

DORMADL - Dataset of human-operated Robot Arm Motion in Activities of Daily Living

Felix Goldau¹, Yashaswini Shivashankar¹, Annalies Baumeister², Lennart Drescher¹, Patrizia Tolle², Udo Frese¹

Abstract—This work presents a dataset of human-operated robot motion to be used within the context of assistive robotics and assorted fields, such as learning from demonstrations, machine-learning based robot control, and activity recognition. The data consists of individual sequences of intentional robot motion performing a task in an environment of daily living. There are 2973 sequences generated in a high-resolution simulation and 986 sequences performed in reality, totaling to 1.16 M datapoints. The data includes labels for the robot's pose, motion and activity. This paper also provides data augmentation methods and a detailed dataset analysis as well as simple models trained on the dataset as a baseline for future research.

The dataset can be downloaded free-of-charge at <https://www.kaggle.com/f371xx/dormadl>.

I. INTRODUCTION

The field of assistive robotics attempts to improve the lives of people who struggle with activities of daily living (ADLs), by using robotic assistance. The intended users often live with physical impairments which restrict their interaction with the environment. Apparently simple tasks, like drinking from a cup, moving a small object from one place to another, or even scratching oneself can become impossible or cumbersome to perform.

Research in this field shows a variety of individual solutions to a lot of tasks and is designed for diverse user groups. The common ground therein is the distinction from the typical application environment of robotics (i.e. industry) and focus on interaction with other people, their homes or private lives. These ADLs refer to the “basic tasks of everyday life, such as eating, bathing, toileting, and transferring” [1] and are well-represented in the literature. The applications range from custom eating utensils for users with spastics [2], simple fetch applications controlled by pointing with a laser pointer [3] or on a touch screen [4] for people with motion-impairments, up to partially autonomous systems to assist people with paraplegia with drinking using their remaining head motion [5] or brain-computer interfaces [6].

The market already provides wheelchair-mounted robotic arms (WMRAs) to be controlled directly by the person sitting in the chair. This creates a great opportunity since it allows for a mobile setup where users can interact with the environment,

*This work was supported by the German Federal Ministry of Education and Research BMBF (Bundesministerium für Bildung und Forschung)-funded projects *DOF-Adaptiv* and *AdaMeKoR* (FKZ 16SV8563, 16SV8534)

¹ The authors are with the German Research Center for Artificial Intelligence (DFKI)

² The authors are with the Frankfurt University of Applied Sciences
Corresponding Author: Felix Goldau, felix.goldau@dfki.de

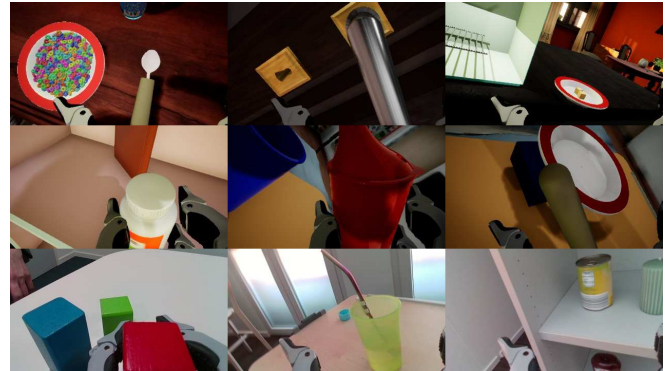


Fig. 1: Example images from the dataset

while also creating challenges, as most places are not designed for a robot arm. In addition to this, the interfaces used to control the robots limit the motion actually possible, as they generally offer less degrees of freedom (DoFs) than the robot is able to perform.

As this is a problem induced by the interface applied, different input modalities have been analyzed, often targeted to specific user groups. These concepts include both physical joystick-alternatives or additional sensors [7], as well as computer-aided control methods such as autonomously switching control modes [8]. More ambitious concepts apply shared control mechanisms, where the user-controlled action, defined by a pre-existing mode, is extended by fusing the result with autonomous solutions [9], possibly also combining this with custom input devices [10].

For modern data-driven approaches, it is necessary to have data representing the desired robot motion to perform a given task, be it for learning or evaluations. This data might also be interesting to the field of activity or intent recognition in order to detect patterns in the users' behavior. However, acquiring this data poses a chicken and egg problem, as the desired motion often cannot be controlled with standard interfaces: Opening a door, for example, requires pulling or pushing the door in an arc, whilst rotating the wrist to keep the alignment with the handle. Given a standard joystick distinguishing between translational and rotational modes, such a motion is simply impossible to perform.

A. Contribution

This work provides a novel dataset of user-controlled robot arm motion in activities of daily living. The main purpose of

the dataset is to learn the user's intended motion given the current situation, i.e. for a situation-adaptive user interface of an assistive robot.

In order to avoid the previously mentioned chicken and egg problem, part of the data is generated with a virtual robot arm controlled by 38 able-bodied participants in a simulation. As the robot arm precisely follows the human hand, no interface-induced motion restrictions apply. The virtual data is padded with a smaller sub-dataset created in our living-lab [11] with a real robot, that is controlled by 4 trained researchers using a 3D-mouse [12].

In short, we contribute a new dataset for assistive robotics of 3 959 recordings (1.16M datapoints), which

- shows realistic and purposeful robot motion in eight simulated and one real scenario,
- provides aligned color and depth images for each datapoint (compare Fig. 1),
- provides poses, velocities, action-labels and the gripper status for machine-learning applications such as activity recognition, AI-based robot control or visual-servoing,
- is preprocessed and ready to use with a provided tensorflow-based [13] dataset loader,
- treats handedness by data-augmentation, and
- is available free-of-charge at
<https://www.kaggle.com/f371xx/dormadl>.

II. STATE OF THE ART

To our knowledge, no dataset exists that includes activity-recognition data of an assistive robot arm or provides sequential pose data of such an arm during the execution of tasks in ADL. Both versions would hold the potential to support the development of shared control algorithms that focus on user intent.

