Interactive and incremental learning of spatial object relations from human demonstrations

Rainer Kartmann* and Tamim Asfour

High Performance Humanoid Technologies Lab, Institute for Anthropomatics and Robotics, Department of Informatics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany

Humans use semantic concepts such as spatial relations between objects to describe scenes and communicate tasks such as "Put the tea to the right of the cup" or "Move the plate between the fork and the spoon." Just as children, assistive robots must be able to learn the sub-symbolic meaning of such concepts from human demonstrations and instructions. We address the problem of incrementally learning geometric models of spatial relations from few demonstrations collected online during interaction with a human. Such models enable a robot to manipulate objects in order to fulfill desired spatial relations specified by verbal instructions. At the start, we assume the robot has no geometric model of spatial relations. Given a task as above, the robot requests the user to demonstrate the task once in order to create a model from a single demonstration, leveraging cylindrical probability distribution as generative representation of spatial relations. We show how this model can be updated incrementally with each new demonstration without access to past examples in a sample-efficient way using incremental maximum likelihood estimation, and demonstrate the approach on a real humanoid robot.

1 Introduction

While growing up, humans show impressive capabilities to continually learn intuitive models of the physical world as well as concepts which are essential to communicate and interact with others. While an understanding of the physical world can be created through exploration, concepts such as the meaning of words and gestures are learned by observing and imitating others. If necessary, humans give each other explicit explanations and demonstrations to purposefully help the learner improve their understanding of a specific concept. These can be requested by the learner after acknowledging their incomplete understanding, or by the teacher when observing a behavior that does not match their internal model (Grusec, 1994). Assistive robots that naturally interact with humans and support them in their daily lives should be equipped with such continual and interactive learning abilities, allowing them to improve their current models and learn new concepts from their users interactively and incrementally.

One important class of concepts children need to learn are the meanings of spatial prepositions such as *right of*, *above* or *close to*. Such prepositions define geometrical relationships between spatial entities (O'Keefe, 2003), such as objects, living beings or conceptual areas, which are referred to as *spatial relations* (Stopp et al., 1994; Aksoy et al., 2011; Rosman and Ramamoorthy, 2011). Spatial relations play an important role in

FIGURE 1
Incremental learning of spatial relations from human demonstrations.
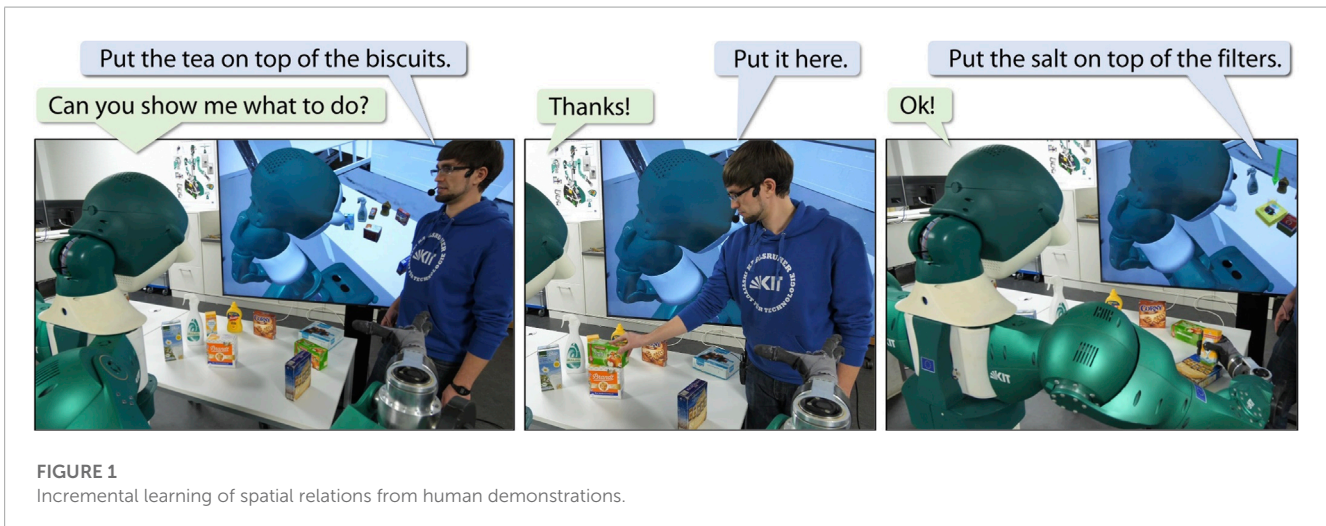
communicating manipulation tasks in natural language, e. g., in "Set the table by placing a plate *on* the table, the fork to *the left* of the plate, and the knife to the *right of* the plate." By abstracting from precise metric coordinates and the involved entities' shapes, spatial relations allow the expression of tasks on a semantic, symbolic level. However, a robot performing such a task must be able to derive subsymbolic placing positions that are needed to parameterize actions. Such *signal-to-symbol gap* remains a grand challenge in cognitive robotics (Krüger et al., 2011). Just like a child, a robot should be able to learn such mapping of spatial object relations from demonstrations provided by humans.

In this work, we consider a robot that has no prior knowledge about the geometric meaning of any spatial relations yet. When given the task to manipulate a scene to fulfill a desired spatial relation between two or more objects, such as a cup *in front of* a bottle, the robot should request a demonstration from the user if it has no model of the spatial relation or if its current model is insufficient (**Figure 1**). Similarly, the robot should be able to receive corrections from the human after executing the task. Finally, having received a new demonstration, the robot should be able to derive a model of the spatial relation from the very first sample and subsequently update its model incrementally with each new demonstration, i. e., without the need to retrain the model with all previously observed demonstrations (Losing et al., 2018).

These goals pose hard requirements for the underlying representation of spatial relations and the cognitive system as a whole. The robot needs to inform the user in case it cannot perform the task by asking for help while maintaining an internal state of the interaction. In addition, the robot will only receive very sparse demonstrations—every single demonstration should be used to update the robot's model of the spatial relation at hand. As a consequence, we require a very sample-efficient representation that can be constructed from few demonstrations and incrementally updated with new ones.

Obtaining a sample-efficient representation can be achieved by introducing bias about the hypothesis space (Mitchell, 1982). Bias reduces the model's capacity, and thereby its potential variance, but more importantly, also reduces the amount of data required to train the model. This effect is also known as the bias-variance tradeoff.

Compared to a partially model-driven approach with a stronger bias, a purely data-driven black-box model can offer superfluous capacity, slowing down training. Bias can be introduced by choosing a model whose structure matches that of the problem at hand. For the problem of placing objects according to desired spatial relations, we proposed to represent spatial relations as parametric probability distributions defined in cylindrical coordinates in our previous work (Kartmann et al., 2020; 2021), observing that capturing relative positions in terms of horizontal distance, horizontal direction, and vertical distance closely matches the notions of common spatial prepositions.

An interesting consideration arises when learning a model from a single demonstration. A single example does not hold any variance; consequently, the learner's only option is to reproduce this demonstration as closely as possible. When receiving more examples, the learner can add variance to its model, thus increasing its ability to adapt to more difficult scenarios. This principle is leveraged by version space algorithms (Mitchell, 1982), which have been used in robotics to incrementally learn task precedence (Pardowitz et al., 2005). While our approach is not a version space algorithm *per se*, it behaves similarly with respect to incremental learning: The model starts with no variance and acquires more variance with new demonstrations, which increases its ability to handle more difficult scenarios such as cluttered scenes.

We summarize our contributions as follows.

1 We present an approach for a robot interacting with a human that allows the robot to (a) request demonstrations from a human for how to manipulate the scene according to desired spatial relations specified by a language instruction if the robot has not sufficient knowledge about how to perform such a task, as well as (b) to use corrections after each execution to continually improve its internal model about spatial relations.

2 We show how our representation of spatial relations based on cylindrical distributions proposed in (Kartmann et al., 2021) can be incrementally learned from few demonstrations based only on the current model and a new demonstration using incremental maximum likelihood estimation.

We evaluate our approach in simulation and real world experiments on the humanoid robot ARMAR-6 (Asfour et al., 2019)[1].

2 Related work

In this section, we discuss related work in the areas of using spatial relations in the context of human-robot interaction and the incremental learning of spatial relation models.

2.1 Spatial relations in human-robot interaction and learning from demonstrations

Spatial relations have been used to enrich language-based human-robot interaction. Many works focus on using spatial relations to resolve referring expressions identifying objects (Tan et al., 2014; Bao et al., 2016) as well as locations (Tellex et al., 2011; Fasola and Matarić, 2013) for manipulation and navigation. In these works, the robot passively tries to parse the given command or description without querying the user for more information if necessary. As a consequence, if the sentence is not understood correctly, the robot is unable to perform the task. Other works use spatial relations in dialog to resolve such ambiguities. Hatori et al. (2018) query for additional expressions if the resolution did not score a single object higher by a margin than all other objects. Shridhar and Hsu (2018); Shridhar et al. (2020) and Doğan et al. (2022) formulate clarification questions describing candidate objects using, among others, spatial relations between them. These works use spatial relations in dialog to identify objects which the robot should interact with. However, our goal is to perform a manipulation task defined by desired spatial relations between objects.

A special form of human-robot interaction arises in the context of robot learning from human demonstrations (Ravichandar et al., 2020). There, the goal is to teach the robot a new skill or task instead of performing a command. Spatial relations have been used in such settings to specify parameters of taught actions. Similar to the works above, given the language command and current context, Forbes et al. (2015) resolve referring expressions to identify objects using language generation to find the most suitable parameters for a set of primitive actions. Prepositions from natural language commands are incorporated as action parameters in a task representation based on part-of-speech tagging by Nicolescu et al. (2019). These works focus on learning the structure of a task including multiple actions and their parameters. However, action parameters are limited to a finite set of values, and spatial relations are implemented as fixed position offsets. In contrast, our goal is learning continuous, geometric models of the spatial relations themselves.

