GoBot: An Autonomous Assistive Robot Using Behavior Trees to Encourage Child Mobility

AMEER HELMI*, Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University, USA
EMILY SCHEIDE*, Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University, USA
TZE-HSUAN WANG, Disability and Mobility Do-it-Yourself Co-Op, Oregon State University, USA
SAMUEL W. LOGAN, Disability and Mobility Do-it-Yourself Co-Op, Oregon State University, USA
GEOFFREY A. HOLLINGER, Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University, USA
NAOMI T. FITTER, Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University, USA

In early motor interventions from clinical rehabilitation to physical activity encouragement, one major challenge is maintaining child engagement and motivation. Robots show unique promise for addressing this challenge, but providing robots with new types of autonomous functionality is vital for promoting robot integration and usefulness in the clinic and home spaces. To provide needed autonomy capabilities for GoBot, our assistive robot for child-robot motion interventions, we propose a behavior tree framework. Within our framework, we build two trees: one manually designed based on expert knowledge of the child-robot interaction domain, and a second automatically synthesized and requiring minimal human input and time to construct. We tested each behavior tree with $N = 11$ children who interacted with GoBot during two behavior tree phases and a stationary-robot control phase. Our results show that both behavior tree phases tended to yield more child motion and significantly higher parent perception of child engagement, compared to the control phase. We showed that GoBot, equipped with our framework, has the potential to encourage movement and interaction in children, and that a synthesized tree can be competitive with a manually-designed tree. The products of this work can benefit researchers of behavior trees and child-robot interaction.

CCS Concepts: • **Computer systems organization → Robotic autonomy**; **Robotics**; • **Human-centered computing → User studies**.

ACM Reference Format:
Ameer Helmi*, Emily Scheide*, Tze-Hsuan Wang, Samuel W. Logan, Geoffrey A. Hollinger, and Naomi T. Fitter. 2025. GoBot: An Autonomous Assistive Robot Using Behavior Trees to Encourage Child Mobility. 1, 1 (February 2025), 17 pages. https://doi.org/XXXXXXX.XXXXXXX

*These authors contributed equally to this work

Authors' addresses: Ameer Helmi*, helmia@oregonstate.edu, Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University, 101 Covell Hall, Corvallis, Oregon, USA, 97331-2409; Emily Scheide*, scheidee@oregonstate.edu, Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University, 101 Covell Hall, Corvallis, Oregon, USA, 97331-2409; Tze-Hsuan Wang, wangtzeh@oregonstate.edu, Disability and Mobility Do-it-Yourself Co-Op, Oregon State University, 160 SW 26th St, Corvallis, Oregon, USA, 97331-2409; Samuel W. Logan, logansa@oregonstate.edu, Disability and Mobility Do-it-Yourself Co-Op, Oregon State University, 160 SW 26th St, Corvallis, Oregon, USA, 97331-2409; Geoffrey A. Hollinger, geoff.hollinger@oregonstate.edu, Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University, 101 Covell Hall, Corvallis, Oregon, USA, 97331-2409; Naomi T. Fitter, naomi.fitter@oregonstate.edu, Collaborative Robotics and Intelligent Systems (CoRIS) Institute, Oregon State University, 101 Covell Hall, Corvallis, Oregon, USA, 97331-2409.

Manuscript submitted to ACM

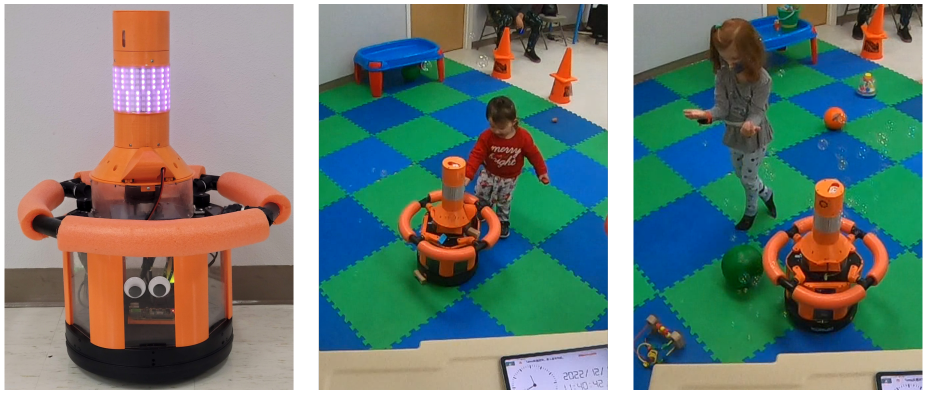

Fig. 1. *Left:* GoBot, our custom assistive robot with bubble, light, and sound stimulus hardware. *Center* & *Right:* GoBot autonomously interacting with participants during a session.

1 INTRODUCTION

In human development, an early and maintained focus on the encouragement of movement and play is critical, not only for building motor skills, but also in the shaping of interconnected social and cognitive skills [15, 29]. Interventions designed to facilitate motor development are particularly important for children with motor impairments and are often carried out through physical therapy methods such as treadmill training [49]. Despite the efficacy that can be achieved by these interventions, maintaining child engagement during sessions is a challenge for clinicians that limits their ability to focus on other vital aspects of an intervention. The same challenge exists in general physical activity-promoting scenarios with children; caregivers need reliable and engaging resources for motivating child motion. Accordingly, a promising new paradigm is the use of robots to assist with engagement during motion interventions, taking advantage of a robot's ability to consistently interact with children per each child's unique needs [46, 47]. Despite the potentially engaging and movement-eliciting behaviors that a robot can perform, modern assistive robots still require a high degree of input effort from the clinician. Unless the robot can successfully operate autonomously, the clinician is fully responsible for teleoperation, which requires training and limits their ability to focus on the child. In this work, we accordingly propose an integrated novel autonomous framework for our assistive robot, GoBot (see Fig. 1).

In robotics, a common way of representing autonomy is with a control architecture, examples of which range from the widespread finite state machine to the increasingly popular behavior tree. Behavior trees are accelerating in prevalence largely due to their notable advantages in recursivity, modularity, and readability [1, 8, 22, 23, 33, 42]. A behavior tree is a directed rooted tree that has internal control flow nodes that guide node activity, and leaf execution nodes that evaluate conditions or trigger actions. As condition node statuses change to reflect the current world state, the logic structure within a behavior tree functions by executing the appropriate action nodes. The advantages of behavior trees make them more feasible to manually design than other common control architectures. However, for those with minimal behavior tree experience, the manual design process can be time-intensive and difficult to validate. Therefore, to enable autonomous operation in GoBot, we manually design a behavior tree and also generate a behavior tree using a method of automatic behavior tree synthesis that we detail in Section 3.2. This synthesis method requires minimal human input and alleviates the need for an expert in a given domain and in behavior trees.

Our main research goal is *the design and evaluation of an effective behavior tree framework for an assistive robot that operates autonomously, promoting physical activity for children with typical development or motor disabilities*. We first provide an overview of related work in assistive robotics, behavior tree functionality, and automatic behavior tree generation methods in Section 2. In Section 3, we describe GoBot's capabilities (Section 3.1), then we present each of the behavior trees we designed and the behavior tree synthesis method we used to generate one of them (Section 3.2). We implemented these behavior trees on GoBot, and in Section 4, we detail the in-person within-subjects data collection evaluation we conducted with $N = 11$ children. This data collection aimed to initially answer if GoBot would tend to encourage similar amounts of child movement during each behavior tree phase, and if GoBot would tend to encourage more child movement during any behavior tree phase compared to a control phase without an active robot. Section 5 reports that each behavior tree performed similarly, and that there was a tendency for the children to move more and be more engaged while the robot was active. Finally, Section 6 discusses the implications of our results, strengths and limitations of this work, and future directions. The main contribution of this work is the integration of an automatic behavior tree synthesis method into an existing assistive robotic system, which provides autonomy to the assistive robot in the form of a behavior tree with the goal of encouraging child physical activity. This use of behavior-tree-based autonomy is novel for the child mobility domain.

