Semantic Scene Manipulation Based on 3D Spatial Object Relations and Language Instructions

Rainer Kartmann, Danqing Liu and Tamim Asfour

***Abstract*— Robot understanding of spatial object relations is key for a symbiotic human-robot interaction. Understanding the meaning of such relations between objects in a current scene and target relations specified in natural language commands is essential for the generation of robot manipulation action goals to change the scene by relocating objects relative to each other to fulfill the desired spatial relations. This ability requires a representation of spatial relations, which maps spatial relation symbols extracted from language instructions to subsymbolic object goal locations in the world. We present a generative model of static and dynamic 3D spatial relations between multiple reference objects. The model is based on a parametric probability distribution defined in cylindrical coordinates and is learned from examples provided by humans manipulating a scene in the real world. We demonstrate the ability of our representation to generate suitable object goal positions for a pick-and-place task on a humanoid robot, where object relations specified in natural language commands are extracted, object goal positions are determined and used for parametrizing the actions needed to transfer a given scene into a new one that fulfills the specified relations.**

I. Introduction

Natural language is a powerful interface for instructing robots to perform tasks such as manipulating objects or navigating a building. Spatial object relations play an important role in communicating the goals or parameters of actions in a human understandable way such as in natural language to the robot, e. g. where to go, which object to pick or where to put it. A key challenge in understanding natural language commands is *grounding* them in concepts of the robot's scene model, i. e. mapping words and phrases to physical entities and locations perceived by the robot [1]. In language understanding, spatial relations are frequently used to resolve referring expressions used in object descriptions, such as in "Pick up the box in front of the table" [2], [3], [4], [5] and are used to find and disambiguate discrete action parameters, such as the object to pick or the landmark to place it at.

Our aim is to endow a robot with the ability to transfer a given scene with a set of objects with certain spatial relations into a new scene fulfilling new spatial relations specified by a natural language command. The overall idea is depicted in Fig. 1. However, when giving instructions such as "Put the apple tea on top of the mint tea" and "Place the plate between the fork and the knife," the set of possible goal locations is not limited to a finite set of landmarks or objects, but could be any position in the workspace of the robot. Therefore, it is not only necessary to ground language phrases referring to objects in the robot's scene model and the spatial relation phrase to a symbol, but also to map this symbol to an arbitrary, i. e. subsymbolic, goal position.

Fig. 1: Semantic scene manipulation based on language commands specifying 3D spatial object relations.

To address this problem, we introduce a parametric and probabilistic generative model for representing 3D spatial relations between multiple objects in a scene. Given an initial scene and a desired spatial relation, the model generates object locations where the object can be placed to fulfill the relation. Thus, the model grounds a spatial relation extracted from a language command in a subsymbolic object goal position. Such a model should have only few parameters which can be intuitively interpreted. A generative model has the benefit that appropriate goal locations (and thus, actions) can directly be sampled from the model instead of sampling and classifying the whole workspace [6]. Following a learning from demonstration paradigm [7], [8], our model is learned from pairwise scene examples that are generated by a human demonstrator manipulating a scene in the real world.

We build on our previous work on 2D spatial relations [9] by extending the model to deal with 3D positions and relations and with two or more reference objects (e. g. A *between* B and C). We consider dynamic spatial relations, which depend on the relative position of objects in the initial scene as well as static relations, which are independent of the relative object positions.

This work has been supported by the German Federal Ministry of Education and Research (BMBF) under the project OML (01IS18040A).

The authors are with the Institute for Anthropomatics and Robotics, Karlsruhe Institute of Technology, Karlsruhe, Germany. {rainer.kartmann, asfour}@kit.edu

II. Related Work

We review work related to spatial relations in the following domains: language understanding, scene understanding including human action recognition, and scene manipulation.

A. Spatial Relations in Language Understanding

In a recent survey, Tellex et al. [1] reviewed the field of using language in robotics. The authors classify works according to the technical approach and the addressed problem. As to the technical approach, our work falls into the category of lexically-grounded methods, as we extract a formal representation of the task (the spatial relations to fulfill and the involved objects) from a language command, which is then grounded to physical placing locations in the scene for execution.

With respect to the addressed problem, Tellex et al. distinguish between (1) human-to-robot communication (mainly concerned with language *understanding*) and (2) robot-to-human communication (dealing with language *generation*). Our work falls into (1), as we extract goal information from a language command to manipulate the current scene. Furthermore, this class contains works on using language to (1a) give robots instructions and (1b) to inform robots about the world. Even when using spatial relations in commands, their main purpose is currently (1b), i. e. describing objects or known landmarks.

Tan et al. [10] use a simple grammar and spatial relations represented as fuzzy memberships to nine image regions around an object to identify a referred object in a command. In [11], spatial relations in daily human instructions are parsed using a syntactic parser and used to learn attributes of new objects. These works do not involve a robot executing actions. Fasola et al. [6] represent 2D spatial prepositions by semantic fields to resolve ambiguous noun phrase groundings in language navigation instructions. Hatori et al. [12] jointly train object recognition and language understanding modules to resolve unconstrained referring expressions to objects and query additional expressions in case of ambiguities. The probabilistic graphical model proposed by Tellex et al. [2] maps constituents of a language command to objects, places and robot actions. Forbes et al. [13] find the most suitable parameters for a set of primitive actions by using language generation to resolve referring expressions given the language command and current context in a programming by demonstration setup. Prepositions from natural language commands are incorporated as action parameters in a task representation based on part-of-speech tagging in [14]. However, in these works action parameters are limited to a finite set of values and position offsets are fixed. Paul et al. [4], [15] build on extensions of [2] to ground abstract spatial concepts such as columns or groups of objects for language understanding. Shridhar et al. present in [5] and [16] a framework to ground referring expressions to objects in an image, using spatial relations to resolve ambiguities when referring to visually similar objects. Placement positions are found based on spatial prepositions in [3] by training a multi-class logistic regression to classify random positions on a table. These works employ discriminative models of spatial relations to identify referred objects, actions or parameters from a finite set of options. In contrast, the problem addressed in this work requires a generative model of spatial relations allowing to place objects at any location without limitation to known entities.

