Inference of Human-derived Specifications of Object Placement via Demonstration

Alex Cuellar[1] , **Ho Chit Siu**[2] and **Julie A Shah**[1]
[1]Massachusetts Institute of Technology
[2]MIT Lincoln Laboratory
alexcuel@mit.edu, julie_a_shah@csail.mit.edu, hochit.siu@ll.mit.edu

Abstract

As robots' manipulation capabilities improve for pick-and-place tasks (e.g., object packing, sorting, and kitting), methods focused on understanding human-acceptable object configurations remain limited expressively with regard to capturing spatial relationships important to humans. To advance robotic understanding of human rules for object arrangement, we introduce positionally-augmented RCC (PARCC), a formal logic framework based on region connection calculus (RCC) for describing the relative position of objects in space. Additionally, we introduce an inference algorithm for learning PARCC specifications via demonstrations. Finally, we present the results from a human study, which demonstrate our framework's ability to capture a human's intended specification and the benefits of learning from demonstration approaches over human-provided specifications.

1 Introduction

As robots become a mainstay of industrial and manufacturing processes, pick-and-place tasks (e.g., packing, sorting, and kitting objects) have become central to many of their applications [Sanneman *et al.*, 2020]. While significant prior research has explored control and manipulation for pick-and-place tasks, less has examined how robots can understand human preferences about object arrangement. Erbayrak et al, for example, designed an algorithm to pack objects into multiple bins while optimizing to keep as many objects of the same predefined "family" together as possible [Erbayrak *et al.*, 2021]. Sun et al studied algorithms designed to instruct warehouse workers where to place items in a box, and observed when workers knowingly deviated from the algorithm's plan [Sun *et al.*, 2022]. The researchers then proposed modified algorithms to minimize human deviation from plans. While such approaches show promise in specific domains, they have limited expressiveness for required spatial relationships during object placement tasks. Therefore, in this work we present Positionally-Augmented Region Connection Calculus (PARCC), a spatial specification language able to capture humans' requirements during object placement tasks. Additionally, we introduce an inference algorithm to infer PARCC specifications from demonstrations.

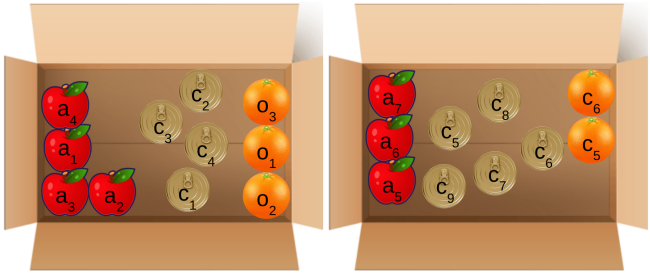

Figure 1: Two example configurations of apples, oranges, and cans in a box.

Broadly speaking, capturing humans' understanding of objects' spatial relationships in a scene is not a new topic to research. Paul et al introduced a framework to "ground" human instructions in a world representation (e.g., understanding the instruction "pick up the middle block in the row of 5 blocks") [Paul *et al.*, 2018], while others have developed methods to generate scenes or object arrangements from descriptions, either via predefined propositions (e.g., "A is left of B") [Wiebrock *et al.*, 2000] or through natural language [Vasardani *et al.*, 2013; Liu *et al.*, 2022].

While these systems could theoretically be used by workers to communicate specifications, there are a few underlying limitations. First, these methods only describe one scene at a time: for example, in Figure 1, prior methods may be able to express that the can c_2 is the furthest-north object in its scene, or generate an approximation of a scene via description of each object's placement. However, when considering a specification describing both scenes, these methods cannot communicate concepts such as "all cans are east of oranges" or "all oranges must touch another orange to the north or south." Capturing these relational rules can ensure, for example, that more fragile objects like apples or oranges are properly supported via contact or that there is consistent spatial placing of objects that may be expected by human workers. Additionally, methods capturing a human's understanding of a scene rely upon the human directly providing a description of that scene [Wiebrock *et al.*, 2000; Vasardani *et al.*, 2013; Liu *et al.*, 2022]. However, humans often under-specify (or misspecify) tasks, relative to what robots require, leading

to undesired behavior when following a human's directly-provided specification [Gross *et al.*, 2016].

To address such limitations, PARCC is designed to both capture descriptions over a "class" of related objects (e.g., apples, oranges, or cans) and use Boolean logic to encode specifications (e.g., "all oranges must touch another orange to the north or south"). Additionally, we present an inference framework to infer specifications from demonstrations instead of relying upon human-provided specifications. To demonstrate the effectiveness of our framework, we performed a human study to test how well the inference method captured specifications via demonstrations compared with direct human-provided specifications. In the field, this method can fit into a larger pipeline for automation of object placement tasks including packing and sorting that maintains patters of human behavior. Continuing the box packing example from Figure 1, a human can demonstrate multiple examples of packing apples, cans, and oranges into boxes and a PARCC specification of object relations can be inferred. In the future, a robot performing the packing task can plan and execute object placements satisfying the demonstrated specification.

2 Related Works

Within the field of task specification via formal logics, descriptions of spatial specifications often use Signal Temporal Logic (STL), since its operation over continuous signals makes it ideal for expressing spatial preferences [Maler and Nickovic, 2004]. Application of STL to spatial problems either uses standard STL operators to express positions and regions as signals [Linard and Tumova, 2020] or modifies notation to include spatial-specific operators [Nenzi *et al.*, 2015; Ma *et al.*, 2020]. For example, Nenzi et al added two spatial modalities ("somewhere" and "surrounds") to STL that operate over an undirected graph representing space (e.g., $\phi_1 \mathcal{S}_{[d_1,d_2]} \phi_2$ expresses that a region where ϕ_1 is true is surrounded by the region where ϕ_2 is true) [Nenzi *et al.*, 2015].

Other spatial specification languages use quad-tree representations [Haghighi *et al.*, 2015]. Here, space is recursively partitioned into quadrants over which a specification reasons. For example, a quad-tree may represent a city and the specification describes power grid requirements across the city.

