





# CoAx: Collaborative Action Dataset for Human Motion Forecasting in an Industrial Workspace

Dimitrios Lagamtzis<sup>1</sup><sup>a</sup>, Fabian Schmidt<sup>1</sup><sup>b</sup>, Jan Seyler<sup>2</sup><sup>c</sup> and Thao Dang<sup>1</sup><sup>d</sup>

<sup>1</sup>Department of Computer Science and Engineering, Esslingen University, Esslingen, Germany

<sup>2</sup>Festo SE & Co. KG, Esslingen, Germany

**Keywords:** Human Robot Collaboration, Industrial Assembly Dataset, Human Motion Forecasting, Action Recognition.

**Abstract:** Human robot collaboration in industrial workspaces where humans perform challenging assembly tasks has become too much; increasingly popular. Now that intention recognition and motion forecasting is being more and more successful in different research fields, we want to transfer that success (and the algorithms making this success possible) to human motion forecasting in an industrial context. Therefore, we present a novel public dataset comprising several industrial assembly tasks, one of which incorporates interaction with a robot. The dataset covers 3 industrial work tasks with robot interaction performed by 6 subjects with 10 repetitions per subject summing up to 1 hour and 58 minutes of video material. We also evaluate the dataset with two baseline methods. One approach is solely velocity-based and the other one is using timeseries classification to infer the future motion of the human worker.


## 1 INTRODUCTION


Collaborative robots (or *cobots* as introduced in (Colligate and Peshkin, 1999)) have received growing interest in both academia and industry over the last decade. While traditional robots usually operate in confined work cells, collaborative robots are not separated from human workers. They allow interaction to solve given tasks in e.g. manufacturing or medical procedures combining cognitive abilities of humans with the repeatability and strength of robot manipulators. Human-Robot-Collaboration (HRC) has the potential to increase efficiency in assembly tasks and lower production costs. A general overview of HRC for manufacturing applications and current research trends is presented in (Matheson et al., 2019).


This article is intended to advance the development of motion and action forecasting methods for assembly tasks with collaborative robots. Motion forecasting is a very active research field (cf. Sec. 2). Its development requires a significant amount of sample data for training and evaluation. While a large body of datasets already exists (as will be discussed in the next section), we find there are still not many sam-


ples available for HRC in industrial assembly tasks. Such tasks involve workpiece components and assembly tools (screwdrivers, soldering irons, ...). Collaborative assembly tasks incorporate actions of humans as well as maneuvers of the robot that modify objects in the workspace to assist the human. To the best of our knowledge, the latter has not been captured in publicly available datasets before. We will provide free access to our dataset<sup>1</sup> of assembly tasks with and without robot interaction. The dataset comprises three tasks recorded by six subjects with ten repetitions per subject. It contains RGBD data, hand and object pose detection results as well as manually defined ground truth action labels that may readily be used for training motion forecasting methods. We also evaluate available hand pose detection methods and compare them using a suitable quality measure. Meaningful performance indices to assess motion forecasting algorithms are proposed (mean and maximum position deviations, and percentage of prediction errors below acceptable bounds). Finally, we have developed two baseline methods, velocity-based motion prediction and time series classification for action recognition and motion prediction, and show benchmark results on our data.

The remainder of this article is organized as follows: Sec. 2 gives an overview of related work on

<sup>a</sup> <https://orcid.org/0000-0003-3342-5083>

<sup>b</sup> <https://orcid.org/0000-0003-3958-8932>

<sup>c</sup> <https://orcid.org/0000-0002-0857-7184>

<sup>d</sup> <https://orcid.org/0000-0001-5505-8953>

<sup>1</sup>Available at: [dkgmtzs.github.io/dataset-coax](https://dkgmtzs.github.io/dataset-coax)

action recognition and forecasting as well as on existing datasets. Sec. 3 describes the exemplary assembly tasks that have been selected for our novel dataset. A summary of object detection algorithms and annotation methods used to generate the data is also given. Sec. 4 proposes quantitative measures to assess the performance of action recognition and forecasting methods and evaluates the performance of two baseline algorithms on the dataset. Sec. 5 concludes the paper and gives an outlook on generating application-driven benchmarks for action forecasting methods.