B. Spatial Relations for Scene Understanding

Spatial relations have been extracted to build semantic scene understanding by classifying the displacement between two objects to distinguish between *on* and *adjacent* [17], learning to classify functional relations from geometric object features in simulation [18], or extracting spatial relations from 3D vision using histograms [19]. Yan et al. [20] employ a neural-logic network trained with loss functions representing logic rules to predict fundamental spatial relations and infer complex spatial relations from them.

For human activity recognition, Lee et al. [21] extract spatial relations from an RGB-D video according to three calculi modeling contact, relative movement and static distance by applying a Dynamic Bayesian Network (DBN) to hand-defined key metrics computed from point cloud masks. While their focus is on the relation extraction and the human activity recognition itself uses simple rules, other authors have employed simpler relation extraction models but more sophisticated action recognition models. (Enriched) Semantic Event Chains encode changes in spatial relations between objects in RGB-D videos and have been used to segment and recognize actions based on distance measures in [22], [23], [24], [25], [26]. Dreher et al. [27] used the same spatial relations model but employed a graph neural network to perform bimanual action recognition.

C. Spatial Relations for Scene Manipulation

The following works focus on changing a scene with the goal of fulfilling given spatial relations. Fasola et al. [6] represent 2D paths as combinations of static spatial prepositions modeled by semantic fields. Although they focus on navigation, their representation could be extended to object manipulation. Mees et al. [28] use metric learning to realize spatial relations from known scene examples with potentially different objects. Jund et al. [29] perform gradient descend on object poses in a scene to create the same relation as in a reference scene. In our work, we derive generative models of spatial relations which abstract from scene examples. Furthermore, the works above only consider static relations. In [30], Mees et al. train a neural network to predict pixel-wise probability maps of placement positions given an input image of the scene and an object to be placed according to a spatial relation in a language command. In contrast to a neural network, our representations based on parametric probability distributions can be intuitively interpreted and estimated from few examples. In addition, we are not restricted to an input image.

III. Problem Formulation

Given a scene with a set of objects and their spatial relations encoded in a semantic scene model together with a language command specifying new spatial relations between these objects, a robot must transfer the initial scene into a new scene fulfilling the specified relations between the same objects by executing actions (see Fig. 2). To this end, the robot must extract the

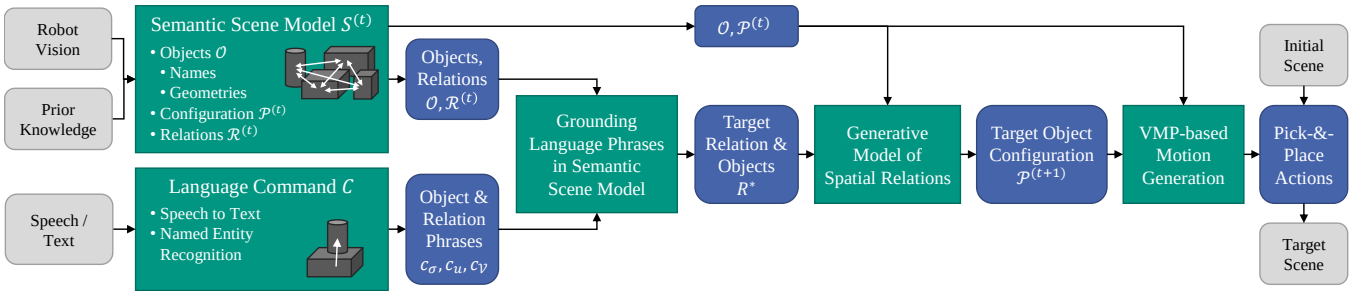


Fig. 2: We pose the problem of transferring a scene $S^{(t_0)}$ to a new scene $S^{(t_1)}$ fulfilling spatial relations R^* between objects which are specified by a language command C.

specified relations and objects from the language command, then find and execute appropriate actions leading to the desired spatial relations.

The semantic scene model contains the configuration of n objects as well as prior knowledge. The configuration $\mathcal{P} = \{P_1, \dots, P_n\}$ consists of the pose $P_i \in \mathrm{SE}(3)$ of each object i with position $\mathbf{p}_i$ and orientation $\mathbf{q}_i$ (as quaternion). The prior knowledge $O_i = (g_i, \eta_i)$ of object i contains the object geometry g_i and a human-readable name η_i. In addition, the scene model contains the spatial relations $\mathcal{R} = \{R_1, \dots, R_m\}$ in the scene. Each relation $R_j = (\sigma_j, u_j, \mathcal{V}_j)$ has a symbol $\sigma_j \in \Sigma$, a *subject* object u_j and a set of *reference* objects $\mathcal{V}_j$ ($u_j \notin \mathcal{V}_j$). Σ is the set of known spatial relation symbols. Therefore, a scene

$$S = (\mathcal{O}, \mathcal{P}, \mathcal{R}) \tag{1}$$

is defined by the prior object information $\mathcal{O} = \{O_1, \dots, O_n\}$, configuration $\mathcal{P}$ and spatial relations $\mathcal{R}$.

A language command is an imperative text sentence C (potentially derived from speech via a speech recognition system) containing object phrases referring to the subject and reference objects as well as the spatial relations between them. In lexically-grounded language understanding [1], the relation and object phrases $c_\sigma, c_u, c_\mathcal{V}$ are parsed from the command,

$$(c_\sigma, c_u, c_\mathcal{V}) \leftarrow \mathrm{parse}(C), \quad \{c_{v_k}\}_{k=1}^{K} \tag{2}$$

where K is the number of reference objects. Then, the extracted phrases (which are still text) must be grounded in the objects $\mathcal{O}$ in the current scene $S^{(t_0)}$ and the known relation symbols Σ, specifying the desired relation R^*,

$$R^* = (\sigma^*, u^*, \mathcal{V}^*) \leftarrow \mathrm{ground}\left(S^{(t_0)}, c_\sigma, c_u, \{c_{v_k}\}_{k=1}^{K}\right).$$

We will refer to u^* as the *target object*. The robot's task is to transfer the current scene $S^{(t_0)}$ to a target scene

$$S^{(t_1)} = \left(\mathcal{O}, \mathcal{P}^{(t_1)}, \mathcal{R}^{(t_1)}\right) \tag{3}$$

with $R^* \in \mathcal{R}^{(t_1)}$. To achieve this, the robot must manipulate the initial object configuration $\mathcal{P}^{(t_0)}$ by executing actions $\mathcal{A}$,

$$\mathcal{P}^{(t_0)} \xrightarrow{\mathcal{A}} \mathcal{P}^{(t_1)} \quad \text{such that } R^* \in \mathcal{R}^{(t_1)}. \tag{4}$$

In this work, we only manipulate the target object by executing pick&place actions. Therefore, the problem is reduced to find a suitable target pose $P_{u^*}^{(t_1)}$ to place u^* at.

