International Journal of Computer Vision (2021) 129:2264–2287
https://doi.org/10.1007/s11263-021-01467-7

Synthetic Humans for Action Recognition from Unseen Viewpoints

Gül Varol[1] · Ivan Laptev[2] · Cordelia Schmid[2] · Andrew Zisserman[3]

Received: 14 September 2020 / Accepted: 5 April 2021 / Published online: 12 May 2021

Abstract
Although synthetic training data has been shown to be beneficial for tasks such as human pose estimation, its use for RGB human action recognition is relatively unexplored. Our goal in this work is to answer the question *whether synthetic humans can improve the performance of human action recognition*, with a particular focus on generalization to unseen viewpoints. We make use of the recent advances in monocular 3D human body reconstruction from real action sequences to automatically render synthetic training videos for the action labels. We make the following contributions: (1) we investigate the extent of variations and augmentations that are beneficial to improving performance at new viewpoints. We consider changes in body shape and clothing for individuals, as well as more action relevant augmentations such as non-uniform frame sampling, and interpolating between the motion of individuals performing the same action; (2) We introduce a new data generation methodology, *SURREACT*, that allows training of spatio-temporal CNNs for action classification; (3) We substantially improve the state-of-the-art action recognition performance on the NTU RGB+D and UESTC standard human action multi-view benchmarks; Finally, (4) we extend the augmentation approach to in-the-wild videos from a subset of the Kinetics dataset to investigate the case when only one-shot training data is available, and demonstrate improvements in this case as well.

Keywords Synthetic humans · Action recognition

1 Introduction

Learning human action representations from RGB video data has been widely studied. Recent advances on convolutional neural networks (CNNs) (LeCun et al. 1989) have shown excellent performance (Carreira and Zisserman 2017; Feichtenhofer et al. 2019, 2016; Hara et al. 2018; Lin et al. 2019; Varol et al. 2018; Wang et al. 2016) on benchmark datasets, such as UCF101 (Soomro et al. 2012). However, the success of CNNs rely heavily on the availability of large-scale training data, which is not always the case. To address the lack of training data, several works explore the use of complementary synthetic data for a range of tasks in computer vision such as optical flow estimation, segmentation, human body and hand pose estimation (Dosovitskiy et al. 2015; Shotton et al. 2011; Su et al. 2015; Varol et al. 2017; Zimmermann and Brox 2017). In this work, we raise the question *how to synthesize videos for action recognition* in the case of limited real data, such as only one viewpoint, or one-shot available at training.

Imagine a surveillance or ambient assisted living system, where a dataset is already collected for a set of actions from a certain camera. Placing a new camera in the environment from a new viewpoint would require re-annotating data because the appearance of an action is drastically different when performed from different viewpoints (Junejo et al. 2011; Liu et al. 2011; Zheng et al. 2016). In fact, we observe that state-of-the-art action recognition networks fail drastically when trained and tested on distinct viewpoints. Specifically, we train the model of Hara et al. (2018) on videos from a benchmark dataset NTU RGB+D (Shahroudy et al. 2016) where people are facing the camera. When we test this network on other front-view (0°) videos, we obtain ~80% accuray. When we test with side-view (90°) videos, the performance drops to ~40% (see Sect. 4). This

Communicated by Gregory Rogez.

✉ Gül Varol
gul.varol@enpc.fr

[1] LIGM, École des Ponts, Univ Gustave Eiffel, CNRS, Champs-sur-Marne, France

[2] Inria, Paris, France

[3] Visual Geometry Group, University of Oxford, Oxford, UK

motivates us to study action recognition from novel viewpoints.

Existing methods addressing cross-view action recognition do not work in challenging setups (e.g. same subjects and similar viewpoints in training and test splits (Shahroudy et al. 2016)). We introduce and study a more challenging protocol with only one viewpoint at training. Recent methods assuming multi-view training data (Li et al. 2018a, b; Wang et al. 2018) also become inapplicable.

A naive way to achieve generalization is to collect data from all views, for all possible conditions, but this is impractical due to combinatorial explosion (Yuille et al. 2018. Instead, we augment the existing real data synthetically to increase the diversity in terms of viewpoints, appearance, and motions. Synthetic humans are relatively easy to render for tasks such as pose estimation, because arbitrary motion capture (MoCap) resource can be used (Shotton et al. 2011; Varol et al. 2017). However, action classification requires certain motion patterns and semantics. It is challenging to generate synthetic data with action labels (De Souza et al. 2017). Typical MoCap datasets (CMU Mocap Database), targeted for pose diversity, are not suitable for action recognition due to lack of clean action annotations. Even if one collects a MoCap dataset, it is still limited to pre-defined set of categories.

In this work, we propose a new, efficient and scalable approach for generating *synthetic videos with action labels* from the target set of categories. We employ a 3D human motion estimation method, such as HMMR (Kanazawa et al. 2019) and VIBE (Kocabas et al. 2020), that automatically extracts the 3D human dynamics from a single-view RGB video. The resulting sequence of SMPL body (Loper et al. 2015) pose parameters are then combined with other randomized generation components (e.g. viewpoint, clothing) to render diverse complementary training data with action annotations. Figure 1 presents an overview of our pipeline. We demonstrate the advantages of such data when training spatio-temporal CNN models for (1) action recognition from unseen viewpoints and (2) training with one-shot real data. We boost performance on unseen viewpoints from 53.6 to 69.0% on NTU, and from 49.4 to 66.4% on UESTC dataset by augmenting limited real training data with our proposed SURREACT dataset. Furthermore, we present an in-depth analysis about the importance of action relevant augmentations such as diversity of motions and viewpoints, as well as our non-uniform frame sampling strategy which substantially improves the action recognition performance. Our code and data will be available at the project page[1].

2 Related Work

Human action recognition is a well-established research field. For a broad review of the literature on action recognition, see the recent survey of Kong et al. Kong and Fu (2018). Here, we focus on relevant works on synthetic data, cross-view action recognition, and briefly on 3D human shape estimation.

Synthetic Humans. Simulating human motion dates back to 1980s. Badler et al. (1993) provide an extensive overview of early approaches. More recently, synthetic images of people have been used to train visual models for 2D/3D body pose and shape estimation (Chen et al. 2016; Ghezelghieh et al. 2016; Liu et al. 2019a; Pishchulin et al. 2012; Shotton et al. 2011; Varol et al. 2018), part segmentation (Shotton et al. 2011; Varol et al. 2017), depth estimation (Varol et al. 2017), multi-person pose estimation (Hoffmann et al. 2019), pedestrian detection (Marin et al. 2010; Pishchulin et al. 2012), person re-identification (Qian et al. 2018), hand pose estimation (Hasson et al. 2019; Zimmermann and Brox 2017), and face recognition (Kortylewski et al. 2018; Masi et al. 2019). Synthetic datasets built for these tasks, such as the recent SURREAL dataset (Varol et al. 2017), however, do not provide action labels.

Among previous works that focus on synthetic human data, very few tackle action recognition (De Souza et al. 2017; Liu et al. 2019b; Rahmani and Mian 2016). Synthetic 2D human pose sequences (Lv and Nevatia 2007) and synthetic point trajectories (Rahmani and Mian 2015; Rahmani et al. 2018; Jingtian et al. 2018) have been used for view-invariant action recognition. However, RGB-based synthetic training for action recognition is relatively new, with (De Souza et al. 2017) being one of the first attempts. De Souza et al. (2017) manually define 35 action classes and jointly estimate real categories and synthetic categories in a multi-task setting. However, their categories are not easily scalable and do not necessarily relate to the target set of classes. Unlike (De Souza et al. 2017), we automatically extract motion sequences from real data, making the method flexible for new categories. Recently, (Puig et al. 2018) has generated the VirtualHome dataset, a simulation environment with programmatically defined synthetic activities using crowd-sourcing. Different than our work, the focus of Puig et al. (2018) is not generalization to real data.

Most relevant to ours, (Liu et al. 2019b) generates synthetic training images to achieve better performance on unseen viewpoints. The work of Liu et al. (Liu et al. 2019b) is an extension of Rahmani and Mian (2016) by using RGB-D as input instead of depth only. Both works formulate a frame-based pose classification problem on their synthetic data, which they then use as features for action recognition. These features are not necessarily discriminative for the target action categories. Different than this direction, we

[1] https://www.di.ens.fr/willow/research/surreact/.

explicitly assign an action label to synthetic videos and define the supervision directly on action classification.

Cross-View Action Recognition. Due to the difficulty of building multi-view action recognition datasets, the standard benchmarks have been recorded in controlled environments. RGB-D datasets such as IXMAS (Weinland et al. 2007), UWA3D II (Rahmani et al. 2016) and N-UCLA (Wang et al. 2014) were state of the art until the availability of the large-scale NTU RGB+D dataset (Shahroudy et al. 2016). The size of NTU allows training deep neural networks unlike previous datasets. Very recently, Ji et al. (Ji et al. 2018) collected the first large-scale dataset, UESTC, that has a 360° coverage around the performer, although still in a lab setting.

Since multi-view action datasets are typically captured with depth sensing devices, such as Kinect, they also provide an accurate estimate of the 3D skeleton. Skeleton-based cross-view action recognition therefore received a lot of attention in the past decade (Ke et al. 2017; Liu et al. 2016, 2017a, b; Zhang et al. 2017). Variants of LSTMs (Hochreiter and Schmidhuber 1997) have been widely used (Liu et al. 2016, 2017a; Shahroudy et al. 2016). Recently, spatio-temporal skeletons were represented as images (Ke et al. 2017) or higher dimensional objects (Liu et al. 2017b)where standard CNN architectures were applied.

RGB-based cross-view action recognition is in comparison less studied. Transforming RGB features to be view-invariant is not as trivial as transforming 3D skeletons. Early work on transferring appearance features from the source view to the target view explored the use of maximum margin clustering to build a joint codebook for temporally synchronous videos Farhadi and Tabrizi 2008. Following this approach, several other works focused on building global codebooks to extract view-invariant representations (Kong et al. 2017; Liu et al. 2019c; Rahmani et al. 2018; Zheng and Jiang 2013; Zheng et al. 2016). Recently, end-to-end approaches used human pose information as guidance for action recognition (Baradel et al. 2017; Liu and Yuan 2018; Luvizon et al. 2018; Zolfaghari et al. 2017). Li et al. (2018a) formulated an adversarial view-classifier to achieve view-invariance. Wang et al. (Wang et al. 2018) proposed to fuse view-specific features from a multi-branch CNN. Such approaches cannot handle single-view training (Li et al. 2018a; Wang et al. 2018). Our method differs from these works by compensating for the lack of view diversity with synthetic videos. We augment the real data automatically at training time, and our model does not involve any extra cost at test time unlike (Wang et al. 2018). Moreover, we do not assume real multi-view videos at training.

3D Human Shape Estimation. Recovering the full human body mesh from a single image has been explored as a model-fitting problem (Bogo et al. 2016; Lassner et al. 2017), as regressing model parameters with CNNs (Kanazawa et al. 2018; Omran et al. 2018; Pavlakos et al. 2018; Tung et al. 2017), and as regressing non-parametric representations such as graphs or volumes (Kolotouros et al. 2019; Varol et al. 2018). Recently, CNN-based parameter regression approaches have been extended to video (Kanazawa et al. 2019; Liu et al. 2019a; Kocabas et al. 2020). HMMR (Kanazawa et al. 2019) builds on the single-image-based HMR (Kanazawa et al. 2018) to learn the human dynamics by using 1D temporal convolutions. More recently, VIBE (Kocabas et al. 2020) adopts a recurrent model based on frame-level pose estimates provided by SPIN (Kolotouros et al. 2019). VIBE also incorporates an adversarial loss that penalizes the estimated pose sequence if it is not a ‘realistic’ motion, i.e., indistinguishable from the real AMASS (Mahmood et al. 2019) MoCap sequences. In this work, we recover 3D body parameters from real videos using HMMR (Kanazawa et al. 2019) and VIBE (Kocabas et al. 2020). Both methods employ the SMPL body model (Loper et al. 2015). We provide a comparison between the two methods for our purpose of action recognition, which can serve as a proxy task to evaluate motion estimation.

3 Synthetic Humans with Action Labels

Our goal is to improve the performance of action recognition using synthetic data in cases where the real data is limited, e.g. domain mismatch between training/test such as viewpoints or low-data regime. In the following, we describe the three stages of: (1) obtaining 3D temporal models for human actions from real training sequences (at a particular viewpoint) (Sect. 3.1); (2) using these 3D temporal models to generate training sequences for new (and the original) viewpoints using a rendering pipeline with augmentation (Sect. 3.2); and (3) training a spatio-temporal CNN with both real and synthetic data (Sect. 3.3).

3.1 3D Human Motion Estimation

In order to generate a synthetic video with graphics techniques, we need to have a sequence of articulated 3D human body models. We employ the parametric body model SMPL (Loper et al. 2015), which is a statistical model, learned over thousands of 3D scans. SMPL generates the mesh of a person given the disentangled pose and shape parameters. The pose parameters ($\mathbb{R}^{72}$) control the kinematic deformations due to skeletal posture, while the shape parameters ($\mathbb{R}^{10}$) control identity-specific deformations such as the person height.