## 2 RELATED WORK

### 2.1 Existing Datasets

A recent review of existing datasets for human action recognition and motion forecasting is given in (Ji et al., 2020).

As mentioned in (Dreher et al., 2020), datasets generated to research human action recognition problems can be categorized into RGB, RGBD and more complex setups based on their recording modalities. Since we target HRC in an industrial context, we focus on RGBD datasets containing samples of assembly tasks.

(Dallel et al., 2020) compiled a very large dataset illustrating industrial actions that a human is executing in a collaborative workspace. They provided RGB data and 3D information of the human actor’s skeleton. As mentioned earlier, considering that we want to research human action recognition and motion forecasting in 3D environments, it is of great importance to also provide the observations of the objects with which humans interact in a 3D format. Additionally, with a view to minimizing the additional hardware requirements, we intend to use a minimum amount of vision sensors for observing the environment. Furthermore, we believe it is essential to use a perspective in which the human, the industrial workspace in which he or she is working, and the collaborating robot itself are visible, so that all the information required for action recognition and motion forecasting can be derived from it. These requirements are not fulfilled by the dataset provided by (Dallel et al., 2020). (Aksoy et al., 2015) published the popular Maniac dataset. It displays different human tasks and provides RGBD information of the video data. The lack of a suitable camera angle, relevant tasks for an industrial context, and the fact that the provided object labels are often inconsistent within a task, made the dataset unsuitable for our goal.

The Bimanual Actions Dataset (Dreher et al., 2020) provides a dataset of kitchen and work tasks with RGBD modality and derived data, namely object and action labels, to support research in human action recognition. The tasks included in the dataset are very useful to research and evaluate baseline algorithms for human action recognition and motion prediction.

However, we believe HRC for industrial tasks requires a strong focus on assembly activities and robot interaction. In addition, the camera and the angle are from the perspective of a robot that observes the scene. This is one option of capturing the scene, yet we seek that the robot itself is part of the scene and thus must be recorded by the camera. Given that the data format and the proposed pipeline for generating these data are designed simple and adaptable, they serve as a blueprint for the dataset published with this paper.

### 2.2 Motion Forecasting

The overall goal of this research is to predict the future motion and thus the behavior of a human actor in a collaborative environment. Motion forecasting can be performed by simply using the temporal information to predict the future position. Also semantical information like the intention or the action the human is about to perform can be used to derive the future motion. In the following, we will review methods of both categories referred to as *with* and *without* action recognition.

#### 2.2.1 Without Action Recognition

The prediction of human motion without action recognition aims to understand the temporal as well as spatial behavior of a subject based on the observed sequences to generate future body poses. For this sequential task, recurrent neural networks are widely employed, which is related to the success of sequence-to-sequence prediction architectures. Using RNNs to model human motion prediction has become the de facto standard that was initially introduced by (Fragkiadaki et al., 2015) who proposed an Encoder-Recurrent-Decoder model. The ERD model includes nonlinear encoder and decoder networks before and after recurrent layers to extend the basic Long Short Term Memory (LSTM) models to jointly learn representations and their dynamics.

A similar approach in the form of deep RNNs for short-term prediction (< 1s) was established by (Martinez et al., 2017) by using a sequence-to-sequence architecture based on Gated Recurrent Units (GRUs) that is predicting velocities to model future human poses. Although (Martinez et al., 2017) outperformed

(Fragkiadaki et al., 2015), both approaches suffer from discontinuities between the observed poses and the predicted future ones since RNNs struggle to maintain the long-term dependencies needed for forecasting further into the future. Even though RNNs seem to be the first choice for sequential data, (Li et al., 2018) introduced a convolutional sequence-to-sequence that is capable of capturing both spatial and temporal correlations and therefore the invariant and dynamic information of human motion.

Despite the fact that a variety of approaches have been developed for predicting human movements, they are often tailored to specific tasks or motions and therefore not universally applicable. For this reason, (Lasota, 2017) introduced the Multiple-Predictor System, which determines the best parameters for each implemented prediction method directly from observed human motions. The system also determines which combination of these predictors produces the best possible result for a variety of different scenarios.

