Learning Spatial Bimanual Action Models Based on Affordance Regions and Human Demonstrations

Björn S. Plonka, Christian Dreher, Andre Meixner, Rainer Kartmann, and Tamim Asfour

***Abstract*— In this paper, we present a novel approach for learning bimanual manipulation actions from human demonstration by extracting spatial constraints between affordance regions, termed affordance constraints, of the objects involved. Affordance regions are defined as object parts that provide interaction possibilities to an agent. For example, the bottom of a bottle affords the object to be placed on a surface, while its spout affords the contained liquid to be poured. We propose a novel approach to learn changes of affordance constraints in human demonstration to construct spatial bimanual action models representing object interactions. To exploit the information encoded in these spatial bimanual action models, we formulate an optimization problem to determine optimal object configurations across multiple execution keypoints while taking into account the initial scene, the learned affordance constraints, and the robot's kinematics. We evaluate the approach in simulation with two example tasks (pouring drinks and rolling dough) and compare three different definitions of affordance constraints: (i) component-wise distances between affordance regions in Cartesian space, (ii) component-wise distances between affordance regions in cylindrical space, and (iii) degrees of satisfaction of manually defined symbolic spatial affordance constraints.**

I. Introduction

Humanoid robots are expected to become increasingly autonomous to assist people in their daily activities. To do this, they must be able to acquire new skills and to perform tasks such as pouring a drink or preparing meals. The ability to learn from demonstration is crucial for robots, as it enables them to acquire knowledge through natural interactions with humans without the need for experts [1].

In this work, we present a method to learn bimanual manipulation actions from human demonstration by extracting and reproducing spatial constraints between affordance regions, so-called *affordance constraints*, of the objects involved. We define an affordance region as a specific object part that supports a particular action, inspired by the affordance concept introduced by Gibson [2]. According to this, a bottle may have several affordance regions: The bottom affords placing the bottle on a surface, its spout affords pouring a contained liquid, and the side of a bottle affords grasping the bottle. Specifically, we learn changes of affordance constraints in human demonstration and store this information in so-called *Spatial Bimanual Action Models* (SBAMs) to capture the interactions between the objects involved as depicted in Fig. 1.

The research leading to these results has received funding from the German Research Foundation (DFG) within the SFB-1574 and the Carl Zeiss Stiftung through the JuBot project and German Federal Ministry of Education and Research (BMBF) under the Robotics Institute Germany (RIG). The authors are with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany. bjoern.plonka@student.kit.edu, {c.dreher,andre.meixner,asfour}@kit.edu

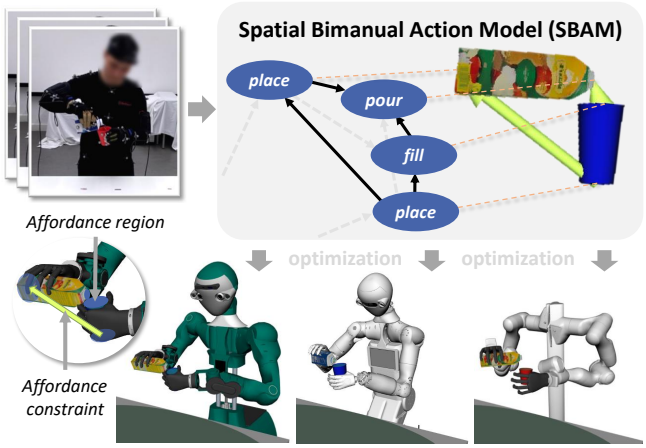

Fig. 1. We learn the spatial constraints between affordance regions (affordance constraints) from human demonstrations to obtain spatial bimanual action models. For execution on a robot, these are used to maximize the similarity between the learned affordance constraints and those present in the current scene subject to the robot's kinematics.

We consider three different types of affordance constraints: (i) component-wise distances between affordance regions in Cartesian space (*Cartesian affordance constraints*), (ii) component-wise distances between affordance regions in cylindrical space (*cylindrical affordance constraints*), and (iii) degrees of satisfaction of manually defined symbolic spatial constraints (*symbolic spatial affordance constraints*). Additionally, we show how to leverage the information in such an SBAM for the execution of the bimanual manipulation action on a humanoid robot. This is done by formulating an optimization problem that maximizes the similarity between the affordance constraints observed in the demonstrations and those present in the current scene. We evaluate the effectiveness of our approach using previously unseen scenes, different objects, three different definitions of affordance constraints, as well as three different humanoid robots in simulation: ARMAR-6, ARMAR-7, and a bimanual Franka Emika Panda setup.

To summarize, our contributions are three-fold: (i) a Spatial Bimanual Action Model that is learned from bimanual human demonstrations and encodes spatial constraints between pairs of affordance regions in a bimanual manipulation task, (ii) formulation of an optimization problem for finding optimal scene arrangements for the execution of the bimanual manipulation task while incorporating constraints extracted from the initial scene state, the robot's kinematics, and the learned SBAM, as well as (iii) a comparison of three definitions of affordance constraints: Cartesian, cylindrical, and symbolic spatial affordance constraints.

II. Related Work

We discuss related work concerning the use of spatial constraints to describe actions (Section II-A), action models that incorporate either spatial or temporal constraints learned from demonstrations (Section II-B), and approaches for incorporating affordance constraints (Section II-C).

A. Spatial Constraints

A vital part in Programming by Demonstrations (PbD) is learning the task constraints needed for a successful task execution. While many representations focus on spatial constraints only in 2D [3], [4] we believe that complex manipulation tasks require the consideration of constraints in 3D space. Ziaeetabar et al. [3] presented an approach for human action recognition by tracking a set of pre-defined symbolic spatial relations between objects approximated by bounding boxes. In our work, instead of considering binary relations between objects, quantify the degree of satisfaction of spatial constraints between objects' affordance regions. O'Keeffe's idea of using probabilistic models to ground semantic constraints [4] was extended to the third dimension by Kartmann et al. [5], representing each spatial constraint by a joint probability function. In this work, we evaluate their performance against the less semantically enriched Cartesian and cylindrical representations.

