

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

Gyeongsik Moon^{*1}Heeseung Kwon^{*2}Kyoung Mu Lee¹Minsu Cho²¹SNU 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 take 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. The codes are available in [here](#).

1. Introduction

Human action recognition in videos aims at classifying an input video of human action into one of pre-defined target classes [4, 7, 13, 14, 24, 27, 36, 46–48]. Following the success of convolutional neural networks (CNNs) on image classification [17, 37], video action recognition has made remarkable progress by developing deep neural models that process RGB image frames via spatio-temporal convolution [4, 13, 27, 46] or two-stream convolution with scene optical flow [14, 36]. 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 [25, 26]. 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 [43]. While recent *pose-based* methods [7, 24, 47, 48], 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 real-world videos in the standard benchmark datasets, human poses are not easy to detect, only partially visible (mostly, close-up faces and hands), or completely absent (first-person view without any persons). Even with successful detection, human poses are often ambiguous without context.

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, the pose-based approach has the strong advantage of robustness in understanding actual human actions. As shown in the example of Fig. 1(b), while the appearance approach is misled by the appearance of barbells, the pose-based approach correctly predicts “hurdling”. Several methods [7, 10, 30, 42, 47, 48, 50] attempt to integrate the two approaches, but the problem has turned out to be non-trivial. They either aggregates the two streams by averaging predicted action scores [7, 10, 42, 47, 48] or fusing features from each stream by fully-connected layers [30, 50]. Their models all inherit a strong bias of contextual information present in the datasets [25, 26] and thus are easily fooled by out-of-context action videos [43].

To address the problem, we propose an effective and robust integration model, dubbed *IntegralAction*, that dynamically combines the appearance and pose information in a pose-driven manner. The main idea of pose-driven feature integration 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 con-

^{*} equal contribution

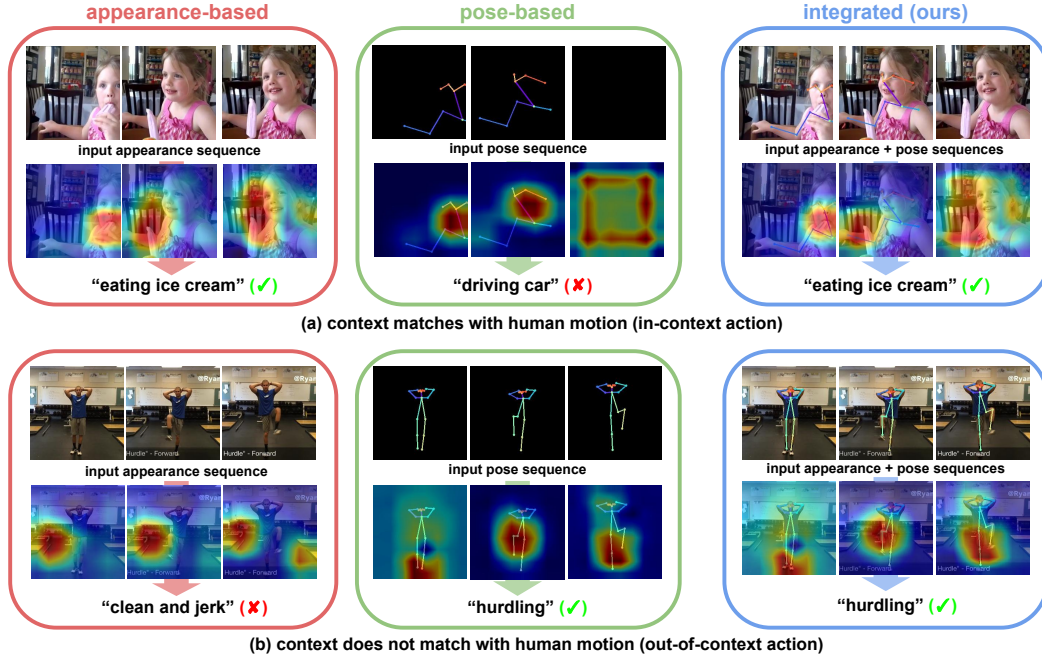


Figure 1: Comparative examples of appearance-based, pose-based, vs. our approaches. For (a) in-context action video and (b) out-of-context action video cases, representative frames and their class activation maps [49] are visualized with the action prediction results. The appearance-based and pose-based approaches are ours with only one stream without the integration. In the input pose sequence, the blank image indicates a pose estimation failure.