2 RELATED WORK

Related work in assistive robotics and behavior trees guided our design choices and provided important context for decisions during our integration and testing efforts.

2.1 Assistive Robots

Robots have been used for a broad variety of assistive applications, ranging from interventions with older adults to support for young children. Some of the applications for assisting adults include post-stroke rehabilitation [35], physical activity coaching [18], and cognitive activity encouragement and memory assistance for people with dementia [13, 25]. For children, applications range from social skill practice for children with autism [5, 30] to eliciting communication, emotion, and motor skills in children with complex cerebral palsy [7]. We note that interactions between robots and children during studies tend to be short, often lasting 15 minutes or less [12, 31].

The majority of the efforts listed above relate to the potential benefits of using robots to encourage physical activity in adults and children. In our own work, we focus specifically on assisting children through robot-mediated interventions that encourage motor skill use and development. In past work in this space, a child with Down syndrome wore a body-weight support harness and both a NAO and a Dash robot with partial autonomy encouraged movement via light, sound, and movement-based actions [24]. In [2, 14], a NAO robot was used to teach and reinforce leg movement in infants via demonstrations and rewards from a robot. A Zeno robot used imitation to encourage children with autism spectrum disorder (ASD) to practice social motor behaviors such as waving hello and goodbye [3]. We conducted initial deployments with GoBot in prior work and found that children stood up more and were more engaged while the robot was active [47].

All of the above work shows promise for robots to encourage movement and engage with children, but most of the work has involved small sample sizes and limited-to-no autonomy. Autonomy for social robots has been studied in domains different to ours, so we reviewed child-robot studies with autonomous robots to serve as inspiration for our methods [6]. In [10], designing fully autonomous robots for child-robot interaction in unstructured and noisy environments proved technically infeasible. Accordingly, this work relied on partial autonomy, using behavior trees on a Pepper robot that engaged in educational games with children in a classroom setting. In another application outside of motion encouragement, a fully autonomous Haru robot guided by a behavior tree was used in a child-robot teaming task [4]. In our work, we use behavior trees to provide GoBot with autonomous capabilities for child motion encouragement, and we assess GoBot's ability to encourage child movement and engagement through in-person testing.

2.2 Behavior Trees

A control architecture is a structure that can represent autonomy by describing the logic that defines the relationship between behavioral responses and certain world states. There are a variety of common control architectures used in robotics applications, such as finite state machines [27], decision trees [26], and behavior trees [22]. Unlike other common architectures, behavior trees are built with respect to tasks rather than states. This difference, among other aspects of behavior tree functionality, translates to behavior trees having advantages over other architectures (e.g., readability, recursivity, modularity, and scalability) [1, 8, 22, 23, 33, 42]. These advantages have led to a notable increase in behavior tree popularity for robotics applications, and are also the reason we chose to employ behavior trees for our child mobility application in this work.

A behavior tree, as shown in Section 3.2, is a directed rooted tree with a variety of nodes that can either be categorized as control flow or execution nodes. Control flow nodes guide the depth-first flow of execution throughout the tree via a process called ticking. Classical control flow nodes include *sequence* nodes ("and" logic, denoted by $\rightarrow$), *fallback* nodes ("or" logic, denoted by ?), *parallel* nodes (concurrent activity, denoted by ||), and customizable *decorator* nodes such as the *not-decorator* (denoted by !), which returns the logical complement of a single node beneath it. Execution nodes are either *condition* nodes that describe components of the world state, or *action* nodes that execute robot behaviors. Any active node can return a status of *success* or *failure*, and all but *condition* nodes can return a status of *running*.

It is often more feasible to manually design a behavior tree than another control architecture for a given application. In Section 3.2, we show the behavior tree we designed manually for the child mobility domain, based on a small set of domain-specific data and our team's expert knowledge of child-robot interaction and behavior trees. However, despite the feasibility of this design, verification that such a manually-designed behavior tree will achieve the maximum possible reward is typically infeasible. Manual verification would require comparison to a large number of other behavior trees to be exhaustive, and this is not realistic, especially in the case that this validation is coupled with deployments with humans.

In an attempt to provide access to high-performing behavior trees with a minimized requirement of user input, methods of automatic behavior tree synthesis have been developed [1, 9, 11, 16, 28, 32, 34, 38, 39, 43]. These methods vary in user input requirements, including approaches like [38] that require and take advantage of a well-defined simulator, methods like [39] that only require specification of the given domain and problem but do not require a simulator, and methods like [16] that learn behavior trees from demonstration. Due to domain complexity and insufficient data, we do not have access to a simulator that represents our child mobility domain accurately enough for a method such as [38] or other comparable methods to learn a behavior tree that would perform well in the real world. The behavior tree synthesis method we use and detail in Section 3.2 solves the problems with manual design, is well suited for the child mobility domain, is generalizable, and does not require a simulator [39]. However, to date, there is a lack of in-person testing to demonstrate the efficacy of this method; we aim to address this gap in our current work.

3 SYSTEM DESIGN

In this section, we describe the design of the assistive robot and the associated sensing methods, as well as the design of our behavior tree framework for autonomous robot control.

Fig. 2. An overhead view from OverTrack while in use during a session. The thick black rectangle surrounding the play space identifies the boundary for tracking bounding boxes, the red-orange bounding box (right) marks the robot, and the green bounding box (left) marks the participant. The software communicates the centroid of each bounding box and the distance between them using ROS.

3.1 Assistive Robot Design

We previously designed GoBot together with experts in kinesiology and pediatric physical therapy with the goal of encouraging child movement through play. We determined that three robotic system requirements were needed to achieve this goal:

(1) Base mobility capable of 2D motion, either holonomic or non-holonomic
(2) Developmentally appropriate and modular reward hardware
(3) Real-time sensing that provides information about the location and movement of children in the play space

GoBot, as shown in Fig. 1, was built on a *TurtleBot2 base* (requirement 1) and is controlled by a Raspberry Pi 4 running the open-source Robot Operating System [36] (ROS) Noetic on Ubuntu 20.04. To ensure developmentally appropriate and modular hardware (requirement 2), we designed a *custom reward stack* which is fixed atop the base and includes:

- *Bubbles:* a module that blows large bubbles.
- *Lights:* an LED array with several light patterns in a variety of colors.
- *Sounds:* speakers that can play a library of engaging sounds.

All of the rewards can be controlled by an operator or activated autonomously. The reward stack is modular, meaning that reward modules can be replaced with new and appropriate rewards for different populations.

