ATTACH Dataset: Annotated Two-Handed Assembly Actions for Human Action Understanding

Dustin Aganian, Benedict Stephan, Markus Eisenbach, Corinna Stretz, and Horst-Michael Gross

***Abstract*— With the emergence of collaborative robots (cobots), human-robot collaboration in industrial manufacturing is coming into focus. For a cobot to act autonomously and as an assistant, it must understand human actions during assembly. To effectively train models for this task, a dataset containing suitable assembly actions in a realistic setting is crucial. For this purpose, we present the ATTACH dataset, which contains 51.6 hours of assembly with 95.2k annotated fine-grained actions monitored by three cameras, which represent potential viewpoints of a cobot. Since in an assembly context workers tend to perform different actions simultaneously with their two hands, we annotated the performed actions for each hand separately. Therefore, in the ATTACH dataset, more than 68% of annotations overlap with other annotations, which is many times more than in related datasets, typically featuring more simplistic assembly tasks. For better generalization with respect to the background of the working area, we did not only record color and depth images, but also used the Azure Kinect body tracking SDK for estimating 3D skeletons of the worker. To create a first baseline, we report the performance of state-of-the-art methods for action recognition as well as action detection on video and skeleton-sequence inputs. The dataset is available at https://www.tu-ilmenau.de/neurob/data-sets-code/attach-dataset.**

I. Introduction

Versatile assembly in small quantities is of high relevance for small and medium-sized enterprises (SMEs). For these, situation-aware cobots are currently a highly relevant research topic [1], [2]. As described in [3], one goal is to achieve situational awareness through general classification and detection algorithms for assembly, since it is not profitable to train new methods for each new manufacturing object. In order to support the situation awareness of cobots, action recognition and action detection of the worker is a suitable tool. However, most of the typical action recognition datasets published so far deal with daily activities [4], [5], [6]. Action datasets that focus on manual activities usually deal with cooking actions [7], [8], [9], [10] and only very recently with assembly actions [11], [12], which are mostly single-label and, thus, only a single action is labeled at any given time for any small-grained action (e.g. *pick up*, *hold*, *screw*).

However, to study situational awareness of assembly actions for cobots, we need a dataset where actions are performed in a natural way and thus can potentially occur simultaneously as the worker can perform an action with each hand. Therefore, we present the novel ATTACH dataset

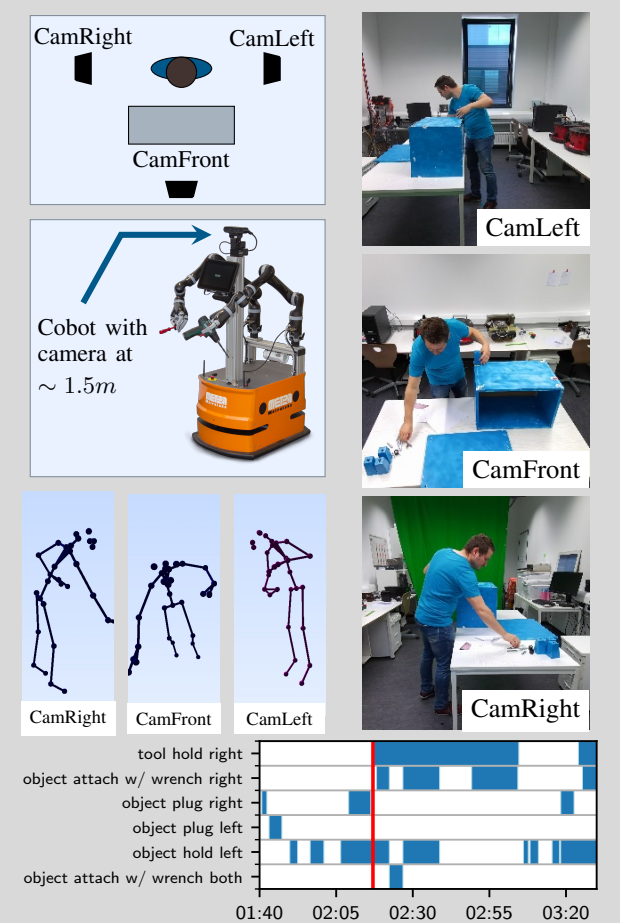

Fig. 1. Setup of the ATTACH dataset (top left), exemplary recorded data, and annotations (bottom). The views of the ATTACH dataset are representative of possible monitoring perspectives of our exemplary cobot. The red line in the annotation diagram (bottom) marks the timestamp at which the above images where recorded. The annotations show which actions were performed before (e.g., *plugging in a leg*) and after (e.g., *holding a wrench*) with each of both hands. It can be seen that several actions temporally overlap, which is the focus of the proposed ATTACH dataset.