We hypothesize that a human action can be captured by the sequence of *pose* parameters, and that the *shape* parameters are largely irrelevant (note, this may not necessarily be true for human-object interaction categories). Given reliable 3D pose sequences from action recognition video datasets, we can transfer the associated action labels to synthetic videos. We use the recent method of Kanazawa et al. (Kanazawa et al. 2019), namely human mesh and motion recovery (HMMR), unless stated otherwise. HMMR extends the single-image reconstruction method HMR (Kanazawa et al. 2018) to video with a multi-frame CNN that takes into account a temporal neighborhood around a video frame. HMMR learns a temporal representation for human dynamics by incorporating large-scale 2D pseudo-ground truth poses for in-the-wild videos. It uses PoseFlow (Zhang et al. 2018)and AlphaPose (Fang et al. 2017) for multi-person 2D pose estimation and tracking as a pre-processing step. Each person crop is then given as input to the CNN for estimating the pose and shape, as well as the weak-perspective camera parameters. We refer the reader to (Kanazawa et al. 2019)for more details. We choose this method for the robustness on in-the-wild videos, ability to capture multiple people, and the smoothness of the recovered motion, which are important for our generalization from synthetic videos to real. Figure 1 presents the 3D pose animated synthetically for sample video frames. We also experiment with the more recent motion estimation method, VIBE (Kocabas et al. 2020), and show that improvements in motion estimation proportionally affect the action recognition performance in our pipeline. Note that we only use the pose parameters from HMMR or VIBE, and randomly change the shape parameters, camera parameters, and other factors. Next, we present the augmentations in our synthetic data generation.

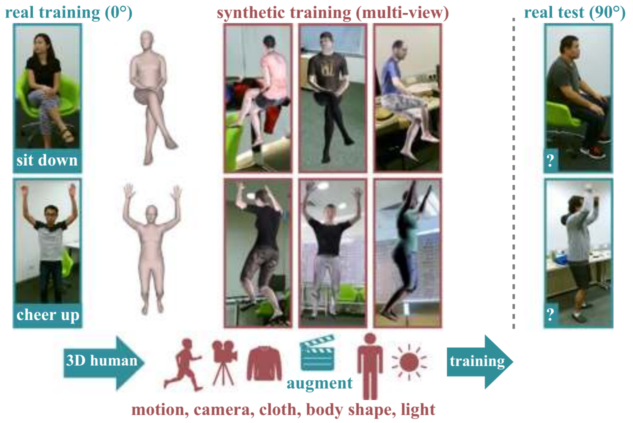

Fig. 1 Synthetic humans for actions: We estimate 3D shape from real videos and automatically render synthetic videos with action labels. We explore various augmentations for motions, viewpoints, and appearance. Training temporal CNNs with this data significantly improves the action recognition from unseen viewpoints

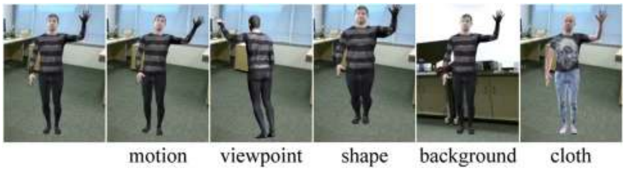

Fig. 2 Augmentations: We illustrate different augmentations of the SURREACT dataset for the hand waving action. We modify the joint angles with additive noise on the pose parameters for *motion* augmentation. We systematically change the camera position to create *viewpoint* diversity. We sample from a large set of body *shape* parameters, *backgrounds*, and *clothing* to randomize appearances

3.2 SURREACT Dataset Components

In this section, we give details on our synthetic dataset, SURREACT (Synthetic hUmans foR REal ACTions).

We follow (Varol et al. 2017) and render 3D SMPL sequences with randomized cloth textures, lighting, and body shapes. We animate the body model with our automatically extracted pose dynamics as described in the previous section. We explore various *motion augmentation* techniques to increase intra-class diversity in our training videos. We incorporate *multi-person* videos which are especially important for two-people interaction categories. We also systematically sample from 8 *viewpoints* around a circle to perform controlled experiments. Different augmentations are illustrated in Fig. 2 for a sample synthetic frame. Visualizations from SURREACT are further provided in Fig. 3.

Each generated video has automatic ground truth for 3D joint locations, part segmentation, optical flow, and SMPL body (Loper et al. 2015) parameters, as well as an action label, which we use for training a video-based 3D CNN for action classification. We use other ground truth modalities as input to action recognition as oracle experiments (see Table 14).We further use the optical flow ground truth to train a flow estimator and use the segmentation to randomly augment the background pixels in some experiments.

Our new SURREACT dataset differs from the SURREAL dataset (Varol et al. 2017) mainly by providing action labels, exploring motion augmentation, and by using automatically extracted motion sequences instead of MoCap recordings (CMU Mocap Database). Moreover, Varol et al. (Varol et al. 2017) do not exploit the temporal aspect of their dataset, but only train CNNs with single-image input. We further employ multi-person videos and a systematic viewpoint distribution.

Motion Augmentation. Automatic extraction of 3D sequences from 2D videos poses an additional challenge in our dataset compared to clean high-quality MoCap sequences. To reduce the jitter, we temporally smooth the estimated SMPL pose parameters by weighted linear averaging. SMPL poses are represented as axis-angle rotations between joints. We convert them into quaternions when we apply linear operations, then normalize each quaternion to have a unit norm, before converting back to axis-angles. Even with this processing, the motions may remain noisy, which is inevitable given that monocular 3D motion estimation is a difficult task on its own.

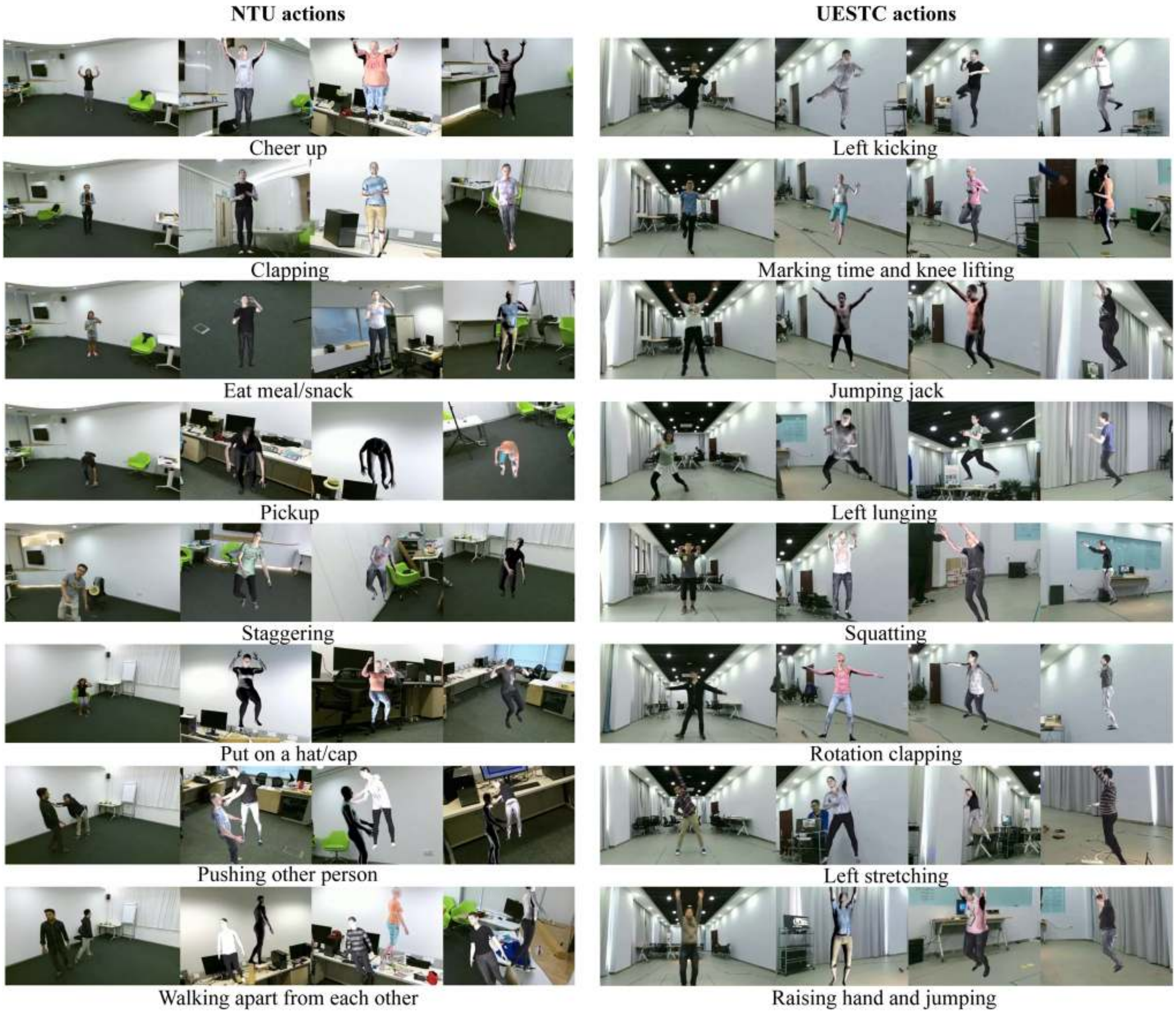

Fig. 3 SURREACT: We visualize samples from SURREACT for the actions from the NTU (left) and the UESTC (right) datasets. The motions are estimated using HMMR. Each real video frame is accompanied with three synthetic augmentations. On the left, we show the variations in clothes, body shapes, backgrounds, camera height/distance from the original 0° viewpoint. On the right, we show the variations in viewpoints for 0°, 45°, and 90° views. The complete list of actions can be found as a video at the project page (SURREACT project page)

Our findings interestingly suggest that the synthetic human videos are still beneficial when the motions are noisy.

To increase motion diversity, we further perturb the pose parameters with various augmentations. Specifically, we use a video-level *additive noise* on the quaternions for each body joint to slightly change the poses, as an intra-individual augmentation. We also experiment with an inter-individual augmentation by interpolating between motion sequences of the same action class. Given a pair of sequences from two individuals, we first align them with dynamic time warping (Sakoe and Chiba 1978), then we linearly interpolate the quaternions of the time-aligned sequences to generate a new sequence, which we refer as *interpolation*. A visual explanation of the process can be found in We show significant gains by increasing motion diversity.

Multi-person. We use the 2D pose information from (Fang et al. 2017; Zhang et al. 2018) to count the number of people in the real video. In the case of a single-person, we center the person on the image and do not add 3D translation to the body, i.e., the person is centered independently for each frame. While such constant global positioning of the body loses information for some actions such as *walking* and *jumping*, we find that the translation estimate adds more noise to consider this information and potentially increases the domain gap with the real where no such noise exists (see Appendix A). If there is more than one person, we insert additional body model(s) for rendering. We translate each person in the xy image plane. Note that we do not translate the person in full xyz space. We observe that the z component of the translation estimation is not reliable due to the depth ambiguity therefore the people are always centered at $z = 0$. More explanations about the reason for omitting the z component can be found in Appendix A.We temporally smooth the translations to reduce the noise. We subtract the mean of translations across the video and across the people to roughly center all people to the frame. We therefore keep the relative distances between people, which is important for actions such as *walking towards each other*.

Viewpoints. We systematically render each motion sequence 8 times by randomizing all other generation parameters at each view. In particular, we place the camera to be rotated at $\{0°, 45°, 90°, 135°, 180°, 225°, 270°, 315°\}$ azimuth angles with respect to the origin, denoted as (0°:45°:360°) in our experiments. The distance of the camera from the origin and the height of the camera from the ground are randomly sampled from a predefined range: [4, 6] meters for the distance, [−1, 3] meters for the height. This can be adjusted according to the target test setting.

Backgrounds. Since we have access to the target real dataset where we run pose estimation methods, we can extract background pixels directly from the training set of this dataset. We crop from regions without the person to obtain static backgrounds for the NTU and UESTC datasets. We experimentally show the benefits of using the target dataset backgrounds in the Appendix (see Table 15).For Kinetics experiments, we render human bodies on top of unconstrained videos from non-overlapping action classes and show benefits over static backgrounds. Note that these background videos might also include human pixels.

3.3 Training 3D CNNs with Non-Uniform Frames

Following the success of 3D CNNs for video recognition (Carreira and Zisserman 2017; Hara et al. 2018; Tran et al. 2015), we employ a spatio-temporal convolutional architecture that operates on multi-frame video inputs. Unless otherwise specified, our network architecture is 3D ResNet-50 (Hara et al. 2018) and its weights are randomly initialized (see Appendix B.4 for pretraining experiments).

To study the generalization capability of synthetic data across different input modalities, we train one CNN for RGB and another for optical flow as in Simonyan and Zisserman (2014). We average the scores with equal weights when reporting the fusion.

We subsample fixed-sized inputs from videos to have a $16 \times 256 \times 256$ spatio-temporal resolution, in terms of number of frames, width, and height, respectively. In case of optical flow input, we map the RGB input to $15 \times 64 \times 64$ dimensional flow estimates. To estimate flow, we train a two-stack hourglass architecture (Newell et al. 2016) with our synthetic data for flow estimation on 2 consecutive frames. We refer the reader to Figure 10 for the qualitative results of our optical flow estimation.

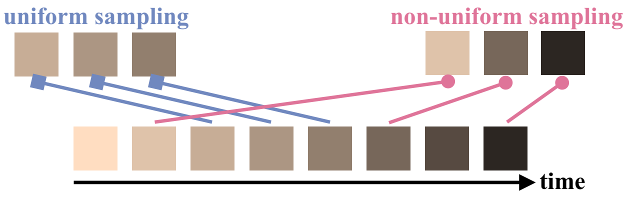

Fig. 4 Frame sampling: We illustrate our non-uniform frame sampling strategy in our 3D CNN training. Compared to the commonly adopted consecutive setting which uniformly samples with a fixed frame rate, non-uniform sampling has random skips in time, allowing speed augmentations and long-term context

Non-Uniform Frame Sampling. We adopt a different frame sampling strategy than most works (Carreira and Zisserman 2017; Feichtenhofer et al. 2019; Hara et al. 2018) in the context of 3D CNNs. Instead of uniformly sampling (at a fixed frame rate) a video clip with consecutive frames, we randomly sample frames across time by keeping their temporal order, which we refer as *non-uniform* sampling. Although recent works explore multiple temporal resolutions, e.g. by regularly sampling at two different frame rates (Feichtenhofer et al. 2019), or randomly selecting a frame rate (Zhu and Newsam 2018), the sampled frames are equidistant from each other. TSN (Wang et al. 2016) and ECO (Zolfaghari et al. 2018) employ a hybrid strategy by regularly sampling temporal segments and randomly sampling a frame from each segment, which is a more restricted special case of our strategy. Moreover, TSN uses a 2D CNN without temporal modelling. Zolfaghari et al. (2018) also has 2D convolutional features on each frame, which are stacked as input to a 3D CNN only at the end of the network. None of these works provide controlled experiments to quantify the effect of their sampling strategy. The concurrent work of Chen et al. (2020) presents an experimental analysis comparing the dense consecutive sampling with the hybrid sampling of TSN.

