AssemblyHands: Towards Egocentric Activity Understanding via 3D Hand Pose Estimation

Takehiko Ohkawa[1,2], Kun He[1], Fadime Sener[1], Tomas Hodan[1], Luan Tran[1], and Cem Keskin[1]
[1]Meta Reality Labs [2]The University of Tokyo
{tohkawa,kunhe,famesener,tomhodan,tranluan07,cemkeskin}@meta.com

Project page: https://assemblyhands.github.io/

Abstract

*We present **AssemblyHands**, a large-scale benchmark dataset with accurate 3D hand pose annotations, to facilitate the study of egocentric activities with challenging hand-object interactions. The dataset includes synchronized egocentric and exocentric images sampled from the recent Assembly101 dataset, in which participants assemble and disassemble take-apart toys. To obtain high-quality 3D hand pose annotations for the egocentric images, we develop an efficient pipeline, where we use an initial set of manual annotations to train a model to automatically annotate a much larger dataset. Our annotation model uses multi-view feature fusion and an iterative refinement scheme, and achieves an average keypoint error of 4.20 mm, which is 85% lower than the error of the original annotations in Assembly101. AssemblyHands provides 3.0M annotated images, including 490K egocentric images, making it the largest existing benchmark dataset for egocentric 3D hand pose estimation. Using this data, we develop a strong single-view baseline of 3D hand pose estimation from egocentric images. Furthermore, we design a novel action classification task to evaluate predicted 3D hand poses. Our study shows that having higher-quality hand poses directly improves the ability to recognize actions.*

1. Introduction

Recognizing human activities is a decades-old problem in computer vision [17]. With recent advancements in user-assistive augmented reality and virtual reality (AR/VR) systems, there is an increasing demand for recognizing actions from the *egocentric* (first-person) viewpoint. Popular AR/VR headsets such as Microsoft HoloLens, Magic Leap, and Meta Quest are typically equipped with egocentric cameras to capture a user's interactions with the real or virtual world. In these scenarios, the user's hands manipulating objects is a very important modality of interaction. In particular, hand poses (*e.g.*, 3D joint locations) play a central role in understanding and enabling hand-object interaction [3, 18], pose-based action recognition [7, 20, 28], and interactive interfaces [10, 11].

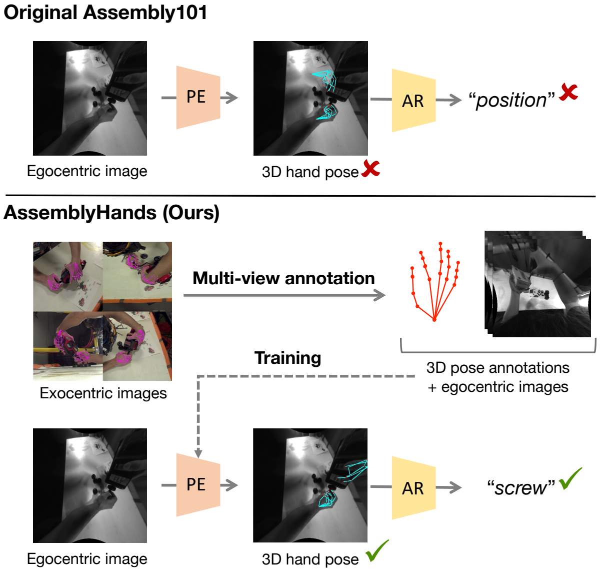


Figure 1. **High-quality 3D hand pose annotations for egocentric activity understanding.** The original Assembly101 [28] recognized actions based on predicted hand poses from the egocentric images; however, suboptimal pose estimation (PE) (*e.g.*, due to the occlusion on the left hand in the top figure) can degrade the performance of action recognition (AR). In contrast, our AssemblyHands benchmark generates 3D hand pose annotations computed from multi-view exocentric images, which are used to train an egocentric pose estimator. As we experimentally demonstrate with an action classification task, having access to more accurate 3D hand pose annotations is critical for better egocentric action recognition.

Recently, several large-scale datasets for understanding egocentric activities have been proposed, such as EPIC-KITCHENS [5], Ego4D [8], and Assembly101 [28]. In particular, Assembly101 highlights the importance of 3D hand

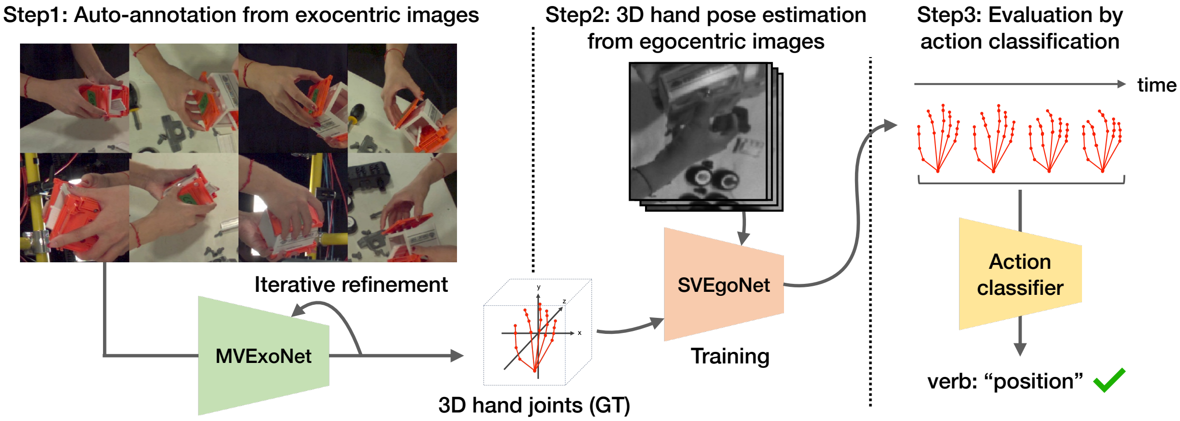

Step1: Auto-annotation from exocentric images
Step2: 3D hand pose estimation from egocentric images
Step3: Evaluation by action classification
time
Iterative refinement
MVExoNet
3D hand joints (GT)
SVEgoNet
Training
Action classifier
verb: "position" ✓