(Annotated Two-Handed Assembly Actions for Human Action Understanding) for video- and skeleton-based action recognition and action detection during assembly. For the ATTACH dataset, we asked 42 participants to assemble cabinets consisting of 26 parts, following three different sets of instructions. An overview of the assembly is given in Fig. 1, which illustrates the three viewpoints we recorded. Thus, we have 17.2 hours of recording time per view (51.6 hours in total) and an average duration of 8.2 minutes per recording (with 378 recordings in total). Each fine-grained action (e.g., *picking up a board with the left hand*, *attach an*

This work has received funding from the Carl-Zeiss-Stiftung as part of the project engineering for smart manufacturing (E4SM)

All authors are with Neuroinformatics and Cognitive Robotics Lab, TU Ilmenau, 98693 Ilmenau, Germany dustin.aganian@tu-ilmenau.de

in: IEEE Int. Conf. on Robotics and Automation (ICRA), London, UK, pp. 11367-11373, IEEE 2023, https://doi.org/10.1109/ICRA48891.2023.10160633

object with a screwdriver in the right hand) was annotated, resulting in a total of 95.2k annotations for 51 distinct action classes (with an average of 252 annotations per video). During the recording of the ATTACH dataset, we focused on the following features:

a) Simultaneous fine-grained labels: During assembly most workers often perform different actions simultaneously with their left and right hands, e.g., *picking something up* with one hand and *holding something* with the other hand (see Fig. 1). In contrast to previous single-label assembly action datasets, which do not represent such behavior, we did not restrict the participants to perform different actions with both hands and also labeled all available actions per hand for each time frame, as visualized in the lower part of Fig. 1. Thus, more than 54% of all frames have more than one label describing the actions that occur and more than 68% of annotations overlap with another annotation.

b) Diverse and dynamic assembly actions: In creating the dataset, we took special care to ensure that participants received as few instructions as possible, i.e., they were not given a script to follow, as is often the case [5], [13]. Instead, they received only various written superficial instructions, such as those typically included with furniture for self-assembly. Due to the variety of the parts to be assembled, actions also varied significantly in time. They ranged from a fifth of a second for actions like *lifting an object* or *rotating a workpiece* to a minute or two for actions such as *attaching an object with a wrench*. Furthermore, each participant has a different level of craftsmanship, resulting in a very large variance in length and execution of the various actions.

We benchmark competitive methods on the ATTACH dataset on various tasks. We evaluate action recognition of video clips and 3D-skeleton sequences, and focus on action detection on video and 3D-skeleton input. We also evaluate the action detection task as a real-world robotic application.

Furthermore, evaluating on skeleton input is necessary to evaluate the problem independently of the background. This is important, since it is often not possible to directly record training data for the targeted environment and skeleton-based methods are mostly independent in this regard. Therefore, we also apply an action detection method on skeleton sequences that previously has been applied on video data only.

Summarized, the main contributions of this paper are:

- The publication of the ATTACH dataset, which is the first dataset to independently label each hand and thus include simultaneous fine-grained labels for the assembly action understanding problem.
- The application and evaluation of different baseline methods for the action recognition and action detection tasks for video-based and skeleton-based methods on the ATTACH dataset, to set a first baseline.
- Evaluation of an action detection method for an online robotic application on the ATTACH dataset.

II. Related work

In the following, we present other related datasets for human action understanding.

A. General and daily actions datasets

Action recognition has become increasingly important in recent years, which is reflected in the large number of different datasets on various problems. For instance, typical representatives for general actions as in NTU RGB+D are 60 [15] and 120 [5], which consist of video clips only a few seconds long, or the popular Kinetics dataset [6], which is generated from YouTube videos and also deals with the classification of very short video clips. On the other hand, many datasets are published in the field of household actions, such as DAHLIA [16], Charades [13], Charades-Ego [17], Toyota Smarthome [18] and TSU [4]. Here, the datasets differ strongly in their recorded perspectives (shooting, egocentric, monitoring) and the length variability of their actions (a few seconds in Charades versus a few seconds to several minutes in TSU).

B. Cooking and instruction actions datasets

In contrast to the assembly datasets, which have only recently been published, the datasets for instructions and for cooking [7], [8], [19], [9], [20], [21], [10] have been of research interest for an extended period of time. However, these datasets are very strongly domain-related, and such domain-specific knowledge is not readily transferable.