2.2 Learning spatial relation models

Many works have introduced models to classify existing spatial relations between objects to improve scene understanding (Rosman and Ramamoorthy, 2011; Sjöö and Jensfelt, 2011; Fichtl et al., 2014; Yan et al., 2020) and human activity recognition (Zampogiannis et al., 2015; Dreher et al., 2020; Lee et al., 2020). These models are either hand-crafted or are not learned incrementally. In contrast, our models are learned incrementally from demonstrations collected during interaction. Few works consider models for classifying spatial relations which could be incrementally updated with new data. Mees et al. (2020) train a neural network model to predict a pixel-wise probability map of placement positions given a camera image of the scene and an object to be placed according to a spatial relation. However, their models are not trained incrementally, and training neural networks incrementally is, in general, not trivial (Losing et al., 2018). In an earlier work, Mees et al. (2017) propose a metric learning approach to model spatial relations between two objects represented by point clouds. The authors learn a distance metric measuring how different the realized spatial relations in two scenes are. Recognizing the spatial relation in a given scene is then reduced to a search of known examples that are similar to the given scene according to the learned metric. Once the metric is learned and kept fixed, this approach inherently allows adding new samples to the knowledge base, which potentially changes the classification of new, similar scenes. However, their method requires storing all encountered samples to keep a notion of known spatial relations, while our models can be updated incrementally with a limited budget of stored examples (Losing et al., 2018). Mota and Sridharan (2018) follow a related idea to learn classification models of spatial relations incrementally. They encode spatial relations as 1D and 2D histograms over the relative distances or directions (encoded as azimuth and elevation angles), of points in two point clouds representing two objects. These histograms can be incrementally updated by merely adding a new observation to the current frequency bins.

However, all of these models are *discriminative*, i. e., they determine the existing relations between two objects in the current scene. In contrast, our goal is to *generate* a new target scene given desired spatial relations. While discriminative models can still be applied by exhaustively sampling the solution space (e. g., possible locations of manipulated objects), classifying the relations in these candidates and choosing one that contains the desired relation, we believe that it is more effective to directly learn and apply generative geometric models of spatial relations. In our previous works, we introduced generative representations of 3D spatial relations in the form of parametric probability distributions over placing positions (Kartmann et al., 2021). These probabilistic models can be sampled to obtain suitable placing positions for an object to fulfill a desired spatial relation to one or multiple reference objects. We have shown how these models can be learned from human demonstrations which were collected offline. In this work, we show how demonstrations can be given interactively and how the models can be updated in a fully incremental manner, i. e., relying solely on the current model and a new demonstration.

---

1 https://youtu.be/x6XKUZd_SeE.

3 Problem formulation and concept

In the following, we formulate the problem of interactive and incremental learning of spatial relation models from human demonstrations and introduce the general concept of the work. In Section 3.1, we summarize the actual task of semantic scene manipulation which the robot has to solve. In Section 3.2, we describe the semantic memory system as part of the entire cognitive control architecture used on the robot. In Section 3.3, we formulate the problem of incremental learning of spatial relation models. In Section 3.4, we describe the envisioned human-robot interaction task and explain how we approached each subproblem in Section 4.

3.1 Semantic scene manipulation

We consider the following problem: Given a scene with a set of objects and a language command specifying spatial relations between these objects, the robot must transfer the initial scene to a new scene fulfilling the specified relations by executing an action of a set of actions. We denote points in time as $t_0, \ldots, t_k, \ldots$ $(t_k \in \mathbb{R})$ that can be viewed as events where the robot is given a command or makes an observation. The scene model is part of the robot's working memory and contains the current configuration

$$\mathbf{P}^{(t_k)} = \left\{ \mathbf{P}_1^{(t_k)}, \ldots, \mathbf{P}_n^{(t_k)} \right\} \tag{1}$$

of $n^{(t_k)}$ objects at time t_k, where $\mathbf{P}_i^{(t_k)} \in SE(3)$ is the pose of object i with position $\mathbf{p}_i^{(t_k)} \in \mathbb{R}^3$ and orientation $\mathbf{Q}_i^{(t_k)} \in SO(3)$. SE(3) and SO(3) denote the special Euclidean group and special orthogonal group, respectively. The desired relation

$$R^* = \left(s^*, u^*, \mathcal{V}^* \right) \leftarrow \mathrm{ground}(C) \tag{2}$$

is obtained by parsing and grounding the natural language command C, i.e., extracting the phrases referring to objects and relations and mapping them to the respective entities in the robot's working and long-term memories[2]. This desired relation consists of a symbol $s^* \in \mathcal{S}$ describing the identity of the relation, a *target* object $u^* \in \{1, \ldots, n^{(t_k)}\}$ and a set of *reference* objects $\mathcal{V}^* \subseteq \{1, \ldots, n^{(t_k)}\} \setminus \{u^*\}$. $\mathcal{S}$ is the set of known spatial relation symbols. The robot's task is to place the target object u^* at an appropriate pose $\mathbf{P}_{u^*}^{(t_{k+1})}$ which fulfills the desired spatial relation R^*. We aim at keeping the object's original orientation, therefore this task is reduced to finding a suitable position $\mathbf{p}_{u^*}^{(t_{k+1})}$.

Our approach to finding suitable placing positions is based on a generative model G of spatial relations. This model is able to generate suitable target object positions fulfilling a relation $R = (s, u, \mathcal{V})$ based on the current scene $\mathbf{P}^{(t_k)}$ and semantic object information $\mathcal{O}$ (object names and geometry, see (5) below), that is formally

$$\mathbf{p}_u^{(t_{k+1})} \sim G_s \left(u, \mathcal{V}, \mathcal{O}, \mathbf{P}^{(t_k)} \right). \tag{3}$$

The generative models G_s of spatial relations s can be learned from human demonstrations. In our previous work, we recorded

---

human demonstrations for each spatial relation using real objects and learned the generative models G_s offline. Each demonstration consisted of the initial scene $\mathbf{P}^{(t_k)}$, a desired relation $R^* = \left(s^*, u^*, \mathcal{V}^* \right)$ verbalized as a language command for the human demonstrator, and the resulting scene $\mathbf{P}^{(t_{k+1})}$ created by the demonstrator by manipulating the initial scene. Therefore, each demonstration has the form

$$D = \left(\mathbf{P}^{(t_k)}, R, \mathbf{P}^{(t_{k+1})} \right), \quad R = (s, u, \mathcal{V}) \tag{4}$$

and can be used to learn the generative model G_s of the relation s. In contrast to the previous work, in this work we consider the problem of interactively collecting samples by querying demonstrations from the user and incrementally updating the generative models of the spatial relations in the robot's memory with each newly collected sample.

3.2 Robot semantic memory

The robot's semantic memory consists of two parts: the *prior knowledge*, i.e., information defined *a priori* by a developer, and the *long-term memory*, i.e., experience gathered by the robot itself (Asfour et al., 2017). In our scenario, the prior knowledge contains semantic information about N known objects,

$$\mathcal{O} = \{O_i\}_{i=1}^N = \{(g_i, \eta_i)\}_{i=1}^N, \tag{5}$$

including object names η_i and 3D models g_i, as well as names of spatial relations, so that language phrases referring to both objects and relations can be grounded in entities stored in the robot's memory by natural language understanding as indicated in (2). We assume no prior knowledge about the geometric meaning of the relations. Instead, this geometric meaning is learned by the robot during interaction in a continual manner, and is thus part of the long-term memory. In other words, the long-term memory contains generative models $G_s^{(t_k)}$ of spatial relations $s \in \mathcal{S}^{(t_k)}$ representing the robot's current understanding of spatial relations at time t_k,

$$\mathcal{G}^{(t_k)} = \left\{ G_s^{(t_k)} \,\middle|\, s \in \mathcal{S}^{(t_k)} \right\}, \tag{6}$$

where $\mathcal{S}^{(t_k)} \subseteq \mathcal{S}$ is the set of spatial relations for which the robot has learned a generative model at t_k. These models are based on samples collected from human demonstrations $\mathcal{D}$, which are contained in the long-term memory as well:

$$\mathcal{D}^{(t_k)} = \left\{ \mathcal{D}_s^{(t_k)} \,\middle|\, s \in \mathcal{S}^{(t_k)} \right\},$$
$$\mathcal{D}_s^{(t_k)} = \left\{ D_{sj} \right\}_{j=1}^{m_s^{(t_k)}} = \left\{ \mathbf{P}_{sj}^{(t_{k_{sj}})}, R_{sj}, \mathbf{P}_{sj}^{(t_{k_{sj}+1})} \right\}_{j=1}^{m_s^{(t_k)}}, \tag{7}$$

where $\mathcal{D}_s^{(t_k)}$ is the set of $m_s^{(t_k)}$ collected samples of spatial relation s at time t_k, and $t_{k_{sj}}, t_{k_{sj}+1}$ refer to the time a sample was collected. At the beginning, the robot's long-term memory is empty, therefore

$$\mathcal{G}^{(t_0)} = \mathcal{D}^{(t_0)} = \mathcal{S}^{(t_0)} = \emptyset. \tag{8}$$

3.3 Learning spatial relations: batch vs. incremental

During interactions, the robot will collect new samples from human demonstrations. When the robot receives a new demonstration for relation s at time t_k $(k > 0)$, it first stores the new sample D_s in its long-term memory:

$$\mathcal{D}_s^{(t_k)} \leftarrow \{D_s\} \cup \begin{cases} \mathcal{D}_s^{(t_{k-1})}, & s \in \mathcal{S}^{(t_{k-1})} \\ \emptyset, & \text{else} \end{cases} \quad (9)$$

Afterwards, the robot may query its long-term memory for the relevant samples $\mathcal{D}_s^{(t_k)}$ collected to date and use them to update its model G_s of s:

$$G_s^{(t_k)} \leftarrow \text{update}\left(G_s^{(t_{k-1})}, \mathcal{D}_s^{(t_k)} \right) \quad (10)$$

Note that such an incremental model update with each new sample requires a very sample-efficient representation. We refer to (10) as the *batch update problem*. The model is expected to adapt meaningfully to each new sample, but all previous samples are needed for the model update. In the machine learning community, incremental learning can be defined as proposed by Losing et al. (2018):

> "We define an incremental learning algorithm as one that generates on a given stream of training data $s_1, s_2, \ldots, s_t$ a sequence of models $h_1, h_2, \ldots, h_t$. In our case [...] $h_i \colon \mathbb{R}^n \to \{1, \ldots, C\}$ is a model function solely depending on h_{i-1} and the recent p examples $s_i, \ldots, s_{i-p}$, with p being strictly limited."