Figure 2. **Construction of AssemblyHands dataset and a benchmark task for egocentric 3D hand pose estimation.** We first use manual annotations and an automatic annotation network (MVExoNet) to generate accurate 3D hand poses for multi-view images sampled from the Assembly101 dataset [28]. These annotations are used to train a single-view 3D hand pose estimation network (SVEgoNet) from egocentric images. Finally, the predicted hand poses are evaluated by the action classification task.

poses in recognizing procedural activities such as assembling toys. 3D hand poses are compact representations, and are highly indicative of actions and even the objects that are interacted with– for example, the "screwing" hand motion is a strong cue for the presence of a screwdriver. Notably, the authors of Assembly101 found that, for classifying assembly actions, learning from 3D hand poses is more effective than solely using video features. However, a drawback of this study is that the 3D hand pose annotations in Assembly101 are not always accurate, as they are computed from an off-the-shelf egocentric hand tracker [11]. We observed that the provided poses are often inaccurate (see Fig. 1), especially when hands are occluded by objects from the egocentric perspective. Thus, the prior work has left us with an unresolved question: *How does the quality of 3D hand poses affect action recognition performance?*

To systematically answer this question, we propose a new benchmark dataset named **AssemblyHands**. It includes a total of 3.0M images sampled from Assembly101, annotated with high-quality 3D hand poses. We not only acquire manual annotations, but also use them to train an accurate automatic annotation model that uses multi-view feature fusion from exocentric (*i.e*., third-person) images; please see Fig. 2 for an illustration. Our model achieves 4.20 mm average keypoint error, which is 85% lower than the original annotations provided in Assembly101. This automatic pipeline enables us to efficiently scale annotations to 490K egocentric images from 34 subjects, making AssemblyHands the largest egocentric hand pose dataset to date, both in terms of scale and subject diversity. Compared to recent hand-object pose datasets, such as DexYCB [3] and H2O [18], our AssemblyHands features significantly more hand-object combinations, as each multi-part toy can be disassembled and assembled at will,

Given the annotated dataset, we first develop a strong baseline for egocentric 3D hand pose estimation, using 2.5D heatmap optimization and hand identity classification. Then, to evaluate the effectiveness of predicted hand poses, we propose a novel evaluation scheme: action classification from hand poses. Unlike prior benchmarks on egocentric hand pose estimation [7, 18, 24], we offer detailed analysis of the quality of 3D hand pose annotation, its influence on the performance of an egocentric pose estimator, and the utility of predicted poses for action classification.

Our contributions are summarized as follows:

- We offer a large-scale benchmark dataset, dubbed AssemblyHands, with 3D hand pose annotations for 3.0M images sampled from the Assembly101 dataset, including 490K egocentric images.
- We propose an automatic annotation pipeline with multi-view feature fusion and iterative refinement, leading to 85% error reduction in the hand pose annotations.
- We define a benchmark task for egocentric 3D hand pose estimation with the evaluation from action classification. We provide a strong single-view baseline that optimizes 2.5D keypoint heatmaps and classifies hand identity. Our results confirm that having high-quality 3D hand poses significantly improves egocentric action recognition performance.

2. Related work

Recognizing actions from pose. The general framework for recognizing people's actions involves extracting low-level states from sensor observations, such as image features or body/hand motion, and then feeding a temporal se-

Dataset	Modality	#img	#ego_img	#views	#subj	Annotation approach
EgoDexter [24]	RGB-D	3K	3K	1 (ego)	4	Manual
Panoptic Studio [30]	RGB	15K	-	31	N/A	2D + triangulation
FPHA [7]	RGB-D	105K	105K	1 (ego)	6	Magnetic sensor
FreiHAND [37]	RGB	37K	-	8	32	Manual + 3D volume + template fitting
HO3D [9]	RGB-D	103K	-	5	10	2D + template fitting
InterHand2.6M [23]	RGB	2.59M	-	80-140	27	Manual + 2D + triangulation
DexYCB [3]	RGB-D	508K	-	8	10	Manual + template fitting
H2O [18]	RGB-D	571K	114K	4 + 1 (ego)	4	2D + template fitting + smoothing
AssemblyHands (M)	RGB/Mono	227K	22K	8 + 4 (ego)	14	Manual + 3D volume + refinement
AssemblyHands (A)		2.81M	468K		20	
AssemblyHands (M + A)		3.03M	490K		34	

Table 1. **Comparison of AssemblyHands with existing 3D hand pose datasets [1].** "M" and "A" stand for manual and automatic annotation, respectively. AssemblyHands is the largest existing benchmark for egocentric 3D hand pose estimation.

quence of states into a recognition model. There is a long history of using full body pose as the state representation in recognizing actions [4, 14, 29, 33, 34], since poses are compact representations that contain discriminative information about actions. Also, in the context of AR/VR, pose information carries the benefit that its availability is less affected by privacy concerns, unlike image/video data. On the modeling side, graph convolutional networks, which treat joints as nodes and bones as edges, have been commonly used in skeleton-based action recognition [20, 35].

In the exocentric setting, action recognition from hand poses is less explored compared to using full body pose, and is only studied on rather small datasets [18]. Instead, hand poses are much more relevant in the egocentric setting. Recently, a large-scale dataset, Assembly101 [28], was proposed to investigate action recognition using 3D hand poses. For Assembly101, 3D hand poses were found to be strong predictors of action; in particular, using hand poses was shown to give higher action classification accuracy compared to using video-based features [19].

Datasets for 3D hand pose estimation. Table 1 shows statistics on existing RGB-based 3D hand pose datasets and our AssemblyHands. Prior works on egocentric hand pose estimation annotate 2D keypoints on a depth image [24] or use magnetic markers attached to hands [7]. Due to the noise from these sensors, as well as the annotation cost, the accuracy and amount of annotation in these benchmarks are not sufficient. Thus, most 3D hand pose estimation works focus on using inputs from static exocentric cameras [3, 9, 12, 23, 30, 31, 36, 37] or utilize such an exocentric dataset to improve egocentric hand pose prediction [26].