C. Assembly actions datasets

So far, only very few papers have been published on assembly action datasets. To the best of our knowledge, the only relevant ones are the toy assembly datasets Meccano [14] and Assembly101 [12] and the furniture assembly dataset IKEA ASM [11], which is most related to our dataset. Meccano as well as Assembly101 focus on the fine-motor assembly of toys. Thus, the camera perspectives are focused on the hands and the assembly object. In the case of Meccano, the assembly process was only recorded with an egocentric view. Likewise, action understanding tasks using hand skeletons were a focus of Assembly101. Their camera perspectives make these two datasets partially to not usable at all for cobot applications.

In contrast, we focus on the assembly task with a worker and a cobot who observes the worker and the workspace. Therefore, an egocentric (worker-centric) perspective would not be suitable. Additionally, the process of assembling a cabinet also requires finer movements, e.g., when *screwing in the legs*, making it necessary to perceive both large body movements and smaller hand movements, when solving general action perception on our dataset. When using cobots in such an assembly process, our camera setup also resembles the actual perspective of the robot more closely rather then the egocentric view of the worker.

The IKEA ASM dataset has a similar recording setup to ours. The important difference, however, is that in the IKEA ASM dataset the natural behavior of the workers

in: IEEE Int. Conf. on Robotics and Automation (ICRA), London, UK, pp. 11367-11373, IEEE 2023, https://doi.org/10.1109/ICRA48891.2023.10160633

TABLE I: Comparison of typical action recognition datasets and related action assembly datasets.

Dataset	Publ. Year	Activity	Frames	Videos	Labelled Instances	Classes	Labelled Frames	Overlapping Labels	Partici-pants	Views	Modalities
NTU RGB+D 120 [5]	2019	General	-	114k	114k	120	-	✗	106	3	RGB, D, IR, 3DPose
Kinetics 700-2020 [6]	2020	General YouTube	-	455k	455k	700	-	✗	-	1	RGB
Charades [13]	2016	Daily	-	10k	67k	157	-	✓	267	1	RGB
TSU [4]	2022	Daily	13.8M	536	41k	51	-	✓	18	7	RGB, D, 3DPose
EPIC-KITCHENS-100 [8]	2022	Kitchen	18M	700	90k	4053	71.6%	28.1%	37	1	RGB
Meccano [14]	2021	Toy Assembly	300K	20	9k	61	84.9%	15.8%	20	1	RGB
Assembly101 [12]	2022	Toy Assembly	111M	4.3k	1,014k	1380	81.4%	7.0%	53	12	RGB, 3DHandP
IKEA ASM [11]	2021	Furniture Assembly	3M	371	17k	33	83.8%	✗	48	3	RGB, D, 3DPose
ATTACH	2023	Furniture Assembly	5.6M	378	95k	51	91.3%	68.3%	42	3	RGB, D, IR, 3DPose

and thus simultaneously executed small-grained actions were not taken into account. In contrast, we explicitly labeled with which hand which fine-grained action was performed resulting in simultaneous action labels. Tab. I summarizes the key points of our dataset compared to typical and relevant action datasets, clearly showing how we focused on the issue of simultaneous labels shown as the percentage of overlapping labels.

III. Dataset

In this section, we describe our overall setup for recording our dataset and give a basic overview of the statistics, such as length of annotated actions and dataset size.

A. Setup

The setup for data recording is shown in Fig. 1. A worktable is monitored by three Azure Kinect cameras[1], which capture the frontal, left, and right views of the worker assembling a piece of furniture, resembling typical observation positions of an assistive robot. Each camera is connected to a separate PC and records RGB images with a resolution of 2560×1440 pixels and depth/IR images with a resolution of 320×288 pixels at 30 FPS. Based on the known camera parameters, registered depth images with the same resolution as the RGB images can be calculated. Extrinsic calibration is realized by a cube with ArUco markers [22] placed in the center of the worktable before a recording took place. This position marks the center of the global coordinate system. For all images captured, we recorded their globally synced timestamps, enabling to match corresponding images across views if necessary. For each camera, the Azure Kinect body tracking SDK is employed, which uses the depth and IR images to extract a 3D skeleton of the worker.

B. Data and Annotations

a) Assembly task: In each recording, the furniture to be assembled are IKEA cabinets, each consisting of 26 parts. We created three versions of the assembly instructions, which differed in the order of the assembly steps and the actions to be performed, such as performing certain actions with bare hands or with a tool. Each of the recorded subjects had to assemble the piece of furniture according to the construction manual. As our manuals only consisted of goal-oriented instructions, like in furniture assembly manuals, we did not specify how to achieve the next step. On average, it took the participants 8.2 minutes to assemble the piece of furniture. Overall, we recorded three complete assemblies for all 42 participants from three different viewpoints each resulting in 378 recordings and 51.6 hours of recordings in total.

