Proceedings of the 35th European Safety and Reliability & the 33rd Society for Risk Analysis Europe Conference
Edited by Eirik Bjorheim Abrahamsen, Terje Aven, Frederic Bouder, Roger Flage, Marja Ylönen

doi: 10.3850/978-981-94-3281-3_ESREL-SRA-E2025-P3449-cd

From Theory to Practice: Achieving Reliable Robot Autonomy through Symbol Grounding

Aayush Jain
Atlantic Technological University & Technological University of Dublin, Ireland. E-mail: aayush.jain@atu.ie

Philip Long
Atlantic Technological University, Ireland. E-mail: philip.long@atu.ie

John D. Kelleher
Trinity College Dublin, Ireland. E-mail: john.kelleher@tcd.ie

Maria Chiara Leva
Technological University of Dublin, Ireland. E-mail: mariachiara.leva@tudublin.ie

In an era where robotic autonomy is becoming increasingly pivotal across various sectors, the symbol grounding problem remains a significant barrier to achieving reliable, context-aware automation. This paper presents novel frameworks to enhance robot autonomy by integrating symbol grounding into autonomous systems, specifically focusing on robot manipulation tasks. We first introduce a method for programming robots using behavior trees, derived from single demonstrations, which embeds symbolic knowledge into robot operations, enabling adaptability. Second, we discuss a framework that integrates spatial scene graphs with BTs to improve task execution through an enhanced understanding of spatial relationships and object interactions, which is crucial for dynamic and cluttered environments. Lastly, we present a neuro-symbolic approach for failure detection and diagnosis in robotic systems. This approach leverages the synergy between symbolic reasoning and neural network capabilities to detect and diagnose operational failures accurately, addressing the critical need for reliability in automated systems. The preliminary evaluation results demonstrate advancements in the programming, execution, and failure detection of robot manipulation tasks, paving the way for more adaptive and intelligent robotic systems in complex real-world applications.

Keywords: Autonomous Robots, Reliable Robot Autonomy, Robot Programming by Demonstration, Scene Graphs, Behavior Trees, Failure Detection and Diagnosis

1. Introduction

Robots are becoming increasingly important in various sectors, including manufacturing, healthcare, and home automation. Robots are expected to perform complex tasks with minimal human intervention, which requires them to understand and interact with their environments effectively (Jain et al., 2022). However, despite advances in artificial intelligence and machine learning, robots still struggle to interpret and execute tasks in dynamic and unstructured environments. A fundamental issue underlying this limitation is the symbol grounding problem, which refers to the challenge of associating abstract symbolic representations with real-world entities and actions (Harnad, 1990). The inability to bridge this gap results in robots that lack adaptability and struggle with generalization across different tasks and environments (Coradeschi and Saffiotti, 2003).

One of the primary reasons for the persistence of this problem is that traditional robotic systems rely on predefined rules and structured environments (Diehl et al., 2021). These systems are not well-suited for real-world applications where variations in object positions, task sequences, and environmental conditions are common. Approaches such as programming by demonstrations (PbD) (Villani et al., 2018) have been proposed to address this issue, but they often fail to integrate symbolic reasoning effectively. Furthermore, while neural networks have shown promise in learning from data, they are require large amount of data (Sünderhauf et al., 2018), lack interpretability and struggle with generalizing beyond their training distributions. This has motivated the need for hybrid approaches that combine symbolic reasoning with data-driven learning to improve robot autonomy and adaptability (Kroemer et al., 2021; Garcez and Lamb, 2023).

To address these challenges, this paper presents three frameworks to enhance robot autonomy through symbol grounding. First, we propose programming robots with Behavior Trees (BTs) derived from single demonstrations, embedding symbolic knowledge for greater adaptability. Second, we integrate spatial scene graphs with BTs, enabling robots to plan and execute tasks with improved spatial awareness. Finally, we introduce a neuro-symbolic approach for failure detection, combining symbolic reasoning with neural networks to enhance reliability. These contributions provide a step forward in addressing the symbol grounding problem and improving the efficiency, adaptability, and reliability of robot manipulation tasks.

The rest of this paper is organized as follows: Section 2 reviews related work, highlighting existing approaches and their limitations. Section 3 describes the proposed frameworks, including their design and implementation. Section 4 presents a discussion about the proposed frameworks, conclusion, limitations and potential directions for future research advancing robotic autonomy.

