Engineering Applications of Artificial Intelligence 163 (2026) 112882

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai

Research paper

Symbolic representation of objects relative poses for robotic manipulation tasks

Isacco Zappa *, Sara Vignali, Andrea Maria Zanchettin, Paolo Rocco

Politecnico di Milano - Department of Electronics, Information and Bioengineering, Via Ponzio 34/5, Milan, 20133, Italy

ARTICLE INFO

Keywords:
Programming by demonstration
Symbolic artificial intelligence
Symbolic knowledge representation
Robot skill programming
Collaborative robotics

ABSTRACT

Collaborative robots (cobots) are democratizing industrial automation with their user-friendly programming approaches. Nevertheless, the Blockly-like interfaces typically available on cobots still require the user to define the program logic flow. Recent advancements in robotics research provide the robotic system with the reasoning capabilities given by symbolic artificial intelligence. This way, the cobot can acquire a new skill from a user demonstration, understand its semantics, and use symbolic planning for grounding and sequencing. Such methodologies rely on a symbolic description of the scene that should adequately represent how the cobot's actions modify the environment. The symbols employed in the literature, however, either lack descriptive accuracy or are too specific for the targeted task, resulting in the application of the proposed teaching methodologies only to simple scenarios. This paper addresses these issues by introducing a methodology for symbolically describing general-purpose spatial relations between entities in a workspace, enhancing the flexibility and the range of application of cobots symbolic reasoning for complex manipulation tasks. The proposed approach involves defining a tunable set of predicates for relative positions and orientations, enabling precise symbolic representations, necessary for real-world tasks. The adoption of these symbols into a Programming by Demonstration framework empowers non-expert users to teach skills and deploy cobots in complex industrial tasks without coding. Experimental results demonstrate the effectiveness of this method, showing that first-time users can deploy cobots for a complex machine tending task comprising parts reorientations.

1. Introduction

Collaborative Robots (cobots) provide industries with safe and flexible solutions to partly automatize production processes. Among the main factors contributing to the widespread adoption of cobots are their novel, more user-friendly programming approaches, which simplify deployment and reduce reliance on skilled personnel (Villani et al., 2018). Indeed, users without extensive programming expertise can quickly install cobots by combining their Graphical User Interface (GUI), physical demonstration of the required movements, and eventually organizing actions in a Blockly-like programming fashion (Weintrop et al., 2017). Furthermore, manufacturers implemented libraries of pre-programmed sequences of parameterized movements that enable the cobot to carry out specific operations, from simply picking an object to more complex ones such as screwing (Pedersen et al., 2016).

Modern cobot user interfaces greatly simplify programming compared to those of traditional industrial robots (George et al., 2023). However, they have inherent limitations. Despite the improved ease of use offered by the new approaches, users must still define the logic of the robot program execution flow. This constraint restricts the range of tasks to those involving relatively simple and fixed movements, like loading boxes on a pallet in a predefined pattern. Traditional robot programming languages may still be necessary for more complex tasks in unstructured environments that may also require decision-making and adaptive behaviors to accomplish the goal. For instance, this can be the case when a robot is deployed to prepare assembly kits with components that can vary depending on the user's request.

This issue has been recently tackled in the literature by providing the robotic system with symbolic task-planning capabilities. These approaches enable the teaching of new skills by a single user demonstration, leveraging the definition of their execution policy with the learning of symbolic models that provide the robot with decision-making capabilities (Steinmetz et al., 2019; Diehl et al., 2021; Liang et al., 2022; Zanchettin, 2023; Eiband et al., 2023; Diehl et al., 2024). Indeed, the flexibility and modularity of the skills are enhanced by inferring from the demonstration the preconditions and effects of their execution. The semantic descriptors are then formalized following the

* Corresponding author.
E-mail address: isacco.zappa@polimi.it (I. Zappa).

https://doi.org/10.1016/j.engappai.2025.112882
Received 8 January 2025; Received in revised form 7 July 2025; Accepted 17 October 2025
Available online 23 October 2025

syntax of a planning language, such as the Planning Domain Description Language (PDDL) (Ghallab et al., 1998), to enable a symbolic task planner to autonomously ground and sequence the skills required to fulfill a given task.

The inference of preconditions and effects relies on the correct description of how the state of the workspace changes during the demonstration. Such a description is acquired by a symbolic representation of the properties and relations of the entities available in the scene (work-objects, robots, grippers, etc.) through Boolean predicates. These predicates must be defined beforehand and should be capable of representing all possible states of the workspace. State-of-the-art teaching techniques provide experimental evidence of their applicability only to simple tasks, such as stacking cubes, peg in hole, and handover (Steinmetz et al., 2019; Diehl et al., 2021; Liang et al., 2022; Zanchettin, 2023). This limitation is mainly due to the predicates used to describe the spatial relations between the objects in the scene. Indeed, the employed symbols only describe ***qualitative*** relations such as *on top* or *near to*. While such a qualitative representation may be sufficient for simple scenarios such as stacking cubes, it lacks the quantitative description required for industrial tasks. As a consequence, it is challenging to use symbolic approaches for real industrial applications.

The definition of a set of symbols that can quickly be tuned to define ***quantitatively*** with the required accuracy the spatial relations between the entities in the robot workspace can pave the way to the application of symbolic task planning for realistic industrial tasks. Indeed, such symbols can be employed to describe the goal of a task, such as the pattern in which workpieces should be fixed on a welding table, or the initial condition, like randomly oriented products coming from a conveyor belt to be placed on pre-formed plastic packages. Therefore, the user is left only with the tuning of the number (resolution) of symbols to describe the relations between the entities in the scene and teach the required skills. Then, the system copes with the definition of the program logic flow thanks to symbolic planning.

This paper introduces an innovative approach to symbolically describe quantitative spatial relations between entities in the scene. The method defines a set of general-purpose predicates representing relative positions and orientations with a tunable resolution. To generate such a set, a proper partition of the relative position and orientation space is required, where each partition element represents a distinct symbol. While a partition of the 3D space can be easily performed by a 3D grid, generating symbols for relative orientations requires a more complex pipeline, whose details will be discussed in Section 4. We showcase how the introduced sets of symbols can be tuned to describe all the relevant spatial relations among the entities being part of a machine tending task. Then, the teaching by demonstration framework presented in Zanchettin (2023) enables ten candidates to teach the required skills. Results demonstrate how the users, on their first experience with robot programming, progressively understand the concept of robot skill and can effectively deploy the cobot for such a complex manipulation task.

The main contributions of this research are as follows:

- The introduction of a set of quantitative relative position symbols that can be quickly tuned and deployed to represent symbolically the relevant spatial relations of the entities in the workspace.
- The definition of a strategy to represent relative orientations symbolically. The symbols grant the semantic understanding of scenarios where the robot, through its actions, can change objects' orientations.
- The application, thanks to our proposed sets of symbols, of a state-of-the-art Programming by Demonstration (PbD) framework to a complex machine tending task involving the semantic description of reorientation operations.

The remainder of this paper is structured as follows. Section 2 reviews the literature on spatial reasoning and semantic analysis of manipulations. Section 3 provides insights into the foundation of our methodology: symbolic planning and semantic PbD. Section 4 presents our approach for generating the set of symbols describing relative positions and orientations. Section 5 outlines the application of the proposed symbols on a mockup of a machine tending task involving workpiece reorientations. Section 6 presents and discusses the results of the experimental campaign. Section 7 concludes and outlines the future development of this work.

2. Related works

2.1. Spatial knowledge representation

The field of spatial knowledge representation, i.e. the representation of relations between entities and the reasoning about these entities and relations, is known as spatial reasoning. Spatial reasoning has been applied to many different fields such as cognitive science (Freksa, 2015), geographical information science (Freksa et al., 2018), engineering design (Park et al., 2007), process plant fault detection (Fathi et al., 1993), and understanding language cues for human–robot interaction (Krishnamurthy and Kollar, 2013). The methods used to represent and understand the spatial relationships between entities are built on manually encoded rules or learning algorithms. According to the definition given by Bredeweg and Struss (2003), it is possible to distinguish between the qualitative and quantitative, also called metric, types of spatial thinking. While quantitative spatial thinking relies on continuous measures of relative poses in terms of angles and distances, qualitative thinking seeks to capture human-level ideas of space by modeling spatial relations using a finite number of symbols. The two categories are compared in Thippur et al. (2015).

A method to represent quantitative spatial relationships between extended entities is proposed in Berretti and Del Bimbo (2006), overcoming the limits given by point-like representations. Sjöö and Jensfelt (2011) proposes a method to categorize functional spatial relations from object features in a simulated environment. In a later work (Sjöö et al., 2012), the same authors manually encode the grounding for the qualitative representation of the prepositions *in* and *on* for visual object search. Rosman and Ramamoorthy (2011) investigates how to extract the qualitative semantic meaning of *on* and *adjacent* starting from data coming from the point clouds of the objects, then elaborated in contact point networks where the spatial relations are interpreted. Mota and Sridharan (2018) employs a declarative language equipped with prepositional relations between objects to enable incremental learning and upgrade the metric grounding of spatial relations. The system deduces relative position and distance-based prepositions from the 3D point cloud of the scene. Lee et al. (2020) employs a neural network to generate masks from a continuous stream of 3D point clouds from an RGB-D scene video. Then, three qualitative key metrics perform connection, trajectory, and distance computation over the masks, from which the spatial relations between the objects are obtained by online filtering through a Dynamic Bayesian Network. The approaches considered in these works provide reasonable solutions for symbolically representing how objects are positioned in the scene concerning one another. Nevertheless, when targeting the acquisition of the semantic meaning of industrial robot skills, a more refined description of the relative positions of the entities in the workspace may be required depending on the relation of interest. Moreover, the studies do not consider the problem of representing the relative orientations of objects symbolically.

Some studies argue that handcrafting of spatial symbols' grounding is tedious and rigid (Kulick et al., 2013; Fichtl et al., 2014; Mees et al., 2017, 2020). Therefore, learning approaches are targeted to let the system acquire knowledge of spatial relations from experience. A method for active learning of spatial symbols' grounding is proposed in Kulick et al. (2013), where a Gaussian Process classifier is employed to generate the symbols and make them more reliant through interaction with the teacher. Fichtl et al. (2014) extracts the spatial relations

such as *on top* or *inside* from 3D vision exploiting a classifier trained on histograms compiled on the 3D patches of the objects. Mees et al. (2017) applies distance metric learning to generalize spatial symbols to novel objects, starting from the point cloud of the scene. The method is proved capable of recognizing relative position relations, such as *on top* and *inside*, but also discerning whether an object is inclined with respect to another. A neural network working on an RGB image of the scene is trained in Mees et al. (2020) to propose a placement position for an object given a spatial relation. A scene graph parsing method using relative depth and layout encoding was introduced by Jiang et al. (2021) to capture geometric relationships, including spatial orientations such as *looking at*, *laying on*, and *against*. Zhu et al. (2022) employs a neural network to generate and reason over higher-order spatial symbols such as ternary projective relations (e.g., *between*), thus enabling more robust inference over noisy data. In Li et al. (2024), multiple convolutional transformer blocks are used to capture the semantic relations between the points of a point cloud reconstructed by a sonar sensor. These methods are not suitable for the problem of describing spatial relations symbolically for complex industrial tasks. Indeed, they do not provide the spatial resolution required to describe assemblies or fine placement actions. Moreover, relative orientations are frequently neglected, and existing symbolic descriptors are unsuitable for industrial applications due to their inherent ambiguity and lack of precise geometric grounding.