[1]Technical specification: https://docs.microsoft.com/en-us/azure/kinect-dk/hardware-specification

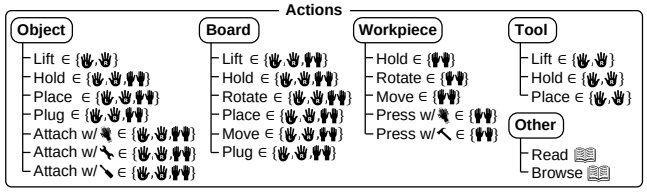

Fig. 2. Action classes used for annotations.

b) Participant statistics: We recorded 42 participants with different level of experience in assembling of which 31 were male and 11 were female. The age of the participants ranged from 21 to 67.

c) Annotations: The recorded data is annotated in detail, as shown in Fig. 2. The type of object on which a particular action is performed is distinguished. This is necessary for follow-up tasks to identify specific assembly steps. We distinguish actions performed on five types of objects: "Object" in Fig. 2 is a small object, such as a screw, that can be enclosed by one hand and of which it is easy to hold multiple instances in one hand. Actions performed on the walls of the cabinet or the like are included in the "board" category. Actions performed on partially assembled furniture are grouped in the "workpiece" category. When the subject uses a tool, the corresponding action is placed in the respective category, unless it is applied directly to a specific object or workpiece. In that case, we annotated it as attaching an object with a specific tool or pressing with a tool. The category "other" contains actions that are performed with the construction manual (e.g. *reading*, *browsing*).

Using this scheme, we get 51 action classes as shown in Fig. 2. For each category, we annotated several actions separately for both hands. This means, that the subject can perform one action with the left hand while simultaneously performing another action with the right hand, e.g., as shown in Fig. 1. This results in more than one label for 54% of all frames and more than 68% of annotations overlapping with another annotation. Overall, the data are annotated with 95.2k annotations, which corresponds to 252 annotations per assembly sequence on average. A histogram of the duration of the performed actions in our dataset can be found in Fig. 3.

in: IEEE Int. Conf. on Robotics and Automation (ICRA), London, UK, pp. 11367-11373, IEEE 2023, https://doi.org/10.1109/ICRA48891.2023.10160633

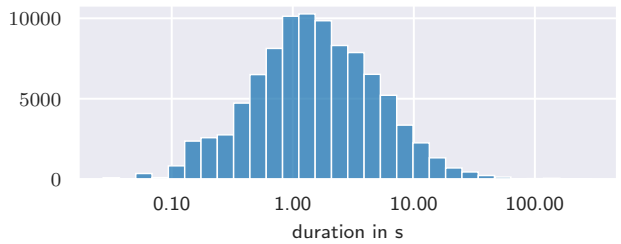

Fig. 3. Histogram of action duration in our dataset plotted on a logarithmic scale. The longest action took 299 seconds ($\sim$ 5 min).

C. Dataset splits

For evaluations on our dataset, we use a person and a view split, similar to other datasets like TSU [4].

Person-split: We split our participants into three groups, with recordings from two-thirds of all participants (28) used for training and the remaining third split into validation (4 participants) and test data (10 participants). Care was taken to ensure that all action classes appear with sufficient frequency in both the test and training splits.

View-split: The camera views shown in Fig. 1 were split as follows: *CamRight* was used as test data, while recordings from *CamFront* and *CamLeft* were used for the training and validation splits. As the views already have a drastically different perception of the scene, we chose not to assign another separate camera for validation. Instead, 10% of all recordings from the front and left camera were assigned for validation. As we will show in our experiments below, splitting the view is a major challenge because the scene looks vastly different from each point of view. Furthermore, in at least one of the views, the person and the action performed are always partially obscured by furniture parts. This split represents the situation of a mobile cobot viewing the scene from a different perspective than those available during training.

IV. Experiments on Action Recognition

In the following, we benchmark state-of-the-art methods on our dataset on the typical action recognition task. Although action detection is more suitable for online robotic applications, this evaluation is still needed because action detectors typically use action recognition methods as a backbone for feature estimation. Thus, we provide first baselines of state-of-the-art methods chosen based on their model architectures (e.g. 3D-CNN vs. transformer) and for our two modalities (video vs. skeleton sequences) respectively.

A. Evaluation protocol

We report the typically used metrics mean class accuracy (mAcc), the top1, and the top5 accuracy on trimmed video clips and trimmed skeleton sequences. We use this evaluation to determine which of the trained networks to use as the backbone for the action detection task (Sec. V).

B. Video-based approaches

a) Setup: As state-of-the-art methods for action recognition on video-sequence inputs, we decided to use TPN [23], a well performing 3D-CNN-based approach, and the novel swin video transformer (Swin) [24]. Implementations for both methods are publicly available[2] and are based on MMAction2 [25].