IV. Approach

We approach the problem formulated in the last section as follows. Based on prior knowledge and visual data of the scene, we build a 3D semantic scene model $S^{(t_0)}$ of the initial scene using prior knowledge and state-of-the-art methods for visual pose estimation of known objects [31], [32] to localize objects in the current scene. We parse the language command with a Named Entity Recognition model (Section IV-A) and ground the extracted phrases by performing substring matching with names of known objects and relations (Section IV-B). Using a generative model of the spatial relation (Section IV-C), we generate a suitable target object pose $P_{u^*}^{(t_1)}$ fulfilling the desired relation R^* (Section V).

A. Extracting the Target Relation from Language Command

Given a speech command such as "Place the apple tea between the juice and the milk," we apply an existing speech recognition system used on our ARMAR-6 robot [33] to obtain the command text C. Following the formulation in Section III, we then extract the object and relation phrases using a state-of-the-art Named Entity Recognition (NER) model. NER classifies parts of a sentence into several class labels, allowing multiple occurrences of the same class label. We define three class labels, `REL`, `TRG`, and `REF`, for spatial relation symbol σ^*, target object u^*, and reference object(s) $\mathcal{V}^*$, respectively.

To train the NER model, we generate language commands based on sentence templates. For object phrases (`TRG` and `REF`), we use the names of all objects from the KIT and YCB object data bases ([34], [35]) after processing them to be human readable and unambiguous. For relation phrases (`REL`), we generate phrases for the relations *left*, *right*, *front*, *behind*, *close to*, *inside*, *on top*, *above*, *under*, *closer*, *farther*, *other side*, *between* and *among* which we manually defined.

To generate a command, a spatial relation phrase, a target object and reference objects are selected at random and inserted into a suitable sentence template, resulting in commands such as "Place the orange juice on top of the potato starch" and "Put apple tea between the coffee filters and the orange juice." The commands are labeled by the selected template inputs. Thus, we generated labeled training data and used them to train a standard NER model from the natural language processing library spaCy (`https://spacy.io/`).

Given the input command sentence, we apply the trained NER model and get a list of sub-phrases for each class label.

We assume to retrieve exactly one phrase each for REL and TRG and one or more for REF. For instance, for the previous example sentence, "Place the apple tea between the juice and the milk," we obtain the phrases $c_\sigma =$ "between", $c_u =$ "apple tea" and $c_\mathcal{V} = \{$"juice", "milk"$\}$.

B. Grounding Command Phrases in Semantic Scene Model

Having obtained the object and relation phrases from the language command, we ground it to the objects in the semantic scene model and the known relations using substring matching. For the target object, we find the longest common substring s_i of c_u and each object name η_i,

$$s_i \leftarrow \underset{s \in c_u \cap \eta_i}{\arg\max} |s|\,, \quad (i = 1, \ldots, n) \tag{5}$$

where $c_u \cap \eta_i$ denotes all commons substrings of c_u and η_i. We sort the matched substrings $\{s_1, \ldots, s_n\}$ decreasingly by their length and discard those which are shorter than a threshold (e. g. 3 characters). The result is a list of object hypotheses in decreasing order of their names' conformity with target object phrase. The same is done for each $c_{v_k} \in c_\mathcal{V}$ as well as the relation phrase c_σ (based on the relation phrases used in command generation).

Despite its simplicity, this kind of substring matching has two benefits. First, it implicitly handles minor errors of the NER model such as "the" being part of an object phrase "the tea", as the phrase will still match the object names "apple tea" and "peppermint tea". Second, it allows using a general term such as "juice" with specific names, such as "orange juice" and "apple juice". In general, we choose the relation and objects with the longest matches. If multiple objects match equally well, we pick one at random. Although substring matching could be applied to the whole command, the NER model focuses it on relevant phrases and provides the label for each phrase.

C. Generative Model of 3D Spatial Relations

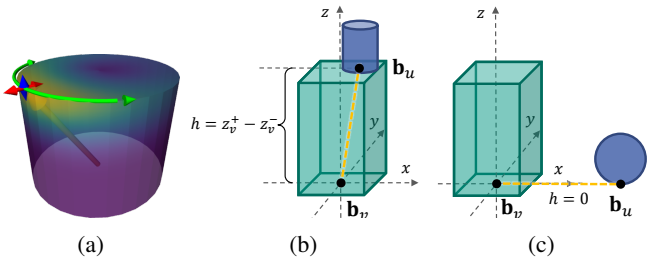
After obtaining $R^* = (\sigma^*, u^*, \mathcal{V}^*)$ from the language understanding components, the next step is finding an appropriate object pose $P_{u^*}^{(t_1)}$ to place the target object u^* at. To this end, we propose a generative model G of spatial relations, which can directly generate suitable target object poses fulfilling a relation $R = (\sigma, u, \mathcal{V})$ based on the current scene

$$P_u \sim G_\sigma\left(u, \mathcal{V}, \mathcal{O}, \mathcal{P}^{(t_0)}\right). \tag{6}$$

With a generative model G, appropriate target poses (and thus, actions) can directly be sampled from G instead of sampling and classifying the whole workspace. In addition, it still allows to discard hypotheses, which are infeasible due to other external constraints, such as collisions or reachability. In addition, our model can be learned from examples in form of pairs of scenes before and after a change induced by humans.

In order to prevent object u^* from tipping after it was placed, we keep its initial orientation in the target scene, $\mathbf{q}_{u^*}^{(t_1)} = \mathbf{q}_{u^*}^{(t_0)}$. Therefore, the problem of generating $P_{u^*}^{(t_1)}$ is reduced to generating a suitable target position $\mathbf{p}_{u^*}^{(t_1)}$.