Significant work to enhance human–robot interaction by making the robot understand spatial relation cues in natural language is available in the literature. Skubic et al. (2004) investigates how to represent spatial information in a symbolic way to enable the interpretation of sub-directions, like "Object is mostly to the left but somewhat forward", common in natural language. Tellex et al. (2011) proposes a framework using spatial representations to enable understanding of natural language commands for navigation and object manipulation through an autonomous forklift system. Bounding boxes of the objects in a scene are employed in Guadarrama et al. (2013) to evaluate spatial features and prepositions to answer queries and execute actions. Tan et al. (2014) addresses the issue of finding the object being referred to in a language command by defining the image regions around the reference object as fuzzy memberships. Gemignani et al. (2015) focuses on achieving qualitative spatial reasoning capabilities for navigation and grounding of vocal commands by computing a representation of the environment in a 2D grid. Shridhar et al. (2020) targets locating objects referred to in a natural language cue by reviewing all the spatial relations identified from a snapshot of the scene. Kartmann et al. (2021) proposes a method for binding spatial relationships describing relative position between two or more objects to goal positions for executing a manipulation, following the instructions given verbally by the operator. To facilitate human–robot collaboration in unstructured environments, Howard et al. (2022) proposes a centralized semantic representation of the scene informed by visual and linguistic inputs. This framework enables the robot to reason about relative object locations with respect to its own frame or other entities, using grounded symbols and inverse semantics for disambiguation. A language-conditioned transformer with spatial self-attention was introduced by Chen et al. (2022) to ground 3D objects by modeling spatial relations, including relative distances and orientations such as *facing*, to disambiguate natural language references. Since these works focus on understanding language cues by the operator, they only define symbols representing approximate spatial relations. Indeed, in the type of verbal cues considered, it is not possible to describe spatial relations with the level of accuracy and information on relative orientations required for industrial tasks.

A growing trend in the research community focuses on enabling Vision-Language Models (VLMs) and Large Multimodal Models (LMMs) to reason about spatial relations from 2D and 3D visual inputs. While not explicitly designed for symbol grounding, recent work has shown that grounded symbolic representations can emerge from such models (Wu et al., 2023; Yang et al., 2024). One recent effort addresses the limitations of VLMs in 3D spatial reasoning by introducing an Internet-scale dataset of 3D spatial visual-question answering examples, enabling models to better capture both qualitative and quantitative spatial relationships (Chen et al., 2024). The work by Cheng et al. (2024) enhances the spatial reasoning capabilities of VLMs by integrating depth information into the visual encoder and learning regional representations from 3D scene graphs, demonstrating strong performance in perceiving relative distances and generalizing across diverse environments. Despite the remarkable progress of VLMs and LMMs in visual and linguistic tasks, several recent works have demonstrated that these models still face significant challenges in spatial reasoning. Benchmarks such as Visual Spatial Reasoning (VSR) (Liu et al., 2023) and the 3D Spatial Reasoning Benchmark (3DSRBench) (Ma et al., 2024) have systematically evaluated a wide range of state-of-the-art models, revealing persistent limitations in understanding spatial relations, particularly relative positions and object orientations. 3DSRBench focuses on 3D spatial questions across diverse viewpoints, highlighting models' poor generalization to changes in camera perspective. VSR tests fine-grained spatial language understanding and shows that models struggle to recognize relational terms, especially those involving orientation. These results underscore the gap between current multimodal models and the spatial reasoning abilities required for real-world applications such as robotics and scene understanding.

Table 1 summarizes the comparison with spatial symbols available in the literature. It is worth noting that the entries corresponding to LMM and VLM-based methods and benchmarks (e.g., (Chen et al., 2024; Cheng et al., 2024; Liu et al., 2023; Ma et al., 2024)) are derived from the types of spatial relations explicitly addressed or demonstrated in the respective papers.

2.2. Semantic analysis of manipulations

A typical application of spatial symbols is to support semantic analysis of manipulations. A significant contribution to the field has been provided by Aksoy et al. (2011), with an early work introducing a set of spatial relations between nodes in a scene. Later, different techniques such as Object-Action Complexes (Wächter et al., 2013) and Semantic Event Chains (Aksoy et al., 2015; Aein et al., 2019) have been applied for extracting the semantics of manipulation actions, building the frameworks on the concept of spatial relations. Zampogiannis et al. (2015) models spatial predicates for object pairs in 3D space from a continuous RGB-D stream of the scene. The study focuses on spatial prepositions common in the natural language, such as *right* and *left*, by partitioning the space surrounding the considered object with hexahedra. The same authors of Lee et al. (2020) in a previous work (Lee et al., 2019) apply a similar approach for human activity recognition. In Dreher et al. (2019), a system is devised for identifying bimanual actions from an RGB-D video capturing a human demonstration by exploiting spatial symbols. The relations between objects have also been exploited by Wang et al. (2024) to gather contextual scene information with the aim of improving the understanding of long-term manipulation actions from videos. When targeting the recognition of human-demonstrated actions, it is sufficient that spatial symbols describe the interaction between the entities and how these interactions change over time. Therefore, a more precise description of the spatial relations is not required. Moreover, even these approaches do not consider the problem of representing relative orientations symbolically.

Our work aims at filling the gap in the literature regarding a more refined and quickly tunable symbolic representation of relative poses for robotic manipulation tasks. The symbols introduced in this paper can enhance the state-of-the-art semantic environment representation techniques required for describing the semantics of complex robotic manipulation skills.

Table 1
Comparison with spatial symbols proposed in the literature.

Paper	Relative position symbols	Relative orientation symbols
Berretti and Del Bimbo (2006)	*before*, *coincident*, *after* on each reference axis (27 in total)	–
Sjöö et al. (2012)	*in*, *on*	–
Rosman and Ramamoorthy (2011)	*on*, *adjacent*	–
Mota and Sridharan (2018)	*in*, *above*, *below*, *front*, *behind*, *right*, *left* + *touching*, *not-touching*, *far*	–
Lee et al. (2020)	*disconnected*, *partially occluded*, *identical*, *proper part of*, *including* + *very far*, *far*, *commensurate*, *close*, *very close*	–
Kulick et al. (2013)	*on*, *close*, *hold*, *inside*	*upright*
Fichtl et al. (2014)	*on top*, *inside*	–
Mees et al. (2017)	*on top*, *inside*, *next to*	*inclined*
Mees et al. (2020)	*inside*, *left*, *right*, *in front*, *behind*, *on top*	–
Jiang et al. (2021)	*at*, *under*, *behind*, *near*, *in front of*, *over*, *above*, *in*, *attached to*, *between*, *across*, *on back of*, *on*	*looking at*, *laying on*, *against*
Zhu et al. (2022)	*left*, *right*, *before*, *after*, *between*	–
Skubic et al. (2004)	*right*, *mostly in front but somewhat to the left*, *right-left*, *mostly to the left but somewhat forward*, *left*, ..., *back*, ..., *right* (16 in total)	–
Tellex et al. (2011)	*on*, *near*, *to the left of*, *to the right of*, *in front of*, *behind of*	–
Guadarrama et al. (2013)	*in front*, *behind*, *close*, *to the left*, *to the right*, *on*, *far from*, *inside*, *near*	–
Tan et al. (2014)	*left of*, *right of*, *behind*, *in front of*, *top left*, *top right*, *bottom left*, *bottom right*, *next to*	–
Gemignani et al. (2015)	*behind*, *in front*, *on the right*, *on the left* + *next to*, *near*, *nearest*, *far*, *not next to*, *furthest*	–
Shridhar et al. (2020)	*on the left on*, *on the right of*, *on the top of*, *at the bottom*, *in the middle*, *bottom-left/right*, *top-left/right*	–
Kartmann et al. (2021)	*left*, *right*, *front*, *behind*, *close to*, *inside*, *on top*, *above*, *under*, *closer*, *farther*, *other side*, *between*, *among*	–
Howard et al. (2022)	*leftmost*, *left*, *behind*, *before*, *right*, *rightmost*	–
Chen et al. (2022)	*closest*, *on the left*, *on the right*, *farther*	*facing*
Chen et al. (2024)	*behind*, *on the left*, *on the right*, *in front of*	–
Cheng et al. (2024)	*in front*, *behind*, *on the right*, *on the left*, *below*, *above*	–
Liu et al. (2023)	*adjacent to*, *alongside*, *at the right side of*, *at the left side of*, *attached to*, *at the back of*, *ahead of*,*below*, *at the edge of*, *on top of*, *beneath*, ...(66 in total)	*facing*, *facing away from*, *parallel to*, *perpendicular to*
Ma et al. (2024)	*next to*, *far*, *above*, *in front*, *beyond*, *left*, *right*	*facing*
Our Approach	**Tunable**[a]	**Tunable**[a]

[a] Tunable depending on the required resolution. Based on the chosen amount of symbols and range, our methodology can reproduce all the symbols above.

3. Background

3.1. Symbolic planning

Symbolic planning aims at computing the sequence of actions required to achieve a desired outcome. The state of the environment, together with the task and the actions, are described using symbols called predicates. Predicates are Boolean functions that represent properties and relations of the entities in the scene. For instance, the predicate *loaded(workpiece, spindle)* can be defined so that it holds true when the workpiece is loaded in the spindle of the machine tool. A state of the system can then be described by the subset of predicates that hold true at that moment. After defining the initial and goal state, actions can be seen as the agent's operations that make the system evolve from one state to another, ideally closer to the goal. Actions are described symbolically by the predicates representing the conditions that must be satisfied for the action to be executed (preconditions) and the changes in the state of the environment after its execution (effects). Therefore, such action models provide robotic systems with enhanced reasoning capabilities and task flexibility.

PDDL offers a standardized language for defining symbolic planning problems and domains. A PDDL symbolic planner works with a collection of predicates used to describe the scene and the available actions, grouped in the so-called *PDDL Domain*. The *PDDL Problem* instead collects the information on the entities available in the scene and the initial and goal state described by predicates. A comprehensive set of ready-to-use symbolic planners working with PDDL is available, making the implementation of a symbolic planning pipeline into a robotic system straightforward.

3.2. Skill programming by demonstration

Kinesthetic teaching enables a human operator to program a robot through hand-guided demonstrations. This approach allows users without robot programming expertise to easily program new robot skills. A robot skill is a pre-programmed execution policy that can be seen as a building block that the robot can use to solve a more complex task. However, while reproducing the recorded trajectory can be sufficient for tasks where the objects are all placed in a known position, the movements the operator has programmed can hardly be autonomously adapted by the robot to use it in different settings. Zanchettin (2023) devised a methodology to enrich the action model with semantics details from a one-shot kinesthetic demonstration. The approach infers the skill semantics based on how the symbolic state of the system changes during the demonstration. Moreover, the system stores parameterized

movements that enable the robot to execute the action in different settings.

When the user provides the input to start recording a skill, the system takes a snapshot of its semantic state. Then, the user proceeds to demonstrate the trajectories and end-effector actions (e.g., open/close the gripper) required to perform the skill while saving waypoints stored by the system as end-effector poses in the operational space. Once the recording is stopped, the approach compares the current symbolic state of the environment with the initial one, assessing which predicates are activated, deactivated, or hold true along the demonstration. Finally, the system computes the preconditions and effects based on the three sets and formalizes the skill model in PDDL. The skill can then be stored in a library and used by a symbolic planner as a building block for computing the sequence required to solve a more complex task.