The GoBot sensing system includes an *RPLIDAR-A1 LIDAR sensor* for *obstacle avoidance* and *OverTrack, an overhead camera-based region-of-interest (ROI) tracking system [20]* which provided real-time data to the robot about the *position and movement of the child*, thus satisfying requirement 3. The LIDAR sensor calculates the distance between nearby objects and the robot at 10 Hz and prevents the robot from driving toward any object less than 0.5 ft (0.15 m) away. OverTrack, shown in Fig. 2, was validated for use as a real-time and post-hoc ROI analysis tool [20]. The system runs using an OpenCV ROI tracker which receives video from an overhead camera and outputs positional data for bounding boxes. At the beginning of a session, a researcher manually drew bounding boxes around the child and robot, and as the session progressed, the researcher corrected any tracking errors by re-drawing bounding boxes. The ROI tracker calculated the centroids of the bounding boxes and the Euclidean distance between them, and then sent the data to the robot using ROS in real time at 10 Hz. Distances were scaled using the 2 ft $\times$ 2 ft (0.61 m $\times$ 0.61 m) colored play mats in

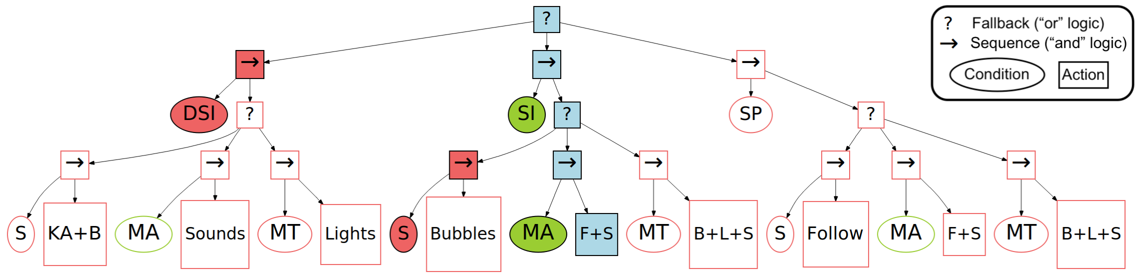

Fig. 3. The manually-designed behavior tree (M-BT), a 37-node behavior tree that we manually designed for our child-robot interaction domain. Behavior tree execution is from left to right, and can be visualized by the coloring of the nodes. Red denotes a node that has failed, green denotes a node that has succeeded, and blue denotes a node that is currently executing. Unshaded nodes are currently inactive, as they are currently not being evaluated or performed by the behavior tree. The behavior tree is shown to be running the follow and sounds (F+S) action, because it has determined that the child is at a social interaction distance (SI succeeded) and is moving away from the robot (MA succeeded). A full listing of the acronyms in this tree appears in the bulleted lists in Section 3.2.

the play space (shown in Fig. 2) as a reference. The robot calculated the angle between the vector from the center of the robot to the front of the robot and the vector from the center of the robot to the child. Next, the robot compared the current positional data against the previous child position to determine if the child was moving. The positional data generated from the overhead tracking system combined with the movement data calculated by the robot was used within the logic of the behavior tree to determine robot actions, as further described below.

3.2 Behavior Trees

We provided our robot with autonomy in the form of a behavior tree. In other words, a behavior tree was responsible for determining when the robot executed actions from a set we defined, using a set of conditions we also defined to describe world states. In Fig. 3, we show a behavior tree that we designed manually based on our combined expert knowledge of child-robot interaction and behavior trees. In Fig. 4, we show a second behavior tree, which we synthesized using the generalizable method detailed later in this section. We first define the robot's acting capabilities as a list of actions and the robot's sensing capabilities as a list of conditions. Next, we provide an overview of the functionality contained within our manually-designed behavior tree, followed by an overview of the behavior tree synthesis method we used and the functionality of the subsequently synthesized behavior tree.

Actions: We defined GoBot's behaviors in three categories: actions relating to movement that are defined with respect to the child's position, actions relating to the reward module, and actions that are a combination of multiple others. The actions within each category are the following:

- *Movement actions:* follow (F) and keep-away (KA)
- *Reward actions:* bubbles (B), lights (L), and sounds (S)
- *Combined actions:* keep-away and bubbles (KA+B), follow and sounds (F+S), and all three reward actions concurrently (B+L+S)

Conditions: In order to specify the states during which an action should occur within the structure of a behavior tree, we also defined GoBot's sensing capabilities as conditions in two categories: child activity conditions that consider the child's movement relative to GoBot, and child-robot interaction distance, which was inspired by the modified Howe's Peer Play Scale [21]. The conditions within each category are the following:

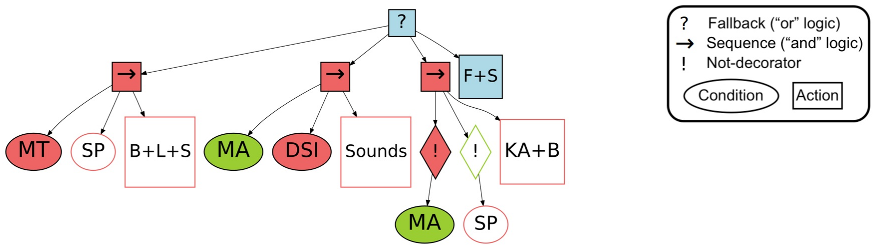

Fig. 4. The synthesized behavior tree (S-BT), a 16-node behavior tree synthesized using the method in Section 3.2 for the child-robot interaction domain. The behavior tree is shown to be running the follow and sounds (F+S) action, because the child is not at a solitary play distance (SP failed), and is moving away from the robot (MA succeeded). A full listing of the acronyms in this tree appears in the bulleted lists in Section 3.2.

- *Child activity conditions:* moving away (MA), moving toward (MT), and stationary (S)
- *Child-robot interaction distance conditions:* direct social interaction (DSI, <3 ft [0.91 m]), social interaction (SI, 3 ft - 6 ft [0.91 m - 1.83 m]), and solitary play (SP, >6 ft [1.83 m])

Manual behavior tree design: Given these actions and conditions, we used our knowledge of the child mobility domain, based on data from previous studies [19, 46], and manually designed a behavior tree that we refer to as the M-BT (shown in Fig. 3). These studies analyzed the success of individual rewards in motivating children to move towards a robot during play sessions, finding that blowing bubbles was a particularly effective action for encouraging child movement. In turn, we emphasized the bubbles action in the M-BT design. Note that to clearly denote node hierarchy, we use the terms parent and child nodes, which pertain only to behavior tree structure, not the human parents and human children in our interaction scenario. We systematically designed the M-BT by dividing the logic into three main subtrees, one for each interaction distance condition (DSI, SI, SP). Each subtree has a *sequence* (⟶) node as its root, with a distance *condition* as the first child node beneath it, and a *fallback* (?) as the second child node. This *fallback* node is the root of the remaining section of the subtree, and has three *sequence* child nodes, one for each child activity condition (MA, MT, S). Beneath each of these *sequence* nodes is a child activity *condition* node, followed by an *action* node for the action that we believed would be most effective in the world state required for behavior tree activity to reach that node. Consider the state of the M-BT shown in Fig. 3, which shows that the combined action follow and sounds (F+S) is running. In order for this to occur, the social interaction (SI) *condition* node returned *success*, the stationary (S) *condition* returned *failure*, and the moving away (MA) *condition* returned *success*. In other words, the robot only executed the follow and sounds (F+S) action when the child in the play space was at a social interaction (SI) distance from the robot and was moving away (MA). Note that in this domain, only one condition in each category defined previously can be true at a time, so there is no potential for conflicting behavior tree action execution commands.

Behavior tree synthesis: To formally validate a manually-designed behavior tree, the creation and comparison of many behavior trees is required. Although the manual design of single behavior tree may not be significantly time-consuming in some cases (depending on domain complexity and designer BT expertise), it is important that there is a feasible form of validation in terms of performance potential. Automatic behavior tree synthesis can more feasibly include this kind of validation, and although it cannot guarantee BT performance in the real world due to the sim-to-real gap, neither can a manually-designed behavior tree. Therefore, we also synthesized a behavior tree that we refer to as the S-BT (shown in Fig. 4) using [39], a behavior tree synthesis method that has a form of performance validation built in. The main

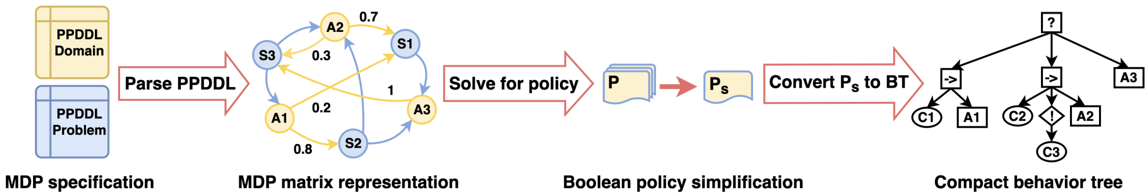

Fig. 5. An overview of the chosen behavior tree synthesis method that is described in more detail in steps (1)-(5) below. Note that this compact behavior tree only shows the general behavior tree structure that can be synthesized using this method (i.e., a *fallback* root, a layer of *sequences* or single *actions*, followed by *conditions* alone or beneath *not-decorators* and *actions* beneath the *sequences*).

prerequisite for this method is that the domain considered and problem to be solved collectively describe a Markov decision process (MDP) that any user with domain knowledge has sufficient information to define using an accessible specification language. It is important to note that this definition of an MDP is not in an immediately solvable form.