### 2.2.2 With Action Recognition

Several methods have been proposed in the literature that learn distinct actions or action classes of the human and employ such prototypes to predict the human’s motion. (Perez-D’Arpino and Shah, 2015) and (Zanchettin and Rocco, 2017) utilize Bayesian inference to classify a human’s reaching intention. (Luo and Berenson, 2015) proposed an unsupervised learning approach predicting human motions using Gaussian Mixture Models (GMM) of the arm and palm. Their two-layer framework consists like (Perez-D’Arpino and Shah, 2015) of a motion/action classifier and a motion predictor. In (Luo and Mai, 2019) Probabilistic Dynamic Movement Primitives (PDMP) have been utilized to classify and predict the human’s intention and future motion in a two-stage approach. (Wang et al., 2017) presented a system that trains a CNN and a LSTM to understand human intentions and predict the future human intention.

(Dreher et al., 2020) presented an approach for action segmentation and recognition that learns object-action relations from bimanual human demonstrations using a graph network to process the scene information, which could be interesting for future research work towards motion forecasting.

Regardless of the described approaches to motion forecasting, the focus of this research was to provide a novel dataset for human action and motion forecasting in an industrial context, and to provide a basis for future research using this dataset. Therefore, initially it is sufficient to develop and implement baseline methods for the aforementioned purpose in or-

der to evaluate the dataset. We hope that our dataset will provide a basis for future research and, in particular, for evaluating state-of-the-art approaches to human action recognition and motion forecasting in an industrial setting.

## 3 DATASET GENERATION

Our dataset presents three industrial work tasks designed to outline collaborative work between a human and robot actor. This section starts by describing the hardware and overall system structure used to record our data. The tasks are defined in detail (Sec. 3.2) and finally the processing pipeline used to generate the dataset is outlined (Sec. 3.3).

### 3.1 System Overview

Our physical system consists of a work cell in our laboratory, equipped with a Festo pneumatic collaborative robot arm (Figure 1). Inside the work cell, a human is observed using the vision system performing work tasks in collaboration with the robot. To

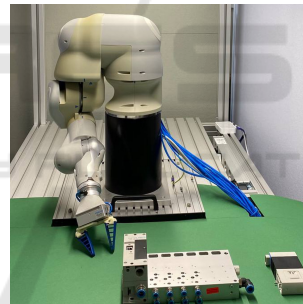


Figure 1: Collaborative robot in work cell.

capture the dataset, an Intel RealSense Depth Camera D435 was used, capturing images at 15 fps with a resolution of 640 px × 480 px. To overlook the space where a human actor and a robot collaborate in solving tasks, the camera was mounted at a height of 0.75m on the work cell relative to the robots base link and tilted downward. The tilt angle is limited by the human pose detection algorithms, which will be discussed later.

In order to represent all objects, including the detected hand pose in the same coordinate system, it was necessary to calibrate the stereo camera used in the setup. Therefore, we used a tool that utilizes ArUco markers in combination with calibration methods from (Tsai and Lenz, 1989). All provided 3D data points are in the robots coordinate system. Our system is based on ROS (Quigley et al., 2009).

## 3.2 Task Description

In this subchapter the data will be characterized in more detail by describing the recorded tasks, the actions executed and the objects interacted with. Every task consists of actions that can be performed by the human and specific objects that need to be interacted with. Due to the importance of collaborative work between human and robot in an industrial workspace, we propose three tasks that have a link to real industrial work tasks: assembly of a valve terminal, assembly of a valve, and soldering a capacitor on an electric circuit board.