While STL and quad-tree representations are highly expressive, they do not lend themselves to cleanly encoding human-intuitive spatial relationships between objects, which tend to use qualitative descriptors and small, countable values. Conversely, region connection calculus (RCC) is a spatial-relational language introduced by Randel et al to formalize human-intuitive concepts of spatial relationships between regions [Randell *et al.*, 1992]. The fundamental relation of RCC is $C(x, y)$ — read as 'x connects with y', meaning that the topological closure of regions x and y share at least one point. The two axioms for C are as follows:

$$\forall x[C(x, x)] \tag{1}$$

$$\forall x \forall y[C(x, y) \rightarrow C(y, x)] \tag{2}$$

The first axiom states that any region x must connect to itself; the second states that if y connects to x, x must connect to y. This fundamental "connect" relation can be used to describe many spatial relations, and several fragments have been proposed for various purposes; RCC8, for example, is a set of eight exhaustive and pairwise disjoint relationships within the RCC framework. However, in this paper we focus on the original 10 relations described in Technical Appendix A. Technical Appendix B further discusses differences in expressing qualitative object relations using PARCC, STL, and quad-trees; as an extension of RCC, PARCC expresses objects relations more easily than other languages.

While RCC itself is not based in logical specification, it has been adopted into specification languages. Ven et al introduced qualitative privacy description language (QPDL), using RCC within linear temporal logic (LTL) to describe technological privacy [van de Ven and Dylla, 2016]. Similarly, spatio-temporal synthesis logic (STSL) combines SU_4 (a spatial language similar to RCC) with STL to characterize spatio-temporal dynamic behaviors in applications such as adaptive cruise control [Li *et al.*, 2020]. However, neither QPDL nor STSL can express specification over "classes" of regions (i.e., "oranges are east of cans," as shown in Figure 1), nor has an algorithm been proposed to infer such specifications from demonstration.

With respect to specification inference via demonstrations, we take inspiration from Vazquez et al, who proposed an inference framework over LTL via a maximum entropy approach [Vazquez-Chanlatte *et al.*, 2018]. This framework derives a likelihood model over specifications based on how frequently demonstrations satisfy a specification versus the probability that the specification would be satisfied by random actions.

3 PARCC Formulation

Applying RCC directly to describe object relations provides some insight into human-intuitive relationships between objects. For example, in Figure 1, RCC can describe that a_1 is in contact with a_4 using the notation $EC(a_1, a_4)$; however, RCC fails to capture two aspects of the examples in Figure 1. First, RCC cannot communicate directionality: while a human may naturally notice that a_1 is west of c_1 or that o_3 contacts o_1 on the north side, RCC cannot capture these distinctions. Second, while RCC describes the relationship between individual objects (e.g., $EC(o_1, o_3)$), it fails to describe a pattern over a "class" of similar objects (e.g. apples are west cans).

In order to include these two capabilities into PARCC, we first define a subset of RCC relations useful for describing object relationships (as opposed to abstract regions), and augment this subset to capture directional information. We then use these object relations to describe patterns between all objects of a particular class using Boolean logic.

We constrain PARCC to exist over axis-aligned rectangular objects on a flat plane; this choice is motivated by the ubiquitous use of rectangular bounding boxes to represent the size and location of objects in a scene [Ali and Zhang, 2024; Zendehdel *et al.*, 2023; He *et al.*, 2021; Jia *et al.*, 2021]. In this paper, we define an object as a tuple, $o = (o_l, o_w, o_x, o_y, o_c)$, where o_l is the object's length, o_w is its height, o_x and o_y designate its position in space, and o_c is the object's "class" —

a label provided to objects subject to the same specifications (e.g., apples, oranges, and cans in Figure 1). In application, this could designate fragile vs. non-fragile objects, shipping destinations for packages, etc.

3.1 PARCC Relations

PARCC reasons over object relations via a subset of the canonical RCC relations (Technical Appendix A) — specifically with "discrete from" and "externally connected to" (written as $DR(x, y)$ and $EC(x, y)$, respectively). As described in Technical Appendix A, $DR(x, y)$ implies the interiors of x and y do not overlap, and $EC(x, y)$ implies the exterior boundaries x and y touch. We exclude the remaining RCC relations since they describe some overlap between regions (disallowed as our regions represent physical objects) or can be described with DR and EC themselves.

Definition 3.1 (PARCC object relation)**.** PARCC object relations include the basic DR and EC relations, with a subscript indicating the relative cardinal position of one object to another (assuming north is aligned with the positive y axis). For example, in Figure 1, we can say that $DR_N(c_2, c_1)$; formally, this requires the following:

$$DR_N(c_2, c_1) \rightarrow DR(c_2, c_1) \wedge y_{c_2} \geq y_{c_1} \quad \forall (x_{c_2}, y_{c_2}) \in c_2, (x_{c_1}, y_{c_1}) \in c_1 \tag{3}$$

meaning c_2 is discrete from c_1, and the y value of every point in c_2 is greater than the y value of every point in c_1.

In addition to the position-augmented object relations, our language must also reason over relations between classes. For this purpose, we define class relations as follows:

Definition 3.2 (PARCC class relation)**.** PARCC class relations use the same notation as position-augmented object relations, but operate over two classes. Given that $\mathcal{A}$ is the set of objects in class A and $\mathcal{B}$ is the set of objects in class B, our class DR relations require all objects in $\mathcal{A}$ to have the provided relation with all objects in $\mathcal{B}$. For example, a DR North class relation would be as follows:

$$DR_N(A, B) \leftrightarrow DR_N(a, b) \quad \forall a \in \mathcal{A} \quad \forall b \in \mathcal{B} \tag{4}$$

This means that, for all objects a of class A and all objects b of class B, $DR_N(a, b)$ must hold. Conversely, EC class relations would require that all objects in $\mathcal{A}$ have the given position augmented relation with at least one object in $\mathcal{B}$. For example, a EC North class relation would be as follows:

$$EC_N(A, B) \leftrightarrow EC_N(a, b) \quad \forall a \in \mathcal{A} \quad \exists b \in \mathcal{B} \tag{5}$$

This means that, for all objects a of class A, there exists an object b of class B such that $EC_N(a, b)$.