In the following, we describe the methods for representing static and dynamic 3D spatial relations and how they are extended to multiple reference objects.



(a) (b) (c)

Fig. 3: (a) Visualization of a cylindrical distribution and its parameters (best viewed in color). The orange arrow runs from the origin to the distribution mean. The cylinder surface is a cross section at the mean radius μ_r (mantle) and mean height μ_h (top), color indicates PDF ranging from 0 (purple) to the maximum PDF at the mean (yellow). Red arrows represent the radius variance, blue arrows represent height variance (radius and height are not covariant in this example), green arrows represent azimuth concentration. (b, c) Using the bottom-projected centroid of objects as reference positions helps to generalize to reference objects of different size.

*1) **3D Spatial Relations as Cylindrical Distributions**:* In our previous work [9], we proposed to represent 2D spatial relations in a horizontal plane (e. g. a tabletop surface) as parametric probability distributions defined in a polar coordinate system (PCS). A polar distribution's probability density function has been defined as

$$p_\theta(r, \phi) = p_{\theta_r}(r) \cdot p_{\theta_\phi}(\phi)\,, \quad \theta = (\theta_r, \theta_\phi) \tag{7}$$

over radial distance $r \in \mathbb{R}_{\geq 0}$ (denoted as distance d in [9]) and azimuthal angle $\phi \in [-\pi, \pi]$, where

$$r \sim \mathcal{N}(\theta_r)\,, \quad \theta_r = \left(\mu_r, \sigma_r^2\right), \tag{8}$$

$$\phi \sim \mathcal{M}(\theta_\phi)\,, \quad \theta_\phi = (\mu_\phi, \kappa_\phi) \tag{9}$$

with mean radius μ_r and radius variance σ_r^2 as well as mean azimuth μ_ϕ and azimuth concentration κ_ϕ. $\mathcal{N}(\cdot)$ denotes a Gaussian while $\mathcal{M}(\cdot)$ denotes a von Mises distribution, which is a circular distribution wrapping over $[-\pi, \pi]$. The reference PCS is centered at the reference object's geometric center, so the polar coordinate (r, ϕ) represents the target object's position relative to the reference object.

To extend the previous model to 3D space, we introduce the height $h \in \mathbb{R}$ as an additional dimension of the probability distribution's definition space. Effectively, this transforms the PCS over (r, ϕ) to a cylindrical coordinate system (CCS) over (r, ϕ, h). As both r and h are distances, we assume that they follow a joint Gaussian distribution

$$(r, h) \sim \mathcal{N}(\theta_{rh})\,, \quad \theta_{rh} = (\mu_{rh}, \Sigma_{rh}) \tag{10}$$

with means $\mu_{rh} = (\mu_r, \mu_h)^\top \in \mathbb{R}^2$ and covariance matrix $\Sigma_{rh} \in \mathbb{R}^{2\times 2}$. For the azimuth, we adopt its von Mises distribution from eq. (9). We also adopt the assumption that (r, h) are independent of ϕ. This assumption is motivated by the observation that spatial relations are typically either a statement about direction (e. g. *left, behind, above*) or about distance (e. g. *close to, far away from*) [36]. Hence, we define a cylindrical distribution as joint probability distribution

$$(r, \phi, h) \sim \mathcal{C}(\theta)\,, \quad \theta = (\theta_{rh}, \theta_\phi) \tag{11}$$

over radius $r \in \mathbb{R}_{\geq 0}$, azimuth $\phi \in [-\pi, \pi]$ and height $h \in \mathbb{R}$ with the joint probability density function (PDF)

$$p_\theta(r, \phi, h) = p_{\theta_{rh}}(r, h) \cdot p_{\theta_\phi}(\phi) \tag{12}$$

with $p_{\theta_\phi}(\phi)$ and $p_{\theta_{rh}}(r, h)$ according to eq. (9) and (10). An example distribution and its parameters is visualized in Fig. 3a.

To sample and evaluate positions in Cartesian world coordinates, they need to be transformed from and to the relation's CCS. The Cartesian world coordinate system (WCS) is aligned to the agent such that $+x$ points to the right and $+z$ points up (i. e. opposite of the vector of gravity), while the relation's CCS is aligned to the reference objects $\mathcal{V}$. For simplicity, we first consider the case of a single reference object $v = v_1$. The alignment of the CCS is based on the reference object's geometry g_v in the WCS. Let

$$B_v^{(t_0)} = [x_v^-, x_v^+] \times [y_v^-, y_v^+] \times [z_v^-, z_v^+] \subset \mathbb{R}^3. \tag{13}$$

be the axis-aligned bounding box (AABB) enclosing g_v at its position $\mathbf{p}_v^{(t_0)}$ in the WCS. The bottom-projected centroid

$$\mathbf{b}_v^{(t_0)} = \left(\frac{x_v^- + x_v^+}{2}, \frac{y_v^- + y_v^+}{2}, z_v^- \right)^\top \in \mathbb{R}^3 \tag{14}$$

of $B_v^{(t_0)}$ represents the reference position of v. At the same time, it is the origin of the relation's CCS. Likewise, the reference position of the subject object u is the bottom-projected centroid $\mathbf{b}_{g_u}^{(t_1)}$ of its own AABB $B_u^{(t_1)}$ at a potential target position $\mathbf{p}_u^{(t_1)}$.

This choice complies with our previous formulation of 2D spatial relations, with the origin at the height of the surface the objects are standing on. A useful property is that two objects standing on the same horizontal surface have the same z-coordinate (height), independent of their respective size.

As in [9], we distinguish between static and dynamic spatial relations. The geometric meaning of static relations such as *left of* or *on top of* mainly depends on the reference objects, while the meaning of dynamic relations such as *closer to* and *on the other side of* depends on the initial configuration of both the subject and reference objects.