In the field of activity recognition, various ADL-describing sequential datasets are in use: The Human Activity Recognition database [14] consists of recordings of activities such as walking or standing combined with inertial measurement unit (IMU) data of smartphones. The Dataset for ADL Recognition [15] relates wrist-worn accelerometer data to activities such as brushing one's teeth or eating soup. Other datasets have multiple IMUs, be it body worn [16] or partially attached to the environment [17] or use vision [18]. However, all of these are recorded by able-bodied participants and describe relatively broad tasks, where most activities are ADLs themselves.

In a rehabilitation or healthcare setting in particular fall or anomaly detection are interesting. The latter was examined by [19], who published a dataset generated in a simulated smart home environment for that express purpose. [20] shows that these virtual environments, though by far not perfect, are sufficiently realistic enough for neurorehabilitation.

From a more robotic perspective, the community prepared multiple datasets to be used in (assistive or service) robotics: The YCB [21] and YCB-Video [22] datasets link images of objects to their respective 6D poses, with YCB having readily available objects to be used as a benchmark. Knowing the poses of seen objects can be very beneficial for automatic

grasping. The Cornell Grasping Dataset [23] expands on this idea by providing grasp rectangles and point clouds for objects in images. Even more specific, the Columbia Grasp Database [24] combines 3D object models with grasp poses for multiple variations of grippers.

In contrast to the datasets of activity recognition, the robotic image datasets mostly provide single-shot information and not video. The exception to this is the YCB-Video dataset; however, in this, the camera simply pans around the object without following any specific purpose (such as grasping the object). This leads to mostly (semi-) autonomous implementations in research of assistive robotics (e.g. visual servoing [25]), even though users prefer manual control [26].

To fill this gap, this work presents a dataset of detailed robotic arm motion in different ADLs. The dataset provides sequential image data of purposeful interactions during the activities and links these to pose and motion information, as well as human-readable action labels. This allows for applications to react to and analyze realistic situations.

III. RECORDING THE DATASET

The presented dataset is aimed to consist of *purposeful* motion for the current task at hand. We define the robot's motion to be such, if the task completion was successful and the robot hand behaved similarly to an able-bodied person using their own hand, given the robot's workspace restrictions.

In order to achieve the desired quality of motions, the data-generating users were instructed to perform only intentional, deliberate and clear motions during data generation. Each scenario was repeated multiple times per user, with recordings being stopped in between runs to prepare the next setup and a few initial runs to get acquainted with the environment. Thus, in this dataset, a *recording* will refer to a single attempt of a user to perform a task once.

During each recording, object poses, including the individual joints of the robot arm [27] were stored regularly to be later used to calculate camera poses and velocities. In addition to the poses, data from an RGBD camera [28] was gathered. This includes aligned color (RGB) and depth data, as well as unaligned data from infrared cameras used to calculate the depth data. In the simulation, this also included segmentation images.

A. Scenario Selection

In order to create a dataset with situations that are both realistic and relevant for the final users, we followed a participatory approach that included both primary users (i.e. people with disabilities who will actually control the arm) as well as secondary users (e.g. therapists or caregiving relatives who are indirectly affected). To develop these scenarios, we followed the action research model by Margaret Riel [29], [30] and the cycle-based participatory development process *Progressive Problem Solving with Action Research* [31], [32].

In two workshops, five primary users (one of them using a WMRA), two caregiving relatives, two physiotherapists and several researchers collectively discussed different scenarios in an open brainstorming session. The results were further

evaluated with the participants (and one additional primary user) in individual one-to-one interviews to allow for more in-depth feedback. All workshops and interviews were transcribed and qualitatively analyzed following Tuckett's thematic analysis [33]. Protocols and interim results were distributed to everyone involved for heightened transparency and short feedback cycles.

During the workshops, the primary users stated that the most meaningful activities for them were eating and drinking prepared food and beverages. One primary user stated that "actually, eating is one of the major things (...) or rather being independent (of another person) during it".

Another activity that was important to all participants was to open and close doors independently. One primary user wished that they "could open the door, drive through it to eat and drink, in a way that (they) could then simply grab it (themselves)" to which a caregiver responded: "If (they) could really eat with it alone, that would be nice; That would be independence".

Though initially dismissed during the workshops because of constantly available human assistance, the scenarios of pick-up tasks, shopping, or microwave-usage, were later reintroduced in the interviews. Four primary users and two caregivers changed their minds and saw these scenarios as chances to gain more independence and relieve the caregivers. One primary user stated, "I now see it as rather interesting in my case (...). I'm home alone and my cell phone falls on the floor or I need something (...) important that is on the floor".

Further scenarios discussed in the workshops were activities towards one's personal hygiene, e.g. brushing teeth or using a sponge, which users regarded as unsafe or unrealistic.

Finally, four scenarios were decided upon. They are shown in Fig. 2 in reading order, starting at the top left. They are:

- **Eating and Drinking:** A prepared meal (e.g. a bowl of cereal or a set of small pre-cut pieces of food) is on the table. The robot arm grasps a fork or spoon, takes food and brings it to the mouth. For drinking, an open bottle or cup is grasped and brought to the mouth. Optionally, the liquid is poured from one container to the other.
- **Opening and Closing Doors:** The wheelchair is positioned close to the door. The door handle is pressed down with the gripper and the door is opened by pushing or pulling with the WMRA. The wheelchair is driven through the door and the door is closed using the robotic arm.
- **Microwave:** A microwave is placed on a table accessible to the robot arm. The gripper either pulls on the door or presses the button that opens the microwave. A plate with a prepared meal is grasped and placed into the microwave. The arm closes the microwave, activates it and retrieves the plate afterwards.
- **Supermarket Shelf / Pick Up from Floor:** This scenario is inspired by the setting of shopping for groceries. It includes a shelf with various objects, such as pasta packages and cans, on the lower levels and at least one additional object on the floor. The robot arm grasps the objects and places them on a table or in a basket.

