# Automatically Evaluating and Generating Clear Robot Explanations

Shen Li

CMU-RI-TR-17-09

May 2017

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

**Thesis Committee:**
Siddhartha Srinivasa (co-chair)
Stephanie Rosenthal (co-chair)
Reid Simmons
Stefanos Nikolaidis

*Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.*

*For my parents*

# *Abstract*

As robots act in the environment, people observe their behaviors and form beliefs about their underlying intentions and preferences. Although people's beliefs often affect their interactions with robots, today's robot behaviors are rarely optimized for ease of human understanding. In this thesis, we contribute studies and algorithms to improve the transparency of robot behaviors for human observers through giving natural language-based and demonstration-based explanations. Our first studies aim to understand how people use natural language to clearly explain their goals of picking up specified blocks in a tabletop manipulation task. We find that the clearest explanations lead people through the visual search task by identifying highly salient visual features, spatial relations with explicit perspective-taking words with respect to the blocks on the table. Based on our findings, we contribute state-of-art graph-based algorithms to automatically generate clear natural language explanations similar to those found in our study, and optimize those algorithms to demonstrate that they are scalable to realistic robot manipulation tasks. In our second studies, we aim to understand features of robot demonstrations that allow people to correctly interpret and generalize robot state preferences in grid world navigation tasks. We identify critical points along a demonstrated trajectory that convey information about robot state preferences - inflection points and compromise points, and contribute an approach for automatically generating trajectory demonstrations with specified numbers of critical points. We show that demonstrated trajectories with more inflection points and fewer compromise points allow observers to more clearly understand and generalize robot preferences compared to other combinations of critical points. We conclude the thesis with areas of future work that can further improve people's understanding of robot behavior.

# *Acknowledgments*

First and foremost, I would like to express my sincere gratitude to my advisors Dr. Siddhartha Srinivasa and Dr. Stephanie Rosenthal, who have greatly helped me in motivating me to keep proactive and hard-working, shaping my interests in human-robot interaction, inspiring, supporting, and formalizing my research ideas, and directing me through social networking and career path planning. I am also extremely grateful to my committee member Dr. Reid Simmons for his ideas and feedbacks all along my Masters research, and to my colleagues Dr. Henny Admoni and Dr. Daqing Yi for their invaluable advices condensed from their solid technical skills, rich life experiences, and kindest hearts. Their vision, guidance and knowledge have been invaluable in my work and life. It has been a great honor to work with them. I would like to thank Rosario Scalise as my "wonder twin" collaborator throughout my thesis work. I enjoy bouncing my ideas with him who always delivers me his innovative and comprehensive explanations, in both natural language and demonstration on computer, optimized for the maximal clarity and fun. I also thank him for encouraging me to network with people in conferences as a sincere friend. Further, I would like to thank all my fellow lab-mates as well, Stefanos Nikolaidis, Clint Liddick, Shushman Choudhury, Rachel Holladay, Gilwoo Lee, Laura Herlant, Jennifer King, Jimmy Jin, Michael Koval, and Shervin Javdani for many inspiring discussions and encouragements in my study and research, as well as balancing my life with lots of fun. It has been a great honor to be part of this hard-working, sociable, versatile, talented, and interesting team. Last but not least, I would express a deep sense of gratitude to my beloved parents and grandparents for their endless support and love throughout my life.

# *Contents*

# *List of Figures*

# *List of Tables*

# 1
# *Introduction*

Robots are deployed to collaborate with humans in many real-world tasks, such as furniture assembly [Knepper et al., 2013], warehouse automation [Everett et al., 1995], or meal serving [Ishii et al., 1995] through various interaction media such as joystick-based teleoperation [Nikolaidis et al., 2017b], gesture-based teleoperation [Nagi et al., 2015], and shared autonomy [Javdani et al., 2015].

There are various factors a robot has to consider in making plans, including objectives, preference, and constraints. For example, in a park navigation scenario (the legend is available in Fig. 1.4), a mobile robot is driving from the start towards the goal position. The robot might prefer navigating on dirt road (Fig. 1.1), on grass (Fig. 1.2), or on rock (Fig. 1.3) [Li et al., 2017]. None of these trajectories are straight lines as humans expect the robot to follow in order to maximize the efficiency based on rationality principle [Gergely et al., 1995, Dennett, 1989, Kamewari et al., 2005]. Humans might attribute biased causes and make errors [Tetlock, 1985] in understanding robot behaviors, which might raises problems like automation surprise [Sarter and Woods, 1997] when a robot takes an unexpected action due to the misalignments between robot plan and human understanding of the robot plan [Fisac et al., 2016].

One approach to resolve this issue is to develop transparency in robot behaviors by enabling a robot to provide comprehensible explanations [Seegebarth et al., 2012, Zhou et al., 2017]. There are multiple media in which explanations could be transferred from robots to humans.

Language-based explanation requires no extra tools and leaves physical bodies free for other tasks, reaches multiple agents in various positions, compensates for visual communication in dark and smoke conditions, is easy to be monitored and recorded [Simon, 1980], effectively influences human visual perception in multi-modal communication [Landau et al., 2010], and affectively responds to human emotions [Scheutz et al., 2006].

There are a lot of works on enabling robot to explain themselves

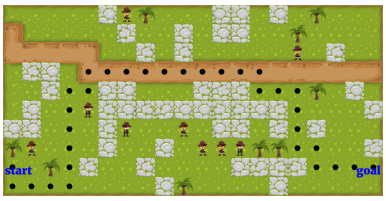



Figure 1.1: An autonomous car which prefers navigating on dirt roads is following the trajectory indicated by black dots [Li et al., 2017].


Figure 1.2: An autonomous car which prefers navigating on grass is following the trajectory indicated by black dots [Li et al., 2017].

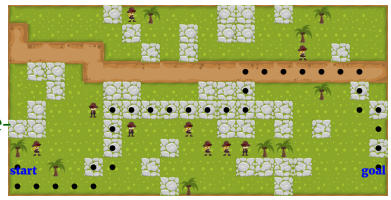



Figure 1.3: An autonomous car which prefers navigating on rock is following the trajectory indicated by black dots [Li et al., 2017].


Figure 1.4: The legend for Fig. 1.1, Fig. 1.2, Fig. 1.3 [Li et al., 2017].

in natural language. Langley [2016] develop an agent to explain its decision-making based on the mental and physical situations in the past. Chakraborti et al. [2017] use multi-model explanations in a prolonged interaction to reconcile the gap between robot and human world models with a minimal change to the human world model. In Human-computer Interaction, a line of research develops *intelligibility* [1] in *context-aware systems* [2] by making the system able to explain its reasonings to users to develop human objective understanding, trust and reliance [Lim et al., 2009, Kulesza et al., 2012, 2013, Bussone et al., 2015] and maintain user satisfaction and usability [Dey, 2009]. Rosenthal et al. [2016] convert visualization to verbalization in natural language to describe mobile robot navigation experiences. Perera et al. [2016] refine and predict the variability of robot explanations to different humans through continued dialog.

[1] Intelligibility: the ability of an application to explain its own behaviors.

[2] Context-aware system: the system which adapts according to the location of use, the collection of nearby people, hosts, and accessible devices, as well as to changes to such things over time. It examines the computing environment and reacts to changes to the environment [Schilit et al., 1994].

These works above mainly focus on explaining the robot decision-making processes or objective functions, but not explaining the robot intentions or constraints. In this work, we contribute a corpus collected from people, evaluate the explanations collected about robot intentions in a tabletop manipulation scenario, and improve the state-of-art algorithm to enable robots to automatically generate explanations about their intentions in natural language.

Inspired from human visual and verbal learnings [Standing, 1973], in addition to language-based explanations, demonstration-based explanations or expressive behaviors could be more effective than language-based in tasks, such as tying shoes.

People have focused on using robot motion as a demonstration to expressively convey robot capabilities [Nikolaidis et al., 2017a], robot learning progresses [Nicolescu and Mataric, 2003], object physical properties [Sciutti et al., 2014, Zhou et al., 2017], and robot goals [Gielniak et al., 2013, Dragan and Srinivasa, 2014, Zhang et al., 2016, Szafir et al., 2014, Takayama et al., 2011]. Zhang et al. [2015] introduce plan explicability as the ease with which humans associate tasks with robot actions in the plan and predictability as the ease with which humans to predict the next task given actions in the previous tasks. Kulkarni et al. [2016] model explicability as the distances between robot plans and the human approximation of robot plan.

These works above mainly focus on expressing the static properties of the robot and related objects, $i.e.$ the system state, but not explain what the robot is optimizing for, $e.g.$ the cost function in MDP. In this work, we propose an algorithm to automatically generate demonstration-based explanations about robot preference in a gridworld navigation scenario and evaluate our generated demonstrations in a user study.

In the following, I will first talk about a set of user studies we ran to evaluate the clarity of a language-based explanation in a tabletop robot

manipulation task. Second, I will contribute several techniques to speed up the state-of-art algorithm for automatically generating clear explanations in natural language. Third, I will propose an algorithm to automatically generate clear demonstration-based explanations in a grid-world navigation task. In the end, I will use a user study to evaluate the clarity of our generated demonstration-based explanations.

# 2

# *Evaluating Explanations in Natural Language*

[1]

In a tabletop manipulation task, it is critical for a robot to convey its intention [Tomasello et al., 2005], *i.e.* the block it is about to pick up, in Fig. 2.1, so that humans could read [Takayama et al., 2011], understand [Alami et al., 2006b] and anticipate robot motion [Gielniak and Thomaz, 2011], infer robot intention [Lichtenthäler et al., 2011], avoid conflicts and effectively collaborate with robots.

Robot intention in this tabletop manipulation *scene* would be the *target object* [2]. One way for a robot to communicate its intention with humans is through a *referring expression (RE)* [3] which is composed by a set of *features*. A clear RE is able to refer to the target object and distinguish it from *distractors* [4] in the scene.

There is a long line of research in robotics related to communicating REs, such as "furthest to the right","near the back", and "closest", for navigation tasks [Skubic et al., 2004, Blisard and Skubic, 2005, MacMahon et al., 2006, Kollar et al., 2010, Tellex et al., 2011, Howard et al., 2014, Trafton et al., 2005a]. However, there are fewer studies on the communication of *spatial references* [5] for tabletop or assembly tasks [Bisk et al., 2016].

In simple scenes, people tend to use *visual features*, such as type and color in their REs for several reasons. Visual features are usually the *conceptual gestalt* [6] [Deemter et al., 2012] , have a higher *perceptual saliency* [7] [Clarke et al., 2013] and *codability* [8] [Deemter et al., 2012] , and require less cognitive efforts from humans [Deemter et al., 2012], For example, in a robot tabletop manipulation scene as shown in Fig. 2.2, "the yellow block" is a RE to identify the target block indicated by the red arrow, where "yellow" is a visual feature - color and "block" is a visual feature - type.

However, in complex scenes, a RE with just visual features might fail to distinguish an object from the others. In another robot tabletop manipulation scene as shown in Fig. 2.3, it is not possible to identify the target object indicated by the red arrow by just using color and type,

[1] This work is done in collaboration with Rosario Scalise

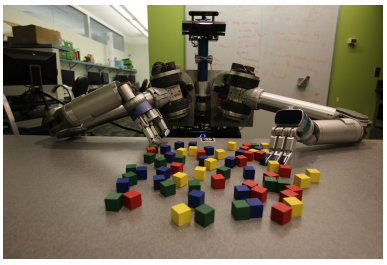


Figure 2.1: Herb (Home Exploring Robotic Butler) [Srinivasa et al., 2010] is picking up blocks from the tabletop.

[2] Target object: the particular object actively involved in the manipulation tasks.

[3] Referring expression (RE): a noun phrase or surrogate for a noun phrase, whose function in discourse is to identify some individual objects [Wikipedia, 2016b]

[4] Distractors: the objects that are not the target object but stay in the same scene with the target object, which might confound the identification of the target object.

[5] Spatial reference: referring to an object using spatial relations.

[6] Conceptual gestalt: the central features forming the speaker's mental representation of the referent, *e.g.* color [Deemter et al., 2012].

[7] Perceptual saliency: the ease with which that the pixels stand out from their surroundings in a scene [Clarke et al., 2013].

[8] Codability: the ease with which that attribute can be included in a mental representation of an object [Deemter et al., 2012].

because there are multiple yellow blocks in the scene. Here "the yellow block" is an *ambiguous* RE because it could identify not only the target object, but also distractors of the target object in the scene.

Note that most visual features are object properties, which are *unary features*. To identify target objects in complex scenes, we can utilize *spatial relations*, either *binary* or *n-ary*, to establish a grounding or certainty about the semantic relationship between two objects. For example, in the complex scene Fig. 2.3, an unambiguous RE for the block indicated by the red arrow would be "the left block among the two yellow blocks which are in between two blue blocks". The spatial relations, such as "left" and "in between two blue blocks" help *disambiguate* the target yellow block and distinguish it from the other blocks. Note that "left" is a binary spatial relation indicating that the target block is on the left to a *landmark* [9] block. Meanwhile, "in between two blue blocks" is a 3-ary spatial relation between the target object and "two blue blocks" as two landmarks. Therefore, a RE typically distinguishes target objects from distractors through unary features, *e.g.* visual features, and binary or n-ary features, *e.g.* spatial relations with references to landmarks.

However, even with the use of visual features and spatial relations, it is still possible to encounter additional ambiguity which originates from the reference frames. Humans often specify their *perspectives* [10] to resolve this ambiguity as in the example "the red cup on your right", in which "your right" specifies the perspective. Robots that collaborate with humans in tabletop tasks have to both understand and generate visual features, spatial relations, and perspectives. We investigate these key components by collecting a corpus of REs and analyzing them with clarity.

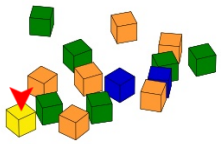


Figure 2.2: A simple scene for tabletop manipulation tasks

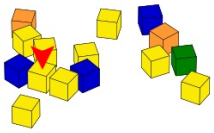


Figure 2.3: A difficult scene for tabletop manipulation tasks

[9] Landmark: people refer to non-target objects which assist disambiguating the target object from distractors.

[10] Perspective: the fixed point where the speaker is issuing a RE.

## 2.1 Related Work

### Visual Feature

*Visual Search* Our task is to generate natural language used for identifying a particular object in the cluttered environment, which is a inverse process to *visual search*, the task of finding the particular objects based on information. [11]. Wolfe [1994] divides visual search into 2 steps: (1) Processing easy information from all locations in parallel; (2) Focusing on the complex information from a few spatial locations. In the first step, people respond to different visual stimuli from the scene, *e.g.* color, stereoscopic depth, line arrangement, curvature, intersection, and terminator, in different response time determined by their *visual salience* [12] [Wolfe, 1994].

[11] Visual search: a human routine visual behavior to find one object in a visual world filled with other distracting items [Wolfe, 1994].

[12] Visual salience: the ease with which that a pixel or a region within a scene stands out from their surroundings [Clarke et al., 2013].

*Visual Salience* People react faster to visual features with high visual salience when searching for an object in a scene with *visual clutter* [13] [Clarke et al., 2013]. Wolfe [1994] survey many visual features studied in literatures and rank them in a sequence based on their visual salience. The partial ranking is object orientation, color, motion, size, stereoscopic depth [14], binocular lustre [15], vernier offset [16], curvature, terminators [17], and intersections.

[13] Visual clutter or feature congestion: the variability of features, *e.g.* color, orientation, and luminance, in a local neighborhood [Clarke et al., 2013].

[14] Stereoscopic depth: a visual feature indicating the distance from the viewer, *e.g.* this object is closest or farthest to me [Wolfe, 1994].

[15] Binocular lustre: If a spot is darker than the background in the image presented to one eye and brighter in the other eye, the perception alternates between darker and lighter which presented to each eye [Wolfe, 1994, Wikipedia, 2016a].

[16] Vernier offset: the disalignment among two line segments [Wolfe, 1994].

[17] Terminators: *e.g.* the tip of a line segment [Wolfe, 1994].

### *Perspective*

When people collaborate together on spatial tasks, they often must take each other's perspectives when referring to objects in the environment [Franklin et al., 1992, Taylor and Tversky, 1996]. In an analysis of 4000 utterances made by NASA astronauts training together for a mission, 25% of the utterances involved perspective takings [Trafton et al., 2005a].

Levelt [1996] categorize perspectives into three types: (1) deictic perspective in reference to the speakers' points of view, *e.g.* "on my left"; (2) intrinsic perspective in reference to the objects' points of view, *e.g.* "in front of the car"; (3) absolute perspective in reference to the world frame, *e.g.* "north". Levinson [1996] merge addressee-centered and deictic perspectives into relative perspective in reference to landmarks.

Most work assumes that people always take robots' perspectives when giving robots instructions for tabletop [Guadarrama et al., 2013, Misra et al., 2014] and navigation [Fischer, 2006] tasks. When a person instructs a robot to perform a task with some ambiguity, the person prefers the robot to take the person's perspective [Trafton et al., 2005b]. In object identification tasks, people intuitively use robots' perspectives [Moratz and Tenbrink, 2006]. Conversely, human-human collaboration literature reveals that solo people with imaginary human partners are uniform in taking their partners' perspectives while people with real human partners are not [Schober, 1993], indicating that there is no consensus on common perspectives. Hence, in our task where participants instruct partners sitting across the table, we analyze and rank different perspectives the participants have used.

### *Ambiguity*

There are three sources of ambiguity in REs which make them very hard for human hearers to understand. (1) unknown perspective-taking of speakers [Levelt, 1996]; (2) ambiguities in target objects, landmarks, and spatial relations between them [Fischer and Moratz, 2001]; (3) ambiguities in *applicability regions* for large and not mutually exclusive spatial relations [Moratz and Tenbrink, 2006].

In an experiment in which people were asked to write navigation

instructions to another person, the other person was only able to successfully follow 69% of the nearly-700 instructions while the others are ambiguous [MacMahon et al., 2006]. In a similar study, subjects were only able to navigate to the final destination 68% of the time [Wei et al., 2009]. We analyze the effects of general ambiguity and the ambiguity caused by unknown perspective on the easiness that people can understand the instruction.

*Human Partner vs. Robot Partner*

Robot is treated as a communication partner who needs more basic instructions than a human interlocutor [Fischer and Moratz, 2001]. This is consistent with another study where half of the participants instruct robots by decomposing the action and describing paths to adapt to assumed robot linguistic and perceptual abilities [Moratz et al., 2001]. Seniors want a streamlined communication with a task-oriented robot and do not want to speak to robots the same way they speak to people [Carlson et al., 2014]. Therefore, we also investigate the difference between the way people speak to a robot and to a human partner in our tabletop manipulation scenario.

## 2.2 *Study 1: Collecting Language Examples*

To understand how humans use visual features and spatial relations in their referring expressions for tabletop manipulation tasks, we collected a corpus of instructions generated by 100 online participants.

*Study design*

To collect referring expressions that represents tasks that required object referring and perspective taking, we created a set of stimulus images. Each image represents a configuration with 15 simplified block objects in different colors (orange, yellow, green, or blue) on a table as shown in Fig. 2.2. We first generated 14 images of configuration independently, each of which included different visual features and spatial relations, such as a single block of one color, pairs of blocks closely placed, blocks separated from a cluster, and blocks within or near clusters of a single color. Then we placed red-arrow indicators above two different target blocks independently in each image and ended up with 14 pairs of configuration (28 images of configuration in total).

This stimulus design is chosen to elicit instructions that rely more on the visual and spatial arrangement of the blocks than their individual appearance for the purposes of human-robot interaction. In order to

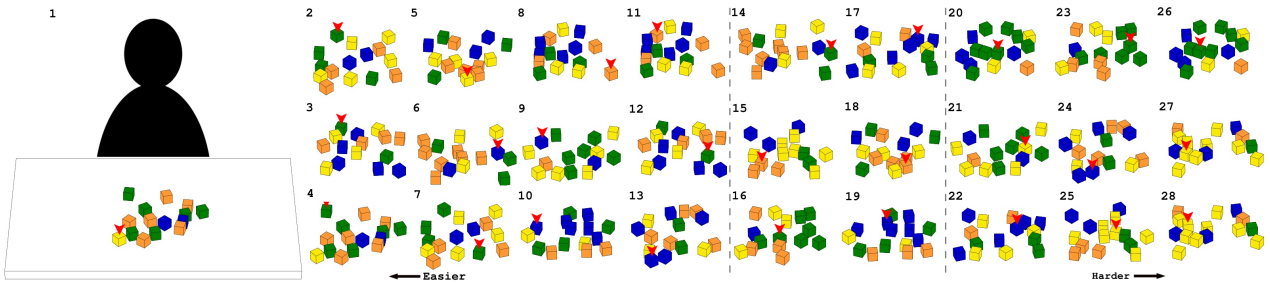



Figure 2.4: Scenes used to elicit spatial references. Online participants were asked to write how they would instruct the silhouetted figure to pick up the block indicated with the red arrow. The block configurations on the left were rated as the easiest to describe, while the configurations on the right were the most difficult.

capture clear instructions for a potential partner, this task asked participants to instruct a hypothetical partner to pick up the indicated block as though that partner could not see the indication arrow. The partner (indicated by the silhouetted figure in the images) was seated across the table from the participant viewing the scene. This setup required participants to be clear about the target blocks and the perspectives where they were describing the blocks.

Prior work indicates that people communicate with robots differently from with other people [Fischer and Moratz, 2001, Carlson et al., 2014, Moratz et al., 2001]. Therefore, we varied whether participants were told that their partner (the silhouette figure) was human or robot. [18] Participants were randomly assigned to either the human or the robot condition, and this assignment was the same for every stimulus they saw. The stimuli were otherwise identical across conditions.

We analyze the results with respect to these hypotheses:

*H1* People use different words when talking to human and robot. Specifically, people are *more verbose*, *more polite*, and use *more partner-based perspective words* to human partners than robot partners.

*H2* The frequency of words used in all instructions correlates with the features used in visual search, including *color, stereoscopic depth, line arrangement, curvature, intersection, and terminator* [Wolfe, 1994].

*H3* Subjective ratings of sentence difficulty correlate with the number of spatial references required to indicate the target objects.

[18] We did not change the visual appearance of the silhouette

### *Study Procedure*

We deployed our study through Amazon's Mechanical Turk[19]. Each participant was randomly assigned a partner condition (human vs robot) and 14 trials. In each trial, participants were presented with an image, like the one on the left side of Fig. 2.2, which was randomly chosen from the two predefined configurations in each of the 14 pairs of configuration. The participants then typed their instructions and rated the diffi-

[19] www.mturk.com

culty of describing that block on a 5-point scale. For each trial, we also collected the completion time. After completing 14 trials, participants were asked (1) if they followed any particular strategies when giving instructions; (2) how challenging the task was overall; (3) for any additional comments they had about the task. Finally, we collected demographics such as age, gender, computer usage, handedness, primary language (English or not), and experience with robots.

*Metrics*

We analyze the collected corpus for language features. To analyze the differences on word choice between human-partner group and robot-partner group (H1), we computed:

- *word count*: number of words for each instruction.
- *politeness*: presence of the word "please" in each instruction.
- *perspective*: whether the instruction explicitly refers to participant's perspective (egocentric), partner's perspective (addressee-centered), neither perspective [20] , or unknown perspective (instruction implicitly refer to some perspectives) (see Table. 2.2 for details) [21].