PARCC Formulas

Prior work has used RCC relations as propositions in logic languages [van de Ven and Dylla, 2016]; we extend this to Boolean logic over PARCC class relations.

Definition 3.3 (PARCC Formula)**.** A PARCC formula is a propositional logic formula over PARCC class relations. Conjunction and disjunction over PARCC class relations can be applied directly using the definitions of class relations in Eqs 4 and 5. For example:

$$DR_N(A, B) \vee DR_S(C, D) \leftrightarrow (DR_N(a, b) \forall a \in \mathcal{A} \forall b \in \mathcal{B}) \vee (DR_S(c, d) \forall c \in \mathcal{C} \forall d \in \mathcal{D}) \tag{6}$$

Negation of DR and EC PARCC class relations are defined as follows:

$$\neg DR_i(A, B) \leftrightarrow \nexists a \in \mathcal{A} \quad s.t.\ DR_i(a, b)\ \forall b \in \mathcal{B} \tag{7}$$

$$\neg EC_i(A, B) \leftrightarrow \nexists a \in \mathcal{A} \quad s.t.\ EC_i(a, b)\ \exists b \in \mathcal{B} \tag{8}$$

In order for $\neg DR_i(A, B)$ to hold, there cannot exist an object a of class A for which $DR_i(a, b)$ holds for all objects of class B. Conversely, for $\neg EC_i(A, B)$ to hold, there cannot exist an object a of class A for which $EC_i(a, b)$ holds for any objects of class B.

Demonstration

In this paper, we describe our inference algorithm over "demonstrations" of objects in a given space.

Definition 3.4 (Demonstration)**.** We define a demonstration as the tuple $D = (\mathcal{O}_D, \mathcal{S}_D, \mathcal{L}_D)$, where $\mathcal{O}_D$ is the set of objects in a particular demonstration, $\mathcal{S}_D$ is the space of x, y points available for object placement (i.e., $(o_x, o_y) \in \mathcal{S}_D \forall o \in \mathcal{O}_D$), and $\mathcal{L}_D$ is the set of classes to which objects in $\mathcal{O}_D$ belong. We notate $\mathcal{O}_D^L$ as the subset of objects belonging to class $L \in \mathcal{L}$.

Throughout this paper, we will use PARCC formulas to describe a demonstration, D. Specifically, we say D can satisfy a formula, ϕ, if all objects in $\mathcal{O}_D$ satisfy ϕ. For example, let D be the demonstration of the example on the left in Figure 1, A be the class of apples, and C be the class of cans. Statement $D \rightarrow DR_E(A, C)$ then evaluates to true, since every apple is discrete from and east of every can. However, the statement $D \rightarrow EC_E(A, A)$ evaluates to false, since not all apples are externally connected to another apple to the east.

4 Specification Inference

For our inference procedure, we assume access to k demonstrations from a human, notated $\mathcal{D} = \{D_1 ... D_k\}$. We assume each D_i has the same space $\mathcal{S}_D$ and classes $\mathcal{L}_D$; note, however, that demonstrations have different object sets, $\mathcal{O}_D$. We also assume each demonstration $D \in \mathcal{D}$ conforms to a specification Φ_h. Our goal is to infer a conjunctive normal-form (CNF) PARCC formula, Φ, that describes universal patterns in $D_1 .... D_k$ as intended by the demonstrator.

The inference process has two steps. First, we use a search-based method to determine a set of disjunctive PARCC formulas, $\bar{\mathcal{C}}$, such that for each $\phi \in \bar{\mathcal{C}}$, $D \rightarrow \phi$ for all $D \in \mathcal{D}$; these will form the disjunctive clauses of the CNF formula, Φ. Second, using a frequentist approach, we determine the probability that each formula $\phi \in \bar{\mathcal{C}}$ was intended by the demonstrator. We evaluate this probability using $\mathcal{R}$, a set of "non-specification" demonstrations generated without the human's specification, and calculate the probability that a demonstration would satisfy ϕ without intent.

Algorithm 1: Candidate Disjunctive Formulas

1 **Function** `FindDisjunctions(`$D_1 ... D_d$, $\mathcal{L}$, N`)`**:**
2 $\bar{\mathcal{C}} = \emptyset$;
3 **for** n = *1 ... N* **do**
4 **for** $\phi \in Template(\mathcal{L}, n)$ **do**
5 **if** $\exists \bar{\phi} \in \bar{\mathcal{C}}$ $s.t.$ $\bar{\phi} \rightarrow \phi$ **then**
6 continue ;
7 **if** $D \rightarrow \phi \; \forall \; D \in \{D_1 .... D_d\}$ **then**
8 $\bar{\mathcal{C}}.add(\phi)$;
9 **return** $\bar{\mathcal{C}}$

Algorithm 2: Inferring Intended Formulas

1 **Function** `Inference(`$\mathcal{D}$, $\bar{\mathcal{C}}$, p_c, k_r`)`**:**
2 $\mathcal{C} = \emptyset$
3 $\mathcal{R} = \{$ `SampleRandDemo(`$\mathcal{D}$`)` $|i \in k_r\}$
4 **for** $\phi \in \bar{\mathcal{C}}$ **do**
5 $p_\phi = \prod_{D \in \mathcal{D}} P(D \rightarrow \phi | \mathcal{R})$
6 **if** $p_\phi < p_c$ **then**
7 $\mathcal{C}.add(\phi)$
8 **return** $\bigwedge_{\phi \in \mathcal{C}} \phi$
9 **Function** `SampleRandDemo(`$\mathcal{D}$`)`**:**
10 $R = ChooseRandom(\mathcal{D})$.copy()
11 **for** $o \in \mathcal{O}_R$ **do**
12 $(o_x, o_y) = ChooseRandom(\mathcal{S}_D)$
13 **return** R

4.1 Finding Satisfying Disjunctive Formulas

Our inference procedure begins by finding a set of disjunctive PARCC formulas, $\bar{\mathcal{C}}$, which satisfy all demonstrations, $\mathcal{D}$ (Algorithm 1). We use an exhaustive search-based method to determine $\bar{\mathcal{C}}$. While one can brute-force a search over all possible disjunctive formulas, this would require checking $\mathcal{O}(|\mathcal{L}|^{2N} 4^N)$ formulas. Therefore, we take inspiration from Shah et al., and allow the use of production rule "templates" limiting the search space based on domain knowledge (see Section 5.2 for an example) [Shah *et al.*, 2018].