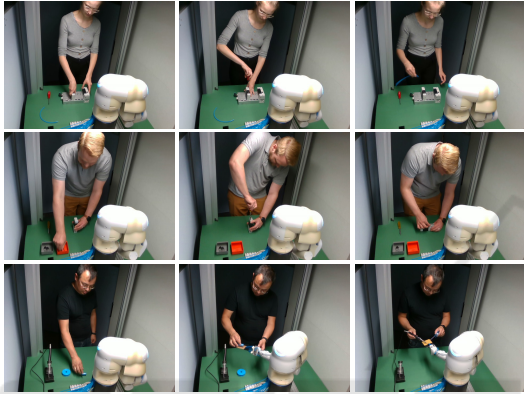


Figure 2: Task overview. Rows: show as follows task 1 to 3. (Detailed description of tasks in 3.2.)

### Task 1: Valve Terminal Plug & Play

In this task a human agent is setting up a valve terminal with a valve. This involves actions like screwing the valve into the valve terminal and also grabbing and installing a hose. The screws are already placed in their position for simplicity reasons. This task is intended to show a typical assembly as it might occur in a collaborative industrial workspace.

### Task 2: Valve Assembly

In this particular task, a valve is assembled from its individual parts. The screws belong to the actual objects of the scene and are located in a designated box. Two screws are used to assemble the main modules of the valve. The last step in the assembly is the attachment of the membrane in its designated place, likewise, stored in a box in the scene.

### Task 3: Collaborative Soldering

The final task explicitly shows a collaboration between the human actor and the robot in the workspace. The robot assists the human in a soldering task by acting as a third arm. Once the robot recognizes the human's intention to solder, it reaches the soldering board, holding it for the human and aligning it so that

the human can solder a capacitor onto it. This task involves the human waiting for the robot in order to continue.

### 3.2.1 Scene Objects

For all tasks, a set of possible objects is defined that can occur in the scene. Possible objects are: *screwdriver*, *hose*, *valve terminal*, *valve*, *box with screws*, *box with membranes*, *soldering station*, *soldering iron*, *soldering tin*, *soldering board* and a *capacitor*. Additional special objects are: the collaborative *robot* with its *end-effector*, the *human* and the *right hand*. The *robot* is described by a 2D and 3D position derived, from the manually labeled bounding box and the momentary position of the motion planner containing the exact position of the robot's base joint. All objects are numbered in sequence.

### 3.2.2 Human Actions

For all tasks, a set of possible actions is defined that can be performed by the human. All possible actions are in general as follows: *approach*, *grab*, *plug*, *join*, *wait for robot*, *screw*, *release*, *solder*, *place*, *reatreat*. This description is adapted from (Dreher et al., 2020). Unlike the aforementioned work, the possible actions in this research include the object being interacted with. This leads to action specification in terms of the given task, action, and object. For instance, the action of a human actor grabbing an object in the workspace is defined as the general action *grab* and the corresponding prominent object specified in the task, which results in the action becoming e.g. *grab screwdriver* for tasks where grabbing occurs. All actions are, as well as the objects, numbered in sequence. So that the action can be fully determined by the 2-tuple combination of action and object id.

## 3.3 Dataset Generation Pipeline

In this subsection, the pipeline for generating the dataset is described in detail. As mentioned in the previous Section 2, one of the most fitting ways of creating a dataset is that from (Dreher et al., 2020). Therefore, it is used as guideline on how the proposed dataset in this paper has been recorded and made available.

The dataset generation pipeline can be divided into two steps. After receiving recorded observations as rosbags, an automated preprocessing of the rosbags is performed. Features like the RGB image, the depth image, the pointcloud as backup information, the 2D and 3D hand pose and finally the robot's 3D base and

end-effector pose are extracted. At this step there are two possible ways to proceed. One can either manually label all scene objects for the entire dataset or use an object detector YOLO (Dreher et al., 2020) or Mask RCNN (He et al., 2018), as this significantly facilitates 3D object segmentation. We decided to manually label all objects for the entire dataset since we would need to label our objects for the detector to function well anyways. The second step in the dataset generation pipeline is therefore the manual labeling of the objects for each frame and the actions for each sequence. Once this manual step is completed, the dataset can be generated. The figure below shows one exemplary frame as a pointcloud with all scene objects 3D box bounded.