Comparing this definition to (10), it becomes clear that it is not an instance of incremental learning in this sense, as the number of samples $|\mathcal{D}_s|$ in the memory is not bounded by a constant. This raises the question of whether representations of spatial relations in the form of cylindrical distributions as proposed in our previous work can be learned in a truly incremental way using a limited budget of stored samples. In this work, we further investigate the question of incremental learning of spatial relations without even storing *any* samples, i. e., solely based on the current model $G_s^{(t_{k-1})}$ and the latest sample D_s, forming the *incremental update problem*:

$$G_s^{(t_k)} \leftarrow \text{update}\left(G_s^{(t_{k-1})}, D_s \right) \quad (11)$$

3.4 Interactive learning of spatial relations

We now describe a scenario of a robot interacting with a human where the human gives a semantic manipulation command to the robot while the robot can gather new samples from human demonstrations. The scheme is illustrated in **Figure 2**. The procedure can be described as follows.

1 The human gives a command to the robot at t_k, specifying a spatial relation $R^* = \left(s^*, u^*, \mathcal{V}^* \right)$ as in (2).
2 The robot observes the current scene $\mathbf{P}^{(t_k)}$ and plans the execution of the given task using its current model $G_s^{(t_k)}$. Planning may be successful or fail due to different reasons. Assuming the task given by the user is solvable, the failure can be attributed to an insufficient model.
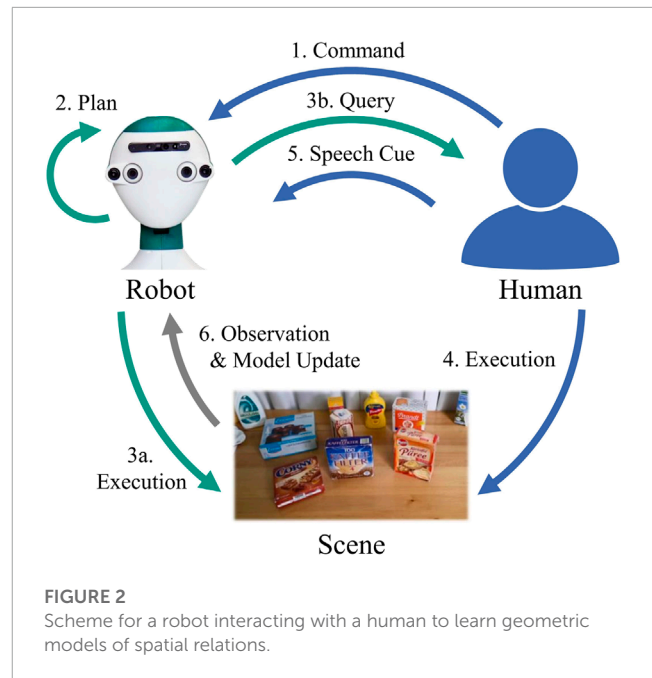
3 Depending on the outcome of 2:

3a. *Planning is successful:* The robot found a suitable placing position and executes the task by manipulating the scene. If the execution was successful and the user is satisfied, the interaction is finished.

3b. *Planning fails:* The robot's current model is insufficient, thus, it queries the human for a demonstration of the task.

4 The human was queried for a demonstration (3b.) or wants to correct the robot's execution (3a.). In both cases, the human performs the task by manipulating the scene.
5 The human signals that the demonstration is complete by giving a speech cue (e. g., "Put it here") to the robot.
6 When receiving the speech cue, the robot observes the changed scene $\mathbf{P}^{(t_{k+1})}$, creates a new sample $D = \left(\mathbf{P}^{(t_k)}, R^*, \mathbf{P}^{(t_{k+1})} \right)$, stores the sample in the long-term memory and updates its model to obtain the new model $G_s^{(t_{k+1})}$ as described in **Section 3.3**.

4 Methods and implementation

To solve the underlying task of semantic scene manipulation, we rely on our previous work which is briefly described in **Section 4.1**. We outline the implementation of the robot's semantic memory in **Section 4.2**. We describe how each new sample is used to update a spatial relation's model in **Section 4.3**. Finally, we explain how we implement the defined interaction scenario in **Section 4.4**.

4.1 3D spatial relations as cylindrical distributions

In (Kartmann et al., 2021), we proposed a model of spatial relations based on a cylindrical distribution $\mathcal{C}$

$$(r, \phi, h) \sim \mathcal{C}(\theta), \quad \theta = (\theta_{rh}, \theta_\phi) \tag{12}$$

over the cylindrical coordinates radius $r \in \mathbb{R}_{\geq 0}$, azimuth $\phi \in [-\pi, \pi]$ and height $h \in \mathbb{R}$. The radius and height follow a joint Gaussian distribution $\mathcal{N}$, while the azimuth, as an angle, follows a von Mises distribution $\mathcal{M}$ (which behaves similar to a Gaussian distribution but is defined on the unit circle),

$$(r, h) \sim \mathcal{N}(\theta_{rh}), \quad \theta_{rh} = (\boldsymbol{\mu}_{rh}, \boldsymbol{\Sigma}_{rh}), \qquad \phi \sim \mathcal{M}(\theta_\phi), \quad \theta_\phi = (\mu_\phi, \kappa_\phi) \tag{13}$$

so the joint probability density function of $\mathcal{C}$ is given by

$$p_\theta(r, \phi, h) = p_{\theta_{rh}}(r, h) \cdot p_{\theta_\phi}(\phi) \tag{14}$$

with $p_{\theta_\phi}(\phi)$ and $p_{\theta_{rh}}(r, h)$ being the respective probability density functions of the distributions in (13). We claim that cylindrical coordinates are a suitable space for representing spatial relations as they inherently encode horizontal distance (*close to*, *far from*), direction (*left of*, *behind*) and vertical distance (*above*, *below*), which are the qualities many spatial relations used by humans are defined by. Therefore, we leverage a cylindrical distribution as a distribution over suitable placing positions of a target object relative to one or more reference objects. The corresponding cylindrical coordinate system is centered at the bottom-projected centroid of the axis-aligned bounding box enclosing all reference objects and scales with the size of that bounding box. By defining the cylindrical coordinate system based on the reference objects' joint bounding box, we can apply the same spatial relation models to single and multiple reference objects. Especially, this allows considering spatial relations that inherently involve multiple reference objects (e. g., *between*, *among*).

4.2 Initialization of robot's semantic memory

We build our implementation of the cognitive architecture of our robot in ArmarX (Vahrenkamp et al., 2015; Asfour et al., 2017). The architecture consists of three-layer for 1) low-level sensorimotor control, 2) high-level for semantic reasoning and task planning and 3) mid-level as a memory system and mediator between the symbolic high-level and subsymbolic low-level. The memory system contains segments for different modalities, such as object instances or text recognized from speech. A segment contains any number of entities that can receive new observations and thus evolve over time. Here, we define three new segments.

- The *sample segment* implements the set of human demonstrations $\mathcal{D}$ in (7). It contains one entity for each $s \in \mathcal{S}^{(t_k)}$, each one representing "all collected samples of relation s." New samples are added as new observations to the respective entity. Thus, $\mathcal{D}_s^{(t_k)}$ in (7) is obtained by querying the sample segment for all observations of entity s.
- The *spatial relation segment* contains the robot's knowledge about spatial relations. There is one entity per $s \in \mathcal{S}$, which holds the information from prior knowledge such as the relations' names for language grounding and verbalization. In addition, each entity contains the current geometric model $G_s^{(t_k)} = \theta$ with θ as in (12).

- The *relational command* segment contains the semantic manipulation tasks $R^* = (s^*, u^*, \mathcal{V}^*)$ extracted from language commands (2). The latest observation is the current or previous task.

In the beginning, the sample segment and the relational command segment are initialized to be empty, i. e., there are no collected samples yet and no command has been given yet. The spatial relation segment is partially initialized from prior knowledge. However, in accordance with the sample segment, the entities contain no geometric model G_s yet, i. e. $G_s^{(t_0)} = \emptyset$.

4.3 Incremental learning of spatial relations

Now, we describe how the batch update problem can be solved using cylindrical distributions. Then, we show how the same mathematical operations can be performed incrementally, i. e., without accessing past samples.

Batch Updates Due to the simplicity of our representation of spatial relations, updating the geometric model of a spatial relation, i. e., its cylindrical distribution, is relatively straightforward. To implement the batch update (10), we query all samples of the relation of interest collected so far, perform Maximum Likelihood Estimation (MLE) to obtain the cylindrical distribution's parameters,

$$G_s^{(t_k)} = \theta_s^{(t_k)} \leftarrow \mathrm{MLE}\left(\mathcal{D}_s^{(t_k)}\right), \tag{15}$$

and update the spatial relation segment.