This procedure takes in demonstrations ($\mathcal{D}$) the set of object classes ($\mathcal{L}$) and the maximum length of the disjunctive formulas (N). First, the algorithm initializes the set of disjunctive formulas, $\bar{\mathcal{C}}$, as the empty set (line 2); we then loop over all possible lengths of the disjunctive phrase from 1 to N (line 3). For each length, we loop over every disjunctive formula of length n allowed by the production rule template given the classes, $\mathcal{L}$ (line 4). For each formula ϕ, we check whether a disjunction is already trivially implied by another formula in $\bar{\mathcal{C}}$ — and, if so, do not further consider it for $\bar{\mathcal{C}}$ (lines 5-6). Next, we check if ϕ is satisfied by all demonstrations in $\mathcal{D}$, and if so add it to $\bar{\mathcal{C}}$ (lines 7-8). Finally, we return the satisfying formulas, $\bar{\mathcal{C}}$ (line 9).

4.2 Determining Intended Disjunctive Formulas

Once our algorithm finds $\bar{\mathcal{C}}$, we determine the formulas $\phi \in \bar{\mathcal{C}}$ that likely describe the demonstrator's intent. To this end, we calculate the probability that any demonstration D will unintentionally satisfy ϕ, which we express as $P(D \rightarrow \phi | \mathcal{R})$, where $\mathcal{R}$ is a set of "non-specification" demonstrations generated without the human's intended specification, Φ_h.

Notice that $D \rightarrow \phi$ if and only if the relations every object o has with other objects in the demonstration satisfy ϕ (which we will notate as $o \rightarrow \phi$). Therefore, assuming the probability that each object satisfies ϕ is independent, the probability is as follows, where we use C to notate the class of objects relevant to ϕ:

$$P(D \rightarrow \phi | \mathcal{R}) = \prod_{o \in \mathcal{O}_D^C} P(o \rightarrow \phi | \mathcal{R}) \tag{9}$$

While not necessarily representative of reality, our independence assumption makes this probabilistic modeling tractable, and provides good results in our human study (see Section 5). In order to approximate $p(o \rightarrow \phi | \mathcal{R})$, we calculate the fraction of objects from $\mathcal{R}$ that satisfy formula ϕ:

$$P(o \rightarrow \phi | \mathcal{R}) = \max\left(\epsilon, \frac{\sum_{R \in \mathcal{R}} \sum_{o' \in \mathcal{O}_R^C} \mathbf{1}(o' \rightarrow \phi)}{\sum_{R \in \mathcal{R}} \sum_{o' \in \mathcal{O}_R^C} 1}\right) \tag{10}$$

where ϵ is a small number that prevents the probability from being 0 (we use .01), and $\mathbf{1}(o' \rightarrow \phi)$ is an indicator variable set to 1 if $o' \rightarrow \phi$, and 0 otherwise. We substitute this probability back into Equation 9, which results in the following:

$$P(D \rightarrow \phi | \mathcal{R}) = \prod_{o \in \mathcal{O}_D^C} \max\left(\epsilon, \frac{\sum_{R \in \mathcal{R}} \sum_{o' \in \mathcal{O}_R^C} \mathbf{1}(o' \rightarrow \phi)}{\sum_{R \in \mathcal{R}} \sum_{o' \in \mathcal{O}_R^C} 1}\right) \tag{11}$$

Algorithm 2 describes the construction of Φ. The algorithm takes a set of human demonstrations ($\mathcal{D}$), the number of non-specification demonstrations to generate (k_r), the set of disjunctive formulas found in Algorithm 1 ($\bar{\mathcal{C}}$), and a cutoff probability parameter (p_c). First, the algorithm constructs an empty set, $\mathcal{C}$ (line 2). Then the algorithm generates k_r non-specification demonstrations via the `SampleRandDemo` function (line 3). `SampleRandDemo` copies a demonstration from $\mathcal{D}$ (line 10) and reassigns each object a point in the demonstration space, $\mathcal{S}_D$ (line 12). Once every object's position is reassigned, the demonstration is returned (line 13). (While the pseudo-random object placement in `SampleRandDemo` may be distinct from human behavior without a defined Φ_h, experiments presented in Section 5 show the process described here provides a reasonable analog to human-provided non-specification demonstrations.)

The algorithm then loops over every disjunctive formula $\phi \in \bar{\mathcal{C}}$, calculating the probability that all human demonstrations unintentionally satisfied ϕ using Equation 11 (lines 4-5). Next, the algorithm checks whether this probability is under the cutoff probability p_c (i.e., whether we are confident that ϕ was not randomly satisfied, we use $p_c = .05$), and adds it to $\mathcal{C}$ (lines 6-7) if so. Finally, the algorithm returns the full specification as the conjunction of all formulas in $\mathcal{C}$ (line 8).

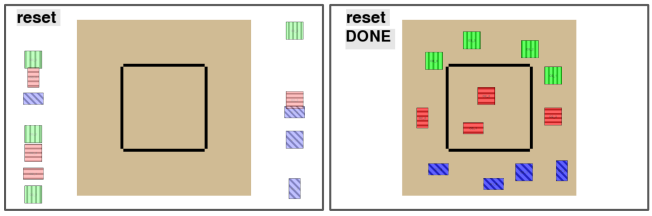

Figure 2: The demonstration interface we used in our experiment. The initial state (left) and completed state (right) is shown.