Given that this prerequisite is satisfied, the behavior tree synthesis can be carried out through the following five steps that are also illustrated in Fig. 5:

(1) Using available data, the user specifies a domain and problem in the well-documented Probabilistic Planning Domain Definition Language (PPDDL) [50] that collectively define an MDP (see Fig. 6). Though more easily specified in PPDDL, an MDP in this form cannot yet be conveniently solved.
(2) The PPDDL specification is converted to the equivalent MDP transition probability and reward matrices that are compatible with an MDP solver. Given A actions and S states in the MDP, the matrices are each $A \times S \times S$.
(3) The matrices are passed to an MDP solver (e.g., value iteration) which returns the optimal policy.
(4) The policy is then simplified to increase compactness of the behavior tree that will subsequently be synthesized from it. Simplification occurs through the representation of the policy as a Boolean sum of products expression. Then, the essential prime implicants of the expression are found and returned as the simplified policy.
(5) The simplified policy is converted to the equivalent, but more compact, behavior tree. This stage includes additional logic that minimizes the number of nodes required to represent disjunctive expressions in the simplified policy.

In order to specify a given domain in PPDDL, the user must have sufficient data to define the relevant conditions (i.e., predicates) and actions. At a high level, each action is defined by a set of preconditions that describe the state in which the action can occur and probabilistic effects that describe how the state changes due to action execution. Specifying a problem in PPDDL is necessary mainly for syntactical completeness in the context of this behavior tree synthesis method. A related benefit of this method is the ease with which the user can update the PPDDL specification given new knowledge of the domain and subsequently receive an updated behavior tree. Note that Boolean simplification that is similar to (4) in some ways has been done in [16, 48], but these methods focused on a converting a decision tree (as opposed to a policy) to a behavior tree.

We were able to effectively use this method by specifying our child mobility domain in PPDDL using data that was collected in prior related work [19, 46]. This past related work, as described earlier, analyzed the success of individual rewards (e.g., bubbles) in motivating children to move towards a robot during play sessions. Given this PPDDL input, the automatic synthesis method provided a synthesized behavior tree (S-BT) that represented the optimal policy, which is a validation guarantee that cannot be easily achieved through manual behavior tree design. The subsequently synthesized behavior tree shown in Fig. 4 notably has only 16 nodes compared to the 37 nodes the M-BT contains. Additionally,

```
(define (domain D)
(:predicates … )
(:constants … )
(:action A1
        :parameters (…)
        :precondition (…)
        :effect (…)
         )
(:action A2
        … )
)
```

```
(define (problem P)
(:domain D)
(:objects … )
(:init … )
(:goal … )
)
```

Fig. 6. The components of a PPDDL domain and problem specification that must be defined by a user and represent an MDP to synthesize a behavior tree using [39]. For simplicity, we do not provide domain-specific details, but more detailed PPDDL examples can be found in [50] and in PPDDL documentation online.

unlike the explicit logic structure of the M-BT, the S-BT contains logic more implicitly. We believe this is a byproduct of the simplification phase of the behavior tree synthesis method that reduces the number of nodes in the tree. By definition, the S-BT must represent the optimal policy in the given MDP. In other words, every state must be associated with the action that is most likely to maximize the reward (i.e., encourage the most movement of the child in the play space). We examined the logic structure of the S-BT and determined that it often differed in state-action pairings, but that there were some state-action pairing matches between the M-BT and the S-BT. One such match is that follow and sounds (F+S) will execute when the moving away (MA) and social interaction (SI) conditions are true, which is the state of the S-BT in Fig. 4. An example of a discrepancy between the two trees is that bubbles, lights and sounds (B+L+S) will be executed concurrently by the M-BT when the social ineraction (SI) and moving toward (MT) conditions are true, but the S-BT under those conditions will execute keep-away and bubbles (KA+B) concurrently.

4 DATA COLLECTION METHODS

Our data collection aimed to answer the following research questions:

(1) Does a synthesized behavior tree (S-BT) tend to perform as well as a manually-designed behavior tree (M-BT) in encouraging children to move and engage with the robot?
(2) Does a fully autonomous behavior-tree-guided robot (with either type of behavior tree) tend to yield more child movement and engagement when compared with an inactive robot?

Positive results would indicate that a synthesized behavior tree can be as effective as a manually-designed behavior tree while minimizing domain expertise and time required to validate the behavior tree performance. Additionally, a fully autonomous robot guided by behavior trees would be able to encourage children to move and stay engaged while also reducing the input needed by a clinician. We evaluated these research questions with a real-world within-subjects data collection. Participant experiences spanned the following three phases that occurred for five minutes each:

- *Control phase*: GoBot was present but not active. This control phase enabled comparison of typical child behavior in a play space with appropriate toys to child behavior in the same play space with an active robot.
- *M-BT phase (experimental phase 1)*: GoBot was autonomously guided by the manually-designed behavior tree (M-BT).
- *S-BT phase (experimental phase 2)*: GoBot was autonomously guided by the synthesized behavior tree (S-BT).

The data collection protocol described below was approved by our university ethics board under #IRB-2020-0723.

4.1 Participants

11 participants (4 male, 7 female) completed the data collection. Participant ages ranged from 1.6 to 8.6 years old (M = 3.8 and SD = 2.0). All participants were typically developing.

4.2 Procedure

Before the start of each session, parents reviewed and signed an informed consent form and completed a demographics survey. ActiGraph sensors were then placed on the right ankle, right wrist, and hip of the child. During each five-minute phase, the child could interact with any of the toys or the robot in the play space, which is shown in Fig. 7. Each of the three phases (i.e., control, M-BT, and S-BT) took place back-to-back, and the ordering of phases was balanced across participants to ensure each possible experience sequence occurred at least once (using a Latin squares method). Throughout each session, a researcher operated OverTrack, as described previously in Section 3.1. At the completion of each phase, parents answered the one-question engagement survey, and at the end of the session, parents completed the closing survey.

4.3 Measurements

In research studies with young children, results can be quite variable across a full participant group. Accordingly, we evaluated the following measurements both across all participants and within each participant's own data. We collected two types of data measurements: *behavioral* and *self-reported*. Behavioral measurements included ActiGraph sensor measurements (acceleration and angular velocity) and overhead video metrics. Parent survey self-reports were also collected to understand perceptions of each child's engagement with the robot.

Behavioral measurements: Tri-axial acceleration and angular velocity data was recorded by three GT9X Link ActiGraph sensors (worn on the child's right wrist, right ankle, and right side of the hip) at 100 Hz. The wrist sensor was placed on the wrist facing upwards (similar to a watch face) while the ankle and hip sensors were placed on the lateral side of the body facing outwards. Overhead video data of the full session was collected from a GoPro Hero Black 10 camera at 30 Hz. A side view was also recorded using a GoPro Hero Black 7 camera at 30 Hz. Positional data collected in real time by OverTrack [20] and recorded to a spreadsheet for post hoc analysis.