A cylindrical distribution is a combination of a bivariate Gaussian distribution over (r, h) and a von Mises distribution over ϕ, see (13). Hence, to perform the MLE of a cylindrical distribution in (15), two independent MLEs are performed for the Gaussian distribution $\mathcal{N}(\theta_{rh})$ and the von Mises distribution $\mathcal{M}(\theta_\phi)$, respectively,

$$\theta_{rh}^* \leftarrow \mathrm{MLE}_\mathcal{N}\left(\{r_j, h_j\}_{j=1}^{m_s^{(t_k)}}\right), \qquad \theta_\phi^* \leftarrow \mathrm{MLE}_\mathcal{M}\left(\{\phi_j\}_{j=1}^{m_s^{(t_k)}}\right), \tag{16}$$

where (r_j, ϕ_j, h_j) are the cylindrical of sample $D_j \in \mathcal{D}_s^{(t_k)}$ (see (Kartmann et al., 2021) for more details).

Updating the model with each new sample requires a representation that can be generated from few examples, including just one, which is the case for cylindrical distributions. However, special attention has to be paid to the case of encountering the first sample ($|\mathcal{D}_s| = 1$). As a single sample holds no variance, an estimated distribution would collapse in a single point. Note that the model's expected behavior is well-defined: It should reproduce the single sample when generating placing positions, which often is a valid candidate. Because cylindrical distributions generalize to objects of different sizes, the model can be directly applied to new scenes. The caveat is that the model is not able to provide alternatives if this placing candidate is, e. g., in collision with other objects. If more samples are added ($|\mathcal{D}_s| \geq 2$), the model is generated using standard MLE. The resulting distribution will have a variance according to the samples, which allows the robot to generate different candidates for placing positions.

Technically, in order to allow deriving a generative model from a single demonstration while avoiding special cases in the mathematical formulation, we perform a small data augmentation step: After transforming the sample to its corresponding cylindrical coordinates $\mathbf{c} = (r, \phi, h)^\top$, we create n copies $\mathbf{c}'_1, \ldots, \mathbf{c}'_n$ of the sample to which we add small Gaussian noise

$$\mathbf{c}'_i \leftarrow \mathbf{c} + \left(\varepsilon_r, \varepsilon_\phi, \varepsilon_h\right)^\top, \quad \varepsilon_r, \varepsilon_\phi, \varepsilon_h \sim \mathcal{N}(0, \sigma_\varepsilon) \qquad (17)$$

for $i \in \{1, \ldots, n\}$, and perform MLE on $\{\mathbf{c}, \mathbf{c}'_1, \ldots, \mathbf{c}'_n\}$. This keeps the distribution concentrated on the original sample while allowing small variance parameters to keep the mathematical formulation simple and computations numerically stable. In our experiments, we used $n = 2$ and $\sigma_\varepsilon = 10^{-3}$. Note that the variance created by this augmentation step is usually negligible compared to the variance created by additional demonstrations.

Incremental Updates Now, we present how the model update (15) can be performed in an incremental manner, i. e., with access only to the current model and the new sample. We follow the method by Welford (1962) for the incremental calculation of the parameters of a univariate Gaussian distribution. By applying this method to the multivarate Gaussian and using a similar method for von Mises distributions to implement the MLEs (16), we can incrementally estimate cylindrical distributions.

Welford (1962) proved that the mean μ and standard deviation σ of a one-dimensional series (x_i), $x_i \in \mathbb{R}$, can be incrementally calculated as follows. Let n be the current time step, $\mu_1 = x_1$ and $\sigma = \frac{1}{n}\tilde{\sigma}$.

$$\mu_n = \left(\frac{n-1}{n}\right) \cdot \mu_{n-1} + \frac{1}{n} \cdot x_n \qquad (n > 1), \qquad (18)$$

$$\tilde{\sigma}_n = \tilde{\sigma}_{n-1} + \left(\frac{n-1}{n}\right) \cdot \left(x_n - \mu_{n-1}\right)^2 \qquad (n \geq 1). \qquad (19)$$

This method can be extended to a multivariate Gaussian $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ over a vector-valued series $(\mathbf{x}_i)$ with $\mathbf{x} \in \mathbb{R}^d$. With analogous conditions as above.

$$\boldsymbol{\mu}_n = \left(\frac{n-1}{n}\right) \cdot \boldsymbol{\mu}_{n-1} + \frac{1}{n} \cdot \mathbf{x}_n \qquad (n > 1), \qquad (20)$$

$$\tilde{\boldsymbol{\Sigma}}_n = \tilde{\boldsymbol{\Sigma}}_{n-1} + \left(\frac{n-1}{n}\right) \cdot \left(\mathbf{x}_n - \boldsymbol{\mu}_{n-1}\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_{n-1}\right)^\top \qquad (n \geq 1). \qquad (21)$$

A von Mises distribution $\mathcal{M}(\mu, \kappa)$ over angles ϕ_i is estimated as follows (Kasarapu and Allison, 2015). Let $\mathbf{x}_i$ be the directional vectors in the 2D plane corresponding to the directions ϕ_i, and $\bar{r}$ the normalized length of their sum,

$$\mathbf{x}_i := \begin{pmatrix} \cos(\phi_i) \\ \sin(\phi_i) \end{pmatrix} \in \mathbb{R}^2 \quad (i = 1, \ldots, n), \qquad \bar{r} := \frac{\left\|\sum_{i=1}^n \mathbf{x}_i\right\|}{n}. \qquad (22)$$

The mean angle μ is the angle corresponding to the mean direction $\tilde{\boldsymbol{\mu}} \in \mathbb{R}^2$,

$$\tilde{\boldsymbol{\mu}} = \begin{pmatrix} \tilde{\mu}_x \\ \tilde{\mu}_y \end{pmatrix} = \frac{1}{n \cdot \bar{r}} \sum_{i=1}^n \mathbf{x}_i, \qquad \mu = \tan^{-1}\left(\frac{\tilde{\mu}_y}{\tilde{\mu}_x}\right). \qquad (23)$$

The concentration κ is computed as the solution of

$$A_d(\kappa) = \bar{r}, \qquad \text{where } A_d(\kappa) := \frac{I_{d/2}(\kappa)}{I_{d/2-1}(\kappa)}, \qquad (24)$$

$I_s(\kappa)$ denotes the modified Bessel function of the first kind and order s, and d denotes the dimension of vectors $\mathbf{x} \in \mathbb{R}^d$ on the $(d-1)$-dimensional sphere $\mathbb{S}^{d-1}$ (since we consider the circle embedded in the 2D plane, $d = 2$ in our case). Eq. 24 is usually solved using closed-form approximations (Sra, 2012; Kasarapu and Allison, 2015). However, the important insight here is that (24) does not depend on the values $\mathbf{x}_i$, but only on the normalized length $\bar{r}$ of their *sum*. This means that the MLE for both μ and κ of a von Mises distribution only depends on the sum $\mathbf{r} := \sum_{i=1}^n \mathbf{x}_i$ of the directional vectors $\mathbf{x}_i$, which can easily be computed incrementally. With $\mathbf{r}_1 = \mathbf{x}_1$.

$$\mathbf{r}_n = \mathbf{r}_{n-1} + \mathbf{x}_n \qquad (n > 1), \qquad (25)$$

$$\bar{r} = \frac{\|\mathbf{r}_n\|}{n}, \quad \tilde{\boldsymbol{\mu}} = \frac{1}{n}\mathbf{r}_n \qquad (n \geq 1), \qquad (26)$$

and the remaining terms as in (23) and (24). Overall, this allows to estimate cylindrical distributions fully incrementally. Note that the batch and incremental updates are mathematically equivalent and thus yield the same results.

4.4 Interactive teaching of spatial relations

Finally, we explain how we implement the different steps of the interaction scenario sketched in **Section 3.4** and **Figure 2**. An example with a real robot is shown in **Figure 3**.

1 *Command*: Following (Kartmann et al., 2021), we use a Named Entity Recognition (NER) model to parse object and relation phrases from a language command and ground them to objects and relations in the robot's memory using substring matching. In addition, the resulting task $R^* = \left(s^*, u^*, \mathcal{V}^*\right)$ is stored in the relational command segment of the memory system. This is required to later construct the sample from a demonstration.