In our prior work, we have shown that it is effective to apply different transformations depending the relation's type to incorporate these dependencies into the CCS. Here, we extend these transformations to 3D. We will refer to

$$\mathbf{p}_{\text{loc}}^{(t_1)}(u, v) = \left(x_{\text{loc}}^{(t_1)}, y_{\text{loc}}^{(t_1)}, z_{\text{loc}}^{(t_1)} \right)^\top = \mathbf{b}_u^{(t_1)} - \mathbf{b}_v^{(t_0)} \tag{15}$$

as u's target position in the local WCS, i. e. relative to v.

*2) **Static Spatial Relations**:* The local target position is scaled proportionally to the reference object's size before transforming it to cylindrical coordinates:

$$\mathbf{p}_{\text{sta}}^{(t_1)}(u, v) = \left(\frac{2\ x_{\text{loc}}^{(t_1)}}{x_v^+ - x_v^-}, \frac{2\ y_{\text{loc}}^{(t_1)}}{y_v^+ - y_v^-}, \frac{z_{\text{loc}}^{(t_1)}}{z_v^+ - z_v^-} \right)^\top \tag{16}$$

This way, if the target object is standing on top of the reference object, $z_{\text{sta}} \approx 1$, independent of the respective object sizes (Fig. 3b). If they are standing on the same horizontal surface, such as a tabletop or shelf, $z_{\text{sta}} \approx 0$ (Fig. 3c). This design helps to generalize spatial relations over objects of different size. Finally, the local scaled world coordinate is converted to cylindrical coordinates

$$\mathbf{c}_{\text{sta}}^{(t_1)}(u, v) = \text{cylindrical}\left(\mathbf{p}_{\text{sta}}^{(t_1)}(u, v) \right) \tag{17}$$

using

$$\text{cylindrical}\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} r \\ \phi \\ h \end{pmatrix} = \begin{pmatrix} \sqrt{x^2 + y^2} \\ \text{atan2}(y, x) \\ z \end{pmatrix}. \tag{18}$$

*3) **Dynamic Spatial Relations**:* Let

$$\left(r^{(t_0)}, \phi^{(t_0)}, h^{(t_0)} \right)^\top = \text{cylindrical}\left(\mathbf{p}_{\text{loc}}^{(t_0)}(u, v) \right) \tag{19}$$

be u's *initial* local position in cylindrical coordinates. For the dynamic cylindrical coordinate

$$\mathbf{c}_{\text{dyn}}^{(t_1)}(u, v) = \left(r_{\text{dyn}}, \phi_{\text{dyn}}, h_{\text{dyn}} \right)^\top, \tag{20}$$

we want to obtain three properties: (i) $r_{\text{dyn}} = 1$ shall correspond to the current radial distance $r^{(t_0)}$ between v and u, (ii) $\phi_{\text{dyn}} = 0$ shall correspond to the current direction $\phi^{(t_0)}$ from v to u, and (iii) $h_{\text{dyn}} = 1$ shall correspond to the current height difference $h^{(t_0)}$ in units of the reference object's height. These properties help generalizing dynamic spatial relations to different initial object configurations [9]. To achieve them, we adopt the alignment of the polar part of the cylindrical coordinates (r, ϕ) from [9] and extend it to the height parameter h by aligning a potential target cylindrical coordinate

$$\left(r^{(t_1)}, \phi^{(t_1)}, h^{(t_1)} \right)^\top = \text{cylindrical}\left(\mathbf{p}_{\text{loc}}^{(t_1)}(u, v) \right) \tag{21}$$

to the initial configuration as follows:

$$\begin{pmatrix} r_{\text{dyn}} \\ \phi_{\text{dyn}} \\ h_{\text{dyn}} \end{pmatrix} = \begin{pmatrix} r^{(t_1)} / r^{(t_0)} \\ \phi^{(t_1)} - \phi_{(t_0)} \\ (h^{(t_1)} - h^{(t_0)}) / \left(z_v^+ - z_v^- \right) \end{pmatrix} \tag{22}$$

*4) **Extension to Multiple Reference Objects**:* To extend to multiple reference objects ($|\mathcal{V}| > 1$), we adopt the notion of *abstract* objects from [4]. In their work, the authors have used abstract concepts such as "row", "column" and "group" in order to ground referential expressions in natural language instructions such as "pick up the middle block from this row of blocks" to colored cubes in an image. Here, we conflate all reference objects into one abstract object and apply the logic for a single object explained above. As the formulations above are based on the AABB $B_v^{(t_0)}$, we can easily extend them from v to $\mathcal{V}$ by computing the AABB $B_{\mathcal{V}}^{(t_0)}$ enclosing all reference objects $v_k \in \mathcal{V}$. This way, we can apply the same spatial relations to single and multiple reference objects without significantly changing the mathematical formulation.

V. Learning Generative Models of 3D Spatial Relations

A. Data Collection in Real World Setup

To learn generative models of spatial relations, we collected a few samples for each spatial relation. As we are using spatial relations to change an initial scene to a target scene, a sample minimally consists of the initial scene $S^{(t_0)}$, the target scene $S^{(t_1)}$ and the relation R^* which has generated the scene

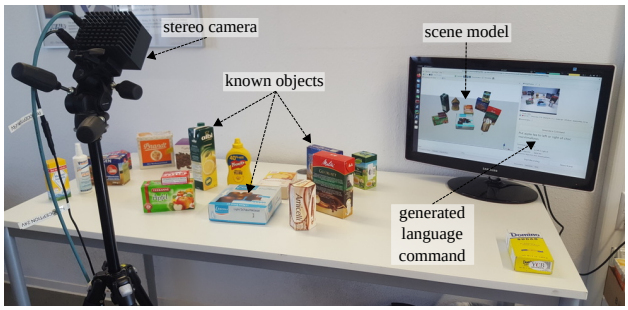


Fig. 4: Real world data collection setup for data collection.

change (which includes the involved objects). In contrast to our previous work, where we collected data on a 2D monitor, for this work, we collected data in a real world demonstration setup shown in Fig. 4. The setup consists of a tabletop scene with multiple known objects from the KIT and YCB object databases [34], [35], a stereo camera and a monitor displaying generated language instructions, which the demonstrator should follow. We calibrated the camera externally to an agent-aligned world coordinate frame as described above. We localize the objects in the scene based on the camera images. The current (left) camera image and scene model are displayed to the human demonstrator on the monitor.