B. Learning Action Models from Demonstrations

As expressed by Billard et al. [1], user friendlyinterfaces for teaching in PbD include visual perception [6]–[8] and kinesthetic teaching [9]–[11]. Since then, also verbal interfaces [12] as well as combinations of these [13] were tested. The kinesthetic approach taken by Ureche et al. [9] learns changes of action constraints over time, enabling a robot to execute the action even in novel scenarios. Gao et al. [6] propose Bi-KVIL, an approach visual imitation learning of bimanual tasks. The approach extracts geometric constraints between keypoints on the object's surface from video and use their constraints to generalize to new tasks. Drawing inspiration from how humans learn through conversation, Nicolescu et al. [12] introduced a system designed to construct a task model based on verbal instructions. The approach is based on synthesizing a symbolic and hierarchical task representation from a single dialogue between the robot and the human by joining spatial information with boolean operators. An example of a multi-model teaching interface was given by Kartmann and Asfour [13]. They revealed how a robot can learn spatial constraints iteratively from visual demonstrations given verbal cues.

With the rise of pre-trained neural networks like GPT-4 [14], [15], several works investigated ways to use such networks as the instructor. DALL-E-Bot [16] queries DALL-E for an image to create human-like arrangements of objects in the real world. Kwonot et al. [17] show that these networks can be used for more than high-level planning, where GPT-4 was tasked to create an end effector trajectory. These works focus on unimanual actions such as pick-and-place tasks while relying on a probabilistic neuronal network. Akbulut et al. [18] has shown that neural networks (NNs) can be used to learn complex movements from a few demonstrations. Using the approach, a bimanual robot could successfully tie a knot. While NNs have been shown to learn generalized trajectories, we have chosen an optimization-based approach that is independent of pre-trained models and also requires only a few demonstrations.

C. Affordance Representations

Affordance regions provide the task model with symbolic abstractions of object properties, as well as subsymbolical groundings in the object's structure. Several works explore methods to find affordance regions in visual data. Often, the problem is approached by finding the corresponding affordances in images [19]–[21] or in point clouds [22], [23]. For example, ToolEENet [24] finds the 6D pose of an affordance region relative to the object. Koppula and Saxena [8] extract affordance regions and their changes over time from videos and use them in their proposed conditional random fields model. The work focused on human action recognition and anticipation to trigger assistive robot behavior rather than learning task models from human demonstration for the reproduction of the task by a robot. While the concept of affordances receives increasing attention in the robotics community [25], few works have explored the potential of learning spatial and temporal constraints in complex manipulation actions by exploiting knowledge about affordance region pairs in human demonstrations.

III. Spatial Bimanual Action Model

This section describes our approach to learning and reproducing bimanual manipulation actions from human demonstrations by exploiting spatial constraints between affordance regions, i. e. affordance constraints, of the objects involved in the task. In this context, a bimanual action is characterized by two distinct actions, each performed by one hand. We propose three different definitions of such affordance constraints (Section III-A) and described how changes of affordance constraints over time can be segmented (Section III-B), a necessary step for generalization. The core of our approach is the Spatial Bimanual Action Model (SBAM) that learns the object motion during the demonstration by generalizing observed affordance constraints based on multiple demonstrations (Section III-C). We propose an optimization problem to reproduce the bimanual action based on the SBAM in a human-like manner, given the current scene and a specified humanoid robot kinematic (Section III-D). Fig. 2 shows a graphical overview of the entire process of learning and executing such an SBAM.

A. Definition of Affordance Constraints

For a robot to effectively perform a bimanual manipulation action based on a small set of demonstrations, it needs to identify the spatial constraints between objects involved in the action. To do so, we assign affordance regions to parts of objects and track affordance constraints throughout the demonstration of the bimanual manipulation action.

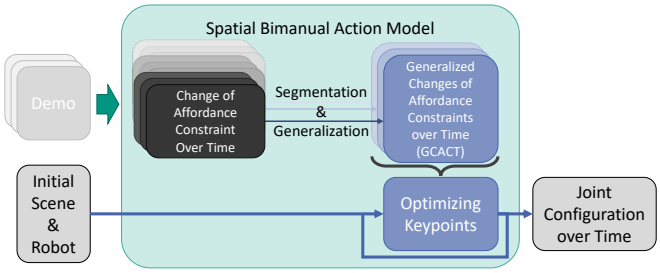

Fig. 2. A simplified visual overview of the Spatial Bimanual Action Model.

Affordance constraints are spatial constraints between a pair of affordance regions of objects involved in the action. We approximate affordance regions of objects with an ellipse relative to the object frame (see Figure 1). For example, a *pour from* affordance region of a milk carton is approximated by an ellipse in the spout.

The affordance constraints we compare are a different way to subsymbolically represent the vector connecting the centers of two affordance regions $A_0 = (A_{0,x}, A_{0,y}, A_{0,z})^\top$ and $A_1 = (A_{1,x}, A_{1,y}, A_{1,z})^\top$. We selected three representations, which we believe are beneficial for the reproduction of the bimanual action: (i) Cartesian affordance constraints, (ii) cylindrical affordance constraints, and (iii) symbolic spatial affordance constraints.

1) Cartesian Affordance Constraint (CaAC): CaACs are calculated as Cartesian distances $x = A_{0,x} - A_{1,x}$, $y = A_{0,y} - A_{1,y}$, and $z = A_{0,z} - A_{1,z}$ between the two affordance regions. Their key advantages are clarity and mathematical simplicity. Cartesian coordinates provide an intuitive way to represent spatial constraints and allow straightforward interpretation of vector differences without ambiguity. Furthermore, they simplify mathematical operations and make calculations easier.

2) Cylindrical Affordance Constraint (CyAC): Cylindrical Affordance Constraints (CyACs) represent the vector between the two affordance regions in cylindrical coordinates with radius ρ, azimuth ϕ, and elevation z as

$$\begin{aligned} \rho &= \sqrt{(A_{0,x} - A_{1,x})^2 + (A_{0,y} - A_{1,y})^2} \\ \phi &= \arctan2(A_{0,y} - A_{1,y}, A_{0,x} - A_{1,x}) \\ z &= A_{0,z} - A_{1,z}. \end{aligned}$$

This representation has the advantage of generalizing rotational symmetry around the z-axis and enables combining approaching movements from multiple different directions, as shown in our previous work [5]. However, since the azimuth ϕ is defined as an angle within the interval $[-\pi, \pi]$, it allows for discontinuities, which adds complexity to the process of learning generalized trajectories.