To ensure the execution in different settings, the robot movement cannot be performed following the waypoints stored in the teaching phase directly in the robot reference frame. Indeed, considering the skill to pick a workpiece, if the position of the object in the scene changes during execution time, the robot would fail to pick it. Therefore, to enable the transferability of the skill to different settings, a strategy must be defined to assess which is the relevant reference frame for the waypoints. A decision tree drives this choice by using the semantic model of the taught skill. The tree considers the sets of predicates, either activated or deactivated, and outputs the most suitable entity whose reference frame should be used to describe the waypoints. More insights about the definition of the decision tree can be found in Zanchettin (2023).

4. Symbols for relative pose description

Concerning the application of a semantic-informed PbD framework for complex industrial tasks, a symbolic representation that is able to discern how the robot's actions modify the scene is essential. Without such descriptive capabilities, the robot could not assess the preconditions and effects of the taught skills and recognize different initial and final conditions for the planning task. However, defining a set of symbols to characterize the scene for complex manipulation tasks is not trivial. Indeed, as outlined in Section 2, symbols available in the literature are too simple and do not provide the accuracy typically required in industrial applications. Therefore, the user is left with the burden of defining the symbols and their grounding depending on the task, and this represents a clear challenge requiring extensive programming expertise. Since semantic-informed PbD is devised for non-experts, the lack of general-purpose, easily tunable symbols goes against its intended simplicity and ease of use.

Complex industrial manipulation tasks involve moving objects to accomplish a specified goal from varying initial conditions, like kitting or loading a box with parts from a conveyor belt. For such tasks, it is relevant to describe symbolically how the parts are initially located with respect to a relevant entity (i.e. *is_box(part, box)* or *is_on_left_side(part, conveyor)*) or how they should be arranged to reach the goal condition (i.e. *is_inside_kit(part, kit)* or *is_placed_in_box(part, box)*). Indeed, manipulation actions involve modifying how the entities are located with respect to each other in the scene.

The idea behind creating a general set of symbols for manipulation tasks is to avoid linking symbols to specific types of relations or objects. Instead, the symbols should describe the spatial relationship by directly addressing the relative pose between the object's reference frames. Such a set of symbols can be defined by an adequate partition of the space representing the relative position and orientation. Each partition element is treated as a symbol, activated when the relevant measure, either relative position or orientation between the two objects, falls inside the partition element boundaries.

The partition must have two properties. Indexability guarantees low computational effort for detecting the activation of the symbols. Moreover, the partition should be uniformly distributed to avoid the undesired scenario where a subset of symbols represents most of the relative position and orientation, limiting the system's ability to distinguish different situations. For the relative position, such a partition can be obtained directly in the Cartesian space with a 3D grid with constant intervals. Adjusting the grid interval size can accommodate a different number of symbols and representation accuracies. The same 3D grid cannot be adopted for the non-Cartesian space of relative orientation since it would not reach the aforementioned desired properties for the partition. Details on how the partitions for the relative positions and orientations are devised, together with an assessment of the suitability of the proposed approach, are presented in the following.

4.1. Relative position symbols

The measure of the relative position between two objects is a vector in the Cartesian space connecting the origins of the two reference frames. In a robotic manipulation scenario, it is not always requested to represent the relative position between two entities in the scene with a symbol. For instance, in a kitting scenario, it is essential to describe how a component is positioned in the kit only when it is placed on top of it and not when it is still in the storage area. Therefore, it is sufficient to generate a symbol only when the distance between two objects falls under a certain threshold ρ that can be tuned according to the relation of interest. Therefore, only the portion of space included in the axis-aligned cube centred in the reference object and identified by the two opposite vertices ($-\rho$,$-\rho$,$-\rho$) (ρ, ρ, ρ) needs to be partitioned. By dividing each direction into finite and constant bins, a cubic 3D grid is created. Finally, the index of each partition element, which can also be seen as the symbol representing the relative position between the two objects, is obtained by unfolding the 3D matrix into a vector.

It is worth noting that the typical description of the relative position found in the literature, such as "on the right" or "on top", can be seen as a particular case of the partition just defined, with just three bins in each direction. The partition and how the symbols meet their activation condition can be seen in Fig. 1.

4.2. Relative orientation symbols

The axis-angle notation describes relative orientations by a unit vector $\bar{r}$, $\|\bar{r}\| = 1$, pointing at the direction of rotation and an angle θ, representing how much the frame of the reference object should be rotated along the unit vector to align with the second object's frame. While $\bar{r}$ can point in any direction, θ is limited to the range $\theta \in [0, \pi]$. By multiplying $\bar{r}$ by θ, we obtain a vector in Cartesian coordinates $\bar{r}\theta$ lying in a sphere with π radius. The space contained in the sphere represents every possible relative orientation. A direct partition of the sphere, in Cartesian or spherical coordinates, entails the aforementioned issues related to the non-uniform distribution of the symbols' activation. To better understand the problem, consider the distance between two points near the sphere's centre divided by a fixed polar angle $\hat{\theta}$. This distance is minimal compared to the distance between two points on the sphere's surface, divided by the same polar angle $\hat{\theta}$. Partitioning the sphere in the same fashion as the 3D grid for the relative position would, therefore, induce the generation of a set of symbols that could fail in discriminating different orientations.

The solution to the problem comes from the intuition that it is possible to trace the spherical space back to a cubic space. Once a cube is obtained, a simple partition along the three axes can be performed. There exist several sphere-to-cube transformations (Lambers, 2020). Radial stretching consists in stretching the sphere points along their radial direction as a function of their position in the sphere, thus obtaining a cube with a side length equal to 2π. The formula for the transformation of the three Cartesian coordinates is:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \frac{\bar{r}\theta^2}{\|\bar{r}\theta\|_\infty} \tag{1}$$

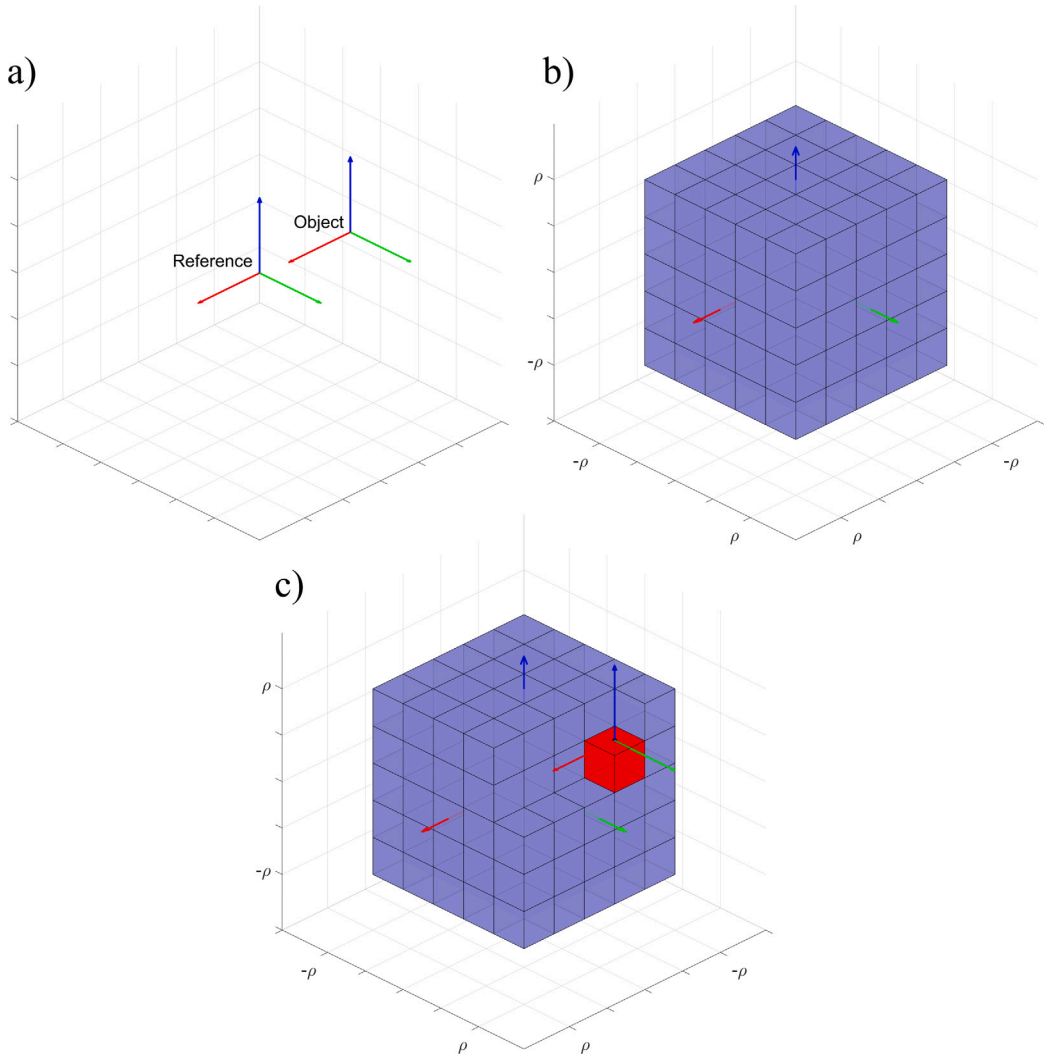

Fig. 1. (a) Position of the reference frame and the object. (b) Axis-aligned partition of the 3D space [$-\rho$, ρ] centered on the reference frame (5 symbols per direction). (c) Activation of a symbol (in red) when the measure falls inside the symbol's boundaries (the surrounding symbols are removed for clarity).

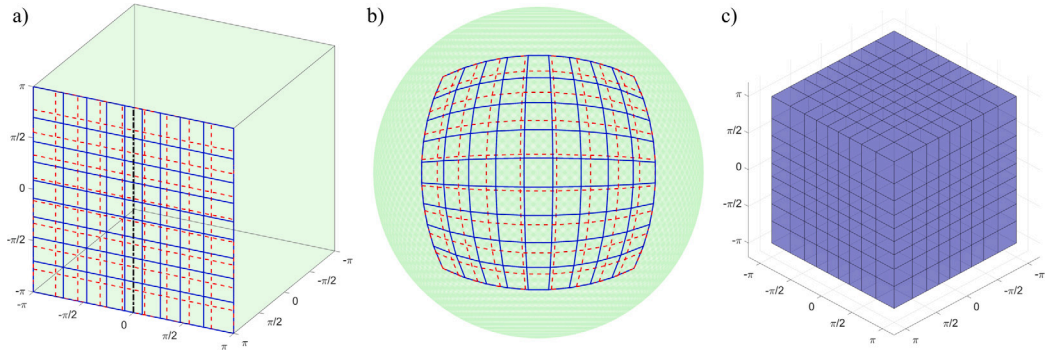

Fig. 2. (a) Constant bin size partition (red dotted line) and our proposed distortion reduction partition (blue line). The black dot-dash line represents the symmetry axis of the cube's face, employed to evaluate the proper bin size for the distortion reduction partition. (b) Comparison between the grid on the sphere surface that results from applying the inverse transformation on the constant size bin (red dotted line) and the distortion reduction partition (blue line). (c) Relative orientation symbols, resulting from our proposed distortion reduction partition with 9 elements per direction.

where $[x_c,\ y_c,\ z_c]^\intercal$ is the corresponding position in the Cartesian space. Radial stretching provides a computationally simple way to map a sphere into a cube and vice-versa. However, the transformation introduces radial and angular distortion. The equally spaced red dotted line grid lying on one of the cube's faces, shown in Fig. 2a, is the result of applying radial stretching to the uneven red dotted grid of Fig. 2b. Partitioning the cube with a 3D grid of regular bin size would, therefore, induce a non-uniform distribution of symbols. To address this problem, the bin size is adjusted based on its position on the 3D grid. The proper size of each bin is evaluated empirically with a procedure

Table 2
Evaluation of the magnitude of the symbols distortion by the standard deviation of the distribution of symbols' activation.