textual information and focus on human motion information when the pose information is sufficient for action recognition. In experimental evaluations, we demonstrate that the proposed pose-driven feature integration greatly improves action recognition on out-of-context action videos, Mimetics [43] dataset, without losing performance on in-context action videos, Kinetics [21] and NTU-RGBD [34] 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) [9], two-stream CNNs [14, 36], and 3D CNNs [4, 38, 39, 46]. Two-stream approaches [14, 36] adopt two-stream networks with RGB and optical flow streams, which process RGB frames together with corresponding optical flow frames. Carreira *et al.* [4] introduce the inflated 3D CNN (I3D) that expands ImageNet [33] pre-trained kernels of 2D CNN to 3D. They verify 3D CNNs with large-scale in-context action datasets such as Kinetics [21] become strong baselines in action recognition. Du *et al.* [39] and Xie *et al.* [46] factorize 3D convolutions into 2D and 1D convolutions for efficient prediction. Recently, several methods propose innovative architectures for efficient action recognition [12, 13, 27]. Lin *et al.* [27] propose the temporal shifting module (TSM) that enables them to

learn spatio-temporal features using 2D convolutions only. Feichtenhofer *et al.* [12, 13] propose SlowFast networks, which capture spatial semantics and motion separately by applying different spatio-temporal resolutions for two different networks. They also propose an efficient set of X3D architectures [12] by controlling scaling factors such as channel widths, layer depths, frame intervals, and spatio-temporal resolutions for action recognition.

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 [22, 28] or off-the-shelf 2D multi-person pose estimation models [3, 16]. The earlier work of Jhuang *et al.* [20] shows that human pose features can improve human action recognition in videos. Cheron *et al.* [6] construct two-stream networks that process RGB frames and optical flows of human parts. Choutas *et al.* [7] introduce a concept of pose motion that represents the temporal movement of human pose in a spatial domain. Yan *et al.* [47] propose a unified 3D CNN model that effectively encodes multiple pose modalities. Du *et al.* [10] and Rohit *et al.* [15] use estimated human poses for attention mechanisms in their model by converting human keypoints to 2-dimensional heatmaps. Recently, GraphCNN-based methods have been proposed for skeleton-based action recognition [5, 24, 29, 35, 48]. Yan *et al.* [48] construct a spatio-temporal human pose graph and process it by GraphCNN. Shi *et al.* [35] introduce a GraphCNN-based action recog-

dition system that learns graph topology. Li *et al.* [24] propose an encoder-decoder structure to capture action-specific latent high-order dependencies in their GraphCNN-based model. Cheng *et al.* [5] propose a shift graph operation for reducing FLOPs of GraphCNNs. Liu *et al.* [29] propose a novel graph convolution operator (G3D) for capturing long-range joint relationship modeling.

Integrating appearance and pose. Several methods have attempted to improve performance in action recognition by integrating the two approaches [7, 10, 15, 30, 42, 47, 48, 50]. Most of the methods [7, 10, 42, 47, 48] use to simply add predicted action scores from the two models in their testing stage. Rohit *et al.* [15] propose to use a bilinear pooling between appearance and pose features for their action classifier. Zolfaghari *et al.* [50] develop a multi-stream 3D CNN for processing multi-frame human poses, optical flows, and RGBs together. Luvizon *et al.* [30] 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 [7, 10, 42, 47, 48] is easily affected by inaccurate action prediction from the pose stream. Our dynamic integration method of IntegralAction effectively suppresses misleading contextual features from the appearance stream and also supplements inaccurate or ambiguous pose 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 [8, 18, 31, 46]. Dauphin *et al.* [8] use a simple gating block for language modeling. Hu *et al.* [18] introduce a squeeze-and-excitation block for image classification. For video classification, Miech *et al.* [31] and Xie *et al.* [46] 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.