For the hyperparameter search during training, we used the original values of the respective methods as a starting point. Furthermore, as the actions performed in our dataset differ in length, compared to the datasets the methods were originally trained with a clip length of 16, we also tested the mean action length of 95 and the median action length of 44 frames respectively. For the process of creating clips from segments for training or evaluation, we have strictly adhered to the original implementations and the usual state-of-the-art practices.

b) Results: Fig. 4 shows the results of our baselines on video input. A significant difference between the person and view split is observable. While Swin and TPN perform comparably on the person split, for the view split Swin outperforms TPN which suggests Swin to be able to generalize better across different views. Generalization across people seems to be the easier task as the performance of both methods is around twice as high as on the view split.

For the different clip lengths tested, Swin performed best with the mean clip length of 95 for the person split. Otherwise, the median clip length of 44 was slightly better.

In direct comparison, Swin outperforms TPN (by a large margin on the view split) on our dataset, making it the better choice as a feature extractor for the action detection methods (evaluated in Sec. V).

C. Skeleton-based approaches

a) Setup: For training skeleton-based approaches, we use the 3D skeleton keypoints estimated by the Azure Kinect body tracking SDK which consists of 32 joints. A visualization of these skeletons can be found in Fig. 1.

As state-of-the-art methods for action recognition on skeleton-sequence inputs, we decided in favor of VA-CNN [26] as a 2D-convolution-based (2D-CNN) approach, and ST-GCN [27] as well as AGCN [28] as graph-convolution-based (GCN) methods. We used publicly available code for VA-CNN[3] and MMaction2 [25] for ST-GCN and AGCN.

For the hyperparameter search during training, again, we oriented on the parameterization in the original implementations as a starting point. Likewise, the skeletons were normalized according to each of the applied methods.

For VA-CNN, the complete trimmed skeleton sequences were transformed into a three channel image (x, y, z) and resized to a resolution of 224×224 pixels (keypoints$\times$frames). Thus, no fixed clip length had to be used for training.

For the GCN methods, we need to train with fixed clip lengths, so similarly to the video-based methods, we chose 44 (median action length) and 95 (mean action length). Furthermore, we also train with the clip length of 313, which corresponds to the 95% quantile of action lengths.

[2]https://github.com/SwinTransformer/Video-Swin-Transformer
[3]https://github.com/microsoft/View-Adaptive-Neural-Networks-for-Skeleton-based-Human-Action-Recognition

in: IEEE Int. Conf. on Robotics and Automation (ICRA), London, UK, pp. 11367-11373, IEEE 2023, https://doi.org/10.1109/ICRA48891.2023.10160633

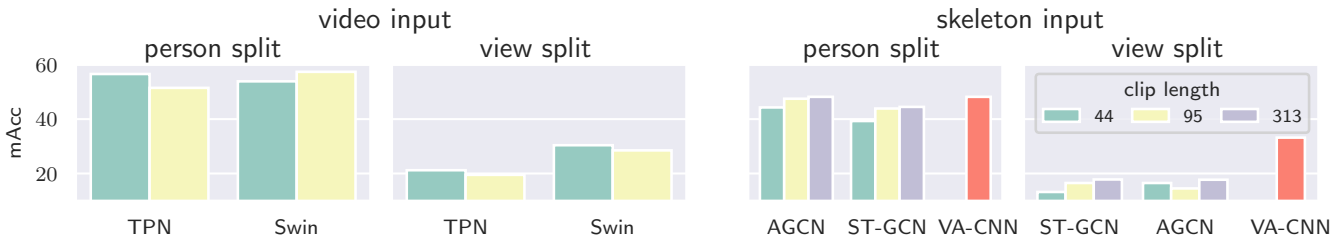

Fig. 4. Baseline results on the action recognition task for different modalities and methods. We only report the mean class accuracy. The top-k metrics for the best models are reported in Tab. II. VA-CNN does not require a fixed number of input frames as the trimmed skeleton-sequence is scaled to size.

TABLE II: Summary of results for action recognition for video- and skeleton-sequence-based inputs. The clip length indicates the best clip length for the respective model and split. mAcc represents mean class accuracy and top1 and top5 represent top-k accuracy.

	model	split	clip length	mAcc	top1	top5
video	Swin	person	95	**57.4**	**61.3**	**92.1**
		view	44	**30.4**	**33.8**	**75.1**
	TPN	person	44	56.6	59.3	90.7
		view	44	21.2	27.0	66.5
skeleton	AGCN	person	313	48.2	55.7	86.9
		view	313	17.7	22.0	55.5
	ST-GCN	person	313	44.5	51.6	84.8
		view	313	17.8	22.6	58.5
	VA-CNN	person	-	**48.2**	**56.5**	**87.2**
		view	-	**33.2**	**43.0**	**76.9**

In addition, 313 is close to the clip length of 300, which is often used for GCNs.

b) Results: The results of our skeleton-based baselines are shown in Fig. 4. Roughly speaking, an increase in clip length results in an increase in recognition performance.