Setups with multiple static cameras have several advantages and have been widely used in the literature [25]. First, the total number of available images proportionately increases with the number of cameras. For instance, InterHand2.6M [23] features numerous camera views (80+), resulting in the largest existing hand pose estimation dataset (non-egocentric) with a moderate amount of distinct frames. Second, 3D keypoint coordinates can be reliably annotated from multiple 2D keypoints by using triangulation [23, 30] or hand template fitting [3, 9, 18, 37] (*e.g.*, MANO [27]).

Recently, a few egocentric activity datasets have installed synchronized egocentric cameras along with exocentric cameras, *e.g.*, Assembly101 [28] and H2O [18]. The availability of exocentric images can significantly reduce the amount of annotation effort required for egocentric images. Compared to the H2O dataset, AssemblyHands provides more than four times egocentric images with accurate ground truth and eight times the number of subjects. Due to the goal-oriented nature of assembly actions, the hand poses in our benchmark are totally natural and unscripted.

For automatic annotation, we utilize a volumetric convolution network similar to the one used by Zimmermann *et al.* [37]. We further augment this model with an iterative refinement scheme that does not require additional training.

3. AssemblyHands dataset generation

The input data in our proposed benchmark comes from the recently introduced Assembly101 [28], a large-scale multi-view video dataset designed for understanding procedural activities, in particular, the assembly and disassembly of take-apart toys. It is recorded with a static rig of 8 RGB cameras, plus 4 monochrome cameras on a synchronized headset worn by the human subject.

The initial hand pose annotations for Assembly101 are generated using an off-the-shelf hand tracker specifically designed for monochrome egocentric images [11]. While it can estimate 3D hand poses with reasonable accuracy, there are several limitations. For example, since the stereo area of the egocentric cameras is relatively narrow, depth estimates become inaccurate as hands move further away from the image center. Also, egocentric-only tracking is prone to

[1] We do not include Assembly101 in Table 1 because it was not intended as a 3D hand pose dataset.

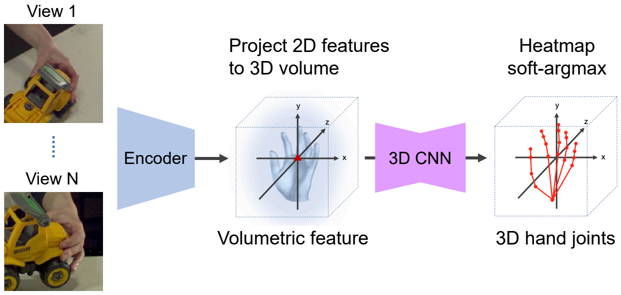


Figure 3. **Architecture of the hand pose annotation model.** We use an EfficientNet encoder [32] to extract 2D features from multi-view images, then aggregate them into a 3D feature volume, and apply volumetric convolution with V2V-Posenet [22]. We apply soft-argmax to extract hand joint locations from 3D heatmaps.

severe failure modes due to heavy occlusion during hand-object interaction. These motivate us to develop a multi-view annotation method using exocentric RGB cameras.

While several existing datasets use off-the-shelf RGB-based models (*e.g.*, OpenPose [2]) to annotate hand poses, we have observed their accuracy is not satisfactory in Assembly101 (see the supplement for details). Since the OpenPose is trained on images with less hand-object occlusions [30], its predictions are often noisy when novel real-world objects (take-apart toys) and higher levels of occlusion are presented in Assembly101. Thus, it is necessary to develop an annotation method tailored to our novel setup.

3.1. Automatic annotation pipeline

We present our proposed automatic annotation pipeline using multi-view exocentric RGB images. We first prepare manual annotation for the frames sampled from the subset of Assembly101 at 1 Hz. Since obtaining manual annotations is laborious, we use them for training an annotation network that can automatically provide reasonable 3D hand pose annotation. We then introduce the detail of our annotation network: (1) an annotation network using volumetric feature fusion (MVExoNet), and (2) iterative refinement during inference of the network. Compared to the manual annotation, this automatic annotation scheme allows us to assign 21 times more labels in another subset of Assembly101 sampled at 30 Hz.

Manual annotation. First, we obtain manual annotations of the 3D locations of 21 joints on both hands in the world coordinate space. We use a setup similar to that of [6, 23], where 2D keypoints are annotated from multiple views and triangulated into 3D. In total, we annotated 62 video sequences from Assembly101 at a sampling rate of 1 Hz, resulting in an annotated set of 22K frames, each having 8 RGB views. We further split it into 54 sequences for training and 8 sequences for testing.

Volumetric annotation network. We next design a neural network model for 3D keypoint annotation. With multi-camera setups, a standard approach is to triangulate 2D keypoint detections; we call this the "2D + Triangulation" baseline. For instance, in InterHand2.6M [23] this approach can achieve an accuracy of 2.78 mm, owing to the high number of cameras (80 to 140). However, for Assembly101, 2D + Triangulation only achieves 7.97 mm given the limited number of 8 RGB cameras (see Table 2). On the other hand, end-to-end "learnable triangulation" methods [1, 16] are known to outperform standard triangulation for human pose estimation in this regime. We thus adopt this principle and design a multi-view hand pose estimation network based on 3D volumetric feature aggregation.

We name our volumetric network MVExoNet, and show its design in Fig. 3. First, a feature encoder extracts 2D keypoint features for each view. We then project the features to a single 3D volume, using the softmax-based weighted average proposed in [16]. Later, an encoder-decoder network based on 3D convolutions refines the volumetric features and outputs 3D heatmaps. We obtain 3D joint coordinates with soft-argmax operation on the heatmaps.

For the architecture, we use EfficientNet [32] as an encoder to extract compact 2D features before volumetric aggregation, in order to save GPU memory. We use V2V-PoseNet [22] as the 3D convolutional network. During training, we generate 2D hand crops by slightly expanding the region enclosing the manually annotated 2D keypoints. The 3D volume is 300 mm long on each side, centered on the bottom of the middle finger (*i.e.*, the third MCP joint). We also augment the volume's root position by adding random noise to each axis, which prevents the model from always predicting the origin of the volume as the third MCP. At test time, we crop hand regions based on the output of a hand detector, and use the predicted third MCP from the 2D + Triangulation baseline as the volume root.