We generate language commands in the same way as to train the NER model (Section IV-A). Here, the object phrases are limited to the currently localized objects. As this work focuses on pick-and-place tasks, we limit the `REF` phrases to the spatial relations, which can be created by relocating a single object in a scene (e. g. we do not generate sentences like "Place the tea *under* the coffee," which would require lifting the coffee first). Reference objects, a target object and a spatial relation are selected randomly and inserted into a suitable sentence template, which is then shown to the demonstrator on the monitor. The demonstrator can then record the initial scene, perform the manipulation, and record the changed scene. To create diverse target scenes, the demonstrator can change and record the scene multiple times (all target scenes are relative to the same initial scene). Eventually, the demonstrator can request the next command. The result is a collection of samples $\mathcal{D}_\sigma$ for each spatial relation $\sigma \in \Sigma$, where each sample consists of an initial and a target scene:

$$\mathcal{D}_\sigma = \left\{ S_k^{(t_0)}, S_k^{(t_1)} \right\}_{k=1}^{K_\sigma} \quad (\sigma \in \Sigma) \tag{23}$$

B. Estimation of Cylindrical Distributions

In our model, each 3D spatial relation σ is represented by a single cylindrical distribution $\mathcal{C}_\sigma$. Therefore, we estimate one cylindrical distribution based on the samples $\mathcal{D}_\sigma$ of each $\sigma \in \Sigma$. For each target scene sample k, we transform the target object's pose to cylindrical coordinates according to its type as defined in eq. (17) and (22).

$$\left(r^k, \phi^k, h^k\right)^\top \leftarrow \begin{cases} \mathbf{c}_{\text{sta}}^{(t_1)}(u,v)\,, & \text{static} \\ \mathbf{c}_{\text{dyn}}^{(t_1)}(u,v)\,, & \text{dynamic} \end{cases} \tag{24}$$

The relations' types (static or dynamic) are predefined. We then perform Maximum Likelihood Estimation (MLE) separately for (1) radius and height and (2) the azimuth according to the cylindrical distribution's structure in eq. (12)

$$\theta_{rh}^* \leftarrow \text{MLE}_{\mathcal{N}}\left(\left\{r^k, h^k\right\}_{k=1}^{K_\sigma}\right), \tag{25}$$

$$\theta_{\phi}^* \leftarrow \text{MLE}_{\mathcal{M}}\left(\left\{\phi^k\right\}_{k=1}^{K_\sigma}\right). \tag{26}$$

With $\theta = (\theta_{rh}, \theta_\phi)$ as in eq. (11), this defines the cylindrical distribution.

C. Sampling and Selection of Target Position

Given the desired relation $R^* = (\sigma^*, u^*, \mathcal{V}^*)$, the previously estimated cylindrical distribution $\mathcal{C}_{\sigma^*}(\theta)$ representing σ^* as well as the initial scene $S^{(t_0)}$, we find a target position $\mathbf{p}_{u^*}^{(t_1)}$ by sampling from $\mathcal{C}_{\sigma^*}(\theta)$ and selecting the best feasible hypotheses. First, we sample a predefined number of cylindrical coordinates

$$\mathbf{c}_k = (r_k, \phi_k, h_k) \sim \mathcal{C}_{\sigma^*}(\theta)\,, \quad (k = 1, \ldots, K). \tag{27}$$

Using the initial scene $S^{(t_0)}$ and type of σ^*, we transform the cylindrical coordinates back to Cartesian space by inverting the steps described above

$$\mathbf{p}_k = (x_k, y_k, z_k) \leftarrow \text{cartesian}\left(\mathbf{c}_k, S^{(t_0)}, \sigma^*\right). \tag{28}$$

These are our initial hypotheses for $\mathbf{p}_{u^*}^{(t_1)}$. To select a final hypothesis for execution, we use a similar approach as in [9]. To ensure that the resulting scene is collision-free and statically stable, we filter the hypotheses using collision and stability checks [37]. Let $H_{\text{feas}} \subseteq \{1, \ldots, K\}$ be the feasible hypotheses. To find the best action, we evaluate the PDF of each hypothesis and keep those with the highest values:

$$H_{\text{best}} = \left\{ k \in H_{\text{feas}} \;\middle|\; p_\theta(\mathbf{c}_k) \geq 0.9 \cdot \max_{l \in I_{\text{feas}}} p_\theta(\mathbf{c}_l) \right\} \tag{29}$$

To avoid unnecessary movement, we select the position $\mathbf{p}_{k^*}$ from the remaining hypotheses which is closest to the target object's current position

$$k^* = \underset{k \in H_{\text{best}}}{\arg\min} \left\| \mathbf{p}_k - \mathbf{p}_{u^*}^{(t_0)} \right\|, \quad \text{thus } \mathbf{p}_{u^*}^{(t_1)} = \mathbf{p}_{k^*}. \tag{30}$$

VI. Evaluation

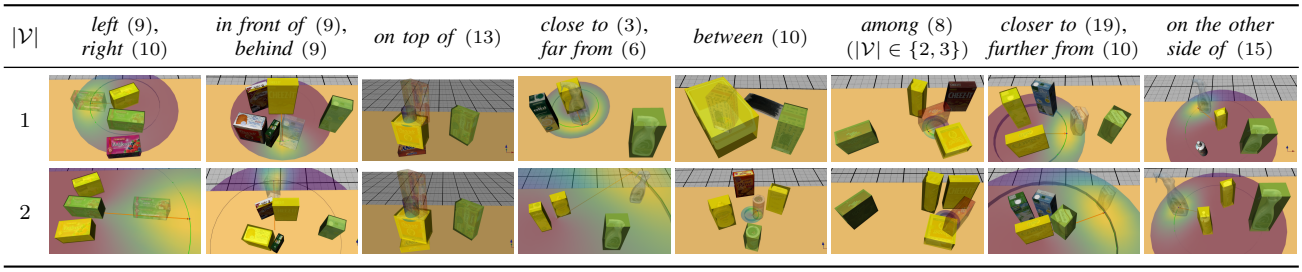
A. Qualitative Analysis of 3D Spatial Relation Representations

Table I shows examples using the learned representations of different spatial relations to select a placing position for an object in simulated scenes. The first observation is that we inherit the representation power of the polar distribution from [9], as "flat" relations such as *left* or *closer to* can easily be represented as cylindrical distribution with a mean height close to zero. This is a result of using the bottom-projected centroids of objects as reference positions. Second, as can be seen in the examples for the relations *in front of* and *behind*, collision-freeness and static stability can be achieved by discarding samples violating any external constraints. Third, the relation *on top* can now be represented as well (with $\mu_h = 0.939 \approx 1$), corresponding roughly to the vertical size of reference object(s). Due to the natural fuzziness of our representation, the relation *among* can also generate positions on top of another object if the tabletop surface is obstructed.