For the person split, VA-CNN and AGCN perform similarly well, with ST-GCN being slightly worse. In contrast, the view split shows a different trend. Here, VA-CNN is clearly superior to GCN methods. This is probably due to the view-adaptive module of VA-CNN, which is supposed to learn a normalization of the skeletal view. However, we also trained the VA-CNN without the view adaptive module on the view split and achieved a mAcc of 22.2%, which is still more than four percentage points better than the best GCN results. This shows, that the examined GCN methods have more difficulties in generalizing a view than the applied CNN method. Thus, VA-CNN is the better choice as a feature extractor for action detection methods on skeletons (evaluated in Sec. V).

When comparing the results of the video-based and the skeleton-based methods, on the person split – as expected – the former perform better. However, the view-split comparison also demonstrates that VA-CNN generalizes even better across views than Swin. This indicates that skeleton-based methods are capable of good generalization. This further highlights the difficulty of the view split and the significant visual differences between the training and test data.

V. Experiments on Action Detection

In action detection, for each frame every occurring class has to be detected. In the following, we begin by presenting our robotic application scenario for action detection. We then briefly describe the chosen method, the evaluation protocol, and the training setup, followed by our experiments for offline and online action detection to get a first baseline on the presented ATTACH dataset.

A. Robotic application scenario

As described in [3], our goal is to use autonomous cobots to assist workers during assembly. In order to achieve this goal, cobots must be able to recognize what is happening live and react to it. Using the ATTACH dataset, we aim to train and evaluate the cobots' environmental awareness regarding action detection. We will first present the experimental results for the typical offline usage, where the complete recording is available during the detection. During offline usage, the detection for a frame is often based not only on past and present frames but also on future frames. Since a cobot is supposed to detect actions live and not after a few minutes, such a detection is not practical for our use case. Therefore, we also do an online evaluation, where only the present and past frames are available for the detection task on each frame.

B. Action detection method

For the action detection task we decided to employ the Pyramid Dilated Attention Network (PDAN) [29] [4] with the chosen recognition methods from experiments in Sec. IV as feature extractors. Unlike many other methods, PDAN is well suited to capture the temporal relationships of both short and long simultaneous actions. This is shown by the state-of-the-art results on the TSU dataset [4], which has simultaneous fine-grained labels for daily activities.

PDAN processes non-overlapping clips of frames of fixed size. Based on these clips, an encoder is used to generate a feature vector for every clip. PDAN then predicts for every frame the occurring actions. In the following experiments, we tested different clip lengths with the minimum clip length being 8 frames. Taking the limitation of our embedded hardware (Jetson Xavier) into account, this minimum length gives enough time to complete the detection on the last clip, before the next clip has to be processed.

PDAN, like many other similar action detectors, has only been used on video-based feature extractors. As shown in our previous experiments, skeleton-based methods tend to generalize better over different views. This is advantageous to our robotic scenario as a cobot can be mobile and therefore has a non-fixed perspective. Thus, we also apply PDAN on the skeleton-based features.

[4]Publicly available code: https://github.com/dairui01/PDAN

in: IEEE Int. Conf. on Robotics and Automation (ICRA), London, UK, pp. 11367-11373, IEEE 2023, https://doi.org/10.1109/ICRA48891.2023.10160633

C. Evaluation protocol

Dai *et al.* [29] evaluated PDAN on the TSU dataset [4] whose varying action duration and overlapping labels make it comparable to our dataset. Therefore, we evaluate the action detection task using the same frame-based mAP evaluation protocol as used in [29]. This performance measure resembles a frame-based accuracy averaged over all action classes, making it suitable for our unbalanced dataset.

D. Setup

As PDAN cannot be applied to video inputs directly, but needs a feature representation generated by an encoder, we first have to choose the models for this purpose. For comparison to the original implementation, we report the detection results when employing the originally used feature extractor I3D [30] as described in [29]. This encoder is trained on clips consisting of 16 frames from the Charades dataset. In addition, we use the best models for both modalities described in Sec. IV, namely the video swin transformer (Swin) and VA-CNN on the respective splits. We like to highlight that this is the first attempt to use PDAN with features that were not extracted from video, but instead from skeleton sequences. Swin and VA-CNN were not only trained on clips consisting of 16 frames and are also directly optimized on our dataset which should enable an improvement in detection quality.