2 *Plan*: We query the current model $G_{s^*}^{(t_k)}$ from the spatial relation memory segment. If $G_{s^*}^{(t_k)} = \emptyset$, the geometric meaning of the spatial relation is unknown, and the robot cannot solve the task. In this case, the robot verbally expresses its lack of knowledge and requests a demonstration of what to do from the human. Otherwise, $G_{s^*}^{(t_k)} = \theta$ defines a cylindrical distribution, which is used to sample a given number of candidates (50 in our experiments). As in (Kartmann et al., 2021), the candidates are filtered for feasibility, ranked according to the distribution's probability density function $p_\theta(r, \phi, h)$, and the best candidate is selected for execution. We consider a candidate to be feasible if it is free of collisions, reachable and stable. To decide whether a candidate is free of collisions, the target object is placed virtually at the candidate position and tested for collisions with other objects within a margin of 25 mm at 8 different rotations around the gravity vector in order to cope with imprecision of action execution, e. g., the object rotating inside the robot's hand while grasping or placing (note that our method aims to keep the object's orientation; the different rotations are only used for collision checking). If a feasible candidate is found, planning is successful. However, after filtering, it is possible that none of the sampled candidates is feasible. This is especially likely when only few samples have been collected so far and, thus, the model's variance is low. Before, this event marked a failure, but in this work, the robot is able to recover by expressing its inability to solve
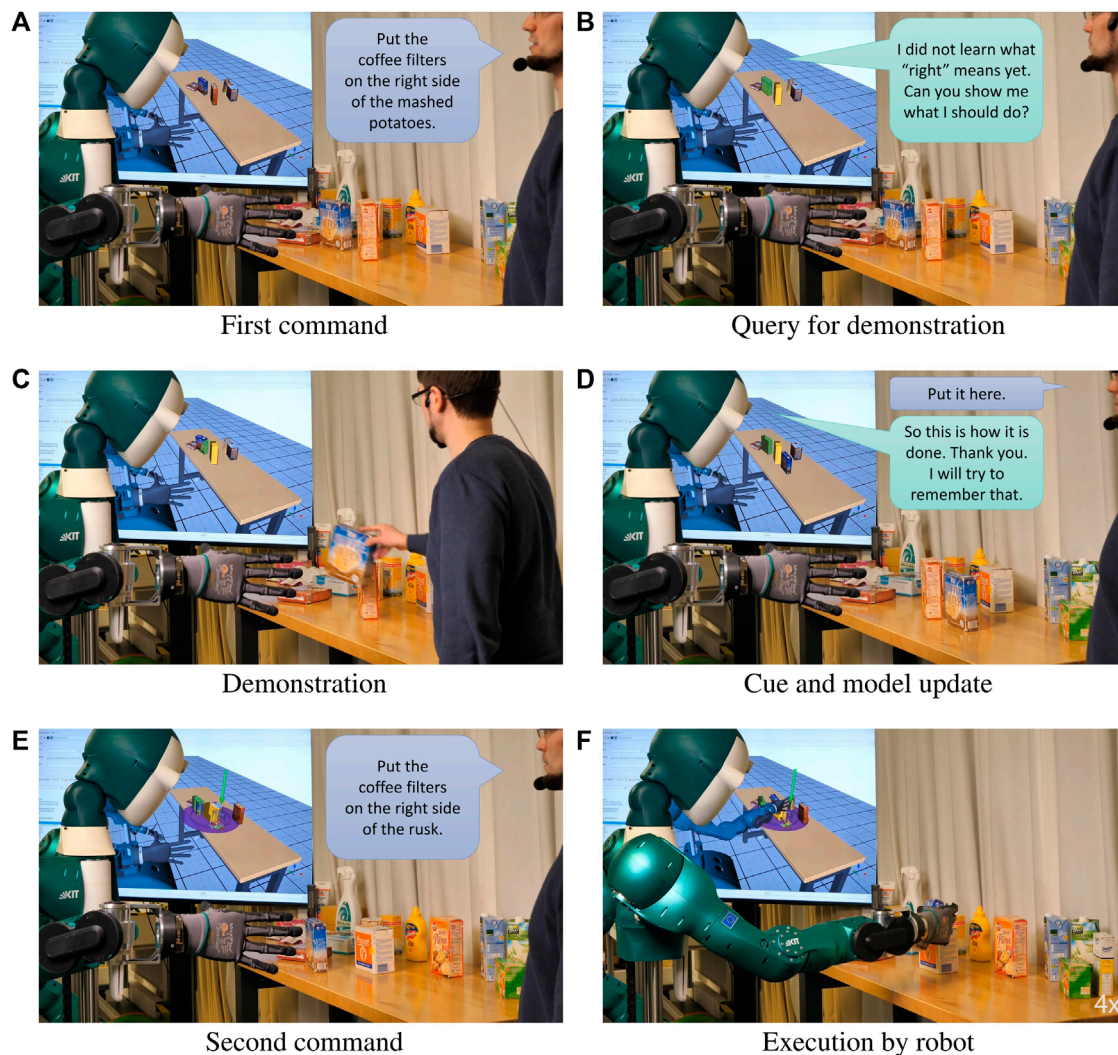
FIGURE 3
Interactively teaching the humanoid robot ARMAR-6 to place objects to the right of other objects. After the first command by the human **(A)**, the robot queries a demonstration **(B)**. The human demonstrates the task **(C)** which is used by the robot to create or update its model **(D)**. When given the next command **(E)**, the robot can successfully perfom the task **(F)**.

the task and ask the human for a demonstration. Note that the query mechanism of both failure cases, i. e. a missing model and an insufficient one, are structurally identical. In both cases, the human is asked to perform the task it originally gave to the robot, i. e., manipulate the scene to fulfill the relation R^*.

3 *Execution by robot and Query:* If planning was successful (3a.), the robot executes the task by grasping the target object and executing a placing action parameterized based on the selected placing position. If planning failed (3b.), the robot verbally requests a demonstration of the task from the human using sentences generated from templates and its speech-to-text system. Examples of generated verbal queries are "I am sorry, I don't know what "right" means yet, can you show me what to do?" (no model) and "Sorry, I cannot do it with my current knowledge. Can you show me what I should do?" (insufficient model). Then, the robot waits for the speech cue signaling the finished demonstration.

4 *Execution by Human and 5. Speech Cue:* After being queried for a demonstration (3b.), the human relocates the target object

to fulfill the requested spatial relation. To signal the completed demonstration, the human gives a speech cue such as "Place it here" to the robot, that is detected using simple keyword spotting and triggers the recording of a new sample. This represents a third case where demonstrations can be triggered: The robot may have successfully executed the task in a qualitative sense (3a.), but the human may not be satisfied with the chosen placing position. In this case, the human can simply change the scene in order to demonstrate what the robot *should have done*, and give a similar speech cue "No, put it here." Again, note that this case is inherently handled by the framework without a special case: When the speech cue is received, the robot has all the knowledge it requires to create a new sample, independently of whether it executed the task before or explicitly asked for a demonstration.

6 *Observation and Model Update:* When the robot receives the speech cue, it assembles a new sample by querying its memory for the relevant information. It first queries its relational command segment for the latest command, which specifies the requested

TABLE 1 Left column: Spatial relation symbols $s \in \mathcal{S}$ used in the experiments. Right columns: Their corresponding baseline model, where $\mathbf{p}_v, \mathbf{p}_{v_i} \in \mathbb{R}^3$ refer to the positions of reference objects, $\mathbf{p}_u$ refers to the initial position of target object u, $\mathbf{x}, \mathbf{y}, \mathbf{z}$ are the unit vectors in the robot's coordinate system, and $\mathrm{dir}(\mathbf{v}) = \mathbf{v}/\|\mathbf{v}\|$ is the direction of a vector $\mathbf{v} \in \mathbb{R}^3$.

Spatial relations	Baseline model	
$(s \in \mathcal{S})$	Description	Placing position $\mathbf{p}_u^{(t_{k+1})}$
right of	20 cm in $+x$ direction	$\mathbf{p}_v + 20\mathrm{cm} \cdot \mathbf{x}$
left of	20 cm in $-x$ direction	$\mathbf{p}_v - 20\mathrm{cm} \cdot \mathbf{x}$
behind	20 cm in $+y$ direction	$\mathbf{p}_v + 20\mathrm{cm} \cdot \mathbf{y}$
in front of	20 cm in $-y$ direction	$\mathbf{p}_v - 20\mathrm{cm} \cdot \mathbf{y}$
on top of	10 cm in $+z$ direction	$\mathbf{p}_v + 10\mathrm{cm} \cdot \mathbf{z}$
close to	20 cm towards initial position of u	$\mathbf{p}_v + 10\mathrm{cm} \cdot \mathrm{dir}(\mathbf{p}_u - \mathbf{p}_v)$
far from	60 cm towards initial position of u	$\mathbf{p}_v + 50\mathrm{cm} \cdot \mathrm{dir}(\mathbf{p}_u - \mathbf{p}_v)$
between	Mid point of reference objects	$1/n \cdot \sum_{i=1}^{n} \mathbf{p}_{v_i}$
among	20 cm from reference objects'	$\mathbf{p}_v + 10\mathrm{cm} \cdot \mathrm{dir}\left(\mathbf{p}_u - 1/n \cdot \sum_{i=1}^{n} \mathbf{p}_{v_i}\right)$
	mid point towards initial position of u	
closer	Half the distance from reference to u	$\mathbf{p}_v + 1/2 \cdot (\mathbf{p}_u - \mathbf{p}_v)$
farther from	Twice the distance from reference to u	$\mathbf{p}_v + 2 \cdot (\mathbf{p}_u - \mathbf{p}_v)$
on the other side of	Distance between reference and target in opposite direction of target	$\mathbf{p}_v - (\mathbf{p}_u - \mathbf{p}_v)$

relation R^* that was just fulfilled by the demonstration. It then queries its object instance segment for the state of the scene $\mathbf{P}^{(t_k)}$ when the command was given and the current state $\mathbf{P}^{(t_{k+1})}$. Combined, this information forms a new sample $D = \left(\mathbf{P}^{(t_k)}, R^*, \mathbf{P}^{(t_{k+1})}\right)$. Afterwards, the robot stores the new sample in its long-term memory and updates G_{s^*} as described in **Section 3.4**. Finally, the robot thanks the human demonstrator for the help, e. g., "Thanks, I think I now know the meaning of 'right' a bit better."

5 Results and discussion

We evaluate our method quantitatively by simulating the interaction scheme with a virtual robot and (Asfour et al., 2019).

5.1 Quantitative evaluation

With our experiments, we aim at investigating two questions: 1) How many demonstrations are necessary to obtain a useful model for a given spatial relation? And 2) how does our model perform compared to baseline models using fixed offsets instead of learned cylindrical distributions? To this end, we implement the proposed human-robot interaction scheme based on human demonstrations collected in simulation.

5.1.1 Experimental setup

For the design of our experiments, a human instructs the robot to manipulate an object according to a spatial relation to other objects. The robot tries to perform the task, and if it is successful, the interaction is finished. If the robot is unable to solve the task,

it requests the user to demonstrate the task and updates its model of the spatial relation from the given demonstration. We call this procedure one *interaction*. We are interested in how the robot's model of a spatial relation develops over the course of multiple interactions, where it only receives a new demonstration if it fails to perform the task at hand.