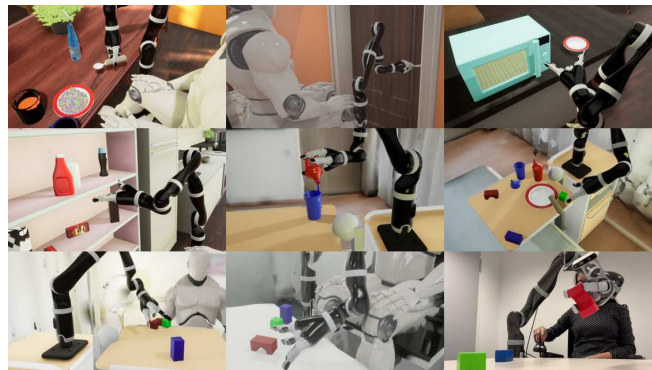


Fig. 2: Overview of the scenarios in the dataset. See media attachment for individual videos

All scenarios were implemented in simulation, with eating and drinking being combined into a single *meal* scenario. In these, the virtual robot arm is mounted to a stationary wheelchair. As the implicit motion of the wheelchair is not part of the dataset, the door scenario does not include the wheelchair moving, but instead only different positions from which to open and close the door.

Another project inspired a scenario where the user is sitting in a bed and the virtual robot arm is mounted to a sideboard with a table attached to one side. In this environment, we added the two scenarios **Fill Cup** and **Cleanup Table**, both based on a photogrammetry-scan of an existing room in our living lab (see [11]).

The *Fill Cup* scenario has two cups on the table with the robot arm grasping one of them to pour water into the other cup, whereas the *Cleanup Table* has various items (e.g. cups, plates, and cutlery) on the table and the robot rearranges them in an orderly fashion. These scenarios were chosen as they support mundane but complex tasks (e.g. pouring water, involving simultaneous rotation and translation), as well as cluttered tasks with various grasps and non-obvious sequential orders (e.g. rearranging objects), thus increasing the difficulty of the dataset.

Finally, we added simple **Block** scenarios for both the bed and wheelchair settings. Here, the robot is used to re-position two blocks to a third block. These simple scenarios work as a baseline but also provide various actions of grasping and reorientation.

B. Recordings in Simulation

In order to create a sufficiently large dataset, part of the data was generated in a virtual reality (VR) simulation environment [34] based on a framework for shared control applications of assistive robots [35]. This includes a virtual version of the same robot arm used in reality. Simulation also allowed to customize the camera's pose or model after the actual recording sessions and automatically assured user anonymity, as only a virtual avatar is rendered.

To record data in the simulation, users were equipped with a VR Headset [36] and motion controllers. Here, they would perceive themselves, depending on the scenario, as either

sitting in a wheelchair with the robot arm attached to its side, or sitting in a bed with the robot mounted to a sideboard. The end effector of the simulated robot arm is connected to the hand-held motion controller, thus enabling the users to basically complete the task using their own hand, only having to adjust for the limiting gripper functionality of the robot. This was hoped to generate human-like but still functional robot motion. The scenarios were developed such that the workspace limitations of the robot would not impede the user.

As this method of control required no initial training of the users, we gathered a variety of people to record data in order to allow different approaches to tasks and variations of motion in the data. An example image created in the simulation can be seen in Fig. 3a.

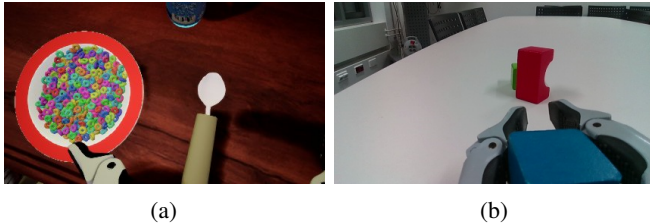


Fig. 3: Example datapoints in simulation (a) and reality (b)

C. Recordings in Reality

For the recordings in reality, an assistive robot [27] was installed to a wheelchair and an RGBD camera [28] was mounted to the last joint of the robot arm. Unlike the setup in the simulation, the real robot cannot simply be moved by following the user's hand, as this would be visible in the image data. Instead, a 3D mouse [12] capable of controlling six DoFs (seven by adding two binary buttons) was used.

Controlling the real robot arm in sufficiently desired motions required training with the 3D mouse and was therefore limited to a selected group. This limited the number of recordings and variety generated in reality. As the setup and implementation of real scenarios takes a lot of time, these were also limited. The real scenarios include versions of Block, Fill Cup (including Drinking with a straw) and Supermarket Shelf / Pick Up from Floor. An example image generated in reality can be seen in Fig. 3b.

D. Data Labeling

As the recording system of the framework automatically stores image-pose pairs of the robot in every frame, no manual labeling of motions is necessary. On a higher level, however, we were able to add activity labels by manually assigning these to time ranges within each recording. As typical activities in the field of activity recognition are relatively broad and rather fit our definition of scenarios, we assign *Actions* instead. These cover shorter ranges of time and are more detailed. Mostly, these consist of a verb defined in reference to an object (e.g. approaching the cup).

Table I lists all components of the actions. Most action-verb combinations exist with only a few exceptions (e.g. the door and handle are an exclusive pair as they are conceptually

connected). The table also lists verbs that only occur with a single object as singular verbs; as well as stand-alone actions without an object.

TABLE I: Overview of action components

verbs	Approach, Grasp, Let go, Push, Retrieve
objects	block, book, bottle, candle, cap, cup, door / handle, food, fork, microwave door, milk carton, plate, spoon, tea
singular verbs	Align [cup], Close [door], Fill [cup], Press [handle], Pull [door]
stand-alone actions	Discard, Drinking, Eating, Idle

As most actions are self-explanatory, we will describe only those with some ambiguity: *Retrieve* moves a held object to another position. For the shelf scenario, a suffix indicates the retrieval to the wheelchair table. *Align cup* positions a cup above another for pouring, which itself is described as *Filling cup*. *Eating* and *Drinking* both move a held item to the mouth, optionally tilting it, and partially retract afterwards.

