | 0.0                   | 6.7             |
| clean and jerk                   | 15.4            | 92.3      | 92.3                  | 92.3            |
| eating cake                      | 0.0             | 11.8      | 0.0                   | 11.8            |
| flying kite                      | 10.0            | 0.0       | 20.0                  | 10.0            |
| opening bottle                   | 0.0             | 0.0       | 0.0                   | 0.0             |
| canoeing or kayaking             | 0.0             | 21.4      | 14.3                  | 21.4            |
| reading newspaper                | 0.0             | 11.1      | 0.0                   | 11.1            |
| skiing slalom                    | 0.0             | 10.0      | 10.0                  | 10.0            |
| playing guitar                   | 7.1             | 14.3      | 14.3                  | 14.3            |
| eating ice cream                 | 0.0             | 0.0       | 0.0                   | 0.0             |
| climbing ladder                  | 7.7             | 15.4      | 7.7                   | 23.1            |
| juggling balls                   | 42.9            | 57.1      | 57.1                  | 71.4            |
| shooting basketball              | 8.3             | 8.3       | 16.7                  | 16.7            |
| catching or throwing frisbee     | 50.0            | 0.0       | 40.0                  | 50.0            |
| sweeping floor                   | 0.0             | 0.0       | 0.0                   | 0.0             |
| playing tennis                   | 5.6             | 16.7      | 11.1                  | 22.2            |
| sword fighting                   | 46.7            | 66.7      | 66.7                  | 73.3            |
| smoking                          | 26.7            | 20.0      | 33.3                  | 33.3            |
| golf driving                      | 14.3            | 64.3      | 50.0                  | 64.3            |
| average                          | 11.2            | 26.0      | 26.5                  | 30.7            |
Table 12: The top-1 accuracy for each action comparison between appearance-only, pose-only, our IntegralAction, and the oracle selection on NTU-RGBD.