3) Symbolic Spatial Affordance Constraint (SSAC): SSACs represent the vector between the two affordance regions with a set of symbolic spatial constraints, similar to our earlier work [13] which describes spatial relations between object centers. How well the vector from A_1 to A_0 satisfies the spatial constraint sc is thus given by $v_{\text{sc}} = \mathbb{P}_{\phi_{\text{sc}}}(\phi) \cdot \mathbb{P}_{\rho_{\text{sc}}}(\rho) \cdot \mathbb{P}_{z_{\text{sc}}}(z)$, where $\mathbb{P}_{\rho_{\text{sc}}}(\rho)$ is the probability density at the radius ρ for the spatial constraint sc, while $\mathbb{P}_{\phi_{\text{sc}}}(\phi)$ and $\mathbb{P}_{z_{\text{sc}}}(z)$ are the the probability density at the azimuth ϕ and elevation z, respectively. We assume that $\rho_{\text{sc}} \sim \mathcal{N}(\mu_\rho, \sigma_\rho^2)$, $\phi_{\text{sc}} \sim \mathcal{M}(\mu_\phi, \kappa_\phi)$, and $z_{\text{sc}} \sim \mathcal{N}(\mu_z, \sigma_z^2)$, where $\mathcal{N}(\cdot)$ denotes a Gaussian distribution, while $\mathcal{M}(\cdot)$ denotes a von Mises distribution, which is a circular distribution defined on the interval $[-\pi, \pi]$, wrapping around periodically. The concrete spatial constraints are given by choosing the mean (μ_ρ, μ_ϕ, μ_z) and variances (σ_ρ^2, κ_ϕ, σ_z^2). An example is shown in Figure 6.

B. Segmenting Changes of Affordance Constraints Over Time

We observe affordance constraints over the course of an action and we are specifically interested in the changes of affordance constraints over time. They can be seen as a trajectory in the "affordance space". Segmenting the observed changes means finding keypoint candidates, i. e. points in time associated with important events. At each candidate keypoint the robot may be required to assume a specific joint configuration in order to best satisfy the learned affordance constraints. Later we will present an approach to derive keypoints from the set of keypoint candidates. We conduct a segmentation based on partial linear approximation. We start by finding a segmentation point, such that the area A_e between the original data and the linear approximation, given by lines between the segmentation points, is minimal (see "$n = 2$" in Figure 3). The same is applied recursively until A_e falls beneath a predetermined threshold ϵ. This threshold is relative to the amplitude of the values of the spatial constraint, ensuring that important information is preserved even when operating at different scales across various spatial constraints. This is important when working with cylindrical data, as the elevation and radius are at a different magnitude from the azimuth that only ranges from $[-\pi, \pi]$. Once all the segments are found, a final pass with a heuristic is performed. Therefore, each pair of consecutive segments is checked to see if the area between the original data and a single segment is smaller than ϵ. If this is the case, the segmentation point is removed, else we find the segmentation point connecting the two consecutive segments, such that A_e is minimal. (see "final" in Figure 3).

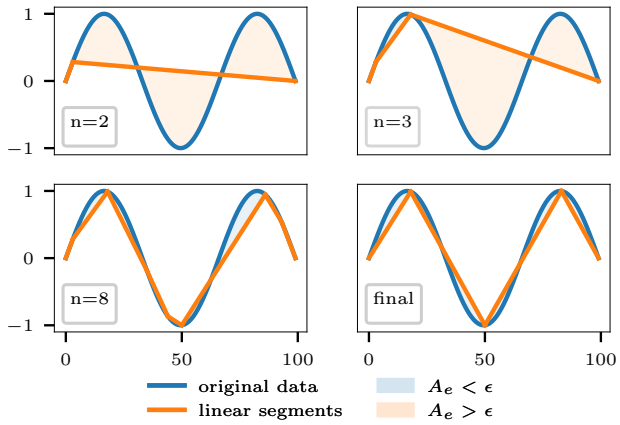

Fig. 3. An exemplary segmentation on synthetic data.

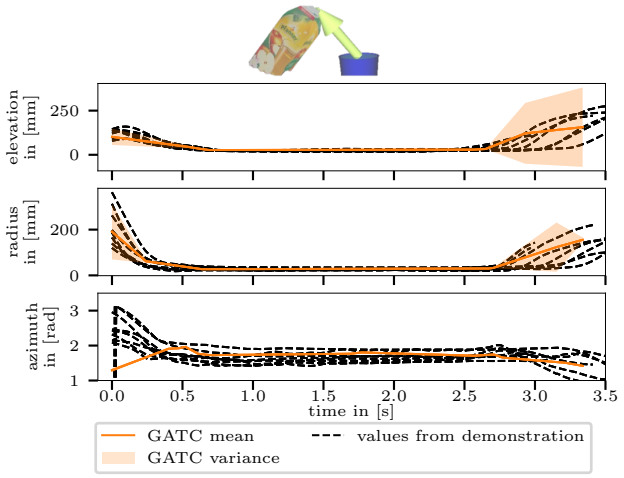

Fig. 4. Segmenting changes of affordance constraints over time allow for generalization. Alongside the mean value, we also compute the standard deviation at the end of each segment. The colored areas show the confidence intervals given the standard deviation and mean value.

C. Learning Spatial Bimanual Action Models

To combine affordance constraints from multiple demonstrations, generalized changes of affordance constraints over time (GCACOT) are needed. These are derived by generalizing the segmented affordance constraints from each individual demonstration in an incremental way. A GCACOT consists of a set of keypoints. Each keypoint is described by the mean value, standard deviation, and its point in time. For the incremental update, the corresponding keypoint candidates have to be matched between demonstrations. However, the number of segments may vary from one demonstration to another. To determine the most probable corresponding keypoint candidates, a method similar to dynamic time warping [26] is employed (see Fig 4). Given a new matched keypoint candidate that occurs at time t_1, we update the time t_0 of the keypoint in the GCACOT as $t_0' = \frac{1}{n}t_1 + \frac{n-1}{n}t_0$ where t_0' is the updated time of the keypoint in the GCACOT and $n-1$ is the amount of previously analyzed demonstrations. The mean value at the keypoint is then updated the same way, while the standard deviation $\hat{v}_0$ is updated by $\hat{v}_0' = \sqrt{\frac{1}{n-1}((v_1 - \bar{v}_0)^2 + (n-2)\hat{v}_0^2)}$ with $\hat{v}_0'$ being the updated standard deviation of the keypoint in the GCACOT, $\bar{v}_0$ being the mean value of the keypoint in the GCACOT, and v_1 being the value of the keypoint in the new demonstration.