Discard is a special label referring to sequences with recording issues. If recordings had Discard-labels at the beginning or end, they were trimmed accordingly or not included in the dataset altogether. *Idle* refers to the rest or pull-back motions of the user.

In addition, every action was given a binary success token, allowing for labeling of failed attempts. Unlike discarded-actions, failures do not refer to software issues, but indicate that the user was unable to complete their intended action (e.g. dropping an object).

E. Preprocessing / Dataset Cleaning

The recorded raw data was processed to cleanup the dataset and prepare it for easier use. For this, the initially measured pose data was smoothed and differentiated over time to generate motion information.

We define $T_{a \leftarrow b} := (\vec{p}, \vec{q}, g)$ as the transformation of a robot's coordinate frame b in reference to a and consisting of a 3-DoFs position \vec{p} , 3-DoFs quaternion-orientation \vec{q} , and 1-DoF gripper opening status g . In addition, we define $\vec{v}^b := (\vec{d}, \vec{r}, f)$ to be the relative velocity of the frame b consisting of a translational, rotational, and gripper-velocity respectively.

Let $\hat{T}_{\text{base} \leftarrow \text{EE}}$ be the raw data measured during recording. Due to the rotational component's dependencies, this data has to be treated as a manifold [37]. We utilized a version of the smoother on boxplus-manifolds proposed by [38] to smooth the raw pose data over time, remove outliers and handle data inconsistencies, thus creating a cleaner $T_{\text{base} \leftarrow \text{EE}}$. To further contextualize the pose with the camera data, the pose is transformed to $T_{\text{base} \leftarrow \text{cam}}$ with the camera frame being at the center of the color lens.

The dataset also provides the robot's velocity $\vec{v}^{\text{cam}}(t)$, which is approximated as the relative camera motion per timestep Δt_k , i.e.

$$\vec{v}^{\text{cam}}(t_k) = \frac{\Delta}{\Delta t_k} T_{\text{base} \leftarrow \text{cam}}(t_k) = \frac{T_{\text{cam}(t_{k-1}) \leftarrow \text{cam}(t_{k+1})}}{t_{k+1} - t_{k-1}}. \quad (1)$$

For this, we define the division of a transformation T by a duration s to create the velocity \vec{v} , mostly following vector multiplication, except for orientation which needs to be converted to a rotation vector first, i.e

$$\vec{v} = T/s = (\vec{p}/s, \ln(\vec{q})/s, g/s) = (\vec{d}, \vec{r}, f). \quad (2)$$

The preparation also includes datatype conversions between simulation and reality, outlier detection and removal, and handling of lost frames. In order to retain time-series information, invalid datapoints not at the beginning or end of recordings were kept in the data and marked as such.

1) *Handedness*: One special case of our dataset is handedness. Our robot arm has a non-symmetric 3-fingered gripper that is intended to be used on the right side of the wheelchair. However, a mirrored version of the same arm exists, that is to be mounted on the left side of a wheelchair. In practice, the chosen side generally aligns with the handedness of the user. As both the arm and users were right-handed during data generation, the dataset conforms to this practice in a way. The position of the robot generally affects from which side objects are approached, so data-points from both positions are needed if any machine learning shall work with both.

We propose to computationally augment the dataset with left-handed datapoints by mirroring the whole scene along the central plane of the robot base, which is parallel to the user's central plane. This means flipping the image and velocities relative to the camera around the YZ-plane (true also for an obliquely looking camera) and flipping poses along the central plane of the robot ($p_x = 0$). The specific formulas for the left-handed $T' := (\vec{p}', \vec{q}', g')$ and $\vec{v}' := (\vec{d}', \vec{r}', f')$ are simple but not trivial to derive:

$$\vec{p}' := \begin{pmatrix} -p_x \\ p_y \\ p_z \end{pmatrix}, \quad \vec{q}' := \begin{pmatrix} -q_x \\ q_y \\ q_z \\ -q_w \end{pmatrix}, \quad g' := g, \quad (3)$$

$$\vec{d}' := \begin{pmatrix} -d_x \\ d_y \\ d_z \end{pmatrix}, \quad \vec{r}' := \begin{pmatrix} r_x \\ -r_y \\ -r_z \end{pmatrix}, \quad f' := f. \quad (4)$$

IV. DATASET STRUCTURE

The dataset is split into subsets for training and testing and is structured in two index files with the labels. The image data is stored in a separate directory structure for each recording and referenced by the index files. The data split was performed on a per-user level, such that recordings of a single user are either in the test or training set in order to maintain more independence.

The recordings are processed and sampled at 10Hz to generate clean time series data. All datapoints of the time series are listed in the csv files, with the recording number, a user number, the scenario, a timestamp within the recording, $T_{\text{base} \leftarrow \text{cam}}$, $\vec{v}^{\text{cam}}(t_k)$, the assigned action class and failure tag, the validity-flag, and file paths for the image data.

This multi-dimensional data structure allows for different settings and usages. The features can be a subset of the camera data, consisting of an RGB color image and an aligned

depth image. In the simulation, a segmentation image is also generated. In addition, but uploaded separately¹ in order to reduce storage size, the infrared camera data used to generate the depth images can be loaded. The labels can be either the pose of the gripper or camera, their velocities, or the assigned actions. A python script is provided to assist loading default dataset configurations.

V. DATASET STATISTICS

This section gives a brief statistical overview. All numbers refer to the complete dataset, with the respective numbers for the subsets in brackets as (*training, test*).

The dataset consists of 1.16 M (871 k, 290 k) datapoints from 3959 (3 165, 794) recordings with an average runtime of 29.3 seconds, thus totaling to a length of approximately 29.3 (24.2, 8.1) hours. Thereof, 502 k datapoints from 986 sequences over 13.9 hours were created in reality. An example for the poses of a single recording are shown in Fig. 4.