2. Related Work

PbD (also known as learning from demonstration or imitation learning), in the domain of robotics, has been extensively explored as an intuitive method for teaching robots complex tasks by leveraging human demonstrations (Chernova and Thomaz, 2014). PbD methods typically focus on two levels of learning: motion-level learning, which involves trajectory imitation and adaptation, and task-level learning, which abstracts task execution into symbolic representations (Osa et al., 2018). Motion-level approaches, such as Dynamic Movement Primitives (DMPs) (Ijspeert et al., 2013), enable robots to generalize learned trajectories but lack semantic understanding of task constraints and objectives (Kyrarini et al., 2019; Lentini et al., 2020; Knaust and Koert, 2021). Conversely, task-level learning, which includes symbolic planning and action segmentation, enhances flexibility but often requires significant manual intervention for labeling and structuring data (Gustavsson et al., 2022; Suddrey et al., 2022; Eiband et al., 2023; Kroemer et al., 2015). In our work, we aim to integrate both motion and task-level learning to improve robot adaptability and generalization (Jain et al., 2024).

BTs have gained prominence as a structured and modular representation for encoding task execution policies in robotic systems (Colledanchise and Ögren, 2018). Originally developed in the gaming industry, BTs offer a hierarchical and reactive control structure that enables task decomposition and execution monitoring (Colledanchise et al., 2019). Unlike traditional Finite State Machines (FSMs) (Grollman and Jenkins, 2010), Hidden Markov Models (HMM) (Kroemer et al., 2015) and Hierarchical Task Networks (HTNs) (Nejati et al., 2006), BTs provide enhanced modularity, reusability, and adaptability to dynamic environments. Several studies have focused on learning BTs from demonstrations, but many require multiple training examples (Gustavsson et al., 2022) or rely on predefined structures (French et al., 2019), limiting their scalability and deployment in real-world applications.

A significant challenge in robot task execution is enabling generalization across different task environments. Scene graphs (Aksoy et al., 2011) have been explored as a means to incorporate spatial and semantic relationships into robotic decision-making (Kroemer et al., 2021). These structured representations capture object affordances and task dependencies, facilitating improved task transfer and execution in varied settings (Ni et al., 2023). However, most existing approaches rely on manually constructed scene graphs or require extensive training data (Zanchettin, 2023; Aksoy et al., 2011), limiting their feasibility in settings with high task variability. In our work, we are investigating the integration of scene graphs with BTs to enhance symbolic reasoning and enable more adaptable robotic task execution.

Failure detection and recovery is another critical research area in autonomous robotics (Khalastchi and Kalech, 2018). Traditional approaches rely on manually defined failure conditions (Akerkar and Sajja, 2009) or static threshold-based anomaly detection methods (Hundman et al., 2018), which often fail in highly dynamic environments. Data-driven failure detection techniques, such as Variational Autoencoders (VAEs) and Long Short-Term Memory (LSTM) networks, have demonstrated the ability to model normal execution patterns and detect deviations indicative of failures (Park et al., 2018). However, purely data-driven methods lack interpretability, making it difficult for the model to detect contextual failures (Mitrevski et al., 2023) and the users to diagnose and rectify failures effectively. In our work, we have introduced neuro-symbolic methods that integrate symbolic reasoning with deep learning techniques to improve failure detection accuracy and provide failure diagnostics.

The proposed research builds upon these existing efforts by integrating PbD, BTs, spatial scene graphs, and neuro-symbolic learning into a cohesive framework for robot programming and execution monitoring. By leveraging a single demonstration to generate adaptable BTs, incorporating spatial reasoning for improved task execution, and employing neuro-symbolic learning for failure detection, this approach seeks to enhance the autonomy, adaptability, and reliability of robotic systems in real-world applications.

3. Proposed Framework

To address the limitations in current robotic autonomy, This work introduces three frameworks that integrate symbol grounding into robotic systems: (1) CoBT, which uses PbD to generate adaptive BTs from a single demonstration; (2) CoBT 2.0, which extends CoBT by incorporating spatial scene graphs for improved object relationship understanding and task execution; and (3) NsAI FDD, which combines symbolic reasoning with neural networks for enhanced failure detection and diagnosis. This section outlines their methodologies and preliminary evaluation.