During the hyperparameter search for training, in addition to the typical parameters and the clip size, we also optimized the PDAN-specific model parameters for the temporal reception field.

E. Results for offline usage

The results of our trained action detection models for offline usage are shown in Fig. 5. As can be seen, the difference across both splits is similar to the models trained on the action recognition task, with the view split being more difficult than the person split. An interesting fact is the impact of the clip length on the detection performance. A significant difference can be observed between the video-based and the skeleton-based models, with the latter being more robust with respect to the clip length. This could result from the size of the feature vectors generated from the different encoders. While VA-CNN produces feature vectors of size 2048 for every clip, Swin only computes feature vectors of size 768. As the clip length increases, more information must be encoded in these feature vectors of constant size. For the smaller sized feature vector of Swin, this might explain the bad performance for large clip lengths in PDAN compared to clips consisting of 16 or less frames.

F. Results for online usage

For our robotic-application-specific evaluation we changed the PDAN model architecture so that the temporal receptive field for every frame only consists of present and past frames and no frames from the. Following this change, we repeated our training and hyperparameter search and present our results in Fig. 5. As can be seen, the performance for online

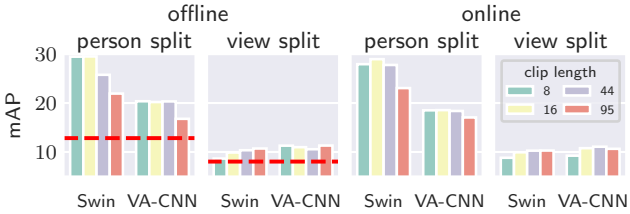

Fig. 5. Results on the action detection task for offline and online usage achieved by PDAN with different backbones. The dashed red line marks results with the I3D encoder trained on Charades. The mAP results of the best models are reported in Tab. III

TABLE III: Results on the action detection task for offline and online usage using PDAN with different encoders and modalities. We report frame-based mAP together with the clip length that achieved the best results.

			offline usage		online usage	
	encoder	split	clip length	mAP	clip length	mAP
video	Swin	person	16	29.5	16	29.0
		view	95	10.7	44	10.3
skeleton	VA-CNN	person	8	20.4	16	18.5
		view	8	11.3	44	11.1

usage of PDAN is comparable to the offline usage and only marginally worse, which is to be expected, as the receptive field now views less relevant frames.

The clip size evaluation is important for the robotic application scenario, because the usage of PDAN with a higher clip size results in longer delay between updates on the current performed actions. Fortunately, this evaluation shows that a smaller clip size, such as 16 (ca. half a second at 30 Hz), achieves very good results compared to the longer clip sizes, for video- and skeleton-based feature extractors.

VI. Conclusion

In this paper, we presented the new ATTACH dataset containing actions performed during assembly. In contrast to existing datasets, we labeled fine-grained actions for each hand individually, resulting in more than 68% of overlapping annotations. Based on our three camera setup, we defined a person and a view split which represent different challenges for action understanding models.

To create a first baseline, we reported results of state-of-the-art methods for both action recognition and detection on our state-of-the-art multi-label assembly dataset, using video and skeleton-sequence inputs respectively. Furthermore, we also evaluated action detection for the cooperative robotic application task of achieving situational awareness for assistance of a worker during assembly.

Future directions: The ATTACH dataset provides a lot of open potential regarding further action understanding tasks. E.g., the estimation of hand poses could provide further information for skeleton-based approaches when trying to perceive actions focused on the fingers, which are only very roughly represented by the Azure Kinect skeletons. For this, our high resolution recordings serve as a good foundation.

To summarize, by providing the ATTACH dataset, we build a foundation for better action perception in the context of assembly tasks, which will contribute to the emerging field of human-robot collaboration.

in: IEEE Int. Conf. on Robotics and Automation (ICRA), London, UK, pp. 11367-11373, IEEE 2023, https://doi.org/10.1109/ICRA48891.2023.10160633

REFERENCES

[1] L. Liu, F. Guo, Z. Zou, and V. G. Duffy, "Application, development and future opportunities of collaborative robots (cobots) in manufacturing: A literature review," *International Journal of Human–Computer Interaction*, pp. 1–18, 2022.

[2] E. Matheson, R. Minto, E. G. Zampieri, M. Faccio, and G. Rosati, "Human–robot collaboration in manufacturing applications: A review," *Robotics*, vol. 8, no. 4, p. 100, 2019.