5 Experiment

We implemented the PARCC inference procedure and evaluated it with human subjects. This study tested the effectiveness of the PARCC specification language and inference framework to capture a human's intent in specifying a spatial configuration. Additionally, our experiment provides support for the hypothesis that demonstration-based specification systems can mitigate issues related to under- or misspecification arising from other modalities, such as natural language. We do not compare against other specification languages common in spatial domains (e.g. STL) as they are ill-equipped to represent object class relations (see Appendix B). [1]

5.1 Experimental Setup

For this study, we designed a box-packing environment as a representative task (Figure 2). The environment initializes with a brown square representing a table acting as the demonstration space, and four objects representing walls of an open box. Off the table were two to four objects each of classes R (red objects), G (green objects), and B (blue objects), which the subject could move. For visual distinction in figures, we shade red objects with horizontal lines, green objects with vertical lines, and blue objects with diagonal lines. Once the subject placed all objects on the table, a "done" button appeared, allowing the subject to complete the demonstration.

The experiment procedure began with a participant training phase, during which we asked the participant to provide five object placement demonstrations (training demonstrations $\mathcal{D}_T$) without having received any prompting with regard to how to arrange the objects on the table. The remainder of the process is shown in Figure 3, along with the notation for demonstrations and specifications generated throughout the experiment. After the training, we showed each participant a set of eight pre-generated demonstrations $\mathcal{D}_I$ (Figure 4). The set of pre-generated demonstrations, $\mathcal{D}_I$, were identical for all participants and were created to satisfy a conjunction of of 12 PARCC formulas described in Technical Appendix C. Upon showing participants $\mathcal{D}_I$, we requested three responses: a set of in-kind demonstrations, and two specifications.

First, the subject provided eight demonstrations, $\mathcal{D}_D$, attempting to follow all spatial patterns they observed in $\mathcal{D}_I$. (Note we had not yet introduced the PARCC language to participants, and they were free to consider "patterns" in whatever representation was most natural.) The participant then provided a natural-language explanation of spatial patterns in

[1] For code and datasets: https://github.com/AlexCuellar/PARCC

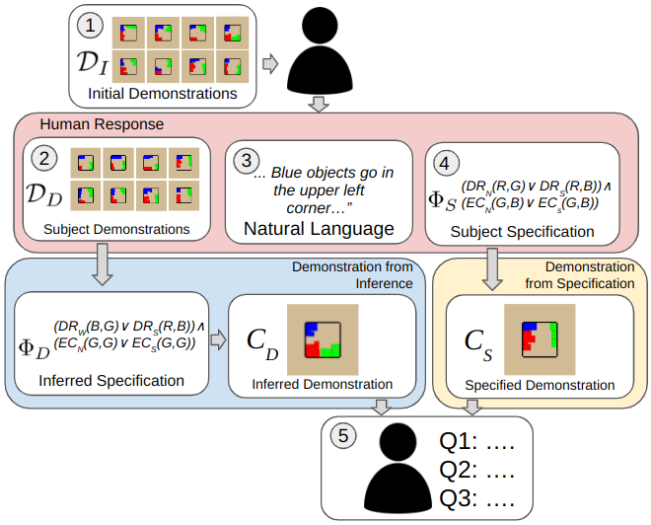

Figure 3: A pipeline showing our human study procedure. Steps directly involving human participation are numbered 1-5. Section 5.1 depicts the final questions given to the human.

$\mathcal{D}_D$ (again, the form of these "patterns" were determined entirely by the participant). Finally, we introduced participants to the PARCC language, and had them provide a PARCC specification Φ_S that best described their demonstrations, $\mathcal{D}_D$. Our algorithm then generated two demonstrations: C_D, which optimized satisfaction of the specification Φ_D inferred from $\mathcal{D}_D$; and C_S, which optimized the specification Φ_S provided directly by the user. These computer-generated demonstrations allowed us to probe subjects' response to system behavior when following specifications inferred from demonstrations (Φ_D) versus those explicitly specified (Φ_S). To generate C_D and C_S, we used a combination of mixed-integer programming and Monte Carlo tree search, both of which have been employed in box-packing domains [Erbayrak *et al.*, 2021; Edelkamp *et al.*, 2014]. Finally, we showed the participant both computer-generated demonstrations simultaneously, C_D on the left and C_S on the right, and asked the following Likert-style questions:

Q1: *The left image matches patterns in my demonstrations.*
Q2: *The right image matches patterns in my demonstrations.*
Q3: *Which computer generated demonstration do you think better matches patterns in your demonstrations?*

Questions were on a 1-5 scale. For Q1 and Q2, 1 indicated "strongly disagree" and 5 indicated "strongly agree;" for Q3, 1 indicated "strongly left" and 5 indicated "strongly right."

We performed this study in two groups for whom the inference procedure used different datasets for non-specification demonstrations (notated as $\mathcal{R}$ in Section 4). *Group A*'s specifications were inferred using the `SampleRandDemo` process (Algorithm 2); *Group B*'s inference used the demonstrations $\mathcal{D}_T$ provided by subjects in Group A during environment training as non-specification demonstrations $\mathcal{R}$. For each group, inference used 100 non-specification demonstrations. Via this distinction, we can identify whether using random object placements as non-specification demonstrations (see Algorithm 2) provides a good enough analog for human

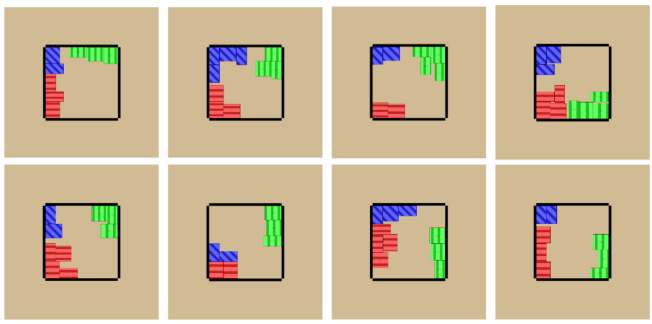

Figure 4: Pre-generated demonstrations of the box packing environment initially shown to study subjects ($\mathcal{D}_I$).

demonstrations without prompting for a specification.