Word count and politeness were automatically extracted from the text. Perspective was manually coded by four raters who coded the same 10% of the data and iterated until high inter-rater reliability, measured by averaging the result of pairwise Cohen's $\kappa$ tests. The average $\kappa$ value for perspective was 0.85, indicating high inter-rater reliability. Once this reliability established, the four raters each processed one quarter of the remainder of the instructions.

| Type | P1 | P2 | Example |
|---|---|---|---|
| **Participant Perspective** | + | - | "the block that is to **my** rightest." |
| | | | "**my** left most blue block" |
| **Partner Perspective** | - | + | "the block on **your** left" |
| | | | "second from the right from **your** view" |
| **Neither Perspective** | - | - | "closest to you" |
| | | | "the top one in a triangle formation" |
| **Unknown Perspective** | ? | ? | "to the **left** of the yellow block" |
| | | | "the block that is on far **right**" |

[20] *Neither Perspective* sentences only use perspective-independent directional information. For example, "closer to you" should be classified as neither perspective instead of partner perspective, because it contains a perspective-independent reference to a landmark, "you," but not perspective-dependent relationships such as "on my left" and "on your right".

[21] Object-centered perspective is not considered because blocks are all the same except color [Levelt, 1996, Levinson, 1996].

Table 2.1: Possible perspectives. (P1=Participant P2=Partner).

To compare the features used in our corpus with visual search (H2), we classify words into categories adapted from visual search literature [Wolfe, 1994]. The categories are listed in Table. 2.2 and presented in the order of *word frequency*, the number of instructions that contain words from the category divided by the size of the corpus.

To verify the correlation between perceived difficulty and the num-

| Word Category | Description |
|---|---|
| **Action** | An action to perform |
| **Object** | An object in configuration |
| **Color** | Color of object |
| **Ordering/Quantity** | Ordering/Quantization of objects |
| **Density** | Concentration of objects (or lack of) |
| **Pattern/Shape** | A readily apparent formation |
| **Orientation** | The direction an object faces |
| **Environmental** | Reference to an object in the environment |
| **Spatial Reference** | Positional reference relating two things |
| **Perspective** | Explicitly indicates perspective |

Table 2.2: Word categories and their brief descriptions

ber of required spatial references (H3), we compare the *subjective difficulty rating* (Likert scale $1$ (easy) to $5$ (difficult)) to the following objective measures:

- *word count*: as computed for H1.
- *spatial reference count*: as computed for H2.
- *ordering and quantity word count*: as computed for H2.
- *completion time*: the duration from when a participant loads a new stimulus to when the participant hits the submit button for his/her instruction.

## 2.3 *Study 1 Results*

In the study, we recruited $120$ participants and over-sampled $1680$ instructions so that we could account for errors in data collection process and invalid responses. We remove $10$ sentences ($0.006\%$) that either do not refer to any blocks or are otherwise nonsensical. For consistent and organized analysis, we randomly select $1400$ sentences from the remaining $1670$ to ensure that each of the $28$ configurations has exactly $50$ instructions divided as evenly as possible between partner conditions. We analyze the $1400$ sentences selected in this manner.

Our data is open-sourced and available online [22].

[22] https://personalrobotics.github.io/collaborative_manipulation_corpus/

### *Visual Features*

To address our belief regarding the correlations between the visual features in our corpus and visual search literature, we analyze how frequently referring expressions contain visual search features. A summary of the results are in Table. 2.3.

First, a reference to color is used in nearly every instruction, so color is such a salient feature in our stimuli as well as in visual search. Next, although orientation, size, and motion are also strongly influential ac-

| Visual Feature | Count | Frequency |
|---|---|---|
| **Color** | 1301 | 0.929 |
| **Ordering/Quantity** | 498 | 0.356 |
| **Density** | 456 | 0.326 |
| **Pattern/Shape** | 60 | 0.043 |
| **Orientation** | 1 | 0.001 |

Table 2.3: Visual feature frequencies and feature-included sentence counts over all 1400 sentences ranked from most to least frequent

cording to visual search literature, they are almost never referenced in our corpus. This is likely due to the fact that in our study, blocks have 4-way symmetry and are not oriented in any particular direction, have the same size, and are static [Wolfe, 1994].

Without many other visual indicators, participants frequently referred to "dense" regions of one particular feature. For example, "edge of table", "end of table", "side of table", and "corner of table" in our data mean the end of an object, which are mapped to the "terminator" feature in Wolfe [1994]. "Isolated", "alone", "apart", and "solitary" in our data mean a vacant area with nothing in it. These words could be understood as the end of all elements, which are mapped to the "terminator" feature in Wolfe [1994]. "Cluster", "pair", "surround", and "sandwiched" in our data mean the boundary between multiple regions, which are mapped to the "intersection" feature in Wolfe [1994]. "Row", "aligned", "column", "string", and "stack" in our data mean a line of consistent features, such as a line of blocks with the same color, which are mapped to the "line" feature in Wolfe [1994]. "Diamond", "rectangle", "triangle", and "square", in our data mean a shape of consistent features, such as a group of blocks which forms a diamond, which are mapped to the "curvature" feature in Wolfe [1994]. These references are observed in the literature with less consistency than color and orientation are [Wolfe, 1994].

Finally, although ordering/quantity does not fit the paradigm of visual search [Wolfe, 1994] as well as the previously mentioned features did, these words are closely related to the concepts of pattern/shape and density. "The third block in the line" and "The second block from the cluster" are examples respectively. We find high occurrence of ordering/quantity words especially in relation to other visual search terms.

In summary, we find that the observed frequency of many categories of words in our corpus, including color, density, shape, and ordering/quantity, closely matched what we expected based upon the visual search literature [Wolfe, 1994].

*Spatial Reference*

Subjective ratings of sentence difficulty correlate with the number of spatial references required to indicate the target blocks.

We evaluate the effect of perceived difficulty on word choice in each instruction by investigating the correlations between subjective rating of difficulty, overall word count, number of spatial references, number of order/quantity words, and completion time. We excluded any trials on which the participant did not provide a subjective rating of difficulty and two outlier trials for which the response times were greater than 10 minutes, which ended up with 1353 sentences.

Because we use ordinal measures in this evaluation (*e.g.* subjective difficulty is rated on a 5-point scale), we conduct a Spearman's rank-order correlation to determine the relationship among the five metrics identified. There are statistically significant correlations across all pairs of metrics ($p < 0.01$ for all, which accounts for multiple comparisons).

Table. 2.3 details these correlations.

- As expected, there is a clear positive correlation (0.528) between word count and difficulty (Fig. 2.5): easier scenes require fewer words to describe.
- As expected, there is a clear positive correlation (0.508) between completion time and difficulty (Fig. 2.6): harder scenes require more time.
- Interestingly, easier rated tasks generally require fewer spatial references (Fig. 2.7): more spatial references in a sentence imply a greater depth of search to find the correct answer.

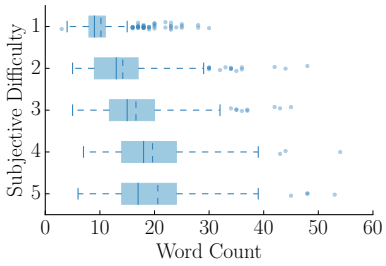
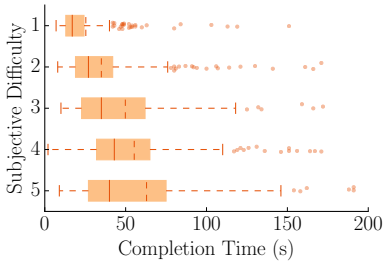



Figure 2.5: The effect of subjective difficulty ratings on word count.



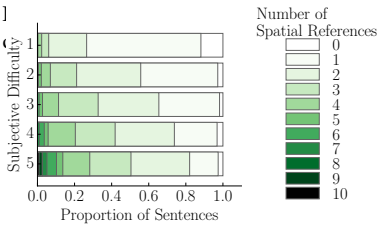




Figure 2.7: The effect of subjective difficulty ratings on the number of spatial references.

| | Difficulty | Word Count | Spatial Reference | Order/Quantity Word | Completion Time |
|---|---|---|---|---|---|
| **Difficulty** | — | 0.528 | 0.213 | 0.338 | 0.508 |
| **Word Count** | 0.528 | — | 0.416 | 0.425 | 0.682 |
| **Spatial Reference** | 0.213 | 0.416 | — | 0.082 | 0.262 |
| **Order/Quantity Word** | 0.338 | 0.425 | 0.082 | — | 0.350 |
| **Completion Time** | 0.508 | 0.682 | 0.262 | 0.350 | — |

Table 2.4: Spearman's rho correlations of sentence features and scene difficulty evaluations. All correlations are statistically significant with $p < 0.01$.

## 2.4 *Study 2: Evaluating Language for Clarity*

To study the principles of clear spatial references in human robot collaboration, we need to validate the clarity of the instructions obtained in Study 1 (Sec. 2.2). First, we manually coded the instructions in terms of two criteria (perspectives had already been coded in Study 1 (Sec. 2.2)):

- *Block ambiguity* - the number of blocks that people could possibly identify from the image based on the given instructions.
- *Perspective* - whether there is an explicitly stated perspective provided in the instructionss.

Subsequently, we ran a follow up study to empirically measure the clarity of the sentences. In this second study, participants were presented with the stimuli from Study 1 (Sec. 2.2) (without red indication arrows) alongside the corresponding block descriptions from Study 1 (Sec. 2.2), and were asked to click on the indicated blocks. We collected responses from ten participants for each instruction from Study 1 (Sec. 2.2).

### *Coding instructions for Clarity*

We manually code each of the instruction from Study 1 (Sec. 2.2) for perspective and general block ambiguity. The coding measures, inter-rater reliability scores, and preliminary findings are described next.

*Perspective* As described in Sec. 2.2 and Table. 2.2, all sentences are labeled with perspective information. Among all the 1400 sentences, 454 (32.4%) sentences use unknown perspective, 339 (24.2%) sentences use partner perspective, 15 (1.07%) sentences use participant perspective, and 589 (42.1%) sentences use neither perspective.

*Block Ambiguity* Block ambiguity is the number of blocks this instruction could possibly identify. For our definition, no inferences are allowed when determining block ambiguity. Every detail which could possibly lead to ambiguity should be considered and expanded to different referred blocks. For example, the spatial relation "surrounded" could mean either partially or fully surrounded, which makes the sentence "the block that is surrounded by three blocks" potentially ambiguous. Unknown perspective could also lead to block ambiguity if different blocks are identified under the assumption of different perspectives.

We manually code each of the instructions from Study 1 (Sec. 2.2) for "high" or "low" block ambiguity. If a sentence could refer to only one single block in the scene, it is rated as "low" ambiguity. Otherwise, it is rated as "high" ambiguity. We use the same process as in Sec. 2.2 to establish inter-rater reliability. On 10% of the data, the average Cohen's $\kappa$ for the four raters is 0.68, indicating high rater agreement. Each rater subsequently code one quarter of the remaining data.

Among all the 1400 sentences coded, 895 (63.9%) sentences are not block ambiguous with only one block being referred to, while 492 (36.1%) sentences possibly refer to more than one block.

*Online Study Design and Procedure*

As mentioned above, the goal of the second study is to investigate the clarity of instructions, which will guide us through the future research on robot-generated instructions. In this online study, new Amazon Mechanical Turk participants were shown 40 configurations randomly chosen from the pool of 28 configurations generated in Study 1 (Sec. 2.2). Each configuration was presented alongside one of the corresponding instruction collected from Study 1 (Sec. 2.2). We would make sure that the clarity of all the collected instructions in Study 1 (Sec. 2.2) were evaluated here. Then the participants were asked to click on the block that best matched each instruction. In Fig. 2.8, as people moved their mouse over the image, a red circle appeared over the blocks to show them which block they would be selecting. When they clicked on the block, a black and white checkered circle would appear around the selected block. Continuing to move the mouse would present a red circle on those blocks which the participants could then click on to change their answer. Then we measured the participant's accuracy at selecting the indicated block.

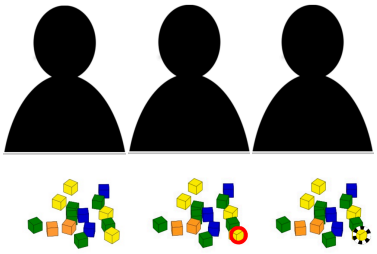


Figure 2.8: A scene in the second study

We compute the following metrics for Study 2:

- *Final Answer* - whether a participant picks the correct block.
- *Accuracy* - average over 10 participants of final answer for each instruction.
- *Completion Time* - duration from moment when the page finishes loading to the moment when a participant clicks the next button to proceed.

Based on our ambiguity measures and the results from Study 2, we hypothesize that:

*H4* *Block ambiguous* sentences will take participants in Study 2 *more time* and participants will be *less accurate* in discerning the referred block.

*H5* Sentences with *unknown perspective* will take participants in Study 2 *more time* and they will be *less accurate* in discerning the referred block. Conversely, sentences with *neither perspective* will take *less time* and participants will be *more accurate* in discerning the referred block.

## 2.5 Study 2 Result

We collect the responses from 356 participants and randomly select 10 responses for each of the 1400 sentences from Study 1 (Sec. 2.2). We evaluate the participant performance in Study 2 on the set of sentences from Study 1 (Sec. 2.2) by measuring their accuracy and completion time as described above. We also compare the objective accuracy measure to our manually-coded block ambiguity and perspective taking.

*Hypothesis H4*

First, we investigate block ambiguity by conducting an independent-samples t-test measuring the effect of block ambiguity (low or high) on accuracy (Fig. 2.9) and completion time (Fig. 2.10). There are significant results for both accuracy ($t(1398) = 13.888, p < 0.005$) and completion time ($t(1398) = -5.983, p < 0.005$). Accuracy is lower and completion time is higher on sentences that contain ambiguous block references (H4). These results confirm that block ambiguous statements take longer amounts of time for participants to process and participants are less accurate in discerning the referred block.

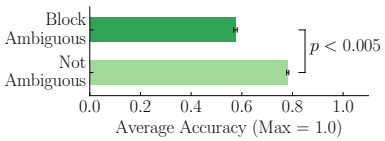



Figure 2.9: The effect of block ambiguity on average selection accuracy.

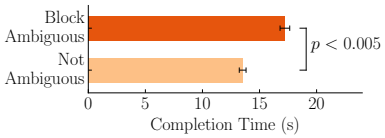



Figure 2.10: The effect of block ambiguity on average completion time.

*Hypothesis H5*

Next, we analyze perspective taking by conducting a one-way ANOVA measuring the effect of perspective type (participant, partner, neither, or unknown) on accuracy (Fig. 2.11) and completion time (Fig. 2.12). Perspective type has a significant effect for both accuracy ($F(3, 1396) = 43.655, p < 0.005$) and completion time ($F(3, 1396) = 34.607, p < 0.005$). Sentences that use neither perspective have higher accuracies ($M = 0.802, SD = 0.240$) than sentences that use partner ($M = 0.662, SD = .278, p = 0.019$) or unknown ($M = 0.619, SD = 0.307, p = 0.017$) perspective (H5). Similarly, average completion time is lower for sentences that use neither perspective ($M = 11.418s, SD = 10.56$) than partner ($M = 16.881, SD = 9.81, p < 0.001$) or unknown ($M = 17.756, SD = 12.03, p < 0.001$) perspective (H5). No other significant differences are found. These results confirm that neither perspective statements take shorter amounts of time for participants to process and participants are more accurate in discerning the referred block. At the same time, unknown perspective statements take participants longer time and participants are less accurate.

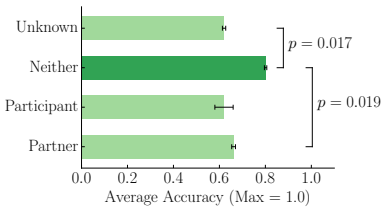



Figure 2.11: The effect of perspective on average selection accuracy.

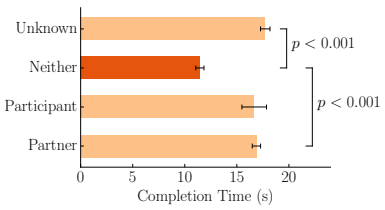



Figure 2.12: The effect of perspective on average completion time.

Additionally, we observe that participants in Study 2 have lower accuracy on sentences that participants in Study 1 (Sec. 2.2) label as more difficult (Fig. 2.13). This result is not surprising as participants who have trouble writing a clear sentence would likely rate the task as difficult.

We conclude that hypotheses 4 and 5 are both supported. Block ambiguity and unknown perspective are both correlated with higher completion times and lower accuracies. The type of perspective in the sentence has a significant effect on accuracy: when the instructions are written in neither perspective, participants in Study 2 have higher accuracy than any of the other perspectives.

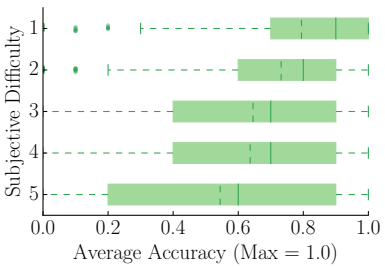



Figure 2.13: The effect of the subjective difficulty ratings from Study 1 (Sec. 2.2) on average selection accuracy from Study 2 (Sec. 2.4).

## 2.6 Conclusion

We created a corpus of REs when identifying objects in a potentially ambiguous setting. We identified a cognitive process which plays a significant role in the formation of these REs. We defined metrics to aid in scoring the optimality of a RE. We designed an evaluation process based on these metrics. And finally, we performed an initial, yet broad, analysis on our corpus that was able to uncover a handful of insights.

Our findings suggest that sentence clarity suffers when there is either an ambiguity related to the number of blocks a sentence can specify or an ambiguity related to perspective. To support our next step in generating language-based explanations, we will follow the insights we collect in this section.

- The visual features and spatial relations people used are based on the frequency of words in our corpus [Li et al., 2016] and visual search literatures [Wolfe, 1994].
- The more spatial relations used in REs, the harder time people will have in understanding these REs.
- A clear RE should eliminate the perspective ambiguity by specifying perspective and eliminate the block ambiguity by making sure that it only identifies the target block.

## 2.7 Future Work

We will discuss a few of these insights in the following section. In analyzing the corpus, we discovered that participants generally followed one of three approaches when writing instructions:

- a *natural* approach where they used embedded clauses linked together by words indicating spatial relationships such as in the instruction "Pick up the yellow block directly to the left of the rightmost blue block.",
- an *algorithmic* approach, which a majority of the users employed, where they partitioned their instructions in stages reflecting a visual search process such as in the instruction "there is an orange block on the right side of the table. Next to this orange block is a yellow block. Please pick up the yellow block touching the yellow block."
- an *active language* approach where they provided instructions asking the partner to move their arms (usually) in a certain way so as to grasp the desired object such as in the instruction "stand up, reach out over the table, and grab the yellow block that is touching the blue block closest to me." In certain instructions, the participant would even offer active guidance (which is of course not an option in a one shot response written in a web form).

Among the three, the algorithmic approach is often the clearest but feels less natural. We believe that these observations about instruction approach types will lend themselves well to further investigation on user instruction preferences. For example, some users might prefer to give algorithmic descriptions which iteratively reduce ambiguity as needed, while other users might prefer to utilize active language where they guide the robots motions via iterative movement-driven instruction.

Further, the descriptions requiring perspective takings usually have perspective-dependent terms like 'right', 'left', 'above' and 'below'. If establishing perspective proves to be difficult in a scenario, robots should prefer to use perspective-independent spatial relations. That is, if the robot is able to generate a description using our definition of 'neither' perspective, it should prefer to do so over other descriptive strategies.

However, there are also exceptions of preferring 'neither perspective. For example, in the scene as shown in Fig. 2.14, if we force ourselves to use 'neither' perspective, we might come up with this instruction, "pick up the blue block that is closer to you and right next to the yellow block." This instruction is clear, but involves many features. This might give hearers a hard time to understand our instruction based on our finding that subjective difficulty ratings are strongly correlated with the number of spatial references required to indicate the target block. On the other hand, if we do not have to use 'neither' perspective, we can have a much more efficient instruction, "pick up the blue block on your far right." Therefore, a trade-off has to be made between preferring 'neither' perspective and producing efficient instruction.

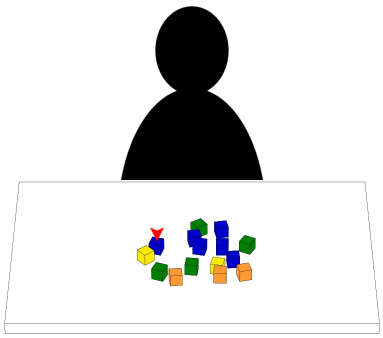


Figure 2.14: In this scene, a tradeoff has to be made between preferring 'neither' perspective and producing efficient instruction.

# 3
# *Generating Explanations in Natural Language*

In Sec. 2, we find that the clarity of referring expressions suffers when there is either a *block ambiguity* related to the number of blocks a sentence can identify or a *perspective ambiguity* related to specified perspectives.

Since REG does not necessarily require a human standing on the opposite side from his partner, we don't need to worry about the issue of perspective ambiguity. We can assume that speakers and hearers share the same perspectives. We mainly focus on the more general problem, which is how to generate clear referring expressions.

To resolve the issue of block ambiguity, we need to select the set of features in two steps. The first step is to choose the set of features to distinguish the target object from its distractors. The second step is to further optimize the set of features for the ease of human understanding based on the frequency of each feature appeared in our corpus [Li et al., 2016].

REG is a process where the speaker first identify the target object, gather perceptual information, and select a set of features for the object, which could be considered as the converse problem to visual search, in which an observer is given a set of features of an object and then identify it within a visual scene [Clarke et al., 2013]. The features people use in visual search are similar to the features they select in content selection [Clarke et al., 2013]. We can also resort to visual search literatures [Wolfe, 1994] for human preferences on features.

## *3.1 Related Work*

### *Visual Feature*

Researchers in the field of REG have considered different sets of visual features in their REG algorithms, such as color, shape [Matuszek et al., 2014], type [FitzGerald et al., 2013] and quantity [Dale, 1989].

### *Spatial Relation*

Qualitative spatial reasoning models a spatial relation over the spatial entities, such as points, lines, planes, and regions, as a constraint in the scene. Checking the consistency between a given expression and the set of constraints imposed by the scene could be reduced to a Constraint Satisfaction Problem (CSP) problem [Chen et al., 2015]. People model spatial relations between two spatial regions as constraints based on the distance of two regions [Cohn and Hazarika, 2001, Cohn and Renz, 2008], and a composition of interior, boundary, and exterior of two regions [Egenhofer and Vasardani, 2007]. To reason about directions between point-based objects, Cone-Shaped Direction [Clementini and Di Felice, 1997], Projection-Based Direction [Isli et al., 2001], and Oriented Point Algebra [Moratz, 2006] split the $360^\circ$ unit circle into multiple ranges. To reason about distances between point-based objects, Rotation, Scaling and Translation [MacMillan et al., 2004] and Qualitative Trigonometry and Qualitative Arithmetic [Liu, 1998] split the 2D scene into multiple areas.

### *Referring Expression Understanding*

Many language understanding works follow an instruction-based learning framework, in which robots extract compositional structures or building generative and discriminative models for understanding route instructions [MacMahon et al., 2006].

One approach is to develop parsers to translates route instructions into formal logics via heuristics [Dzifcak et al., 2009] and into attribute-value pairs and robot actions via context-free grammar [MacMahon et al., 2006]. Forbes et al. [2015] develop a model to understand manipulation commands based on reachable space and object referring history.

More other works include a world model in interpreting language. Hsiao et al. [2008] develop the notion of object schema as a discrete structure of object attributes and address language grounding via schema searching and matching. Tellex et al. [2011] develop Generalized Grounding Graphs ($G^3$) as a world model which dynamically instantiates a factor graph to understand a route instruction based on its hierarchical and compositional semantic structure. Howard et al. [2014] develop Distributed Correspondence Graph (DCG) as a model for planning constraints from natural language instructions [Howard et al., 2014]. Paul et al. [2016] further extend DCG model to support abstract concepts in the world.