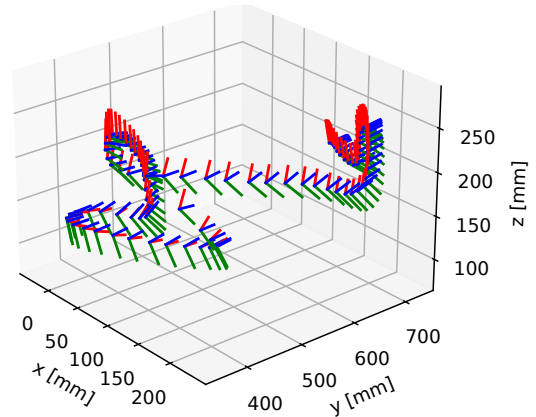


Fig. 4: Camera poses during a single example recording with (red, green, blue) being (right, up, forward)

The mean and a scaled representation of the sampled covariance of camera motion is shown in Fig. 5. The scaling is performed to be able to compare the translational, rotational, and gripper velocities with different units. For this, a rotational velocity of 180 °/s corresponds to a translational velocity of 1000 mm/s, and, respectively, opening the gripper once per second corresponds to 300 mm/s. To enhance visibility, the covariance's components are color-coded based on their absolute ratios to visualize relationships.

	unit	mean																																																																									
Translation X	[mm/s]	-0.564	<table border="1"> <thead> <tr> <th colspan="8">covariance</th> </tr> <tr> <th>X</th> <th>Y</th> <th>Z</th> <th>Rx</th> <th>Ry</th> <th>Rz</th> <th>G</th> <th></th> </tr> </thead> <tbody> <tr> <td>X</td> <td>9315</td> <td>-392</td> <td>2091</td> <td>1534</td> <td>1951</td> <td>5619</td> <td>-147</td> </tr> <tr> <td>Y</td> <td>-392</td> <td>7026</td> <td>528</td> <td>-988</td> <td>-703</td> <td>412</td> <td>66</td> </tr> <tr> <td>Z</td> <td>2091</td> <td>528</td> <td>17167</td> <td>5367</td> <td>554</td> <td>-280</td> <td>323</td> </tr> <tr> <td>R_x</td> <td>1534</td> <td>-988</td> <td>5367</td> <td>5288</td> <td>57</td> <td>-1105</td> <td>83</td> </tr> <tr> <td>R_y</td> <td>1951</td> <td>-703</td> <td>554</td> <td>-57</td> <td>13232</td> <td>1217</td> <td>120</td> </tr> <tr> <td>R_z</td> <td>5619</td> <td>412</td> <td>-280</td> <td>1105</td> <td>1217</td> <td>20442</td> <td>78</td> </tr> <tr> <td>G</td> <td>-147</td> <td>66</td> <td>323</td> <td>83</td> <td>120</td> <td>78</td> <td>4788</td> </tr> </tbody> </table>	covariance								X	Y	Z	Rx	Ry	Rz	G		X	9315	-392	2091	1534	1951	5619	-147	Y	-392	7026	528	-988	-703	412	66	Z	2091	528	17167	5367	554	-280	323	R _x	1534	-988	5367	5288	57	-1105	83	R _y	1951	-703	554	-57	13232	1217	120	R _z	5619	412	-280	1105	1217	20442	78	G	-147	66	323	83	120	78	4788
covariance																																																																											
X	Y	Z		Rx	Ry	Rz	G																																																																				
X	9315	-392		2091	1534	1951	5619	-147																																																																			
Y	-392	7026		528	-988	-703	412	66																																																																			
Z	2091	528		17167	5367	554	-280	323																																																																			
R _x	1534	-988		5367	5288	57	-1105	83																																																																			
R _y	1951	-703		554	-57	13232	1217	120																																																																			
R _z	5619	412	-280	1105	1217	20442	78																																																																				
G	-147	66	323	83	120	78	4788																																																																				
Translation Y	[mm/s]	5.629																																																																									
Translation Z	[mm/s]	4.686																																																																									
Rotation X	[°/s]	-0.177																																																																									
Rotation Y	[°/s]	0.131																																																																									
Rotation Z	[°/s]	0.331																																																																									
Gripper G	[1/s]	0.010																																																																									

Fig. 5: Mean and scaled covariance of camera motion

Fig. 6 shows the distribution of the labeled actions over the complete dataset. It can be clearly seen that objects are

¹refer <https://www.kaggle.com/f371xx/dormadl>

represented more often which either have dedicated scenarios (block, cup, microwave door) or are reoccurring (plate, cup, bottle). The verbs show a focus on approaching and retrieving, as well as opening and closing of the gripper. This is expected as it is at the core of robot manipulation.

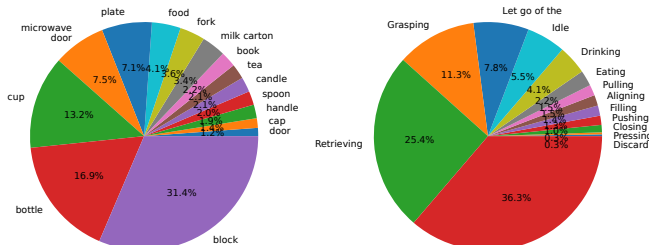


Fig. 6: Distribution of action labels in the dataset

VI. BASELINE MODEL STATISTICS

As a baseline for comparison, we trained two simple networks on the dataset: One for activity (here action) recognition and one to predict the camera’s motion direction for direct robot control, robot servoing, or approaches of shared control. Both can be thought of as predicting the user’s *intent* from the current situation.

A. Baseline Action Recognition Model

MobileNetV2 [39] is used as a base model to predict the robot’s action. The model backbone is extended with a 2D convolution layer with ReLU activation function, global average 2D-pooling, as well as a dropout and a dense layer with a softmax activation function. The model is trained to classify the images into one of the actions performed by the robot arm. A categorical cross-entropy loss function was used during the training. Table II shows the baseline model’s results for both the test and training datasets for the loss and an accuracy metric.