We recruited 35 subjects, 20 in Group A and 15 in Group B. All were aged between 18 and 34 years and had at least a high-school education. The protocol was approved by the MIT Committee on the Use of Humans as Experimental Subjects (protocol E-3748) and the United States Department of Defense Human Research Protection Office (protocol MITL20220002). All subjects provided informed consent before the experiment, and received $15 for their participation.

5.2 Choice of Search Template

For inference, we allow use of a "template" limiting the possible disjunctive formulas based on domain knowledge (Section 4.1). For this experiment, we intuited that class relations within disjunctive formulas would (1) share the same "related" class (i.e., the first class in a PARCC class relation) and (2) always contain either EC or DC class relations, but not both within the same disjunction. This intuition comes from considering each disjunctive phrase as its own "constraint," all of which must be satisfied by the overall conjunction. In this mindset, we found it likely that each "constraint" would reason over one class and only consider one RCC relation.

Similar to Dwyer's identification of LTL templates applicable to many real-world tasks, this template is intended to limit the search space to the most natural "constraints" for human operators across domains [Dwyer *et al.*, 1999]. While a full investigation of template choice can constitute its own work, we compare our choice of template to two others in Technical Appendix D. One template is less restrictive than the one we use (i.e. allowing a larger set of possible disjunctive formulas) and one is more restrictive (i.e. allowing a smaller set of disjunctive formulas). The specifications inferred from human demonstrations (Φ_D) are identical between our choice of template and the less restrictive template, showing that our template imposes minimal inductive bias on the specifications inferred in this experiment, and provides evidence that our template represents rules relevant to human demonstrators more generally. Additionally, we compare our template to a more restrictive template to show how operators can intentionally eliminate rules irrelevant to a particular domain.

5.3 Hypotheses

To evaluate the inference framework's ability to capture humans' specification, we proposed four hypotheses.

First, we expected that the inferred specification, Φ_D, would capture a human's intended specification, and therefore hypothesized that participants would respond to (Q1) by asserting that C_D matched their demonstrations:
H1: *Participants agree C_D matches patterns in $\mathcal{D}_D$. (Q1)*
Next, we expected that participants' provided specification, Φ_S, would not capture their intended specification. Therefore, we hypothesized that participants would respond to (Q2) asserting that C_S did not match their demonstrations:
H2: *Participants disagree C_S matches patterns in $\mathcal{D}_D$. (Q2)*
In our inference procedure (Algorithm 2), we automatically generate non-specification data $\mathcal{R}$; we expected this process to provide a reasonable analog to human demonstrations without prompting a specification. Therefore, we hypothesized that Groups A and B would respond similarly to (Q1):
H3: *Response to Q1 does not significantly vary between groups A and B.*
Finally, we expected direct specifications to differ from inferred specifications due to under- or misspecification.
H4: *Inferred specifications Φ_D are distinct from human-provided specifications Φ_S.*

5.4 Results

Figure 5 and Table 1 summarize our major results. To characterize how successfully the inference procedure captured participants' intended specification across both groups, we determined via a Wilcoxon one-sample signed-rank test that responses to Q1 significantly exceeded 3 (i.e., that participants responded either "agree" or "strongly agree") ($p = 5.1\text{e}{-8}$), implying that the inference algorithm captured the humans' intended specifications, and supporting $H1$.

Similarly, to characterize how successfully participants' provided specification Φ_S captured their own intended specification across both groups, a Wilcoxon one-sample signed-rank test also revealed that responses to Q2 were significantly less than 3 (i.e., that participants responded "disagree" or "strongly disagree") ($p = 3.8\text{e}{-7}$), implying that humans' provided specifications differed from their intended specifications, and supporting $H2$.

To determine whether the generation of non-specification data in Algorithm 2 is an appropriate analog for human demonstrations without prompting for a specification, we computed whether the responses to Q1 were significantly different between groups A and B via a Mann-Whitney U test. The results indicated no significant difference between the groups' responses ($p = 0.87$), supporting $H3$.

To determine any differences between the human's provided specification (Φ_S) and the inferred specification (Φ_D), we compared these two specifications to the specification employed when creating the pre-generated "initial" demonstrations $\mathcal{D}_I$ (Figure 4). $\mathcal{D}_I$ used a specification of 12 PARCC formulas ($\phi_1 ... \phi_{12}$) in conjunction with each other. These 12 formulas are described in Technical Appendix C. Table 1 shows the proportion of subjects for whom each of the 12 formulas appeared in Φ_D, and the proportion of subjects who successfully encoded each formula in their specification Φ_S. Across the 12 formulas, we used a two-sample Wilcoxon signed-rank test to characterize significant difference in the proportion of formulas specified by Φ_D and Φ_S, and found

Inclusion of Initial Specification in Response Mode												
	ϕ_1	ϕ_2	ϕ_3	ϕ_4	ϕ_5	ϕ_6	ϕ_7	ϕ_8	ϕ_9	ϕ_{10}	ϕ_{11}	ϕ_{12}
Φ_D (Human Demonstrations)	1	1	1	1	1	1	.88	.91	.91	.77	.74	.88
Φ_S (Human Specification)	0.86	0.83	0.8	0.2	0.2	0.2	0.06	0.06	0.09	0.03	0.00	0.03
Natural Language	0.94	0.94	0.94	0.83	0.83	0.83	.63	0.63	0.63	0.03	0.03	0.03

Table 1: Proportion of subjects including formulas $\phi_1 ... \phi_{12}$ from $\mathcal{D}_I$ across response types (see Technical Appendix C for the formulas). Φ_D was inferred from subject demonstrations $\mathcal{D}_D$. Φ_S is the human-provided PARCC specification. "Natural Language" refers to the human's written specification. The proportion of subjects including each formula in their demonstration Φ_D is higher than either direct specification Φ_S or natural language. This supports hypothesis H4, and shows that inferring human specifications from demonstrations is more reliable than direct specification or language.