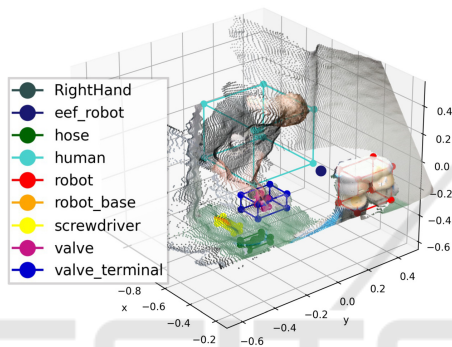


Figure 3: Processed dataset example: Pointcloud with 3D box bounded objects from task 1.

The dataset provided in this research consists of the 3 tasks described in Section 3.2. Each task was recorded 10 times with 6 different human subjects performing. This results to 180 recordings in total, which accumulate to approximately 1h 58min of runtime. The system described in Section 3.1 produces the input for the dataset generation pipeline.

### 3.3.1 Object Labeling

Inferred from (Dreher et al., 2020), 2D bounding boxes have been used to label the captured image data. Since the dataset is generated offline, there are two possible ways of labeling the objects recorded in each video sequence. Either one can manually label the objects or use a trained classification network to generate labels automatically. Training a network requires data or rather object data that has already been labeled and can be used to retrain an existing network, since the objects in the proposed tasks are not within the objects available in any pretrained network. Because no labeled data existed, we decided to label the object data for the whole dataset manually using an annotation tool (Dutta and Zisserman, 2019). Using this labeling tool enables the earlier mentioned way of

training a neural network that can then be used for automatic labeling of the objects. The annotation tool is compatible with annotation formats that can be used to train a Mask R-CNN (He et al., 2018). That means that it is possible to reuse the manually labeled object data in a second step to classify new data when the dataset is extended.

### 3.3.2 Human- and Handpose Detection

For two-dimensional human pose estimation and the associated detection of human joints we focus on the widely used tool OpenPose (Cao et al., 2019) and on the pose and hand detection of Googles framework MediaPipe (Zhang et al., 2020). Since both approaches to human pose estimation are compatible to the software as well as hardware of our system, we performed a comparison of the accuracy of the two detectors. To assess the accuracy, we use an adopted version of the metric Percentage of Correct Keypoints (Simon et al., 2017) that indicates the probability that a detected keypoint  $p$  is within a distance threshold  $t$ , given in pixel, of its true location  $q$ :

$$PCK_t = \frac{1}{n} \sum_{i=1}^n d(p_i, q_i) < t \quad (1)$$

Based on the CMU Panoptic Hand Dataset (Simon et al., 2017), we evaluated both hand detectors using  $PCK_t$  as performance measure by varying the distance threshold  $t$  from 1 to 20 pixels. The results have clearly shown that OpenPose outperforms MediaPipe in terms of quality by about 26%, so we decided to use OpenPose as the method for human pose estimation. To transform the two-dimensional pixel coordinates of the human pose estimation into three-dimensional space, the depth image of the Intel RealSense D435 is used. Experience has shown that the depth image does not hold a valid value for the depth information for every corresponding pixel of the RGB image. As a result, the 3D coordinates cannot be determined for every pixel. To filter invalid depth information, not only a single pixel but also its surrounding pixels are considered and then the median of the available depth information is formed and used to transform the keypoint coordinates.

### 3.3.3 Action Labeling

Regarding the format of the action labels for each task, we refer to (Dreher et al., 2020). The format for the action label consists of the initial frame number of the action, followed by the action label identifier and enclosed by the terminal frame number (initial frame number, action label identifier, terminal frame number). For each task we manually labeled actions

according to these from Section 3.2.2. As already described, the action is a composition of the general action and the specified object that is directly related to the action. Nonetheless one can easily just decode and use the first ID, which is the action, to follow a more general approach in action recognition.

## 4 BASELINE MOTION FORECASTING METHODS

In this section, methods will be discussed that have been implemented and evaluated to set a baseline in motion forecasting for the proposed dataset. An overview is given first, followed by a description of the methods and presentation of experimental results.

### 4.1 Overview of the Methods

For the evaluation of the dataset, two baseline algorithms for motion forecasting have been implemented. As already mentioned in Section 2, we categorize motion forecasting into approaches with and without action recognition and employ one method of each category.