Iterative refinement. During the inference of MVExoNet, we propose a simple iterative refinement heuristic that improves the model's input over several rounds. As mentioned above, MVExoNet requires hand bounding boxes to crop input images and the root position to construct the 3D volume. At test time, the bounding box and volume root come from a hand detector and triangulation of initial 2D keypoint predictions, respectively, which may contain inaccuracies. We found that MVExoNet performs worse than the hypothetical upper bound of having the manually annotated crops and root positions as input.

Our iterative refinement is motivated by this observation: since MVExoNet already generates reasonable predictions, we can use its output to re-initialize the hand crops and volume root position. This gives the network better inputs with each successive round. We call the original model MVExoNet-R1 (the first round of inference), and name the

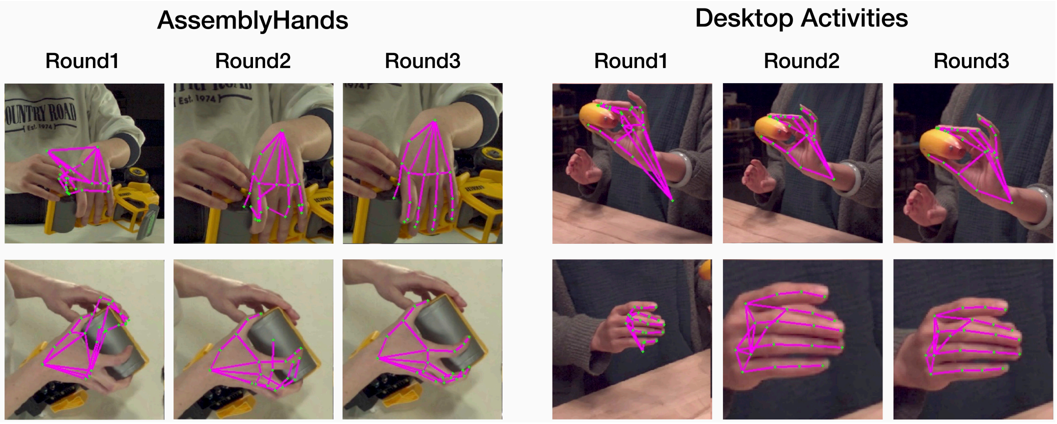


Figure 4. **Example visualization of iterative refinement on AssemblyHands and Desktop Activities [21].** Over the refinement iterations, the cropped image progressively becomes better centered on the hand, and the predicted hand pose becomes more accurate.

Annotation method	MPJPE	PCK-AUC
Egocentric-only [28]	27.55	29.4
2D + Triangulation	7.97	63.8
MVExoNet-R1 (Ours)	5.42	79.2
MVExoNet-R2 (Ours)	4.30	83.1
MVExoNet-R3 (Ours)	**4.20**	**83.4**

Table 2. **Evaluation of hand pose annotation on manually annotated subset of AssemblyHands.** We use MPJPE (mm) and PCK-AUC (%) as the evaluation metrics.

following rounds as MVExoNet-R2, etc. In each additional round, we define input hand crops from projected 2D keypoints generated by the MVExoNetin the previous round, and center the 3D volume on the predicted root position. Note that we freeze MVExoNet during the iterative refinement inference and only update the input (*i.e.*, bounding box and volume root) to the model.

3.2. Evaluation of annotated 3D hand poses

We now compare the accuracy of our proposed annotation method to several baselines, including egocentric hand tracker [11] used in the original Assembly101. First, to evaluate in-distribution generalization, we use the manually annotated test set from Assembly101, which contains frames sampled from 8 sequences at 1 Hz. We also consider the generalization to unseen multi-camera setups; for this purpose, we use the *Desktop Activities* subset from the recently released Aria Pilot Dataset [21].

Comparison to egocentric hand pose annotation. We compare the accuracy of annotation methods on a manually-annotated evaluation set in Table 2. The original hand annotations in Assembly101 [28] are computed by an egocentric hand pose estimator, UmeTrack [11], using monochrome images from egocentric cameras. The egocentric annotation (Egocentric-only) achieved a error of 27.55mm, which is significant higher than methods using exocentric cameras, namely 2D + Triangulation and our proposed method. We found that the annotation from egocentric cameras becomes inaccurate when in-hand objects block the user's perspective. For these cases, the keypoint predictions from multiple exocentric cameras help localize the occluded keypoints. By fusing volumetric features from multi-view exocentric images, our MVExoNet performs much better than the standard 2D + Triangulation baseline.

Ablation study of MVExoNet. As shown in Table 2, our initial inference result (MVExoNet-R1) achieved reasonable performance with 5.42 mm error. The iterative refinement further boosts in reducing annotation errors from 5.42 mm to 4.20 mm (22.5% reduction) after two rounds.

In Fig. 4, we visualize the transition of the hand crops and MVExoNet's predictions over the rounds on both Assembly101 and Desktop Activities. Hand crops in the first round are not optimal for both datasets. For example, the model cannot distinguish which hand to annotate because both hands are centered on the image in Assembly101 (left). Also, the hand moves above in the image (top right) and appears to be tiny (bottom right). Given these suboptimal hand crops, the prediction becomes noisy, such as keypoint predictions going to the other hand and detaching from the hand position. However, in the later rounds, the hand crops gradually focus on the target hand (*e.g.*, left hand on the top left figure), which improves the keypoint localization.

Generalization to novel camera configurations. To evaluate the cross-dataset generalization ability of our annotation method, we use the Desktop Activities dataset, which also features hand-object interactions in a multi-camera setup. It is recorded with a multi-view camera rig similar to that of Assembly101, but with 12 exocentric RGB cameras and different camera placements. The objects are from the YCB

Annotation method	MPJPE	PCK-AUC
2D + Triangulation	49.21	23.9
MVExoNet-R1 (Ours)	21.20	51.3
MVExoNet-R2 (Ours)	14.57	67.2
MVExoNet-R3 (Ours)	**13.38**	**70.4**

Table 3. **Evaluation of multi-view annotation on the Desktop Activities dataset [21].** We use MPJPE (mm) and PCK-AUC (%) as the evaluation metrics.