[3] M. Eisenbach, D. Aganian, M. Köhler, B. Stephan, C. Schröter, and H.-M. Gross, "Visual scene understanding for enabling situation-aware cobots," in *IEEE International Conference on Automation Science and Engineering; 17 (Lyon, France): 2021.08. 23-27*, 2021.

[4] R. Dai, S. Das, S. Sharma, L. Minciullo, L. Garattoni, F. Bremond, and G. Francesca, "Toyota smarthome untrimmed: Real-world untrimmed videos for activity detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[5] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 10, pp. 2684–2701, 2020.

[6] L. Smaira, J. Carreira, E. Noland, E. Clancy, A. Wu, and A. Zisserman, "A short note on the kinetics-700-2020 human action dataset," *arXiv preprint arXiv:2010.10864*, 2020.

[7] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price, *et al.*, "Scaling egocentric vision: The epic-kitchens dataset," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 720–736.

[8] D. Damen, H. Doughty, G. M. Farinella, A. Furnari, E. Kazakos, J. Ma, D. Moltisanti, J. Munro, T. Perrett, W. Price, *et al.*, "Rescaling egocentric vision: collection, pipeline and challenges for epic-kitchens-100," *International Journal of Computer Vision*, vol. 130, no. 1, pp. 33–55, 2022.

[9] M. Rohrbach, S. Amin, M. Andriluka, and B. Schiele, "A database for fine grained activity detection of cooking activities," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 1194–1201.

[10] L. Zhou, C. Xu, and J. J. Corso, "Towards automatic learning of procedures from web instructional videos," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[11] Y. Ben-Shabat, X. Yu, F. Saleh, D. Campbell, C. Rodriguez-Opazo, H. Li, and S. Gould, "The ikea asm dataset: Understanding people assembling furniture through actions, objects and pose," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021, pp. 847–859.

[12] 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)*, June 2022, pp. 21 096–21 106.

[13] G. A. Sigurdsson, G. Varol, X. Wang, A. Farhadi, I. Laptev, and A. Gupta, "Hollywood in homes: Crowdsourcing data collection for activity understanding," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 510–526.

[14] F. Ragusa, A. Furnari, S. Livatino, and G. M. Farinella, "The meccano dataset: Understanding human-object interactions from egocentric videos in an industrial-like domain," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021, pp. 1569–1578.

[15] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[16] G. Vaquette, A. Orcesi, L. Lucat, and C. Achard, "The daily home life activity dataset: a high semantic activity dataset for online recognition," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*. IEEE, 2017, pp. 497–504.

[17] G. A. Sigurdsson, A. Gupta, C. Schmid, A. Farhadi, and K. Alahari, "Charades-ego: A large-scale dataset of paired third and first person videos," *arXiv preprint arXiv:1804.09626*, 2018.

[18] S. Das, R. Dai, M. Koperski, L. Minciullo, L. Garattoni, F. Bremond, and G. Francesca, "Toyota smarthome: Real-world activities of daily living," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[19] A. Miech, D. Zhukov, J.-B. Alayrac, M. Tapaswi, I. Laptev, and J. Sivic, "Howto100m: Learning a text-video embedding by watching hundred million narrated video clips," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2630–2640.

[20] Y. Tang, D. Ding, Y. Rao, Y. Zheng, D. Zhang, L. Zhao, J. Lu, and J. Zhou, "Coin: A large-scale dataset for comprehensive instructional video analysis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1207–1216.

[21] S. Toyer, A. Cherian, T. Han, and S. Gould, "Human pose forecasting via deep markov models," in *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. IEEE, 2017, pp. 1–8.

[22] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognition*, vol. 47, no. 6, pp. 2280–2292, 2014.

[23] C. Yang, Y. Xu, J. Shi, B. Dai, and B. Zhou, "Temporal pyramid network for action recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[24] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2022, pp. 3202–3211.

[25] M. Contributors, "Openmmlab's next generation video understanding toolbox and benchmark," https://github.com/open-mmlab/mmaction2, 2020.

[26] P. Zhang, C. Lan, J. Xing, W. Zeng, J. Xue, and N. Zheng, "View adaptive neural networks for high performance skeleton-based human action recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[27] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[28] L. Shi, Y. Zhang, J. Cheng, and H. Lu, "Two-stream adaptive graph convolutional networks for skeleton-based action recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 026–12 035.

[29] R. Dai, S. Das, L. Minciullo, L. Garattoni, G. Francesca, and F. Bremond, "Pdan: Pyramid dilated attention network for action detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021, pp. 2970–2979.

[30] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

in: IEEE Int. Conf. on Robotics and Automation (ICRA), London, UK, pp. 11367-11373, IEEE 2023, https://doi.org/10.1109/ICRA48891.2023.10160633