Figure 4 compares the consecutive sampling with our non-uniform sampling. In our experiments, we report results for both and show improvements for the latter. Our videos are temporally trimmed around the action, therefore, each video is short, i.e. spans several seconds. During training we randomly sample 16 video frames as a fixed-sized input to 3D CNN. Thus, the convolutional kernels become speed-invariant to some degree. This can be seen as a data augmentation technique, as well as a way to capture long-term cues.

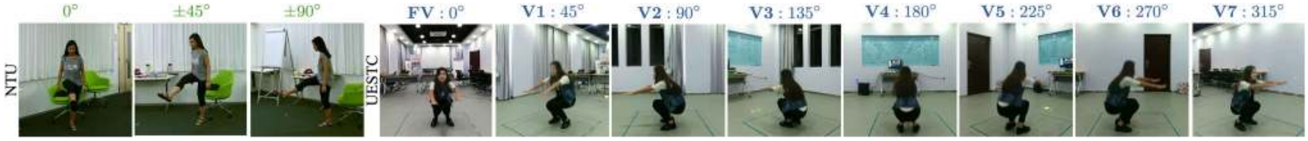

Fig. 5 Datasets: We show sample video frames from the multi-view datasets used in our experiments. NTU and UESTC datasets have 3 and 8 viewpoints, respectively. NTU views correspond to 0°, 45°, and 90° from left to right. UESTC covers 360° around the performer

At test time, we sample several 16-frame clips and average the softmax scores. If we test the uniform case, we sample non-overlapping consecutive clips with sliding window. For the non-uniform case, we randomly sample as many non-uniform clips as the number of sliding windows for the uniform case. In other words, the number of sampled clips is proportional to the video length. More precisely, let T be the number of frames in the entire test video, F be the number of input frames per clip, and S be the stride parameter. We sample N clips where $N = \lceil (T - F)/S) \rceil + 1$. In our case $F = 16$, $S = 16$. We apply sliding window for the uniform case. For the non-uniform case, we sample N clips, where each clip is an ordered random (without replacement) 16-frame subset from T. We observe that it is important to train and test with the same sampling scheme, and keeping the temporal order is important. More details can be found in Appendix B.5.

Synth+Real. Since each real video is augmented multiple times (e.g. 8 times for 8 views), we have more synthetic data than real. When we add synthetic data to training, we balance the real and synthetic datasets such that at each epoch we randomly subsample from the synthetic videos to have equal number for both real and synthetic.

We minimize the cross-entropy loss using RMSprop (Tieleman and Hinton 2012) with mini-batches of size 10 and an initial learning rate of 10^{-3} with a fixed schedule. Color augmentation is used for the RGB stream. Other implementation details are given in Appendix A.

4 Experiments

In this section, we start by presenting the action recognition datasets used in our experiments (Sect. 4.1). Next, we present extensive ablations for action recognition from unseen viewpoints (Sect. 4.2). Then, we compare our results to the state of the art for completeness (Sect. 4.3). Finally, we illustrate our approach on in-the-wild videos (Sect. 4.4).

4.1 Datasets and Evaluation Protocols

We briefly present the datasets used in this work, as well as the evaluation protocols employed.

NTU RGB+D Dataset (NTU). This dataset (Shahroudy et al. 2016) captures 60 actions with 3 synchronous cameras (see Fig. 5). The large scale (56K videos) of the dataset allows training deep neural networks. Each sequence has 84 frames on average. The standard protocols (Shahroudy et al. 2016) report accuracy for cross-view and cross-subject splits. The cross-view (CV) split considers 0° and 90 views as training and 45° view as test, and the same subjects appear both in training and test. For the cross-subject (CS) setting, 20 subjects are used for training, the remaining 20 for test, and all 3 views are seen at both training and test. We report on the standard protocols to be able to compare to the state of the art (see Table 8). However, we introduce a new protocol to make the task more challenging. From the cross-subject training split that has all 3 views, we take only 0° viewpoint for training, and we test on the 0°, 45°, 90° views of the cross-subject test split. We call this protocol cross-view-subject (CVS). Our focus is mainly to improve for the unseen and distinct view of 90°.

UESTC RGB-D Varying-view 3D Action Dataset (UESTC). UESTC is a recent dataset (Ji et al. 2018) that systematically collects 8 equally separated viewpoints that cover 360° around a person (see Fig. 5). In total, the dataset has 118 subjects, 40 actions categories, and 26500 videos of more than 200 frames each. This dataset allows studying actions from unusual views such as behind the person. We use the official protocol Cross View I (CV-I), suitable for our task, which trains with 1 viewpoint and tests with all other 7 for each view. The final performance is evaluated as the average across all tests. For completeness, we also report the Cross View II (CV-II) protocol that concentrates on multi-view training, i.e., training with even viewpoints (FV, V2, V4, V6) and testing with odd viewpoints (V1, V3, V5, V7), and vice versa.

One-shot Kinetics-15 Dataset (Kinetics-15). Since we wish to formulate a one-shot scenario from in-the-wild Kinetics (Kay et al. 2017) videos, we need a pre-trained model to serve as feature extractor. We use a model pre-trained on Mini-Kinetics-200 (Xie et al. 2017), a subset of Kinetics-400. We define the novel classes from the remaining 200 categories which can be described by body motions. This procedure resulted in a 15-class subset of Kinetics-400: *bending back, clapping, climbing a rope, exercising arm, hugging, jogging, jumpstyle dancing, krumping, push up, shaking hands, skip-*

Table 1 Training jointly on synthetic and real data substantially boosts the performance compared to only real training on NTU CVS protocol, especially on unseen views (45°, 90°) (e.g., 69.0% vs 53.6%). The improvement can be seen for both RGB and Flow streams, as well as the fusion. We note the marginal improvements with the addition of flow unlike in other tasks where flow has been used to reduce the synthetic-real domain gap (Doersch and Zisserman 2019). We render two different versions of the synthetic dataset using HMMR and VIBE motion estimation methods, and observe improvements with VIBE. Moreover, training on synthetic videos alone is able to obtain 63.0% accuracy

	RGB			Flow			RGB + Flow		
Training data	0°	45°	90°	0°	45°	90°	0°	45°	90°
Real(0°)	86.9	74.5	53.6	82.8	70.6	49.7	88.8	78.2	57.3
Synth$_{HMMR}$(0°:45°:360°)	54.0	49.5	42.7	51.7	46.9	38.6	60.6	55.5	47.8
Synth$_{HMMR}$(0°:45°:360°) + Real(0°)	**89.1**	**82.0**	**67.1**	**85.9**	**76.4**	**58.9**	**90.5**	**83.3**	**68.0**
Synth$_{VIBE}$(0°:45°:360°)	58.1	52.8	45.3	54.1	47.2	37.9	63.0	57.6	48.3
Synth$_{VIBE}$(0°:45°:360°) + Real(0°)	**89.7**	**82.0**	**69.0**	**85.9**	**77.7**	**61.8**	**90.6**	**83.4**	**71.1**

Table 2 Real baselines: Training and testing with our cross-view-subject (CVS) protocol of the NTU dataset using only real RGB videos. Rows and columns correspond to training and testing sets, respectively. Training and testing on the same viewpoint shows the best performance as can be seen by the diagonals of the first three rows. This shows the domain gap present between 0°, 45°, 90° viewpoints. If we add more viewpoints to the training (last two rows) we account for the domain gap. Non-uniform frame sampling (right) consistently outperforms the uniform frame sampling (left)

		Test views					
		uniform			non-uniform		
		0°	45°	90°	0°	45°	90°
Train views	0°	**83.9**	67.9	42.9	**86.9**	74.5	53.6
	45°	72.1	**81.6**	66.8	78.1	**85.2**	75.7
	90°	41.7	63.4	**81.4**	52.3	71.2	**85.4**
	0° + 45°	86.0	85.3	69.9	89.7	88.9	79.3
	0° + 45° + 90°	**86.8**	**86.9**	**84.1**	**89.4**	**89.4**	**87.8**

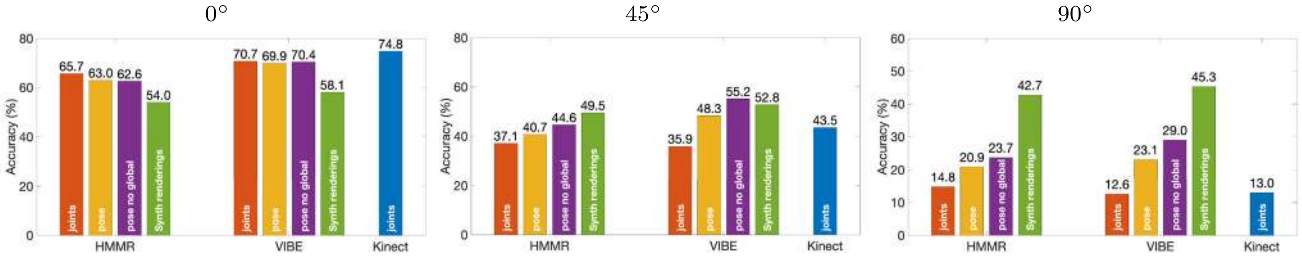

Fig. 6 Inputting raw motion parameters performs significantly worse for the 90° unseen viewpoint compared to synthetic renderings on the NTU CVS protocol. We compare various input representations with increasing view-independence (joint coordinates, SMPL pose parameters, SMPL pose parameters without the global rotation). Experiments are carried out with SMPL model recovered with RGB-based methods HMMR (Kanazawa et al. 2019) and VIBE (Kocabas et al. 2020), and depth-based Kinect joints. A 2D ResNet architecture is used for motion parameter inputs similar to Ke et al. (2017). We also present an architecture study in Table 3. Note that significant gains are further possible when mixing the synthetic renderings with real videos. See text for interpretation

ping rope, stretching arm, swinging legs, sweeping floor, wrestling. Note that many of the categories such as *waiting in line, dining, holding snake* cannot be recognized solely by their body motions, but additional contextual cues are needed. From the 15 actions, we randomly sample 1 training video per class (see Fig. 8 for example videos with their synthetic augmentations). The training set therefore consists of 15 videos. For testing, we report accuracy on all 725 validation videos from these 15 classes. The limitation of this protocol is that it is sensitive to the choice of the 15 training videos, e.g., if 3D motion estimation fails on one video, the model will not benefit from additional synthetic data of one class. Future work can consider multiple possible training sets (e.g., sampling videos where 3D pose estimation is confident) and report average performance.

4.2 Ablation Study

We first compare real-only (Real), synthetic-only (Synth), and mixed synthetic and real (Synth+Real) training. Next, we explore the effect of the motion estimation quality and inputting raw motion parameters as opposed to synthetic renderings. Then, we experiment with the different synthetic data generation parameters to analyze the effects of viewpoint and motion diversity. In all cases, we evaluate our models on real test videos.

Real Baselines. We start with our cross-view-subject protocol on NTU by training only with real data. Table 2 summarizes the results of training the model on a single-view and testing on all views. We observe a clear domain gap between different viewpoints, which can be naturally reduced by adding more views in training. However, in the case when a single view is available, this would not be possible. If we train only with 0°, the performance is high (83.9%) when tested on 0°, but significantly drops (42.9%) when tested on 90°. In the remaining of our experiments on NTU, we assume that only the frontal viewpoint (0°) is available.

Non-Uniform Frame Sampling. We note the consistent improvement of non-uniform frame sampling over the uniform consecutive sampling in all settings in Table 2. Additional experiments about video frame sampling, such as the optical flow stream, can be found in Appendix B.5. We use our non-uniform sampling strategy for both RGB and flow streams in the remainder of experiments unless specified otherwise.

Synth+Real Training. Next, we report the improvements obtained by synthetically increasing view diversity. We train the 60 action classes from NTU by combining the real 0° training data and the synthetic data augmented from real with 8 viewpoints, i.e. 0°:45°:360°. Table 1 compares the results of Real, Synth, and Synth+Real trainings for RGB and Flow streams, as well as their combination. The performance of the flow stream is generally lower than that of the RGB stream, possibly due to the fine-grained categories which cannot be distinguished with coarse motion fields.

It is interesting to note that training only with synthetic data (Synth) reaches 63.0% accuracy on real 0° test data which indicates a certain level of generalization capability from synthetic to real. Combining real and synthetic training videos (Real+Synth), the performance of the RGB stream increases from 53.6% to 69.0% compared to only real training (Real), on the challenging unseen 90° viewpoint. Note that the additional synthetic videos can be obtained ‘for free’, i.e. without extra annotation cost. We also confirm that even the noisy motion estimates are sufficient to obtain significant improvements, suggesting that the discriminative action information is still present in our synthetic data.

The advantage of having a controllable data generation procedure is to be able to analyze what components of the synthetic data are important. In the following, we examine a few of these aspects, such as quality of the motion estimation, input representation, amount of data, view diversity, and motion diversity. Additional results can be found in Appendix B.

Quality of the Motion Estimation: HMMR vs VIBE. 3D motion estimation from monocular videos has only recently demonstrated convincing performance on unconstrained videos, opening up the possibility to investigate our problem of action recognition with synthetic videos. One natural question is whether the progress in 3D motion estimation methods will improve the synthetic data. To this end, we compare two sets of synthetic data, keeping all the factors the same except the motion source: Synth$_{HMMR}$ extracted with HMMR (Kanazawa et al. 2019), Synth$_{VIBE}$ extracted with VIBE (Kocabas et al. 2020). Table 1 presents the results. We observe consistent improvements with more accurate pose estimation from VIBE over HMMR, suggesting that our proposed pipeline has great potential to further improve with the progress in 3D recovery.