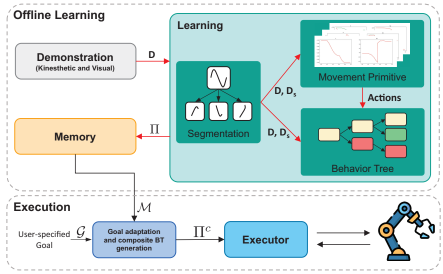

Fig. 1. Workflow of offline learning and execution in CoBT. The red lines depict the offline learning phase and the black lines depict the execution phase.

3.1. *CoBT Framework*

Industrial robots typically require expert-driven coding, making conventional programming rigid, time-consuming, and unsuitable for dynamic environments. Small and medium-sized enterprises struggle with frequent reprogramming, hindering scalability. A major challenge in flexible automation is enabling robots to execute complex tasks with minimal human input. Programming by Demonstration (PbD) offers a solution by teaching tasks through demonstration rather than coding. However, many PbD methods demand multiple demonstrations or extensive manual intervention, limiting their real-world applicability.

Collaborative Programming of Behavior Trees (CoBT) (Jain et al., 2024) framework addresses these challenges by providing a structured, data-efficient framework that generates reactive and modular BTs from a single demonstration. Unlike traditional methods that require manual task decomposition or extensive training data, CoBT leverages a combination of data-driven learning and logic-based declarative reasoning to segment demonstrations into meaningful task sequences. The generated BTs enable robots to execute tasks flexibly, adapting to variations in object placement, environmental changes, and new goal configurations without requiring additional demonstrations. This approach ensures that robots can generalize learned behaviors while maintaining transparency and explainability, which are critical for industrial deployment.

The CoBT framework consists of several key components, as shown in Fig. 1 that collectively transform a single human demonstration into an adaptive robotic program. First, a user provides a demonstration, which is recorded as multivariate time-series data capturing the robot's end-effector movement, object positions, and gripper states. This data is then processed through a velocity-based change-point detection algorithm to segment the demonstration into discrete actions, as shown in Fig. 2 Top. Each segment is analyzed to extract symbolic states, which represent task constraints and motion preconditions. By mapping these symbolic states to Dynamic Movement Primitives (DMPs), CoBT ensures that the robot can adapt its motions to varying conditions without losing the overall task structure. Once the primitive actions are learned, they are composed into a BT, which enables the robot to execute the task reactively. The generated BT continuously evaluates task preconditions and dynamically selects actions based on real-time sensory feedback, ensuring that the robot can recover from errors and adapt to changes in the environment.

One of the advantages of CoBT is its ability to generate task plans that are both modular and hierarchical. The learned BTs can be adapted to new goals by modifying high-level task parameters without requiring a new demonstration. Additionally, multiple BTs can be combined to form composite task sequences, enabling robots to execute long-horizon tasks with minimal reprogramming. This modularity allows CoBT to scale across a wide range of robotic applications, from industrial assembly to household assistance. Furthermore, because BTs provide a transparent representation of task execution, they enhance user understanding and facilitate human-robot collaboration.

The effectiveness of CoBT has been validated through experimental evaluations on seven different robotic manipulation tasks, as shown in Fig. 3, including pick-and-place, insertion, drawer opening, pouring, and kitting. The results demonstrate that CoBT achieves a high success rate of approximately 93%, with an average programming time of just 7.5 seconds per task. Importantly, the framework exhibits strong reactivity, successfully adapting to variations in object placement and environmental conditions in real time. A pilot study conducted with non-expert users further highlights the accessibility and usability of CoBT, showing that individuals with no prior robotics experience can program robots effectively using this approach. Participants in the study reported low cognitive load and high ease of use, indicating that CoBT significantly reduces the complexity of robot programming. Video of the demonstrations and evaluation tasks are available at: https://youtu.be/uz768FNIAgM

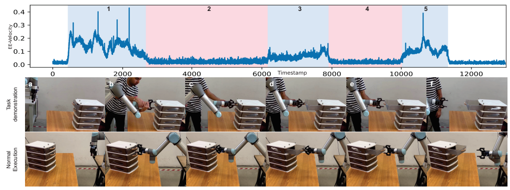

Fig. 2. (Top) Segmentation based on velocity and gripper state. (Middle) Transitions during the *drawer* task demonstration. (Bottom) a trial example of the generated policy under normal conditions.