TABLE I

ESTIMATED CYLINDRICAL DISTRIBUTIONS USED TO SELECT PLACING POSITIONS IN SIMULATED SCENES.

The number of samples used to estimate each distribution is given in parentheses. The target object is marked green, reference objects are yellow. The target object is drawn transparently at a selected placing position. In the first row, one reference object is specified, while two are used in the second row (for *among*, two and three have been specified, respectively). The distributions are visualized similarly to Figure 3a, with the cylinder tops (colored disks) spanning $1.5\mu_r$ to completely show the distributions' high-likelihood areas.



(a) Initial scene (b) Grasp (c) VMP placing motion (d) Place and release (e) Retreat

Fig. 5: Changing a scene according to the command "Place the mustard on the rusk." on the humanoid robot ARMAR-6.

Finally, we can specify more than one reference object as easily as one for all relations. For instance, *on the other side of* generates a position on the opposite side of the mid point between the two reference objects. Similarly, *on top of* yields similar results when generating positions relative to a stack of objects or just the highest object of the stack. A very interesting observation is that using *between* with a container as single reference object (which never occurred in the demonstrated data) results in a behavior expected from an *inside* relation (which was demonstrated as containers were not used in data collection). This is plausible, as *between* behaves like an *inside* for the abstract object comprising all reference objects.

Overall, the results demonstrate that our previous representation of 2D spatial relations have successfully been extended to 3D and multiple reference objects.

B. Validation Experiments on Real Robot

We demonstrate the usefulness of the learned representations of 3D spatial relations in a robot pick-and-place task (Fig. 5) on the humanoid robot ARMAR-6 [33]. We place several known objects on a table, give a language command to the robot and follow the steps described in Section IV to parse the command, ground the relation and object phrases in the robot's scene model and retrieve the learned cylindrical distribution, sample and select a target position and execute the grasping and placing motions.

For the generation of robot motions, we leverage our previous work on Via-Point Movement Primitives (VMP) [38] that can be learned from kinesthetic teaching and adapted to new start and goal positions as well as arbitrary intermediate via-points. Once the target position $\mathbf{p}_{u^*}^{(t_1)}$ has been selected, the robot has to grasp the target object u^* and place it at the new position. Several VMPs (approach, move, place, retreat) are used to approach and grasp the target object, lift it and place it at the target position. Vision-based localization of the involved objects on the table as well as in the robot's hand is used to determine goal positions for the adaptation of the VMPs. The experiments show that the learned cylindrical distributions can generate suitable placing positions.

VII. CONCLUSION AND FUTURE WORK

We presented a generative model for learning 3D spatial relations with multiple reference objects that extends our previous work on 2D spatial relations. The model is based on a parametric cylindrical distribution and enables a robot to manipulate objects in the current scene to fulfill spatial relations specified in a language command by generating suitable placing positions of the manipulated object. To understand and ground the language command, we have leveraged a Named Entity Recognition model and substring matching with the names of objects and relations in the robot's semantic scene model.

Future work and extensions of this work will address the questions of whether the type of a spatial relation (static or dynamic) can be derived automatically based on the respective likelihood of the demonstration data. Further, we will investigate the use of ambiguities in grounding object phrases as entry points for a dialog for disambiguation. In addition, we will extend the spatial relation models to encode more complex manipulation actions.

REFERENCES

[1] S. Tellex, N. Gopalan, H. Kress-Gazit, and C. Matuszek, "Robots That Use Language," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 3, no. 1, pp. 25–55, 2020.

[2] S. Tellex, T. Kollar, S. Dickerson, M. R. Walter, A. G. Banerjee, S. Teller, and N. Roy, "Understanding Natural Language Commands for Robotic Navigation and Mobile Manipulation," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2011.

[3] S. Guadarrama, L. Riano, D. Golland, D. Göhring, Y. Jia, D. Klein, P. Abbeel, and T. Darrell, "Grounding Spatial Relations for Human-Robot Interaction," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2013, pp. 1640–1647.

[4] R. Paul, J. Arkin, N. Roy, and T. M. Howard, "Efficient Grounding of Abstract Spatial Concepts for Natural Language Interaction with Robot Manipulators," in *Robotics: Science and Systems (RSS)*, vol. 12, 2016.

[5] M. Shridhar and D. Hsu, "Interactive Visual Grounding of Referring Expressions for Human-Robot Interaction," in *Robotics: Science & Systems (RSS)*, 2018.

[6] J. Fasola and M. J. Matarić, "Using semantic fields to model dynamic spatial relations in a robot architecture for natural language instruction of service robots," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013, pp. 143–150.

[7] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, "Robot Programming by Demonstration," in *Handbook of Robotics*. Springer, 2008, pp. 1371–1394.

[8] B. D. Argall, S. Chernova, M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robotics and Autonomous Systems*, vol. 57, no. 5, pp. 469–483, 2009.

[9] R. Kartmann, Y. Zhou, D. Liu, F. Paus, and T. Asfour, "Representing Spatial Object Relations as Parametric Polar Distribution for Scene Manipulation Based on Verbal Commands," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 8373–8380.

[10] J. Tan, Z. Ju, and H. Liu, "Grounding Spatial Relations in Natural Language by Fuzzy Representation for Human-Robot Interaction," in *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, July 2014, pp. 1743–1750.

[11] J. Bao, Z. Hong, H. Tang, Y. Cheng, Y. Jia, and N. Xi, "Teach robots understanding new object types and attributes through natural language instructions," in *International Conference on Sensing Technology (ICST)*, vol. 10, Nov. 2016, pp. 1–6.