Other objects or humans in the environment could affect or facilitate language understanding. Matuszek et al. [2010], Vogel and Jurafsky [2010] parse instructions to path descriptions based on a labeled topol-

ogy. Kollar et al. [2010] present language understanding as inferring the most probable path given detected objects from route instructions. Liu and Chai [2015] enable robot to assess its perceptual differences with humans and mediate perceptual differences by interacting with humans via dialog. Yi et al. [2014, 2016a,b] extend DCG to model human constraints in robot plans.

### *Referring Expression Generation*

*Referring Expression Generation (REG)* [1] usually has 2 steps:

- *content selection or content determination* where the speaker determines the set of visual features and spatial relations to distinguish the target object from distractors [Krahmer et al., 2003].
- *surface realization* where the speaker realizes the selected visual features and spatial relations into natural language [Krahmer et al., 2003].

[1] Referring Expression Generation (REG): a discrimination task, where the system needs to communicate sufficient information (words or phrases) to identify one domain entity and distinguish it from other domain entities [Reiter et al., 2000, Reiter and Dale, 1997].

This thesis mainly focuses on content determination with the assumption that we have a good surface realizer. The REG algorithms in early stage follow Gricean maxim [Grice, 1975] to search for a referring expression with neither too much information which could be overwhelming, misleading, and boring, nor too little information which could be ambiguous [Gatt and Krahmer, 2017]. Full Brevity Algorithm (FB) conduct an exhaustive breadth-first search over all the sets of features until a smallest set can distinguish the target object [Dale, 1992]. But FB is NP hard and psychologically unrealistic. Greedy Heuristic algorithm conduct a depth-first search to find an available set of features by greedily adding the features with the most descriptive power [2] [Dale, 1989, 1992]. But the solution may not be optimal. Incremental algorithm (IA) is more psychologically realistic because it selects features based on a domain-specific preference or cognitive salience of all the features [Dale and Reiter, 1995]. For example, the algorithm would select color in prior to shape because people usually prefer color over shape. But IA might not return the shortest solution.

[2] Descriptive power: how many distractors a feature can rule out [Dale, 1989, 1992].

*Graph-based Algorithm (GBA)* REG algorithm was proposed to resolve the trade-off between psychological realism and language efficiency. GBA develops a *labeled directed multigraph*, called *REG graph*, to represent the scene with objects, visual features, spatial relations, and the preference ordering over all the features Krahmer et al. [2003]. Each object in the scene is represented as a node in the graph. Each unary feature, $e.g.$ visual feature "red", is represented as a self-loop. Each binary features, $e.g.$ spatial relation "next to", is represented as a binary edge [Krahmer et al., 2003]. We then make this graph psychologically realistic by assigning a cost to each edge based on the human preference on the associated feature. For example, Fig. 3.2(c) is the REG graph $G$

representing the objects, visual features, spatial relations, and assigned costs in the scene as shown in Fig. 3.2(a). GBA would search through all the possible subgraphs $g \in G$ and find the *unique* subgraph $g_u \in G$ as shown in Fig. 3.2(d), such that $g_u$ is *graph isomorphic* [3] to one and only one subgraph $g'_u \in G$. A clear referring expression Fig. 3.2(b) to identify the target object $c$ could be extracted from $g_u$ [Krahmer et al., 2003].

[3] Graph isomorphism is introduced in Sec. 3.1.Graph Isomorphism.

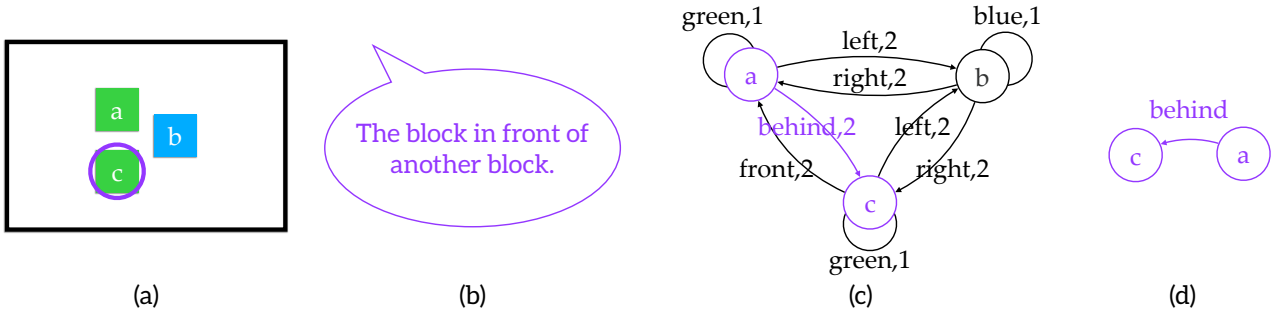



Figure 3.1: **(a)** a scene with a target object $c$ indicated by the purple circle; **(b)** a clear referring expression $r$ for the target object $c$ in (a); **(c)** the REG graph $G$ for the scene (a); **(d)** a unique subgraph $g \in G$ for identifying $c$. $g$ is only isomorphic to the subgraph $g' \in G$ in purple.

Pechmann [1989] shows that speakers tend to overspecify by including redundant features which have no contrastive value to refer to objects. For example, to identify the target block indicated in the purple circle in scene Fig. 3.2(a), people might prefer to say "the green block under a *green* block" over "the green block under a block" although the italicized feature "green" is redundant. Following overspecification, Viethen et al. [2013] extend GBA by modeling user's individual variation in overspecification. There are many other works on extending GBA to have more expressiveness in the generated referring expressions by introducing plurals [Krahmer and Van Deemter, 2012], basic-level category descriptor [Krahmer and Van Deemter, 2012] or entry-level category attribute [Kazemzadeh et al., 2014], conceptual gestalt [Deemter et al., 2012], underspecification [Van Deemter et al., 2012], egocentricity [Gatt et al., 2014, Van Deemter et al., 2012], serial dependency [Gatt et al., 2014], hierarchically structured domains [Paraboni et al., 2007], feature salience [Mitchell et al., 2013, Gatt et al., 2014], individual variation [Ferreira and Paraboni, 2014, Dale and Viethen, 2009], and non-determinism [Van Deemter et al., 2012]. The state of art GBA is Longest First algorithm [Viethen et al., 2013] [4], a branch and bound algorithm, which exhaustively search for all the referring expressions with the lowest cost and return the longest one among them for overspecification.

[4] Full code is available `http://www.m-mitchell.com/code/`.

Besides GBA, there are other approaches for REG. FitzGerald et al. [2013] train a log-linear model from a corpus for the probability distribution of a referring expression formulated as a logical expression that identifies target objects. Fang et al. [2014], Fang [2014] model REG as a collaborative decision making process to bridge the discrepancy between the robot and human world models.

Another line of work develops algorithms to jointly perform both content selection and surface realization. Engonopoulos and Koller [2014] define a synchronous grammar that relates surface strings with the target object and compute a chart to represent all the valid referring expressions. Tellex et al. [2014] enable robots to generate questions by using $G^3$ to model the human ability to understanding a question [Tellex et al., 2011].

More recent work has been focusing on developing the interface between computer vision and referring expressions to produce descriptions for objects in complex and realistic visual scenes [Mitchell et al., 2013, Kazemzadeh et al., 2014, Mao et al., 2016].

In this work, we choose to use GBA because REG graph gives us a qualitative model of the world. Based on this world model, we could bridge the gap between the robot's and human's world models and achieve a shared mental model [Converse, 1993] through human-robot interaction or physical robot manipulation guided by optimizing the REG graph [Liu and Chai, 2015]. Meanwhile, the REG graphs with assigned costs paves the way for a natural fusion between traditional rule-based approaches and more recent statistical approaches in a single algorithm [Krahmer et al., 2003].

The state of art GBA - Longest First algorithm [Viethen et al., 2013] and a well-developed indeterministic REG algorithm - Visible Objects Algorithm [5] [Mitchell et al., 2013] are evaluated on two well-known REG corpora, the GRE3D3 corpus [Viethen and Dale, 2008] as shown in Fig. 3.2 and the singular furniture section of the TUNA corpus [van Deemter et al., 2006]. However, both corpora only contains simple scenes like Fig. 3.2, instead of complex scenes as shown in Fig. 3.3. The current GBA algorithm does not extend well to a kitchen scene and does not support for the n-ary features that involve more than two objects we found in our corpus in Sec. 2, such as "a line of three blocks". In this chapter, we will propose several techniques to speed up the algorithm so that it scales to a larger scene in our corpus and a more comprehensive hierarchical graph structure to support n-ary features.

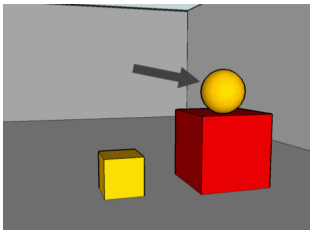


Figure 3.2: A simple scene from GRE3D3 corpus [Viethen and Dale, 2008].

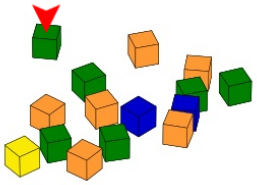


Figure 3.3: A complex scene from our corpus [Li et al., 2016].

[5] Full code is available https://github.com/mmitchellai/VisibleObjectsAlgorithm.

### *Graph isomorphism*

A labeled graph is defined as $G(V, E, L_V, L_E, \varphi)$, where $V$ is a set of vertices; $E \subseteq V \times V$ is a set of edges; $L_V$ and $L_E$ are sets of vertex labels and edge labels respectively; and $\varphi$ is a label function that defines the mappings $V \rightarrow L_V$ and $E \rightarrow L_E$.

A labeled graph $G_1(V_1, E_1, LV_1, LE_1, \varphi_1)$ is *isomorphic* to another graph $G_2(V_2, E_2, LV_2, LE_2, \varphi_2)$, if and only if there exists a bijection $f : V_1 \rightarrow V_2$ such that:

- $\forall u \in V_1, \varphi_1(u) = \varphi_2(f(u))$

- $\forall(u,v) \in E_1 \Leftrightarrow (f(u), f(v)) \in E_2$
- $\forall(u,v) \in E_1, \varphi_1(u,v) = \varphi_2(f(u), f(v))$

The bijection $f$ is an *isomorphism* between $G_1$ and $G_2$ [Jiang et al., 2013].

*Time Complexity* Graph isomorphism which is neither known to be solvable in polynomial time nor NP-complete [Conte et al., 2004] The guaranteed upper bound (worst-case analysis) [6] for checking graph isomorphism for arbitrary graphs with $n$ vertices is subexponential $exp(\sqrt{O(nlogn)})$ [Babai and Luks, 1983]. Recently a new paper shows that it could be solved in quasipolynomial time $exp((logn)^{O(1)})$ through group theoretic "local certificates" and combinatorial canonical partitioning techniques [Babai, 2015].

[6] http://dabacon.org/pontiff/?p=4148

*Topological Based Algorithms* Some algorithms reduce the computational complexity by imposing topological restrictions. We can enforce these topological restriction onto our REG to leverage the faster algorithms. Tree isomorphism can be solved in linear time by associating each node with a tuple that describes the complete history of its descendants [Aho and Hopcroft, 1974, Campbell and Radford, 1991]. However, it is hard to transform REG graph to a tree because of the undirected cycles, such as, "object 1 is on the left to 2, which is on the left to 3, which is behind 1." Isomorphism for planar graphs [Hopcroft and Wong, 1974] and graphs with bounded genus [Miller, 1980] can be solved in almost linear time. We can planarizing the REG graph into a closely related planar graph via weighted greedy pruning algorithm. The downside is that we might lose some edges with low cost, which damages our solution optimality [Krahmer et al., 2003]. Bounded valence (degree) graph isomorphism could be solved in polynomial time by being reduced to color automorphism problem for groups [Luks, 1982]. Our REG graph has a very large bound on valence which is equivalent to the number of visual features plus the number of possible spatial relations with all the other objects. It would be hard to improve the efficiency a lot even we transfer our REG graph to a bounded valence graph.

*Tree Search Based Algorithms* Most of the algorithms of exact graph matching are based on tree search with backtracking. The basic idea here is that a partial match is iteratively expanded by adding a new pair of matched nodes based on compatible constraints, preference orderings of node and edge attributes, and heuristics to prune unfruitful search paths. If the current partial mapping cannot be expanded furthermore, then the algorithm backtracks [Conte et al., 2004]. Some algorithms search for matches in the adjacency matrix [Ullmann, 1976], distance matrix [Schmidt and Druffel, 1976], or Constraint Satisfaction Problem

(CSP) [Larrosa and Valiente, 2002] representation of a graph and pruning the search with backtracks and refinements. The VF [Cordella et al., 1998a] and VF2 [Cordella et al., 2001] algorithms search for matches in a depth-first search manner and apply a heuristic that is based on the analysis of the sets of nodes adjacent to the ones already considered in the partial mapping to prune the search tree.

*Group Theory Based Algorithms* `nauty and Traces` [7] checks for isomorphism between graphs by verifying the equality of the adjacency matrices of their canonical forms [McKay et al., 1981, McKay and Piperno, 2014]. The equality verification can be done in $O(N^2)$ but the construction of canonical labeling can require exponential time in the worst case. It is really effective for matching a single small graph against a large fixed database of graphs by pre-computing canonical labels, but it doesn't exploit node and edge attributes of the graphs, compared to VF2 [Conte et al., 2004]. Other algorithms are available, such as `Saucy3` [8], `bliss` [9], `conauto` [10].

[7] http://pallini.di.uniroma1.it/index.html

[8] http://vlsicad.eecs.umich.edu/BK/SAUCY/

[9] http://www.tcs.hut.fi/Software/bliss/index.html

[10] https://sites.google.com/site/giconauto/

## 3.2 *Referring Expression Generation Algorithm*

### *Referring Expression Generation Graph*

The GBA algorithm would construct a REG graph to represent the scene [Krahmer et al., 2003], which create a mapping between the scene Fig. 3.2(a) and the REG graph Fig. 3.2(c).

To use REG graph to reason about referring expressions (RE), we need to map a RE to the REG graph. Considering a RE as a set of features, we can use a set of edges to represent these features, which means that we can use a subgraph to represent a RE. Now we have two mappings. We map a scene in Fig. 3.2(a) to a REG graph in Fig. 3.2(c). We also map a RE in Fig. 3.2(b) to a subgraph in Fig. 3.2(d).

A clear RE only identifies one object in the scene - the target object. Similarly, a unique subgraph is only isomorphic to one subgraph in the REG graph. We can further map the clarity of a RE to the uniqueness of a subgraph. For example, in the scene Fig. 3.2(a), we have a target object $c$ indicated by the red circle. The RE in Fig. 3.2(b) is ambiguous because it can identify both block $a$ and $c$. Correspondingly, the subgraph in Fig. 3.2(d) is isomorphic to two subgraphs in the REG graph Fig. 3.2(c) (one in orange and one in blue). An example of a clear RE is that in the scene Fig. 3.2(e), we have a target object $c$ indicated by the purple circle. The RE in Fig. 3.2(f) is clear because it can only identify block $c$. Correspondingly, the subgraph in Fig. 3.2(h) is isomorphic to only one subgraph in the REG graph Fig. 3.2(g) in purple. Therefore, generating a clear RE is equivalent to searching for a unique subgraph inside the

REG graph.

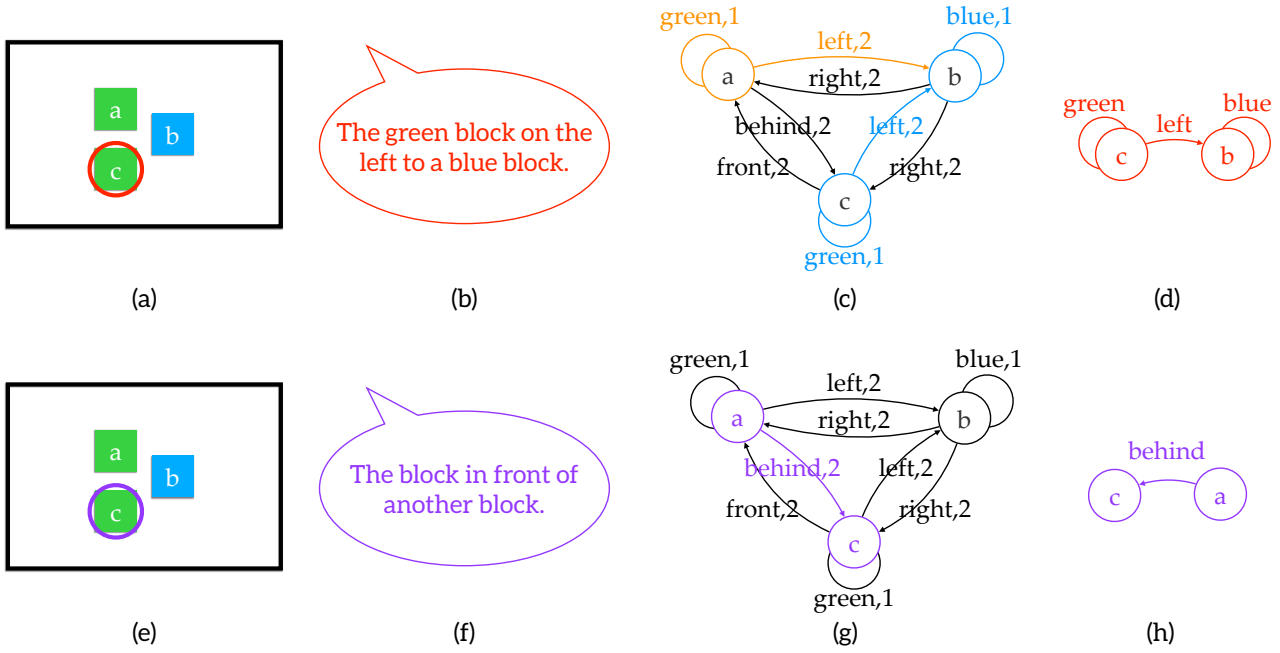



Figure 3.4: **(a)** a scene with a target object $c$ indicated by the red circle; **(b)** an ambiguous referring expression $r$ for the target object $c$ in (a); **(c)** the REG graph $G$ for the scene (a); **(d)** a non-unique subgraph $g \in G$ for identifying $c$. $g$ is isomorphic to the subgraphs $g_1 \in G$ in orange and $g_2 \in G$ in blue; **(e)** the same scene as (a), with the target object $c$ indicated by the purple circle; **(f)** a clear referring expression $r'$ for $c$ in (e); **(g)** the same REG graph $G$ as (c) for the scene (e); **(h)** a subgraph $g' \in G$ for identifying $c$ in (e). $g'$ is only isomorphic to the subgraph $g_3 \in G$ in purple.

*Graph-based Referring Expression Generation*

The original REG algorithm would search for the unique subgraph $g(V_g, E_g) \in G(V, E)$ with the minimum cost, as described in Equ. 3.1. Then this subgraph could distinguish the target object $o \in O$ from other objects.

$$g = \arg\min_{g \in G} \sum_{e \in E_g} \text{Cost}(e)$$
$$\text{subject to} \quad o \in V_g \text{ and } \nexists g' \neq g \in G \; s.t. \; g' \simeq g \tag{3.1}$$

This main loop of the algorithm has two steps, search process and isomorphism process. In the search process, the algorithm would start from a minimal subgraph which has only one node associated the target object. Then the program will iteratively generating subgraphs by adding new edges based on their costs in a breadth-first manner. In this way, all the possible subgraphs with the node associated with the target object will be generated. In the isomorphism process, the algorithm would check the uniqueness of each newly generated subgraph $g$ in the REG graph $G$ through checking graph isomorphism between $g$ and any subgraphs $g'$ of $G$. To verify graph isomorphism between $g$ and $g'$, the algorithm would iteratively map all the nodes and the edges derived

from the nodes in $g$ to $g'$ in a depth-first manner with backtracking. Note that the matching here means both graph structure and semantic, $i.e.$, the labels associated with the nodes and edges. For example, if we can find two subgraphs $g_1 \in G$ and $g_2 \in G$ which are isomorphic to $g \in G$, $i.e.$ $g \simeq g_1$ and $g \simeq g_2$, then the RE realized from $g$ should be able to identify two different objects in the scene, instead of the target object only, which means that the RE is ambiguous. The algorithm will repeat these two steps until a $g$ with a minimal cost among all the unique subgraphs is found. The full algorithm is described in [Krahmer et al., 2003] and the implementation of the state-of-art graph-based Longest First REG algorithm [Viethen et al., 2013] has implementation available [11].

[11] http://www.m-mitchell.com/code/index.html

### *Inefficiency in GBA*

As the number of objects increases, the number of possible subgraphs will increase. Then there are more possible graphs in the search process and the graphs to be matched are more complex in the isomorphism process.

We test the implementation in finding a RE for the block 10 in a scene with 15 blocks which was previously used in our user studies, as shown in Fig. 3.5. The algorithm generates a scene graph for REG, as shown in Fig. 3.6(a). The solution with a minimal cost is `[('9', 'left', '5'), ('10', 'behind', '9'), ('5', 'color', 'red')]` which represents a RE "block 10 is behind another block which is on the left to a red block." The search process takes 13.983 seconds and the isomorphism process takes 5.613 seconds. Therefore, we aim for speeding up both search and isomorphism processes.

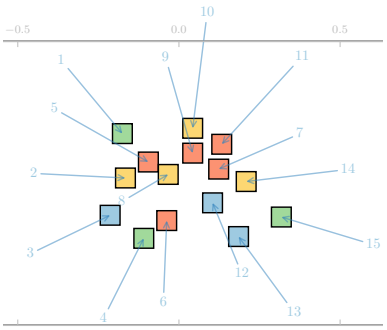


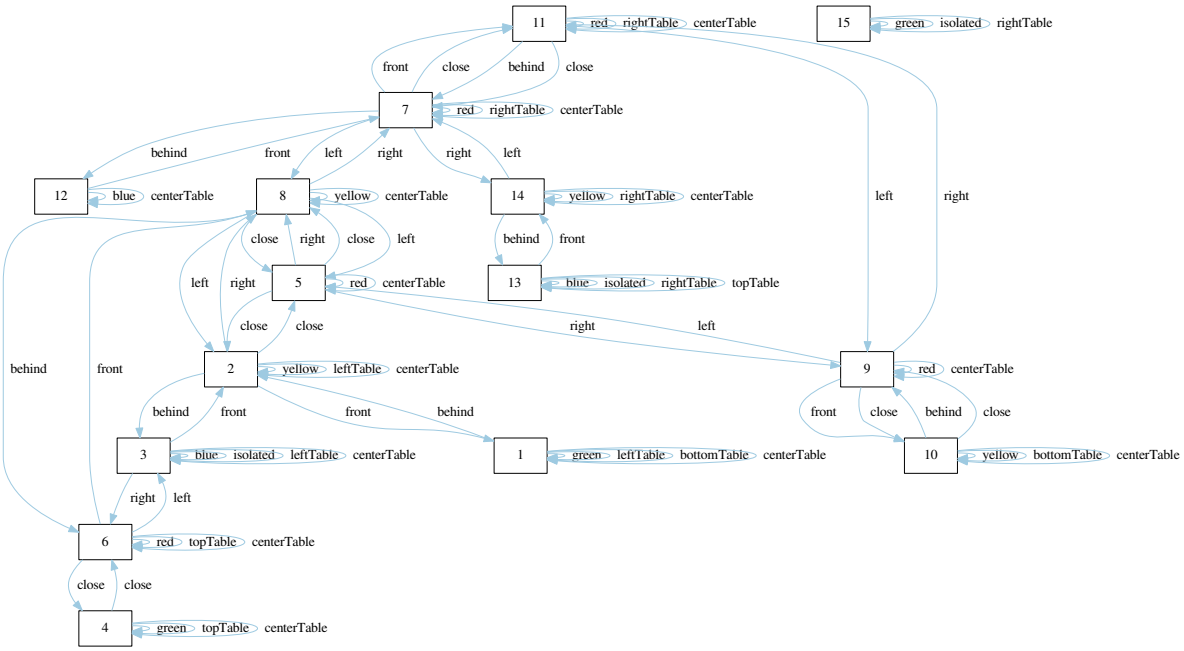

Figure 3.5: A simple scenario.