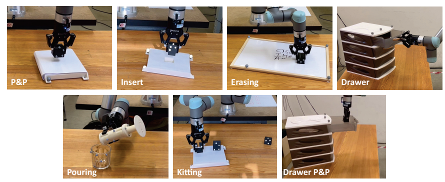

Fig. 3. 7 evaluation tasks that include mix of complex and P2P trajectory executions, and short and long-horizon tasks with multi-level goals.

3.2. *CoBT2.0 Framework*

CoBT has certain limitations. It struggles with handling spatial and semantic relationships beyond two objects, limiting its adaptability to new environments and varying object arrangements. The absence of structured spatial reasoning reduces flexibility and reactivity to changes in object positions. Additionally, without semantic knowledge, the generated programs can be ambiguous, leaving operators uncertain about how the system perceives entities, actions, and relationships. Integrating spatial scene graphs and semantic task knowledge enhances CoBT's ability to capture and reason about these relationships effectively.

To this end, CoBT2.0 is proposed to integrate spatial scene graphs and semantic task knowledge within the BTs. In particular, a single demonstration is used to generate action specific spatial scene graphs containing geometric constraints specifications of the action. The geometric constraints are used as action specific conditional nodes for the BTs where the dynamic scene graph is constantly analyzed. Further, the spatial knowledge and visual data from the demonstration is used to disambiguate the BT and generate semantic labels about the relationships of the entities interacting in the task. This approach enhances framework's ability to generalize, adapt during task execution, transfer learned behaviors to new environments, and improve system transparency and interpretability for operators.

CoBT2.0 generates dynamic scene graphs for each action in a demonstrated task. Nodes represent entities such as objects or the robot, and edges denote spatial relationships like proximity or movement. Using transformation matrices embedded in the edges, the algorithm captures detailed spatial relationships between entities. The graph is iteratively updated as actions progress, ensuring that the evolving spatial configurations are accurately represented. This structured representation allows the robot to generalize and adapt tasks based on real-world object configurations.

Further, the framework refines the spatial scene graphs by assigning semantic labels to entities, actions, and relationships. Using a Vision-Language Model (VLM), it generates descriptive labels based on the spatial data and pre-/post-condition images for each action. These labels provide clear insights into the robot's interactions, such as "Gripper grasps metal rod" or "Block near drill." The enriched scene graphs improve interpretability and transparency, enabling the operator to understand and validate the system's perception and reasoning effectively.

The BT generated by CoBT 2.0 executes tasks by integrating spatial scene graphs as conditional nodes, ensuring precise adherence to demonstrated spatial configurations. During execution, the BT sequences actions represented as motion primitives while continuously evaluating the current spatial relationships between entities against the stored scene graph constraints. Each action-specific node checks whether the real-time transformations of objects match the expected transformations stored in the graph. If the spatial constraints are satisfied, the BT allows the execution to proceed; otherwise, it triggers corrective actions or retries to align with the demonstrated conditions. This combination of scene graph-based conditional nodes and motion primitives enables adaptive, accurate, and robust task execution in dynamic environments.

3.3. *NsAI FDD Framework*

Autonomous robots face inevitable execution failures due to environmental changes, sensor inaccuracies, or mechanical issues. Traditional failure detection relies on predefined conditions set by experts, but this approach is unscalable for diverse failure modes. As industries embrace mass customization and agile manufacturing, robots must autonomously detect and handle failures. The key challenge is developing an adaptive system that learns from normal executions while ensuring interpretability and explainability.

A neuro-symbolic approach for failure detection integrates symbolic reasoning with sub-symbolic data-driven learning to address this challenge. The neuro-symbolic framework presented in this work combines raw sensory data from robot execution with symbolic task representations encoded in Behavior Trees (BTs). This hybrid approach leverages the strengths of both paradigms: neural networks effectively model complex, high-dimensional execution data, while BTs provide contextual information to the model. By combining these two elements, the system can detect failures by identifying deviations from normal execution patterns.