These GCACOTs are of interest because the standard deviation at any given time indicates how important it is to satisfy that constraint, while the mean simultaneously provides the desired target values. The lower the standard deviation is, the more important it is to fulfill the corresponding constraint. Note that the same methodology applies to all three kinds of affordance constraints used in this work. In the following, we will show how this representation is used to formulate an optimization problem for the execution of the bimanual action on a robot.

D. Executing Learned Bimanual Manipulation Actions

We assume that a bimanual manipulation action can be represented as a sequence of specific object configurations at certain keypoints in time. To find these keypoints, a histogram of all keypoint candidates from all GCACOTs (presented in Section III-C) is created. The keypoints are determined by searching for clusters of keypoint candidates from all GCACOTs in a smaller time window. This is achieved by applying a Butterworth filter to the histogram data and a peak detection (see Fig 5). For the execution of a learned bimanual manipulation action through an SBAM, we formulate an optimization problem to find optimal object placements at the identified keypoints. Optimal object placements are those that satisfy the learned affordance constraints between the objects involved.

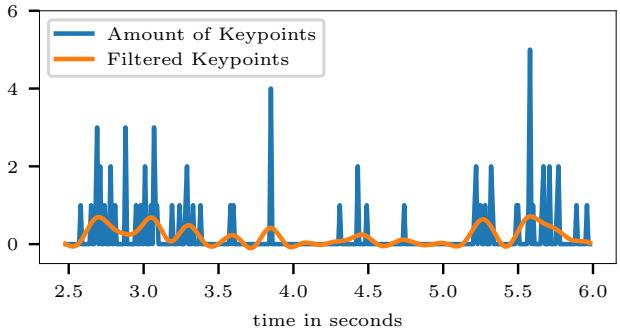

Fig. 5. The cumulative number of keypoint candidates that fall in the time window of the corresponding bin. Butterworth filter and a peak detection are used to determine the optimal keypoints.

We define the objective function of our optimization problem from three terms as follows:

$$\underset{\theta \in \Theta_{\text{lim}}}{\operatorname{argmin}} \; t_s(\theta) + t_h(\theta) + t_d(\theta) \tag{1}$$

The first term, t_s, is the similarity term that brings the objects into the desired configuration by satisfying the objects' respective affordance region constraints. The second term, t_h, is the human-likeness term that favors human-like configurations during the optimization. The third term, t_d, is a damping term that reduces unnecessary object movements during optimization. As can be seen, we optimize the robot's configuration $\theta \in \Theta$ so that the resulting poses of the objects in its hands satisfy the observed affordance constraints in the SBAM as close as possible at each identified keypoint. Thus, we use the robot's kinematics to naturally constrain the optimization problem to yield object poses that are reachable by the robot. These three terms will be defined in the following in more detail.

The current affordance constraints are calculated by first finding an initial grasp for the manipulated objects. Given the current robot configuration θ and the grasp poses we calculate the pose of the grasped objects in the global frame, which in turn allows computing the position of the affordance regions in the global frame. Meanwhile, the desired affordance constraints are given by the SBAM. Let $w_{i,j,k} = \frac{1}{1+\hat{v}_{i,j,k}}$ be the weight of the spatial constraint k between affordance regions i and j with $\hat{v}_{i,j,k}$ denoting the corresponding standard

deviation. An affordance constraint that is similar across all demonstrations will have a low standard deviation and thus a high weight as it is chosen to be inversely proportional to the standard deviation. Such a weighting serves as a measure of importance to favor affordance constraint segments in the optimization that have been observed similarly many times and to disregard those that were observed by coincidence. We define the similarity term as:

$$t_s(\theta) = \sum_{i,j} \left(s_{k,i,j}(\theta) \right)^{1/2},$$

where $k \in \{\text{ca, cy, sc}\}$ defines the used type of affordance constraint, and i and j correspond to affordance regions, while $s_{k,i,j}(\theta)$ is the weighted similarity.

For each type of spatial constraint, a different way of calculating the weighted similarity is used. For the CaACs, we chose to calculate the weighted similarity as

$$\begin{aligned} s_{ca} &= w_x^2(x_1 - x_2)^2 + w_y^2(y_1 - y_2)^2 + w_z^2(z_1 - z_2)^2 \\ &= (w_x x_1 - w_x x_2)^2 + (w_y y_1 - w_y y_2)^2 + (w_z z_1 - w_z z_2)^2, \end{aligned}$$

with current (x_1, y_1, z_1) and target CaACs (x_2, y_2, z_2) and the corresponding weights (w_x, w_y, w_z).

The usage of the same metric is unsuitable for CyAC as a slight change in azimuth does not have the same impact as the same change for elevation or radius, in contrast to CaACs. For this reason, we chose to calculate the similarity s_{cy} between the weighted current $(w_\rho \rho_1, w_\phi \phi_1, w_z z_1)$ and weighted target CyACs $(w_\rho \rho_2, w_\phi \phi_2, w_z z_2)$ by transforming them into Cartesian coordinates:

$$\begin{aligned} s_{cy} &= (\cos(w_\phi \phi_1) w_\rho \rho_1 - \cos(w_\phi \phi_2) w_\rho \rho_2)^2 \\ &\quad + (\sin(w_\phi \phi_1) w_\rho \rho_1 - \sin(w_\phi \phi_2) w_\rho \rho_2)^2 \\ &\quad + (w_z z_1 - w_z z_2)^2. \end{aligned}$$

This similarity is closely related to the Cartesian similarity, where $w_x x_1$ corresponds to $\cos(w_\phi \phi_1) w_\rho \rho_1$. In order to mitigate numerical precision issues, we opted to use the logarithm of the SSAC to compute the similarity, as it keeps the values in a manageable range. Thus, the similarity s_{sc} between the current $(v_{sc_1,1}, \ldots, v_{sc_n,1})$ and target SSACs $(v_{sc_1,2}, \ldots, v_{sc_n,2})$ is given by

$$s_{sc} = \sum_{i=1}^{n} \left(w_i \left(\log(v_{sc_i,1}) - \log(v_{sc_i,2}) \right) \right)^2.$$

To generate more human-like executions, we added the human-likeness term $t_h(\theta)$ using the $\text{SOA}_{\boldsymbol{q}}$ criterion presented in our previous work [27].