Raw Motion Parameters as Input. Another question is whether the motion estimation output, i.e., body pose parameters, can be directly used as input to an action recognition model instead of going through synthetic renderings. We implement a simple 2D CNN architecture similar to Ke et al. (2017) that inputs 16-frame pose sequence in the form of 3D joint coordinates (24 joints for SMPL, 25 joints for Kinect) or 3D joint rotations (24 axis-angle parent-relative rotations for SMPL, or 23 without the global rotation). In particular, we use a ResNet-18 architecture (He et al. 2015). We experiment with both HMMR and VIBE to use SMPL parameters as input, as well as Kinect joints provided by the NTU dataset for comparison. Figure 6 reports the results of various pose representations against the performance of synthetic renderings for three test views. We make several observations: (1) Removing viewpoint-dependent factors, e.g., pose parameters over joints, degrades performance on seen viewpoint, but consistently improves on unseen viewpoints; (2) Synthetic video renderings from all viewpoints significantly improve over raw motion parameters for the challenging unseen viewpoint; (3) VIBE outperforms HMMR; (4) Both RGB-based motion estimation methods are competitive with the depth-based Kinect joints.

We note the significant boost with renderings (45.3%) over pose parameters (29.0%) for the 90° test view despite the same source of motion information for both. There are three main differences which can be potential reasons. First, the architectures 3D ResNet and 2D ResNet have different capacities. Second, motion estimation from non-frontal viewpoints can be challenging, negatively affecting the performance of pose-based methods, but not affecting 3D ResNet (because pose estimation is not a required step). Third, the renderings have the advantage that standard data augmentation

Table 3 Architecture comparison: We explore the influence of architectural improvements for pose-based action recognition models: 2D ResNet with temporal convolutions versus ST-GCN with graph convolutions on the SMPL pose parameters obtained by VIBE. While ST-GCN improves over 2D ResNet, the performance of the synthetic-only training with renderings remain superior for the unseen 90° viewpoint

Arch.	Input	0°	45°	90°
2D ResNet	Pose	69.9	48.3	23.1
2D ResNet	Pose no global	70.4	55.2	29.0
ST-GCN	Pose	74.8	59.8	31.4
ST-GCN	Pose no global	75.6	60.9	36.2
3D ResNet	Synth	58.1	52.8	45.3

Table 4 Viewpoint diversity: The effect of the views in the synthetic training on the NTU CVS split. We train only with synthetic videos obtained from real data of 60 sequences per action. We take a subset of views from the synthetic data: 0°, ±45°, ±90°. Even when synthetic, the performance is better when the viewpoints match between training and test. The best performance is obtained with all 8 viewpoints combined

	0°	45°	90°
Synth(0°)	38.3	27.1	17.9
Synth(45°, 315°)	35.9	34.2	26.8
Synth(90°, 270°)	13.9	18.3	23.2
Synth(0°:45°:360°)	**48.3**	**44.3**	**38.8**

techniques on image pixels can be applied, unlike the pose parameters which are not augmented. More importantly, the renderings have the advantage that they can be mixed with the real videos, which showed to substantially improve the performance in Table 1.

To explore the architectural capacity question, we study the pose-based action recognition model further and experiment with the recent ST-GCN model (Yan et al. 2018) that makes use of graph convolutions. For this experiment, we use VIBE pose estimates and compare ST-GCN with the 2D ResNet architecture in Table 3. Although we observe improvements with using ST-GCN (29.0% vs 36.2%), the synthetic renderings provide significantly better generalization to the unseen 90° view (45.3%).

Amount of Data. In the NTU CVS training split, we have about 220 sequences per action. We take subsets with {10, 30, 60, 100} sequences per action, and train the three scenarios: Real, Synth, Synth+Real, for each subset. Figure 7 plots the performance versus the amount of data for these scenarios, for both RGB and Flow streams. We observe the consistent improvement of complementary synthetic training, especially for unseen viewpoints. We also see that it is more effective to use synthetic data at a given number of sequences per action. For example, on the 90° viewpoint, increasing the number of sequences from 100 to 220 in the real data results only in 4.6% improvement (49.0% vs 53.6%, Real), while one can synthetically augment the existing 100 sequences per action and obtain 64.7% (Synth+Real) accuracy without spending extra annotation effort.

View Diversity. We wish to confirm that the improvements presented so far are mainly due to the viewpoint variation in synthetic data. The "Synth(v=0°) + Real" plot in Fig. 7 indicates that only the 0° viewpoint from synthetic data is used. In this case, we observe that the improvement is not consistent. Therefore, it is important to augment viewpoints to obtain improvements. Moreover, we experiment with having only ±45° or ±90° views in the synthetic-only training for 60 sequences per action. In Table 4, we observe that the test performance is higher when the synthetic training view matches the real test view. However, having all 8 viewpoints at training benefits all test views.

Motion Diversity. Next, we investigate the question whether motions can be diversified and whether this is beneficial for synthetic training. There are very few attempts towards this

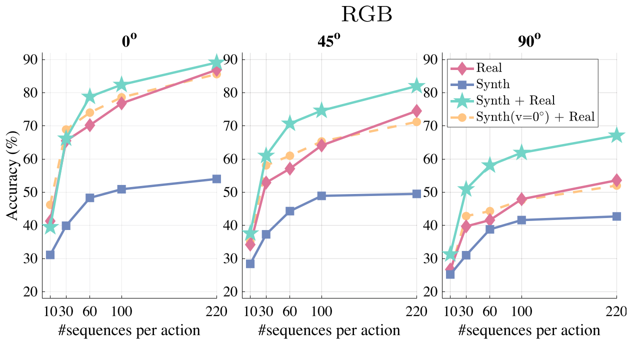

Fig. 7 Amount of data: The number of real sequences per action for: Real, Synth, Synth+Real training on NTU CVS split. Generalization to unseen viewpoints is significantly improved with the addition of synthetic data (green) compared to training only with real (pink). Real training contains the 0° view. We experiment with all 8 views (green) or only the 0° view (yellow) in the additional synthetic data. See text for interpretation

Table 5 Motion diversity: We study the effect of motion diversity in the synthetic training on a subset of the NTU CVS split. The results indicate that clothing, body shape diversity is not as important as motion diversity (second and last rows). We can significantly improve the performance by motion augmentations, especially with a video-level additive noise on the joint rotations (second and sixth rows). Here, each dataset is rendered with all 8 views and the training is only performed on synthetic data. At each rendering, we randomly sample clothing, body shape, lighting etc

#Seq./		Motion	Test views		
action	#Render	augm.	0°	45°	90°
10	1	–	31.1	28.4	25.2
10	6	–	33.1	31.6	26.2
10	6	Interp.	35.7	31.5	26.9
10	6	Add. noise [frame]	25.0	24.2	21.4
10	6	Add. noise [every 25f]	32.5	31.3	28.0
10	6	Add. noise [video]	**37.9**	**35.9**	**31.5**
60	1	–	48.3	44.3	38.8

direction (De Souza et al. 2017) since synthetic data has been mainly used for static images. Recently, (Liu et al. 2019a) introduced interpolation between distinct poses to create new poses in synthetic data for training 3D pose estimation; however, its contribution over existing poses was not experimentally validated. In our case, we need to preserve the action information, therefore, we cannot generate unconstrained motions. Generating realistic motions is a challenging research problem on its own and is out of the scope of this paper. Here, we experiment with motion augmentation to increase diversity.

As explained in Sect. 3.1, we generate new motion sequences by (1) interpolating between motion pairs of the same class, or by (2) additive noise on the pose parameters. Table 5 presents the results of this analysis when we train only with synthetic data and test on the NTU CVS protocol. We compare to the baseline where 10 motion sequences per action are rendered once per viewpoint (the first row). We render the same sequences without motion augmentation 6 times (the second row) and obtain marginal improvement. On the other hand, having 60 real motion sequences per action significantly improves (last row) and is our upper bound for the motion augmentation experiments. That means that the clothing, body shape, lighting, i.e. appearance diversity is not as important as motion diversity. We see that generating new sequences with interpolations improves over the baseline. Moreover, perturbing the joint rotations across the video with additive noise is simple and effective, with performance increase of about 5% (26.2% vs 31.5%) over rendering 6 times without motion augmentation. To justify the video-level noise (i.e., one value to add to all frames), in Table 5, we also experiment with frame-level noise and a hybrid version where we independently sample a noise at every 25 frames, which are interpolated for the frames in between. These renderings qualitatively remain very noisy, reducing the performance in return.

4.3 Comparison with the State of the Art

In the following, we employ the standard protocols for UESTC and NTU datasets, and compare our performance with other works. Tables 6 and 7 compare our results to the state-of-the-art methods reported by Ji et al. (2018) on the recently released UESTC dataset, on CV-I and CV-II protocols. To augment the UESTC dataset, we use the VIBE motion estimation method. We outperform the RGB-based methods JOULE (Hu et al. 2017) and 3D ResNeXt-101 (Hara et al. 2018) by a large margin even though we use a less deep 3D ResNet-50 architecture. We note that we have trained the ResNeXt-101 architecture (Hara et al. 2018) with our implementation and obtained better results than our ResNet-50 architecture (45.2% vs 36.1% on CV-I, 82.5% vs 76.1% on CV-II). This contradicts the results reported in Ji et al. (2018). We note that a first improvement can be attributed to our non-uniform frame sampling strategy. Therefore, we report

Table 6 UESTC dataset Cross View I protocol: Training on 1 viewpoint and testing on all the others. The plots on the right show individual performances for the RGB networks. The rows and columns of the matrices correspond to training and testing views, respectively. We obtain significant improvements over the state of the art, due to our non-uniform frame sampling and synthetic training

Method		Modality	Accuracy (%)
VS-CNN (Ji et al. 2018)		Skeleton	29.0
JOULE (Hu et al. 2017) by (Ji et al. 2018)		RGB	31.0
ResNeXt-101 (Hara et al. 2018) by (Ji et al. 2018)		RGB	32.0
ResNeXt-101 (Hara et al. 2018) (ours)		RGB	45.2
RGB	Real [uniform]	RGB	36.1
RGB	Real	RGB	49.4
	Synth + Real	RGB	**66.4**
Flow	Real	RGB	63.5
	Synth + Real	RGB	**73.1**
RGB + Flow	Real	RGB	63.2
	Synth + Real	RGB	**76.1**

Table 7 UESTC dataset Cross View II protocol: Training on 4 odd viewpoints, testing on 4 even viewpoints (left), and vice versa (right). We present the results on both splits and their average for the RGB and Flow streams, as well as the RGB+Flow late fusion. Real+Synth training consistently outperforms the Real baseline

Training views:			V1, V3, V5, V7					FV, V2, V4, V6					
Test views:			FV	V2	V4	V6	Avg$_{even}$	V1	V3	V5	V7	Avg$_{odd}$	Avg
VS-CNN (Ji et al. 2018)		Skeleton	87.0	54.0	71.0	60.0	68.0	87.0	58.0	60.0	87.0	73.0	70.5
JOULE (Hu et al. 2017) by (Ji et al. 2018)		RGB	74.0	49.0	57.0	55.0	58.8	74.0	48.0	47.0	80.0	62.3	60.6
ResNeXt-101 (Hara et al. 2018) by (ji et al. 2018)		RGB	51.0	40.0	54.0	39.0	46.0	52.0	44.0	48.0	52.0	49.0	47.5
ResNeXt-101 (Hara et al. 2018) (ours)		RGB	78.0	71.7	79.4	65.4	73.6	94.0	91.3	89.9	89.9	91.3	82.5
RGB	Real [uniform]	RGB	78.1	63.1	76.6	46.4	66.1	92.1	80.9	85.5	86.1	86.2	76.1
RGB	Real	RGB	69.9	57.1	79.1	51.1	64.3	91.3	86.8	89.4	88.9	89.1	76.7
	Synth + Real	RGB	**79.4**	**75.8**	**83.6**	**73.3**	**78.0**	**95.8**	**91.2**	**92.8**	**93.9**	**93.4**	**85.7**
Flow	Real	RGB	73.5	68.4	81.2	60.3	70.9	94.6	83.4	**89.8**	90.8	89.7	80.3
	Synth + Real	RGB	**79.0**	**73.1**	**84.6**	**73.7**	**77.6**	**95.2**	**87.8**	89.2	**92.7**	**91.2**	**84.4**
RGB + Flow	Real	RGB	74.5	68.4	82.4	59.4	71.2	95.8	88.9	91.4	92.3	92.1	81.7
	Synth + Real	RGB	**82.4**	**77.7**	**85.6**	**76.8**	**80.6**	**96.6**	**92.3**	**92.9**	**94.9**	**94.2**	**87.4**

Table 8 State of the art comparison: We report on the standard protocols of NTU for completeness. We improve previous RGB-based methods (bottom) due to non-uniform sampling and synthetic training. Additional cues extracted from RGB modality are denoted in parenthesis. We perform on par with skeleton-based methods (top) without using the Kinect sensor

Method		Modality	CS	CV
Shahroudy et al. (2016)	Part-LSTM	Skeleton	62.9	70.3
Liu et al. (2016)	ST-LSTM	Skeleton	69.2	77.7
Liu et al. (2017a)	GCA-LSTM	Skeleton	74.4	82.8
Ke et al. (2017)	MTLN	Skeleton	79.6	84.8
Liu et al. (2017b)	View-invariant	Skeleton	80.0	87.2
Baradel et al. (2017)	Hands attention	RGB+Skeleton	84.8	90.6
Liu and Yuan (2018)	Pose evolution	RGB+Depth	91.7	95.3
Si et al. (2019)	Attention LSTM	Skeleton	89.2	95.0
Shi et al. (2019b)	2s-AGCN	Skeleton	88.5	95.1
Shi et al. (2019a)	DGNN	Skeleton	89.9	96.1
Baradel et al. (2017)	Hands attention	RGB (Pose)	75.6	80.5
Liu and Yuan (2018)	Pose evolution	RGB (Pose)	78.8	84.2
Zolfaghari et al. (2017)	Multi-stream	RGB (Pose+Flow)	80.8	–
Luvizon et al. (2018)	Multi-task	RGB (Pose)	85.5	–
Baradel et al. (2018)	Glimpse clouds	RGB (Pose)	86.6	93.2
Wang et al. (2018)	DA-Net	RGB (Flow)	88.1	92.0
Luo et al. (2018)	Graph distillation	RGB (Pose+Flow+Depth)	89.5	–
Real RGB [uniform]		RGB	86.3	90.8
Real RGB		RGB	89.0	93.1
Real Flow		RGB (Flow)	84.4	90.9
Real RGB+Flow		RGB (Flow)	90.0	94.3
Synth+Real RGB		RGB	89.6	94.1
Synth+Real Flow		RGB (Flow)	85.6	91.4
Synth+Real RGB+Flow		RGB (Flow)	**90.7**	**95.0**

our uniform real baseline as well. A significant performance boost is later obtained by having a mixture of synthetic and real training data. Using only RGB input, we obtain 17.0% improvement on the challenging CV-I protocol over real data (66.4 vs 49.4). Using both RGB and flow, we obtain 44.1% improvement over the state of the art (76.1 vs 32.0). We also report on the even/odd test splits of the CV-II protocol that have access to multi-view training data. The synthetic data again shows benefits over the real baselines. Compared to NTU, which contains object interactions that we do not simulate, the UESTC dataset focuses more on the anatomic movements, such as body exercises. We believe that these results convincingly demonstrate the generalization capability of our efficient synthetic data generation method to real body motion videos.