Type of partition:	Sphere, Constant Bin Size Partition	Radial Stretching, Constant Bin Size Partition (Lambers, 2020)	Radial Stretching, Adjusted Bin Size Partition (this paper)
Standard Deviation:	101.31	47.19	24.63

that starts by finding the cube lines affected by the highest magnitude of distortion. Considering the Radial Stretching mapping, these lines are the axes of symmetry of the faces of the cube, as the one shown in black in Fig. 2a, which are mapped as arcs on the sphere's surface. A constant bin size partition would divide the line as a composition of segments of the same length. When these segments are mapped to arcs on the sphere, they are distorted, making them uneven. A proper partition should be capable of dividing the line into segments that, once mapped on the sphere, produce arcs of the same length. To this end, it is possible to generate a set of equidistant points on the sphere's arc and map them to the cube line to evaluate the distribution of the points on the only varying coordinates affected by the distortion. Indeed, by evaluating the percentiles of the distribution, we can find the boundaries of the bins that ensure a more uniform size of symbols. The percentiles coinciding with the bins' boundaries can be chosen depending on the number of symbols one needs to reach the required descriptive accuracy. An example of the partition and the reduction of distortion this choice provides, visualized on the sphere surface, is represented with a blue grid on Fig. 2a–b. The resulting 3D partition, representing the relative orientation symbols, is shown in Fig. 2c.

To quantify the improvement in terms of a more homogeneous symbols' activation, we performed a comparison between a baseline obtained by a direct partition of uniform size that circumscribes the 3D sphere representing the space of relative orientation, a constant bin size of the cube obtained by stretching the sphere, and the proposed distortion reduction partition. Each partition comprises 10^6 symbols, and their activation is counted in front of 10^8 random rotations. The ideal partition would have its symbols' activated, on average, 100 times, with a minimal deviation. Therefore, we used the standard deviation of each symbol's activation as the evaluation criterion. Table 2 lists the results of the study. Our proposed distortion reduction partition achieves a reduction of standard deviation with respect to the direct sphere partition and the constant bin size partition equal to, respectively, 75.68% and 22.27%. Even if the magnitude of the distortion problem is significantly reduced, it is not entirely solved. With the Radial Stretching mapping, considering one face of the cube, the points on one edge and the points on the symmetry axis parallel to that edge are mapped to two arcs of the sphere with different lengths. Therefore, depending on where the size of the bins for the distortion reduction is evaluated, we can solve the distortion in that area, but the result is not valid everywhere else on the sphere. Nevertheless, results show that our approach indeed reduces the deviation of the symbols' activation, providing the descriptive capability required to distinguish different robotic manipulation skills, as shown in the case study of Section 5.

4.3. Effectiveness of the proposed spatial symbols

The methodology introduced in this Section defines two novel sets of general-use symbols to quantitatively describe the relative pose between entities. The effectiveness of the proposed spatial symbolic representation is provided by three key properties, whose relevance was highlighted in the introduction of Section 4:

1. **Generalizability:** Since the symbols are grounded solely based on the relative pose measure between the two objects' reference frames, they are designed to be agnostic to the specific relation they describe. Moreover, the symbols are generally applicable to all object types, as they disregard the shape of the objects and consider only the spatial pose of their reference frames. Finally, the symbols can be quickly tuned to meet the required descriptive accuracy by adjusting the number of elements in the 3D partitions.
2. **Light Computational Grounding:** The symbols are retrieved from an indexable 3D partition of the space of relative positions and orientations. Therefore, assessing the activation of a relative position or orientation symbol consists of discretizing the three components of the measure (either p_x, p_y, p_z for the relative position, or x_c, y_c, z_c from Eq. (1) for the relative orientation) into the 1D finite sets of ordered bins found along each of the three directions of the partition. For the relative position symbols, since the 1D partitions are indexable and bins have equal size, the bin search has a complexity of $\mathcal{O}(1)$ (Cormen et al., 2022). Conversely, for the indexable 1D partition with bins of non-constant size used for the relative orientation symbols, the bin search is implemented efficiently via binary search, with a worst-case complexity of $\mathcal{O}(\log n)$, where n is the number of bins (Cormen et al., 2022). Once the three bin indices are found, it is straightforward to determine which element of the 3D partition the measure falls into, making the symbols' grounding computationally lightweight.
3. **Uniform Distribution of the Symbols:** The 3D partition with constant element size in the Cartesian space of relative positions ensures a uniform distribution of symbols, avoiding the undesired scenario in which only a subset of symbols represents most of the relative position space. When considering the non-Cartesian space of relative orientation, a direct partition was found to be unsuitable, as summarized in Table 2. However, the adopted sphere-to-cube transform, combined with a non-constant size 3D partition, has been shown to greatly improve the uniformity of symbol activation, providing the descriptive capability required to distinguish different relative orientations.

5. Use case: Machine tending

The symbolic representation of spatial relations presented in this paper allows one to describe the semantics of robotic manipulation skills with an accuracy that can be tuned depending on the relation of interest. The validation of the methodology addresses a machine tending task, where the cobot is placed in proximity of a machine tool and has to tend a workpiece between the machine jaws. Machine tending is a repetitive task well-suited for automation with cobots. Indeed, after structuring the work-cell so that the workpieces are brought with a predefined pose, the robot can perform the task consistently with fixed movements. However, this setup increases re-configuration time whenever a new product is introduced. Enabling the cobot to reorient the workpiece to the target orientation required to feed it in the machine jaws removes these constraints, reducing downtimes due to production changes.

Defining a cobot program that adapts to random initial orientations and computes a regrasping plan is difficult, time-consuming, and requires programming expertise. Indeed, semantic-informed PbD can greatly simplify cobot deployment for machine tending tasks involving reorientations. In the teaching phase, non-experts can program the necessary actions, comprising reorientations. At the execution, the system retrieves the initial symbolic state and uses a symbolic planner such as Fast Downward (Helmert, 2006), employed in this study, to compute the skill sequence to reorient and load the part in the

Table 3
Parameters for the symbolic description of the machine tending skills. The number of symbols depicted is intended for one direction, with the total being written between brackets. (PS: Position Symbols; OS: Orientation Symbols; TCP: Tool Center Point; WRF: World Reference Frame).

Entity #1	Entity #2	PS - Threshold	PS - Number	OS - Number	Meaning
Gripper TCP	Workpiece	0.7 m	1 (1)	5 (125)	Different poses of the Workpiece
WRF	Door	1.1 m	4 (64)	1 (1)	Door is open or closed
Door	Workpiece	0.3 m	1 (1)	1 (1)	Support symbol to include Door in the loading skill
Machine Jaws	Workpiece	0.1 m	1 (1)	1 (1)	Workpiece being loaded

machine. Moreover, we introduced a replanning strategy that updates the symbolic state after each planning step and recomputes the skill sequence as needed. This provides robustness to execution failures or external interventions.

Semantic-informed PbD works on top of a symbolic description of the scene that could represent how the robot's action modifies the state. Since machine tending involves part reorientation, a symbolic representation of relative orientations is essential. With any of the approaches listed in Table 1, the system would not infer the meaning of a reorientation action, as it would lack the capability to spot any change in the symbolic state. However, it may be argued that a set of symbols could be manually defined with a hand-tailored grounding for this application. For instance, the workpiece orientation could be described using the face on which the workpiece is lying. Such information can be gathered by comparing the object pose with the most similar stable pose computed from a 3D model of the object. Indeed, semantic-informed PbD is a programming approach that allows non-experts to teach robot skills and deploy the robot for a complex task. However, manually defining symbols is time-consuming and requires robotics and possibly advanced programming expertise. Therefore, a non-expert still has to face a challenge before being able to use the methodology to program the robot. The proposed symbolic representation helps overcome this problem by letting non-experts easily adapt general-purpose symbols to any spatial relation. The use case demonstrates that the proposed symbols generalize across spatial relations, removing the need for hard-coded groundings.

The remainder of the Section is structured as follows. Section 5.1 introduces the laboratory setup of the machine tending task, together with the predicates employed and the skills to be taught by the users. Section 5.2 provides the criteria for evaluating the effectiveness of the proposed symbols. Moreover, the section outlines the user study conducted to assess the usability of semantic-informed PbD in a machine tending task.

5.1. Experimental setup

The experimental validation is conducted on a laboratory mockup of a machine tending task, shown in Fig. 3. The 3D-printed workpiece has nine stable poses, as shown in Fig. 4, but only one, *Pose 0*, allows it to be fixed between the machine jaws. A Doosan A0509s robot performs the task and moves the machine tool door. An in-hand Intel RealSense D435i camera captures images during a predefined scanning routine, estimating the workpiece and door poses using a contour-based algorithm and ArUco marker detection, respectively. A push-button interface allows users to control the demonstration phase. The entire system runs on a laptop with an Intel Core i7-11800H processor.

To setup the cobot for the machine tending task, the user has to program the skills for reorienting the part, opening/closing the door, and loading the workpiece in the machine's jaws. Fig. 5 reports the set of skills. The predicates employed to semantically characterize the scene should be capable of describing the changes in the workspace taking place during the demonstrations. At first, the entities of interest must be picked. Then, for each relation, a proper threshold for the relative position symbol activation and the number of symbols to describe the relative position and orientation should be selected based on the required descriptive accuracy. Table 3 provides the symbols' parameters employed in the user study. Referring to the skills listed in Fig. 5, the spatial relation symbols used to semantically describe the actions have been tuned as follows:

- **Door opening/closing:** To characterize the door state, the position of the ArUco marker, acting as a reference frame for the door, must be assessed in the robot base frame, which acts as a fixed world reference frame. As shown in Fig. 6-a, the sliding door stroke is aligned with the y-direction of the robot base frame, with its midpoint located 0.55 m from the origin of the frame along that direction. The portion of space described by the relative position symbols is a cube centered on the reference entity. Therefore, setting the symbols with a threshold of 1.1 m and four symbols per direction makes the midpoint of the sliding door stroke the boundary between two symbols along the y-direction, ensuring that two distinct symbols mark the door opening and closing. Given the irrelevance of the door's orientation, the number of relative orientation symbols has been set to one.
- **Workpiece reorientation:** To characterize on which face the workpiece is resting on the stand placed in front of the robot, the relative orientation between the gripper tool center point in the scanning position and the workpiece reference frame is evaluated. Since rotating the workpiece along one of the three directions of its reference frame can place it in at most five stable poses, as shown in Fig. 6-b, five relative orientation symbols per direction, making a total of 125 symbols, are sufficient to represent the entire range of possible orientations in which the workpiece can be placed on the stand. Given that when the robot is moved to the scanning pose, the distance between the gripper tool center point and the workpiece is at most 0.7 m, that value is set as the threshold for the relative position symbol. A single symbol is sufficient due to the irrelevance of how the relative position between the two entities is modified by the reorientation skill.
- **Workpiece loading:** The loading action requires a symbol to characterize when the workpiece is placed between the machine jaws. Therefore, a single relative position symbol can be defined, activated whenever the workpiece is placed within 0.1 m from the origin of the machine jaws reference frame. Given the irrelevance of its pose, knowing that only a single orientation is permitted inside the machine jaws, the number of relative orientation symbols is set to one. Finally, a relative position symbol is added to describe the relation between the workpiece and the machine door at the end of the loading skill. This guarantees that the requirement for the door to be open during loading is correctly captured in the preconditions (Zanchettin, 2023).

As illustrated by the explanation above, deploying the solution requires the user to specify only a few task-related parameters, without the need to define new symbols or manage their grounding manually.