In our experiments, we consider the 12 spatial relations $\mathcal{S}$ listed in **Table 1**. Consider one relation $s \in \mathcal{S}$, and assume we have 10 demonstrations of this relation. As defined in (4), each demonstration D_i $(i = 1, \ldots, 10)$ has the form $D_i = \left(\mathbf{P}_i^{(t_k)}, R_i, \mathbf{P}_i^{(t_{k+1})}\right)$, with initial object poses $\mathbf{P}_i^{(t_k)}$, desired spatial relation R_i, and object poses after the demonstration $\mathbf{P}_i^{(t_{k+1})}$. Each demonstration D_i also implicitly defines a *task* $T_i = \left(\mathbf{P}_i^{(t_k)}, R_i\right)$ consisting of only the initial scene and the desired relation. Given this setup, we define a *learning scenario* as follows: First, we initialize a virtual robot that has no geometric model of s as it has not received any demonstrations yet. Then, we consecutively perform one interaction for each task $T_1, \ldots, T_{10}$. After each interaction, we evaluate the robot's current model of s on all tasks.

More precisely, in the beginning we ask the robot to perform the first task T_1. As it has no model of s at this point, it will always be unable to solve the task. Following our interaction scheme, the robot requests the corresponding demonstration and is given the target scene $\mathbf{P}_i^{(t_{k+1})}$. The robot learns its first model of s from this demonstration, which concludes the first interaction. We then evaluate the learned model on all tasks $T_1, \ldots, T_{10}$, i. e., for each task we test whether the model can generate a feasible placing position for the target object. Here, we consider T_1 as *seen*, and the remaining tasks $T_2, \ldots, T_{10}$ as *unseen*. Accordingly, we report the success ratios, i. e., the proportion of solved tasks, among the seen task, the unseen tasks and all tasks. We then proceed with the second interaction,

where the robot is asked to perform T_2. This time, the model learned from the previous interaction may already be sufficient to solve the task. If this is the case, the robot does *not* receive the corresponding demonstration and thus keeps its current model. Otherwise, it is given the new demonstration D_2 and can incrementally update its model of s. Again, we test the new model on all tasks, where T_1 and T_2 are now considered as seen and $T_3, \ldots, T_{10}$ as unseen. We continue in this manner for all tasks left. After the last interaction, all tasks $T_1, \ldots, T_{10}$ are seen and there are no unseen tasks. For completeness, we also perform an evaluation on all tasks before the first interaction; here, the robot trivially fails on all tasks since it initially has no model of s.

Overall, for one learning scenario of relation s, we report the proportions of solved tasks among all tasks, the seen tasks and the unseen tasks after each interaction (and before the first interaction). Note that, as explained above, the number of seen and unseen tasks change over the course of a learning scenario. In addition, we report the number of demonstrations the robot has been given after each interaction. Note that this number may be smaller than the number of performed interactions, as a demonstration is only given during an interaction if the robot fails to perform the task at hand. As the results depend on the order of the tasks and demonstrations, we run 10 repetitions of each learning scenario with the tasks randomly shuffled for each repetition, and report all metrics' means and standard deviations over the repetitions. Finally, to obtain an overall result over all spatial relations, we aggregate the results of all repetitions of learning scenarios of all relations $s \in \mathcal{S}$, and report the resulting means and standard deviations of success ratios and number of demonstrations with respect to the number of performed interactions.

We compare our method with baseline models for all relations which place the target object at a fixed offset from the reference objects. **Table 1** gives their descriptions and the definitions of the placing position $\mathbf{p}_u^{(t_{k+1})}$ they generate. The baseline models of direction-based static relations (*right of, left of, behind, in front of, on top of*) yield a candidate at a fixed distance towards the respective direction. Distance-based static relations (*close to, far from, among*) place the target object at a fixed distance towards its initial direction. The baseline models of dynamic spatial relations (*closer, farther from, on the other side*) are similar, but the distance of the placing position to the reference object is defined relative to the current distance between the objects. We conduct the experiments on the baseline models in the same way as described above, except that (1) the robot has access to the models from the beginning, and (2) we only report the success ratio among all tasks over one repetition, as the baseline models are constant and do not change during a learning scenario or depending on the order of tasks.

For collecting the required demonstrations for each relation, we use a setup with a human, several objects on a table in a simulated environment and a command generation based on sentence templates. Given the objects present in the initial scene and the set of spatial relation symbols $\mathcal{S}$, we generate a verbal command specifying a desired spatial relation $R = (s, u, \mathcal{V})$ such as "Place the cup between the milk and the plate." The human is given the generated command and performs the task by moving the objects in the scene. We record the initial scene $\mathbf{P}^{(t_k)}$, the desired spatial relation R, and the final scene $\mathbf{P}^{(t_{k+1})}$, describing a full demonstration $D = \left(\mathbf{P}^{(t_k)}, R, \mathbf{P}^{(t_{k+1})} \right)$ as required for the experiment. Aside from the

objects involved in the spatial relation, the scenes also contain inactive objects which can lead to collisions when placing the object at a generated position, thus rendering some placements infeasible.

5.1.2 Results

Figure 4 shows the means and standard deviations of the percentage of solved tasks among all tasks, the seen tasks and the unseen tasks after each interaction aggregated over all relations and repetitions as described above. Furthermore, it shows the averages of total number of demonstrations the robot has received after each interaction. As explained above, note that not all interactions result in a demonstration. Also, note that the number of seen and unseen tasks change with the number of interactions; especially, there are no seen tasks before the first interaction and there are no unseen tasks after the last one. For comparison, the success ratio of the baseline models averaged over all relations are shown as well.

In addition, **Figure 5** gives examples of success ratios and number of received demonstrations aggregated only over the repetitions of learning scenarios of single spatial relations. Note that here, the baseline's performance is constant over all repetitions as the number of tasks it can solve is independent of the order of tasks in the learning scenario. Finally, **Figure 6** presents concrete examples of solved and failed tasks involving the spatial relations in **Figure 5** demonstrating the behavior of our method and the baseline during the experiment. Note that the cases represented by the rows in **Figure 6** are not equally frequent; our method more often outperforms the baseline models than *vice versa* as indicated by the success ratios in **Figures 4**, **5**.

5.1.3 Discussion

The baseline models achieved only 52.5% ± 24.9% success rate over all relations, showing that many tasks in our demonstrations were not trivially solvable using fixed offsets. Also, the standard deviation of 52.5% ± 24.9%. The models were successful if their candidate placing position was free and on the tables. However, if the single candidate was, e. g., obstructed by other objects, the models could not fall back to other options. This was especially frequent among relations such as *close to*, *between* and *among* which tend to bring the target object close to other objects. Other collisions were caused because the sizes of the involved objects were not taken into account, especially among relations with fixed distances.

As expected, using our method the robot could never solve a task before the first interaction. However, after the first interaction (which always results in a demonstration), the robot could already solve about 83.3% ± 13.1% of seen and about 51.9% ± 18.3% of unseen tasks on average, almost equaling the baseline models on the unseen tasks. After just two interactions, the mean success ratios on seen tasks reaches a plateau of at 89%–92%, with 1.53 ± 0.19 demonstrations on average. Importantly, the success ratios among the seen tasks stays high and does not decrease after more interactions, which indicates that our method generally does not "forget" what it has learned from past demonstrations. The success ratios among unseen tasks rise consistently with the number of interactions, although more slowly than among seen tasks. Nonetheless, after five interactions, the robot could solve 85.3 ± 9.9% of unseen tasks after having received 2.13 ± 0.41 demonstrations on average, which shows that our method can generalize from few
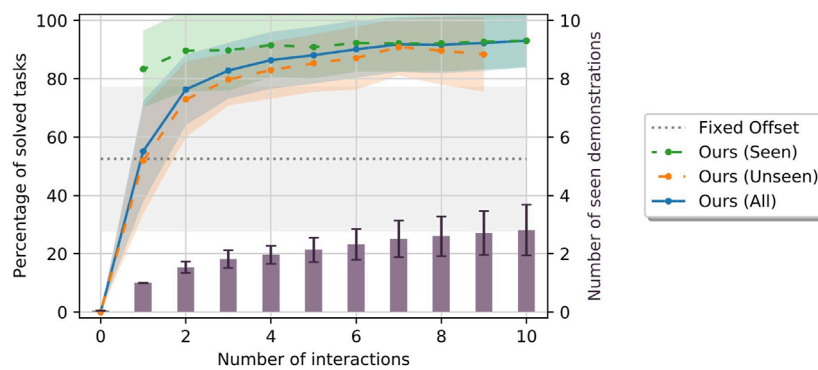
FIGURE 4
Means and standard deviations of percentage of tasks solved by our method among the seen (green), unseen (orange) and all (blue) tasks, by the baseline models using fixed offsets (gray) as well as number of demonstrations (purple) the robot has been given after a given number of interactions. All metrics are aggregated over multiple repetitions and all spatial relations.
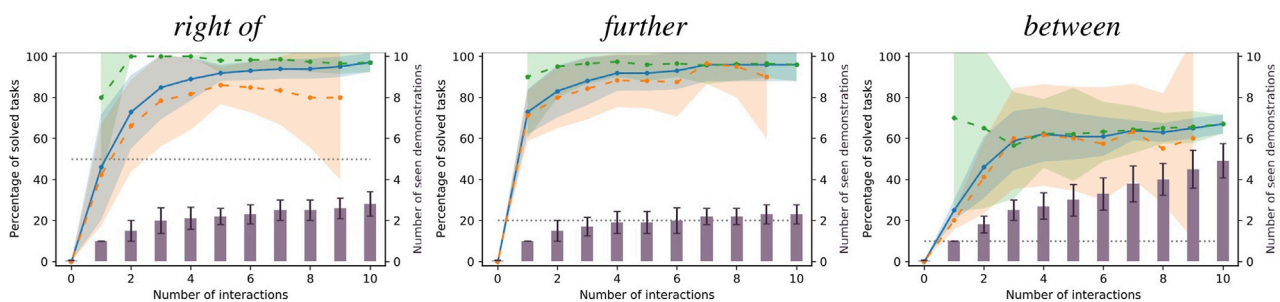