### 4.2 Velocity based Approach

As an approach to motion prediction without action recognition, (Lasota, 2017) introduced the method called Velocity-Based Position Projection, that is based on projecting the current position of the human by estimating its current velocity. To estimate the current velocity, our approach uses the spatial as well as the temporal change of two consecutive human positions. As suggested by (Lasota, 2017), we use the Savitzky-Golay-Filter (Savitzky and Golay, 1964), which basically performs a polynomial regression on the data series, smoothing the human position data. Since there is no universal method for finding the optimal parameters of the filter, we recorded exemplary motions in this regard and determined the optimal set of parameters based on the mean prediction error. Regarding the methods of performance measurement, we used the mean and maximum prediction error, calculated by evaluating the Euclidian distance of a predicted position to its true location, and the Percentage of Correct Predictions  $PCP_t$ . The metric  $PCP_t$  is an adaption of  $PCK_t$ , as it was also used by (Diller et al., 2020), to measure the probability that a predicted keypoint is within a specified distance threshold within the true location. Therefore, Equation 1 still applies, however the Euclidian distance to the

true location is now computed in three-dimensional space and the distance threshold  $t$  is given in meters.

We evaluated the implemented velocity-based approach on our dataset regarding the mentioned performance measures. We used a sliding window approach to capture the smoothed human position data that is then used to produce several sets of predictions of up to 3s in the future, leading to the exemplary task-specific results shown in Table 1.

Table 1: Motion forecasting experiment: Results for exemplary test data sequence of task 1 performed by a specific subject predicting up to 3s into the future. Mean and maximum prediction error are measured in meters. Percentage of Correct Predictions  $PCP_t$  was calculated based on a distance threshold  $t$  of 0.1m.

Time [ms]	Mean Error	Max. Error	$PCP_{0,1}$
500	0.096m	0.385m	0.585
600	0.122m	0.431m	0.527
1200	0.237m	0.958m	0.307
1800	0.339m	1.474m	0.185
2400	0.434m	1.918m	0.126
3000	0.537m	2.383m	0.094

### 4.3 Motion Prototype Approach

The motion prototype approach is an adaption to the time series classification method proposed in (Perez-D’Arpino and Shah, 2015) and (Lasota, 2017). In our version we proceed as follows. We use all hand trajectories of our dataset that belong to one specific task. We then derive the subtrajectories for each action that are received from the action labels for the given task sequence. Subsequently we calculate the mean  $\mu$  and the covariances  $\Sigma$  for every family of trajectories that we concluded for each action, resulting to our motion prototypes. For the alignment of the trajectories, we use an approximate dynamic time warping algorithm (FastDTW) (Salvador and Chan, 2007) to overcome limitations in time and space complexity. In a next step we then predict, for an unseen test data window of size  $\alpha$ , the motion prototype  $x^p$ , namely the action. This is done by computing the log posterior as described in (Perez-D’Arpino and Shah, 2015). When the motion prototype for the window is predicted, we search for a representative point  $x_r^p$ , in the motion prototype by calculating the mean squared error. After finding the representative point  $x_r^p$  we can forecast the future point, based on the time horizon  $t_h$  by computing  $x_{t+t_h} = x_{r+t_h}^p$ . In order to not be susceptible to the fact that objects moved by the human hand can be manipulated in slightly different places from person to person and from task to task, we calculated a position delta between the representative point  $x_r^p$  and the previously predicted point  $x_{r+t_h}^p$  and added it to the