A graph-like representation of the workpiece stable poses is provided to the users to better organize the reorientation skills that have been programmed so far, avoiding repetitions. Fig. 7 shows a sample graph, where the stable poses of the workpiece are represented by nodes and every feasible reorientation skill as an oriented arc connecting the nodes. The graph is not fully connected because some reorientations from a pose to a target one are not executable in a single step due to the gripper encumbrance and require the robot to regrasp the object, leaving it in an intermediate stable pose. Finally, as a reference, a sample demonstration of a reorientation skill, along with its encoding in PDDL, is shown in Fig. 8.

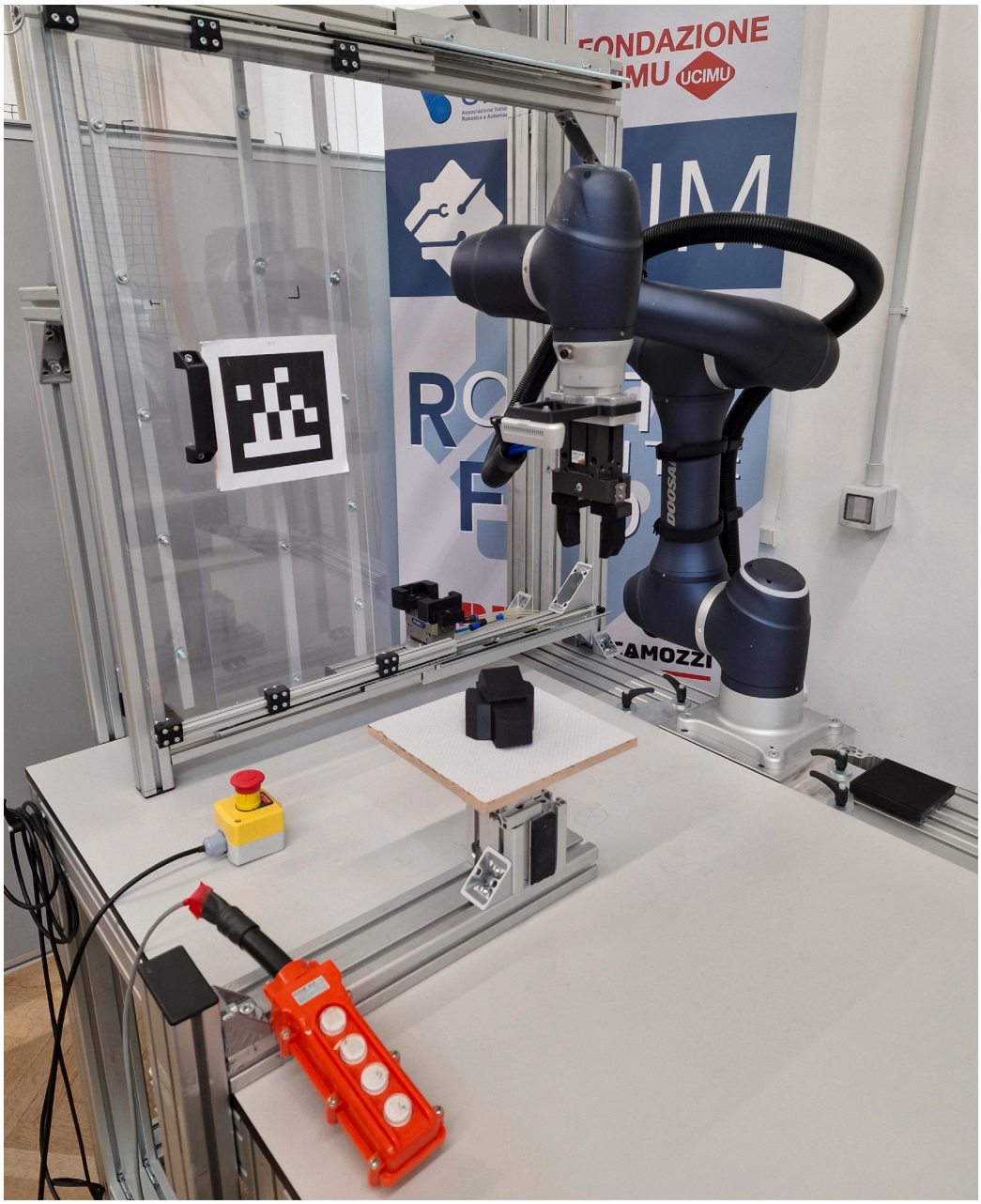

Fig. 3. Experimental setup, consisting in a Doosan A0509 robot, a mockup machine tool, an in-hand mounted Intel Realsense d435i camera, a push-button user interface, and the workpiece to be loaded in the machine.

5.2. Evaluation criteria

The validation addresses two main claims. First, we investigate whether the symbols introduced in this paper adequately describe all the spatial relations for a machine tending task comprising parts reorientation. The demonstrations are performed by an expert operator teaching the complete set of feasible reorientation, shown in Fig. 7, together with the skills to open and close the machine tool door and to load the workpiece in the machine jaws. The workspace is set with the machine door closed and the workpiece in one of the available stable poses. Then, after setting the machine tool loaded with a closed door as the goal condition, the symbolic planner evaluates the plan to solve the task. Following the computed plan, the system starts executing the sequence and, after every skill is executed, runs the routine to update the pose of the entities in the scene and the semantic state before passing on to the next action. After repeating the experiment from every stable pose, it is investigated how long it takes the planner, on average, to solve the task and whether the execution leads to a successful loading of the part, even in the case of unforeseen changes in the environment.

The second stage of the experimental validation involves ten candidates, all of whom stated none or minimal experience with robots (less than 1 h). The users are provided with guidelines on how to interact with the system, an overview of the task, and what is intended as a robot skill. The experiment was conducted to evaluate the ease of use and intuitiveness of the programming solution when deployed for a real industrial task, assessing the following aspects:

- **Test 1:** The user can correctly identify the skills required for the robot to complete the task. After receiving preliminary instructions, the user is asked to list the actions that must be demonstrated. The test is considered passed if the four basic skills depicted in Fig. 5 are correctly identified.

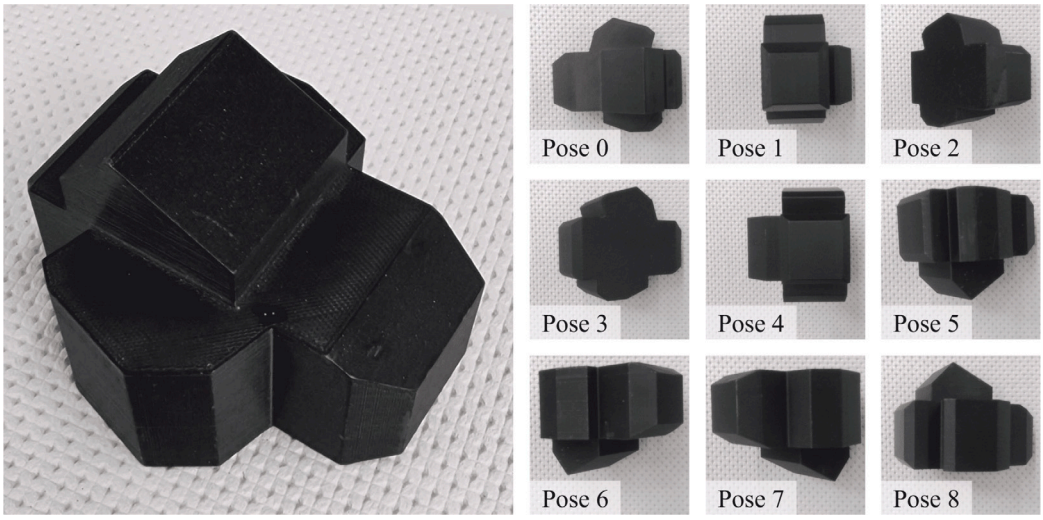

Fig. 4. Workpiece to be reoriented with its 9 stable poses.

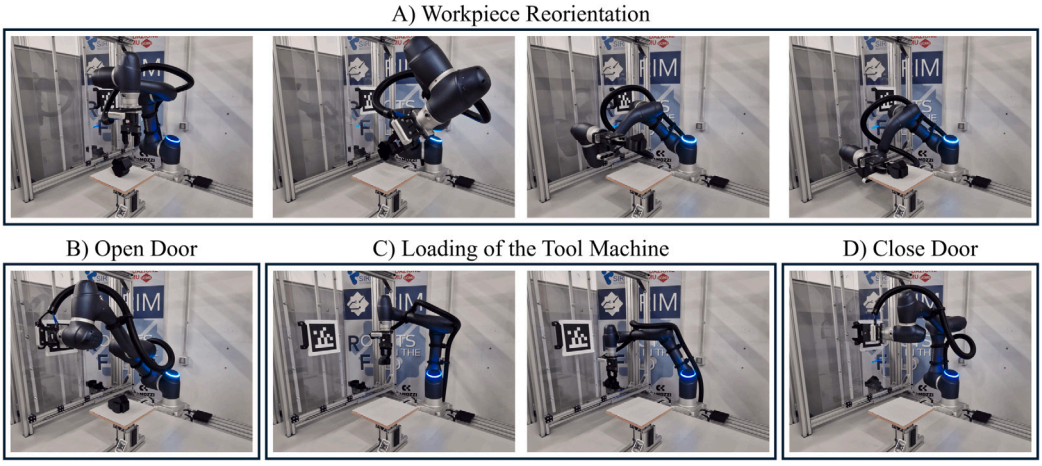

Fig. 5. Set of skills the robot requires to perform the machine tending task.

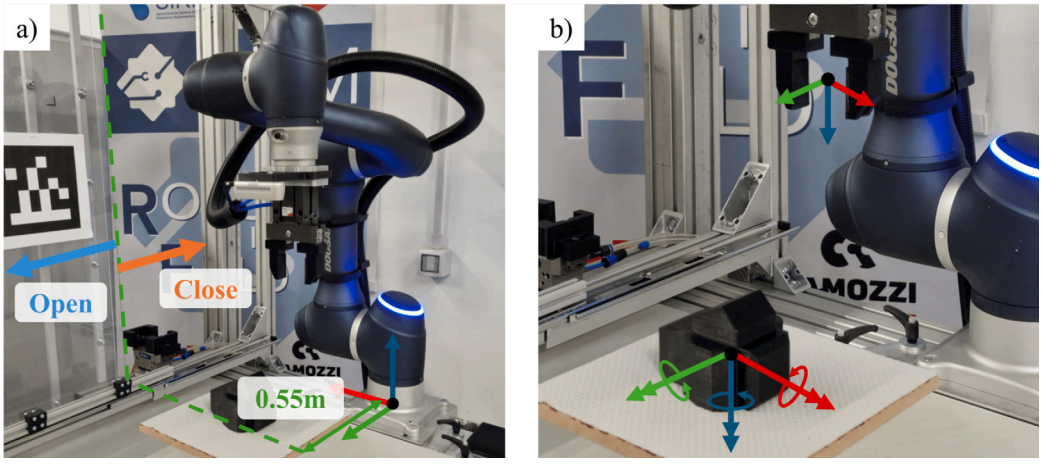

Fig. 6. Measures taken for tuning the symbols to describe (a) the machine door state via the relative position between its marker and the world reference frame, and (b) the workpiece pose via the relative orientation between the gripper tool center point and the workpiece reference frame.