Self-reported measurements: Parents were given two surveys. At the end of each phase, parents completed one Likert-type question that asked them to rate their child's engagement with GoBot on a scale of Strongly Disagree (1) to Strongly Agree (7). Parents completed a closing survey that included Likert-type and free-response questions. Likert-type questions used the same scale as the phase-wise survey question and covered the following three topics: general perception of child engagement with GoBot, belief in robot usefulness for child well-being, and interest in participating in future studies. Parents responded to the following three free-response questions: 1) How do you think robots can be useful to improve the well-being of children?, 2) In general, how did your child interact with the robot throughout the session?, and 3) In general, what is your perception of the robot used in this session?

4.4 Analysis

We evaluated measurements both in terms of trends in the data as well as with significance testing. Our data collection sought to provide an initial understanding of the capabilities of behavior trees in this domain, so both the trending and statistical testing results can be helpful for reasoning about next steps. We evaluated trends in the data by phase and across participants both with descriptive statistics and by calculating how many participants had equal to or

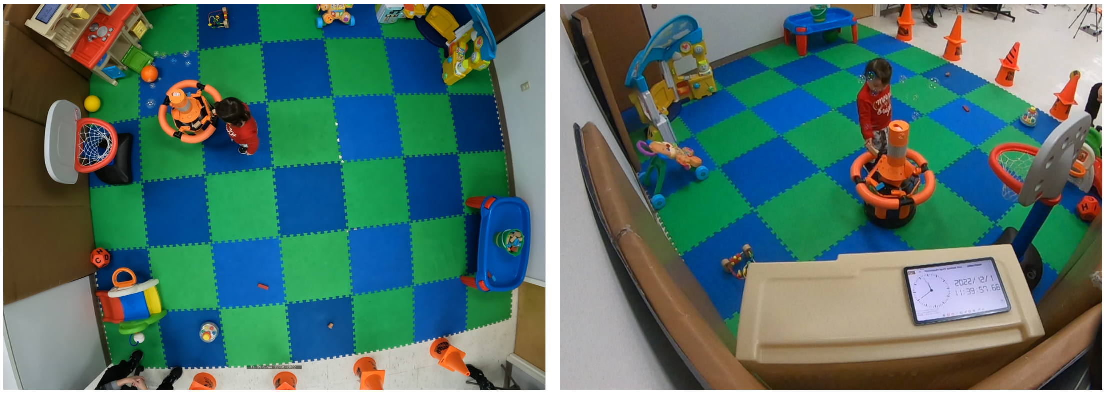

Fig. 7. Overhead and side view of play space showing the data collection setup with GoBot, toys, and the participant.

higher behavioral and survey values in the M-BT or S-BT phases when compared with the control phase. We also tested each measurement for significant differences across phases in the jamovi 2.3.21 software [37, 41, 44] using repeated-measures analysis of variance (RM-ANOVA) tests with α = 0.05 significance level. We used Tukey's HSD test for pairwise comparisons in the case of significant main effects. We report effect sizes using η^2, where η^2 = 0.010 is considered a small effect, η^2 = 0.040 is a medium effect, and η^2 = 0.090 is a large effect [17].

ActiGraph data: We first used the ActiLife v6.13.4 software to transfer recorded accelerometer and gyroscope data from the ActiGraph sensors to a computer for processing. We used the algorithm described in [45] to determine events that were likely to be *ankle movements*. Figure 8 shows an example of ankle movements counted by the algorithm over 10 seconds of one participant's inertial data. We analyzed only the ankle sensor recordings since we were most interested in walking movement. The algorithm works by first calculating the root mean square (RMS) acceleration and angular velocity from each participant's raw ankle sensor recordings. We used the rejection ranges provided by the related work [45] to filter the acceleration and gyroscope signal before the algorithm computed participant-specific thresholds for both acceleration and angular velocity. After incorporating a 0.5-second window moving average filter, the algorithm identified the start of a significant ankle movement as an instance when both the acceleration and gyroscope thresholds were exceeded and the end of a movement as when the gyroscope data dropped below the associated threshold. We used the algorithm to calculate the count of ankle movements for each phase and participant.

Overhead tracking data: To calculate the *total movement* of each participant and the *mean child-robot distance* during each phase, we used the positional data collected by OverTrack [20]. We computed the total amount of child movement during each phase by summing the change in the bounding box centroids of the child between subsequent frames, after excluding position changes smaller than 0.06 ft (0.02 m, likely to be noise), and larger than 0.5 ft (0.15 m, highly unlikely based on maximum child ambulation speed [40]). The recorded centroids for the child and robot were used to compute the per-phase mean child-robot Euclidean distance.

Survey responses: We assessed *engagement* using parent responses on the one-question engagement survey for each phase. We report the mean overall parent perception of child engagement with the robot, in addition to the results of the statistical tests.

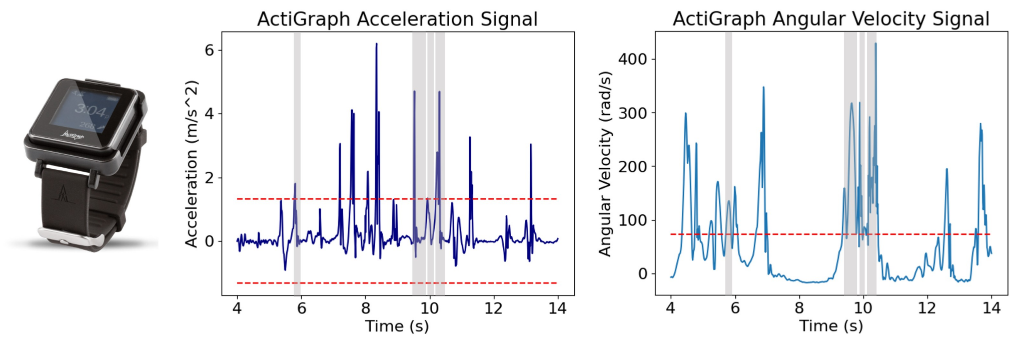

Fig. 8. Image of the ActiGraph GT9X sensor and example algorithm output for identifying ankle movements over 10 seconds of one participant's ActiGraph accelerometer and gyroscope sensor data (values are the root mean square (RMS) of each sensor data minus the median). Red lines indicate participant-specific thresholds and gray boxes indicate ankle movement periods as determined by the algorithm (i.e., periods when both the RMS accelerometer and RMS gyroscope readings exceeded the participant-specific thresholds). For this segment of sensor data, the algorithm identified four ankle movements.

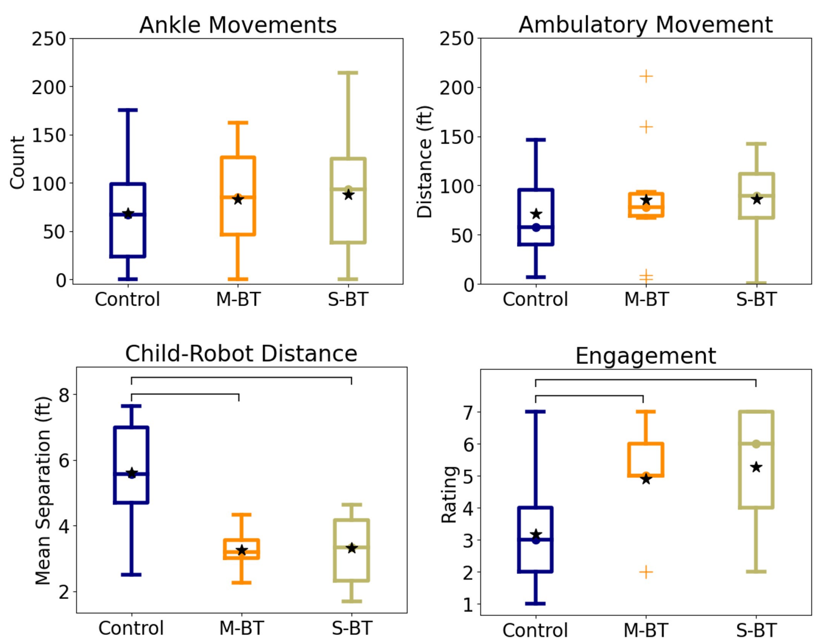

Fig. 9. Data collection results for ankle movement, ambulatory movement, child-robot distance, and engagement. Boxplots include boxes from the 25th to the 75th percentiles, center lines with a circle marker for medians, asterisks for means, whiskers up to 1.5 times the inter-quartile range, and "+" marks for outliers. Brackets indicate significant pairwise differences.