In Table 8, we compare our results to the state-of-the-art methods on standard NTU splits. The synthetic videos are generated using the HMMR motion estimation method. Our results on both splits achieve state-of-the-art performance only with the RGB modality. In comparison, (Baradel et al. 2018; Luvizon et al. 2018; Zolfaghari et al. 2017) use pose information during training. Luo et al. (2018) uses other modalities from Kinect such as depth and skeleton during training. Similar to us, (Wang et al. 2018) uses a two-stream approach. Our non-uniform sampling boosts the performance. We have moderate gains with the synthetic data for both RGB and flow streams, as the real training set is already large and similar to the test set.

4.4 One-Shot Training

We test the limits of our approach on unconstrained videos of the Kinetics-15 dataset. These videos are challenging for several reasons. First, the 3D human motion estimation fails often due to complex conditions such as motion blur, low-resolution, occlusion, crowded scenes, and fast motion. Second, there exist cues about the action context that are difficult to simulate, such as object interactions, bias towards certain clothing or environments for certain actions. Assuming that body motions alone, even when noisy, provide discriminative information for actions, we augment the 15 training videos of one-shot Kinetics-15 subset synthetically using HMMR (see Fig. 8) by rendering at 5 viewpoints (0°, 30°, 45°, 315°, 330°).

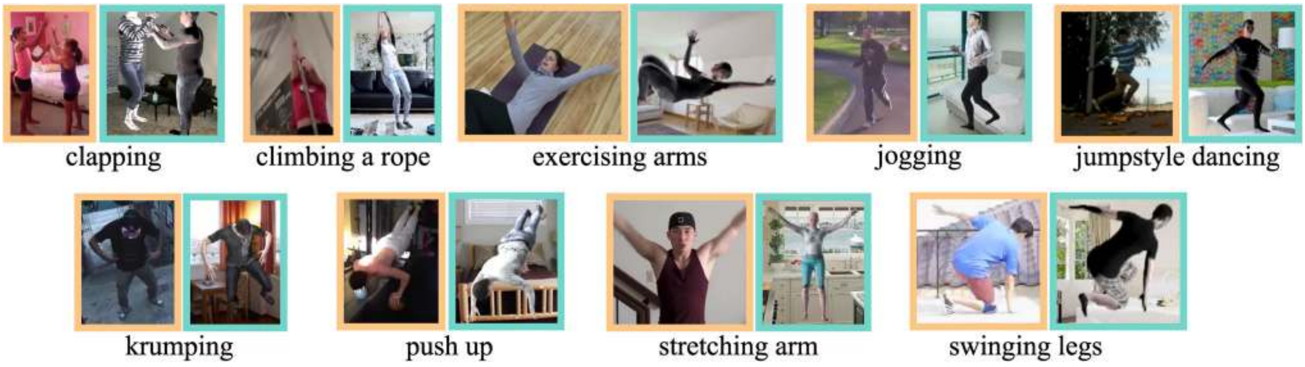

Fig. 8 Sample video frames from the one-shot Kinetics-15 dataset. We provide side-by-side illustrations for real frames and their synthetically augmented versions from the original viewpoint. Note that we render the synthetic body on a static background for computational efficiency, but augment it during training with random real videos by using the segmentation mask

Table 9 One-shot Kinetics-15: Real training data consists of 1 training sample per category, i.e., 15 videos. Random chance and nearest neighbor rows present baseline performances for this setup. We augment each training video with 5 different viewpoints by synthetically rendering SMPL sequences extracted from real data (i.e., 75 videos), blended on random backgrounds from the Mini-Kinetics training videos and obtain 6.5% improvement over training only with real data. For the last 4 rows, we train only the last linear layer of the ResNeXt-101 3D CNN model pre-trained on Mini-Kinetics 200 classes

Method	Synth background	Accuracy (%) RGB	Flow	RGB+Flow
Chance	–	6.7	6.7	6.7
Real (Nearest n.)	–	8.6	13.1	13.9
Synth	Mini-Kinetics	9.4	10.3	11.6
Real	–	26.2	20.6	28.4
Synth + Real	LSUN	26.3	21.1	29.2
Synth + Real	Mini-Kinetics	**32.7**	**22.3**	**34.6**

We use a pre-trained feature extractor model and only train a linear layer from the features to the 15 classes. We observe over-fitting with higher-capacity models due to limited one-shot training data. We experiment with two pre-trained models, obtained from Crasto et al. (2019): RGB and flow. The models follow the 3D ResNeXt-101 architecture from Hara et al. (2018) and are pre-trained on Mini-Kinetics-200 categories with 16 × 112 × 112 resolution with consecutive frame sampling.

In Table 9 (top), we first provide simple baselines: nearest neighbor with pre-trained features is slightly above random chance (8.6% vs 6.7% for RGB). Table 9 (bottom) shows training linear layers. Using only synthetic data obtains poor performance (9.4%). Training only with real data on the other hand obtains 26.2%, which is our baseline performance. We obtain ∼6% improvement by adding synthetic data. We also experiment with static background images from the LSUN dataset (Yu et al. 2015) and note the importance of realistic noisy backgrounds for generalization to in-the-wild videos.

5 Conclusions

We presented an effective methodology for automatically augmenting action recognition datasets with synthetic videos. We explored the importance of different variations in the synthetic data, such as viewpoints and motions. Our analysis emphasizes the question on how to diversify motions within an action category. We obtain significant improvements for action recognition from unseen viewpoints and one-shot training. However, our approach is limited by the performance of the 3D pose estimation, which can fail in cluttered scenes. Possible future directions include action-conditioned generative models for motion sequences and simulation of contextual cues for action recognition.

Acknowledgements This work was supported in part by Google Research, EPSRC grant ExTol, Louis Vuitton ENS Chair on Artificial Intelligence, DGA project DRAAF, and the French government under management of Agence Nationale de la Recherche as part of the Investissements d'Avenir program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute). We thank Angjoo Kanazawa, Fabien Baradel, and Max Bain for helpful discussions, Philippe Weinzaepfel and Nieves Crasto for providing pre-trained models.

APPENDIX

This appendix provides detailed explanations for several components of our approach (Sect. A). We also report complementary results for synthetic training, and our non-uniform frame sampling strategy (Sect. B).

A Additional Details

SURREACT Rendering. We build on the implementation of Varol et al. (2017) and use the Blender software. We add support for multi-person images, for using estimated motion inputs, for systematic viewpoint rendering, and different sources for background images. We use the cloth textures released by Varol et al. (2017), i.e., 361/90 female, 382/96 male textures for training/test splits, respectively. The resolution of the video frames is similarly 320x240 pixels. For background images, we used 21567/8790 train/test images extracted from NTU videos, and 23034/23038 train/test images extracted from UESTC videos, by sampling a region outside of the person bounding boxes. The rendering code takes approximately 6 seconds per frame, for saving RGB, body-part segmentation and optical flow data. We parallelize the rendering over hundreds of CPUs to accelerate the data generation.

Motion Sequence Interpolation. As explained in Sect. 3.2 of the main paper, we explore creating new sequences by interpolating pairs of motions from the same action category. Here, we visually illustrate this process. Figure 9 shows two sequences of *sitting down* that are first aligned with dynamic time warping, and then linearly interpolated. We only experiment with equal weights when interpolating (i.e. 0.5), but one can sample different weights when increasing the number of sequences further.

3D Translation in SURREACT. In Sect. 3.2 of the main paper, we explained that we translate the people in the xy image plane only when there are multiple people in the scene. HMMR (Kanazawa et al. 2019) estimates the weak-perspective camera scale, jointly with the body pose and shape. We note that obtaining 3D translation of the person in the camera coordinates is an ambiguous problem. It requires the size of the person to be known. This becomes more challenging in the case of multi-person videos.

HMMR relies on 2D pose estimation to locate the bounding box of the person which then becomes the input to a CNN. The CNN outputs a scale estimation s_b together with the $[x_b, y_b]$ normalized image coordinates of the person center with respect to the bounding box. We first convert these values to be with respect to the original uncropped image: s and $[x, y]$. We can recover an approximate value for the z coordinate of the person center, by assuming a fixed focal length $F = 500$. The translation in z then becomes: $z = F/(0.5 * W * s)$, where W is the image resolution and s is the estimated camera scale. The translation of the person center then becomes $[x, y, z]$. In practice, the z values are very noisy whereas $[x, y]$ values are more reliable. We therefore assume that the person is always centered at $z = 0$ and apply the translation only in the xy plane.

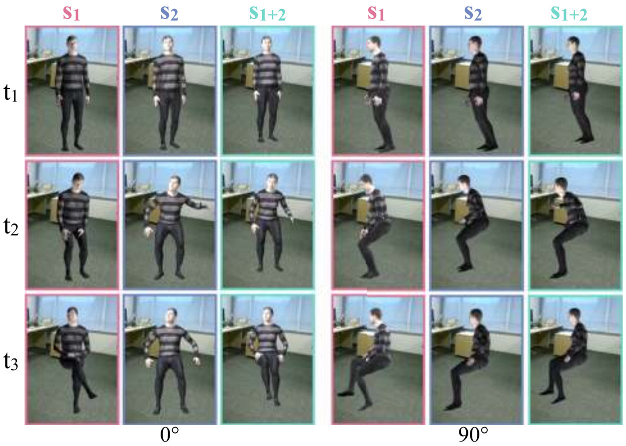

Fig. 9 Motion interpolation procedure for the *sitting* action. Two temporally aligned sequences s_1 and s_2 from different individuals are interpolated to create s_{1+2}, from two viewpoints. Note the new arm and leg angles that contribute to motion diversity

Table 10 Training on different versions of the synthetic data generated from 10 sequences per action from the NTU CVS protocol. We train only on synthetic and test on the real test set. Multi-person videos in the synthetic training improves performance, especially in the interaction categories (see Fig. 14). The noisy translation estimates degrades generalization, therefore, we use only xy translation and only in the case of multi-person. See text for further details

#People	Translation	0°	45°	90°
Single	xyz	18.3	17.8	15.0
Multi	xyz	21.0	21.2	17.3
Multi	xy	26.8	26.0	21.9
Multi	xy (when multi-person)	**28.5**	**27.2**	**23.0**

We observe that due to the noisy 2D person detections the estimated translation is noisy even in the xy image plane, leading to less generalization performance on real data when we train only with synthetic data. We validate this empirically in Table 10. We render multiple versions of the synthetic dataset with 10 motion sequences per action, each rendered from 8 viewpoints. We train only with this synthetic data and evaluate on the real NTU CVS protocol. Including multiple people improves performance (first and second rows), mainly because 11 out of 60 action categories in NTU are two-person interactions. Figure 14 also shows the confusion matrix of training only with single-person, resulting in the confusion of the interaction categories. Dropping the z component from the translation further improves (second and third rows). We also experiment with no translation if there is a single person,

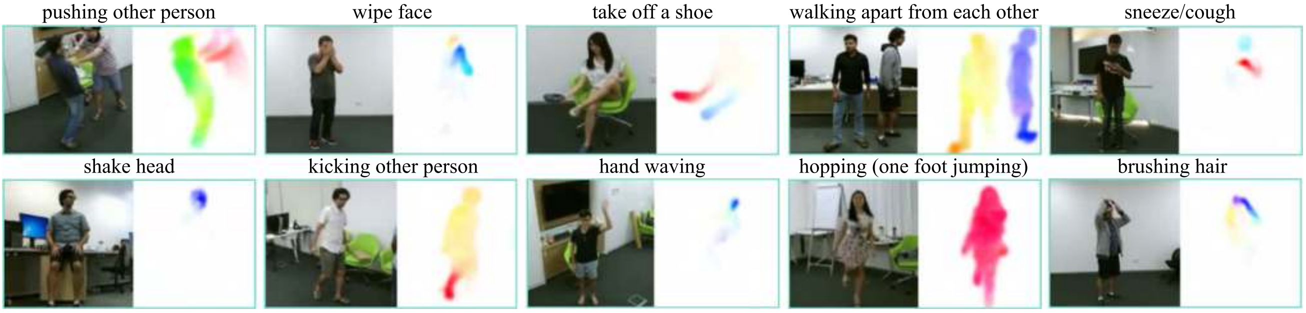

Fig. 10 Qualitative results for our optical flow estimation network trained on SURREACT, tested on the NTU dataset

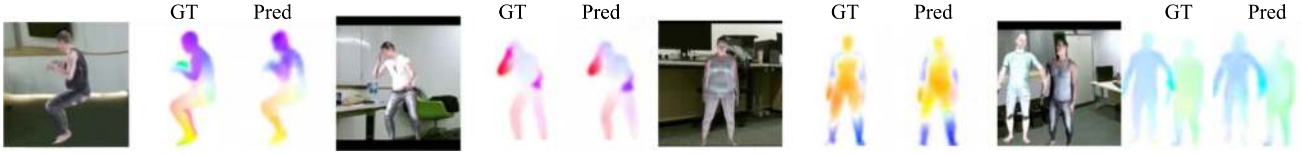

Fig. 11 Qualitative results for our optical flow estimation network trained and tested on SURREACT, together with the ground truth

and xy translation only for the multi-person case (fourth row), which has the best generalization performance. This is unintuitive since some actions such as *jumping* are not realistic when the vertical translation is not simulated. This indicates that the translation estimations from the real data need further improvement to be incorporated in the synthetic data. Our 3D CNN is otherwise sensitive to the temporal jitter induced by the noisy translations of the people.