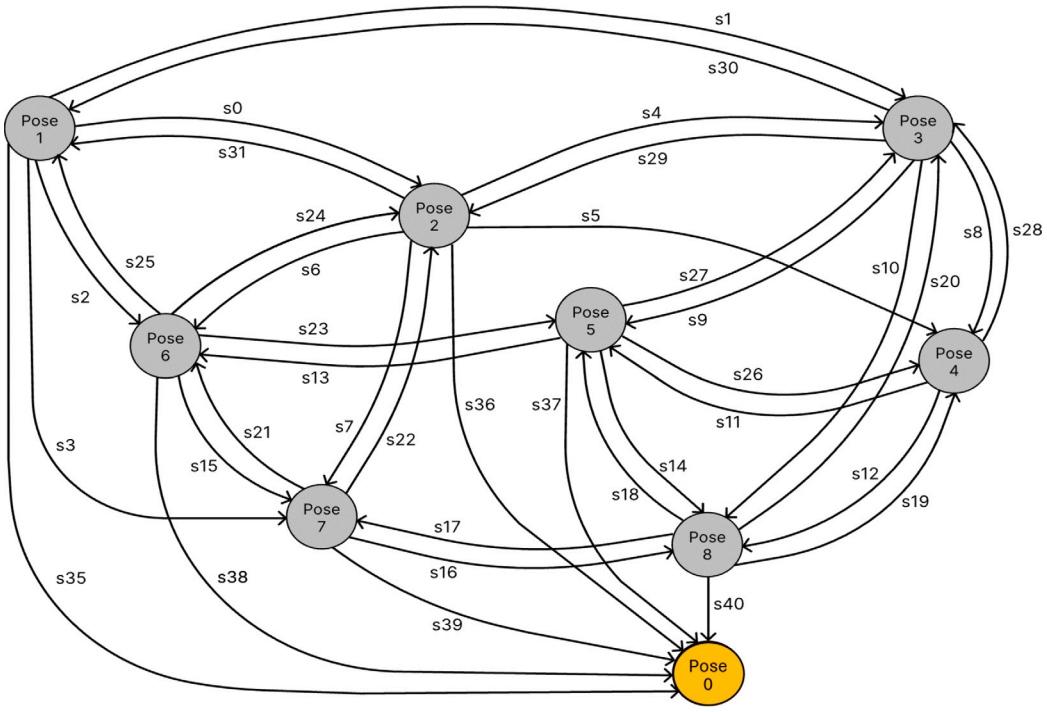

Fig. 7. Graph-like representation to show the object poses and the reorientation skills. Pose 0 is highlighted in yellow since it is the only pose from which the workpiece can be loaded in the machine jaws.

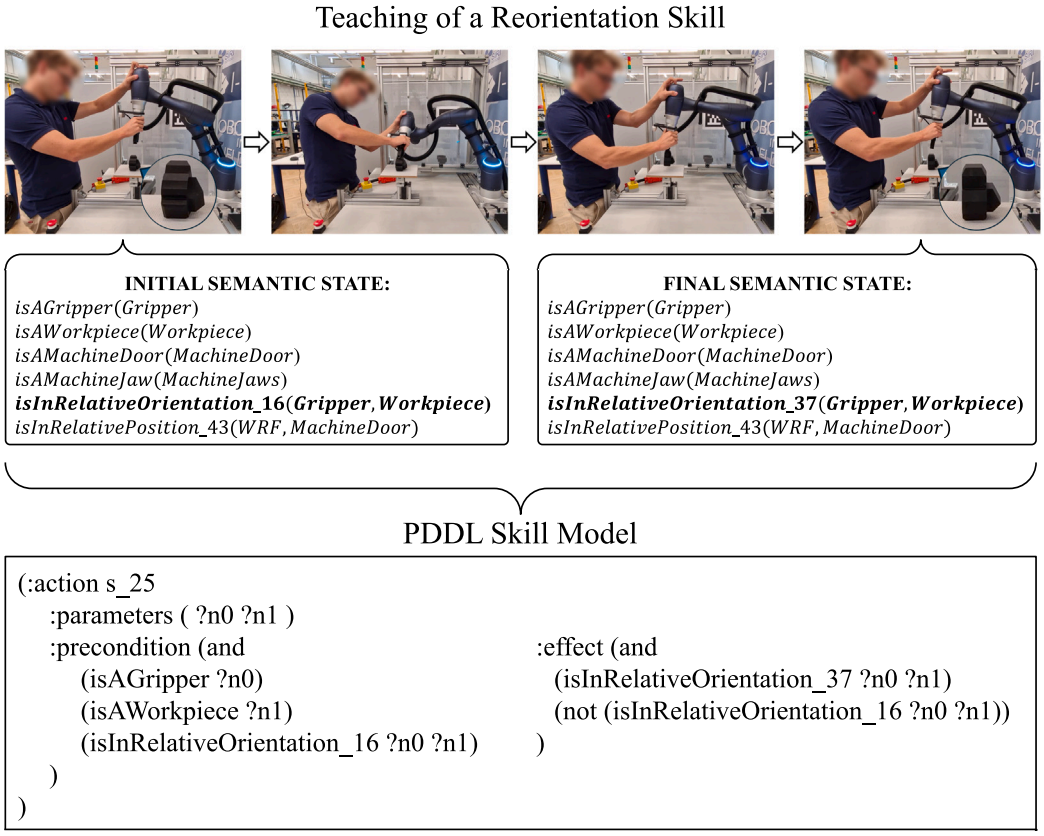

Fig. 8. Sample demonstration of the reorientation skill (s_25 in Fig. 7) to bring the workpiece from Pose 6 to Pose 1, and its corresponding symbolic encoding in PDDL. The figure exemplifies the teaching process described in Section 3.2, as proposed by Zanchettin (2023).

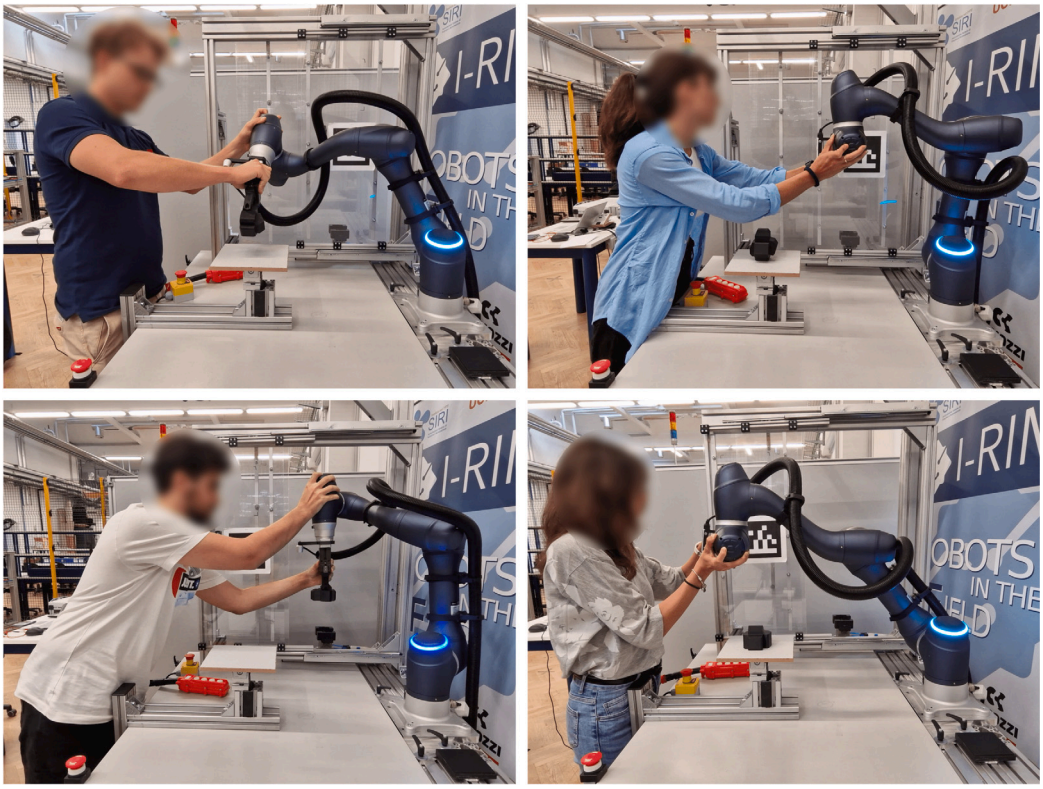

Fig. 9. Candidates teaching the skills by hand-guiding the robot arm.

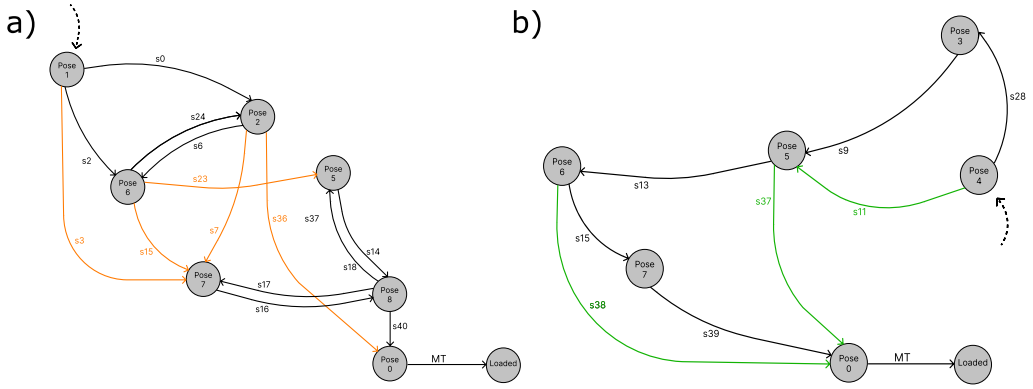

Fig. 10. Graphs provided to the users for Test 3 (a) and Test 4 (b). The dashed arrow indicates the initial pose of the workpiece. Black arrows represent the reorientation skills already available in the skill library. Orange and green arrows illustrate possible solutions to the respective tests: these arrows were not included in the versions shown to the users.

- **Test 2:** The user can properly teach the actions required for the task without the help of a supervisor. The user is provided only with the general guidelines for performing a demonstration. After the teaching phase is completed, it is assessed whether the planner can compute an executable plan that results in the workpiece being successfully loaded into the machine. Fig. 9 shows some snapshots from the teaching phase.
- **Test 3:** The user can understand whether a failure is caused by something missing in the skill library. The system has an incomplete library that misses some reorientation actions. Fig. 10-a illustrates the situation, with the workpiece initially placed in Pose 1 and the skills already available in the library shown as black arrows. As can be seen from the image, without any additional reorientation skills, the planner is unable to compute a sequence of actions capable of loading the machine tool. The user is asked to assess the situation by examining the graph shown in Fig. 10-a, excluding the possible solutions represented by orange arrows, and to teach one of the reorientation skills required to bridge the two unconnected parts of the graph. It is assessed whether the user can teach one of the skills represented by the orange arrows, thereby leading to a successful tending of the machine.
- **Test 4:** The user can cope with inefficiencies in the tending process given by an incomplete skill library. As shown in Fig. 10-b, the workpiece is initially placed in Pose 4, and the current library, represented by the black arrows connecting the nodes, is sufficient to execute the task but results in a longer sequence of actions due to missing reorientation skills between poses. The user is asked to address this inefficiency by examining the graph shown in Fig. 10-b (excluding the possible solutions represented by the green arrows), identifying and teaching the skills that could reduce the sequence length. It is then assessed whether the

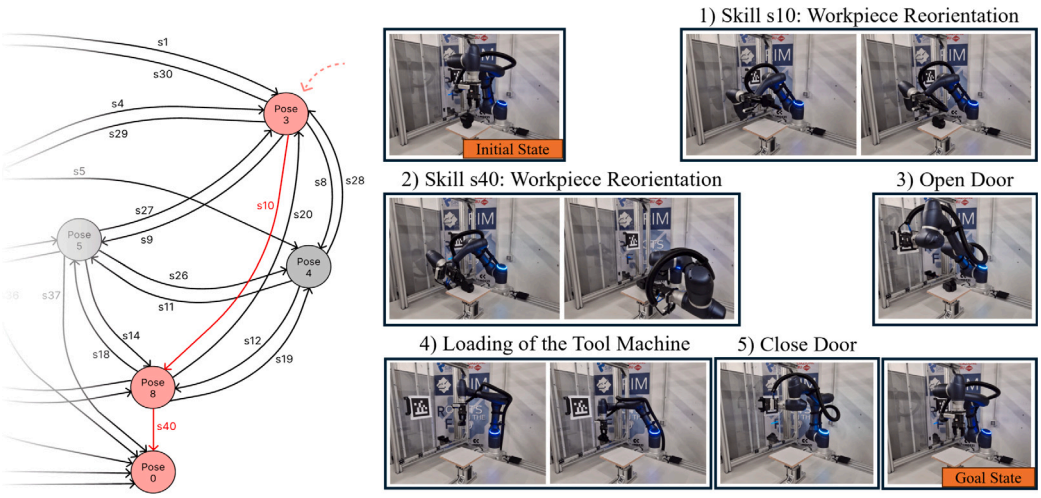

Fig. 11. Optimal plan computed by the semantic planner and execution by the robot. The graph on the left shows only the part of the plan concerning the reorientations.