[12] J. Hatori, Y. Kikuchi, S. Kobayashi, K. Takahashi, Y. Tsuboi, Y. Unno, W. Ko, and J. Tan, "Interactively Picking Real-World Objects with Unconstrained Spoken Language Instructions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 3774–3781.

[13] M. Forbes, R. P. N. Rao, L. Zettlemoyer, and M. Cakmak, "Robot Programming by Demonstration with Situated Spatial Language Understanding," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2015, pp. 2014–2020.

[14] M. Nicolescu, N. Arnold, J. Blankenburg, D. Feil-Seifer, S. B. Banisetty, M. Nicolescu, A. Palmer, and T. Monteverde, "Learning of Complex-Structured Tasks from Verbal Instruction," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, Oct. 2019, p. 8.

[15] R. Paul, J. Arkin, D. Aksaray, N. Roy, and T. M. Howard, "Efficient grounding of abstract spatial concepts for natural language interaction with robot platforms," *International Journal of Robotics Research*, vol. 37, no. 10, pp. 1269–1299, 2018.

[16] M. Shridhar, D. Mittal, and D. Hsu, "INGRESS: Interactive visual grounding of referring expressions," *International Journal of Robotics Research*, vol. 39, no. 2-3, pp. 217–232, 2020.

[17] B. Rosman and S. Ramamoorthy, "Learning spatial relationships between objects," *International Journal of Robotics Research*, vol. 30, no. 11, pp. 1328–1342, 2011.

[18] K. Sjöö and P. Jensfelt, "Learning spatial relations from functional simulation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011, pp. 1513–1519.

[19] S. Fichtl, A. McManus, W. Mustafa, D. Kraft, N. Krüger, and F. Guerin, "Learning spatial relationships from 3D vision using histograms," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2014, pp. 501–508.

[20] F. Yan, D. Wang, and H. He, "Robotic Understanding of Spatial Relationships Using Neural-Logic Learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 8358–8365.

[21] S. U. Lee, S. Hong, A. Hofmann, and B. Williams, "QSRNet: Estimating Qualitative Spatial Representations from RGB-D Images," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2020, pp. 8057–8064.

[22] K. Zampogiannis, Y. Yang, C. Fermuller, and Y. Aloimonos, "Learning the Spatial Semantics of Manipulation Actions Through Preposition Grounding," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2015, pp. 1389–1396.

[23] E. E. Aksoy, Y. Zhou, M. Wächter, and T. Asfour, "Enriched Manipulation Action Semantics for Robot Execution of Time Constrained Tasks," in *IEEE/RAS International Conference on Humanoid Robots (Humanoids)*, Nov. 2016, pp. 109–116.

[24] F. Ziaeetabar, T. Kulvicius, M. Tamosiunaite, and F. Wörgötter, "Prediction of Manipulation Action Classes Using Semantic Spatial Reasoning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 3350–3357.

[25] ——, "Recognition and Prediction of Manipulation Actions Using Enriched Semantic Event Chains," *Robotics and Autonomous Systems (RAS)*, vol. 110, pp. 173–188, Dec. 2018.

[26] T. R. Savarimuthu, A. G. Buch, C. Schlette, N. Wantia, J. Roßmann, D. Martínez, G. Alenyà, C. Torras, A. Ude, B. Nemec, A. Kramberger, F. Wörgötter, E. E. Aksoy, J. Papon, S. Haller, J. Piater, and N. Krüger, "Teaching a Robot the Semantics of Assembly Tasks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 5, pp. 670–692, 2018.

[27] C. R. G. Dreher, M. Wächter, and T. Asfour, "Learning Object-Action Relations from Bimanual Human Demonstration Using Graph Networks," *IEEE Robotics and Automation Letters (RA-L)*, vol. 5, no. 1, pp. 187–194, 2020.

[28] O. Mees, N. Abdo, M. Mazuran, and W. Burgard, "Metric Learning for Generalizing Spatial Relations to New Objects," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 3175–3182.

[29] P. Jund, A. Eitel, N. Abdo, and W. Burgard, "Optimization Beyond the Convolution: Generalizing Spatial Relations with End-to-End Metric Learning," in *IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 4510–4516.

[30] O. Mees, A. Emek, J. Vertens, and W. Burgard, "Learning Object Placements For Relational Instructions by Hallucinating Scene Representations," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[31] P. Azad, T. Asfour, and R. Dillmann, "Combining Harris Interest Points and the SIFT Descriptor for Fast Scale-Invariant Object Recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2009, pp. 4275–4280.

[32] K. Pauwels and D. Kragic, "SimTrack: A Simulation-Based Framework for Scalable Real-Time Object Pose Detection and Tracking," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sept. 2015, pp. 1300–1307.

[33] T. Asfour, M. Wächter, L. Kaul, S. Rader, P. Weiner, S. Ottenhaus, R. Grimm, Y. Zhou, M. Grotz, and F. Paus, "ARMAR-6: A High-Performance Humanoid for Human-Robot Collaboration in Real-World Scenarios," *IEEE Robotics Automation Magazine*, vol. 26, no. 4, pp. 108–121, 2019.

[34] A. Kasper, Z. Xue, and R. Dillmann, "The KIT object models database: An object model database for object recognition, localization and manipulation in service robotics," *The International Journal of Robotics Research*, vol. 31, no. 8, pp. 927–934, 2012. [Online]. Available: https://doi.org/10.1177/0278364912445831

[35] B. Calli, A. Singh, J. Bruce, A. Walsman, K. Konolige, S. Srinivasa, P. Abbeel, and A. M. Dollar, "Yale-CMU-Berkeley dataset for robotic manipulation research," *International Journal of Robotics Research*, vol. 36, no. 3, pp. 261–268, Mar. 2017.

[36] J. O'Keefe, "Vector Grammar, Places, and the Functional Role of the Spatial Prepositions in English," in *Representing Direction in Language and Space*. Oxford University Press, 2003, pp. 69–85.

[37] R. Kartmann, F. Paus, M. Grotz, and T. Asfour, "Extraction of Physically Plausible Support Relations to Predict and Validate Manipulation Action Effects," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 4, pp. 3991–3998, Oct. 2018.

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