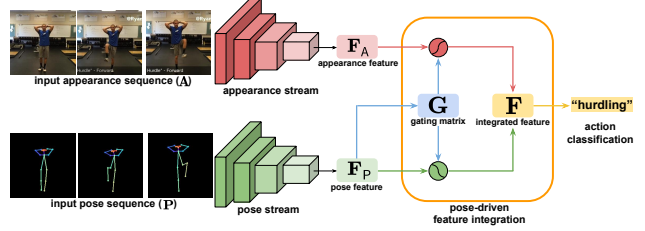


Figure 2: The overall pipeline of the proposed IntegralAction.

- 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 \mathbf{A} and the corresponding pose estimation sequence \mathbf{P} , and then predicts action probabilities of the video over target classes \mathcal{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, \mathbf{A} and \mathbf{P} , via appearance and pose streams; these two streams transform \mathbf{A} and \mathbf{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 \mathcal{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 $\mathbf{A} \in \mathbb{R}^{T \times 3 \times H_A \times W_A}$ where T , H_A , and W_A denote the numbers of frames, height, and width of the sequence, respectively. The sampled frames are selected by randomly choosing the first frame within the video clip and then collecting subsequent frames at τ frame intervals [4, 46]. The frames are resized and cropped into the size of $H_A \times W_A$. Given the sequence of RGB frames \mathbf{A} , the appearance stream transforms it into an appearance feature $\mathbf{F}_A \in \mathbb{R}^{T \times C}$.

For an architecture of the stream, we employ ResNet [17] and add the temporal shift module (TSM) [27] for each residual block of the ResNet for efficient and effective spatio-temporal feature extraction following Lin *et al.* [27]; 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 [27], 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 \mathbf{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 [2, 3, 16]. The keypoints for each frame are then translated into K keypoint heatmaps [2, 45] and B part affinity fields [3], which will be described below. Combining the heatmaps and fields along channels, we construct a sequence $\mathbf{P} \in \mathbb{R}^{T \times (K+B) \times H_P \times W_P}$ where H_P and W_P represent the height and width of the sequence, respectively. Given the sequence of human pose frames \mathbf{P} , the pose stream transforms it into a pose features $\mathbf{F}_P \in \mathbb{R}^{T \times C}$.

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 \mathbf{P} , $K + B$, is relatively large, we reduce the spatial size of \mathbf{P} for efficient feature extraction. To adapt the channel dimension of \mathbf{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 [19], and ReLU function.

Keypoint heatmap. Each keypoint is translated into a heatmap \mathbf{H}_k using a Gaussian blob [2, 45]: $\mathbf{H}_k(x, y) = \exp\left(-\frac{(x-x_k)^2 + (y-y_k)^2}{2\sigma^2}\right)$, where x_k and y_k are the coordinates of k th keypoint and σ 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 heatmap. 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 \mathbf{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: $\mathbf{L}_b(x, y) = \mathbf{v}_b$ where $\mathbf{v}_b \in \mathbb{R}^2$ is the 2D unit vector in the direction of the b th bone [3]. Like the keypoint heatmap, we sort pose estimation results by the score and select top-5 persons to make PAFs. The PAF of a person whose score is lower than 0.1 becomes zero PAF. When a position is assigned bones of multiple persons, it obtains the average

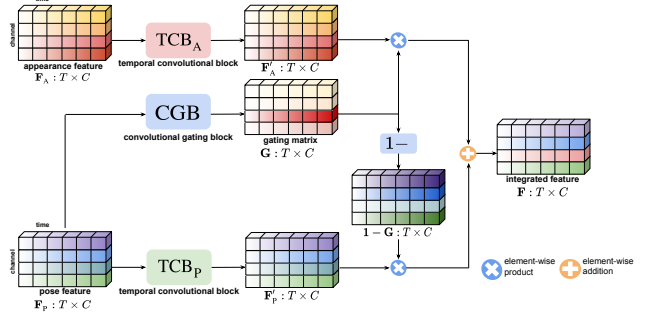


Figure 3: Pose-driven feature integration.