### *Speeding up GBA*

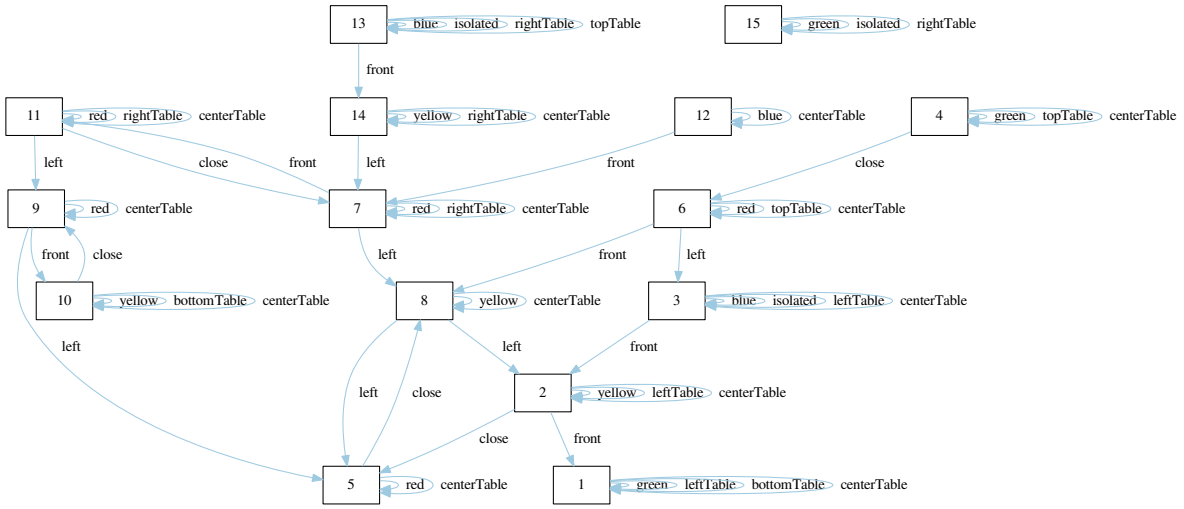
*Speeding Up the Search Process* The original algorithm in Krahmer et al. [2003], Viethen et al. [2013] has to generate all the possible subgraphs and find the one that has the minimal cost and could uniquely identify the target object, which is very time-consuming. In scenario Fig. 3.5, to find a RE for block 10, the search process takes 13.983 seconds.

To speed the algorithm up, we could prune the redundancy in the search tree without sacrificing the optimality in our solution based on the following characteristics of a REG graph.

- If the parent subgraph is unique, then whatever edges you add to it, the child subgraph will be always unique.
- Since a subgraph with less edges could have a higher cost than a subgraph with more edges, and vice versa. We cannot prune a branch in the search tree just because the corresponding subgraph has many




(a) The original REG graph for the scene in Fig. 3.5.




(b) The reduced REG graph for the scene in Fig. 3.5 based on the commutative rule.

Figure 3.6: The REG graph in (a) could be reduced to the one in (b) via the commutative rule.

edges. But instead, we can prune a branch in the search tree because the corresponding subgraph has a high cost.

Accordingly, we can reduce the number of generated subgraphs by pruning the search space in two ways.

- If a search branch reaches a unique subgraph, this subgraph becomes a new possible solution and we prune its child subgraphs.
- If a search branch reaches a subgraph that has a higher cost than the cost of the current best solution, then we eliminate this subgraph and prune its child subgraphs.

In the same scenario in Fig. 3.5, the more efficient algorithm finds the same referring expression for block 10 in 6.156 seconds. We also test this on all of our scenes used in the user studies as shown in Fig. 2.2. In average, this pruning search technique would reduce the total running time of REG about 50.77% as shown in in Fig. 3.2.

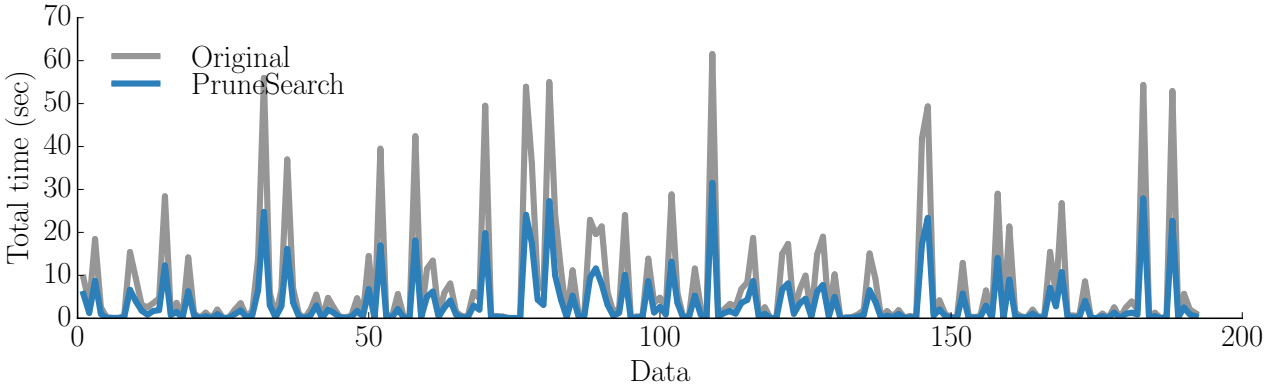


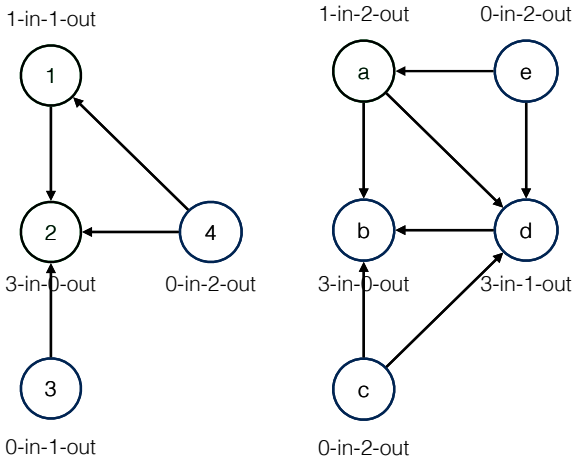

Figure 3.7: The effect of search pruning on the total running time to generate RE's for all the scenes as shown in Fig. 2.2.

*Speeding up the Isomorphism Process* The algorithm has to match each generated subgraph $g$ with the REG graph $G$ to check the uniqueness of $g$ in $G$. To match $g$ within $G$, the algorithm has to iterate through all the possible subgraphs $g' \in G$ and match $g$ with each of them, which is very time-consuming. In scenario Fig. 3.5, to find a RE for block 10, the isomorphism process takes 5.613 seconds.

To expedite the isomorphism process, we apply heuristics originally designed for Constraint Satisfaction Problems (CSP), inspired by Larrosa and Valiente [2002] who model graph isomorphism as a CSP. Our goal is to find a match for a subgraph $g(V_g, E_g)$ within the REG graph $G(V, E)$. The main loop of the depth-first graph matching algorithm would start from matching $v_g \in V_g$ with $v \in V$, then expand $v_g$ and $v$ in parallel, and match the newly derived edges and nodes respectively. The termination condition is all the nodes in $V_g$ are matched to $V$ and all the edges derived from $V_g$ are also matched to the edges derived from

the corresponding nodes in $V$. If so, $g$ is graph isomorphic to $g'$, denoted as $g \simeq g'$.

We can speed up the algorithm by matching all the $v_g \in V_g$ in an order determined by *Minimum Remaining Values (MRV)* heuristic. Based on this heuristic, the program would choose to match the most constrained node with the fewest legal possible matches as the next node $g$ to try. For example, we want to find a match for the subgraph $g$ as shown in Fig. 3.8(a) in the REG graph $G$ as shown in Fig. 3.8(b). Assuming we have already matched node 1 and $a$, based on the MRV heuristic, the next node to be matched in $g$ would be node 2 instead of 4 because 2 has a higher degree than 4, *i.e.* it would be harder to find a match for 2 than 4. The benefit of MRV is to make the program fail fast so that we could stop wasting our time on searching for a impossible match for $g$ in $G$.




(a) A subgraph $g$ refers to object 1.

(b) A REG graph $G$.

Figure 3.8: Our task is to match $g$ within $G$.

Assuming now we are trying to find a node $v \in V$ that could match $v_g \in V_g$. Let $D(v_g) = \{v_1, ..., v_k\} \subseteq V$ be the domain of possible matches, or candidates. The original algorithm would iteratively try all the node $v \in D(v_g)$ until finding a match $v_g \leftrightarrow v$, $s.t.$ all the edges derived from $v_g$ are matched to the edges derived from $v$ respectively.

One way to improve the efficiency is to match $v_g$ with all the $v \in D(v_g)$ in an order determined by *Least Constraining Value (LCV)* heuristic. Based on LCV, the algorithm would choose the node $v \in D(v_g)$ that leaves the most flexibility for the future unmatched neighboring nodes $v'_g \neq v_g \in V_g$. For example, assuming we have matched 1 in $g$ Fig. 3.8(a) with $a$ in $G$ Fig. 3.8(b), our task in this iteration is to find a match for node 2 derived from 1. Our candidates for 2 are the nodes derived from $a$, such as $b$, $d$, and $e$, *i.e.* the matching could be $2 \leftrightarrow b$, $2 \leftrightarrow d$, or $2 \leftrightarrow e$ if $g \simeq G$. First of all, $e$ is not possible because $e$ has less in-degree than

2. Therefore, we have to try matching 2 to $b$ and $d$ iteratively. Based on LCV heuristic, since $d$ has a higher or equal in- and out-degree than $b$, we will try $b$ before trying $d$ to leave $d$ a higher matching chance for the unmatched nodes 3 and 4 in $g$ Fig. 3.8(a). The benefit of LCV is to reduce failures by leaving more flexibility for the future. Note that here we do prefer algorithms which fail slow over the ones which fail fast. The reason is that even the program fails to match 2 and $b$, it has to try 2 and $d$ anyway. Fast failing will not prune the search.

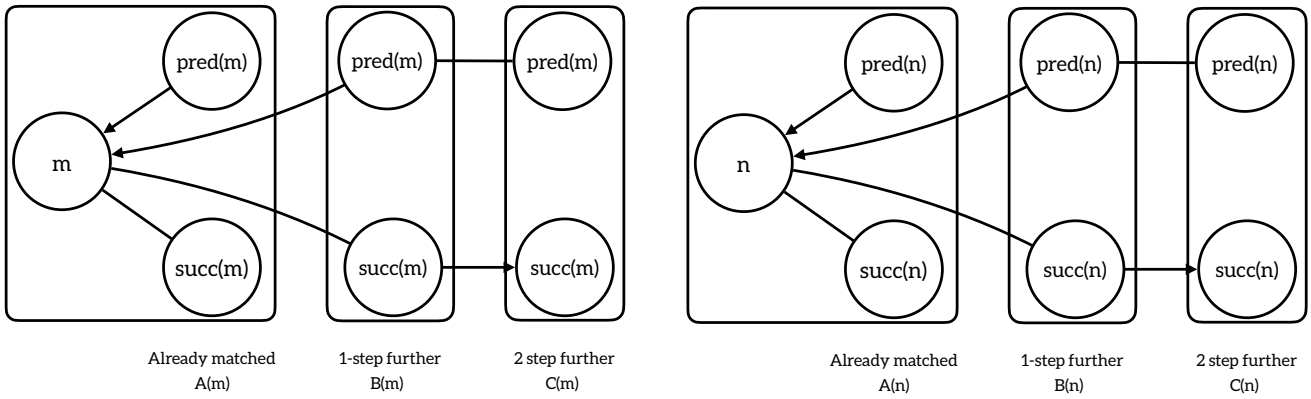
In addition, we also prune the match graph by applying constraints on the to-be-matched nodes. VF2 graph matching algorithm treat graph matching problem as a CSP and apply heuristics to speed it up [Cordella et al., 1998a, 2000, 1999, 1998b, 2001]. Here we apply 1 and 2 look-ahead heuristics [Cordella et al., 1998b] on the to-be-matched nodes.

For example, our goal is to match the REG graph $g$ in Fig. 3.9(a) with the full graph $G$ in Fig. 3.9(b). Assuming in this iteration, we are trying to match the node $m \in g$ with node $n \in G$. Since we only focus on $m$ and $n$, we view $g$ and $G$ as chains centered at $m$ and $n$. The set of nodes in $A(m)$ ($A(n)$) contains $m$ ($n$), and the successors and predecessors of $m$ ($n$) which have already been matched. The set of nodes in $B(m)$ ($B(n)$) contains the predecessors and successors of $m$ ($n$) that are 1 step further from $m$ ($n$) and have not been matched yet. The set of nodes in $C(m)$ ($C(n)$) contains the predecessors and successors of $m$ ($n$) that are 2 steps further from $m$ ($n$) and have not been matched yet.

Before iteratively matching each nodes and edges derived from $m$ and $n$, we can check if it is possible to match $m$ and $n$ in advance using the two following rules, 1 look-ahead and 2 look-ahead. From these two looking-ahead, if we find that it is impossible to match $m$ and $n$, then we don't need to check the derived nodes and edges.

- Check if the degrees of all the predecessors and successors of $m$ that are 1 step further from $m$ but not already mapped, $i.e.$ $B(m)$, are less than or equal to the ones of $B(n)$ respectively. In Fig. 3.9, we are comparing the degrees of nodes inside the $B(m)$ box with the degrees of nodes inside $B(n)$ box.
- Check if the degrees of all the predecessors and successors of $m$ that are 2 steps further from $m$ but not already mapped, $i.e.$ $C(m)$, are less than or equal to the ones of $C(n)$ respectively. In Fig. 3.9, we are comparing the degrees of nodes inside the $C(m)$ box with the degrees of nodes inside $C(n)$ box.

In the same scenario in Fig. 3.5, the algorithm with these heuristics would find the same RE for block 10 in 6.217 seconds. We also test this on all of our scenes used in the user studies as shown in Fig. 2.2. In average, this pruning search technique would reduce the total running time of REG about 16.65% as shown in in Fig. 3.2.




(a) A subgraph $g$.

(b) A full graph $G$.

Figure 3.9: In this iteration, we are trying to match $m \in g$ with $n \in G$.

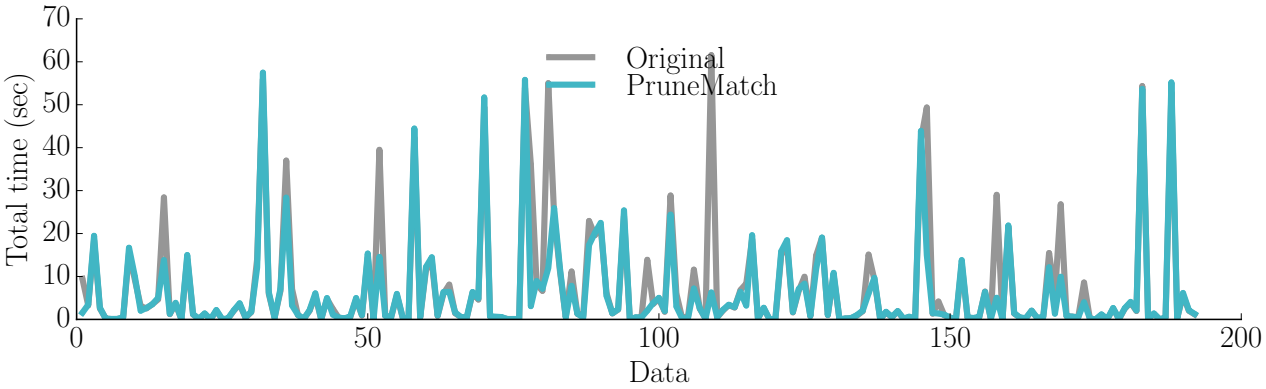



Figure 3.10: The effect of match pruning on the total running time to generate RE's for all the scenes as shown in Fig. 2.2.

*Speeding up Graph Building* Many binary relations are *commutative*. For example, if "object A is on the left to B", then "B is on the right to A." In REG, for each pair of binary edges labeled as commutative relations, we only need one of them in the REG graph and we can always deduct the other edge by applying the commutative rule on the existing edge. Therefore, we can significantly reduce the REG graph size by applying commutative rule. For example, we want to generate a new subgraph $g'$ from the current subgraph $g = (10 \rightarrow \text{behind} \rightarrow 9)$ by expanding the node 9 as shown in Fig. 3.6(a). Without commutative rule, $g'$ might be $(10 \rightarrow \text{behind} \rightarrow 9), (9 \rightarrow \text{front} \rightarrow 10)$. However, both $g$ and $g'$ correspond to the same referring expression for object 10, "the object behind another object."

One way to eliminate this redundancy in searching for new subgraphs is to apply commutative rule so that only one of the edge "front" and the edge "behind" remains in the full graph as shown in Fig. 3.6(b).

In the same scenario in Fig. 3.5, the algorithm after applying the com-

mutative rule finds the same referring expression for block 10 in 12.356 seconds. We also test this on all of our scenes used in the user studies as shown in Fig. 2.2. In average, this graph shrinking would reduce the total running time of REG about 21.50% as shown in Fig. 3.2.

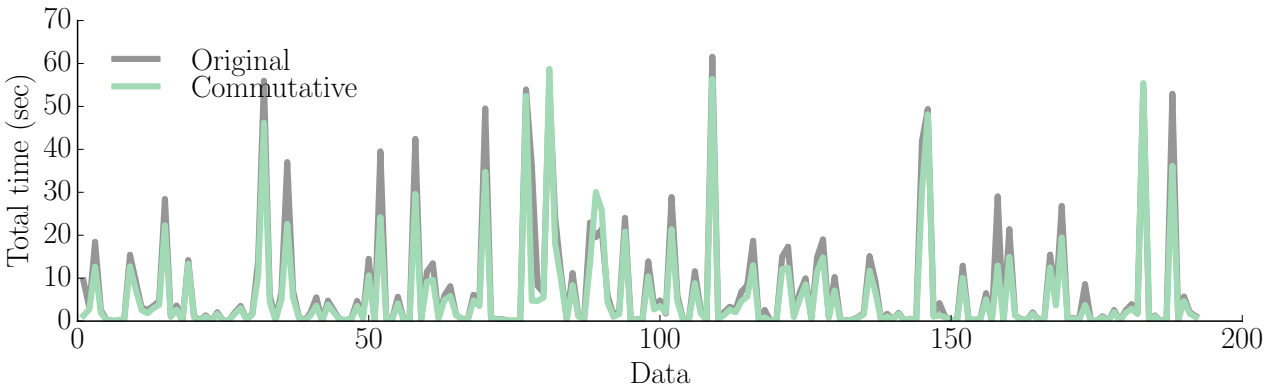



Figure 3.11: The effect of applying commutative rule on the total running time to generate RE's for all the scenes as shown in Fig. 2.2.

We composite the three techniques from above and test the algorithm on all of our scenes used in the user studies as shown in Fig. 2.2. In average, this pruning search technique would reduce the total running time of REG about 55.80% as shown in Fig. 3.12.

## 3.3 *Hierarchical REG Structure*

Based on our corpus, people tend to use *qualitative* expressions to describe a spatial information, including topology, orientation and distance. For example, people usually describe orientation by using qualitative expressions, *e.g.* "to the left of" and "northeast of" instead of quantitative expressions, *e.g.* "53 degrees". Similarly, people describe distance via qualitative categories, *e.g.* "A is close to B", or qualitative distance comparatives, *e.g.* "A is closer to B than to C", instead of quantitative values, *e.g.* "A is 1 meter away from B" [Renz and Nebel, 2007].

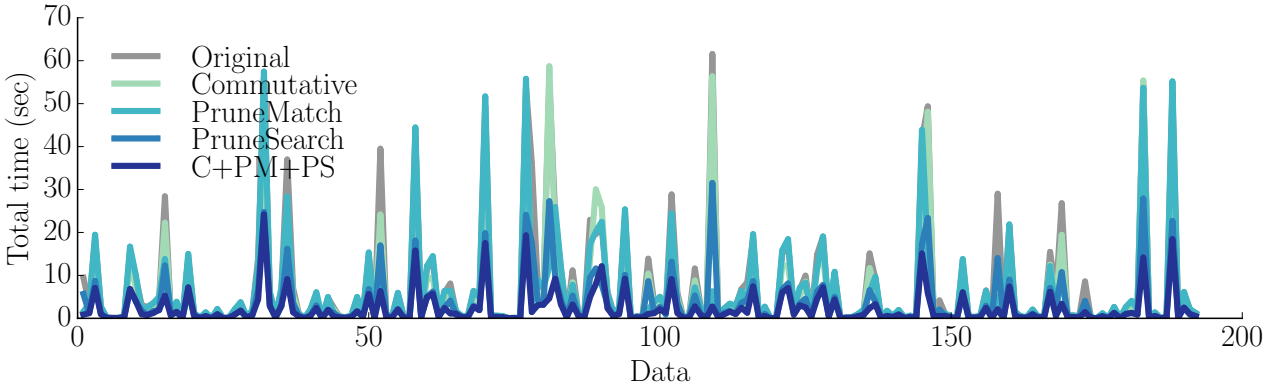
Constraints are widely used in modeling semantic spatial information by determining the positions of objects in the physical world [Renz and Nebel, 2007]. Constraints are determined by features, which based on our corpus contain unary [12] , binary [13] , and n-ary [14] features. We develop a set of constraints accordingly based on these features.

*Unary Absolute Qualitative Constraint* A unary absolute qualitative constraint represents the unary qualitative features on a single object. *Color constraint* defines the color of an object, *e.g.*"green", "yellow", "blue", "orange". *Isolation constraint* defines whether an object is surrounded by other objects or "isolated". *Absolute position constraint* defines the absolute position of an object viewing from the frame of the tabletop, *e.g.*

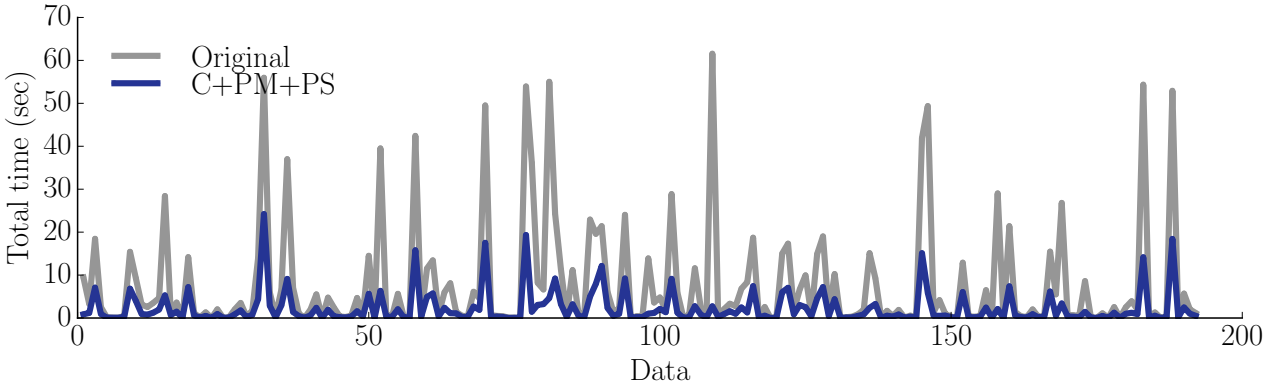
[12] Unary feature: an object property or visual feature, represented as a tuple <attribute,value>. For example, in a unary feature "a red block", the attribute is "color" and the value is "red".