| classes            | appearance-only | pose-only | IntegralAction (ours) | oracle selection |
|--------------------|-----------------|-----------|-----------------------|------------------|
| drink water        | 90.9            | 82.8      | 92.0                  | 96.4             |
| eat meal           | 80.0            | 66.9      | 82.5                  | 82.9             |
| brush teeth        | 92.3            | 80.1      | 91.5                  | 95.6             |
| brush hair         | 95.2            | 87.5      | 96.7                  | 97.4             |
| drop               | 97.1            | 78.5      | 97.1                  | 98.5             |
| pick up            | 96.7            | 93.8      | 98.2                  | 98.2             |
| throw              | 90.2            | 83.3      | 90.2                  | 94.2             |
| sit down           | 95.6            | 96.0      | 98.5                  | 98.9             |
| stand up           | 98.5            | 96.7      | 99.3                  | 99.6             |
| clapping           | 79.5            | 64.8      | 78.8                  | 89.4             |
| reading            | 73.9            | 48.2      | 70.6                  | 82.0             |
| writing            | 64.3            | 40.4      | 64.7                  | 77.2             |
| tear up paper      | 93.0            | 80.8      | 93.7                  | 95.9             |
| put on jacket      | 100.0           | 97.8      | 99.6                  | 100.0            |
| take off jacket    | 98.2            | 94.6      | 96.4                  | 98.9             |
| put on a shoe      | 90.8            | 58.6      | 83.2                  | 92.7             |
| take off a shoe    | 82.8            | 55.8      | 73.5                  | 89.8             |
| put on glasses     | 85.7            | 85.3      | 92.6                  | 96.0             |
| take off glasses   | 89.8            | 86.1      | 92.3                  | 93.4             |
| put on a hat/cap   | 99.6            | 92.3      | 98.2                  | 100.0            |
| take off a hat/cap | 98.2            | 94.1      | 98.5                  | 99.3             |
| cheer up           | 94.5            | 90.9      | 93.8                  | 96.7             |
| hand waving        | 87.2            | 87.2      | 90.9                  | 93.4             |
| kicking something  | 98.2            | 93.1      | 98.2                  | 99.3             |
| reach into pocket  | 81.8            | 72.6      | 81.4                  | 85.8             |
| hopping            | 94.9            | 96.4      | 96.4                  | 96.7             |
| jump up            | 100.0           | 99.6      | 100.0                 | 100.0            |
| phone call         | 89.1            | 78.9      | 78.2                  | 94.9             |
| play with phone/tablet | 70.2        | 58.2      | 76.4                  | 81.5             |
| type on a keyboard | 88.4            | 66.5      | 89.8                  | 93.8             |
| point to something | 88.8            | 75.7      | 90.6                  | 92.4             |
| taking a selfie    | 88.0            | 85.5      | 93.1                  | 94.6             |
| check time (from watch) | 88.4        | 84.8      | 92.8                  | 97.1             |
| rub two hands      | 75.2            | 75.4      | 79.7                  | 91.7             |
| nod head/bow       | 93.8            | 90.2      | 95.7                  | 97.1             |
| shake head         | 94.9            | 83.2      | 98.2                  | 97.8             |
| wipe face          | 83.7            | 76.4      | 91.7                  | 95.7             |
| salute             | 95.7            | 90.6      | 95.7                  | 97.8             |
| put palms together | 82.2            | 88.8      | 92.4                  | 95.7             |
| cross hands in front | 96.0        | 94.9      | 97.5                  | 97.8             |
| sneeze/cough       | 68.5            | 69.6      | 77.7                  | 80.4             |
| staggering         | 98.6            | 96.7      | 98.9                  | 99.6             |
| falling down       | 98.2            | 96.0      | 98.9                  | 98.5             |
| headache           | 68.5            | 68.1      | 75.0                  | 84.1             |
| chest pain         | 88.0            | 88.0      | 92.0                  | 96.4             |
| back pain          | 95.3            | 85.5      | 97.1                  | 98.9             |
| neck pain          | 83.7            | 77.2      | 89.1                  | 92.8             |
| nausea/vomiting    | 87.6            | 83.6      | 90.5                  | 93.1             |
| fan self           | 86.5            | 86.9      | 88.4                  | 96.0             |
| punch/slap         | 88.3            | 89.1      | 92.0                  | 95.3             |
| kissing            | 98.6            | 91.7      | 97.8                  | 99.3             |
| pushing            | 97.5            | 94.9      | 98.5                  | 99.6             |
| pat on back        | 97.8            | 85.5      | 98.2                  | 99.3             |
| point finger       | 97.5            | 89.1      | 97.8                  | 98.6             |
| hugging            | 99.3            | 97.4      | 99.6                  | 100.0            |
| giving object      | 95.3            | 87.3      | 94.2                  | 97.1             |
| touch pocket       | 96.7            | 92.0      | 96.0                  | 99.3             |
| shaking hands      | 98.2            | 94.6      | 100.0                 | 99.6             |
| walking towards    | 100.0           | 98.9      | 100.0                 | 100.0            |
| walking apart      | 100.0           | 97.1      | 98.9                  | 100.0            |
| average            | 90.4            | 83.7      | 91.7                  | 95.1             |
Figure 6: Visualized confusion matrices of appearance-only, pose-only, and our IntegralAction on Kinetics50, Mimetics, and NTU-RGBD.
Figure 7: Qualitative results of appearance-only, pose-only, and the proposed IntegralAction.

Figure 8: Qualitative results of appearance-only, pose-only, and the proposed IntegralAction.
Figure 9: Qualitative results of appearance-only, pose-only, and the proposed IntegralAction.

Figure 10: Qualitative results of appearance-only, pose-only, and the proposed IntegralAction.
Figure 11: Qualitative results of appearance-only, pose-only, and the proposed IntegralAction.
Table 13: The network architecture details of IntegralAction in Section 4.2 of the main manuscript. The dimensions of kernels are denoted by \((T \times S^2, C)\) for the temporal, spatial, and channel sizes. The strides and output size are denoted by \((T \times S^2)\) for the temporal and spatial sizes.