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, \mathbf{F}_A and \mathbf{F}_P , into an *adaptive* and *robust* action feature \mathbf{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, \mathbf{F}_A and \mathbf{F}_P , are transformed into \mathbf{F}'_A and \mathbf{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 \mathbf{F}_A and \mathbf{F}_P , respectively, and produce outputs with the same channel dimension as inputs. This semantic alignment along channels is necessary for the two features, \mathbf{F}'_A and \mathbf{F}'_P , to be integrated by addition after gating.

Pose-driven gating. The pose-driven gating takes as input the pose feature \mathbf{F}_P , predicts a channel-wise gating matrix $\mathbf{G} \in \mathbb{R}^{T \times C}$, and imposes it on semantically aligned features \mathbf{F}'_A and \mathbf{F}'_P . The gating matrix \mathbf{G} is predicted by a convolutional gating block, CGB , that consists of convolution with kernel size 1, batch normalization, and sigmoid activation function. The gating matrix \mathbf{G} , where each element lies $[0, 1]$, is used to perform gating on \mathbf{F}'_A and \mathbf{F}'_P in opposite directions: $\mathbf{G} \odot \mathbf{F}'_A$ and $(1 - \mathbf{G}) \odot \mathbf{F}'_P$, 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 $\mathbf{F} \in \mathbb{R}^{T \times C}$ as follows:

$$\mathbf{F} = \mathbf{G} \odot \mathbf{F}'_A + (1 - \mathbf{G}) \odot \mathbf{F}'_P. \quad (1)$$

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:

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

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 strong 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 \mathbf{F} to predict probabilities over target action classes. The classification is performed for each frame of \mathbf{F} by a fully-connected layer with softmax output. As in [27, 41], 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:

$$L = L_{\text{cls}} + \lambda L_{\text{gate}}, \quad (3)$$

where L_{cls} represents the standard cross entropy loss and L_{gate} is the gate regularizer described above. λ 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 \mathbf{F}_A and \mathbf{F}_P for classification, respectively, by minimizing L_{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 [32] is used for implementation. The ResNet is initialized with the publicly released weight pre-trained on the ImageNet [33], 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 224×224 , and that of the input pose sequence as 56×56 . We pre-train the appearance and pose streams with the initial learning rate

of 10^{-2} . The learning rate is reduced by a factor of 10 at 30th and 60th epoch until 70th epoch when trained on Kinetics [21] and reduced by a factor of 10 at 20th and 30th epoch until 40th epoch when trained on Kinetics50 [21] and NTU-RGBD [34]. After the pre-training, we continue the training of full IntegralAction. To this end, the initial learning rate is set to 10^{-3} and reduced by a factor of 10 at 10th and 15th epoch until 20th epoch. Random scaling, translation, and horizontal flip augmentations are applied during the training following Lin *et al.* [27]. 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 Lin *et al.* [27].

4. Experiments

4.1. Datasets and evaluation metrics

Kinetics. Kinetics [21] 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 [43], 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 to human body motions rather than background scene and objects. For human pose estimation on Kinetics, we use the results released by Yan *et al.* [48], which are obtained using OpenPose [2].

Mimetics. Mimetics [43] 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.* [43]. For human pose estimation on Mimetics, we use ResNet50-based Mask R-CNN [16, 44].

NTU-RGBD. NTU-RGBD [34] 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 [34].