In addition, a damping term $t_d(\theta) = \sum_{o \in M} w_0' k_0(\theta)$ is defined, with M as the set of all currently manipulated objects, w_0' a cumulative weight and k_0 the deviation between the current position and the position at the previous keypoint for the manipulated object o. When the deviation between the pose of the objects is large in the demonstrations, the weight of the connected affordance constraints is lower. Thus, the optimizer ignores the similarity between the current and the desired values of these affordance constraints, increasing the probability of unnecessary object movement. This is counteracted by the norm of the difference between the current position of the object o and the position at the previous keypoint to the additional terms k_o. As a weight, we define w_o' to be proportional to the time delta and the standard deviation of all pairs of affordance regions, where at least one is part of the object o, such that the auxiliary terms are weighted less when the similarity is more important.

In this work, we employed the gradient-free Nelder–Mead method [28] to find the pose of the end effectors. Additionally, we used a non-linear optimization-based inverse kinematics solver with the human-likeness criterion $\text{SOA}_{\boldsymbol{q}}$ described in [27] to find a human-like posture to reach both end effector poses at each keypoint. The trajectories are then computed by interpolating the joint values linearly.

Overall, we generate optimal and human-like robot poses for each keypoint. They are optimal in the sense that the affordance regions of objects manipulated by the robot in the given configuration best satisfy the affordance constraints, independent of the constraint set.

IV. Experiments and Evaluation

To evaluate our proposed spatial bimanual action model (SBAM) we show in Section IV-A qualitative results of the generated robot behavior using two bimanual actions of the KIT Bimanual Manipulation Dataset [29]: *pouring drink* and *rolling dough*. In Section IV-B, we perform a quantitative cross-validation resulting from the SBAMs.

A. Qualitative Evaluation

To demonstrate the performance of a learned SBAM we refer to the video attachment[1], in which individual clips are referenced by the symbol ⓝ for the n-th clip.

The first three executions (cf. ①–③) show ARMAR-6 executing *pouring drink* with an *apple juice* and a *large cup*. The SBAMs were learned from 11 demonstrations (8 demonstrations of pouring apple juice into a large cup and 3 of pouring milk into a small cup), using three distinct types of affordance constraints: ① Cartesian affordance constraints (CaACs), ② cylindrical affordance constraints (CyACs), and ③ symbolic spatial affordance constraints (SSACs).

For the SSACs, we define the spatial constraints as in our previous work [13] and parameterize them as in Table I.

TABLE I
DEFINITION OF SSACS USED IN OUR EVALUATIONS

constraint	μ_ρ	σ_ρ^2	μ_ϕ	κ_ϕ	μ_z	σ_z^2
above	0	250	0	10^{-5}	250	100
below	0	250	0	10^{-5}	-250	100
close	0	100	0	10^{-5}	0	250
far away	500	100	0	10^{-5}	0	250
in front	0	250	$\frac{3}{2}\pi$	5	0	250
behind	0	250	$\frac{1}{2}\pi$	5	0	250
left	0	250	π	5	0	250
right	0	250	0	5	0	250

For all three constraints, our approach was able to generate human-like joint configurations, so that the spout of the bottle

[1] https://youtu.be/SuCRJvBpQNs

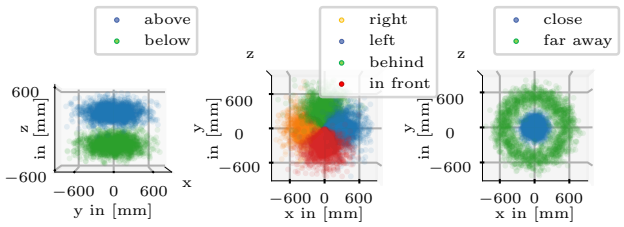

Fig. 6. Displaying the chosen spatial constraints by drawing 2000 random

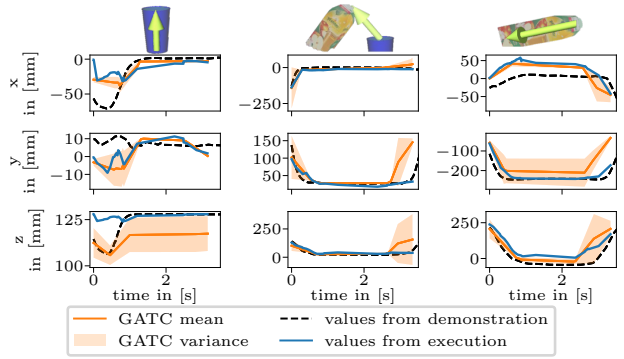

Fig. 7. Cartesian affordance constraints (CaACs) in comparison to the generalized linear segments as well as the values of the action as performed by the human demonstrator for the *pouring drink*-action

is above the cup while the bottle is being rotated. Several demonstrations exhibited abrupt movements at the end of the action. This can be attributed to a higher observed variance in the GCACOT towards the end of the action when the affordance constraints relevant for the bimanual action will relax, allowing for more object movement.