Subsets	Eval-M	Eval-A	Eval-M+A
Train-M	24.38	28.58	28.35
Train-A	25.18	22.29	22.45
Train-M+A	**23.46**	**21.84**	**21.92**

Table 4. **Effect of automatic annotation for the training of SVEgoNet.** We use egocentric image sets with manual (M), automatic (A), and manual and automatic (M + A) annotation for training and evaluation. We report MPJPE (mm) as the evaluation metric (lower is better).

benchmark [3], which are also unseen in Assembly101. To our knowledge, there are no existing hand pose annotations for Desktop Activities. We use the same manual annotation approach to construct an evaluation set with 1105 annotated frames from three different sequences.

As shown in Table 3, due to the new camera configuration and the presence of novel objects, all methods obtain higher errors than in the Assembly101 setting. In particular, the baseline annotation method 2D + Triangulation degrades significantly when applied to Desktop Activities, to nearly 50 mm MPJPE. In contrast, our MVExoNet is quite robust to the new setting, achieving an initial MPJPE of 21.20 mm, and 13.38 mm after two rounds of iterative refinement (a 36.9% error reduction).

4. Egocentric 3D hand pose estimation

To build hand pose estimators for egocentric views, we train models on egocentric images with annotations generated in Section 3. Training on egocentric images is necessary because existing exocentric datasets do not fully capture egocentric-specific biases in terms of the viewpoint, camera characteristics (egocentric cameras are typically fisheye), and blur from the head motion. Hence, the generalization of exocentric models to egocentric data tends to be limited: for example, in [26], the model trained on DexYCB [3] (exocentric) achieves 14% PCK on FPHA [7] (egocentric), compared to 63% when fine-tuned on FPHA.

We conduct an evaluation of 3D hand pose estimation from egocentric views. Given a single egocentric image, the task aims to predict the 3D coordinates of 21 joints in the wrist-relative space. We split both the manually annotated and the automatically annotated datasets (M/A) into training and evaluation. Manually annotated training and evaluation sets contain 19.2K and 3.0K images, respectively, which are sampled at 1 Hz from 62 video sequences with 14 subjects. Automatically annotated sets include 405K and 63K images, respectively, which are sampled at 30 Hz from a disjoint set of 20 sequences with 20 subjects.

Single-view baseline. Following standard heatmap-based hand pose estimators [15, 23], we build a single-view network (SVEgoNet) trained on monochrome egocentric images. The model consists of 2.5D heatmap optimization and hand identity classification. The 2.5D heatmaps represent 2D keypoint heatmaps in x-y axis and the wrist-relative distance from the camera in z axis. We use the ResNet-50 [13] backbone. The 3D joint coordinates are computed by applying the argmax operation on the 2.5D heatmaps.

In addition, we observe that learning the correlations between hand poses and the identity of hand is effective in our task. For instance, during the "screw" motion, participants in Assembly101 are more likely to hold the toy with their left hand and turn the screwdriver with their right hand. When handling small parts, both hands tend to be closer and appear in the same hand crop. To capture such correlations, we add a hand identity classification branch to SVEgoNet, inspired by [23]. We let the branch classify whether *left*, *right*, or *both* hands appear in a given hand crop.

Evaluation. We compare the predictions from our model and UmeTrack [11] with the ground truth in wrist-relative coordinates. We use two standard metrics: mean per joint position error (MPJPE) in millimeters, and area under curve of percentage of correct keypoints (PCK-AUC).

4.1. Results

Effect of automatic annotation. In Table 4, we compare the performance of SVEgoNet trained on datasets with manual (M), automatic (A), and manual + automatic (M+A) annotations, respectively. We provide Eval-M results as the canonical reference and the other results on all evaluation sets. We observe that using Train-A alone, which is 21 times larger than Train-M, slightly increases error on Eval-M by 3% relative. On the other hand, the model trained on the combined annotations, Train-M+A, consistently gives the lowest error, which validates our efforts in scaling annotations with automatic methods. This study also shows that having a hybrid of manual and automatic annotations is a pragmatic solution to improving the model performance.

Qualitative results. Fig. 5 shows qualitative examples of 3D hand poses generated by UmeTrack [11], our automatic annotation pipeline, and our trained egocentric baseline SVEgoNet. We visualize the prediction of each model from different viewpoints. The egocentric baseline UmeTrack can estimate hand poses reasonably well when seen