The proposed approach, as shown in Fig. 4 consists of two primary components: offline learning and online failure detection. In the offline learning phase, a LSTM-VAE is trained on nominal execution data. This model learns a compact latent representation of normal execution behavior using both sub-symbolic sensory inputs (e.g., robot joint states, forces, and object positions) and symbolic

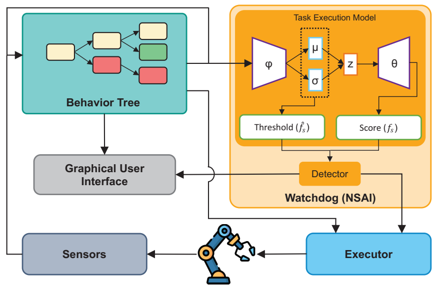

Fig. 4. Overview of the proposed neuro-symbolic AI failure detection and diagnosis.

task states extracted from the BT. The model reconstructs expected execution patterns, allowing it to identify discrepancies when encountering anomalous execution data. A failure score function is learned based on reconstruction errors, and a custom threshold function is defined to distinguish between normal and anomalous executions.

During the online failure detection phase, real-time execution data is processed through the trained LSTM-VAE model. The failure score is continuously monitored, and if it deviates beyond the learned threshold, the system triggers a failure event, stopping execution and alerting human operators. The BT structure provides a high-level understanding of execution status, allowing operators to trace failures back to specific sub-tasks, aiding in rapid diagnosis and recovery. This capability is particularly useful in industrial settings, where minimizing downtime is critical.

The neuro-symbolic failure detection approach was validated through a pick-and-place robotic task. The model was trained on 40 successful trials and refined with a threshold function from 20 additional trials. It was then tested under normal and failure conditions, including vision errors causing incorrect placements and unexpected tool collisions, as shown in Fig. 5. Results confirm the system's ability to detect failures in real time, with the failure score function distinguishing normal executions from anomalies. The threshold dynamically adapted to execution conditions, enabling timely execution halts and meaningful feedback via BT visualization. Compared to traditional methods relying on predefined thresholds or rule-based classifiers, the neuro-symbolic approach demonstrated superior adaptability, eliminating the need for domain-specific failure predicates.

4. Discussion and Conclusion

This paper presents a framework that embeds symbolic representations into robotic task execution through three frameworks: CoBT, CoBT 2.0, and NsAI FDD. Each framework leverages symbol grounding to bridge the gap between high-level task abstraction and real-world execution, enhancing flexibility in dynamic and unstructured environments.

CoBT leverages PbD to learn BTs from a single demonstration. The grounding of abstract task descriptions into structured BTs allows the robot to react dynamically to environmental changes, without the need for extensive manual programming. By decomposing a demonstration into symbolic task representations and associating them with executable actions, CoBT ensures that symbols retain meaning in varying contexts, making task adaptation more fluid and interpretable.

CoBT 2.0 enhances the CoBT framework by integrating spatial scene graphs, which provide a structured representation of object relationships and task constraints. Symbol grounding in this case is achieved through the dynamic encoding of spatial relationships within a scene, allowing the robot to reason about its surroundings. By continuously updating the scene graph during execution, the robot can ensure that symbolic representations remain contextually relevant, improving its ability to generalize across different task settings. This results in a higher level of spatial awareness, reducing errors in interactions and enabling more robust manipulation in cluttered environments.

NsAI FDD applies symbol grounding to failure detection and diagnosis by combining symbolic task states with data-driven failure detection. NsAI FDD grounds symbolic task states into the robot's execution model, allowing failures to be detected based on deviations from expected symbolic behavior rather than solely numerical failures. This approach improves detection capability and enables the system to provide meaningful diagnostics, making failure recovery more intuitive

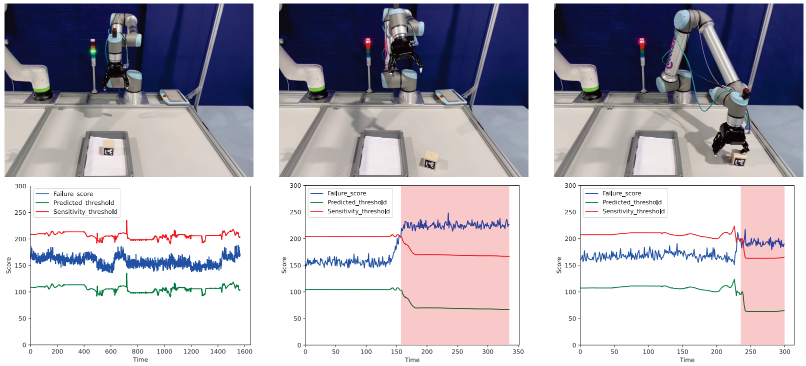

Fig. 5. NsAI FDD results under normal conditions (left), vision error (middle), and tool collision (right).

and effective.