While ①–③ used the same combination of objects as 8 of the demonstrations, ④ and ⑤ show new combinations of the objects with different robots, displaying the versatility and generalizability of the approach. ④ uses the same SBAM as ②, applied to ARMAR-7 pouring milk into a large cup. Similarly, ⑤ uses also the SBAM of ① in combination with the new initial scene and a bimanual Franka Emika Panda setup to create a successful execution of *pouring drink*. While the majority of the execution appears to be as expected, each action experiences a short interruption, where the milk carton is rotated but the spout is not above the cup. Clips ⑥ and ⑦ show ARMAR-6 and ARMAR-7 respectively rolling dough with a *rolling pin* on the *table*. While the SBAM in ⑥ uses SSACs, the SBAM optimizing the configurations for ARMAR-7 in ⑦ utilizes CyACs to represent the same two rolling demonstrations. In both executions, the desired motions of rolling dough are visible.

B. Quantitative Evaluation

To assess the effectiveness of our approach, we employed a cross-validation for two different bimanual actions: *Pouring Drink* IV-B.1 and *Rolling Dough* IV-B.2.

1) Pouring Drink: To train the spatial bimanual action model we used 8 demonstrations of the *pouring drink* action from *apple juice* into the *large cup* and 3 demonstrations of the *pouring drink* action from the *small milk* into the *small cup* and evaluated its performance on the 9th unseen demonstration of *pouring drink* from *apple juice* into the *large cup* in a simulation. The reported numbers are the mean of all nine folds. At each keypoint, we measure the similarity of the unseen demonstration and the execution by calculating the Euclidean distance between vectors connecting the centers of the affordance regions in the demonstrated action and the optimized object poses derived from Equation 1. Our evaluation results, including mean, standard deviation, and minimum values across all keypoints and scenarios, are summarized in Table II, providing a comprehensive analysis of the approach's performance.

TABLE II

AFFORDANCE CONSTRAINT SIMILARITY FOR *pouring drink* IN *mm*

affordance region pair	type	avg	std	min
place onto of table ↓ place of cup large	CaAC	135.233	**123.702**	10.385
	CyAC	**132.965**	131.112	10.379
	SSAC	155.870	138.069	**10.363**
place onto of table ↓ pour into of cup large	CaAC	136.563	**136.543**	9.445
	CyAC	**132.257**	144.325	**9.442**
	SSAC	157.483	155.158	9.454
place onto of table ↓ pour from of apple juice	CaAC	**175.586**	203.581	**0.444**
	CyAC	176.766	203.715	**0.444**
	SSAC	190.240	**191.454**	**0.444**
place onto of table ↓ place of apple juice	CaAC	**214.453**	**216.694**	**0.396**
	CyAC	216.057	219.126	**0.396**
	SSAC	218.582	218.675	**0.396**
place of cup large ↓ pour into of cup large	CaAC	23.087	22.889	**0.208**
	CyAC	**22.072**	**21.133**	**0.208**
	SSAC	24.532	29.096	**0.208**
place of cup large ↓ pour from of apple juice	CaAC	**174.932**	**211.721**	1.849
	CyAC	176.445	221.118	**1.575**
	SSAC	180.333	214.044	2.708
place of cup large ↓ place of apple juice	CaAC	182.554	**192.510**	7.894
	CyAC	**182.544**	200.068	7.327
	SSAC	197.645	208.331	**5.384**
pour into of cup large ↓ pour from of apple juice	CaAC	**165.770**	**208.665**	3.357
	CyAC	169.718	218.626	**1.805**
	SSAC	173.982	215.371	1.963
pour into of cup large ↓ place of apple juice	CaAC	**172.947**	**184.193**	**1.688**
	CyAC	175.450	193.651	7.585
	SSAC	189.538	203.633	2.313
pour from of apple juice ↓ place of apple juice	CaAC	123.095	97.722	**0.621**
	CyAC	**121.581**	94.565	**0.621**
	SSAC	133.062	**93.486**	**0.621**

The evaluation results reveal that the usage of the three representations results in very similar deviations between the demonstration and the execution. The values of the mean difference between the demonstration and the execution for each affordance region pair are tightly clustered with the most deviation observed in the *place onto* affordance region for the *table* and the *pour into* affordance region for *cup large*, exhibiting a mean difference of $25.226\,\mathrm{mm}$. The values are quite high (ca. $17\,\mathrm{cm}$ for the top of the bottle and the top of the cup). Note that most keypoints lie in phases of the action with prevalent nonlinear object movement, e. g. at the start of the *pouring drink* action. Nonlinear object movements lead to an over-representation of keypoint candidates resulting from the segmentation through linear approximation. Additionally, these candidates also show a high variance as the observed object movements coincide with large deviations between

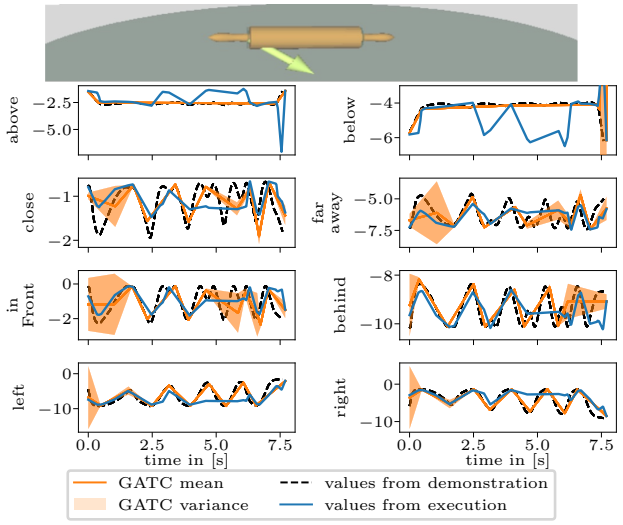

Fig. 8. Symbolic spatial affordance constraints (SSACs) in comparison to the generalized linear segments as well as the values of the action as performed by the human demonstrator for the *rolling dough*-action

the demonstrations used in learning. Because of the high variance, these affordance constraints are not relevant to the execution, but they impact the reported similarities.

Surprisingly, the SSACs, for which we used the same parameters as detailed in Table I, consistently achieved the highest and thus worst mean similarity across nearly all instances, while the CaACs and CyACs demonstrated the lowest mean similarities. This outcome is particularly intriguing given the substantial time investment required for optimizing SSACs – averaging 8 hours for 30 keypoints – compared to ca. 10 minutes each for cylindrical and Cartesian representations on a Ryzen 9 5900X processor running at 4.5 GHz. The keypoints have to be optimized in sequential order as the final robot configuration serves as the initial configuration at the next keypoint. Therefore, parallel optimization was deemed impractical due to the introduction of non-consistent object movement.