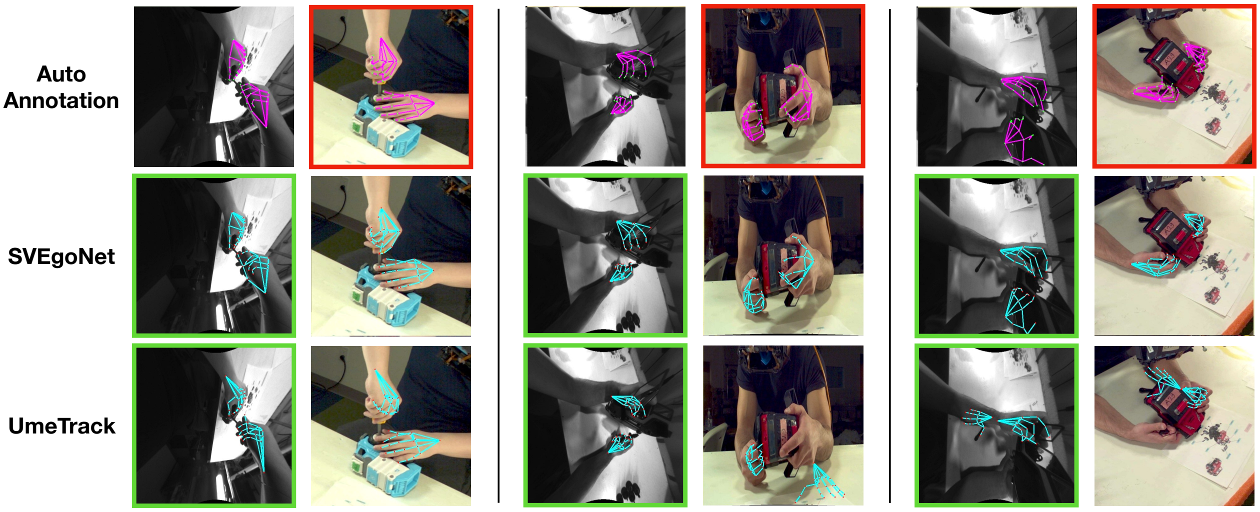


Figure 5. **Qualitative examples of 3D hand poses given by our automatic annotation, SVEgoNet, and UmeTrack [11].** We visualize the 2D projection of 3D poses in one egocentric image and another synchronized exocentric image. We use colored borders to indicate the source images from which hand poses are computed: exocentric (red, additional views omitted) or egocentric (green). The egocentric-based UmeTrack exhibits multiple failure modes, such as inaccurate relative depth prediction of keypoints (left) and entire hand (middle), and completely losing track during occlusion (right). Our multi-view automatic annotation overcomes these failures, resulting in a more robust SVEgoNet when trained on such annotations.

from the egocentric view; however, visualization in exocentric views reveals that it tends to make errors along the z-axis. In particular, the accuracy of the prediction degrades in hard examples with self-occlusion (left example) or hand-object occlusion (middle and right examples). On the other hand, our multi-view automatic annotation overcomes these failures using the cues of multiple exocentric images. Owing to it, the SVEgoNet trained on the annotation achieves more robust results to these occlusion cases.

5. Action classification from 3D hand poses

Finally, we revisit our motivating question: *How does the quality of 3D hand poses affect action recognition performance?* We answer this question with a novel evaluation scheme: verb classification with hand poses as input. In Assembly101 [28], an action is defined at a fine-grained level as the combination of a single verb describing a movement plus an interacting object, *e.g.*, *pick up a screwdriver*. We use six verb labels to evaluate predicted hand poses, including *pick up*, *position*, *screw*, *put down*, *remove*, and *unscrew* (see the left figure in Fig. 6). This is because these verbs heavily depend on the user's hand movements, which hand pose estimation aims to encode. For classifying verbs, we train MS-G3D [20], a graph convolutional network, using the output of egocentric hand pose estimators. Following the experiments of Assembly101, for each segment, we input the sequence of 42 keypoints (21 for each hand). We use the same train/eval split as our automatic annotation, AssemblyHands-A, sampled at a frequency of 30 Hz (*vs*. the original 60 Hz). The model constructs time-series graphs from 3D hand poses and classifies each segment into verbs.

5.1. Results

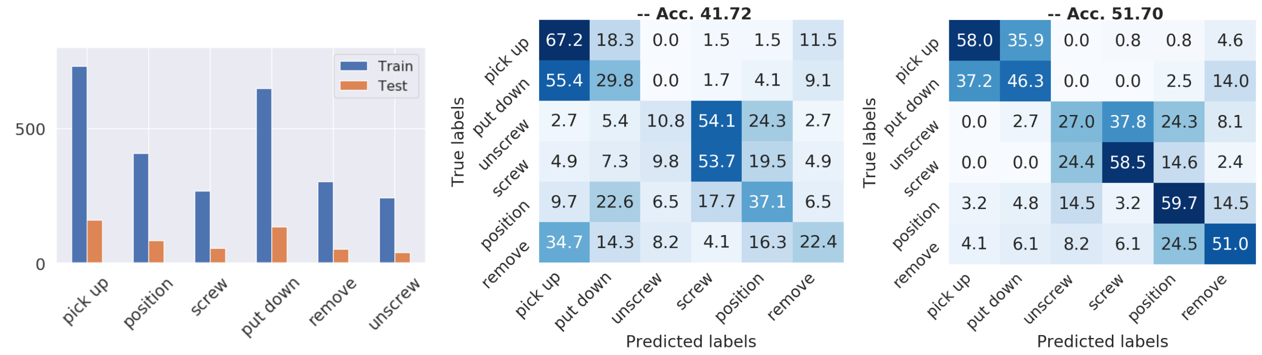
In Table 5, we report the verb classification accuracy given 3D hand poses estimated from the egocentric cameras. First, we establish an empirical upper bound for verb classification accuracy in AssemblyHands-A using the annotated hand poses. We train a verb classifier on our automatic annotations, which achieves 56.5% verb accuracy on average. We note that the lower sampling rate of 10 Hz affects the recognition of rapid nonlinear motions; in particular, the accuracy for *unscrew* is quite low, mainly due to confusion with the *screw* motion.

We then compare our single-view SVEgoNet to the off-the-shelf egocentric hand pose estimator UmeTrack [11], which was used to provide the original annotations for Assembly101, and uses a feature fusion module from multiple egocentric images. First, we report on the pose estimation metric, where SVEgoNet achieves 22.96 mm MPJPE, which is 38% lower than UmeTrack. Next, for verb classification accuracy, using hand poses predicted by SVEgoNet also outperforms using UmeTrack by a large margin (51.7 *vs*. 41.8). When using the upper bound performance of 56.5 as a reference, using SVEgoNet poses attains 91.5% relative performance, which is significantly better than the 73.9% that can be achieved with UmeTrack.

Additionally, we present classification confusion matrices for UmeTrack and SVEgoNet in Fig. 6. Using SVEg-

Method	MPJPE	pick up	position	screw	put down	remove	unscrew	Avg. Verb Acc.
UmeTrack [11]	32.91	**67.2**	37.1	53.7	29.8	22.5	10.8	41.8 (73.9%)
SVEgoNet (Ours)	**21.92**	58.0	**59.7**	**58.5**	**46.3**	**51.0**	**27.0**	**51.7** (91.5%)
AssemblyHands-A	-	63.4	61.3	65.9	59.5	49.0	13.5	56.5 (100%)

Table 5. **Evaluation of action classification from hand poses.** We train and evaluate a MS-G3D [20] action classification model using hand pose sequences as input, and report Verb Accuracy (%). AssemblyHands-A represents the empirical upper bound where automatically annotated hand poses are used as input. Our SVEgoNet predicts more accurate 3D hand poses, which leads to better classification accuracy.