Table 4
Results of the user study outlined in Section 5.2.

Test 1: Skill Segmentation	Test 2: Skill Teaching	Test 3: Missing Skill	Test 4: Path Optimization
5/10 (50%)	8/10 (80%)	8/10 (80%)	7/10 (70%)

additional teaching by the candidate leads to a better solution in terms of reduced plan length.

At the end of the experiments, the user is provided with a System Usability Scale (SUS) Questionnaire (Brooke et al., 1996) to assess the overall programming experience.

6. Results and discussion

6.1. First phase: Assessing symbols descriptive capabilities

The first phase of validation explores the capability of the proposed symbols to describe complex industrial tasks. Results show that the planner consistently computes the shortest sequence of actions to feed the part into the machine jaws, with an average computation time of 150 ms. Plan executions achieved a 100% success rate. A sample plan and its execution are shown in Fig. 11. These results demonstrate that the symbols effectively capture the semantics of the skills taught by demonstration, ensuring both reliable and optimal execution. The short planning time, easily masked by running the planner during the robot's homing phase after pose estimation, enables replanning after each action. Indeed, if the post-action scene does not match the expected symbolic state due to execution failures or external disturbances, the planner recomputes a new plan to recover and complete the task. Examples of replanning are included in the accompanying video.[1]

6.2. Second phase: User study results

The second phase of the experimental campaign aims to assess the system's usability and intuitiveness by engaging ten volunteers without expertise in robot programming. The results collected in the user study outlined in Section 5.2, are listed in Table 4.

Results from the segmentation experiment (Test 1) show that half of the first-time users struggled to grasp the concept of robot skills. Instead of viewing skills as modular blocks of parameterized movements with preconditions and effects, they often defined basic motions (e.g., lifting the object) or gripper actions (e.g., closing the gripper) as standalone skills. Although this over-segmentation would not affect planning outcomes, it conflicts with the experimental setup. Specifically, the in-hand camera cannot estimate the pose once the object is grasped, preventing semantic state updates after such actions.

Results from the teaching phase (Test 2) show that demonstrations led to successful executions in 80% of attempts. Hand-guiding was intuitive and accessible for first-time users. The failures were mainly due to poorly chosen waypoints, which, when projected into the current object reference frame, caused the robot to reach joint limits. With more experience, such issues could likely be avoided. Nonetheless, the results are highly satisfactory, as 80% of first-time users were able to deploy the cobot for such a complex task.

Test 3 and Test 4 assessed the users' ability to either overcome a failure or optimize the process by adding skills to the library. Results show a clear improvement from the initial segmentation test, indicating that after the teaching phase, users were more likely to approach the task as a composition of robot skills. Indeed, only two users failed to teach the missing reorientation skill, and 70% understood how to speed up the execution.

Finally, the SUS questionnaires administered to the candidates provided insights into the usability and intuitiveness of the methodology. Fig. 12 shows the aggregate distribution of responses to the ten questions of the SUS questionnaire. The computed SUS scores for each candidate are listed in Table 5. The average score of the questionnaires equals 84.83/100, reaching an A grade and placing the methodology in the 90th–95th percentile (Lewis and Sauro, 2009) of SUS scores, indicating very high usability. In particular, 80% of the users found this robot programming approach intuitive and easy to learn without requiring support, as highlighted by Q3 and Q4 in Fig. 12.

7. Conclusions

This paper introduced a novel formulation of symbols to semantically characterize spatial relations between entities in the robot workspace. We address the limitations of state-of-the-art semantic PbD interfaces, which use hand-tailored symbols to characterize the skills.

[1] https://youtu.be/6wDCu9XEbXI

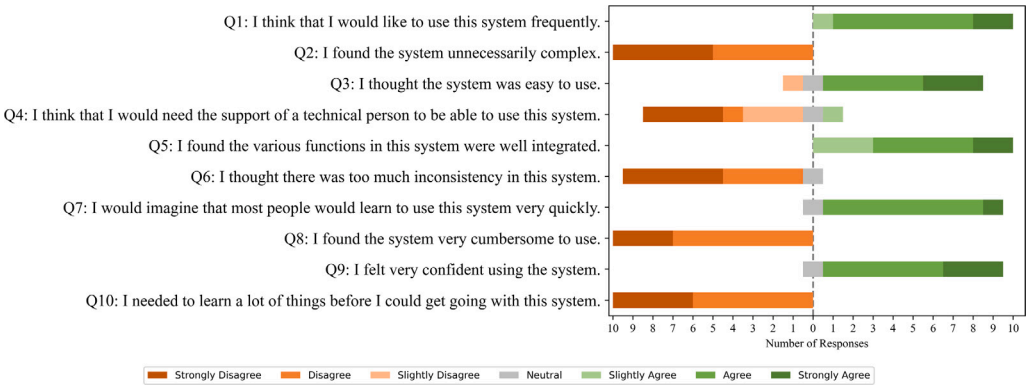

Fig. 12. Aggregate distribution of responses from the ten candidates to the SUS questionnaire.

Table 5
SUS scores from the ten candidates.

Candidate ID	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
SUS Score	76.67	91.67	88.33	81.67	81.67	81.67	80.00	85.00	86.67	95.00

These symbols only describe qualitative relations that are difficult to generalize and lack the descriptive accuracy required for complex tasks. We propose to retrieve the symbols from a proper partition of the three-dimensional spaces describing relative positions and relative orientations. The partitions can be tuned to meet the required accuracy and can be applied to any relation of interest. The procedure guarantees to define a set of indexable symbols that describe spatial relations in a quantitative manner and can be grounded with low computational efforts. We validated our symbols' capabilities to describe all the relevant relations of a machine tending task comprising part reorientations. Results show that our symbols enable the system to properly understand the semantics of the skills taught by demonstration, leading to the workpiece being loaded into the machine at the end of the execution of the plan computed by the symbolic planner. Finally, we investigated the usability of the teaching system when applied to a real industrial task through a series of tests involving non-expert users. First-time users struggled to understand the concept of a task as a composition of robot skills, with half of them failing to understand what actions are required for the task. However, after the teaching phase, which resulted in 80% of the users being able to deploy the cobot for the task, the users grew confident with the programming approach and were able to manipulate the skill library to overcome execution failures and optimize the process.

To advance the methodology presented in this paper toward real-world deployment, future research should focus on increasing its technology readiness level, particularly when applied to robot programming by demonstration scenarios. More precisely, future studies can address the development of an interface to support the user in defining the symbols' parameters required for the task. Indeed, in this work, we did not investigate how easily the users can tune the symbols since the focus was on validating the descriptive capabilities of the symbols themselves. However, given the geometrical nature and interpretability of the symbols, the user should be capable of assessing the location of the entities in the workspace and defining the symbols accordingly. We also foster the application of the methodology on factory ground, involving operators to test the programming approach thoroughly. To this end, more focus should be put on developing a graphical user interface to support the teaching process and help the user interact with the cobot. Moreover, within an industrial setting, the intuitiveness and ease of use of the programming approach should be thoroughly tested through both objective and subjective measures. For example, tracking the number of support requests from operators, counting the interactions required before the cobot autonomously completes tasks, and analyzing the learning curve over repeated programming sessions with different tasks, can provide valuable insights. Additionally, assessing the operator workload using tools such as the NASA-TLX (Hart and Staveland, 1988) can further substantiate claims of ease of use.

We also propose enhancements to the symbolic description that would relieve users from manually tuning parameters during pre-deployment. This can be achieved by enabling the system to autonomously identify relevant spatial relations and incrementally adjust parameters throughout the teaching phase, as it observes all possible scene transitions following skill demonstrations. Indeed, some studies target to extract spatial symbols directly from multiple demonstrations, without relying on a pre-coded, tunable symbol grounding framework like the one proposed in this paper (Konidaris et al., 2018; Ahmetoglu et al., 2022). However, such approaches conflict with the industrial requirement of a rapid deployment of the robot. Therefore, developing methods to autonomously define relevant symbols from a single demonstration, without limiting applicability to a narrow set of tasks, remains a promising direction.

Finally, although this paper applies the proposed methodology to kinesthetic Programming by Demonstration with collaborative robots, the proposed spatial symbols have broader applicability. These symbols can be integrated into skill programming frameworks using alternative teaching modalities such as teleoperation, virtual reality, or observational learning. In fact, extending the approach to a diverse set of teaching modalities can pave the way for applying the semantic skill programming methodology to traditional industrial robots and humanoids. Furthermore, these symbols could support other areas of spatial reasoning, including language cue processing for defining objectives in human–robot interaction, initial and goal specification for automated planning, and assembly quality inspection by comparing the current and target positioning of components. In general, whenever a symbolic representation of the system's state requires information about the relative poses of entities, the symbols introduced in this paper can be particularly beneficial due to their adaptability and low computational grounding cost. This highlights the versatility of the proposed spatial symbolic representation beyond the specific cobot Programming by Demonstration scenario described in this paper.

CRediT authorship contribution statement

Isacco Zappa: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Sara Vignali:** Validation, Software, Investigation, Formal analysis, Data curation. **Andrea Maria Zanchettin:** Writing – review & editing, Supervision, Resources, Conceptualization. **Paolo Rocco:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Paolo Rocco reports financial support was provided by MUR. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This study was partially carried out within the MICS (Made in Italy – Circular and Sustainable) Extended Partnership and received funding from Next-Generation EU (Italian PNRR – M4 C2, Invest 1.3 – D.D. 1551.11-10-2022, PE00000004). CUP MICS D43C22003120001.

Data availability

Data will be made available on request.

References

Aein, M.J., Aksoy, E.E., Wörgötter, F., 2019. Library of actions: Implementing a generic robot execution framework by using manipulation action semantics. Int. J. Robot. Res. 38 (8), 910–934.

Ahmetoglu, A., Seker, M.Y., Piater, J., Oztop, E., Ugur, E., 2022. Deepsym: Deep symbol generation and rule learning for planning from unsupervised robot interaction. J. Artificial Intelligence Res. 75, 709–745.