5 DATA COLLECTION RESULTS

All participants successfully completed the data collection protocol. The results of our ActiGraph, overhead tracking system, and survey data analysis appear below. As a reminder, we considered both trends in the data as well as the results of statistical testing to understand early benefits and limitations of using behavior trees in the child mobility domain.

ActiGraph results: As evidenced by the results plot in Fig. 9, the number of ankle movements tended to be greater for the experimental phases (active robot with the M-BT or S-BT) when compared to the control phase (inactive robot). When tabulating how many participants displayed more ankle movements during each experimental phase when compared to the control phase, we found that seven out of the 11 participants had equal or more ankle movements during the M-BT phase and nine out of the 11 participants had equal or more ankle movements during the S-BT phase. However, RM-ANOVA tests across phases showed no significant results (p = 0.226).

Overhead tracker results: Like for the ankle movements, the measurements from the overhead tracking data, including ambulatory movement and child-robot spacing, tended to be greater for the experimental phases when compared to the control phase (see Fig. 9). Seven out of the 11 participants moved farther during the M-BT phase when compared to the control and seven out of the 11 participants moved farther during the S-BT phase when compared to the control. There was no significant difference in ambulatory movement across phases (p = 0.669).

Child-robot spacing data showed positive trends in addition to significant results, as elucidated in Fig. 9. Within participants, nine out of 11 children were closer to the robot during the M-BT phase when compared to the control and 10 out of 11 children were closer to the robot during the S-BT phase when compared to the control. RM-ANOVA tests showed significant differences (p<0.001, F(2, 20) = 16.5, η^2 = 0.466). Tukey's HSD tests revealed that children were significantly closer to GoBot in both experimental phases when compared to the control and that there was no significant difference in the distance between the child and robot when comparing the M-BT and S-BT phases.

Survey results: Responses to the per-phase one-question engagement survey, as presented in Fig. 9, showed that parents perceived their children to be most engaged with the robot while it was active. We evaluated individual parent ratings of child engagement and found that seven out of the 11 parents rated their child to be more engaged during the M-BT phase when compared to the control. Seven out of 11 parents rated their child to be more engaged during the S-BT phase when compared to the control. Parents rated their children as significantly more engaged with GoBot during the M-BT and S-BT phases when compared to the control (p = 0.003, F(2, 20) = 7.8, η^2 = 0.211). At the end of each session, parents rated their perception of overall engagement with the robot. The mean overall rating was 5.1 (*SD*=1.6) on the 7-point scale described previously.

6 DISCUSSION

Our research goal was to provide GoBot with autonomous capabilities and to test how effectively each behavior tree could encourage child movement and engagement. We conducted an in-person data collection using the manually-designed behavior tree (M-BT) or the synthesized behavior tree (S-BT) on GoBot, and we discuss the results, implications, strengths and limitations, and future directions of our work below.

6.1 Design Implications

The results of our data collection provides evidence that the behavior tree synthesis method is a viable replacement for manual behavior tree design in the child-robot interaction domain, despite the minimal input data available and the lack of a relevant simulator. GoBot performed similarly with respect to child movement and engagement whether it was using the manually-designed behavior tree (M-BT) or the synthesized behavior tree (S-BT). This suggests that behavior tree synthesis could replace manual design, providing benefits like feasible (i.e., faster) validation and no requirement of behavior tree expertise. A potential contributing factor behind the similar success of both behavior trees is their inclusion of actions that showed promise in prior child mobility work [19]. More specifically, both behavior trees emphasize the follow and sounds action when a child is moving away from GoBot, and actions involving bubbles when a children is approaching GoBot.

Even if a synthesized behavior tree is successful as defined above, it does not mean that the approach is able to effectively encourage the desired child behaviors. However, the behavioral and survey results demonstrate that our autonomous robot did tend to encourage more ankle movements and ambulatory movement and significantly increased engagement compared to the control phase. Additionally, parents' written feedback shows that their perceptions of the children's engagement with GoBot was high. One parent wrote "I think [my child] likes a lot this robot." Another parent mentioned that their child was "happy to see [the robot], slow to warm [to the robot], and then excited to see bubbles, lights, and play blocks with [the robot]." Furthermore, a different parent noted that their child "loved playing with bubbles, and was curious about the robot." Overall, parent feedback showed that they found their children to be engaged and enjoying interactions with the robot. This is a promising sign for our envisioned robot-mediated motor interventions, since a robot that requires less human input (i.e., is more autonomous) and can engage children is more likely to be adopted in clinical or home settings.

6.2 Strengths & Limitations

Our work presents an autonomous assistive robot guided by behavior trees, one manually-designed and one synthesized automatically. Our results suggest that synthesizing a behavior tree using the method detailed in Section 3.2 will not only lessen the expertise and time required from a user, but will also result in a competitively-performing behavior tree in comparison to a manually-designed behavior tree. Preliminary results further show that the synthesized trees may be easier to understand due to their compactness. Additionally, even with sufficient behavior tree and domain expertise and time to design a behavior tree manually, it is very difficult to validate the performance of a manually-designed behavior tree because it requires the creation and comparison of many other behavior trees. The behavior tree synthesis method we employ in this work avoids this issue with performance validation because it inherently returns a behavior tree that has verifiably equivalent performance to the optimal policy in the MDP that the input data represents. Neither manual behavior tree design nor automatic synthesis can fill the sim-to-real gap (e.g., MDPs do not take partial observability into account). However, the synthesis method we use alleviates the need for the user to understand BT logic or structure, and produces a behavior tree that will perform increasingly well in the real world as the MDP PPDDL specification is made to be a more accurate representation of reality. We tested GoBot's autonomous performance during free play with children and found that the robot engaged children and tended to encourage movement. Limited work has been done with autonomous robots using a behavior tree in child-robot domains and, to the best our knowledge, this is the first instance of a behavior-tree-guided robot for child mobility.

Main limitations of this work include the small sample size in our experiment and the relatively brief interaction that each of the participants had with GoBot. We aim to conduct longitudinal studies with more participants in the future to further test our behavior trees and validate the robot's performance, with the potential for stronger empirical claims. Additional control conditions (such as an active robot informed by a simple LIDAR-based policy) could also provide new context on how the BT policies compare to other simpler policies for moving GoBot. The participants represented in the current work were not gender- and age-balanced and do not include children with motor disabilities. Depending on goals of future validations, we will need to adjust the recruited samples; for example, we will need to work with children with motor disabilities to draw conclusions about the robot's influence in actual motor interventions, rather than for general free play with children. Prior to the experiment we performed in this work, we only had limited data to inform the synthesis of the S-BT in this work. In future work, we aim to incorporate the data collected from our evaluation sessions discussed here, in order to more accurately represent the reality of the child mobility domain, and subsequently allow the synthesis method to result in a behavior tree more specifically suited for the actual domain.