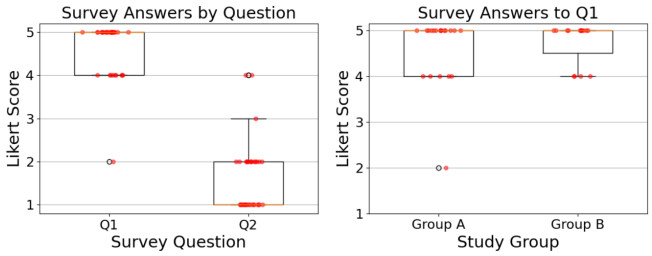

Figure 5: Box and whisker plots of Likert responses. (left) Likert responses to how well C_D matched patterns in subjects' demonstrations (Q1), and how well C_S matched patterns in subjects' demonstrations (Q2). Responses to Q1 were significantly greater than 3 ($p = 5.1e - 8$), and responses to Q2 were significantly less than 3 ($p = 3.8e - 7$). (right) Likert responses indicating how well C_D matched patterns in subjects' demonstrations between Groups A and B. Responses did not differ significantly between the two groups.

that significantly fewer formulas were correctly specified by Φ_S compared with Φ_D ($p = 4.8\text{e}{-4}$), supporting $H4$.

5.5 Discussion

The statistical support for $H1$ and $H2$ confirms that obtaining human PARCC specification via demonstrations provides an advantage over direct specification. Additionally, support for $H4$ suggests the perceived difference between C_D and C_S results from human misspecification, even when specification is performed via a formal language with inherent semantics translatable into natural language. This result is not surprising, given prior research demonstrates that humans often have difficulty interpreting and providing specifications in formal specification languages, with or without translation to natural language [Loomes and Vinter, 1997; Vinter *et al.*, 1996; Vinter, 1998; Greenman *et al.*, 2023].

Our findings also suggest that human specification via natural language has shortfalls. Table 1 shows that subjects often underspecified when using natural language (though determining one-to-one correlations between natural language and PARCC specifications is subjective). Additionally, subjects' natural language often contain inconsistencies with regard to word choice; for example, some subjects described objects of a class "clustering" around a corner of the box. However, some such participants always placed objects in contact with the walls comprising the corner, and some placed objects in the corner's vicinity. Such discrepancy shows natural language is an imprecise way to capture humans' intended specifications — and the underspecification of a human's own intention indicates that there would be shortcomings to this approach even if the language were precise.

Finally, support for $H3$ suggests that non-specification demonstrations for inference do not significantly vary between pseudo-randomly generated and human-generated data. Therefore, we conclude that pseudo-random object placement for non-specification demonstrations $\mathcal{R}$ (as in Algorithm 2) provides a reasonable analog for human-generated non-specification demonstrations.

6 Limitations and Future Work

While PARCC and the inference method in this paper captures spatial specifications relevant to human demonstrators, there are several open areas for future work. First, PARCC is limited to rectangular objects in two dimensions. Extending to three dimensions and more varied geometries can improve applicability in situations where objects cannot be treated as bounding boxes in an image. However, considering a third dimension requires modeling object stacking and stability, which is outside this paper's scope. Second, the inference procedure relies on search over all possible disjunctive formulas allowed by a template. While this works for a few object classes, it may be intractable for a larger number of classes due to combinatorial explosion. In these cases, a sample based approach similar to Shah et al's use of Markov Chain Monte Carlo may be useful [Shah *et al.*, 2018]. Finally, inference over PARCC only considers rules that exist over all demonstrations. In many situations, learning preferences in addition to strict rules may provide a route to capture more nuanced aspects of human demonstrations.

7 Conclusion

In this work, we presented Positionally Augmented RCC, a specification language expressing spatial relationships between classes of objects. By utilizing RCC as the basis of our language, our method expresses human-intuitive spatial relationships between objects more easily than traditional spatial languages (e.g. STL). We also present an inference framework to learn PARCC specifications from demonstrations. Finally, via a human study, we show our framework's effectiveness in capturing human-intended spatial specifications and the advantage of learning-from-demonstration approaches to specification over direct human specification due to humans' tendency to mis- or under-specify.

Acknowledgments

DISTRIBUTION STATEMENT A. Approved for public re589 lease. Distribution is unlimited. This material is based upon work supported by the Under Secretary of Defense for Research and Engineering under Air Force Contract No. FA8702-15-D-0001. Any opinions, findings, conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Under Secretary of Defense for Research and Engineering.

References

[Ali and Zhang, 2024] Momina Liaqat Ali and Zhou Zhang. The yolo framework: A comprehensive review of evolution, applications, and benchmarks in object detection. *Computers*, 13(12):336, 2024.

[Dwyer *et al.*, 1999] Matthew B Dwyer, George S Avrunin, and James C Corbett. Patterns in property specifications for finite-state verification. In *Proceedings of the 21st international conference on Software engineering*, pages 411–420, 1999.

[Edelkamp *et al.*, 2014] Stefan Edelkamp, Max Gath, and Moritz Rohde. Monte-carlo tree search for 3d packing with object orientation. In *Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz)*, pages 285–296. Springer, 2014.

[Erbayrak *et al.*, 2021] Seda Erbayrak, Vildan Özkır, and U Mahir Yıldırım. Multi-objective 3d bin packing problem with load balance and product family concerns. *Computers & Industrial Engineering*, 159:107518, 2021.

[Greenman *et al.*, 2023] Ben Greenman, Sam Saarinen, Tim Nelson, and Shriram Krishnamurthi. Little tricky logic: Misconceptions in the understanding of LTL. *The Art, Science, and Engineering of Programming*, 7, 2023.