[13] Binary feature: a spatial relation between two objects, represented as a tuple <attribute,value>. For example, in a binary feature "the block is far from another block", the attribute is "distance" and the value is "far".

[14] N-ary feature: a spatial relation between more than two objects, represented as a tuple <attribute,value>. For example, in an n-ary feature "the three blocks in a line", the attribute is "shape" and the value is "line".




(a) The comparison between the effects of applying different techniques on the total running time to generate RE's for all the scenes as shown in Fig. 2.2.




(b) The effect of applying all the techniques on the total running time to generate RE's for all the scenes as shown in Fig. 2.2.

Figure 3.12: The effect of applying different techniques on the total running time. "Original" = brutal force algorithm; "PruneSearch" = the algorithm after speeding up the search process; "PruneMatch" = the algorithm after speeding up the isomorphism process; "Commutative" = the algorithm after reducing the size of the REG graph.

"on the left/right/top/bottom half of the table" and "on the center of the table."

*Binary Relative Qualitative Constraint* A binary relative qualitative constraint represents the *qualitative spatial relation* between two objects. We use spatial relations instead of absolute $(x, y)$ coordinates for the ease of natural language generation because people identify target objects by referring to *landmarks* via spatial relations, *e.g.* "the target object is on the left to the green block" or "far from the red block." Qualitative spatial relations consist of orientation and distance [15,16] [Renz and Nebel, 2007]. *Qualitative distance* qualitatively describes the distance between two objects, *e.g.* "A is touching B", "A is close/far from B". *Qualitative relative orientation* qualitatively describes the quaternary direction between two objects, *e.g.* "A is in front of B","A is behind B","A is on the left/right to B"

*Binary Relative Quantitative Constraint* An binary relative qualitative constraint represents a low-level feature - *quantitative spatial relation* which

[15] We include orientation and distance as spatial relations because orientation and distance interact with each other, *e.g.* distance cannot usually be summed unless they are in the same direction [Cohn and Renz, 2008].

[16] Topology is the third type of spatial relation according to Renz and Nebel [2007]. But we don't consider topology because we treat each object as point particle while topological relations are usually described between spatial regions rather than points.

quantitatively states the relative position between two objects. Similar to qualitative spatial relations, we categorize quantitative spatial relations based on whether the relation refers to distance or orientation. *Quantitative distance* states the precise distance between two objects, $e.g.$ "object 1 is 2 cm away from object 2" as shown in Fig. 3.13(c). *Quantitative relative orientation* quantitatively describes the direction of the vector from the landmark object to the target object, $e.g.$ "object 1 is $60°$ north of east in the view of object 2" as shown in Fig. 3.13(c).

*N-ary Relative Qualitative Constraint* An n-ary relative qualitative constraint represents a container [Paul et al., 2016] of more than two objects, all of which share some common object properties or similar positions, $e.g.$, "a cluster/pair of objects", "a string/column/stack/row of objects", and "objects form a diamond/rectangle/square/triangle."

*Hierarchical REG Structure* We propose to model referring expressions as constraints in a scene $(O, C)$, in which $O$ is a set of objects, $C$ is a set of constraints over $O$. $C$ includes object properties, $i.e.$ unary constraints, denoted as $C^1$ and object relations, $i.e.$ binary constraints denoted as $C^2$, and n-ary constraints denoted as $C^n$. We build a labeled directed multigraph to support semantic expressiveness, where each node represents an object and each edge represents a constraint. This graph is similar to the REG graph which only supports unary absolute qualitative constraints and binary relative qualitative constraints [Krahmer et al., 2003]. To incorporate n-ary relative qualitative constraints and binary relative quantitative constraints as we mentioned previously, we will extend the current REG graph to a semantic hierarchy [Kuipers, 2000]. This is inspired by human cognitive map, which separates spatial information to power the flexibility and robustness in expression [Kuipers, 2000].

Our world model is a hierarchy $G = \{G^M, G^S, G^A\}$, which consists of three layers. Each layer is a labeled graph denoted as $G(V, E, L_V, L_E, \varphi)$, where $V$ is a set of vertices; $E \subseteq V \times V$ is a set of edges; $L_V$ and $L_E$ are sets of vertex labels and edge labels respectively; and $\varphi$ is a label function that defines the mappings $V \to L_V$ and $E \to L_E$.

The three layers include the *quantitative layer* $G^M(V, E^M, L_V, L_E^M, \varphi^M)$, the *qualitative layer* $G^S(V, E^S, L_V, L_E^S, \varphi^S)$, and the *abstract layer* $G^A(V, E^A, L_V, L_E^A, \varphi^A)$. Each layer is a local graph structure, which defines a type of representation of objects in a world, similar to a REG graph [Krahmer et al., 2003]. Each vertex $v \in V$ is 1-1 mapped to an object $o \in O$ , denoted in Equ. 3.2. Each edge, self-loop, binary edge, or n-ary edge, $e \in E^M \cup E^S \cup E^A$ is 1-1 mapped to a constraint $c \in C$ , denoted in Equ. 3.2.

Vertex: $v \in V \Leftrightarrow o \in O$ (3.2)

Self-loop: $e(v_i) \in E^M \cup E^S \cup E^A \Leftrightarrow (v_i \rightarrow v_i)$ in graph $\Leftrightarrow c(o_i) \in C^1$

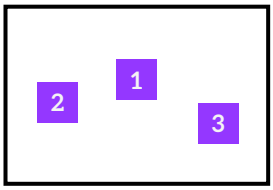
$e.g.$ In Fig. 3.13(e), $e(v_1) \Leftrightarrow c(o_1) \Leftrightarrow$ "$o_1$ is purple. "

Binary edge: $e(v_i, v_j) \in E^M \cup E^S \cup E^A \Leftrightarrow (v_i \rightarrow v_j)$ in graph $\Leftrightarrow c(o_i, o_j) \in C^2$

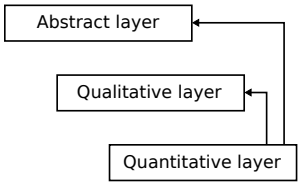
$e.g.$ In Fig. 3.13(e), $e(v_1, v_2) \Leftrightarrow c(o_1, o_2) \Leftrightarrow$ "$o_1$ is on the right to $o_2$."

N-ary edge: $e(v_i, ..., v_k) \in E^M \cup E^S \cup E^A \Leftrightarrow c(o_i, ..., o_k) \in C^n$ $(n > 2)$

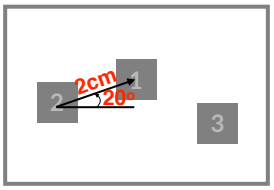
$e.g.$ In Fig. 3.13(f), $e(v_1, v_2, v_3) \Leftrightarrow c(o_1, o_2, o_3) \Leftrightarrow$ "$o_1, o_2, o_3$ are in a line."



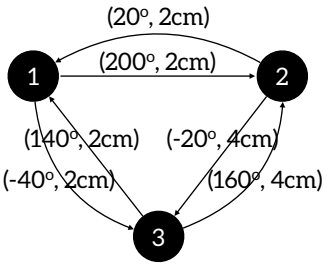

(a) Scene.



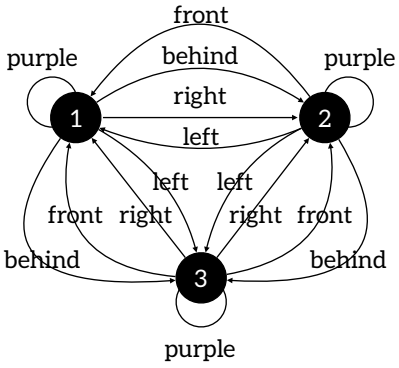

(b) Hierarchy.



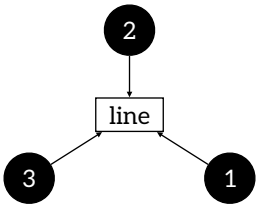

(c) Quantitative Spatial Relation.




(d) Quantitative layer of graph.




(e) Qualitative layer of graph.




(f) Abstract layer of graph.

Figure 3.13: We reason about spatial relations using a hierarchical structure as indicated in (b) where abstract and qualitative layers are initialized and updated by the quantitative layer. The three layers for the scene as shown in (a) are quantitative layer (d), qualitative layer (e), and abstract layer (f). In particular, each edge in the quantitative layer as shown in (d) represents both the direction and distance between the two objects based on the reasoning described in (c).

*Quantitative Layer* A quantitative layer is a graph $G^M(V, E^M, L_V, L_E^M, \varphi^M)$. Its binary edges $\subseteq E^M$ represent binary relative quantitative constraints $\subseteq C^2$, including quantitative distances, $e.g.$ "object 1 is 2 cm away from object 2" and quantitative relative orientations, $e.g.$ "object 1 is 180° north of east in the view of object 2". Because a quantitative relation represent a direction from the landmark object to the target object, $e.g.$ "the target object is 180° north of east in the view of the landmark object", the graph is directed. Because there could be multiple edges between the same pair of objects, the graph is defined as a multigraph. Fig. 3.13(d) is an example quantitative layer describing the scene in Fig. 3.13(a).

We need quantitative layer to integrate the ambiguities of the qualitative information and the metric precision of the quantitative information [Walter et al., 2014]. The quantitative layer stores low-level metric information that are invariant to perspective change, where $E^M$ represents binary relative quantitative constraints. Since there are so many different possible n-ary relative qualitative constraints, $e.g.$ "line", "triangle", "L shape", "circle", and so on, it would be inefficient if we preprocess all the n-ary constraints in the beginning and save them in the system. It would be better to dynamically reason about them and deduct them from other constraints. However, with only qualitative spatial relations, $e.g.$ "right" and "behind", it is sometimes impossible to reason and deduce n-ary constraints, $e.g.$ "triangle" and "line" as shown in Fig. 3.14. Therefore, The low-level quantitative information would support the initialization and update of a qualitative layer and an abstract layer.

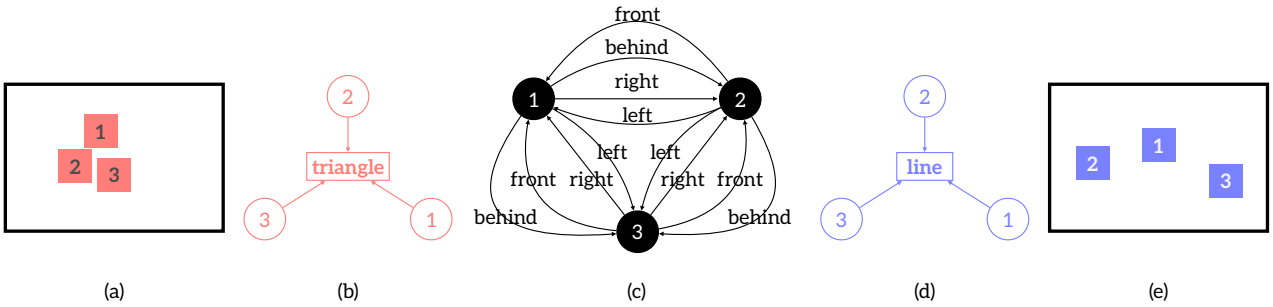



Figure 3.14: The example to illustrate that it is necessary to have a quantitative layer. (a) and (e) shows two scenes which have the same qualitative layer but different high level features and different abstract layers. In particular, (a) shows "the three objects which form a triangle", while (e) shows "the three objects in a line." Therefore, without a quantitative layer, qualitative layer alone cannot represent the gap between these two different scenes.

*Qualitative Layer* The qualitative layer is a graph $G^S(V, E^S, L_V, L_E^S, \varphi^S)$, which is similar to REG graph [Krahmer et al., 2003]. Its self-loops $\subseteq E^S$ represent unary absolute qualitative constraints $i.e.$ $C^1$, $e.g.$ "object 1 is red." Its binary edges $\subseteq E^S$ represent binary relative qualitative constraint $\subseteq C^2$, $e.g.$ "object 1 is on the left to object 2."

Because a qualitative relation represents a direction from the landmark object to the target object, $e.g.$ "the target object is on the left to the landmark object", the graph is directed. Because there could be multiple edges between the same pair of objects, such as distance and relation, the graph is defined as a multigraph. Fig. 3.13(e) is an example quantitative layer describing the scene in Fig. 3.13(a).

The qualitative relations could be extracted from the quantitative layer, $e.g.$ "A is far from B" could be deducted from "A is 10 meters far from B" plus a threshold.

*Abstract Layer* The abstract layer is a graph $G^A(V, E^A, L_V, L_E^A, \varphi^A)$. Its n-ary edges $E^A$ represent the n-ary relative qualitative constraints $\subseteq$

$C^n$, $e.g.$ "in a triangle", "in an L shape" or "in a cluster of nearby red blocks".

These abstract properties defines abstract groups of objects. It is a bipartite graph structure that represents how vertices (objects) belongs to different abstract groups.

The abstract properties are extracted from low-level information in a quantitative layer. For example, a 3-ary feature "a line of three objects" Fig. 3.13(f) could be deducted from the quantitative angles about $0°$ or $180°$ between each other in the quantitative layer as indicated in Fig. 3.13(d). But it is hard to deduct the 3-ary feature only from the qualitative layer Fig. 3.13(e).

## *3.4 Future Work*

In our implementation, all the visual features and spatial relations are predefined. But people are varied in defining spatial relations. For example, people without cars think that $100$ miles is a far distance but people with cars do not think so. How to exactly model spatial relations, $e.g.$ "close", "left to", "tilted" and "big", is critical to REG. We could further optimize clear referring expressions to be more human preferable by building models for these features based on our corpus, and actively updating the models by incorporating the individual variation in human-robot interactions.

In a real life, the scene could become very complex due to a large number of objects and a high similarity between these objects. For example in a table in the kitchen, there might be many spoons which look similar to each other placed randomly on the table. In this situation, the REG graph might become too complicated so it would take too much time to search for a clear RE.

One approach is to reduce the size of the REG graph by resolving a trade-off between graph efficiency and expressiveness. We are seeking for a minimum structure that supports expressing fully spatial relation reasoning. We could eliminate unnecessary constraints or edges to improve graph efficiency, while make sure that all the constraints have their representations in the graph. We attempt to apply binary edge commutativity to eliminate half of the binary edges in the graph. In addition to that, there are many graph structures we can use, $e.g.$ Relative Neighborhood Graph [Jaromczyk and Toussaint, 1992] and Delaunay Triangulation, which represent information efficiently, so that we only keep a minimum number of edges in the REG graph, but dynamically re-build the edges or constraints when necessary.

Another approach is to physically change the scene to make the scenarios easier for humans and robots to talk about. We can treat this referring process as a multi-modal interaction between the speaker and

the hearer. One mode is interactive dialog where a robot refers to a fuzzy region by saying "the blocks on the left." The human will usually follow the robot guidance and move their eyesight to focus on the blocks on the left. By capturing human feedback by eye trackers or microphones, the robot could predict the uncertainty in this human visual search task and produce further instructions in natural language to adapt to the human confusion by assigning probabilities into the REG graph. Another mode is physical manipulation, where the robot could manipulate the scene by removing clutters to make it easier for it to generate clear and understandable referring expressions. The robot could also move an object with a high saliency closer to the target object so that this object could act as a critical landmark for the robot to refer to. Another thought here is to make the robot swap an object with a low visual saliency, but close to the target object, and an object with a high saliency, but far from the target object, to increase the saliency around the target object.

# 4
# *Generating Explanations as Demonstrations*

[1]

Different robots optimize for different objective functions while satisfy different constraints, but not usually optimize for the ease of human understanding. Different robot reasoning might lead to the same robot behavior, which makes it difficult for humans to understand robot preferences, anticipate robot future plans, and adapt to robots in advance for safe [Alami et al., 2006a] and effective collaboration [Fisac et al., 2016]. Consider the trajectory shown in Fig. 4.1. It appears that the robot does its best to avoid rocks while navigating to the goal, implying it has a preference for traversing grassy states over rocky states. However, this trajectory could have also been generated by a robot with an objective function that has no preference for either terrain type if it arbitrarily chose where to turn. Similarly, a person observing the robot in Fig. 4.2 may be unclear about whether the robot has no terrain preference or a strong preference for grass.

In a system where language is not available, such as a noisy airport or a quiet library, to describe a task which is hard to be explained in natural language, such as tying our shoes, a robot could use demonstration-based explanations to help people understand the system and its task. prior work has focused on using robot motion to effectively convey robot capabilities and goals [Nikolaidis et al., 2017a, Dragan et al., 2013]. In contrast, we focus on using robot motion to convey its own objective function and show that it prefers to navigate through states with particular features.

We are interested in producing robot motion trajectories that help people understand the robot's feature preferences and that improve their ability to generalize that behavior to new environments. Based on the observation that people assign rational meaning to agent actions [Gergely et al., 1995, Dennett, 1989, Kamewari et al., 2005], we define two types of critical points in a trajectory, *inflection points* and *compromise points*, as points that are information-rich and convey information about the relationship between the planned trajectory and the features in the en-

[1] This work is done in collaboration with Rosario Scalise

vironment. Fig. 4.1 is an extreme example of how inflection points (*i.e.*, changes in direction) may lead an observer to infer preference for grass because the trajectory traverses only that terrain feature. The single rock compromise point in Fig. 4.2 may similarly lead an observer to believe there is no preference for grass over rocks when in fact all alternative paths have more rocks and therefore a lower overall value.

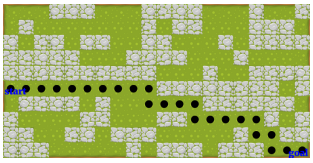


Figure 4.1: Many possible objective functions could generate this trajectory.

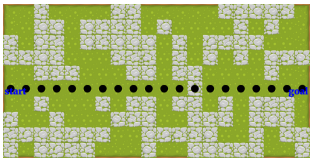


Figure 4.2: Many possible objective functions could generate this trajectory.

## 4.1 *Related Work*

Towards the goal of helping people accurately understand robot behavior, prior research has focused on ways to plan robot motion that is more interpretable or understandable to humans. Nikolaidis et al. [2017a] have contributed action planning algorithms that allow their robot to reveal its capabilities adaptively through a game theoretic model of human expectations. Sciutti et al. [2014], Zhou et al. [2017] have developed expressive robotic lifting motions to help humans understand the weights of the objects that robots are manipulating. The ease with which a person could recognize a robot's goals by observing its action execution improves robot legibility [Dragan et al., 2013, Dragan and Srinivasa, 2014], predictability [Fisac et al., 2016], acceptance [Cha et al., 2015], and naturalness [Szafir et al., 2014], which are important for human recognition of robot tasks [Zhang et al., 2016] and human-robot collaboration [Powers and Kiesler, 2006, Gielniak et al., 2013]. However, the prior works aim to make current executed behavior and goals more understandable and does not focus on helping people more easily predict future actions and generalize current behavior to new environments.

Our approach to making robot behavior more understandable is to communicate robot preferences for different states or state features (robot reward function) via robot actions. Inspired by the idea that people attribute decision-making at critical points in behaviors to rationality [Chajewska et al., 2000], we propose critical points along a trajectory that could be more informative than others about the robot's preferences. We analyze how these critical points in a trajectory affect a person's understanding of the robot's reward function by systematically creating demonstration trajectories with particular sets of points. The demonstrations (either in simulations like ours or real robots like Nicolescu and Mataric [2003]) motivate people to observe new robot behaviors and infer the robot's preferences [Zhang and Parkes, 2008]. We think demonstration as another medium of robot explanation, *i.e. demonstration-based explanation*, in addition to language-based explanation.

## 4.2 Problem Formulation

We formulate our robots' behaviors as a standard Markov Decision Process [Puterman, 2014] which is a tuple of the form: $\{\mathcal{S}, \mathcal{A}, \mathcal{T}, R\}$.

This includes a set of world states $s \in \mathcal{S}$ with a single absorbing goal state $s_g \in \mathcal{S}$ and a set of robot actions $a \in \mathcal{A}$. The MDP has a deterministic state transition function $\mathcal{T} : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$ and an immediate reward function $R : \mathcal{S} \rightarrow \mathbb{R}_+$. A robot behaves according to a deterministic policy $\pi : \mathcal{S} \rightarrow \mathcal{A}$. The optimal policy is denoted as $\pi^*$ and describes the policy that maximizes the overall reward.

A trajectory $\xi(s_0|\pi) \in \Xi$ is defined as a sequence of states $[s_0, s_1, s_2, ..., s_g]$ where $\forall s_t \in \xi(s_0|\pi)$, $\mathcal{T}(s_{t-1}, \pi(s_{t-1})) = s_t$. The total reward of $\xi$ is $R_\xi(\xi) = \sum_{s_t \in \xi} R(s_t)$. An optimal trajectory $\xi^*$ is yielded by following $\pi^*$.

To ensure there are no cycles in a trajectory, there is one and only one $s \in \mathcal{S}$ such that $R(s) \geq 0$.

### *Experimental Setup*

As an example domain, we consider a gridworld representation of a park which has a single terrain feature such as grass or rock assigned to each state (tile) on the grid.

- State $s \in \mathcal{S}$ is defined as $s = (x, y)$
- Action $a$ is a 4-connected movement where $a \in \mathcal{A} = \{\rightarrow, \uparrow, \downarrow, \leftarrow\}$
- We define $\phi : \mathcal{S} \rightarrow \mathbb{N}_+^3$ as a mapping from states to features. $\boldsymbol{\phi}(s) = [\mathbb{1}_{\text{goal}}(s), \mathbb{1}_{\text{grass}}(s), \mathbb{1}_{\text{rock}}(s)] \in \{0, 1\}^3$ subject to $\|\phi(s)\| = 1$, where each $\mathbb{1}(s)$ is an indicator function (e.g., $\mathbb{1}_{\text{grass}}(s) = 1$ if the tile type at $s$ is grass and $\mathbb{1}_{\text{grass}}(s) = 0$ otherwise)
- We define $\mathcal{T}$ as a transition mapping with deterministic 4-connected movements within the gridworld.
- $\boldsymbol{\theta} \in \mathbb{R}^3$ are the weights for the feature vector $\boldsymbol{\phi}$. The reward for a state $s$ with weights $\boldsymbol{\theta}$ is given by $R(\boldsymbol{\theta}, s) = \boldsymbol{\theta}^\intercal \boldsymbol{\phi}(s) \in \mathbb{R}$.
- When deriving the optimal policy, we break action ties with the ordering $[\rightarrow, \uparrow, \downarrow, \leftarrow]$.

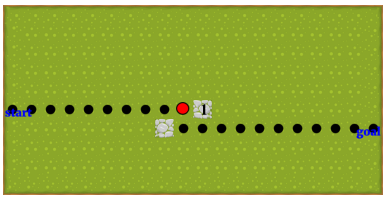


Figure 4.3: An inflection point indicated as the red dot (the black dot under the red dot is another inflection point).

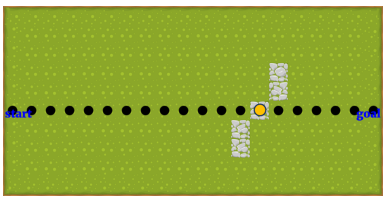


Figure 4.4: A compromise point indicated as the yellow dot.