TABLE II: Baseline results of action recognition model

	train	test
categorical cross-entropy	0.8554	0.9223
categorical accuracy	0.7066	0.6796

B. Baseline Motion Direction Prediction Model

For the motion direction prediction, we implemented a baseline model able to output a multi-dimensional Gaussian distribution of the motion, which allows for the use of statistical tools. Even if a single output is required, one can simply take the most likely value of the distribution.

The model is designed with a MobileNetV2 backbone [39] extended with a 2D-convolutional layer with a batch normalization, ReLU activation, as well as three fully-connected layers, the first two of which with a ReLU activation. For the final activation, a layer calculating a sample-based covariance (compare [40]) was used to generate the probability distribution. As this is intended as a simple baseline model, no further extensions, such as recurrences, were added. The only

preprocessing was dimensional scaling to align translational, rotational, and gripper velocity (Section V). The model was trained using a Mahalanobis-loss [40].

To provide more intuitive values than the Mahalanobis-loss, we propose a new metric: Let $b = (\vec{b}_1, \dots, \vec{b}_n)$ be the n -dimensional base spanned by the covariance’s normalized principal components \vec{b}_i (i.e. its eigenvectors), where \vec{b}_1 has the largest corresponding eigenvalue and \vec{b}_n the smallest. We can now calculate the projection \vec{p} of our labeled vector \vec{v} onto a k -dimensional sub-base $s = (\vec{b}_1, \dots, \vec{b}_k)$

$$\vec{p} = \sum_{i=1}^k \langle \vec{v} \cdot \vec{b}_i \rangle \cdot \vec{b}_i, \tag{5}$$

where $\langle \cdot \rangle$ is the scalar product of two vectors and $k < n$. Together with the projection \vec{q} on the complementing sub-basis $b = (\vec{b}_{k+1}, \dots, \vec{b}_n)$, \vec{p} and \vec{v} form a right triangle such that pythagorean theorem yields $|\vec{v}|^2 = |\vec{p}|^2 + |\vec{q}|^2$. This allows us to define $|\vec{p}|^2 / |\vec{v}|^2$ as a metric: the percentage of the squared length of true motion that is represented by the k first principal components of the probability distribution. Note that the first k principal components maximize this metric among all choices of $\vec{b}_1, \dots, \vec{b}_n$. This can also be seen as how well one could follow the true motion, while only moving along the axes $\vec{b}_1, \dots, \vec{b}_k$.

Within a direction-prediction task, the percentage-of-motion-metric is to a Mahalanobis- or log-likelihood-loss what the accuracy-metric is to a cross-entropy-loss in a classification task: Both metrics reinterpret the evaluation to a more human-readable form by simplifying (reducing) the data. Here, accuracy assesses only the binary equivalence of the label and the most likely class while ignoring the actual probability value. Similarly, the percentage-of-motion-metric assesses only the projection of the true direction in the the sub-basis s while ignoring the remaining $n - k$ dimensions.

Table III shows the baseline model’s results for the Mahalanobis-loss distance, negative log-likelihood and the percentage of motion metrics for $k = 1$ and $k = 2$, as well as the root-mean-squared-error between the label and the first principal component.

TABLE III: Baseline results of motion prediction model

	train	test
Mahalanobis-distance	4.6991	5.5219
negative log-likelihood	-11.1994	-10.4222
percentage of motion ($k = 1$)	0.5844	0.4975
percentage of motion ($k = 2$)	0.7538	0.6741
root mean squared error	0.5229	0.5268

VII. LIMITATIONS

The quality of the dataset is limited by two main factors: Issues in the methodology and the simulation reality gap. Even though the simulation is designed to be very close to reality, there are some aspects of robot interactions in reality that were not implemented for various reasons.

A. Methodology

As with typical learning-from-demonstration applications, the dataset can only be assumed to provide accurate information for situations similar to those recorded.

Apart from that, our method of robot control is based on the assumption that users control the robot arm similarly to a regular arm. This assumption might not be correct and therefore make the data partially invalid. The data generated in reality should improve this.

Only a small portion of the data is recorded in reality and on a small subset of scenarios. The dataset could be greatly improved by adding more real recordings.

Due to our participatory approach, we continued end-user interviews during data recording. This resulted in a request for the *Meal* scenario to be adjusted to eat cereals from a bowl instead of using a fork to eat fruit. We adjusted this and recorded additional data in a separate instance. This results in an imbalance in the number of scenarios and users.

B. Simulation-Reality-Gap: Camera Data

The simulated camera follows the real camera in terms of camera parameters and effects (compare [35]). In order to verify the simulated camera quality, we generated data with both the real and simulated cameras in environments as similar as possible (Fig. 7).

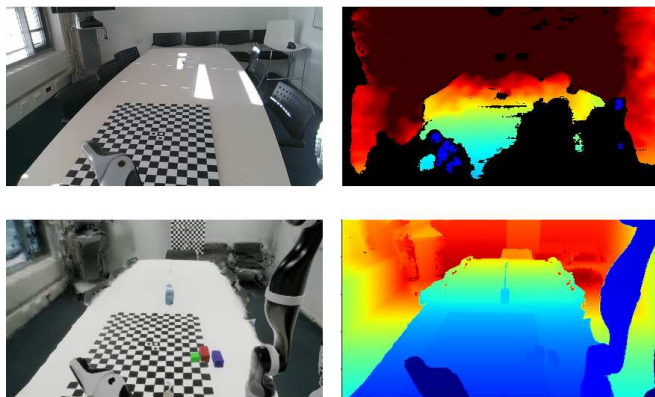


Fig. 7: Comparison of real (top) and simulated (bottom) camera data

It can be seen that the color data (left) is close to identical, whereas the depth data (right) shows vast differences, as the simulated camera is perfect while the real camera suffers from multiple image effects. These mostly stem from the stereo depth algorithm used by the real camera, which uses grayscale image data generated from two additional built-in cameras. We provide simulated versions of these images in a separate repository for users interested in calculating more realistic depth images.

There is no additional noise or image effects in the simulated data. If required, users can simply add these manually.