The good performance of CaACs can be attributed to their inherent simplicity for generalization and ease of comparability during optimization. In Cartesian coordinates, each dimension possesses an identical size, facilitating a straightforward comparison of the distance between current affordance constraint values and their target counterparts using respective weights. In contrast, CyACs pose challenges for optimization due to the different value ranges in the different dimensions – ranging from $(-\infty, \infty)$ for both elevation and radius, and $[-\pi, \pi]$ for the azimuth. This inherent characteristic means that even slight changes in azimuth values can disproportionately impact the scene compared to changes in elevation or radius. For instance, a small change in azimuth can equate to a significant deviation, while the same change in radius or elevation would be next to negligible, resulting in difficulties in optimizing effectively solely within cylindrical space. To address these challenges, we transformed cylindrical values into Cartesian coordinates, where differences in values exert uniform influence across all dimensions during optimization. One potential explanation for the relatively inferior performance of the spatial constraint set could stem from challenges associated with optimizing joint configurations within a non-linear, high-dimensional space. Additionally, the mean value ranges of individual constraints within this set tend to be smaller compared to constraints such as *elevation* (cf video ① with approximately 200 units versus video ③ with around 10 units). Consequently, these smaller value ranges lead to correspondingly smaller deviations between the executions, resulting in smaller weights assigned to each constraint. The smaller weights incentivize the optimizer to treat all constraints nearly equally, diminishing the potential advantage of identifying and discounting irrelevant dimensions. Consequently, the optimizer may struggle to effectively prioritize and leverage relevant constraints for optimizing the overall action performance.

TABLE III

AFFORDANCE CONSTRAINT SIMILARITY FOR *rolling dough* IN *mm*

affordance region pair	type	avg	std	min
left handle of rolling pin	CaAC	**8.762**	**5.996**	**0.782**
↓	CyAC	11.802	22.036	1.552
right handle of rolling pin	SSAC	12.006	12.215	1.387
left handle of rolling pin	CaAC	134.810	72.513	21.103
↓	CyAC	134.788	87.605	25.146
place onto of table	SSAC	**115.850**	**71.209**	**17.747**
right handle of rolling pin	CaAC	135.882	75.134	21.034
↓	CyAC	136.227	90.163	29.437
place onto of table	SSAC	**118.420**	**74.384**	**15.387**

2) Rolling Dough: To ensure that our findings were not limited to the *pouring drink* action, we also trained models to perform a *rolling dough* action involving moving a rolling pin back and forth across a table five times. Our analysis of the leave-one-out cross-validation focuses on comparing the mean and variance of the difference of relative affordance region positions during execution from an unseen scene. The results of this evaluation are summarized in Table III. Similar to our previous observations, the evaluation exhibits similar values across different methods. However, it is noteworthy that the SSACs appear to yield the most accurate executions, whereas the CyACs show the least similarity. Figure 8 presents the courses of all the SSACs between *place onto* of the table and *left handle* of the rolling pin of the target demonstration, the execution, and the learned values. Remarkably, the zig-zag pattern is present, albeit at minimal amplitudes, in 6 out of the 8 SSACs. This incentivizes the optimizer to execute the motion more precisely. In comparison, only two of the three constraints manifest the zig-zag pattern in the cylindrical coordinate system (cf. video ⑦).

V. CONCLUSION AND FUTURE WORK

In this paper, we introduced a novel approach for learning spatial bimanual action models from affordance constraints observed in human demonstrations. We formulated an optimization problem that finds optimal object arrangements across multiple keypoints for the execution of a bimanual action on a humanoid robot. It takes into account the affordance constraints in the spatial bimanual action model,

the present affordance constraints in the current scene, as well as the robot's kinematics. We evaluated our approach qualitatively and quantitatively in simulation with two tasks and compared the influence of different objects, different robots, and three different affordance constraint types. The results show that given such a spatial bimanual action model, a humanoid robot is able to execute observed bimanual manipulation actions learned from human demonstration.

In future work, we aim to further improve the execution on the robot by utilizing movement primitives such as in our previous work [30]. Although the symbolic spatial affordance constraints performed worse than expected, we still believe that this representation has merit and will continue to improve its performance. Additionally, we plan to include collision avoidance between the manipulation objects, the robot, and the environment (e. g. using constraints similar to [31]) in order to validate our approach in real experiments with the humanoid robots ARMAR-6 and ARMAR-7, demonstrating the model's applicability in real-world settings. This also includes improving the performance of the approach by refining the underlying code and using gradient-based optimization methods. Furthermore, incorporating the capability to identify relevant affordance regions will allow us to execute bimanual actions in complex environments. In the long term, we want to investigate, how to combine various additional modalities other than the spatial constraints between affordance regions, such as temporal constraints between actions [32] or force constraints in our strive to create unifying manipulation task models that are learned from human demonstration.

REFERENCES

[1] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot Programming by Demonstration," in *Handbook of Robotics*, B. Siciliano and O. Khatib, Eds., 2008, pp. 1371–1394.

[2] J. J. Gibson, "The Theory of Affordances," *Hilldale, USA*, vol. 1, no. 2, pp. 67–82, 1977.

[3] M. Mansouri and F. Pecora, "A Representation for Spatial Reasoning in Robotic Planning," in *International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[4] J. O'Keefe, "Vector Grammar, Places, and the Functional Role of the Spatial Prepositions in English," in *Representing Direction in Language and Space*, E. Van Der Zee and J. Slack, Eds., 2003, pp. 69–85.

[5] R. Kartmann, D. Liu, and T. Asfour, "Semantic Scene Manipulation Based on 3D Spatial Object Relations and Language Instructions," in *International Conference on Humanoid Robots (Humanoids)*, 2021.

[6] J. Gao, X. Jin, F. Krebs, N. Jaquier, and T. Asfour, "Bi-KVIL: Keypoints-based Visual Imitation Learning of Bimanual Manipulation Tasks," in *International Conference on Robotics and Automation (ICRA)*, 2024, pp. 16 850–16 857.