### *Critical Points of Trajectories*

Depending on a robot's functional objective, the trajectory it follows can vary significantly. We characterize the information-rich states and actions within a trajectory as *critical points*. Based on the rationality principle, we focus on two types of critical points – *inflection points* in which people assign meaning to changing direction and *compromise points* in which a robot traverses over states with different features. Although

this set of characteristics is not exhaustive, we believe it provides an effective starting point in analyzing trajectories. We will demonstrate that critical points can be beneficial in guiding the observer's understanding of robot behavior, or they can be detrimental to an observer's understanding, confounding their beliefs and leading to misinterpretation.

### *Inflection Points*

Inflection points are defined as $s_t \in \xi(s_0|\pi)$ where the robot changes its direction. In other words, inflection points are all points at which the robot's action is not identical to its prior action, $i.e.$, $\pi(s_{t-1}) \neq \pi(s_t)$). In Fig. 4.3, an inflection point is indicated by the red dot where the robot moves down. This change of behavior gives people information about the robot's aversion towards the rock tile annotated as 1. In our park environment, inflection points come in pairs, $e.g.$, there is another inflection point indicated as the black dot right under the red dot in Fig. 4.3, because the robot typically resumes moving rightward after changing direction.

### *Compromise Points*

Compromise points are defined as states $s_t \in \xi^*(s_0|\pi^*)$ in which the myopic reward of entering $s_t$ is not the maximum obtainable from $s_{t-1}$, yet the total reward for the trajectory is maximized. In particular, $\exists\, a_{t-1} \in \mathcal{A}, a_{t-1} \neq \pi^*(s_{t-1}), \mathcal{T}(s_{t-1}, a_{t-1}) = s'_t.s.t.R(s'_t) > R(s_t)$, but $R_\xi(\xi^*(s'_t|\pi^*)) < R_\xi(\xi^*(s_t|\pi^*))$.

The trajectory in Fig. 4.4 contains one compromise point (orange dot). To reach the goal, the robot must traverse a terrain feature, $e.g.$ rock, which incurs a higher cost than another possible terrain feature (grass) accessible from the previous state. Any attempt to move around the rock frontier would result in lower total trajectory reward compared to the straight path over the one compromise point.

## 4.3 *Generating Demonstrations*

We develop a method for synthesizing trajectories through environments that demonstrate the robot's reward function $R(\boldsymbol{\theta}, s)$ by changing $\boldsymbol{\phi}$ by iteratively inserting inflection and compromise points into the trajectory $\xi^*$.

### *Inflection Points*

To create an inflection point at $s_i \in \xi^*(s_0|\pi^*)$, we can decrease the reward of $s_{i+1}$ which alters $\pi^*(s_i)$ to avoid $s_{i+1}$. In Fig. 4.5, grass is pre-

ferred and has lower cost than rock. To create an inflection point at $s_i$ indicated as the red dot, we place a rock terrain tile at $s_{i+1}$ annotated as state **1**.

One side effect of changing state **1** is that it might introduce multiple optimal policies yielding multiple optimal trajectories. The ambiguity of multiple optimal trajectories (or policies) can mislead people as it requires more complex reasoning to identify. One solution is to change some states to make all but one of the optimal trajectories sub-optimal. We treat this as a set cover problem. Universe $\mathbb{U}$ is the set of all the available optimal trajectories $\mathbb{U} = \{\xi | \xi = \xi^* \leftarrow \pi^*\}$. $\forall$ state $s \in \xi \in \mathbb{U}$, we define $\mathbb{S}(s) \subseteq \mathbb{U}$ to include all the optimal trajectories that go through $s$, *i.e.*, $\mathbb{S}(s) = \{\xi | s \in \xi \in \mathbb{U}\}$. The family set contains all the $\mathbb{F}(s)$, *i.e.*, $\mathbb{F} = \{\mathbb{S}(s) | s \in \xi \in \mathbb{U}\}$ $s.t.$ $\bigcup_{ss \in \mathbb{F}(s)} ss = \mathbb{U}$.

Our goal is to find the minimum number of states that all but one of the optimal paths go through, *i.e.*, to find the minimal set $\mathbb{C}$ of states subject to $\bigcup_{cc \in \mathbb{C}} \mathbb{S}(cc) = U \backslash \xi^{**}$ where $\xi^{**} \in U$ is the only path $s.t.$ $\forall cc \in cover, \xi^{**} \notin \mathbb{S}(cc)$. Then when we have a set $\mathbb{C}$, $\forall s \in \mathbb{C}$, we can reduce $R(s)$ to make all the $\xi \in \mathbb{S}(s)$ become sub-optimal and leave $\xi^{**}$ the only optimal trajectory.

In Fig. 4.5, there are 9 extra optimal trajectories available after changing states **1** (yellow arrows). By placing a rock terrain at state **2**, we could prevent the robot from moving downwards before reaching the red dot and make the trajectory indicated by black dots the only optimal trajectory. We generate 4 inflection points accordingly as shown in Fig. 4.6.

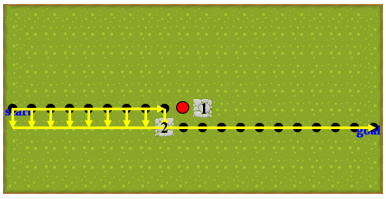



Figure 4.5: Generating 1 inflection point (red dot) by placing rock tiles at state 1 and 2.

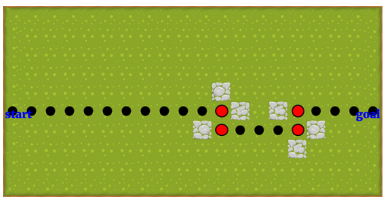



Figure 4.6: 4 inflection points (red dots).

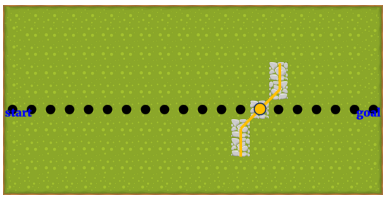



Figure 4.7: Generating 1 compromise point (orange dot) by building a frontier (orange line segments).

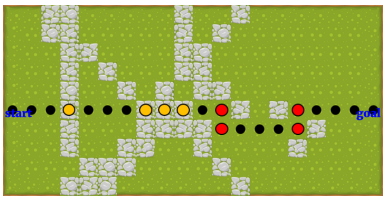



Figure 4.8: 4 compromise points (orange dots).

### *Compromise Points*

Similar to generating an inflection point, to generate a compromise point at $s_i \in \xi^*(s_0|\pi^*)$, we could decrease the reward of $s_{i+1}$ $s.t.$ $R(s_{i+1}) < R_{\text{max}}$. But the difference is that now we want robots to keep $\pi^*(s_i)$ and head to $s_{i+1}$ inevitably. Hence, we could decrease the rewards of a set of states in a neighboring area close to $s_{i+1}$ to make it too costly for robots to detour around $s_{i+1}$. We could initiate the area as one state and iteratively increase its size until the new optimal trajectory passes through $s_{i+1}$. In each iteration, we could grow the area by making all the optimal trajectories which do not go through $s_{i+1}$ become sub-optimal using the same technique we introduced in Sec. 4.3. Inflection Points.

In Fig. 4.7, to create an compromise point at $s_i$ (orange dot), we can build a frontier of states filled with rock terrain from top to bottom for a certain length (orange frontier). This frontier with low reward will force the robot to pass through $s_{i+1}$ and follow the trajectory indicated as the black dots. In our implementation, we use cubic Bezier curves [Farin, 2006] randomly generated through De Casteljau's Algo-

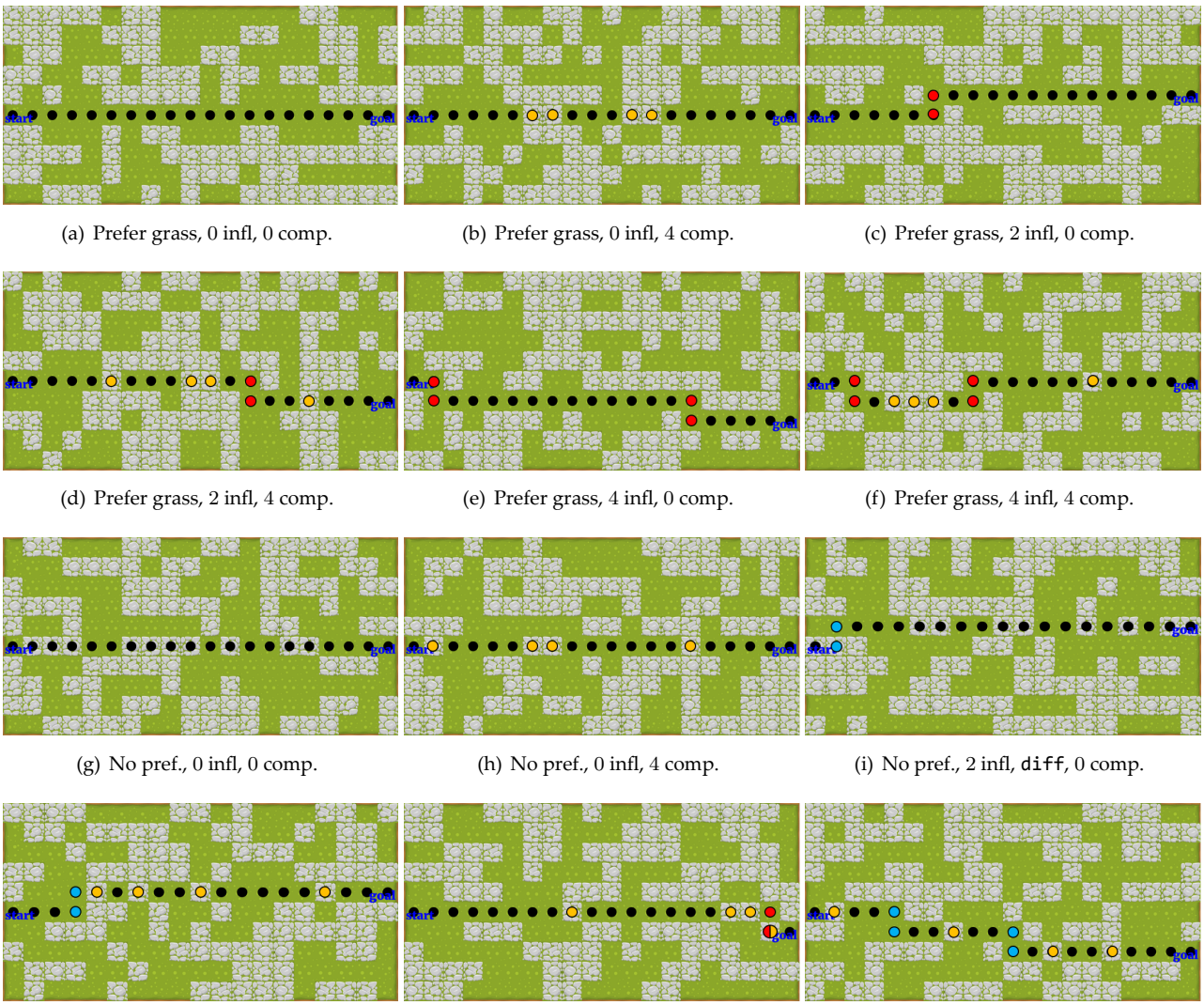
rithm [Farin, 2014] to represent natural-looking frontiers. We generate 4 compromise points accordingly as shown in Fig. 4.8.

*Extra Points*

We uniformly distribute different $\phi$'s across our demonstration maps to so that all the maps are consistent with each other regarding the frequency of each feature. In our implementation, we ensure that each map contains $50\%$ rocks and $50\%$ grass adding complementary rock tiles to grass-dominant maps and vice versa. To make maps look natural, we place terrain types based on 2D Perlin noise [Perlin, 1985, Perlin and Hoffert, 1989, Perlin, 2002]. Final maps and trajectories are shown in Fig. 4.9.

*Generated Demonstrations*

We have used the algorithm in Sec. 4.3 to create the maps for our study. We show some of the maps in Fig. 4.9. Note that we did not show the maps where the robots prefer rock because we can get the maps where the robots prefer rocks by replacing all the rocks with grass and all the grass with rocks in maps where the robots prefer grass.



(a) Prefer grass, 0 infl, 0 comp.

(b) Prefer grass, 0 infl, 4 comp.

(c) Prefer grass, 2 infl, 0 comp.

(d) Prefer grass, 2 infl, 4 comp.

(e) Prefer grass, 4 infl, 0 comp.

(f) Prefer grass, 4 infl, 4 comp.

(g) No pref., 0 infl, 0 comp.

(h) No pref., 0 infl, 4 comp.

(i) No pref., 2 infl, `diff`, 0 comp.

(j) No pref., 2 infl, `diff`, 4 comp.

(k) No pref., 2 infl, `same`, 4 comp.

(l) No pref., 4 infl, `diff`, 4 comp.

Figure 4.9: Robot preference type, number of inflection points (red dots), inflection point configuration (when the robots have no preference, inflection points with `diff` configuration = blue dots, inflection points with `same` configuration = red dots), number of compromise points (orange dots) for demonstration examples.

# 5
# *Evaluating Explanations as Demonstrations*

[1]

We ran a study to test the effects of the demonstration-based explanations in Sec. 4 with different critical points along trajectories on human understanding of robot terrain preferences. Our goal is to determine, in detail, the roles critical points play in trajectories that lead to good understanding of robot behavior. Towards this, we conducted a large-scale study to systematically examine how varying the critical points in trajectories affects peoples' understandings of robot behaviors.

We generated trajectories through synthetic environments according to different robot behaviors and showed them to people via Amazon Mechanical Turk. We conducted a within-subjects study in which we varied the parameterizations of the robot's reward function as well as the combinations of critical points along each trajectory and asked people to specify their understandings as well as generalize new plans in different environments. We show that people understand and can generalize the robot's terrain preferences more accurately as the number of inflection points increases and compromise points decreases within trajectories. However, when a robot has no preference for terrain types, the addition of either type of critical point within a trajectory reduces a participant's understanding.

We conclude that our critical points in trajectories do provide observers more information about a robot's state preferences. A robot that can take these points into consideration while planning its trajectories can reduce observer uncertainty about its behavior while still acting optimally.

[1] This work is done in collaboration with Rosario Scalise

## *5.1 User Study*

### *Independent Variables*

We tested six terrain preference conditions and ten no-preference conditions. The six preference conditions comprise all combinations of $\{0, 2, 4\}$

inflection and $\{0, 4\}$ compromise points. The no-preference conditions are combinations of $\{0, 2, 4\}$ inflection points, $\{0, 4\}$ compromise points, and {`same`, `different`} inflection point configurations [2].

[2] When there are no inflection points, there are no inflection point configurations, hence there are 10 'no preference' maps instead of 12.

*Terrain Preferences* We compared trajectories through maps when there is a terrain feature preference versus when there was no preference between terrain features. We randomly selected half of the terrain preference conditions to prefer rock and half to prefer grass. For example,

- In the demonstrations Fig. 4.9(a), Fig. 4.9(b), Fig. 4.9(c), Fig. 4.9(d), Fig. 4.9(e), Fig. 4.9(f), the robots prefer grass.
- We do not show the maps where the robots prefer rock because they are symmetric with the maps where the robots prefer grass. We can switch grass with rocks to get a map with the other preference.
- In the demonstrations, Fig. 4.9(g), Fig. 4.9(h), Fig. 4.9(i), Fig. 4.9(j), Fig. 4.9(k), Fig. 4.9(l), the robots have no preference.

*Inflection Points* Each demonstration trajectory had $0$, $2$, or $4$ inflection points. Locations of the inflection points were randomly chosen along the path.

- The demonstrations in Fig. 4.9(a), Fig. 4.9(b), Fig. 4.9(g), Fig. 4.9(h), have $0$ inflection points.
- The demonstrations in Fig. 4.9(d), Fig. 4.9(c), Fig. 4.9(i), Fig. 4.9(j), Fig. 4.9(k), have $2$ inflection points.
- The demonstrations in Fig. 4.9(e), Fig. 4.9(f), Fig. 4.9(l), have $4$ inflection points.

*Compromise Points* We set the number of compromise points in each demonstration trajectory to be one of two values. When the reward function had preferences, these two values were $\{0, 4\}$. We were interested in observing the differences between having no compromise points versus having several points where the robot must "make a compromise". The number of compromise points is $20\%$ of the total trajectory length. When the reward function had no preferences, compromises could not technically occur. Therefore, we arbitrarily assigned a "simulated" preference and then divided the number of terrain features along the trajectory in the two levels: $\{50 - 50, 20 - 80\}$. The former level resulted in a trajectory where there was no preference illustrated by compromise points. The latter resulted in a trajectory where the robot simulated a compromise on $20\%$ of the states.

- The demonstrations in Fig. 4.9(a), Fig. 4.9(c), Fig. 4.9(e), have $0$ compromise points when the robots have preferences.

- The demonstrations in Fig. 4.9(b), Fig. 4.9(d), Fig. 4.9(f), have $4$ compromise points when the robots have preferences.
- The demonstrations in Fig. 4.9(g), Fig. 4.9(i), have $0$ compromise points when the robots have no preferences.
- The demonstrations in Fig. 4.9(h), Fig. 4.9(j), Fig. 4.9(k), Fig. 4.9(l), have $4$ compromise points, $i.e.$ $20-80$ terrain type distribution along the trajectories, when the robots have no preferences.

*Inflection Point Configuration* At each inflection point, there is a 'decision' corresponding to the change in direction. The robot's direction switches from continuing onto one tile (B in both Fig. 5.1 and Fig. 5.2) to moving onto another tile (C in both Fig. 5.1 and Fig. 5.2). We test whether human understanding changes if the terrain types of those tiles are the `same`, $e.g.$, the robot chooses to turn from one grass tile to another grass tile, as shown in Fig. 5.1 or `diff`, $e.g.$, the robot turns from a grass tile onto a rock tile, as shown in Fig. 5.2. This condition is only tested when there is no preference in the terrain type and there are inflection points along the trajectories.

- The inflection points in Fig. 4.9(k), have `same` configuration when the robots have no preferences and there are inflection points along the trajectories.
- The inflection points in Fig. 4.9(i), Fig. 4.9(j), Fig. 4.9(l), have `diff` configuration when the robots have no preferences and there are inflection points along the trajectories.

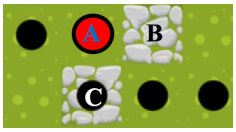


Figure 5.1: An inflection point with `same` configuration.

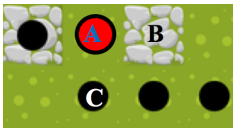


Figure 5.2: An inflection point with `diff` configuration.

*Response Types*

*Sliders* We included a slider for each terrain feature type and labeled them {"Strongly Avoid", "Slightly Avoid", "Neutral", "Slightly Prefer", "Strongly Prefer"}. We asked participants to indicate the preference the robot had demonstrated for each terrain type using the sliders. Participants were free to place the sliders anywhere along the scale. We mapped their slider placements to a value between $[0, 1000]$, where $0$ corresponds to "Strongly Avoid", $500$ corresponds to "Neutral", and $1000$ corresponds to "Strongly Prefer".

*Text Free-Response* Participants were asked to explain the reasoning they believe the robot used as it planned its path through the map. Unlike the sliders, free response allows an unconstrained representation of the users' mental models of the robot behaviors. Due to space constraints, we do not present the results from the free response.

*Drawing Trajectories* Last, we presented the participants with a new map (without a demonstration trajectory pre-drawn on it) and asked them to draw the trajectory they believed the robot would take if it were using the same reasoning to plan its new trajectory. Participants were required to start at a predefined point and could add 4-connected waypoints until reaching the goal position. Each map was generated to ensure it had a single optimal trajectory with respect to a fixed terrain preference. The maps were filled $50/50$ with rock and grass tiles. In order to reduce the bias in our test maps, each participant received a randomized test map for each experimental condition. This measure allowed us to test participants' understanding of the robot's behaviors by comparing their drawn path to the optimal one.

*Subjective Confidence* We asked participants to indicate on a $5$-point Likert scale how confident they were that the trajectory they drew would be the one the robot would take.

### *Dependent Variables*

Our measures of accuracy in understanding robot preferences are based on the drawn trajectories, sliders, and subjective ratings of confidence.

*Optimality Ratio* The *optimality ratio* $= |\frac{\text{total cost of optimal trajectory}}{\text{total cost of user drawn trajectory}}| \in [0, 1]$. As people understand the robot reward function more accurately, the optimality ratio increases.

*Preference Range* We assume that people use the distance between the grass and rock slider placements to indicate their certainty about inferring the robot preferences. We map the distance between the grass and rock slider placements to *preference range* $\in [0, 2000]$. A value of 0 corresponds to the user inferring no preference between the grass and rock terrains while a value of 2000 corresponds to the user inferring a difference with a high certainty, regardless of what the robot actually prefers.

We map the user slider placement for grass $s_{grass}$ to $[0, 1000]$, the user placement for rock $s_{rock}$ to $[0, 1000]$. Then we define the *preference range* $p$ as in Equ. 5.1, so that in all the 3 conditions, the larger $p$ is, the more certainty the users have in their understandings of the robot preferences.

$$p = \begin{cases} s_{grass} - s_{rock} + 1000 \in [0, 2000] \text{ if the robot prefers grass} \\ s_{rock} - s_{grass} + 1000 \in [0, 2000] \text{ if the robot prefers rock} \\ 2 \times (1000 - |s_{grass} - s_{rock}| \in [0, 2000] \text{ if the robot has no preferences} \end{cases} \tag{5.1}$$

*Subjective Confidence* We use *subjective confidence* $\in \{1, 2, 3, 4, 5\}$ to represent the user's self-reported confidence in understanding robot reasoning, with higher values indicating more confidence.

### *Hypotheses*

*H1* When Robots have preferences, the more inflection points, the higher optimality ratio, preference range, and subjective confidence.

*H2* When Robots have preferences, the more compromise points, the lower optimality ratio, preference range, and subjective confidence.

*H3* When Robots have no preferences, the more inflection points, the lower optimality ratio, preference range, and subjective confidence.

*H4* When Robots have no preferences, the more compromise points, the lower optimality ratio, preference range, and subjective confidence.

*H5* When Robots have no preferences, the optimality ratio, preference range, and subjective confidence are lower when each inflection point has a `diff` configuration than when each inflection point has a `same` configuration.

### *Study Deployment*

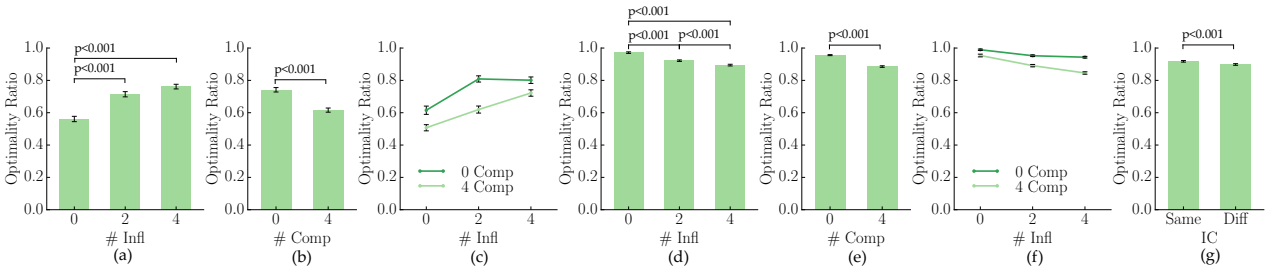
We recruited $90$ participants via Amazon Mechanical Turk. We used a within-subjects design where each subject was shown the total $16$ conditions $(6 + 10)$ in the same order. This order was pre-determined to ensure that no three consecutive conditions had the same terrain preference, which allowed us to avoid users inferring incorrectly based on coincidental patterns. Upon completion of the study, we collected demographic information from participants, including their age, gender, occupation, primary language, and experience with robots, video games, and RC-cars. We also asked for general comments as well as how difficult they found the tasks. Due to space constraints, these results are omitted.