-- Acc. 41.72

	pick up	put down	unscrew	screw	position	remove
pick up	67.2	18.3	0.0	1.5	1.5	11.5
put down	55.4	29.8	0.0	1.7	4.1	9.1
unscrew	2.7	5.4	10.8	54.1	24.3	2.7
screw	4.9	7.3	9.8	53.7	19.5	4.9
position	9.7	22.6	6.5	17.7	37.1	6.5
remove	34.7	14.3	8.2	4.1	16.3	22.4

-- Acc. 51.70

	pick up	put down	unscrew	screw	position	remove
pick up	58.0	35.9	0.0	0.8	0.8	4.6
put down	37.2	46.3	0.0	0.0	2.5	14.0
unscrew	0.0	2.7	27.0	37.8	24.3	8.1
screw	0.0	0.0	24.4	58.5	14.6	2.4
position	3.2	4.8	14.5	3.2	59.7	14.5
remove	4.1	6.1	8.2	6.1	24.5	51.0

Figure 6. **Verb label distribution and confusion matrices of verb classification.** We show the distribution of the six verb labels (left) used in our experiments and confusion matrices of UmeTrack [11] (middle) and our SVEgoNet (right).

oNet predictions significantly reduces the off-diagonal confusions, especially for the challenging verb pairs, *(pick up, put down)* and *(screw, unscrew)*. Measuring the performance individually per verb, SVEgoNet improves the verb accuracy from the UmeTrack by 22%, 5%, 16%, 28%, and 17% for *position*, *screw*, *put down*, *remove* and *unscrew*, respectively, while dropping the accuracy for *pick up* by 5%. For the confusing verb pair *(pick up, put down)*, UmeTrack tends to predict both verbs as *pick up* due to the most frequent verb class. Thus, the accuracy of *put down* is particularly low (29.8%), while the accuracy of 67.2% for *pick up* is slightly higher than SVEgoNet's 58.0%. Notably, our model's improvement on *position* and *remove* verbs is significant because for these verbs, one hand is most of the time heavily occluded, and UmeTrack fails to predict accurate poses for the occluded hands.

The fact that we achieve more than 90% relative performance compared to the upper bound is very encouraging, as SVEgoNet only uses a single egocentric image as input, as opposed to performing complex inference with multi-view exocentric images. This again speaks to the large potential in recognizing activities using lightweight egocentric setups, such as head-mounted monochrome cameras.

6. Conclusion

We present **AssemblyHands**, a novel benchmark dataset for studying egocentric activities in the presence of strong hand-object interactions. We provide accurate 3D hand pose annotations on a large scale, using an automatic annotation method based on multi-view feature aggregation, which far outperforms the egocentric-based annotation from the original Assembly101. The accurate annotations allow us to carry out in-depth analysis of how hand pose estimates inform action recognition. We provide a baseline for single-view egocentric hand pose estimation, and propose a novel evaluation scheme based on verb classification. Our results have confirmed that the quality of 3D hand poses significantly affects verb recognition performance. We hope that AssemblyHands inspires new methods and insights for understanding human activities from the egocentric view.

Limitations and future work. We focus on hand pose annotations and action classification from hand poses. While object cues (*e.g*., object pose) would further benefit the task, its annotation creates a bigger challenge due to the presence of many small object parts in the assembly task. In future work, we first plan to extend hand pose annotation to the entire Assembly101 at higher sampling rates. We also plan to obtain object-level annotation, *e.g*., object bounding boxes. Finally, we are interested in exploring the interplay between hands, objects, and actions with multi-task learning.

Acknowledgments

The authors would like to thank Kevin Harris for help with data collection, and Lingni Ma, Svetoslav Kolev, Bugra Tekin, Edoardo Remelli, Shangchen Han, Robert Wang for helpful discussions.

References

[1] K. Bartol, D. Bojanić, T. Petković, and T. Pribanić. Generalizable human pose triangulation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 4

[2] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 4

[3] Y.-W. Chao, W. Yang, Y. Xiang, P. Molchanov, A. Handa, J. Tremblay, Y. S. Narang, K. Van Wyk, U. Iqbal, S. Birchfield, J. Kautz, and D. Fox. DexYCB: A benchmark for capturing hand grasping of objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9044–9053, 2021. 1, 2, 3, 6

[4] G. Chéron, I. Laptev, and C. Schmid. P-cnn: Pose-based cnn features for action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3218–3226, 2015. 3

[5] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, J. Ma, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, and M. Wray. Rescaling egocentric vision. *International Journal of Computer Vision (IJCV)*, early access, 2021. 1

[6] Q. Feng, K. He, H. Wen, C. Keskin, and Y. Ye. Active learning with pseudo-labels for multi-view 3d pose estimation. *CoRR*, abs/2112.13709, 2021. 4

[7] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. Kim. First-person hand action benchmark with RGB-D videos and 3D hand pose annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 409–419, 2018. 1, 2, 3, 6

[8] K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, M. Martin, T. Nagarajan, I. Radosavovic, S. K. Ramakrishnan, F. Ryan, J. Sharma, M. Wray, M.g Xu, E. Zhongcong Xu, C. Zhao, S. Bansal, D. Batra, V. Cartillier, S. Crane, T. Do, M. Doulaty, A. Erapalli, C. Feichtenhofer, A. Fragomeni, Q. Fu, C. Fuegen, A. Gebreselasie, C. Gonzalez, J. Hillis, X. Huang, Y. Huang, W. Jia, W. Khoo, J. Kolar, S. Kottur, A. Kumar, F. Landini, C. Li, Y. Li, Z. Li, K. Mangalam, R. Modhugu, J. Munro, T. Murrell, T. Nishiyasu, W. Price, P. R. Puentes, M. Ramazanova, L. Sari, K. Somasundaram, A. Southerland, Y. Sugano, R. Tao, M. Vo, Y. Wang, X. Wu, T. Yagi, Y. Zhu, P. Arbelaez, D. Crandall, D. Damen, G. M. Farinella, B. Ghanem, V. K. Ithapu, C. V. Jawahar, H. Joo, K. Kitani, H. Li, R. Newcombe, A. Oliva, H. Soo Park, J. M. Rehg, Y. Sato, J. Shi, M. Z. Shou, A. Torralba, Lo Torresani, M.i Yan, and J. Malik. Ego4D: Around the world in 3, 000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18973–18990, 2022. 1