C. Simulation-Reality-Gap: Robot Arm

As mentioned in [35], the simulated robot arm is built from original robot data, including mechanical dimensions

and meshes. However, in order to avoid movement limitations during data generation, the simulated robot is not controlled using an inverse kinematic and therefore not limited to the motion limits of the real robot. Instead, the simulated joints are spring-based and can, to some extent, move beyond the intended angle limits.

The simulated robot is controlled by moving the motion controller and having the end effector follow it. This results in each robot link in row being pulled along, such that the robot behaves similar to a rope in zero gravity. While this is a major difference to reality, this should not change the relative motion per datapoint which depends only on the camera pose.

D. Simulation-Reality-Gap: Grasping

The simulated grasping is not physics-based but instead a software solution, such that an object is considered attached to the robot hand if squeezed by opposite fingers. This can sometimes cause unrealistic behavior.

In our setup, however, objects are easily graspable with the robot. Poorly-graspable or very heavy objects would have more issues that are therefore avoided. Another factor weakening the effect is due to the use of human operators instead of a script. It can be assumed that humans instinctively prefer realistic grasps.

However, the two doors (room door and microwave) could not be designed as such, as they are not lifted but instead opened or closed. Custom interactions were designed for these, where the robotic hand would retain a relative pose to the handle whilst grasped.

VIII. CONCLUSIONS

In summary, we present a novel dataset applicable to multiple fields associated with assistive robotics. The dataset is easily accessible free of cost and can be used for both robot control as well as activity recognition tasks.

We provided detailed descriptions of the specific method applied to generate the dataset, using both a simulation environment and an associated setup in reality. The capabilities and limitations of the dataset were discussed in detail and metrics were presented as baselines for machine learning research.

Future work should focus on utilizing the provided data to gain insights into user behavior and optimize shared control implementations based on this knowledge. This can, for example, be achieved by analyzing the recorded motions and manually implementing specific interactions, possibly dependent on the current action or activity. Alternatively, data-driven machine-learning models could be trained to predict the user's intended motion in order to offer the most likely direction of control as part of a user interface.

REFERENCES

- [1] J. M. Wiener, R. J. Hanley, R. Clark, and J. F. Van Nostrand, "Measuring the activities of daily living: Comparisons across national surveys," *Journal of gerontology*, vol. 45, no. 6, pp. S229–S237, 1990.
- [2] P. Turgeon, M. Dubé, T. Laliberté, P. S. Archambault, V. H. Flamand, F. Routhier, and A. Campeau-Lecours, "Mechanical design of a new device to assist eating in people with movement disorders," *Assistive Technology*, vol. 0, no. 0, pp. 1–8, 2020, pMID: 32105199. [Online]. Available: <https://doi.org/10.1080/10400435.2020.1734111>