## *5.2 Results*

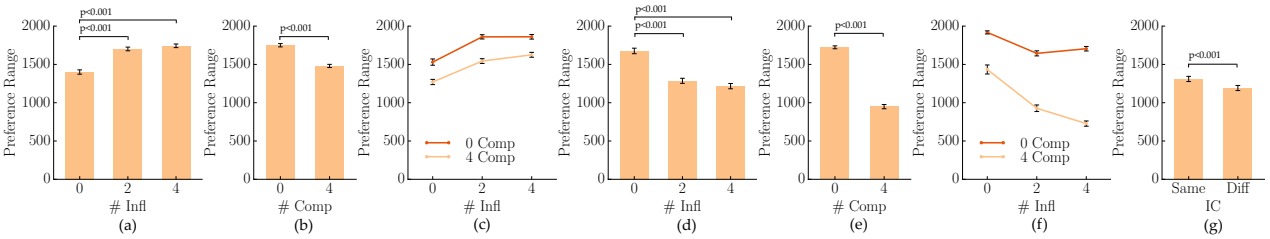
### *Preference*

*Optimality Ratio* We use a two-way repeated measures ANOVA to find the effect of inflection points and compromise points on optimality ratio (Table. 5.2).

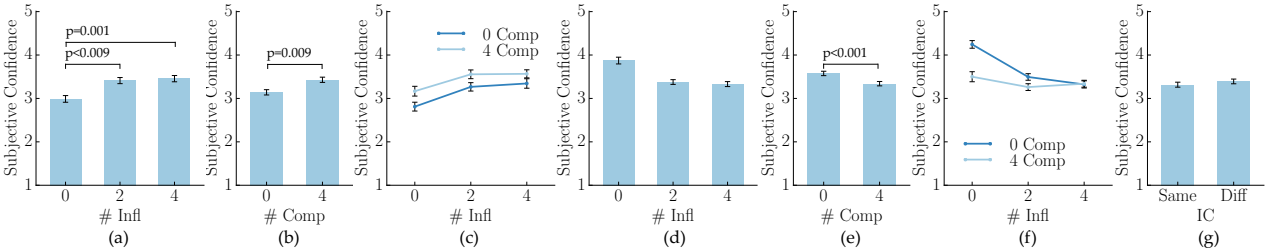
The number of inflection points has a significant effect on the optimality ratio $(F(2, 178) = 46.159, p < 0.001)$. Post hoc analysis with a Bonferroni adjustment reveals that the optimality ratio is significantly




I *Optimality ratio* vs different conditions.




II *Preference range* vs different conditions.




III *Subjective confidence* vs different conditions.

Figure 5.3: When there is a preference, *optimality ratio* / *preference range* / *subjective confidence* vs (a) the number of inflection points (b) the number of compromise points (c) the interaction between the number of inflection points and compromise points. When there is no preference,*optimality ratio* / *preference range* / *subjective confidence* vs (d) the number of inflection points (e) the number of compromise points (f) the interaction between the number of inflection points and compromise points (g) inflection point configuration

increased from 0 to 2 ($p < 0.001$) and from 0 to 4 ($p < 0.001$), but not from 2 to 4 inflection points ($p = 0.052$), though it is close (Fig. 5.3I(a)). This suggests that inflection points help users understand robot preferences. For example, it is easier for people to understand that the robot prefers grass over rock terrains by looking at Fig. 4.9(c) than Fig. 4.9(a). Additionally, in these maps there is little benefit to demonstrating the situation with more than 2 inflection points. For example, it is not much easier for people to understand that the robot prefers grass over rock terrains by looking at Fig. 4.9(e) than Fig. 4.9(c) although Fig. 4.9(e) has more inflection points. The first part of H1 is supported.

The optimality ratio is significantly decreased from 0 to 4 compromise points ($F(1, 89) = 74.476, p < 0.001$) (Fig. 5.3I(b)). For example, it is easier for people to understand that the robot prefers grass over rock terrains by looking at Fig. 4.9(c) than Fig. 4.9(d). The first part of H2 is supported.

There is a significant interaction between the numbers of inflection and compromise points on optimality ratio ($F(2, 178) = 5.291, p =$

$0.006$). When there are no compromise points, there is no significant difference between 2 and 4 inflection points ($p = 0.730$). However, when there are 4 compromise points, optimality ratio is significantly increased from 2 to 4 inflection points ($p = 0.001$) (Fig. 5.3I(c)). This indicates that as the number of compromise points increases, people need more inflection points to mitigate their confusion about the compromise points. For example, it is easier for people to understand that the robot prefers grass over rock terrains by looking at Fig. 4.9(f) than Fig. 4.9(d).

*Preference Range* We used a two-way repeated measures ANOVA to determine the effects of inflection points and compromise points on preference range (Table. 5.2).

The number of inflection points has a significant effect on preference range ($F(2, 178) = 65.759, p < 0.001$). A post hoc analysis with a Bonferroni adjustment reveals that the preference range is significantly increased from 0 to 2 ($p < 0.001$) and from 0 to 4 ($p < 0.001$), but not from 2 to 4 ($p = 0.385$) inflection points (Fig. 5.3II(a)). This suggests that more inflection points lead to greater certainty about the robot's preference. Similar to optimality ratio, increasing beyond 2 inflection points does not improve preference range. The second part of H1 is supported.

Preference range is also significantly decreased from 0 to 4 compromise points ($F(1, 89) = 91.050, p < 0.001$) (Fig. 5.3II(b)). The second part of H2 is supported.

There are no other significant effects on preference range.

*Subjective Confidence* To measure the effect of inflection and compromise points on the Likert scale responses for subjective confidence, we ran a generalized ordinal logistic model and estimated the model parameters through a generalized estimating equation (GEE) with AR(1) covariance structure (Table. 5.2).

Subjective confidence significantly increased from 0 to 2 ($p = 0.009$) and from 0 to 4 ($p = 0.001$), but not from 2 to 4 ($p = 0.907$) inflection points (Fig. 5.3III(a)). This suggests that inflection points help people feel more confident about their evaluations, but that increasing beyond 2 inflection points does not necessarily lead to more confidence. The third part of H1 is supported.

Subjective confidence is significantly increased from 0 to 4 compromise points ($p = 0.009$) (Fig. 5.3III(b)). Interestingly, the third part of H2 is rejected.

There are no other significant effects for subjective confidence.

| | Optimality Ratio | | Preference Range | | Subjective Confidence | |
|---|---|---|---|---|---|---|
| | 0 Comp | 4 Comp | 0 Comp | 4 Comp | 0 Comp | 4 Comp |
| 0 Infl | 0.62(0.24) | 0.51(0.18) | 1530(391) | 1271(324) | 2.81(0.97) | 3.17(1.07) |
| 2 Infl | 0.81(0.18) | 0.62(0.21) | 1860(265) | 1544(298) | 3.27(0.92) | 3.56(0.96) |
| 4 Infl | 0.80(0.19) | 0.72(0.19) | 1861(286) | 1626(304) | 3.34(1.04) | 3.57(0.89) |

Table 5.1: Mean (std. dev.) optimality ratio, preference range, and subjective confidence for preference maps

*No Preference*

Analysis for no preference maps follows the analysis for preference maps above. Results for inflection point configuration are only available for demonstrations with 2 or 4 inflection points, since 0 inflection points mean there cannot be inflection point configurations.

*Optimality Ratio.* We conducted a three-way repeated measures ANOVA to determine the effect of inflection points, compromise points, and inflection point configuration on optimality ratio (Table. 5.2).

The number of inflection points has a significant effect on optimality ratio ($F(2, 178) = 42.050, p < 0.001$). Post hoc analysis with a Bonferroni adjustment reveals that optimality ratio is significantly decreased from 0 to 2 ($p < 0.001$), from 0 to 4 ($p < 0.001$), and from 2 to 4 ($p < 0.001$) inflection points (Fig. 5.3I(d)). This suggests that people's ability to identify the robot's true preferences continues to decrease as inflection points are added. For example, it is easier for people to understand that the robot has no preference by looking at Fig. 4.9(g) than Fig. 4.9(i). The first part of H3 is supported.

Optimality ratio is significantly decreased from 0 to 4 compromise points ($F(1, 89) = 62.649, p < 0.001$) (Fig. 5.3I(e)), For example, it is easier for people to understand that the robot has no preference from Fig. 4.9(g) than Fig. 4.9(h). The first part of H4 is supported.

There is a significant interaction between the numbers of inflection and compromise points on optimality ratio ($F(2, 178) = 12.652, p < 0.001$). When the number of compromise points is high, the optimality ratio is significantly decreased from 2 to 4 inflection points ($p < 0.001$), while when number of compromise points is low, there is no significant difference ($p = 0.883$) (Fig. 5.3I(f)). This indicates that when there are many compromise points, more inflection points exacerbates the detrimental effect of compromise points on optimality ratio, while when the number of compromise points is low, the detrimental effect is gone.

Optimality ratio is significantly higher when inflection points have `same` configuration than when they have `diff` configuration ($F(1, 89) = 12.793, p = 0.001$) (Fig. 5.3I(g)). This indicates that for maps without a preference, inflection points that move to the same type of terrain better

reveal the robot's true (lack of) preference. For example, it is easier for people to understand that the robot has no preference by looking at Fig. 4.9(k) than Fig. 4.9(j). The first part of H5 is supported.

No other significant results were found.

| | 0 Compromise | | 4 Compromise | |
|---|---|---|---|---|
| | Same | Different | Same | Different |
| 0 Inflection | 0.99 (0.04) | 0.99 (0.04) | 0.95 (0.08) | 0.95 (0.08) |
| 2 Inflection | 0.95 (0.06) | 0.95 (0.07) | 0.91 (0.10) | 0.87 (0.10) |
| 4 Inflection | 0.95 (0.07) | 0.93 (0.08) | 0.86 (0.11) | 0.83 (0.09) |

Table 5.2: Mean (std. dev.) optimality ratio for no preference maps

*Preference Range.*

We use a three-way repeated measures ANOVA to determine the effect of the number of inflection points, compromise points, and inflection point configuration on preference range (Table. 5.2).

The number of inflection points has a significant effect on preference range ($F(2, 178) = 67.728, p < 0.001$). Post hoc analysis with a Bonferroni adjustment reveals that preference range is significantly decreased from 0 to 2 ($p < 0.001$) and from 0 to 4 ($p < 0.001$), but not from 2 to 4 inflection points ($p = 0.069$) (Fig. 5.3II(d)). The second part of H3 is supported.

Preference range is also significantly decreased from 0 to 4 compromise points ($F(1, 89) = 181.118, p < 0.001$) (Fig. 5.3II(e)). The second part of H4 is supported.

There is a significant interaction between the numbers of inflection points and compromise points on preference range ($F(2, 178) = 18.848, p < 0.001$). When there are 4 compromise points, preference range is significantly decreased from 2 to 4 inflection points ($p = 0.003$), while when there are 0 compromise points, there is no significant difference ($p = 0.611$) (Fig. 5.3II(f)). This indicates that inflection points have a detrimental effect on preference range only when they are exacerbated by compromise points, but that without the compromise points there is no detrimental effect.

Preference range was significantly decreased from `same` to `diff` inflection point configuration ($F(1, 89) = 13.802, p < 0.001$) (Fig. 5.3II(g)). This indicates that for maps without a preference, the preference range is lower when all inflection points have the `diff` configuration than when the same number of inflection points have the `same` configuration. The second part of H5 is supported.

No other significant differences are found.

| | 0 Compromise | | 4 Compromise | |
|---|---|---|---|---|
| | Same | Different | Same | Different |
| 0 Inflection | 1434 (563) | 1434 (563) | 1918 (194) | 1918 (194) |
| 2 Inflection | 1716 (395) | 1572 (482) | 989 (655) | 866 (485) |
| 4 Inflection | 1738 (351) | 1671 (436) | 797 (485) | 658 (428) |

Table 5.3: Mean (std. dev.) preference range for no preference maps

*Subjective Confidence.* To determine the effect of inflection points, compromise points, and inflection point configurations on subjective confidence, we conducted a generalized ordinal logistic model and estimated the model parameters through a generalized estimating equation (GEE) with AR(1) covariance structure (Table. 5.2).

There is no significant effect of inflection points on subjective confidence (Fig. 5.3III(d)). People are not significantly less confident about inferring the robot reasoning when dealing with demonstrations with more inflection points. The third part of H3 is rejected.

Subjective confidence is significantly decreased from 0 to 4 compromise points ($p < 0.001$) (Fig. 5.3III(e)). People are less confident about the robot's reasoning when dealing with demonstrations with more compromise points. The third part of H4 is supported.

There were no significant effects of inflection point configuration on subjective confidence (Fig. 5.3III(g)). The third part of H5 is rejected.

| | 0 Compromise | | 4 Compromise | |
|---|---|---|---|---|
| | Same | Different | Same | Different |
| 0 Inflection | 4.24 (0.84) | 4.24 (0.84) | 3.50 (1.10) | 3.50 (1.10) |
| 2 Inflection | 3.40 (1.04) | 3.59 (0.99) | 3.26 (1.13) | 3.27 (0.97) |
| 4 Inflection | 3.24 (1.13) | 3.40 (1.09) | 3.37 (1.09) | 3.31 (1.06) |

Table 5.4: Mean (std. dev.) subjective confidence for no preference maps

## 5.3 *Future Work*

People derive expectations about robot behavior by observing robot trajectories. Our work serves as a basis for enabling robots to use trajectories to convey information about their reward functions. In this work, we introduce the concept of critical points and give two examples, inflection points and compromise points. Using these, we develop a method for systematically generating trajectories that possess the critical points we specify. We then test how trajectories with varying combinations of critical points affect human understanding of robot reward functions. We show that inflection points can have different ef-

fects on human understanding depending on whether a robot's reward function has particular terrain feature preferences or not. Specifically, when there is a preference for terrain features, adding inflection points improves human understanding, while when there is no preference, adding inflection points hinders understanding. In both cases, increasing the number of compromise points decreases human understanding of the robot's preferences.

Interestingly, our results showed that the subjective confidence did not increase with fewer compromise points as we expected. Future work is needed to understand why this is the case. For example, it is possible that if participants never saw the robot navigate over a rock, they would not be confident about what would happen if it *had* to navigate over a rock.

Additionally, our results showed that there was a significant effect of one pair of inflection points but no benefit to the second pair of inflection points suggesting that there is a "law of diminishing returns" in information conveyed by inflections. Because we only investigated two terrain types, one pair of inflection points is all that is necessary to indicate which terrain type is preferred. More work is needed to investigate whether our finding holds for more complex environments. For example, while we believe that one inflection point is needed to show relative preference between pairs of features, it is unclear whether the complexity of the path will overwhelm an observer rather than help them.

Finally, our study was performed in an online study and not on a real robot. We acknowledge that it may be difficult to modify real environments in order for optimal trajectories to include critical points. In environments where a real robot cannot demonstrate its reward function by adding inflection points, for example, it may be possible for the robot to display a simulated environment with a trajectory, such as those we generated, to efficiently teach an observer about its preferences. Another option may be to demonstrate a non-optimal path that has more critical points to resolve a tradeoff between trajectory efficiency and ease of human understanding. Future work is needed to understand whether our findings translate to real robots in real environments, and also whether other methods of demonstration are effective.

# 6
# *Conclusion and Future Work*

People observe robot behaviors, understand robot intentions and preferences, and anticipate their future behaviors. However, as robots become more and more complicated but still rarely optimize their behavior for the ease with which humans understand them, it is more likely that humans would attribute biased reasons to robot behaviors and feel confused and surprised during collaboration. To make robot behavior more transparent and explicable, our approach is to enable robots to proactively give clear language-based explanations about their intentions and clear demonstration-based explanations about their preferences to humans.

In this thesis, we contribute a set of studies to collect and evaluate language-based explanations in tabletop manipulation tasks. We find that clear language-based explanations distinguish target objects from clutters via a set of salient visual features and spatial relations with perspective specified explicitly. To generate natural explanations, we enables robots to choose visual features and spatial relations based on the feature frequencies in our corpus. To make our generated explanations more varied, consistent, and extensible, we extend the graph structure for explanation generation from previous work to a hierarchical structure with both higher and lower levels of information for reasoning. To make our explanation generation more efficient, we leverage heuristics from graph isomorphism and constraint satisfaction problem to prune and guide our search. On the other hand, we introduce the idea of critical points along robot trajectories which are informative in helping people understand robot preferences in grid world navigation tasks. We fist propose an algorithm to generate explanations about robot preferences in the form of demonstration with specified numbers of critical points. We then verify the clarity of our generated demonstration-based explanations and find that some critical points are beneficial in conveying information about robot preferences while some critical points are harmful.

We have investigated both language-based and demonstration-based

explanations. Tradeoffs have to be resolved when choosing to use language or demonstrations. Language-based explanations are clearer for declarative information, more efficient in many environments except noisy settings, and more understandable because humans use languages a lot in social interaction. Demonstration-based explanations are clearer for procedural information, and could facilitate task completion and human understanding in parallel. For example, it is very hard to explain how to tie a shoe in natural language, while it is rather easy to demonstrate. Therefore, one future direction could be to investigate how to switch between these two types of explanations to effectively leverage both advantages.

From a psychology perspective, human process language-based explanations through the auditory channel and process demonstration-based explanations through the visual channel [Wuyts and Buekers, 1995]. Inspired from the dual processing in multimedia learning where reading could help listening [Mousavi et al., 1995, Moreno and Mayer, 2002], another prominent future direction is to combine language with demonstrations as a multi-modal explaining system.

# 7
# *Bibliography*


Alfred V. Aho and John E. Hopcroft. *The Design and Analysis of Computer Algorithms*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1st edition, 1974. ISBN 0201000296.

Rachid Alami, Alin Albu-Schäffer, Antonio Bicchi, Rainer Bischoff, Raja Chatila, Alessandro De Luca, Agostino De Santis, Georges Giralt, Jérémie Guiochet, Gerd Hirzinger, et al. Safe and dependable physical human-robot interaction in anthropic domains: State of the art and challenges. In *Proc. IROS*, pages 1–16. IEEE, 2006a.

Rachid Alami, Aurélie Clodic, Vincent Montreuil, Emrah Akin Sisbot, and Raja Chatila. Toward human-aware robot task planning. In *AAAI spring symposium: To Boldly Go Where No Human-Robot Team Has Gone Before*, pages 39–46, 2006b.

László Babai. Graph isomorphism in quasipolynomial time. *CoRR*, abs/1512.03547, 2015.

László Babai and Eugene M Luks. Canonical labeling of graphs. In *Proc. ACM symposium on Theory of computing*, pages 171–183. ACM, 1983.

Yonatan Bisk, Daniel Marcu, and William Wong. Towards a dataset for human computer communication via grounded language acquisition. In *Proc. AAAI Workshop on Symbiotic Cognitive Systems*, 2016.

S. N. Blisard and M. Skubic. Modeling spatial referencing language for human-robot interaction. In *Proc. IEEE International Workshop on Robot and Human Interactive Communication*, pages 698–703, 2005.

Adrian Bussone, Simone Stumpf, and Dympna O'Sullivan. The role of explanations on trust and reliance in clinical decision support systems. In *Proc. International Conference on Healthcare Informatics (ICHI)*, pages 160–169. IEEE, 2015.

Douglas M Campbell and David Radford. Tree isomorphism algorithms: speed vs. clarity. *Mathematics Magazine*, 64(4):252–261, 1991.

Laura Carlson, Marjorie Skubic, Jared Miller, Zhiyu Huo, and Tatiana Alexenko. Strategies for human-driven robot comprehension of spatial descriptions by older adults in a robot fetch task. *Topics in Cognitive Science*, 6(3):513–533, 2014.

Elizabeth Cha, Anca D Dragan, and Siddhartha S Srinivasa. Perceived robot capability. In *Proc. RO-MAN*, pages 541–548. IEEE, 2015.

Urszula Chajewska, Daphne Koller, and Ronald Parr. Making rational decisions using adaptive utility elicitation. In *Proc. National Conference on Artificial Intelligence*, pages 363–369, 2000.

Tathagata Chakraborti, Sarath Sreedharan, Yu Zhang, and Subbarao Kambhampati. Explanation generation as model reconciliation in multi-model planning. *arXiv preprint arXiv:1701.08317*, 2017.

Juan Chen, Anthony G Cohn, Dayou Liu, Shengsheng Wang, Jihong Ouyang, and Qiangyuan Yu. A survey of qualitative spatial representations. *The Knowledge Engineering Review*, 30(01):106–136, 2015.

Alasdair Daniel Francis Clarke, Micha Elsner, and Hannah Rohde. Where's wally: the influence of visual salience on referring expression generation. *Frontiers in psychology*, 4:329, 2013.

Eliseo Clementini and Paolino Di Felice. Approximate topological relations. *International Journal of Approximate Reasoning*, 16(2):173–204, 1997.

Anthony G. Cohn and Shyamanta M. Hazarika. Qualitative spatial representation and reasoning: An overview. *Fundamenta Informaticae*, 46 (1-2):1–29, 2001.

Anthony G Cohn and Jochen Renz. Qualitative spatial representation and reasoning. *Foundations of Artificial Intelligence*, 3:551–596, 2008.

Donatello Conte, Pasquale Foggia, Carlo Sansone, and Mario Vento. Thirty years of graph matching in pattern recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 18(03):265–298, 2004.

Sharolyn Converse. Shared mental models in expert team decision making. *Individual and Group Decision Making: Current issues*, 221, 1993.

Luigi P Cordella, Pasquale Foggia, Carlo Sansone, Francesco Tortorella, and Mario Vento. Graph matching: a fast algorithm and its evaluation. In *Proc. ICPR*, volume 2, pages 1582–1584. IEEE, 1998a.

Luigi P Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. Subgraph transformations for the inexact matching of attributed relational graphs. In *Graph Based Representations in Pattern Recognition*, pages 43–52. Springer, 1998b.

Luigi P Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. Performance evaluation of the vf graph matching algorithm. In *Proc. International Conference on Image Analysis and Processing*, pages 1172–1177. IEEE, 1999.

Luigi P Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. Fast graph matching for detecting cad image components. In *Proc. ICPR*, volume 2, pages 1034–1037. IEEE, 2000.

Luigi Pietro Cordella, Pasquale Foggia, Carlo Sansone, and Mario Vento. An improved algorithm for matching large graphs. In *Proc. IAPR-TC15 workshop on graph-based representations in pattern recognition*, pages 149–159, 2001.

Robert Dale. Cooking up referring expressions. In *Proc. Association for Computational Linguistics*, pages 68–75. Association for Computational Linguistics, 1989.

Robert Dale. *Generating referring expressions: constructing descriptions in a domain of objects and processes*. Bradford Books, 1992.

Robert Dale and Ehud Reiter. Computational interpretations of the gricean maxims in the generation of referring expressions. *Cognitive science*, 19(2):233–263, 1995.

Robert Dale and Jette Viethen. Referring expression generation through attribute-based heuristics. In *Proc. European Workshop on Natural Language Generation*, pages 58–65. Association for Computational Linguistics, 2009.

Kees van Deemter, Albert Gatt, Ielka van der Sluis, and Richard Power. Generation of referring expressions: Assessing the incremental algorithm. *Cognitive science*, 36(5):799–836, 2012.