Following the previous results reported on Kinetics, Mimetics [4, 43], and NTU-RGBD [24, 35, 43, 48], 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-only model and use ResNet-18 for the pose-only model. We set $T = 8$ and $\tau = 8$ for the appearance-only model, and $T = 32$ and $\tau = 2$ for the pose-only model. The same settings are used for appearance and pose streams of our integrated model. We apply $4 \times 1 \times 1$ average pool-

Table 1: Top-1 and top-5 accuracy comparison between baselines and the proposed method on Kinetics, Mimetics, and NTU-RGBD.

methods	Kinetics		Mimetics		NTU-RGBD		avg rank
	top-1	top-5	top-1	top-5	top-1	top-5	
appearance-only model	73.5	91.0	6.3	16.7	90.4	99.0	2.7
pose-only model	30.2	51.3	15.2	33.1	83.7	97.0	3.2
IntegralAction (ours, $\lambda = 1.5$)	73.3	90.8	12.8	26.0	91.7	99.4	1.8
IntegralAction (ours, $\lambda = 5.0$)	65.0	85.9	15.3	31.5	91.0	99.2	2.2

Table 2: Top-1 and top-5 accuracy comparison with state-of-the-art methods on Kinetics and Mimetics. Methods with * initialize weights with ImageNet pre-trained ones.

methods	Kinetics		Mimetics	
	top-1	top-5	top-1	top-5
Appearance-based methods				
I3D (RGB)* [4]	71.1	89.3	-	-
R(2+1)D (RGB)* [39]	72.0	90.0	-	-
TSM* [27]	73.5	91.0	6.3	16.7
3D ResNext-101 (RGB)* [43]	74.5	-	8.6	20.1
3D ResNext-101 (two-stream)* [43]	-	-	10.5	26.9
SlowFast Res-101 [13]	78.9	93.5	-	-
X3D-XL [12]	79.1	93.9	-	-
Pose-based methods				
Deep LSTM [34]	16.4	35.3	-	-
TCN [23]	20.3	40.0	-	-
ST-GCN [48]	30.7	52.8	12.6	27.4
SIP-Net [43]	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

ing along the temporal axis for \mathbf{F}_P to set the same feature dimension as that of \mathbf{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 λ 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 the 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, 27, 39, 43],

Table 3: Top-1 accuracy comparison with state-of-the-art methods on NTU-RGBD.

methods	accuracy
Lie Group [40]	50.1
Du <i>et al.</i> [11]	59.1
Deep LSTM [34]	60.7
SIP-Net [43]	64.8
Zolfaghari <i>et al.</i> [50]	67.8
TCN [23]	74.3
ST-GCN [48]	81.5
AS-GCN [24]	86.8
2S-AGCN [35]	88.5
Shift-GCN [5]	90.7
MS-G3DNet [29]	91.5
IntegralAction (ours, $\lambda = 1.5$)	91.7
IntegralAction (ours, $\lambda = 5.0$)	91.0

pose-based methods [23, 34, 43, 48], and the proposed method, respectively. 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, 27, 39, 43], and outperforms all the other pose-based methods [23, 34, 43, 48] by a large margin in Kinetics. At the same time, our method achieves state-of-the-art performance in Mimetics. Table 3 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 [40], RNN methods [11, 34], CNN methods [23, 43, 50], and GraphCNN methods [24, 35, 48]. 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 various feature integration methods including feature fuse [30, 50] and score average in testing stage [7, 10, 42, 47, 48] in Table 4. We additionally train variants of our IntegralAction by disabling the gating or gating without L_{gate} . Among all the integration methods, our setting 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 methods and ours on Mimetics is 23%, while that on Kinetics50 is 3%. For all models, the network architectures are based on that of

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 fuse [30, 50]	73.8	93.4	19.5	42.3
score average [7, 47]	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	92.3	26.5	50.5
	(↓ 3%)	(↓ 1%)	(↑ 23%)	(↑ 6%)

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 fuse. The feature fuse method [30, 50] suffers from low accuracy on Mimetics because their networks are trained without any gating or regularization; therefore, theirs cannot filter out a strong contextual 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. To further validate this, we show the Gaussian distribution of the gating matrix \mathbf{G} averaged over the channel and temporal dimensions from models trained with and without L_{gate} on each frame of Kinetics50 and Mimetics in Figure 4. As the figure 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. Thus, our IntegralAction is optimized to utilize the pose feature when the input pose sequence provides sufficient information for the action recognition without being dominated by the appearance feature.

To implement the feature fuse, we pass \mathbf{F}_P and \mathbf{F}_A to an additional fully-connected layer, of which output is used for the action prediction.