The integration of symbol grounding across these three frameworks ensures that abstract knowledge is continuously mapped to real-world execution. This results frameworks that are not only adaptable but also intuitive, allowing human operators to better understand and trust robotic decision-making. By embedding symbolic reasoning at different stages of task execution—from initial programming to real-time monitoring and failure detection—this work demonstrates how symbol grounding can significantly enhance robotic autonomy. The preliminary experimental results validate the effectiveness of this approach, demonstrating significant improvements in task generalization, responsiveness to environmental changes, and failure detection accuracy.

State-of-the-art approaches to robot autonomy often rely on large datasets, brittle rule-based scripts, or domain-specific heuristics, which limit adaptability to changing environments and leave crucial gaps in transparency and real-time failure management. Many learning-based methods also neglect higher-level spatial and semantic relations, hampering a robot's ability to generalize from small or single demonstration sets. In contrast, this symbol-grounded autonomy framework integrates data-driven learning with declarative, symbolic reasoning to address these limitations more comprehensively. Thus, building on the shortcomings in existing approaches, this work provides a cohesive and more resilient solution for flexible, interpretable, and fail-safe robot autonomy in real-world scenarios.

While effective, CoBT relies on single demonstrations, limiting its ability to capture task variability. Future work will focus on incremental learning and integrating multiple demonstrations to enhance generalization. CoBT 2.0 is still under development, and future work will involve extensive validation across diverse tasks. Efforts will focus on optimizing data sampling rates and execution accuracy while exploring modular task design for greater flexibility and scalability. NsAI FDD excels at detecting failures but lacks a human-understandable explanation mechanism. Future research will integrate Large Language Models to generate natural language explanations for detected failures, aiding operator decision-making. Additionally, NsAI FDD will be evaluated against the new Regulation (EU) 2023/1230 on machinery safety, ensuring compliance with evolving autonomy and safety requirements while enhancing human-robot interaction through interpretable diagnostics.

Acknowledgement

This work is supported by the CISC project which is funded by the European Union's Horizon 2020 MSCA-ITN grant agreement no. 955901; ADAPT Research Centre which is funded by SFI Research Centers Programme and is co-funded under the European Regional Development Fund (ERDF) through Grant 13/RC/2106_P2; and EU Commission Recovery and Resilience Facility under the Science Foundation Ireland Future Digital Challenge Grant Number 22/NCF/FD/10929.

References

Akerkar, R. and P. Sajja (2009). *Knowledge-based systems*. Jones & Bartlett Publishers.

Aksoy, E. E., A. Abramov, J. Dörr, K. Ning, B. Dellen, and F. Wörgötter (2011). Learning the semantics of object–action relations by observation. *The International Journal of Robotics Research 30*(10), 1229–1249.

Chernova, S. and A. L. Thomaz (2014). Robot learning from human teachers. *Synthesis lectures on artificial intelligence and machine learning 8*(3), 1–121.

Colledanchise, M., D. Almeida, and P. Ögren (2019). Towards blended reactive planning and acting using behavior trees. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 8839–8845. IEEE.

Colledanchise, M. and P. Ögren (2018). *Behavior trees in robotics and AI: An introduction*. CRC Press.

Coradeschi, S. and A. Saffiotti (2003). An introduction to the anchoring problem. *Robotics and autonomous systems 43*(2-3), 85–96.

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

Eiband, T., C. Willibald, I. Tannert, B. Weber, and D. Lee (2023). Collaborative programming of robotic task decisions and recovery behaviors. *Autonomous Robots 47*(2), 229–247.

French, K., S. Wu, T. Pan, Z. Zhou, and O. C. Jenkins (2019). Learning behavior trees from demonstration. In *2019 International Conference on Robotics and Automation (ICRA)*, pp. 7791–7797. IEEE.

Garcez, A. d. and L. C. Lamb (2023). Neurosymbolic ai: The 3 rd wave. *Artificial Intelligence Review*, 1–20.