Daniel Clement Dennett. *The intentional stance*. MIT press, 1989.

Anind K Dey. Explanations in context-aware systems. In *ExaCt*, pages 84–93, 2009.

Anca Dragan and Siddhartha Srinivasa. Integrating human observer inferences into robot motion planning. *Autonomous Robots*, 37(4):351–368, 2014.

Anca D Dragan, Kenton CT Lee, and Siddhartha S Srinivasa. Legibility and predictability of robot motion. In *Proc. HRI*, pages 301–308. ACM/IEEE, 2013.

Juraj Dzifcak, Matthias Scheutz, Chitta Baral, and Paul Schermerhorn. What to do and how to do it: Translating natural language directives

into temporal and dynamic logic representation for goal management and action execution. In *Proc. ICRA*, pages 4163–4168. IEEE, 2009.

Max J Egenhofer and Maria Vasardani. Spatial reasoning with a hole. In *Proc. International Conference on Spatial Information Theory*, pages 303–320. Springer, 2007.

Nikos Engonopoulos and Alexander Koller. Generating effective referring expressions using charts. *Proc. INLG and SIGDIAL*, page 6, 2014.

Hobart R Everett, Douglas W Gage, Gary A Gilbreath, Robin T Laird, and Richard P Smurlo. Real-world issues in warehouse navigation. In *Photonics for Industrial Applications*, pages 249–259. International Society for Optics and Photonics, 1995.

Rui Fang. *Referring expression generation towards mediating shared perceptual basis in situated dialogue*. PhD thesis, Michigan State University, 2014.

Rui Fang, Malcolm Doering, and Joyce Y Chai. Collaborative models for referring expression generation in situated dialogue. In *Proc. AAAI*, pages 1544–1550, 2014.

Gerald Farin. Class a bezier curves. *Computer Aided Geometric Design*, 23(7):573–581, 2006.

Gerald Farin. *Curves and surfaces for computer-aided geometric design: a practical guide*. Elsevier, 2014.

Thiago Castro Ferreira and Ivandré Paraboni. Referring expression generation: taking speakers' preferences into account. In *Proc. International Conference on Text, Speech, and Dialogue*, pages 539–546. Springer, 2014.

Jaime F Fisac, Chang Liu, Jessica B Hamrick, Shankar Sastry, J Karl Hedrick, Thomas L Griffiths, and Anca D Dragan. Generating plans that predict themselves. In *Proc. WAFR*, 2016.

Kerstin Fischer. The role of users? concepts of the robot in human-robot spatial instruction. In *Spatial Cognition V Reasoning, Action, Interaction*, pages 76–89. Springer, 2006.

Kerstin Fischer and Reinhard Moratz. From communicative strategies to cognitive modelling. In *Proc. Workshop Epigenetic Robotics*, 2001.

Nicholas FitzGerald, Yoav Artzi, and Luke S Zettlemoyer. Learning distributions over logical forms for referring expression generation. In *Proc. Conference on Empirical Methods on Natural Language Processing*, pages 1914–1925, 2013.

Maxwell Forbes, Rajesh PN Rao, Luke Zettlemoyer, and Maya Cakmak. Robot programming by demonstration with situated spatial language understanding. In *Proc. ICRA*, pages 2014–2020. IEEE, 2015.

Nancy Franklin, Barbara Tversky, and Vicky Coon. Switching points of view in spatial mental models. *Memory & Cognition*, 20(5):507–518, 1992.

Albert Gatt and Emiel Krahmer. Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *arXiv preprint arXiv:1703.09902*, 2017.

Albert Gatt, Emiel Krahmer, Kees Van Deemter, and Roger PG Van Gompel. Models and empirical data for the production of referring expressions. *Language, Cognition and Neuroscience*, 29(8):899–911, 2014.

György Gergely, Zoltán Nádasdy, Gergely Csibra, and Szilvia Bíró. Taking the intentional stance at 12 months of age. *Cognition*, 56(2): 165–193, 1995.

Michael J Gielniak and Andrea L Thomaz. Generating anticipation in robot motion. In *Proc. RO-MAN*, pages 449–454. IEEE, 2011.

Michael J Gielniak, C Karen Liu, and Andrea L Thomaz. Generating human-like motion for robots. *IJRR*, 32(11):1275–1301, 2013.

H Paul Grice. Logic and conversation. *1975*, pages 41–58, 1975.

Sergio Guadarrama, Lorenzo Riano, Dave Golland, Daniel Gouhring, Yangqing Jia, David Klein, Pieter Abbeel, and Trevor Darrell. Grounding spatial relations for human-robot interaction. In *Proc. IROS*, pages 1640–1647. IEEE, 2013.

John E Hopcroft and Jin-Kue Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In *Proc. ACM symposium on Theory of computing*, pages 172–184. ACM, 1974.

Thomas M Howard, Stefanie Tellex, and Nicholas Roy. A natural language planner interface for mobile manipulators. In *Proc. ICRA*, pages 6652–6659. IEEE, 2014.

Kai-yuh Hsiao, Soroush Vosoughi, Stefanie Tellex, Rony Kubat, and Deb Roy. Object schemas for responsive robotic language use. In *Proc. HRI*, pages 233–240. ACM, 2008.

Sumio Ishii, Shinji Tanaka, and Fumiaki Hiramatsu. Meal assistance robot for severely handicapped people. In *Proc. ICRA*, volume 2, pages 1308–1313. IEEE, 1995.

Amar Isli, Volker Haarslev, Ralf Möller, et al. *Combining cardinal direction relations and relative orientation relations in qualitative spatial reasoning*. University, Bibliothek des Fachbereichs Informatik, 2001.

Jerzy W Jaromczyk and Godfried T Toussaint. Relative neighborhood graphs and their relatives. *Proc. IEEE*, 80(9):1502–1517, 1992.

Shervin Javdani, Siddhartha Srinivasa, and J. Andrew (Drew) Bagnell. Shared autonomy via hindsight optimization. In *Proc. RSS*, Rome, Italy, July 2015.

Chuntao Jiang, Frans Coenen, and Michele Zito. A survey of frequent subgraph mining algorithms. *The Knowledge Engineering Review*, 28 (01):75–105, 2013.

Kazunori Kamewari, Masaharu Kato, Takayuki Kanda, Hiroshi Ishiguro, and Kazuo Hiraki. Six-and-a-half-month-old children positively attribute goals to human action and to humanoid-robot motion. *Cognitive Development*, 20(2):303–320, 2005.

Sahar Kazemzadeh, Vicente Ordonez, Mark Matten, and Tamara L Berg. Referitgame: Referring to objects in photographs of natural scenes. In *Proc. EMNLP*, pages 787–798, 2014.

Ross A Knepper, Todd Layton, John Romanishin, and Daniela Rus. Ikeabot: An autonomous multi-robot coordinated furniture assembly system. In *Proc. ICRA*, pages 855–862. IEEE, 2013.

Thomas Kollar, Stefanie Tellex, Deb Roy, and Nicholas Roy. Toward understanding natural language directions. In *Proc. HRI*, pages 259–266. IEEE Press, 2010.

Emiel Krahmer and Kees Van Deemter. Computational generation of referring expressions: A survey. *Computational Linguistics*, 38(1):173–218, 2012.

Emiel Krahmer, Sebastiaan Van Erk, and André Verleg. Graph-based generation of referring expressions. *Computational Linguistics*, 29(1): 53–72, 2003.

Benjamin Kuipers. The spatial semantic hierarchy. *Artificial intelligence*, 119(1-2):191–233, 2000.

Todd Kulesza, Simone Stumpf, Margaret Burnett, and Irwin Kwan. Tell me more?: the effects of mental model soundness on personalizing an intelligent agent. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pages 1–10. ACM, 2012.

Todd Kulesza, Simone Stumpf, Margaret Burnett, Sherry Yang, Irwin Kwan, and Weng-Keen Wong. Too much, too little, or just right? ways explanations impact end users' mental models. In *Proc. Visual Languages and Human-Centric Computing (VL/HCC)*, pages 3–10. IEEE, 2013.

Anagha Kulkarni, Tathagata Chakraborti, Yantian Zha, Satya Gautam Vadlamudi, Yu Zhang, and Subbarao Kambhampati. Explicable robot planning as minimizing distance from expected behavior. *arXiv preprint arXiv:1611.05497*, 2016.

Ayelet N Landau, Lisa Aziz-Zadeh, and Richard B Ivry. The influence of language on perception: listening to sentences about faces affects the perception of faces. *Journal of Neuroscience*, 30(45):15254–15261, 2010.

Pat Langley. Explainable agency in human-robot interaction. 2016.

Javier Larrosa and Gabriel Valiente. Constraint satisfaction algorithms for graph pattern matching. *Mathematical structures in computer science*, 12(04):403–422, 2002.

Willem JM Levelt. Perspective taking and ellipsis in spatial descriptions. *Language and Space*, pages 77–107, 1996.

Stephen C Levinson. Frames of reference and molyneux's question: Crosslinguistic evidence. *Language and space*, pages 109–169, 1996.

Shen Li, Rosario Scalise, Henny Admoni, Stephanie Rosenthal, and Siddhartha S Srinivasa. Spatial references and perspective in natural language instructions for collaborative manipulation. In *Proc. RO-MAN*, pages 44–51. IEEE, 2016.

Shen Li, Rosario Scalise, Henny Admoni, Siddhartha S Srinivasa, and Stephanie Rosenthal. Evaluating critical points in trajectories. In *Proc. RO-MAN*, 2017.

Christina Lichtenthäler, Tamara Lorenz, and Alexandra Kirsch. Towards a legibility metric: How to measure the perceived value of a robot. *ICSR work-in-progress-track*, 2011.

Brian Y Lim, Anind K Dey, and Daniel Avrahami. Why and why not explanations improve the intelligibility of context-aware intelligent systems. In *Proc. SIGCHI Conference on Human Factors in Computing Systems*, pages 2119–2128. ACM, 2009.

Changsong Liu and Joyce Yue Chai. Learning to mediate perceptual differences in situated human-robot dialogue. In *Proc. AAAI*, pages 2288–2294, 2015.

Jiming Liu. A method of spatial reasoning based on qualitative trigonometry. *Artificial Intelligence*, 98(1):137–168, 1998.

Eugene M Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. *Journal of Computer and System Sciences*, 25 (1):42–65, 1982.

Matt MacMahon, Brian Stankiewicz, and Benjamin Kuipers. Walk the talk: Connecting language, knowledge, and action in route instructions. In *Proc. National Conference on Artificial Intelligence*, 2006.

Jean MacMillan, M Paley, Eileen B Entin, and Elliot E Entin. Questionnaires for distributed assessment of team mutual awareness. *Handbook of human factors and ergonomics methods*, pages 51–1, 2004.

Junhua Mao, Jonathan Huang, Alexander Toshev, Oana Camburu, Alan L Yuille, and Kevin Murphy. Generation and comprehension of unambiguous object descriptions. In *Proc. CVPR*, pages 11–20, 2016.

Cynthia Matuszek, Dieter Fox, and Karl Koscher. Following directions using statistical machine translation. In *Proc. HRI*, pages 251–258. IEEE, 2010.

Cynthia Matuszek, Liefeng Bo, Luke Zettlemoyer, and Dieter Fox. Learning from unscripted deictic gesture and language for human-robot interactions. In *Proc. AAAI*, pages 2556–2563, 2014.

Brendan D McKay and Adolfo Piperno. Practical graph isomorphism, ii. *Journal of Symbolic Computation*, 60:94–112, 2014.

Brendan D McKay et al. Practical graph isomorphism. 1981.

Gary Miller. Isomorphism testing for graphs of bounded genus. In *Proc. ACM Symposium on Theory of Computing*, pages 225–235. ACM, 1980.

Dipendra Kumar Misra, Jaeyong Sung, Kevin Lee, and Ashutosh Saxena. Tell me dave: Context-sensitive grounding of natural language to manipulation instructions. In *Proc. RSS*, 2014.

Margaret Mitchell, Kees Van Deemter, and Ehud Reiter. Generating expressions that refer to visible objects. In *Proc. HLT-NAACL*, pages 1174–1184, 2013.

Reinhard Moratz. Representing relative direction as a binary relation of oriented points. In *Proc. ECAI*, volume 6, pages 407–411, 2006.

Reinhard Moratz and Thora Tenbrink. Spatial reference in linguistic human-robot interaction: Iterative, empirically supported development of a model of projective relations. *Spatial cognition and computation*, 6(1):63–107, 2006.

Reinhard Moratz, Kerstin Fischer, and Thora Tenbrink. Cognitive modeling of spatial reference for human-robot interaction. *International Journal on Artificial Intelligence Tools*, 10(04):589–611, 2001.

Roxana Moreno and Richard E Mayer. Verbal redundancy in multimedia learning: When reading helps listening. *Journal of Educational Psychology*, 94(1):156, 2002.

Seyed Yaghoub Mousavi, Renae Low, and John Sweller. Reducing cognitive load by mixing auditory and visual presentation modes. *Journal of Educational Psychology*, 87(2):319, 1995.

Jawad Nagi, H Ngo, Luca Maria Gambardella, and Gianni A Di Caro. Wisdom of the swarm for cooperative decision-making in human-swarm interaction. In *Proc. ICRA*, pages 1802–1808. IEEE, 2015.

Monica N Nicolescu and Maja J Mataric. Natural methods for robot task learning: Instructive demonstrations, generalization and practice. In *Proc. AAMAS*, pages 241–248. ACM, 2003.

Stefanos Nikolaidis, Swaprava Nath, Ariel D Procaccia, and Siddhartha Srinivasa. Game-theoretic modeling of human adaptation in human-robot collaboration. In *Proc. HRI*, 2017a.

Stefanos Nikolaidis, Yu Xiang Zhu, David Hsu, and Siddhartha Srinivasa. Human-robot mutual adaptation in shared autonomy. In *Proc. HRI*. ACM/IEEE, 2017b.

Ivandré Paraboni, Kees Van Deemter, and Judith Masthoff. Generating referring expressions: Making referents easy to identify. *Computational Linguistics*, 33(2):229–254, 2007.

R Paul, J Arkin, N Roy, and TM Howard. Efficient grounding of abstract spatial concepts for natural language interaction with robot manipulators. *Proc. RSS*, 2016.

Thomas Pechmann. Incremental speech production and referential overspecification. *Linguistics*, 27(1):89–110, 1989.

Vittorio Perera, Sai P Selveraj, Stephanie Rosenthal, and Manuela Veloso. Dynamic generation and refinement of robot verbalization. In *Proc. RO-MAN*, pages 212–218. IEEE, 2016.

Ken Perlin. An image synthesizer. *Proc. SIGGRAPH Computer Graphics*, 19(3):287–296, 1985.

Ken Perlin. Improving noise. In *Transactions on Graphics*, volume 21, pages 681–682. ACM, 2002.

Ken Perlin and Eric M Hoffert. Hypertexture. In *Proc. SIGGRAPH Computer Graphics*, volume 23, pages 253–262. ACM, 1989.

Aaron Powers and Sara Kiesler. The advisor robot: tracing people's mental model from a robot's physical attributes. In *Proc. HRI*, pages 218–225. ACM/IEEE, 2006.

Martin L Puterman. *Markov decision processes: discrete stochastic dynamic programming*. John Wiley & Sons, 2014.

Ehud Reiter and Robert Dale. Building applied natural language generation systems. *Natural Language Engineering*, 3(01):57–87, 1997.

Ehud Reiter, Robert Dale, and Zhiwei Feng. *Building natural language generation systems*, volume 33. MIT Press, 2000.

Jochen Renz and Bernhard Nebel. Qualitative spatial reasoning using constraint calculi. In *Handbook of spatial logics*, pages 161–215. Springer, 2007.

Stephanie Rosenthal, Sai P Selvaraj, and Manuela Veloso. Verbalization: Narration of autonomous robot experience. In *Proc. IJCAI*, pages 862–868. AAAI Press, 2016.

Nadine B Sarter and David D Woods. Team play with a powerful and independent agent: Operational experiences and automation surprises on the airbus a-320. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 39(4):553–569, 1997.

Matthias Scheutz, Paul Schermerhorn, and James Kramer. The utility of affect expression in natural language interactions in joint human-robot tasks. In *Proc. HRI*, pages 226–233. ACM, 2006.

Bill Schilit, Norman Adams, and Roy Want. Context-aware computing applications. In *Proc. WMCSA Workshop on Mobile Computing Systems and Applications*, pages 85–90. IEEE, 1994.

Douglas C Schmidt and Larry E Druffel. A fast backtracking algorithm to test directed graphs for isomorphism using distance matrices. *Journal of the ACM (JACM)*, 23(3):433–445, 1976.

Michael F. Schober. Spatial perspective taking in conversation. *Cognition*, 47:1–24, 1993.

Alessandra Sciutti, Laura Patane, Francesco Nori, and Giulio Sandini. Understanding object weight from human and humanoid lifting actions. *IEEE Transactions on Autonomous Mental Development*, 6(2):80–92, 2014.

Bastian Seegebarth, Felix Müller, Bernd Schattenberg, and Susanne Biundo. Making hybrid plans more clear to human users—a formal approach for generating sound explanations. In *Proc. International Conference on International Conference on Automated Planning and Scheduling*, pages 225–233. AAAI Press, 2012.

J.C. Simon. *Spoken Language Generation and Understanding*, volume 59. Springer Netherlands, 1980.

M. Skubic, D. Perzanowski, S. Blisard, A. Schultz, W. Adams, M. Bugajska, and D. Brock. Spatial language for human-robot dialogs. *Proc. SMC Part C (Applications and Reviews)*, 34(2):154–167, 2004.

Siddhartha S Srinivasa, Dave Ferguson, Casey J Helfrich, Dmitry Berenson, Alvaro Collet, Rosen Diankov, Garratt Gallagher, Geoffrey Hollinger, James Kuffner, and Michael Vande Weghe. Herb: a home exploring robotic butler. *Autonomous Robots*, 28(1):5–20, 2010.

Lionel Standing. Learning 10000 pictures. *The Quarterly Journal of Experimental Psychology*, 25(2):207–222, 1973.

Daniel Szafir, Bilge Mutlu, and Terrence Fong. Communication of intent in assistive free flyers. In *Proc. HRI*, pages 358–365. ACM/IEEE, 2014.

Leila Takayama, Doug Dooley, and Wendy Ju. Expressing thought: improving robot readability with animation principles. In *Proc. HRI*, pages 69–76. ACM, 2011.

Holly A Taylor and Barbara Tversky. Perspective in spatial descriptions. *Journal of memory and language*, 35(3):371–391, 1996.

Stefanie Tellex, Thomas Kollar, Steven Dickerson, Matthew R. Walter, Ashish G. Banerjee, Seth Teller, and Nicholas Roy. Understanding natural language commands for robotic navigation and mobile manipulation. In *Proc. AAAI*, pages 1507–1514, San Francisco, CA, August 2011.

Stefanie Tellex, Ross A Knepper, Adrian Li, Daniela Rus, and Nicholas Roy. Asking for help using inverse semantics. In *Proc. RSS*, volume 2, 2014.

Philip E Tetlock. Accountability: A social check on the fundamental attribution error. *Social Psychology Quarterly*, pages 227–236, 1985.

Michael Tomasello, Malinda Carpenter, Josep Call, Tanya Behne, and Henrike Moll. Understanding and sharing intentions: The origins of cultural cognition. *Behavioral and Brain Sciences*, 28(05):675–691, 2005.

J Gregory Trafton, Nicholas L Cassimatis, Magdalena D Bugajska, Derek P Brock, Farilee E Mintz, and Alan C Schultz. Enabling effective human-robot interaction using perspective-taking in robots. *Proc. SMC Part A (Systems and Humans)*, 35(4):460–470, 2005a.

J Gregory Trafton, Alan C Schultz, Magdalena Bugajska, and Farilee Mintz. Perspective-taking with robots: experiments and models. In *Proc. International Workshop on Robot and Human Interactive Communication*, pages 580–584. IEEE, 2005b.

Julian R Ullmann. An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 23(1):31–42, 1976.

Kees van Deemter, Ielka van der Sluis, and Albert Gatt. Building a semantically transparent corpus for the generation of referring expressions. In *Proc. International Natural Language Generation Conference*, pages 130–132. Association for Computational Linguistics, 2006.

Kees Van Deemter, Albert Gatt, Roger PG Van Gompel, and Emiel Krahmer. Toward a computational psycholinguistics of reference production. *Topics in cognitive science*, 4(2):166–183, 2012.

Jette Viethen and Robert Dale. The use of spatial relations in referring expression generation. In *Proc. the International Natural Language Generation Conference*, pages 59–67. Association for Computational Linguistics, 2008.

Jette Viethen, Margaret Mitchell, and Emiel Krahmer. Graphs and spatial relations in the generation of referring expressions. In *Proc. the European Workshop on Natural Language Generation*, pages 72–81, 2013.

Adam Vogel and Dan Jurafsky. Learning to follow navigational directions. In *Proc. Association for Computational Linguistics*, ACL '10, pages 806–814, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

Matthew R Walter, Sachithra Hemachandra, Bianca Homberg, Stefanie Tellex, and Seth Teller. A framework for learning semantic maps from grounded natural language descriptions. *The International Journal of Robotics Research*, 33(9):1167–1190, 2014.

Yuan Wei, Emma Brunskill, Thomas Kollar, and Nicholas Roy. Where to go: Interpreting natural directions using global inference. In *Proc. ICRA*, pages 3761–3767. IEEE, 2009.

Wikipedia. Binocular rivalry — wikipedia, the free encyclopedia, 2016a. URL https://en.wikipedia.org/w/index.php?title=Binocular_rivalry&oldid=756393747. [Online; accessed 5-May-2017].

Wikipedia. Referring expression — wikipedia, the free encyclopedia, 2016b. URL https://en.wikipedia.org/w/index.php?title=Referring_expression&oldid=720297786. [Online; accessed 24-April-2017].

Jeremy M Wolfe. Guided search 2.0 a revised model of visual search. *Psychonomic bulletin & review*, 1(2):202–238, 1994.

Irina J Wuyts and Martinus J Buekers. The effects of visual and auditory models on the learning of a rhythmical synchronization dance skill. *Research quarterly for exercise and sport*, 66(2):105–115, 1995.

Daqign Yi, Michael A Goodrich, and Kevin D Seppi. Informative path planning with a human path constraint. In *Proc. SMC*, pages 1752–1758. IEEE, 2014.

Daqing Yi, Michael A Goodrich, and Kevin D Seppi. Homotopy-aware rrt*: Toward human-robot topological path-planning. In *Proc. HRI*, pages 279–286. IEEE, 2016a.

Daqing Yi, Thomas M Howard, Michael A Goodrich, and Kevin D Seppi. Expressing homotopic requirements for mobile robot navigation through natural language instructions. In *Proc. IROS*, pages 1462–1468. IEEE, 2016b.

Haoqi Zhang and David C Parkes. Value-based policy teaching with active indirect elicitation. In *Proc. AAAI*, volume 8, pages 208–214, 2008.

Yu Zhang, Hankz Hankui Zhuo, and Subbarao Kambhampati. Plan explainability and predictability for cobots. *CoRR*, abs/1511.08158, 2015.

Yu Zhang, Sarath Sreedharan, Anagha Kulkarni, Tathagata Chakraborti, Hankz Hankui Zhuo, and Subbarao Kambhampati. Plan explicability for robot task planning. In *Proc. RSS Workshop on Planning for Human-Robot Interaction: Shared Autonomy and Collaborative Robotics*, 2016.

Allan Zhou, Dylan Hadfield-Menell, Anusha Nagabandi, and Anca D Dragan. Expressive robot motion timing. In *Proc. HRI*, 2017.