- [3] M. Zhong, Y. Zhang, X. Yang, Y. Yao, J. Guo, Y. Wang, and Y. Liu, "Assistive grasping based on laser-point detection with application to wheelchair-mounted robotic arms," *Sensors*, vol. 19, no. 2, p. 303, 2019.
- [4] K. M. Tsui, D.-J. Kim, A. Behal, D. Kontak, and H. A. Yanco, "i want that": Human-in-the-loop control of a wheelchair-mounted robotic arm," *Applied Bionics and Biomechanics*, vol. 8, no. 1, pp. 127–147, 2011.
- [5] F. F. Goldau, T. K. Shastha, M. Kyrarini, and A. Gräser, "Autonomous multi-sensory robotic assistant for a drinking task," in *2019 IEEE 16th International Conference on Rehabilitation Robotics (ICORR)*. IEEE, 2019, pp. 210–216.
- [6] L. R. Hochberg, D. Bacher, B. Jarosiewicz, N. Y. Masse, J. D. Simeral, J. Vogel, S. Haddadin, J. Liu, S. S. Cash, P. Van Der Smagt *et al.*, "Reach and grasp by people with tetraplegia using a neurally controlled robotic arm," *Nature*, vol. 485, no. 7398, pp. 372–375, 2012.
- [7] P. Try, S. Schöllmann, L. Wöhle, and M. Gebhard, "Visual sensor fusion based autonomous robotic system for assistive drinking," *Sensors*, vol. 21, no. 16, 2021. [Online]. Available: <https://www.mdpi.com/1424-8220/21/16/5419>
- [8] S. Jain and B. Argall, "Robot learning to switch control modes for assistive teleoperation," in *RSS 2016 Workshop on Planning for Human-Robot Interaction: Shared Autonomy and Collaborative Robotics*, 2016.
- [9] B. Driessen, T. T. Kate, F. Liefhebber, A. Versluis, and J. Van Woerden, "Collaborative control of the manus manipulator," *Universal Access in the Information Society*, vol. 4, no. 2, pp. 165–173, 2005.
- [10] S. Jain, A. Farshchiansadegh, A. Broad, F. Abdollahi, F. Mussa-Ivaldi, and B. Argall, "Assistive robotic manipulation through shared autonomy and a body-machine interface," in *2015 IEEE international conference on rehabilitation robotics (ICORR)*. IEEE, 2015, pp. 526–531.
- [11] S. Autexier, C. Lüth, and R. Drechsler, *Künstliche Intelligenz im Gesundheitswesen*. Springer Fachmedien Verlag Wiesbaden, march 2022, ch. Das Bremen Ambient Assisted Living Lab und darüber hinaus – Intelligente Umgebungen, smarte Services und Künstliche Intelligenz in der Medizin für den Menschen, pp. 835–850.
- [12] 3Dconnexion. (2023) SpaceMouse Compact. [Online]. Available: <https://3dconnexion.com/us/product/spacemouse-compact/>
- [13] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: <https://www.tensorflow.org/>
- [14] D. Anguita, A. Ghio, L. Oneto, X. Parra, J. L. Reyes-Ortiz *et al.*, "A public domain dataset for human activity recognition using smartphones." in *Esann*, vol. 3, 2013, p. 3.
- [15] B. Bruno, F. Mastrogiovanni, A. Sgorbissa, T. Vernazza, and R. Zaccaria, "Analysis of human behavior recognition algorithms based on acceleration data," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 1602–1607.
- [16] M. Ruzzon, A. Carfi, T. Ishikawa, F. Mastrogiovanni, and T. Murakami, "A multi-sensory dataset for the activities of daily living," *Data in Brief*, vol. 32, p. 106122, 2020.
- [17] D. Roggen, A. Calatroni, M. Rossi, T. Holleczeck, K. Förster, G. Tröster, P. Lukowicz, D. Bannach, G. Pirkel, A. Ferscha *et al.*, "Collecting complex activity datasets in highly rich networked sensor environments," in *2010 Seventh international conference on networked sensing systems (INSS)*. IEEE, 2010, pp. 233–240.
- [18] P. Patek and A. Thakkar, "A survey on video-based human action recognition: recent updates, datasets, challenges, and applications," *Artificial Intelligence Review*, vol. 54, pp. 2259–2322, 2021.
- [19] T. Alshammari, N. Alshammari, M. Sedky, and C. Howard, "Simadl: Simulated activities of daily living dataset," *Data*, vol. 3, no. 2, 2018. [Online]. Available: <https://www.mdpi.com/2306-5729/3/2/11>
- [20] S. M. Gerber, R. M. Müri, U. P. Mosimann, T. Nef, and P. Urwyler, "Virtual reality for activities of daily living training in neurorehabilitation: a usability and feasibility study in healthy participants," in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2018, pp. 1–4.
- [21] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*. IEEE, 2015, pp. 510–517.
- [22] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes," 2017. [Online]. Available: <https://arxiv.org/abs/1711.00199>
- [23] I. Lenz, H. Lee, and A. Saxena, "Deep learning for detecting robotic grasps," *Int. J. Rob. Res.*, vol. 34, no. 4–5, p. 705–724, apr 2015. [Online]. Available: <https://doi.org/10.1177/0278364914549607>
- [24] C. Goldfeder, M. Ciocarlie, H. Dang, and P. K. Allen, "The columbia grasp database," in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 1710–1716.
- [25] D.-J. Kim, R. Lovelett, and A. Behal, "Eye-in-hand stereo visual servoing of an assistive robot arm in unstructured environments," in *2009 IEEE International Conference on Robotics and Automation*. IEEE, 2009, pp. 2326–2331.
- [26] D. Kim, R. Hazlett-Knudsen, H. Culver-Godfrey, G. Rucks, T. Cunningham, D. Portee, J. Bricout, Z. Wang, and A. Behal, "How autonomy impacts performance and satisfaction: Results from a study with spinal cord injured subjects using an assistive robot," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 42, no. 1, pp. 2–14, 2012.
- [27] Kinova inc., "Jaco assistive robot - user guide," (last visited on 13.02.2023), 2021, "EN-UG-007-r05-202111". [Online]. Available: <https://assistive.kinovarobotics.com/uploads/EN-UG-007-Jaco-user-guide-R05.pdf>
- [28] Intel Corporation. (2022, november) Intel realSense - product family d400 series. (last visited on 13.02.2023). [Online]. Available: <https://www.intelrealsense.com/wp-content/uploads/2022/11/Intel-RealSense-D400-Series-Datasheet-November-2022.pdf>
- [29] M. Riel. (2020) Understanding action research. online. [Online]. Available: <https://www.ccarweb.org/what-is-action-research>
- [30] A. Baumeister, M. Pascher, Y. Shivashankar, F. F. Goldau, U. Frese, J. Gerken, E. Gardo, B. Klein, and P. Tolle, "Ai for simplifying the use of an assistive robotic arm for people with severe body impairments," *Gerontechnology*, vol. 21, 10 2022. [Online]. Available: <https://doi.org/10.4017/gt.2022.21.s.578.5.sp7>
- [31] M. T. Wright, M. Block, H. Kilian, and K. Lemmen, "Förderung von qualitätswissenschaft durch partizipative gesundheitsforschung," *Prävention und Gesundheitsförderung*, vol. 3, no. 8, pp. 147–154, 2013.
- [32] A. Baumeister, E. Gardo, P. Tolle, B. Klein, M. Pascher, J. Gerken, F. Goldau, Y. Shivashankar, and U. Frese, "The importance of participatory design for the development of assistive robotic arms: initial approaches and experiences in the research projects mobile and dof-adaptiv," in *Connected Living: international and interdisciplinary conference (2021)*, Frankfurt am Main, 2021.
- [33] A. G. Tuckett, "Applying thematic analysis theory to practice: A researcher's experience," *Contemporary nurse*, vol. 19, no. 1-2, pp. 75–87, 2005.
- [34] Epic Games. (2023) Unreal engine. (last visited on 13.02.2023). Version 4.27.2. [Online]. Available: <https://www.unrealengine.com>
- [35] M. Pascher, F. F. Goldau, K. Kronhardt, U. Frese, and J. Gerken, "AdaptiX – A Transitional XR Framework for Development and Evaluation of Shared Control Applications in Assistive Robotics," in review.
- [36] HTC Corporation, "Vive pro 2 headset," 2021. [Online]. Available: <https://www.vive.com/us/product/vive-pro2/overview/>
- [37] C. Hertzberg, R. Wagner, U. Frese, and L. Schröder, "Integrating generic sensor fusion algorithms with sound state representations through encapsulation of manifolds," *Information Fusion*, vol. 14, no. 1, pp. 57–77, 2013.
- [38] T. L. Koller and U. Frese, "The interacting multiple model filter and smoother on boxplus-manifolds," *Sensors*, vol. 21, no. 12, p. 4164, 2021.
- [39] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.
- [40] F. F. Goldau and U. Frese, "Learning to map degrees of freedom for assistive user control: Towards an adaptive dof-mapping control for assistive robots," in *The 14th Pervasive Technologies Related to Assistive Environments Conference*, 2021, pp. 132–139.