Grollman, D. H. and O. C. Jenkins (2010). Incremental learning of subtasks from unsegmented demonstration. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 261–266. IEEE.

Gustavsson, O., M. Iovino, J. Styrud, and C. Smith (2022). Combining context awareness and planning to learn behavior trees from demonstration. In *2022 31st IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1153–1160. IEEE.

Harnad, S. (1990). The symbol grounding problem. *Physica D: Nonlinear Phenomena 42*(1-3), 335–346.

Hundman, K., V. Constantinou, C. Laporte, I. Colwell, and T. Soderstrom (2018). Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 387–395.

Ijspeert, A. J., J. Nakanishi, H. Hoffmann, P. Pastor, and S. Schaal (2013). Dynamical movement primitives: learning attractor models for motor behaviors. *Neural computation 25*(2), 328–373.

Jain, A., P. Long, V. Villani, J. D. Kelleher, and M. Chiara Leva (2024). Cobt: Collaborative programming of behaviour trees from one demonstration for robot manipulation. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 12993–12999.

Jain, A., S. Mehak, P. Long, J. D. Kelleher, M. Guilfoyle, and M. C. Leva (2022). Evaluating safety and productivity relationship in human-robot collaboration. *Health 2022*, 08–28.

Khalastchi, E. and M. Kalech (2018). On fault detection and diagnosis in robotic systems. *ACM Computing Surveys (CSUR) 51*(1), 1–24.

Knaust, M. and D. Koert (2021). Guided robot skill learning: A user-study on learning probabilistic movement primitives with non-experts. In *2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids)*, pp. 514–521. IEEE.

Kroemer, O., C. Daniel, G. Neumann, H. Van Hoof, and J. Peters (2015). Towards learning hierarchical skills for multi-phase manipulation tasks. In *2015 IEEE international conference on robotics and automation (ICRA)*, pp. 1503–1510. IEEE.

Kroemer, O., S. Niekum, and G. Konidaris (2021). A review of robot learning for manipulation: Challenges, representations, and algorithms. *The Journal of Machine Learning Research 22*(1), 1395–1476.

Kyrarini, M., M. A. Haseeb, D. Ristić-Durrant, and A. Gräser (2019). Robot learning of industrial assembly task via human demonstrations. *Autonomous Robots 43*, 239–257.

Lentini, G., G. Grioli, M. G. Catalano, and A. Bicchi (2020). Robot programming without coding. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 7576–7582. IEEE.

Mitrevski, A., P. G. Plöger, and G. Lakemeyer (2023). A hybrid skill parameterisation model combining symbolic and subsymbolic elements for introspective robots. *Robotics and Autonomous Systems 161*, 104350.

Nejati, N., P. Langley, and T. Konik (2006). Learning hierarchical task networks by observation. In *Proceedings of the 23rd international conference on Machine learning*, pp. 665–672.

Ni, Z., X. Deng, C. Tai, X. Zhu, Q. Xie, W. Huang, X. Wu, and L. Zeng (2023). Grid: Scene-graph-based instruction-driven robotic task planning. *arXiv preprint arXiv:2309.07726*.

Osa, T., J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, J. Peters, et al. (2018). An algorithmic perspective on imitation learning. *Foundations and Trends® in Robotics 7*(1-2), 1–179.

Park, D., Y. Hoshi, and C. C. Kemp (2018). A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robotics and Automation Letters 3*(3), 1544–1551.

Suddrey, G., B. Talbot, and F. Maire (2022). Learning and executing re-usable behaviour trees from natural language instruction. *IEEE Robotics and Automation Letters 7*(4), 10643–10650.

Sünderhauf, N., O. Brock, W. Scheirer, R. Hadsell, D. Fox, J. Leitner, B. Upcroft, P. Abbeel, W. Burgard, M. Milford, et al. (2018). The limits and potentials of deep learning for robotics. *The International journal of robotics research 37*(4-5), 405–420.

Villani, V., F. Pini, F. Leali, and C. Secchi (2018). Survey on human–robot collaboration in industrial settings: Safety, intuitive interfaces and applications. *Mechatronics 55*, 248–266.

Zanchettin, A. M. (2023). Symbolic representation of what robots are taught in one demonstration. *Robotics and Autonomous Systems 166*, 104452.