Flow Estimation. We train our own optical flow estimation CNN, which we use to compute the flow in an online fashion, during action classification training. In other words, we do not require pre-processing the videos for training. To do so, we use a light-weight stacked hourglass architecture (Newell et al. 2016) with two stacks. The input and output have 256×256 and 64×64 spatial resolution, respectively. The input consists of 2 consecutive RGB frames of a video, the output is the downsampled optical flow ground truth. We train with mean squared error between the estimated and ground truth flow values. We obtain the ground truth from our synthetic SURREACT dataset. Qualitative results of our optical flow estimates can be seen in Figs. 10 and 11 on real and synthetic images, respectively. When we compute the flow *on-the-fly* for action recognition, we loop over the 16-frame RGB input to compute the flow between every 2 frames and obtain 15-frame flow field as input to the action classification network.

Training Details. We give additional details to Sect. 3.3 of the main paper on the action classification training. We train our networks for 50 epochs with an initial learning rate of 10^{-3} which is decreased twice with a factor of 10^{-1} at epochs 40 and 45, respectively. For NTU, UESTC, and (SURREACT project page) datasets, we spatially crop video frames around the person bounding box with random augmentations in scale and the center of the bounding box. For the Kinetics dataset, we crop randomly with a bias towards the center. We scale the RGB values between [0, 1] and jitter the color channels with a multiplicative coefficient randomly generated between [0.8, 1.2] for each channel. We subtract 0.5 and clip the values between [−0.5, 0.5] before inputting to the CNN.

NTU CVS Protocol. We provide Table 11 with the number of videos used for each NTU protocol, summarizing the difference of our new CVS protocol from the official cross-view (CV) and cross-subject (CS) splits. The CVS protocol addresses two problems with the official splits: (1) while CV uses same subjects across splits and CS uses same views across splits, CVS uses different subjects and different views between train and test; (2) our cross-view setup of train(0), test(90) is much more challenging than train(0+90), test(45) due to viewpoints being more distinct, especially a problem with CV where the same subjects are used in train and test.

B Additional Analyses

We analyze further the synthetic-only training (Sect. B.1) and synthetic+real training (Sect. B.2). We define a synthetic test set and report the results of the models in the main paper also on this test set. We present additional ablations. We report the confusion matrix on the synthetic test set, as well as on the real test set, which allows us to gain insights about which action categories can be represented better synthetically. Finally, we explore the proposed non-uniform sampling more in Sect. B.5.

Table 11 Statistics of NTU splits: The above table summarizes the partition of the dataset into views and subjects. The below table provides the number of videos we use for each NTU protocol. *The difference between these numbers is because we filter out some videos for which there exist no synchronized camera, to keep the number of test videos same for each test view. Similarly we use synchronized cameras in training, therefore slightly lower number of training videos in CV (37344 instead of 37644=18889+18755) and CS (39675 instead of 40089) but the test sets reflect the official list of videos

	0°	45°	90°	Total
Train subjects	13386	13415	13288	40089
Test subjects	5503	5517	5467	16487
Total	18889	18932	18755	56576

		0°	45°	90°	Total	
CS	Train	13225	13225	13225	39675	Diff. sub., same view
	Test	5503	5517	5467	16487*	
CV	Train	18672	–	18672	37344	Same sub., diff. view
	Test	–	18932	–	18932	(easy: 0+90 train, 45 test)
CVS	Train	13225	–	–	13225	Diff. sub., diff. view
	Test	5447	5447	5447	16341*	(challenging: 0 train, 90 test)

Table 12 The performance of the view-augmented models from Table 4 of the main paper on the synthetic test set. We train only with synthetic videos obtained from 60 sequences per action. We confirm that the viewpoints should match also for the synthetic test set. We report the viewpoint breakdown, as well as the average

	Synth test views								
	0°	45°	90°	135°	180°	225°	275°	315°	Avg
0	**81.7**	41.7	10.0	43.3	40.0	30.0	10.0	50.0	38.3
45-315	58.3	65.0	36.7	53.3	43.3	48.3	50.0	66.7	52.7
90-270	15.0	35.0	61.7	23.3	13.3	35.0	56.7	33.3	34.2
All 8	78.3	**73.3**	**61.7**	**68.3**	**73.3**	**75.0**	**65.0**	**71.7**	**70.8**

B.1 Synthetic-Only Training

Here, we define a synthetic test set based on the NTU actions, and perform additional ablations on our synthetic data such as different input modalities beyond RGB and flow, effect of backgrounds, effect of further camera augmentations, and confusion matrix analysis.

Synthetic Test Set. Similar to SURREAL (Varol et al. 2017), we separate the assets such as cloth textures, body shapes, backgrounds into train and test splits, which allows us to validate our experiments also on a synthetic test set. Here, we use one sequence per action from the real 0° test set to generate a small synthetic test set, i.e. 60 motion sequences in total, rendered for the 8 viewpoints, using the test set assets.

We report the performance of our models from Tables 4 and 5 of the main paper on this set. Table 12 confirms that the viewpoints should match between training and test for best results. Augmenting with all 8 views benefits the overall results. Table 13 presents the gains obtained by motion augmentations on the synthetic test set. Both interpolations and the additive noise improves over applying no augmentation.

Different Input Types. The advantage of having a synthetic dataset is to be able to perform experiments with different modalities. Specifically, we have ground-truth optical flow, body-part segmentation for each video. We compare training with these input modalities as opposed to RGB, or the estimated flow in Table 14. We evaluate on the real NTU CSV test set when applicable, and on the synthetic test set. We see that even when ground truth, the optical flow performs worse than RGB, indicating difficulty of distinguishing fine-grained actions only with flow fields. Body-part segmentation on the other hand, outperforms other modalities due to providing precise locations for each body part and an abstraction which reduces the gap between the training and test splits. In other words, body-part segmentation is independent of clothing,

Table 13 The performance of the motion-augmented models from Table 5 of the main paper on the synthetic test set. We train only with synthetic videos obtained from 60 sequences per action. Both augmentation approaches improve over the baseline

#Sequences per action	#Renders	Motion augmentation	Synth All
10	1	–	55.4
10	6	–	55.0
10	6	Interpolation	**58.8**
10	6	Additive noise	57.7
60	1	–	70.8

Table 14 Different input types when training only with synthetic data and testing on the synthetic test set, as well as the real NTU CVS test set when applicable. The data is generated from 60 sequences per action. The results indicate that body part segmentation can be an informative representation for action classification. Optical flow, even when ground truth (GT) is used, is less informative for fine-grained action classes in NTU

	Real			Synth
Input type	0°	45°	90°	All
Flow (Pred)	38.3	34.6	29.3	58.4
Flow (GT)	–	–	–	61.2
RGB	48.3	44.3	38.8	70.8
Body-part segm (GT)	–	–	–	**71.7**

Table 15 Effect of synthetic data backgrounds for synthetic-only training. Results are reported both on the real NTU CVS set and the synthetic test set. The synthetic training is generated from 60 sequences per action. Matching the target background statistics improves generalization to real. See text for details

		Real			Synth
Backgrounds		0°	45°	90°	All
Random	LSUN	39.1	37.3	32.5	70.8
Random	NTU	42.7	39.8	34.3	67.9
Fixed	NTU	**48.3**	**44.3**	**38.8**	**70.8**

Table 16 Training on different versions of the synthetic data generated from 10 sequences per action from the NTU CVS protocol. We ablate the importance of augmentations of the camera height and distance. We train only on synthetic and test on the real test set. We observe improvements with randomized camera positions besides the azimuth rotation

Camera height & distance	0°	45°	90°
Fixed	28.5	27.2	23.0
Random	**31.3**	**28.1**	**24.3**

lighting, background effects, but only contains motion and body shape information. This result highlights that we can improve action recognition by improving body part segmentation as in Zolfaghari et al. (2017).

Effect of Backgrounds. As explained in Sect. 3.2 of the main paper, we use 2D background images from the target action recognition domain in our synthetic dataset. We perform an experiment whether this helps on the NTU CVS setup. The NTU dataset is recorded in a lab environment, therefore has specific background statistics. We train models by replacing the background pixels of our synthetic videos randomly by LSUN (Varol et al. 2017; Yu et al. 2015) images or the original NTU images outside the person bounding boxes. Table 15 summarizes the results. Using random NTU backgrounds outperform using random LSUN backgrounds. However, we note that the process of using the segmentation mask creates some unrealistic artifacts around the person, which might contribute to the performance degradation. We therefore use the fixed backgrounds from the original renderings in the rest of the experiments.

Effect of Camera Height/Distance Augmentations. As stated in Sect. 3.2 of the main paper, we randomize the height and the distance of the camera to increase the viewpoint diversity within a certain azimuth rotation. We evaluate the importance of this with a controlled experiment in Table 16. We render two versions of the synthetic training set with 10 sequences per action from 8 viewpoints. The first one has a fixed distance and height at 5 meters and 1 meter, respectively. In the second one, we randomly sample from [4, 6] meters for the distance, and [−1, 3] meters for the height. We see that the generalization to real NTU CVS dataset is improved with increased randomness in the synthetic training. Visuals corresponding to the pre-defined range can be found in Fig. 12.

Confusion Matrices. We analyze two confusion matrices in Fig. 13: training only on the synthetic data and (1) testing on the synthetic test set; and (2) testing on the real NTU CVS 0° view test set. The confused classes are highlighted on the figure. The confusions on both test sets suggest that the fine-grained action classes require more precise body motions, such as {*clapping*, *rub two hands together*}, and {*reading*, *writing*}. Other confusions include object interaction categories (e.g. {*put on a hat*, *brushing hair*} and {*typing on a keyboard*, *writing*}), which can be explained by the fact that synthetic data does not simulate objects. These confusions are mostly resolved when training with both real and synthetic data.

B.2 Synthetic+Real Training

Amount of Additional Synthetic Data. In Fig. 7 of the main paper, we plotted the performance against the amount of action sequences in the training set for both synthetic and real datasets. Here, we also report Synth+Real training per-

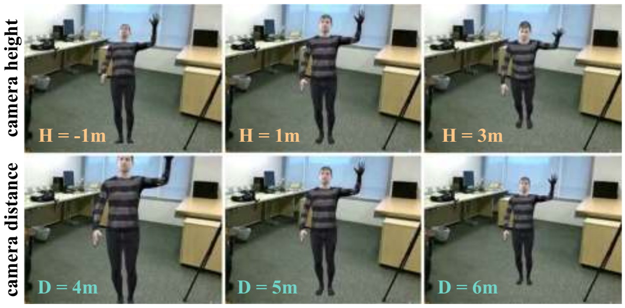

Fig. 12 We illustrate the limits for the camera height and distance parameters in SURREACT. We randomly sample between [-1, 3] and [4, 6] meters for the height and distance, respectively

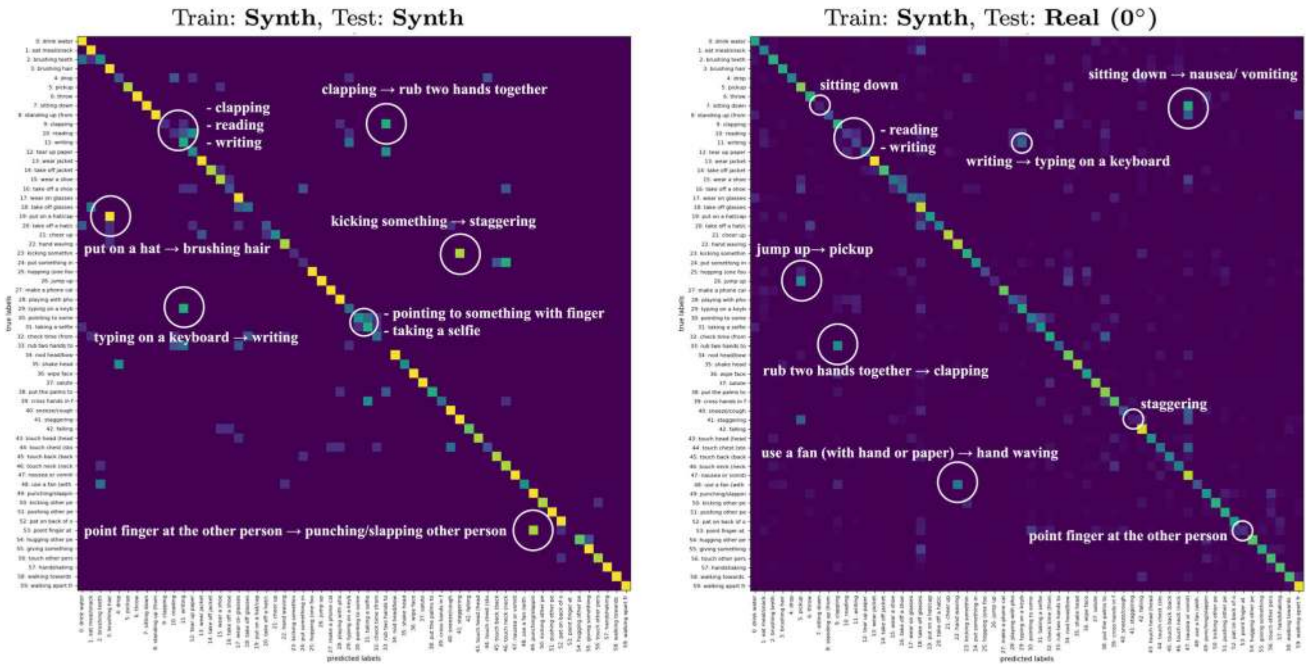

Fig. 13 The confusion matrices for training only with the final synthetic data with all the 220 sequences per action. Both on the synthetic test set (left) and the real 0° view test (right), the confusions are often between classes that are characterized by fine-grained body movements and object-interaction classes

formance when the Real data is fixed and uses all the action sequences available (i.e., 220 sequences per action), and the Synth data is gradually increased. Table 17 summarizes the results. Increasing the amount of synthetic data improves the performance. The improvement can be observed already at the challenging 90° view with as little synthetic data as 10 sequences per action (57.2% vs 53.6%). Using all the motions shows the most benefit as expected.