FIGURE 5
Examples of success ratios and number of demonstrations aggregated over the repetitions of single relations. Colors are as in **Figure 4**. Note that for single relations, the success ratios of the baseline models are constant and, thus, have no variance.
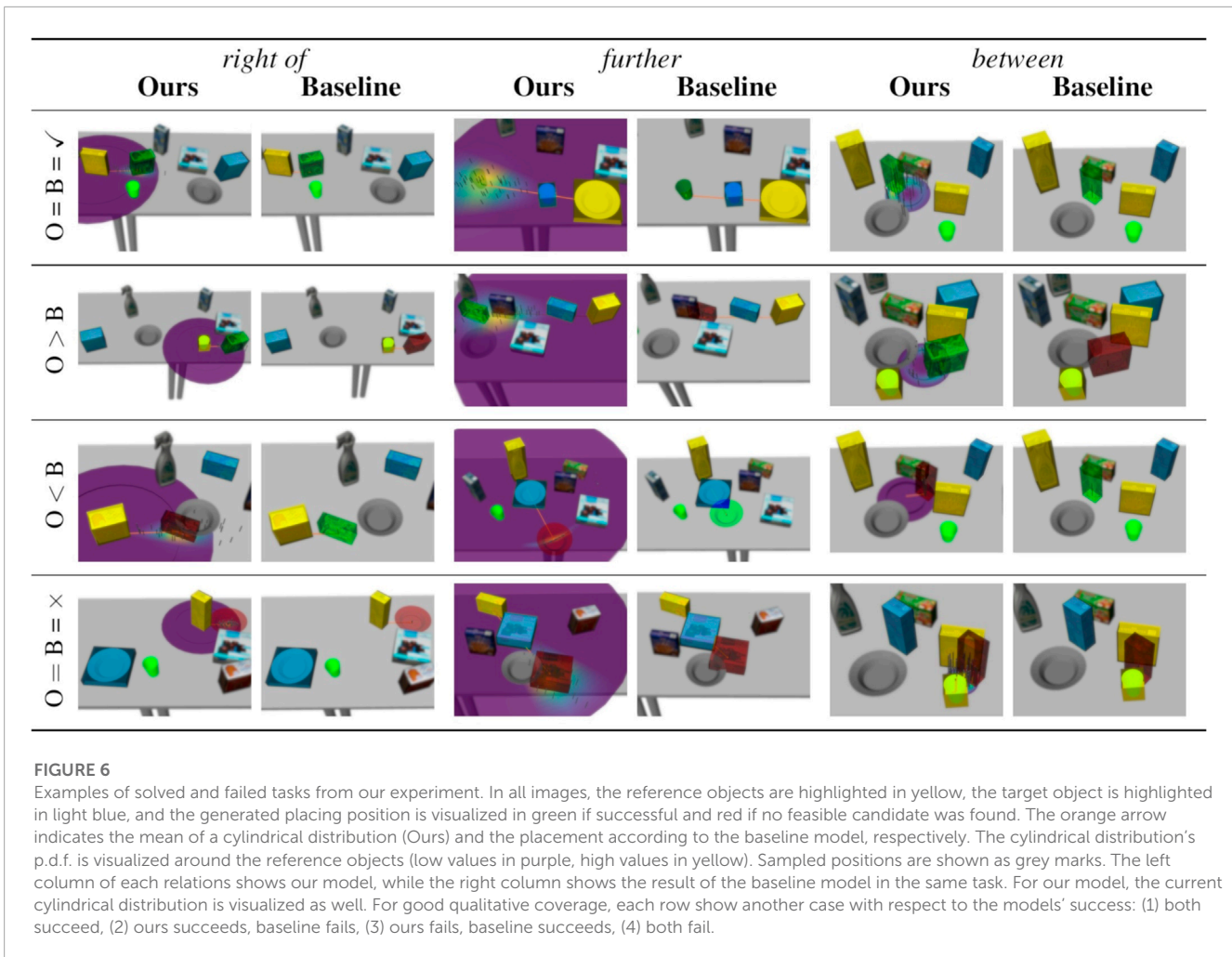
demonstrations. After completing each learning scenario, i. e., after all interactions have been performed, the robot could solve 93.0 ± 9.1% of all tasks while having received 2.81 ± 0.87 demonstrations on average.

One might wonder why the robot is not always able to successfully reproduce the first demonstration on the single seen task after the first interaction. This can be explained in two ways: First, the human is free to change the target object's orientation during the demonstration, which can allow the human to place it closer to other objects than without rotating it. However, as our method only generates new positions while trying to keep the original object orientation, the robot cannot reproduce this demonstration as it would lead to a collision. Second, we use a rather conservative strategy for collision detection in order to anticipate inaccuracies in action execution (e. g., the object rotating in the robot's hand during grasping or placing). More precisely, we use a collision margin of 25 mm to check for collisions at different hypothetical rotations of the target object (see **Section 4.4**). Therefore, if the human placed the object very closely to another in the demonstration, the robot might discard this solution to avoid collisions. These are situations that can only be solved after increasing the model's variance through multiple demonstrations.

We can observe the behavior of our method in more detail by focusing on the results of single relations shown in **Figure 5** and the

examples in **Figure 6**. First, the standard deviation over the success ratios among the unseen tasks tends to increase towards the end of the learning scenarios. This is likely due to the decreasing number of unseen tasks towards the end: Before the final interaction, there is only one unseen task left, so the success ratio is either 0% or 100% in each repetition, leading to a higher variance than when the success ratio is averaged over more tasks. As for the relation *right of*, common failure cases were caused by the conservative collision checking in combination with a finite number of sampled candidates (**Figure 6**, third row), or the mean distance of the learned distribution being too small for larger objects such as the plate (**Figure 6**, fourth row). With the relation *further*, failures were often caused by the candidate positions being partly off the table in combination with the distance variance being too small to generate alternatives (**Figure 6**, third row), or scenes where the sampled area was either blocked by other objects or off the table (**Figure 6**, fourth row).

The relation *between* was one of the relations that were more difficult to learn, with a success ratio of only 67.0 ± 4.6% among all tasks after receiving an average of 4.90 ± 0.83 demonstrations at the end of the learning scenario (the baseline model achieved only 10%). In the demonstrations, the area between the two reference objects was often cluttered, which prevented our method from finding a collision free placing location. The success ratio among

FIGURE 6
Examples of solved and failed tasks from our experiment. In all images, the reference objects are highlighted in yellow, the target object is highlighted in light blue, and the generated placing position is visualized in green if successful and red if no feasible candidate was found. The orange arrow indicates the mean of a cylindrical distribution (Ours) and the placement according to the baseline model, respectively. The cylindrical distribution's p.d.f. is visualized around the reference objects (low values in purple, high values in yellow). Sampled positions are shown as grey marks. The left column of each relations shows our model, while the right column shows the result of the baseline model in the same task. For our model, the current cylindrical distribution is visualized as well. For good qualitative coverage, each row show another case with respect to the models' success: (1) both succeed, (2) ours succeeds, baseline fails, (3) ours fails, baseline succeeds, (4) both fail.

the seen tasks starts at $70.0 \pm 45.8\%$ after one interaction. The large variance indicates that, compared to other relations, there were many demonstrations that could not be reproduced after the first interaction, with the success ratio among seen tasks being either 0% or 100% leading to a high variance, similar to the success ratios among unseen tasks towards the end of the learning scenarios. Moreover, the success ratio among the seen tasks decreases to $56.7 \pm 26.0\%$ after the third interaction, although it slightly increases again afterwards. In this case, the 1.50 ± 0.50 additional demonstrations caused the model to "unlearn" to solve the first tasks in some cases. However, after the third interaction, the success ratios among seen and unseen tasks stabilize and do not change significantly with more demonstrations. Apparently, the models reached their maximum variance after a few interactions, with new demonstrations not changing the model significantly; however our conservative collision detection often caused all candidates to be considered infeasible. One especially difficult task is shown in **Figure 6** (fourth row), where the two reference objects were standing very close to each other, leaving little space for the target object. Finally, note that the tasks for the *between* relation shown in the first and the third row of **Figure 6** are the same. This is because the baseline could only solve this single task. The corresponding failure example of our model (third row) was shows a model learned from only one demonstration. Indeed,

with two or more demonstrations, our method was always able to solve the this task (example in first row).

To summarize, while some aspects can still be improved, the overall results demonstrate that our generative models of spatial relations can be effectively learned in an incremental manner from few demonstrations, while our interaction scheme allows the robot to obtain new demonstrations to improve its models if they prove insufficient for a task at hand.

5.2 Validation experiments on real robot

We performed validation experiments on the real humanoid robot (Asfour et al., 2019) which are shown in the video. An example is shown in **Figure 3** and described in more detail here: In the beginning, the robot has no geometric model of the *right of* relation. First, the user commands the robot to place an object *on the right side of* a second object (step 1. in **Section 3.4**). The robot grounds the relation phrase "on the right side of" to the respective entry in its memory 2.), and responds that it has "not learned what *right* means yet," and requests the user to show it what to do (3a.). Consequently, the user gives a demonstration by performing the requested task 4.) and gives the speech cue "Put it here" 5.). The robot observes

the change in the scene and creates a model of the spatial relation *right of* 6.). Afterwards, the user starts a second interaction by instructing the robot to put the object *on the right side of* a third one 1.). This time, the robot has geometric model of *right of* in its memory 2.) and is able to perform the task 3b.). Beyond that, we show examples of demonstrating and manipulation the scene according to the relations *in front of*, *on top of*, *on the opposite side of*, and *between*.