6.3 Conclusions

In this work, our main goal was to encourage children to be more physically active through engagement with a robot. We manually designed a behavior tree and synthesized another for the same child mobility domain, in order to provide behavior-tree-based autonomy to our assistive robot, GoBot. We then evaluated whether the behavior trees were an effective form of autonomy for this goal and whether they performed comparably to one another. The behavioral results showed that GoBot tended to be able to encourage child movement and that children were significantly more engaged with both behavior trees, as compared to a control phase with an inactive robot. Parent perception of child engagement with GoBot corroborated these results. Ultimately, our work have shown that behavior tree synthesis methods and the union of planning, autonomy, and human-robot interaction have great potential to improve child mobility in assistive robotic domains.

ACKNOWLEDGMENTS

This work was supported by funding from NSF award CMMI-2024950.

REFERENCES

[1] Bikramjit Banerjee. 2018. Autonomous acquisition of behavior trees for robot control. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*. IEEE, Madrid, Spain, 3460–3467.

[2] Janelle Boyd, Jgenisius Harris, Michelle Smith, Sergio García-Vergara, Yu-Ping Chen, Ayanna Howard, et al. 2017. An infant smart-mobile system to encourage kicking movements in infants at-risk of cerebral palsy. In *Proc. of the IEEE Workshop Advanced Robotics and Its Social Impacts (ARSO)*. IEEE, Austin, TX, USA, 1–5.

[3] Nicoleta Bugnariu, Carolyn Young, Katelyn Rockenbach, Rita M Patterson, Carolyn Garver, Isura Ranatunga, Monica Beltran, Nahum Torres-Arenas, and Dan Popa. 2013. Human-robot interaction as a tool to evaluate and quantify motor imitation behavior in children with Autism Spectrum Disorders. In *Proc. of the IEEE Int. Conf. on Virtual Rehabilitation (ICVR)*. IEEE, Philadelphia, PA, USA, 57–62.

[4] Vicky Charisi, Luis Merino, Marina Escobar, Fernando Caballero, Randy Gomez, and Emilia Gómez. 2021. The Effects of Robot Cognitive Reliability and Social Positioning on Child-Robot Team Dynamics. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, Xi'an, China, 9439–9445.

[5] Caitlyn Clabaugh, Kartik Mahajan, Shomik Jain, Roxanna Pakkar, David Becerra, Zhonghao Shi, Eric Deng, Rhianna Lee, Gisele Ragusa, and Maja Matarić. 2019. Long-term personalization of an in-home socially assistive robot for children with autism spectrum disorders. *Frontiers in Robotics and AI* 6 (2019), 110.

[6] Caitlyn Clabaugh and Maja Matarić. 2019. Escaping oz: Autonomy in socially assistive robotics. *Annual Review of Control, Robotics, and Autonomous Systems* 2, 1 (2019), 33–61.

[7] Cecilia Clark, Levin Sliker, Jim Sandstrum, Brian Burne, Victoria Haggett, and Cathy Bodine. 2019. Development and preliminary investigation of a semiautonomous Socially Assistive Robot (SAR) designed to elicit communication, motor skills, emotion, and visual regard (engagement) from young children with complex cerebral palsy: A pilot comparative trial. *Advances in Human-Computer Interaction* 2019 (2019), 2614060.

[8] Michele Colledanchise and Petter Ögren. 2018. *Behavior Trees in Robotics and AI: An Introduction.* CRC Press, Boca Raton, FL, USA.

[9] Michele Colledanchise, Ramviyas Parasuraman, and Petter Ögren. 2018. Learning of behavior trees for autonomous agents. *IEEE Trans. on Games* 11, 2 (2018), 183–189.

[10] Enrique Coronado, Xela Indurkhya, and Gentiane Venture. 2019. Robots Meet Children, Development of Semi-Autonomous Control Systems for Children-Robot Interaction in the Wild. In *Proc. of the IEEE Int. Conf. on Advanced Robotics and Mechatronics (ICARM)*. IEEE, Toyonaka, Japan, 360–365.

[11] Rahul Dey and Chris Child. 2013. QL-BT: Enhancing behaviour tree design and implementation with Q-learning. In *Proc. of the IEEE Conf. on Computational Intelligence in Games (CIG)*. IEEE, Niagara Falls, ON, Canada, 1–8.

[12] Chiara Filippini, Edoardo Spadolini, Daniela Cardone, Domenico Bianchi, Maurizio Preziuso, Christian Sciarretta, Valentina del Cimmuto, Davide Lisciani, and Arcangelo Merla. 2021. Facilitating the child–robot interaction by endowing the robot with the capability of understanding the child engagement: The case of Mio Amico Robot. *International Journal of Social Robotics* 13, 4 (2021), 677–689.

[13] David Fischinger, Peter Einramhof, Konstantinos Papoutsakis, Walter Wohlkinger, Peter Mayer, Paul Panek, Stefan Hofmann, Tobias Koertner, Astrid Weiss, Antonis Argyros, et al. 2016. Hobbit, a care robot supporting independent living at home: First prototype and lessons learned. *Robotics and Autonomous Systems* 75 (2016), 60–78.

[14] Naomi T Fitter, Rebecca Funke, José Carlos Pulido, Lauren E Eisenman, Weiyang Deng, Marcelo R Rosales, Nina S Bradley, Barbara Sargent, Beth A Smith, and Maja J Mataric. 2019. Socially assistive infant-robot interaction: Using robots to encourage infant leg-motion training. *IEEE Robotics & Automation Magazine* 26, 2 (2019), 12–23.

[15] Sharon E Fox, Pat Levitt, and Charles A Nelson III. 2010. How the timing and quality of early experiences influence the development of brain architecture. *Child Development* 81, 1 (2010), 28–40.

[16] Kevin French, Shiyu Wu, Tianyang Pan, Zheming Zhou, and Odest Chadwicke Jenkins. 2019. Learning behavior trees from demonstration. In *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. IEEE, Montreal, QC, Canada, 7791–7797.

[17] David C Funder and Daniel J Ozer. 2019. Evaluating effect size in psychological research: Sense and nonsense. *Advances in Methods and Practices in Psychological Science* 2, 2 (2019), 156–168.

[18] Binnur Görer, Albert Ali Salah, and H Levent Akın. 2017. An autonomous robotic exercise tutor for elderly people. *Autonomous Robots* 41, 3 (2017), 657–678.

[19] Ameer Helmi, Samantha Noregaard, Natasha Giulietti, Samuel W Logan, and Naomi T Fitter. 2022. Let Them Have Bubbles! Filling Gaps in Toy-Like Behaviors for Child-Robot Interaction. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, Philadelphia, PA, USA, 7417–7422.

[20] Ameer Helmi, Connor Phillips, Fernando Castillo, Samuel W Logan, and Naomi T Fitter. 2023. OverTrack: Overhead Camera Tracking Tool for Child-Robot Interaction. In *Proc. of the Social Robot Navigation: Advances and Evaluation Workshop, held in conjunction with the Int. Conference on Intelligent Robots and Systems (IROS)*. IEEE, Detroit, Michigan, USA, 7417–7422.

[21] Carollee Howes and Catherine C Matheson. 1992. Sequences in the development of competent play with peers: Social and social pretend play. *Developmental Psychology* 28, 5 (1992), 961.

[22] Matteo Iovino, Edvards Scukins, Jonathan Styrud, Petter Ögren, and Christian Smith. 2022. A survey of Behavior Trees in Robotics and AI. *Robotics and Autonomous Systems* 154 (2022), 104096.