Synth+Real Training Strategies. In all our experiments, we combine the training sets of synthetic and real data to train jointly for both datasets, which we referred as Synth+Real. Here, we investigate whether a different strategy, such as using synthetic data as pre-training (as in Varol et al. (Varol et al. 2017)), would be more effective. In Table 18, we present several variations of training strategies. We conclude that our Synth+Real, is simple yet effective, while marginal gains can be obtained by continuing with fine-tuning only on Real data.

B.3 Performance Breakdown for Object-Related Actions

While the NTU dataset is mainly targeted for skeleton-based action recognition, many actions involve object interactions. In Table 19, we analyze the performance breakdown into action categories with and without objects. We notice that the object-related actions have lower performance than body-only counterparts even when trained with Real data. The gap is higher when only synthetic training is used since we simulate only humans, without objects.

Table 17 Amount of synthetic data addition: We experiment with Synth+Real RGB training while changing the number of sequences per action in the synthetic data and using all the real data. We conclude that using all available motion sequences improves the performance over taking a subset, confirming the importance of motion diversity. Results are reported on the NTU CVS protocol

	0°	45°	90°
Real(220)	86.9	74.5	53.6
Synth(10) + Real(220)	85.5	74.7	57.2
Synth(30) + Real(220)	85.2	77.4	61.8
Synth(60) + Real(220)	87.6	78.7	62.2
Synth(100) + Real(220)	87.7	78.8	63.7
Synth(220) + Real(220)	**89.1**	**82.0**	**67.1**

B.4 Pretraining on Kinetics

Throughout the paper, the networks for NTU training are randomly initialized (i.e., scratch). Here, we investigate whether there is any gain from Kinetics (Kay et al. 2017) pretraining. Table 20 summarizes the results. We refer to the table caption for the interpretation.

Table 18 Training strategies with Synthetic+Real: The bottom part of this table presents additional results for different training strategies on the NTU CVS protocol and the synthetic test set. The arrow A→B denotes training first on A and then fine-tuning on B dataset. S and R stand for Synthetic and Real, respectively. Pre-training only on one dataset is suboptimal (last three rows). Our choice of training by mixing S+R from scratch is simple yet effective. Marginal gains can be obtained by continuing training only on Real data (S+R→R)

Training	Real 0°	45°	90°	Synth All
S	54.0	49.5	42.7	70.4
R	86.9	74.5	53.6	18.1
S+R	89.1	**82.0**	67.1	71.0
S+R→R	**89.9**	81.9	**67.5**	73.1
S→R	84.1	77.5	66.2	65.6
S→S+R	81.6	75.1	63.6	**73.3**
R→S+R	84.3	75.1	59.9	56.7

Table 19 Object-related vs human-body actions: We report the performance breakdown into 28 object-related and 32 human-body actions for the NTU CVS protocol. In all three setups, Real, Synth, and Synth+Real, human-body action categories have higher performance than object-related categories

		0°	45°	90°
Real	All actions	86.9	74.5	53.6
	Human-body	88.3	75.8	58.3
	Object-related	85.2	73.0	48.1
Synth	All actions	58.1	52.8	45.3
	Human-body	62.9	60.5	55.9
	Object-related	52.4	43.9	33.0
Synth+Real	All actions	89.7	82.0	69.0
	Human-body	90.3	83.2	71.2
	Object-related	89.1	80.5	66.3

B.5 Non-Uniform Frame Sampling

In this section, we explore the proposed frame sampling strategy further.

First, we confirm that the benefits of non-uniform sampling applies also to the flow stream. Since flow is estimated online during training, we can compute flow between any two frames. Note that the flow estimation method is learned on 2 consecutive frames, therefore it produces noisy estimates for large displacements. However, even with this noise, in Table 21, we demonstrate advantages of non-uniform sampling over consecutive for the flow stream.

Next, we present our experiments about the testing modes as mentioned in Sect. 3.3 of the main paper. Table 22 suggests that the training and testing modes should be the same for both uniform and non-uniform samplings. The convolutional filters adapt to certain training statistics, which should be preserved at test time.

Table 21 Frame sampling for the flow stream: Training and testing on the real NTU CVS split. We confirm that the non-uniform sampling is beneficial also for the flow stream even though the flow estimates can be noisy between non-uniform frames

	0°	45°	90°
Flow [uniform]	80.6	68.3	44.7
Flow [non-uniform]	**82.8**	**70.6**	**49.7**

Table 22 Train/test modes: Training and testing on the real NTU CVS split. The frame sampling mode should be the same at training and test times

	Test mode Uniform 0°	45°	90°	Non-uniform 0°	45°	90°
Train uniform	**83.9**	**67.9**	**42.9**	27.5	20.6	13.8
Train non-uniform	32.1	21.1	12.4	**86.9**	**74.5**	**53.6**

Table 20 Effect of pretraining: We measure the effect of pretraining with Kinetics (Kay et al. 2017) over random initialization (real setting on the NTU CVS split). Interestingly, we do not observe improvements when the RMSProp optimizer is used, whereas SGD can improve the baselines from 53.6% to 55.4%. We note that this is still marginal compared to the boost we gain from synthetic data (69.0% in Table 1). Training only a linear layer on frozen features as opposed to end-to-end (e2e) finetuning is also suboptimal

Optimizer	Pretraining	Uniform 0°	45°	90°	Non-uniform 0°	45°	90°
RMSProp	Kinetics [linear]	42.5	33.3	27.2	48.1	37.7	31.1
	Kinetics [e2e]	81.9	66.9	43.8	85.6	72.2	51.3
	Scratch	83.9	67.9	42.9	**86.9**	**74.5**	**53.6**
SGD	Kinetics [linear]	39.8	31.7	26.2	45.4	35.1	28.8
	Kinetics [e2e]	89.4	69.8	40.5	**91.9**	**78.9**	**55.4**
	Scratch	81.9	65.8	41.4	85.3	72.8	52.5

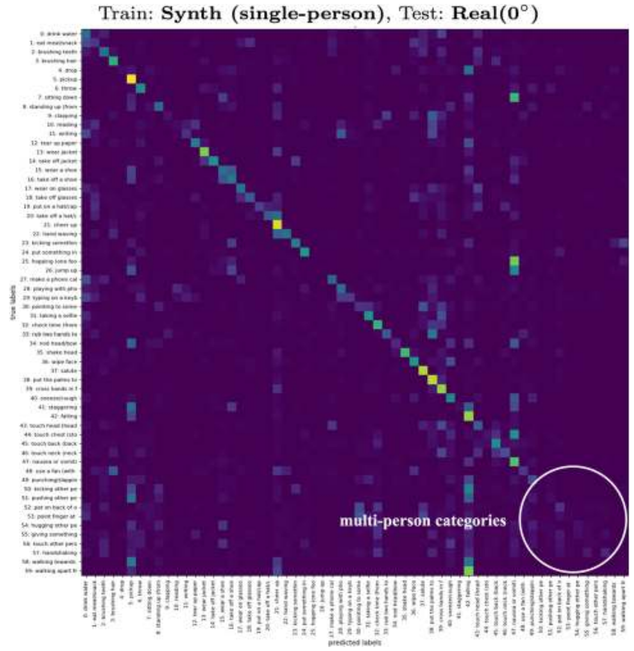

Fig. 14 We render a version of the synthetic data with 10 sequences per action, where we only insert a single person per video. When trained with this data, the two-person interaction categories (last 11 classes) are mostly misclassified on the real NTU CVS 0° view test data The confusions suggest that it is important to model multi-person cases in the synthetic data

Table 23 Frame order: Training and testing on the real NTU CVS split. Preserving the order of frames in non-uniform sampling is important. The confusion matrix in Fig. 15 shows that the mistakes are often among ‘symmetric’ action classes such as *sitting up* and *standing up*. Order-aware models fail drastically when tested non-ordered, as expected

	Test mode					
	Ordered			Non-ordered		
	0°	45°	90°	0°	45°	90°
Train ordered	**86.9**	**74.5**	**53.6**	16.4	13.1	8.1
Train non-ordered	67.2	50.2	32.8	**72.7**	**57.8**	**37.2**

We then investigate the importance of the frame order when we randomly sample non-uniformly. We preserve the temporal order in all our experiments, except in Table 23, where we experiment with a shuffled order. In this case, we observe a significant performance drop which can be explained by the confusion matrix in Fig. 15. The action classes such as *wearing* and *taking off* are heavily confused when the order is not preserved. This experiment allows detecting action categories that are temporally symmetric (Price and Damen 2019). We also observe that the ordered model fails when tested in non-ordered mode, which indicates that the convolutional kernels become highly order-aware.

Finally, we experiment with other frame sampling alternatives in Table 24. See the table caption for interpretation of the results.

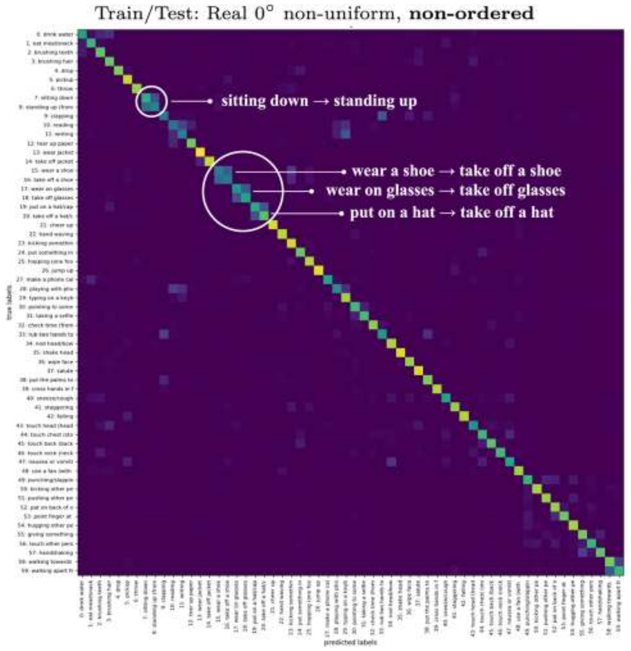

Fig. 15 We present the confusion matrix for the non-ordered training explained in Table 23. The classes that require the temporal order to be distinguished are confused as expected. The training and test is performed on the real NTU CVS 0° view split

Table 24 Frame sampling alternatives: Training and testing on the real NTU CVS split. We explore alternative random sampling schemes besides uniform baseline (1) and the random non-uniform (5). (2) applies a random frame rate similar to Zhu and Newsam (2018). However, (3) with a fixed frame rate of maximum temporal skip outperforms the random fps, suggesting the importance of long-term span. (4) the hybrid approach similar to Wang et al. (2016); Zolfaghari et al. (2018) increases the data augmentation while ensuring long-term context. (5) our fully random sampling maximizes the data augmentation. The approaches (3)(4)(5) perform similarly outperforming (1)(2)

	0°	45°	90°
1. Uniform, random shift, original fps	83.9	67.9	42.9
2. Uniform, random shift, random fps	84.1	69.9	48.9
3. Uniform, random shift, with smallest fps	85.6	74.0	54.0
4. Hybrid (random within uniform segments)	85.3	73.9	**54.3**
5. Non-uniform random	**86.9**	**74.5**	53.6

References

Carnegie-Mellon Mocap Database. http://mocap.cs.cmu.edu/.

Badler, N. I., Phillips, C. B., & Webber, B. L. (1993). *Simulating Humans: Computer Graphics Animation and Control*. New York, NY, USA: Oxford University Press Inc.

Baradel, F., Wolf, C., & Mille, J. (2017). Pose-conditioned spatio-temporal attention for human action recognition. *CoRR*. (**abs/1703.10106**).

Baradel, F., Wolf, C., Mille, J., & Taylor, G.W. (2018). Glimpse clouds: Human activity recognition from unstructured feature points. In: CVPR.

Bogo, F., Kanazawa, A., Lassner, C., Gehler, P., Romero, J., & Black, M.J. (2016). Keep it SMPL: Automatic estimation of 3D human pose and shape from a single image. In: ECCV.

Carreira, J., & Zisserman, A. (2017). *Quo vadis, action recognition?* A new model and the Kinetics dataset. In: CVPR.

Chen, C.F., Panda, R., Ramakrishnan, K., Feris, R., Cohn, J., Oliva, A., & Fan, Q. (2020). Deep analysis of CNN-based spatio-temporal representations for action recognition. arXiv preprint arXiv:2010.11757.