6 Conclusion and future work

In this work, we described how a learning humanoid robot which has the task to manipulate the scene based on desired spatial object relations can query and use demonstrations from a human during interaction to incrementally learn generative models of spatial relations. We demonstrated how the robot can communicate its inability to solve a task in order to collect more demonstrations in a continual manner. In addition, we showed how a parametric representation of object spatial relations can be learned incrementally from few demonstrations. In future work, we would like to make the human-robot interaction even more natural by detecting when a demonstration is finished, thus releasing the requirement of a speech cue indicating this event. Furthermore, we want to explore how knowledge about different spatial relations can be transferred between them and leveraged for learning new ones.

Data availability statement

The raw data supporting the conclusion of this article will be made available by the authors, without undue reservation.

Author contributions

RK developed the methods and their implementation and performed the evaluation experiments. The entire work was conceptualized and supervised by TA. The initial draft of the manuscript was written by RK and revised jointly by RK and TA. All authors listed have made a substantial, direct, and intellectual contribution to the work and approved it for publication.

Funding

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frobt.2023.1151303/full#supplementary-material

References

Aksoy, E. E., Abramov, A., Dörr, J., Ning, K., Dellen, B., and Wörgötter, F. (2011). Learning the semantics of object–action relations by observation. *Int. J. Robotics Res.* 30, 1229–1249. doi:10.1177/0278364911410459

Asfour, T., Dillmann, R., Vahrenkamp, N., Do, M., Wächter, M., Mandery, C., et al. (2017). "The karlsruhe ARMAR humanoid robot family," in *Humanoid robotics: A reference*. Editors A. Goswami, and P. Vadakkepat (Springer Netherlands), 1–32.

Asfour, T., Wächter, M., Kaul, L., Rader, S., Weiner, P., Ottenhaus, S., et al. (2019). ARMAR-6: A high-performance humanoid for human-robot collaboration in real world scenarios. *Robotics Automation Mag.* 26, 108–121. doi:10.1109/mra.2019.2941246

Bao, J., Hong, Z., Tang, H., Cheng, Y., Jia, Y., and Xi, N. (2016). "CXCR7 suppression modulates microglial chemotaxis to ameliorate experimentally-induced autoimmune encephalomyelitis," *International conference on sensing technology* (Nanjing, China, 10, 1–7. doi:10.1016/j.bbrc.2015.11.059

Doğan, F. I., Torre, I., and Leite, I. (2022). "Asking follow-up clarifications to resolve ambiguities in human-robot conversation," in Proceeding of the ACM/IEEE International Conference on Human-Robot Interaction, Sapporo, Hokkaido, Japan, March 2022 (IEEE), 461–469.

Dreher, C. R. G., Wächter, M., and Asfour, T. (2020). Learning object-action relations from bimanual human demonstration using graph networks. *IEEE Robotics Automation Lett. (RA-L)* 5, 187–194. doi:10.1109/lra.2019.2949221

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

Fichtl, S., McManus, A., Mustafa, W., Kraft, D., Krüger, N., and Guerin, F. (2014). "Learning spatial relationships from 3D vision using histograms," in Proceeding of the IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, June 2014 (IEEE), 501–508.

Forbes, M., Rao, R. P. N., Zettlemoyer, L., and Cakmak, M. (2015). "Robot programming by demonstration with situated spatial language understanding," in Proceeding of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, May 2015 (IEEE), 2014–2020.

Grusec, J. E. (1994). *Social learning theory and developmental psychology: The legacies of robert R. Sears and albert bandura. A century of developmental psychology*. Washington, DC, US: American Psychological Association.

Hatori, J., Kikuchi, Y., Kobayashi, S., Takahashi, K., Tsuboi, Y., Unno, Y., et al. (2018). "Interactively picking real-world objects with unconstrained spoken language instructions," in Proceeding of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, May 2018 (IEEE), 3774–3781.

Kartmann, R., Zhou, Y., Liu, D., Paus, F., and Asfour, T. (2020). "Representing spatial object relations as parametric polar distribution for scene manipulation based on verbal commands," in Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, January 2021 (IEEE), 8373–8380.

Kartmann, R., Liu, D., and Asfour, T. (2021). "Semantic scene manipulation based on 3D spatial object relations and language instructions," in Proceeding of the IEEE-RAS International Conference on Humanoid Robots (Humanoids), Munich, Germany (IEEE), 306–313.

Kasarapu, P., and Allison, L. (2015). Minimum message length estimation of mixtures of multivariate Gaussian and von Mises-Fisher distributions. Mach. Learn. 100, 333–378. doi:10.1007/s10994-015-5493-0

Krüger, N., Geib, C., Piater, J., Petrick, R., Steedman, M., Wörgötter, F., et al. (2011). Object–action complexes: Grounded abstractions of sensory–motor processes. Robotics Aut. Syst. 59, 740–757. doi:10.1016/j.robot.2011.05.009

Lee, S. U., Hong, S., Hofmann, A., and Williams, B. (2020). "QSRNet: Estimating qualitative spatial representations from RGB-D images," in Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, January 2021 (IEEE), 8057–8064.

Losing, V., Hammer, B., and Wersing, H. (2018). Incremental on-line learning: A review and comparison of state of the art algorithms. Neurocomputing 275, 1261–1274. doi:10.1016/j.neucom.2017.06.084

Mees, O., Abdo, N., Mazuran, M., and Burgard, W. (2017). "Metric learning for generalizing spatial relations to new objects," in Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September 2017 (IEEE), 3175–3182.

Mees, O., Emek, A., Vertens, J., and Burgard, W. (2020). "Learning object placements for relational instructions by hallucinating scene representations," in Proceeding of the IEEE International Conference on Robotics and Automation (ICRA), Paris, France, August 2020 (IEEE), 94–100.

Mitchell, T. M. (1982). Generalization as search. Artif. Intell. 18, 203–226. doi:10.1016/0004-3702(82)90040-6

Mota, T., and Sridharan, M. (2018). "Incrementally grounding expressions for spatial relations between objects," in Proceeding of the International Joint Conferences on Artificial Intelligence (IJCAI) (IEEE), 1928–1934.

Nicolescu, M., Arnold, N., Blankenburg, J., Feil-Seifer, D., Banisetty, S. B., Nicolescu, M., et al. (2019). "Learning of complex-structured tasks from verbal instruction," in Proceeding of the IEEE/RAS International Conference on Humanoid Robots (Humanoids) (IEEE), 770–777.

O'Keefe, J. (2003). "Vector grammar, places, and the functional role of the spatial prepositions in English," in Representing direction in language and space (Oxford University Press), 69–85.

Pardowitz, M., Zollner, R., and Dillmann, R. (2005). "Learning sequential constraints of tasks from user demonstrations," in Proceeding of the IEEE-RAS International Conference on Humanoid Robots, Tsukuba, Japan, December 2005 (IEEE), 424–429.

Ravichandar, H., Polydoros, A. S., Chernova, S., and Billard, A. (2020). Recent advances in robot learning from demonstration. Annu. Rev. Control, Robotics, Aut. Syst. 3, 297–330. doi:10.1146/annurev-control-100819-063206

Rosman, B., and Ramamoorthy, S. (2011). Learning spatial relationships between objects. Int. J. Robotics Res. 30, 1328–1342. doi:10.1177/0278364911408155

Shridhar, M., and Hsu, D. (2018). "Interactive visual grounding of referring expressions for human-robot interaction," in Robotics: Science & systems (RSS).

Shridhar, M., Mittal, D., and Hsu, D. (2020). Ingress: Interactive visual grounding of referring expressions. Int. J. Robotics Res. 39, 217–232. doi:10.1177/0278364919897133

Sjöö, K., and Jensfelt, P. (2011). "Learning spatial relations from functional simulation," in Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), San Francisco, CA, USA, September 2011 (IEEE), 1513–1519.

Sra, S. (2012). A short note on parameter approximation for von Mises-Fisher distributions: And a fast implementation of Is(x). Comput. Stat. 27, 177–190. doi:10.1007/s00180-011-0232-x

Stopp, E., Gapp, K.-P., Herzog, G., Laengle, T., and Lueth, T. C. (1994). "Utilizing spatial relations for natural language access to an autonomous mobile robot," in KI-94: Advances in artificial intelligence. Lecture notes in computer science. Editors B. Nebel, and L. Dreschler-Fischer (Berlin, Heidelberg: Springer), 39–50.

Tan, J., Ju, Z., and Liu, H. (2014). "Grounding spatial relations in natural language by fuzzy representation for human-robot interaction," in Proceeding of the IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Beijing, China, July 2014 (IEEE), 1743–1750.

Tellex, S., Kollar, T., Dickerson, S., Walter, M. R., Banerjee, A. G., Teller, S., et al. (2011). Understanding natural language commands for robotic navigation and mobile manipulation. Proc. AAAI Conf. Artif. Intell. 25, 1507–1514. doi:10.1609/aaai.v25i1.7979

Vahrenkamp, N., Wächter, M., Kröhnert, M., Welke, K., and Asfour, T. (2015). The robot software framework ArmarX. it - Inf. Technol. 57, 99–111. doi:10.1515/itit-2014-1066

Welford, B. P. (1962). Note on a method for calculating corrected sums of squares and products. Technometrics a J. statistics Phys. Chem. Eng. Sci. 4, 419–420. doi:10.1080/00401706.1962.10490022

Yan, F., Wang, D., and He, H. (2020). "Robotic understanding of spatial relationships using neural-logic learning," in Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, January 2021 (IEEE), 8358–8365.

Zampogiannis, K., Yang, Y., Fermuller, C., and Aloimonos, Y. (2015). "Learning the spatial semantics of manipulation actions through preposition grounding," in Proceeding of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, May 2015 (IEEE), 1389–1396.