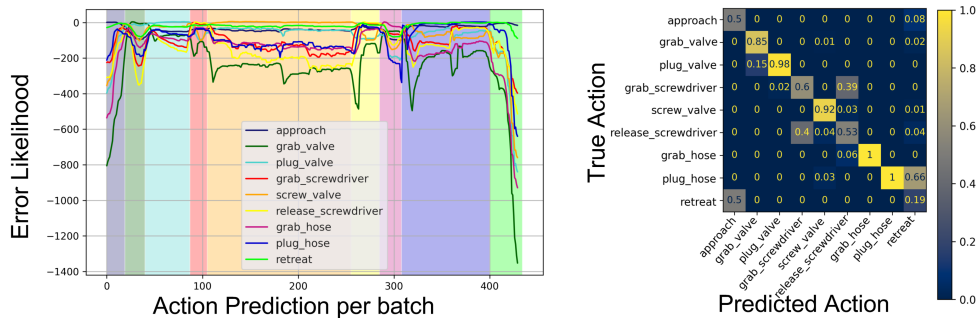


Figure 4: Experiment results with motion prototype approach for task 1. Left figure: Error likelihood for predicted actions. The maximum likelihood translates to the predicted action. The background color highlights the ground truth action for the given sequence. Every prediction was made for every batch. Right figure: Normalized confusion matrix action classification.

last known position of the test window, resulting in the following equation for the prediction.

$$x_{t+t_h} = x_t + (|x_t^p - x_{t+t_h}^p|) \quad (2)$$

Since we know the true point, we can calculate the error as the difference of the true and predicted point, as described in the next paragraphs.

The experiments with the motion prototype approach have been set up as follows. For each task introduced in the dataset, a train test split has been performed such that 45 sequences of the same task performed by different subjects have been used for training of the motion prototypes and 15 for the testing. For the 15 test sequences that have not been respected within the creation of the prototypes, we evaluated the action prediction and the subsequent motion forecasting. Using overlapping sliding windows, we created batches of the size 5 (less than a half second) for the action prediction. The result of the action prediction for each batch was then used to do the motion forecasting. We predicted the future position for a time horizon of 0.5 seconds up to 3 seconds. We evaluated the performance of the predictions by measuring the error for all batches of one sequence. We used the same performance measurements as for the velocity-based approach in Section 4.2.

The results for the action prediction in Figure 4, more precisely on the confusion matrix show clearly that most actions can be predicted accurately. Though actions that can be seen as complements to each other, namely *grab* and *release* or *approach* and *retreat* show a certain behaviour of misclassification leading to a bad prediction. The correlation between these actions, where the associated prototypes have a similar spatial magnitude, could be one reason for this. Regarding the subsequent motion and position forecasting results, the three performance measures for this experiment scored as follows:

The motion prototype approach constrains the time (number of steps) that can be predicted into the

Table 2: Motion forecasting experiment: Results for exemplary test data sequence of task 1 predicting 0.5s up to 3s into the future.

Time [ms]	Mean Error	Max. Error	$PCP_{0,1}$
500	0.136m	0.597m	0.534
600	0.151m	0.622m	0.506
1200	0.208m	0.851m	0.317
1800	0.238m	0.947m	0.254
2400	0.305m	1.105m	0.214
3000	0.327m	1.105m	0.122

future by its prototype length for each motion class. A possible solution to overcome this problem would be to concatenate logically consecutive motion prototypes. However, since this experiment shows only results of our baseline approach with this method, we did not pursue this approach any further and simply accounted the last existing point of the prototype as representative when no other point was left.

Comparing the two baseline methods, it is clear that the motion prototype approach achieves better results, especially at longer time horizons. This is because semantic information, such as the action occurring, is considered in the prediction. Nevertheless, for a very short time horizon, when a velocity-based motion forecasting is sufficient, the results are significantly better. We hope that this baseline can serve as a starting point for comparison in future research.

## 5 CONCLUSION AND OUTLOOK

In this work we present a novel RGBD dataset with direct context to the industry, showing subjects performing different work tasks in a collaborative human robot working environment. Besides the RGBD dataset we provide object labels per frame and action ground truths for the right hand for each sequence. In addition, we have developed a dataset generation pipeline for feature extraction and thus for the generation of the dataset, which can be easily reused to

implement new tasks and generate new data. We envision that our dataset can be a useful basis for training object detection methods and develop motion forecasting and action recognition algorithms in the context of HRC.

We plan to further research algorithms for motion forecasting in the industrial context with the use of the proposed dataset. Therefore, we not only want to use hand motion information to predict the future, but combine it with semantic information. Above all, we want to address problems inherent in time series classification and other approaches, such as limitation to short time horizons and the lack of generalizability with respect to variations in the scene.