[7] M. Diehl, C. Paxton, and K. Ramirez-Amaro, "Automated Generation of Robotic Planning Domains from Observations," in *International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 6732–6738.

[8] H. S. Koppula and A. Saxena, "Anticipating Human Activities Using Object Affordances for Reactive Robotic Response," *Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, vol. 38, no. 1, pp. 14–29, 2016.

[9] A. L. P. Ureche, K. Umezawa, Y. Nakamura, and A. Billard, "Task Parameterization Using Continuous Constraints Extracted From Human Demonstrations," *Transactions on Robotics (T-RO)*, vol. 31, no. 6, pp. 1458–1471, 2015.

[10] S. Niekum, S. Osentoski, G. Konidaris, and A. G. Barto, "Learning and Generalization of Complex Tasks From Unstructured Demonstrations," in *International Conference on Intelligent Robots and Systems (IROS)*, 2012, pp. 5239–5246.

[11] M. Koskinopoulou, M. Maniadakis, and P. Trahanias, "Learning Spatio-temporal Characteristics of Human Motions Through Observation," in *Advances in Service and Industrial Robotics*, N. A. Aspragathos, P. N. Koustoumpardis, and V. C. Moulianitis, Eds., 2019, vol. 67, pp. 82–90.

[12] M. Nicolescu, N. Arnold, J. Blankenburg, D. Feil-Seifer, S. Banisetty, M. Nicolescu *et al.*, "Learning of Complex-Structured Tasks from Verbal Instruction," in *International Conference on Humanoid Robots (Humanoids)*, 2019, pp. 747–754.

[13] R. Kartmann and T. Asfour, "Interactive and Incremental Learning of Spatial Object Relations from Human Demonstrations," *Frontiers in Robotics and AI*, vol. 10, 2023.

[14] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou *et al.*, "A Survey of Large Language Models," 2023, arXiv:2303.18223 [cs].

[15] OpenAI, J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya *et al.*, "GPT-4 Technical Report," 2024, arXiv:2303.08774 [cs].

[16] I. Kapelyukh, Y. Ren, I. Alzugaray, and E. Johns, "Dream2real: Zero-shot 3d object rearrangement with vision-language models," in *International Conference on Robotics and Automation (ICRA)*, 2024, pp. 4796–4803.

[17] T. Kwon, N. D. Palo, and E. Johns, "Language Models as Zero-shot Trajectory Generators," *Robotics and Automation Letters (RA-L)*, vol. 9, no. 7, pp. 6728–6735, 2024.

[18] B. Akbulut, T. Girgin, A. Mehrabi, M. Asada, E. Ugur, and E. Oztop, "Bimanual Rope Manipulation Skill Synthesis through Context Dependent Correction Policy Learning from Human Demonstration," in *International Conference on Robotics and Automation (ICRA)*, 2023, pp. 3904–3910.

[19] W. Zhai, H. Luo, J. Zhang, Y. Cao, and D. Tao, "One-shot Object Affordance Detection in the Wild," *International Journal on Computer Vision (IJCV)*, 2022.

[20] D. Hadjivelichkov, S. Zwane, L. Agapito, M. P. Deisenroth, and D. Kanoulas, "One-Shot Transfer of Affordance Regions? AffCorrs!" in *Conference on Robot Learning (CoRL)*, 2023, pp. 550–560.

[21] W. Qu, X. Li, and X. Jin, "Knowledge Enhanced Bottom-up Affordance Grounding for Robotic Interaction," *PeerJ Computer Science*, vol. 10, 2024.

[22] T. Nguyen, M. N. Vu, B. Huang, T. Van Vo, V. Truong, N. Le *et al.*, "Language-conditioned affordance-pose detection in 3d point clouds," in *International Conference on Robotics and Automation (ICRA)*, 2024, pp. 3071–3078.

[23] A. Delitzas, A. Takmaz, F. Tombari, R. Sumner, M. Pollefeys, and F. Engelmann, "SceneFun3D: Fine-grained Functionality and Affordance Understanding in 3D Scenes," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2024.

[24] Y. Wang, L. Zhang, Y. Tu, H. Zhang, K. Bai, Z. Chen *et al.*, "ToolEENet: Tool Affordance 6D Pose Estimation," 2024, arXiv:2404.04193 [cs].

[25] P. Ardon, E. Pairet, K. S. Lohan, S. Ramamoorthy, and R. P. A. Petrick, "Affordances in Robotic Tasks – A Survey," 2020, arXiv:2004.07400 [cs].

[26] T. Giorgino, "Computing and Visualizing Dynamic Time Warping Alignments in R: The dtw Package," *Journal of Statistical Software*, vol. 31, no. 7, pp. 1–24, 2009.

[27] A. Meixner, M. Carl, F. Krebs, N. Jaquier, and T. Asfour, "Towards Unifying Human Likeness: Evaluating Metrics for Human-Like Motion Retargeting on Bimanual Manipulation Tasks," in *International Conference on Robotics and Automation (ICRA)*, 2024, pp. 13 015–13 022.

[28] F. Gao and L. Han, "Implementing the Nelder-Mead Simplex Algorithm With Adaptive Parameters," *Computational Optimization and Applications*, vol. 51, no. 1, pp. 259–277, 2012.

[29] F. Krebs, A. Meixner, I. Patzer, and T. Asfour, "The KIT Bimanual Manipulation Dataset," in *International Conference on Humanoid Robots (Humanoids)*, 2021, pp. 499–506.

[30] Y. Zhou, J. Gao, and T. Asfour, "Learning Via-Point Movement Primitives with Inter- and Extrapolation Capabilities," in *International Conference on Intelligent Robots and Systems (IROS)*, 2019, pp. 4301–4308.

[31] D. Rakita, H. Shi, B. Mutlu, and M. Gleicher, "Collisionik: A per-instant pose optimization method for generating robot motions with environment collision avoidance," in *International Conference on Robotics and Automation (ICRA)*, 2021, pp. 9995–10 001.

[32] C. Dreher and T. Asfour, "Learning Symbolic and Subsymbolic Temporal Task Constraints from Bimanual Human Demonstrations," in *International Conference on Intelligent Robots and Systems (IROS)*, 2024.