[Gross *et al.*, 2016] Stephanie Gross, Brigitte Krenn, and Matthias Scheutz. Multi-modal referring expressions in human-human task descriptions and their implications for human-robot interaction. *Interaction Studies*, 17(2):180–210, 2016.

[Haghighi *et al.*, 2015] Iman Haghighi, Austin Jones, Zhaodan Kong, Ezio Bartocci, Radu Gros, and Calin Belta. Spatel: a novel spatial-temporal logic and its applications to networked systems. In *Proceedings of the 18th International Conference on Hybrid Systems: Computation and Control*, pages 189–198, 2015.

[He *et al.*, 2021] Yuhang He, Wentao Yu, Jie Han, Xing Wei, Xiaopeng Hong, and Yihong Gong. Know your surroundings: Panoramic multi-object tracking by multimodality collaboration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2969–2980, 2021.

[Jia *et al.*, 2021] Dan Jia, Mats Steinweg, Alexander Hermans, and Bastian Leibe. Self-supervised person detection in 2d range data using a calibrated camera. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 13301–13307. IEEE, 2021.

[Li *et al.*, 2020] Tengfei Li, Jing Liu, JieXiang Kang, Haiying Sun, Wei Yin, Xiaohong Chen, and Hui Wang. Stsl: A novel spatio-temporal specification language for cyber-physical systems. In *2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)*, pages 309–319. IEEE, 2020.

[Linard and Tumova, 2020] Alexis Linard and Jana Tumova. Active learning of signal temporal logic specifications. In *2020 IEEE 16th International Conference on Automation Science and Engineering (CASE)*, pages 779–785. IEEE, 2020.

[Liu *et al.*, 2022] Weiyu Liu, Chris Paxton, Tucker Hermans, and Dieter Fox. Structformer: Learning spatial structure for language-guided semantic rearrangement of novel objects. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 6322–6329. IEEE, 2022.

[Loomes and Vinter, 1997] Martin Loomes and Rick Vinter. Formal methods: No cure for faulty reasoning. In *Safer Systems*, pages 67–78. Springer, 1997.

[Ma *et al.*, 2020] Meiyi Ma, Ezio Bartocci, Eli Lifland, John Stankovic, and Lu Feng. Sastl: Spatial aggregation signal temporal logic for runtime monitoring in smart cities. In *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, pages 51–62. IEEE, 2020.

[Maler and Nickovic, 2004] Oded Maler and Dejan Nickovic. Monitoring temporal properties of continuous signals. In *Formal Techniques, Modelling and Analysis of Timed and Fault-Tolerant Systems*, pages 152–166. Springer, 2004.

[Nenzi *et al.*, 2015] Laura Nenzi, Luca Bortolussi, Vincenzo Ciancia, Michele Loreti, and Mieke Massink. Qualitative and quantitative monitoring of spatio-temporal properties. In *Runtime Verification: 6th International Conference, RV 2015, Vienna, Austria, September 22-25, 2015. Proceedings*, pages 21–37. Springer, 2015.

[Paul *et al.*, 2018] Rohan Paul, Jacob Arkin, Derya Aksaray, Nicholas Roy, and Thomas M Howard. Efficient grounding of abstract spatial concepts for natural language interaction with robot platforms. *The International Journal of Robotics Research*, 37(10):1269–1299, 2018.

[Randell *et al.*, 1992] David A Randell, Zhan Cui, and Anthony G Cohn. A spatial logic based on regions and connection. *KR*, 92:165–176, 1992.

[Sanneman *et al.*, 2020] Lindsay Sanneman, Christopher Fourie, and Julie A Shah. The state of industrial robotics: Emerging technologies, challenges, and key research directions. *arXiv preprint arXiv:2010.14537*, 2020.

[Shah *et al.*, 2018] Ankit Shah, Pritish Kamath, Julie A Shah, and Shen Li. Bayesian inference of temporal task specifications from demonstrations. *Advances in Neural Information Processing Systems*, 31, 2018.

[Sun *et al.*, 2022] Jiankun Sun, Dennis J Zhang, Haoyuan Hu, and Jan A Van Mieghem. Predicting human discretion to adjust algorithmic prescription: A large-scale field experiment in warehouse operations. *Management Science*, 68(2):846–865, 2022.

[van de Ven and Dylla, 2016] Jasper van de Ven and Frank Dylla. Qualitative privacy description language: Integrating privacy concepts, languages, and technologies. In *Privacy Technologies and Policy: 4th Annual Privacy Forum, APF 2016, Frankfurt/Main, Germany, September 7-8, 2016, Proceedings 4*, pages 171–189. Springer, 2016.

[Vasardani *et al.*, 2013] Maria Vasardani, Sabine Timpf, Stephan Winter, and Martin Tomko. From descriptions to depictions: A conceptual framework. In *International Conference on Spatial Information Theory*, pages 299–319. Springer, 2013.

[Vazquez-Chanlatte *et al.*, 2018] Marcell Vazquez-Chanlatte, Susmit Jha, Ashish Tiwari, Mark K Ho, and Sanjit Seshia. Learning task specifications from demonstrations. *Advances in neural information processing systems*, 31, 2018.

[Vinter *et al.*, 1996] RJ Vinter, MJ Loomes, and D Kornbrot. Seven lesser known myths of formal methods: uncovering the psychology of formal specification. Technical report, University of Hertfordshire, 1996.

[Vinter, 1998] RJ Vinter. *Evaluating formal specifications: a cognitive approach*. PhD thesis, University of Hertfordshire, 1998.

[Wiebrock *et al.*, 2000] Sylvia Wiebrock, Lars Wittenburg, Ute Schmid, and Fritz Wysotzki. Inference and visualization of spatial relations. In *Spatial Cognition II*, pages 212–224. Springer, 2000.

[Zendehdel *et al.*, 2023] Niloofar Zendehdel, Haodong Chen, and Ming C Leu. Real-time tool detection in smart manufacturing using you-only-look-once (yolo) v5. *Manufacturing Letters*, 35:1052–1059, 2023.