## REFERENCES

- Aksoy, E. E., Tamosiunaite, M., and Wörgötter, F. (2015). Model-free incremental learning of the semantics of manipulation actions. *Robotics and Autonomous Systems*, 71:118–133.
- Cao, Z., Hidalgo Martinez, G., Simon, T., Wei, S., and Sheikh, Y. A. (2019). Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.
- Colgate, J. E. and Peshkin, M. A. (1999). Cobots. US Patent No. 5952796A.
- Dallel, M., Havard, V., Baudry, D., and Savatier, X. (2020). InHARD - Industrial Human Action Recognition Dataset in the Context of Industrial Collaborative Robotics. In *2020 IEEE International Conference on Human-Machine Systems (ICHMS)*, pages 1–6.
- Diller, C., Funkhouser, T. A., and Dai, A. (2020). Forecasting characteristic 3d poses of human actions. *CoRR*, abs/2011.15079.
- Dreher, C. R. G., Wächter, M., and Asfour, T. (2020). Learning Object-Action Relations from Bimanual Human Demonstration Using Graph Networks. *IEEE Robotics and Automation Letters*, 5(1):187–194.
- Dutta, A. and Zisserman, A. (2019). The VIA Annotation Software for Images, Audio and Video. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2276–2279, Nice France. ACM.
- Fragkiadaki, K., Levine, S., and Malik, J. (2015). Recurrent network models for kinematic tracking. *CoRR*, abs/1508.00271.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2018). Mask R-CNN. *arXiv:1703.06870 [cs]*.
- Ji, Y., Yang, Y., Shen, F., Shen, H. T., and Li, X. (2020). A Survey of Human Action Analysis in HRI Applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(7):2114–2128.
- Lasota, P. A. (2017). A multiple-predictor approach to human motion prediction. *IEEE International Conference on Robotics and Automation (ICRA)*.
- Li, C., Zhang, Z., Lee, W. S., and Lee, G. H. (2018). Convolutional sequence to sequence model for human dynamics. *CoRR*, abs/1805.00655.
- Luo, R. and Berenson, D. (2015). A framework for unsupervised online human reaching motion recognition and early prediction. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2426–2433.
- Luo, R. and Mai, L.-C. (2019). Human Intention Inference and On-Line Human Hand Motion Prediction for Human-Robot Collaboration. *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*.
- Martinez, J., Black, M. J., and Romero, J. (2017). On human motion prediction using recurrent neural networks. *CoRR*, abs/1705.02445.
- Matheson, E., Minto, R., Zampieri, E. G. G., Faccio, M., and Rosati, G. (2019). Human–Robot Collaboration in Manufacturing Applications: A Review. *Robotics*, 8(4):100.
- Perez-D’Arpino, C. and Shah, J. A. (2015). Fast target prediction of human reaching motion for cooperative human-robot manipulation tasks using time series classification. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6175–6182, Seattle, WA, USA. IEEE.
- Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., and Ng, A. (2009). ROS: an open-source Robot Operating System. page 6.
- Salvador, S. and Chan, P. (2007). FastDTW: Toward Accurate Dynamic Time Warping in Linear Time and Space. page 11.
- Savitzky, A. and Golay, M. J. E. (1964). Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639.
- Simon, T., Joo, H., Matthews, I., and Sheikh, Y. (2017). Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*.
- Tsai, R. and Lenz, R. (1989). A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Transactions on Robotics and Automation*, 5(3):345–358.
- Wang, Z., Wang, B., Liu, H., and Kong, Z. (2017). Recurrent convolutional networks based intention recognition for human-robot collaboration tasks. In *2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1675–1680, Banff, AB. IEEE.
- Zanchettin, A. M. and Rocco, P. (2017). Probabilistic inference of human arm reaching target for effective human-robot collaboration. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 6595–6600, Vancouver, BC. IEEE.
- Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C., and Grundmann, M. (2020). Mediapipe hands: On-device real-time hand tracking. *CoRR*, abs/2006.10214.