| layers | appearance stream | pose stream | output size |
|--------|-------------------|-------------|-------------|
| input  | RGB frames        | keypoint heatmaps+PAFs | appearance: \(8 \times 224^2\) pose: \(32 \times 56^2\) |
| conv1  | \(1 \times 7^2, 64, \text{stride} 1 \times 2^2\) | \(1 \times 3^2, 64\) | appearance: \(8 \times 112^2\) pose: \(32 \times 56^2\) |
|        | \(1 \times 3^2\) max pool, \text{stride} 1 \times 2^2\) |            |             |
| res2   | \[
\begin{bmatrix}
\text{TSM} \\
1 \times 2^2, 256 \\
1 \times 3^2, 256 \\
1 \times 1^2, 256
\end{bmatrix}
\] \(\times 3\) | \[
\begin{bmatrix}
\text{TSM} \\
1 \times 3^2, 64 \\
1 \times 3^2, 64
\end{bmatrix}
\] \(\times 2\) | appearance: \(8 \times 56^2\) pose: \(32 \times 56^2\) |
| res3   | \[
\begin{bmatrix}
\text{TSM} \\
1 \times 1^2, 512 \\
1 \times 3^2, 512 \\
1 \times 1^2, 512
\end{bmatrix}
\] \(\times 4\) | \[
\begin{bmatrix}
\text{TSM} \\
1 \times 3^2, 128 \\
1 \times 3^2, 128
\end{bmatrix}
\] \(\times 2\) | appearance: \(8 \times 28^2\) pose: \(32 \times 28^2\) |
| res4   | \[
\begin{bmatrix}
\text{TSM} \\
1 \times 1^2, 1024 \\
1 \times 3^2, 1024 \\
1 \times 1^2, 1024
\end{bmatrix}
\] \(\times 6\) | \[
\begin{bmatrix}
\text{TSM} \\
1 \times 3^2, 256 \\
1 \times 3^2, 256
\end{bmatrix}
\] \(\times 2\) | appearance: \(8 \times 14^2\) pose: \(32 \times 14^2\) |
| res5   | \[
\begin{bmatrix}
\text{TSM} \\
1 \times 1^2, 2048 \\
1 \times 3^2, 2048 \\
1 \times 1^2, 2048
\end{bmatrix}
\] \(\times 3\) | \[
\begin{bmatrix}
\text{TSM} \\
1 \times 3^2, 512 \\
1 \times 3^2, 512
\end{bmatrix}
\] \(\times 2\) | appearance: \(8 \times 7^2\) pose: \(32 \times 7^2\) |
| pool   | global average pool | global average pool | appearance: \(8 \times 1^2\) pose: \(32 \times 1^2\) |
| feature align (TCB\(_A\), TCB\(_P\)) | \(1 \times 1^2, 512\) layer normalization | \(4 \times 1^2\) avg pool, \text{stride} \(4 \times 1^2\) | \(1 \times 1^2, 512\) layer normalization both: \(8 \times 1^2\) |
| pose-driven gating (CGB) | \((1 - G)\) element-wise product | \((G : 1 \times 1^2, 512)\) \(G\) element-wise product |
| aggregation | element-wise addition |
| classifier | fully-connected layer | \# of classes |
Table 14: The network architecture details of IntegralAction in Section 4.3 and 4.4 of the main manuscript. The dimensions of kernels are denoted by \((T \times S^2, C)\) for the temporal, spatial, and channel sizes. The strides and output size are denoted by \((T \times S^2)\) for the temporal and spatial sizes.

| layers | appearance stream | pose stream | output size |
|--------|-------------------|-------------|-------------|
| input  | RGB frames        | keypoint heatmaps+PAFs | appearance: 8×224² pose: 8×56² |
| conv₁  | \(1 \times 7^2, 64\), stride \(1 \times 2^2\) | \(1 \times 3^2, 64\) | appearance: 8×112² pose: 8×56² |
| res₂   | \(1 \times 3^2\) max pool, stride \(1 \times 2^2\) | \[ \begin{bmatrix} \text{TSM} \\ 1 \times 3^2, 64 \\ 1 \times 3^2, 64 \end{bmatrix} \times 2 \] | both: 8×56² |
|        | \[ \begin{bmatrix} \text{TSM} \\ 1 \times 3^2, 128 \\ 1 \times 3^2, 128 \end{bmatrix} \times 2 \] | \[ \begin{bmatrix} \text{TSM} \\ 1 \times 3^2, 128 \\ 1 \times 3^2, 128 \end{bmatrix} \times 2 \] | both: 8×28² |
| res₃   | \[ \begin{bmatrix} \text{TSM} \\ 1 \times 3^2, 256 \\ 1 \times 3^2, 256 \end{bmatrix} \times 2 \] | \[ \begin{bmatrix} \text{TSM} \\ 1 \times 3^2, 256 \\ 1 \times 3^2, 256 \end{bmatrix} \times 2 \] | both: 8×14² |
| res₄   | \[ \begin{bmatrix} \text{TSM} \\ 1 \times 3^2, 512 \\ 1 \times 3^2, 512 \end{bmatrix} \times 2 \] | \[ \begin{bmatrix} \text{TSM} \\ 1 \times 3^2, 512 \\ 1 \times 3^2, 512 \end{bmatrix} \times 2 \] | both: 8×7² |
| pool   | global average pool | global average pool | |
| feature align (TCBₐ,TCBₚ) | \(1 \times 1², 512\) | \(1 \times 1², 512\) | both: 8×1² |
| pose-driven gating (CGB) | \((1 - G)\) element-wise product | \(G \cdot 1 \times 1 \times 1, 512\) | |
| aggregation | element-wise addition | \(G\) element-wise product | |
| classifier | fully-connected layer | \# of classes | |