[23] Andreas Klöckner. 2013. Interfacing behavior trees with the world using description logic. In *Proc. of the AIAA Guidance, Navigation, and Control (GNC) Conf.* AIAA, Boston, MA, USA, 4636.

[24] Elena Kokkoni, Effrosyni Mavroudi, Ashkan Zehfroosh, James C Galloway, Renè Vidal, Jeffrey Heinz, and Herbert G Tanner. 2020. GEARing smart environments for pediatric motor rehabilitation. *Journal of Neuroengineering and Rehabilitation* 17, 1 (2020), 1–15.

[25] Ioannis Kostavelis, Dimitrios Giakoumis, Georgia Peleka, Andreas Kargakos, Evangelos Skartados, Manolis Vasileiadis, and Dimitrios Tzovaras. 2018. RAMCIP robot: a personal robotic assistant; demonstration of a complete framework. In *Proc. of the European Conf. on Computer Vision (ECCV) workshops*. Springer, Munich, Germany, 0–0.

[26] Sotiris B Kotsiantis. 2013. Decision trees: a recent overview. *Artificial Intelligence Review* 39 (2013), 261–283.

[27] David Lee and Mihalis Yannakakis. 1996. Principles and methods of testing finite state machines-a survey. *Proc. of the IEEE* 84, 8 (1996), 1090–1123.

[28] Chong-U Lim, Robin Baumgarten, and Simon Colton. 2010. Evolving behaviour trees for the commercial game DEFCON. In *Proc. of the European Conf. on the Applications of Evolutionary Computation (EvoCOP)*. Springer, Berlin, Germany, 100–110.

[29] Samuel W Logan, Melynda Schreiber, Michele Lobo, Breanna Pritchard, Lisa George, and James Cole Galloway. 2015. Real-world performance: Physical activity, play, and object-related behaviors of toddlers with and without disabilities. *Pediatric Physical Therapy* 27, 4 (2015), 433–441.

[30] Flavia Marino, Paola Chilà, Stefania Trusso Sfrazzetto, Cristina Carrozza, Ilaria Crimi, Chiara Failla, Mario Busà, Giuseppe Bernava, Gennaro Tartarisco, David Vagni, et al. 2020. Outcomes of a robot-assisted social-emotional understanding intervention for young children with autism spectrum disorders. *Journal of Autism and Developmental Disorders* 50, 6 (2020), 1973–1987.

[31] François Michaud, Tamie Salter, Audrey Duquette, Henri Mercier, Michel Lauria, Helene Larouche, and Francois Larose. 2007. Assistive technologies and child-robot interaction. In *Proc. of the AAAI Spring Sym. on Multidisciplinary Collaboration for Socially Assistive Robotics*. AAAI, Stanford, CA, USA.

[32] Aadesh Neupane and Michael Goodrich. 2019. Learning swarm behaviors using grammatical evolution and behavior trees. In *Proc. of the Int. Joint Conf. on Artificial Intelligence (IJCAI)*. International Joint Conferences on Artificial Intelligence Organization, Macao, China, 513–520.

[33] Petter Ögren. 2012. Increasing modularity of UAV control systems using computer game behavior trees. In *Proc. of the AIAA Guidance, Navigation, and Control Conf.* AIAA, Minneapolis, MN, USA, 4458.

[34] Diego Perez, Miguel Nicolau, Michael O'Neill, and Anthony Brabazon. 2011. Evolving behaviour trees for the Mario AI competition using grammatical evolution. In *Proc. of the European Conf. on the Applications of Evolutionary Computation (EvoCOP)*. Springer, Berlin, Germany, 123–132.

[35] Lihui Pu, Wendy Moyle, Cindy Jones, and Michael Todorovic. 2019. The effectiveness of social robots for older adults: a systematic review and meta-analysis of randomized controlled studies. *The Gerontologist* 59, 1 (2019), e37–e51.

[36] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, Andrew Y Ng, et al. 2009. ROS: an open-source Robot Operating System. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA) Workshop on Open Source Software*, Vol. 3. IEEE, Kobe, Japan, 5.

[37] R Core Team. 2020. R: A language and environment for statistical computing (Version 4.0) [Computer software]. https://cran.r-project.org/

[38] Emily Scheide, Graeme Best, and Geoffrey A Hollinger. 2021. Behavior tree learning for robotic task planning through Monte Carlo DAG search over a formal grammar. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, Xi'an, China, 4837–4843.

[39] Emily Scheide, Graeme Best, and Geoffrey A Hollinger. 2025. Synthesizing compact behavior trees for probabilistic robotics domains. *Autonomous Robots* 49, 1 (2025), 3.

[40] Bénédicte Schepens, PA Willems, and GA Cavagna. 1998. The mechanics of running in children. *The Journal of Physiology* 509, Pt 3 (1998), 927.

[41] Henrik Singmann. 2018. afex: Analysis of Factorial Experiments [R package]. https://cran.r-project.org/package=afex

[42] Christopher Iliffe Sprague, Özer Özkahraman, Andrea Munafo, Rachel Marlow, Alexander Phillips, and Petter Ögren. 2018. Improving the modularity of AUV control systems using behaviour trees. In *Proc. of the IEEE/OES Autonomous Underwater Vehicle Workshop (AUV)*. IEEE, Porto, Portugal, 1–6.

[43] Jonathan Styrud, Matteo Iovino, Mikael Norrlöf, Mårten Björkman, and Christian Smith. 2022. Combining planning and learning of behavior trees for robotic assembly. In *Proc. of the IEEE Int. Conf. on Robotics and Automation (ICRA)*. IEEE, Philadelphia, PA, USA, 11511–11517.

[44] The jamovi project. 2020. jamovi (Version 1.6) [Computer software]. https://www.jamovi.org

[45] Ivan A Trujillo-Priego, Christianne J Lane, Douglas L Vanderbilt, Weiyang Deng, Gerald E Loeb, Joanne Shida, and Beth A Smith. 2017. Development of a wearable sensor algorithm to detect the quantity and kinematic characteristics of infant arm movement bouts produced across a full day in the natural environment. *Technologies* 5, 3 (2017), 39.

[46] Ashwin Vinoo, Layne Case, Gabriela R Zott, Joseline Raja Vora, Ameer Helmi, Samuel W Logan, and Naomi T Fitter. 2021. Design of an assistive robot for infant mobility interventions. In *Proc. of the IEEE Int. Conf. on Robot & Human Interactive Communication (RO-MAN)*. IEEE, Vancouver, BC, Canada, 604–611.

[47] Joseline Raja Vora, Ameer Helmi, Christine Zhan, Eliora Olivares, Tina Vu, Marie Wilkey, Samantha Noregaard, Naomi T Fitter, and Samuel W Logan. 2021. Influence of a Socially Assistive Robot on Physical Activity, Social Play Behavior, and Toy-Use Behaviors of Children in a Free Play Environment: A Within-Subjects Study. *Frontiers in Robotics and AI* 8 (2021), 368.

[48] Adam Wathieu, Thomas R Groechel, Haemin Jenny Lee, Chloe Kuo, and Maja J Matarić. 2022. Re: Bt-espresso: Improving interpretability and expressivity of behavior trees learned from robot demonstrations. In *Proc. of the Int. Conf. on Robotics and Automation (ICRA)*. IEEE, Philadelphia, PA, USA, 11518–11524.

[49] Kate L Willoughby, Karen J Dodd, and Nora Shields. 2009. A systematic review of the effectiveness of treadmill training for children with cerebral palsy. *Disability and Rehabilitation* 31, 24 (2009), 1971–1979.

[50] Håkan LS Younes and Michael L Littman. 2004. PPDDL 1.0: An extension to PDDL for expressing planning domains with probabilistic effects. *Techn. Rep. CMU-CS-04-162* 2 (2004), 99.