[9] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit. Honnotate: A method for 3D annotation of hand and object poses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3196–3206, 2020. 3

[10] S. Han, B. Liu, R. Cabezas, C. Twigg, P. Zhang, J. Petkau, T.-H. Yu, C.-J. Tai, M. Akbay, Z. Wang, A. Nitzan, G. Dong, Y. Ye, L. Tao, C. Wan, and R. Wang. MEgATrack: monochrome egocentric articulated hand-tracking for virtual reality. *ACM Transactions on Graphics (ToG)*, 39(4):87–1, 2020. 1

[11] S. Han, P.-C. Wu, Y. Zhang, B. Liu, L. Zhang, Z. Wang, W. Si, P. Zhang, Y. Cai, T. Hodan, R. Cabezas, L. Tran, M. Akbay, T.-H. Yu, C. Keskin, and R. Wang. UmeTrack: Unified multi-view end-to-end hand tracking for VR. *CoRR*, abs/2211.00099, 2022. 1, 2, 3, 5, 6, 7, 8

[12] Y. Hasson, G. Varol, D. Tzionas, I. Kalevatykh, M. J. Black, I. Laptev, and C. Schmid. Learning joint reconstruction of hands and manipulated objects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11807–11816, 2019. 3

[13] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[14] U. Iqbal, M. Garbade, and J. Gall. Pose for action - action for pose. In *In Proceedings of IEEE International Conference on Automatic Face Gesture Recognition*, pages 438–445, 2017. 3

[15] U. Iqbal, P. Molchanov, T. M. Breuel, J. Gall, and J. Kautz. Hand pose estimation via latent 2.5D heatmap regression. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 125–143, 2018. 6

[16] K. Iskakov, E. Burkov, V. Lempitsky, and Y. Malkov. Learnable triangulation of human pose. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7718–7727, 2019. 4

[17] Y. Kong and Y. Fu. Human action recognition and prediction: A survey. *International Journal of Computer Vision*, 130(5):1366–1401, 2022. 1

[18] T. Kwon, B. Tekin, J. Stühmer, F. Bogo, and M. Pollefeys. H2O: two hands manipulating objects for first person interaction recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10118–10128, 2021. 1, 2, 3

[19] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7083–7093, 2019. 3

[20] Z. Liu, H. Zhang, Z. Chen, Z. Wang, and W. Ouyang. Disentangling and unifying graph convolutions for skeleton-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 143–152, 2020. 1, 3, 7, 8

[21] Z. Lv, E. Miller, J. Meissner, L. Pesqueira, C. Sweeney, J. Dong, L. Ma, P. Patel, P. Moulon, K. Somasundaram, O. Parkhi, Y. Zou, N. Raina, S. Saarinen, Y. M. Mansour, P.-K. Huang, Z. Wang, A. Troynikov, R. M. Artal, D. DeTone, D. Barnes, E. Argall, A. Lobanovskiy, D. J. Kim, P. Bouttefroy, J. Straub, J. J. Engel, P. Gupta, M. Yan, R. D. Nardi, and R. Newcombe. Aria pilot dataset. `https://about.facebook.com/realitylabs/projectaria/datasets`, 2022. 5, 6

[22] G. Moon, J. Y. Chang, and K. M. Lee. V2V-PoseNet: Voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5079–5088, 2018. 4

[23] G. Moon, S.-I. Yu, H. Wen, T. Shiratori, and K. M. Lee. InterHand2.6M: A dataset and baseline for 3D interacting hand pose estimation from a single RGB image. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 548–564, 2020. 3, 4, 6

[24] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt. Real-time hand tracking under occlusion from an egocentric RGB-D sensor. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1163–1172, 2017. 2, 3

[25] T. Ohkawa, R. Furuta, and Y. Sato. Efficient annotation and learning for 3d hand pose estimation: A survey. *CoRR*, abs/2206.02257, 2022. 3

[26] T. Ohkawa, Y.-J. Li, Q. Fu, R. Furuta, K. M. Kitani, and Y. Sato. Domain adaptive hand keypoint and pixel localization in the wild. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 68—-87, 2022. 3, 6

[27] J. Romero, D. Tzionas, and M. J. Black. Embodied hands: Modeling and capturing hands and bodies together. *ACM Transactions on Graphics (ToG)*, 36(6):245:1–245:17, 2017. 3

[28] F. Sener, D. Chatterjee, D. Shelepov, K. He, D. Singhania, R. Wang, and A. Yao. Assembly101: A large-scale multi-view video dataset for understanding procedural activities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 21096–21106, 2022. 1, 2, 3, 5, 7

[29] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang. Ntu rgb+ d: A large scale dataset for 3d human activity analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1010–1019, 2016. 3

[30] T. Simon, H. Joo, I. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4645–4653, 2017. 3, 4

[31] S. Sridhar, F. Mueller, M. Zollhoefer, D. Casas, A. Oulasvirta, and C. Theobalt. Real-time joint tracking of a hand manipulating an object from RGB-D input. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 294–310, 2016. 3

[32] M. Tan and Q. V.!Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning (ICML)*, volume 97, pages 6105–6114, 2019. 4

[33] C. Wang, Y. Wang, and A. Yuille. An approach to pose-based action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 915–922, 2013. 3

[34] A. Yao, J. Gall, G. Fanelli, and L. Van Gool. Does human action recognition benefit from pose estimation?". In *Proceedings of the British Machine Vision Conference (BMVC)*. BMV press, 2011. 3

[35] B. Yu, H. Yin, and Z. Zhu. Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting. *arXiv preprint arXiv:1709.04875*, 2017. 3

[36] C. Zimmermann and T. Brox. Learning to estimate 3D hand pose from single RGB images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4913–4921, 2017. 3

[37] C. Zimmermann, D. Ceylan, J. Yang, B. Russell, M. J. Argus, and T. Brox. FreiHAND: A dataset for markerless capture of hand pose and shape from single RGB images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 813–822, 2019. 3