Aksoy, E.E., Abramov, A., Dörr, J., Ning, K., Dellen, B., Wörgötter, F., 2011. Learning the semantics of object–action relations by observation. Int. J. Robot. Res. 30 (10), 1229–1249.

Aksoy, E.E., Tamosiunaite, M., Wörgötter, F., 2015. Model-free incremental learning of the semantics of manipulation actions. Robot. Auton. Syst. 71, 118–133.

Berretti, S., Del Bimbo, A., 2006. Modeling spatial relationships between 3d objects. ICPR'06, In: 18th International Conference on Pattern Recognition, vol. 1, IEEE, pp. 119–122.

Bredeweg, B., Struss, P., 2003. Current topics in qualitative reasoning. AI Mag. 24 (4), 13–13.

Brooke, J., et al., 1996. SUS-A quick and dirty usability scale. Usability Eval. Ind. 189 (194), 4–7.

Chen, S., Guhur, P.-L., Tapaswi, M., Schmid, C., Laptev, I., 2022. Language conditioned spatial relation reasoning for 3d object grounding. Adv. Neural Inf. Process. Syst. 35, 20522–20535.

Chen, B., Xu, Z., Kirmani, S., Ichter, B., Sadigh, D., Guibas, L., Xia, F., 2024. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14455–14465.

Cheng, A.-C., Yin, H., Fu, Y., Guo, Q., Yang, R., Kautz, J., Wang, X., Liu, S., 2024. SpatialRGPT: Grounded spatial reasoning in vision language models. arXiv preprint arXiv:2406.01584.

Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C., 2022. Introduction to Algorithms. MIT Press.

Diehl, M., Paxton, C., Ramirez-Amaro, K., 2021. Automated generation of robotic planning domains from observations. In: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 6732–6738.

Diehl, M., Zappa, I., Zanchettin, A.M., Ramirez-Amaro, K., 2024. Learning robot skills from demonstration for multi-agent planning. In: 2024 IEEE 20th International Conference on Automation Science and Engineering. CASE, IEEE, pp. 2348–2355.

Dreher, C.R., Wächter, M., Asfour, T., 2019. Learning object-action relations from bimanual human demonstration using graph networks. IEEE Robot. Autom. Lett. 5 (1), 187–194.

Eiband, T., Liebl, J., Willibald, C., Lee, D., 2023. Online task segmentation by merging symbolic and data-driven skill recognition during kinesthetic teaching. Robot. Auton. Syst. 162, 104367.

Fathi, Z., Ramirez, W.F., Tavares, A., Gilliland, G., 1993. Symbolic reasoning and quantitative analysis for fault detection and isolation in process plants. Eng. Appl. Artif. Intell. 6 (3), 203–218.

Fichtl, S., McManus, A., Mustafa, W., Kraft, D., Krüger, N., Guerin, F., 2014. Learning spatial relationships from 3D vision using histograms. In: 2014 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 501–508.

Freksa, C., 2015. Strong spatial cognition. In: International Conference on Spatial Information Theory. Springer, pp. 65–86.

Freksa, C., van de Ven, J., Wolter, D., 2018. Formal representation of qualitative direction. Int. J. Geogr. Inf. Sci. 32 (12), 2514–2534.

Gemignani, G., Capobianco, R., Nardi, D., 2015. Approaching qualitative spatial reasoning about distances and directions in robotics. In: Congress of the Italian Association for Artificial Intelligence. Springer, pp. 452–464.

George, P., Cheng, C.-T., Pang, T.Y., Neville, K., 2023. Task complexity and the skills dilemma in the programming and control of collaborative robots for manufacturing. Appl. Sci. 13 (7), 4635.

Ghallab, M., Nau, D., Traverso, P., 1998. PDDL - The Planning Domain Definition Language. Tech. rep., Technical Report, DCS TR-1165, Yale Center for Computational Vision and Control.

Guadarrama, S., Riano, L., Golland, D., Go, D., Jia, Y., Klein, D., Abbeel, P., Darrell, T., et al., 2013. Grounding spatial relations for human-robot interaction. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 1640–1647.

Hart, S.G., Staveland, L.E., 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In: Advances in Psychology, vol. 52, Elsevier, pp. 139–183.

Helmert, M., 2006. The fast downward planning system. J. Artificial Intelligence Res. 26, 191–246.

Howard, T.M., Stump, E., Fink, J., Arkin, J., Paul, R., Park, D., Roy, S., Barber, D., Bendell, R., Schmeckpeper, K., et al., 2022. An intelligence architecture for grounded language communication with field robots. Field Robot. 2, 468–512.

Jiang, J., He, Z., Zhang, S., Zhao, X., Tan, J., 2021. Learning to transfer focus of graph neural network for scene graph parsing. Pattern Recognit. 112, 107707.

Kartmann, R., Liu, D., Asfour, T., 2021. Semantic scene manipulation based on 3D spatial object relations and language instructions. In: 2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids). IEEE, pp. 306–313.

Konidaris, G., Kaelbling, L.P., Lozano-Perez, T., 2018. From skills to symbols: Learning symbolic representations for abstract high-level planning. J. Artificial Intelligence Res. 61, 215–289.

Krishnamurthy, J., Kollar, T., 2013. Jointly learning to parse and perceive: Connecting natural language to the physical world. Trans. Assoc. Comput. Linguist. 1, 193–206.

Kulick, J., Toussaint, M., Lang, T., Lopes, M., 2013. Active learning for teaching a robot grounded relational symbols. In: IJCAI. Citeseer, pp. 1451–1457.

Lambers, M., 2020. Survey of cube mapping methods in interactive computer graphics. Vis. Comput. 36 (5), 1043–1051.

Lee, S.U., Hofmann, A., Williams, B., 2019. A model-based human activity recognition for human–robot collaboration. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 736–743.

Lee, S.U., Hong, S., Hofmann, A., Williams, B., 2020. QSRNet: Estimating qualitative spatial representations from RGB-D images. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 8057–8064.

Lewis, J.R., Sauro, J., 2009. The factor structure of the system usability scale. In: Human Centered Design: First International Conference, HCD 2009, Held As Part of HCI International 2009, San Diego, CA, USA, July 19-24, 2009 Proceedings 1. Springer, pp. 94–103.

Li, C., Li, H., Chen, K., 2024. Convolutional point transformer for semantic segmentation of sewer sonar point clouds. Eng. Appl. Artif. Intell. 138, 109456.

Liang, Y.S., Pellier, D., Fiorino, H., Pesty, S., 2022. iRoPro: An interactive robot programming framework. Int. J. Soc. Robot. 14 (1), 177–191.

Liu, F., Emerson, G., Collier, N., 2023. Visual spatial reasoning. Trans. Assoc. Comput. Linguist. 11, 635–651.

Ma, W., Chen, H., Zhang, G., Chou, Y.-C., de Melo, C.M., Yuille, A., 2024. 3Dsrbench: A comprehensive 3d spatial reasoning benchmark. arXiv preprint arXiv:2412.07825.

Mees, O., Abdo, N., Mazuran, M., Burgard, W., 2017. Metric learning for generalizing spatial relations to new objects. In: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS, IEEE, pp. 3175–3182.

Mees, O., Emek, A., Vertens, J., Burgard, W., 2020. Learning object placements for relational instructions by hallucinating scene representations. In: 2020 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 94–100.

Mota, T., Sridharan, M., 2018. Learning the grounding of expressions for spatial relations between objects. In: Workshop on Perception, Inference and Learning for Joint Semantic, Geometric and Physical Understanding at ICRA 2018.

Park, J., Kim, Y.S., et al., 2007. Visual reasoning and design processes. In: DS 42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07. 2007. pp. 333–334.

Pedersen, M.R., Nalpantidis, L., Andersen, R.S., Schou, C., Bøgh, S., Krüger, V., Madsen, O., 2016. Robot skills for manufacturing: From concept to industrial deployment. Robot. Comput.-Integr. Manuf. 37.

Rosman, B., Ramamoorthy, S., 2011. Learning spatial relationships between objects. Int. J. Robot. Res. 30 (11), 1328–1342.
Shridhar, M., Mittal, D., Hsu, D., 2020. Ingress: Interactive visual grounding of referring expressions. Int. J. Robot. Res. 39 (2–3), 217–232.
Sjöö, K., Aydemir, A., Jensfelt, P., 2012. Topological spatial relations for active visual search. Robot. Auton. Syst. 60 (9), 1093–1107.
Sjöö, K., Jensfelt, P., 2011. Learning spatial relations from functional simulation. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems. IEEE, pp. 1513–1519.
Skubic, M., Perzanowski, D., Blisard, S., Schultz, A., Adams, W., Bugajska, M., Brock, D., 2004. Spatial language for human-robot dialogs. IEEE Trans. Syst. Man, Cybern. Part C (Applications Reviews) 34 (2), 154–167.
Steinmetz, F., Nitsch, V., Stulp, F., 2019. Intuitive task-level programming by demonstration through semantic skill recognition. IEEE Robot. Autom. Lett. 4 (4), 3742–3749.
Tan, J., Ju, Z., Liu, H., 2014. Grounding spatial relations in natural language by fuzzy representation for human-robot interaction. In: 2014 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE, IEEE, pp. 1743–1750.
Tellex, S., Kollar, T., Dickerson, S., Walter, M., Banerjee, A., Teller, S., Roy, N., 2011. Understanding natural language commands for robotic navigation and mobile manipulation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 25, (1), pp. 1507–1514.
Thippur, A., Burbridge, C., Kunze, L., Alberti, M., Folkesson, J., Jensfelt, P., Hawes, N., 2015. A comparison of qualitative and metric spatial relation models for scene understanding. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 29, (1).
Villani, V., Pini, F., Leali, F., Secchi, C., Fantuzzi, C., 2018. Survey on human-robot interaction for robot programming in industrial applications. Ifac-PapersOnline 51 (11), 66–71.
Wächter, M., Schulz, S., Asfour, T., Aksoy, E., Wörgötter, F., Dillmann, R., 2013. Action sequence reproduction based on automatic segmentation and object-action complexes. In: 2013 13th IEEE-RAS International Conference on Humanoid Robots (Humanoids). IEEE, pp. 189–195.
Wang, B., Chang, F., Liu, C., Wang, W., 2024. Human–robot interaction-oriented video understanding of human actions. Eng. Appl. Artif. Intell. 133, 108247.
Weintrop, D., Shepherd, D.C., Francis, P., Franklin, D., 2017. Blockly goes to work: Block-based programming for industrial robots. In: 2017 IEEE Blocks and beyond Workshop (B&B). IEEE.
Wu, X., Li, Y.-L., Sun, J., Lu, C., 2023. Symbol-LLM: leverage language models for symbolic system in visual human activity reasoning. Adv. Neural Inf. Process. Syst. 36, 29680–29691.
Yang, Z., Garrett, C., Fox, D., Lozano-Pérez, T., Kaelbling, L.P., 2024. Guiding long-horizon task and motion planning with vision language models. arXiv preprint arXiv:2410.02193.
Zampogiannis, K., Yang, Y., Fermüller, C., Aloimonos, Y., 2015. Learning the spatial semantics of manipulation actions through preposition grounding. In: 2015 IEEE International Conference on Robotics and Automation. ICRA, IEEE, pp. 1389–1396.
Zanchettin, A.M., 2023. Symbolic representation of what robots are taught in one demonstration. Robot. Auton. Syst. 166, 104452.
Zhu, R., Janowicz, K., Cai, L., Mai, G., 2022. Reasoning over higher-order qualitative spatial relations via spatially explicit neural networks. Int. J. Geogr. Inf. Sci. 36 (11), 2194–2225.