Comparison with the score average. The score averaging methods [7, 10, 42, 47, 48] average the predicted action probability from the appearance-only and pose-only models in the 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 distributions of the averaged gating matrix \mathbf{G} from Kinetics50 and Mimetics have very different variances, which indicates that the estimated gating matrix

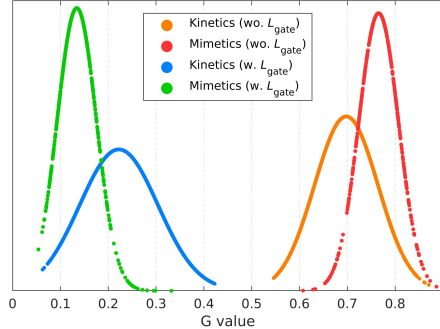


Figure 4: The Gaussian distributions of averaged \mathbf{G} from models trained with and without L_{gate} on Kinetics50 and Mimetics.

Table 5: Averaged \mathbf{G} values for each action class on Kinetics50.

action classes	averaged \mathbf{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

varies according to the input sequence. As some videos of Kinetics50 contain human-central videos while others contain context-central videos, whether the input pose sequence is sufficient for the action recognition or not varies a lot, which makes the gating matrix diverse. In contrast, most of the videos of Mimetics contain human-central videos, which results in a less diverse gating matrix. We additionally show the gating matrix \mathbf{G} averaged over the channel dimension and all frames of an action class in Table 5. The table shows that the averaged gating matrix becomes 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 increases 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. We tried to utilize the action recognition test results on the training set to supervise the gating matrix differently for each frame; however, it did not improve the action recognition accuracy in a meaningful margin. We leave designing a frame (or video)-adaptive regularizer as future work.

Table 6: Top-1 and top-5 accuracy comparison between models that estimate \mathbf{G} from appearance feature, pose feature, and both features on Kinetics50 and Mimetics. The numbers in parentheses represent relative difference between other best performing method and ours.

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 and top-5 accuracy comparison between models trained with various λ values on Kinetics50 and Mimetics.

λ 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

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 between models that estimate the gating matrix \mathbf{G} from the appearance feature \mathbf{F}_A , pose feature \mathbf{F}_P , and both ones in Table 6. The both-driven method predicts \mathbf{G} from a combined feature, concatenation of \mathbf{F}_A and \mathbf{F}_P 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 the worse performance of the appearance-driven and both-driven ones on Mimetics is that the appearance feature provides strongly biased contextual information from out-of-context action videos. Therefore, CGB, which estimates the gating matrix, is dominated by biased contextual information. In contrast, the pose feature only provides human motion information without contextual information. This can make CGB robust to the biased context, which leads to more robust performance on out-of-context action videos.

Various training settings. We provide the top-1 and top-5 accuracy from various λ values in Table 7. As the table shows, a larger λ 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

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
X	X	72.2	93.6	17.8	43.1
✓	X	72.5	92.0	23.5	46.6
✓	✓	72.2	92.3	26.5	50.5

on Mimetics while marginally decreases the accuracy on Kinetics50; therefore, we set $\lambda = 1.5$ through the whole experiments.

In addition, we show the top-1 and top-5 accuracy from various possible training strategies of IntegralAction in Table 8. For the comparison, we trained three models, including ours. The first one is trained without pre-training, and the second one is trained from pre-trained appearance and pose streams, while the two 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. Therefore, we trained our IntegralAction from pre-trained appearance and pose streams and fixed them during the final training stage.