Chen, W., Wang, H., Li, Y., Su, H., Wang, Z., Tu, C., Lischinski, D., Cohen-Or, D., & Chen, B. (2016). Synthesizing training images for boosting human 3D pose estimation. In: 3DV.

Crasto, N., Weinzaepfel, P., Alahari, K., & Schmid, C. (2019). MARS: Motion-augmented RGB stream for action recognition. In: CVPR.

De Souza, C.R., Gaidon, A., Cabon, Y., & López Peña, A.M. (2017) Procedural generation of videos to train deep action recognition networks. In: CVPR.

Doersch, C., & Zisserman, A. (2019). Sim2real transfer learning for 3D pose estimation: Motion to the rescue. *CoRR*. (**abs/1907.02499**).

Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., van der Smagt, P., Cremers, D., & Brox, T. (2015). FlowNet: Learning optical flow with convolutional networks. In: ICCV.

Fang, H.S., Xie, S., Tai, Y.W., & Lu, C. (2017). RMPE: Regional multi-person pose estimation. In: ICCV.

Farhadi, A., & Tabrizi, M.K. (2008). Learning to recognize activities from the wrong view point. In: ECCV.

Feichtenhofer, C., Fan, H., Malik, J., & He, K. (2019). SlowFast networks for video recognition. In: ICCV.

Feichtenhofer, C., Pinz, A., & Zisserman, A. (2016). Convolutional two-stream network fusion for video action recognition. In: CVPR.

Ghezelghieh, M.F., Kasturi, R., & Sarkar, S. (2016). Learning camera viewpoint using CNN to improve 3D body pose estimation. In: 3DV.

Hara, K., Kataoka, H., & Satoh, Y. (2018). Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet? In: CVPR.

Hasson, Y., Varol, G., Tzionas, D., Kalevatykh, I., Black, M.J., Laptev, I., Schmid, C. (2019). Learning joint reconstruction of hands and manipulated objects. In: CVPR.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep residual learning for image recognition. In: CVPR.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735–1780.

Hoffmann, D.T., Tzionas, D., Black, M.J., & Tang, S. (2019). Learning to train with synthetic humans. In: GCPR.

Hu, J. F., Zheng, W. S., Lai, J., & Jianguo, Z. (2017). Jointly learning heterogeneous features for RGB-D activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 39*(11), 2186–2200.

Ji, Y., Xu, F., Yang, Y., Shen, F., Shen, H.T., & Zheng, W. (2018). A large-scale RGB-D database for arbitrary-view human action recognition. In: ACMMM.

Jingtian, Z., Shum, H., Han, J., & Shao, L. (2018). Action recognition from arbitrary views using transferable dictionary learning. *IEEE Transactions on Image Processing, 27,* 4709–4723.

Junejo, I. N., Dexter, E., Laptev, I., & Perez, P. (2011). View-independent action recognition from temporal self-similarities. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 33*(1), 172–185.

Kanazawa, A., Black, M.J., Jacobs, D.W., & Malik, J.(2018). End-to-end recovery of human shape and pose. In: CVPR.

Kanazawa, A., Zhang, J.Y., Felsen, P., & Malik, J. (2019) Learning 3D human dynamics from video. In: CVPR.

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., et al. (2017). The Kinetics human action video dataset. *CoRR*. (**abs/1705.06950**).

Ke, Q., Bennamoun, M., An, S., Sohel, F., & Boussaid, F. (2017). A new representation of skeleton sequences for 3D action recognition. In: CVPR.

Kocabas, M., Athanasiou, N., & Black, M.J. (2020). VIBE: Video inference for human body pose and shape estimation. In: CVPR.

Kolotouros, N., Pavlakos, G., Black, M.J., & Daniilidis, K. (2019) Learning to reconstruct 3D human pose and shape via model-fitting in the loop. In: ICCV.

Kolotouros, N., Pavlakos, G., & Daniilidis, K. (2019) Convolutional mesh regression for single-image human shape reconstruction. In: CVPR.

Kong, Y., Ding, Z., Li, J., & Fu, Y. (2017). Deeply learned view-invariant features for cross-view action recognition. *IEEE Transactions on Image Processing, 26*(6), 3028–3037.

Kong, Y., & Fu, Y. (2018). Human action recognition and prediction: A survey. *CoRR*. (**abs/1806.11230**).

Kortylewski, A., Egger, B., Schneider, A., Gerig, T., Morel-Forster, A., & Vetter, T. (2018). Empirically analyzing the effect of dataset biases on deep face recognition systems. In: CVPRW.

Lassner, C., Romero, J., Kiefel, M., Bogo, F., Black, M.J., & Gehler, P.V. (2017) Unite the people: Closing the loop between 3D and 2D human representations. In: CVPR.

LeCun, Y., Boser, B. E., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. E., & Jackel, L. D. (1989). Backpropagation applied to handwritten zip code recognition. *Neural Computation, 1*(4), 541–551.

Li, J., Wong, Y., Zhao, Q., & Kankanhalli, M. (2018). Unsupervised learning of view-invariant action representations. In: NeurIPS.

Li, W., Xu, Z., Xu, D., Dai, D., & Gool, L. V. (2018). Domain generalization and adaptation using low rank exemplar SVMs. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(5), 1114–1127.

Lin, J., Gan, C., Han, & S. (2019). TSM: Temporal shift module for efficient video understanding. In: ICCV.

Liu, J., Akhtar, N., & Mian, A. (2019). Temporally coherent full 3D mesh human pose recovery from monocular video. *CoRR*. (**abs/1906.00161**).

Liu, J., Rahmani, H., Akhtar, N., & Mian, A. (2019) Learning human pose models from synthesized data for robust RGB-D action recognition. International Journal of Computer Vision (IJCV), 127, 1545-1564.

Liu, J., Shah, M., Kuipers, B., & avarese, S. (2011). Cross-view action recognition via view knowledge transfer. In: CVPR.

Liu, J., Shahroudy, A., Xu, D., & Wang, G. (2016). Spatio-temporal LSTM with trust gates for 3D human action recognition. In: ECCV.

Liu, J., Wang, G., Hu, P., Duan, L.Y., & Kot, A.C. (2017). Global context-aware attention LSTM networks for 3D action recognition. In: CVPR.

Liu, M., Liu, H., & Chen, C. (2017). Enhanced skeleton visualization for view invariant human action recognition. *Pattern Recognition, 68*(C), 346–362.

Liu, M., & Yuan, J. (2018). Recognizing human actions as the evolution of pose estimation maps. In: CVPR.

Liu, Y., Lu, Z., Li, J., & Yang, T. (2019). Hierarchically learned view-invariant representations for cross-view action recognition. *IEEE Transactions on Circuits and Systems for Video Technology, 29,* 2416–2430.

Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M.J. (2015). SMPL: A skinned multi-person linear model. ACM transactions on graphics (TOG), 34(6), 1-16.

Luo, Z., Hsieh, J.T., Jiang, L., Niebles, J.C., & Fei-Fei, L. (2018). Graph distillation for action detection with privileged information. In: ECCV.

Luvizon, D.C., Picard, D., & Tabia, H. (2018). 2D/3D pose estimation and action recognition using multitask deep learning. In: CVPR.

Lv, F., & Nevatia, R.(2007). Single view human action recognition using key pose matching and viterbi path searching. In: CVPR.

Mahmood, N., Ghorbani, N., Troje, N.F., Pons-Moll, G., & Black, M.J. (2019). AMASS: Archive of motion capture as surface shapes. In: ICCV.

Marin, J., Vazquez, D., Geronimo, D., & Lopez, A.M. (2010). Learning appearance in virtual scenarios for pedestrian detection. In: CVPR.

Masi, I., Tran, A.T., Hassner, T., Sahin, G., & Medioni, G. (2019). Face-specific data augmentation for unconstrained face recognition. International Journal of Computer Vision (IJCV), 127, 642-667.

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In: ECCV.

Omran, M., Lassner, C., Pons-Moll, G., Gehler, P.V., & Schiele, B. (2018). Neural body fitting: Unifying deep learning and model-based human pose and shape estimation. In: 3DV.

Pavlakos, G., Zhu, L., Zhou, X., & Daniilidis, K. (2018). Learning to estimate 3D human pose and shape from a single color image. In: CVPR.

Pishchulin, L., Jain, A., Andriluka, M., Thormählen, T., & Schiele, B. (2012).Articulated people detection and pose estimation: Reshaping the future. In: CVPR.

Price, W., & Damen, D. (2019). Retro-Actions: Learning 'close' by time-reversing 'open' videos.

Puig, X., Ra, K., Boben, M., Li, J., Wang, T., Fidler, S., & Torralba, A. (2018). VirtualHome: Simulating household activities via programs. In: CVPR.

Qian, X., Fu, Y., Xiang, T., Wang, W., Qiu, J., Wu, Y., Jiang, Y.G., & Xue, X. (2018). Pose-normalized image generation for person re-identification. In: ECCV.

Rahmani, H., Mahmood, A., Huynh, D., & Mian, A. (2016). Histogram of oriented principal components for cross-view action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 38*(12), 2430–2443.

Rahmani, H., & Mian, A. (2015) Learning a non-linear knowledge transfer model for cross-view action recognition. In: CVPR.

Rahmani, H., & Mian, A. (2016). 3D action recognition from novel viewpoints. In: CVPR.

Rahmani, H., Mian, A., & Shah, M. (2018). Learning a deep model for human action recognition from novel viewpoints. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(3), 667–681.

Sakoe, H., & Chiba, S. (1978). Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 26*(1), 43–49.

Shahroudy, A., Liu, J., Ng, T.T., & Wang, G. (2016). NTU RGB+D: A large scale dataset for 3D human activity analysis. In: CVPR.

Shi, L., Zhang, Y., Cheng, J., & Lu, H. (2019). Skeleton-based action recognition with directed graph neural networks. In: CVPR.

Shi, L., Zhang, Y., Cheng, J., & Lu, H. (2019). Two-stream adaptive graph convolutional networks for skeleton-based action recognition. In: CVPR.

Shotton, J., Fitzgibbon, A., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., & Blake, A. (2011). Real-time human pose recognition in parts from single depth images. In: CVPR.

Si, C., Chen, W., Wang, W., Wang, L., &Tan, T. (2019). An attention enhanced graph convolutional LSTM network for skeleton-based action recognition. In: CVPR.

Simonyan, K., &Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. In: NeurIPS.

Soomro, K., Roshan Zamir, A., & Shah, M. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild. CRCV-TR-12-01.

Su, H., Qi, C.R., Li, Y., &Guibas, L.J. (2015). Render for CNN: Viewpoint estimation in images using CNNs trained with rendered 3D model views. In: ICCV.

SURREACT project page. https://www.di.ens.fr/willow/research/surreact/.

Tieleman, T., &Hinton, G. (2012). Lecture 6.5—RMSprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., & Paluri, M. (2015). Learning spatiotemporal features with 3D convolutional networks. In: ICCV.

Tung, H.Y.F., Tung, H.W., Yumer, E., & Fragkiadaki, K. (2017). Self-supervised learning of motion capture. In: NeurIPS.

Varol, G., Ceylan, D., Russell, B., Yang, J., Yumer, E., Laptev, I., Schmid, C. (2018). BodyNet: Volumetric inference of 3D human body shapes. In: ECCV.

Varol, G., Laptev, I., & Schmid, C. (2018). Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 40*(6), 1510–1517.

Varol, G., Romero, J., Martin, X., Mahmood, N., Black, M.J., Laptev, I., & Schmid, C.(2017). Learning from synthetic humans. In: CVPR.

Wang, D., Ouyang, W., Li, W., & Xu, D. (2018). Dividing and aggregating network for multi-view action recognition. In: ECCV.

Wang, J., Nie, X., Xia, Y., Wu, Y., & Zhu, S.C.(2014). Cross-view action modeling, learning, and recognition. In: CVPR.

Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., & Van Gool, L.:(2016) Temporal segment networks: Towards good practices for deep action recognition. In: ECCV.

Weinland, D., Boyer, E., & Ronfard, R. (2007). Action recognition from arbitrary views using 3D exemplars. In: ICCV.

Xie, S., Sun, C., Huang, J., Tu, Z., & Murphy, K. (2017). Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: ECCV.

Yan, S., Xiong, Y., & Lin, D. (2018). Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI.

Yu, F., Zhang, Y., Song, S., Seff, A., & Xiao, J. (2015). LSUN: Construction of a large-scale image dataset using deep learning with humans in the loop. *CoRR*. (**abs/1506.03365**).

Yuille, A.L., Liu, C.: Deep nets: What have they ever done for vision? CoRR **abs/1805.04025** (2018).

Zhang, D., Guo, G., Huang, D., & Han, J. (2018). PoseFlow: A deep motion representation for understanding human behaviors in videos. In: CVPR.

Zhang, P., Lan, C., Xing, J., Zeng, W., Xue, J., & Zheng, N. (2017). View adaptive recurrent neural networks for high performance human action recognition from skeleton data. In: ICCV.

Zheng, J., & Jiang, Z. (2013). Learning view-invariant sparse representations for cross-view action recognition. In: ICCV.

Zheng, J., Jiang, Z., & Chellappa, R. (2016). Cross-view action recognition via transferable dictionary learning. *IEEE Transactions on Image Processing*, *25*(6), 2542–2556.

Zhu, Y., & Newsam, S. (2018). Random temporal skipping for multirate video analysis. In: ACCV.

Zimmermann, C., & Brox, T. (2017). Learning to estimate 3D hand pose from single RGB images. In: ICCV.

Zolfaghari, M., Oliveira, G.L., Sedaghat, N., & Brox, T.(2017). Chained multi-stream networks exploiting pose, motion, and appearance for action classification and detection. In: ICCV.

Zolfaghari, M., Singh, K., & Brox, T. (2018). ECO: efficient convolutional network for online video understanding. In: ECCV.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.