We think 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. 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 NRF grant (NRF-2017R1E1A1A01077999 -50%), the Visual Turing Test project (IITP-2017-0-01780 -50%), and the IITP grant (No.2019-0-01906, AI Graduate School Program - POSTECH) 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] Ke Cheng, Yifan Zhang, Xiangyu He, Weihan Chen, Jian Cheng, and Hanqing Lu. Skeleton-based action recognition with shift graph convolutional network. In *CVPR*, 2020.
- [6] Guilhem Chéron, Ivan Laptev, and Cordelia Schmid. P-CNN: Pose-based cnn features for action recognition. In *ICCV*, 2015.
- [7] Vasileios Choutas, Philippe Weinzaepfel, Jérôme Revaud, and Cordelia Schmid. PoTion: Pose motion representation for action recognition. In *CVPR*, 2018.
- [8] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *ICML*, 2017.
- [9] 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.
- [10] Wenbin Du, Yali Wang, and Yu Qiao. RPA: An end-to-end recurrent pose-attention network for action recognition in videos. In *ICCV*, 2017.
- [11] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *CVPR*, 2015.
- [12] Christoph Feichtenhofer. X3D: Expanding architectures for efficient video recognition. In *CVPR*, 2020.
- [13] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *ICCV*, 2019.
- [14] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *CVPR*, 2016.
- [15] Rohit Girdhar and Deva Ramanan. Attentional pooling for action recognition. In *NeurIPS*, 2017.
- [16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *ICCV*, 2017.
- [17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.
- [18] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.
- [19] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ICML*, 2015.
- [20] Hueihan Jhuang, Juergen Gall, Silvia Zuffi, Cordelia Schmid, and Michael J Black. Towards understanding action recognition. In *ICCV*, 2013.
- [21] 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.
- [22] Leonid Keselman, John Iselin Woodfill, Anders Grunnet-Jepsen, and Achintya Bhowmik. Intel realsense stereoscopic depth cameras. In *CVPRW*, 2017.
- [23] Tae Soo Kim and Austin Reiter. Interpretable 3D human action analysis with temporal convolutional networks. In *CVPRW*, 2017.
- [24] 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.
- [25] Yingwei Li, Yi Li, and Nuno Vasconcelos. RESOUND: Towards action recognition without representation bias. In *ECCV*, 2018.
- [26] Yi Li and Nuno Vasconcelos. REPAIR: Removing representation bias by dataset resampling. In *CVPR*, 2019.
- [27] Ji Lin, Chuang Gan, and Song Han. TSM: Temporal shift module for efficient video understanding. In *ICCV*, 2019.
- [28] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3D human action recognition. In *ECCV*, 2016.
- [29] Ziyu Liu, Hongwen Zhang, Zhenghao Chen, Zhiyong Wang, and Wanli Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *CVPR*, 2020.
- [30] Diogo C Luvizon, David Picard, and Hedi Tabia. 2D/3D pose estimation and action recognition using multitask deep learning. In *CVPR*, 2018.
- [31] Antoine Miech, Ivan Laptev, and Josef Sivic. Learnable pooling with context gating for video classification. *arXiv preprint arXiv:1706.06905*, 2017.
- [32] 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.
- [33] 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.
- [34] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. NTU RGB+D: A large scale dataset for 3D human activity analysis. In *CVPR*, 2016.
- [35] Lei Shi, Yifan Zhang, Jian Cheng, and Hanqing Lu. Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In *CVPR*, 2019.
- [36] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NeurIPS*, 2014.
- [37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2014.

- [38] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3D convolutional networks. In *ICCV*, 2015.
- [39] 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.
- [40] Raviteja Vemulapalli, Felipe Arrate, and Rama Chellappa. Human action recognition by representing 3D skeletons as points in a lie group. In *CVPR*, 2014.
- [41] 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.
- [42] 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.
- [43] Philippe Weinzaepfel and Grégory Rogez. Mimetics: Towards understanding human actions out of context. *arXiv preprint arXiv:1912.07249*, 2019.
- [44] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. <https://github.com/facebookresearch/detectron2>, 2019.
- [45] Bin Xiao, Haiping Wu, and Yichen Wei. Simple baselines for human pose estimation and tracking. *ECCV*, 2018.
- [46] 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.
- [47] An Yan, Yali Wang, Zhifeng Li, and Yu Qiao. PA3D: Pose-action 3D machine for video recognition. In *CVPR*, 2019.
- [48] Sijie Yan, Yuanjun Xiong, and Dahua Lin. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *AAAI*, 2018.
- [49] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.
- [50] 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.