Addressing Failures in Robotics using Vision-Based Language Models (VLMs) and Behavior Trees (BT)[⋆]

Faseeh Ahmad[1], Jonathan Styrud[2] and Volker Krueger[1]

[1] Lund University, Lund, Sweden,
{faseeh.ahmad, volker.krueger}@cs.lth.se
[2] KTH Royal Institute of Technology, Stockholm, Sweden
jstyrud@kth.se

Abstract. In this paper, we propose an approach that combines Vision Language Models (VLMs) and Behavior Trees (BTs) to address failures in robotics. Current robotic systems can handle known failures with pre-existing recovery strategies, but they are often ill-equipped to manage unknown failures or anomalies. We introduce VLMs as a monitoring tool to detect and identify failures during task execution. Additionally, VLMs generate missing conditions or skill templates that are then incorporated into the BT, ensuring the system can autonomously address similar failures in future tasks. We validate our approach through simulations in several failure scenarios.

Keywords: Robotics, Failure Detection, Behavior Trees, Vision Language Models, Recovery Behaviors

1 Introduction

Modern robotic systems can handle complex tasks in controlled environments, but transitioning into dynamic, small-batch manufacturing introduces new challenges, especially around failure management. Failures; unforeseen disturbances that prevent task completion; can lead to costly delays and risks, particularly in shared workspaces [1]. The ability to detect, identify, and recover from failures autonomously is crucial for ensuring the robustness of robotic systems.

Traditional failure management strategies in robotics include human intervention, failure analysis [2], and automated recovery strategies [1]. These approaches have limitations: human intervention is time-consuming, failure analysis requires expertise, and automated strategies often lack flexibility in handling unforeseen scenarios. Our recent work [3] introduced a novel method using automated recovery behaviors modeled as robotic skills with parameters, preconditions, and postconditions, executed through Behavior Trees and Motion Generators (BTMG) policy representation [4]. This approach optimizes recovery policies using Reinforcement Learning (RL) [5] and also adapts the parameters to different task variations [6].

However, two key limitations remain: (1) the system assumes failure detection and identification are already solved, requiring prior knowledge of the failure, and (2) it only handles known failures with predefined solutions. These limitations make it difficult to address unforeseen failures. We propose addressing these gaps by utilizing Vision Language Models (VLMs) to detect, identify, and generate solutions for unknown failures. By integrating VLMs with Behavior Trees (BTs), our approach autonomously monitors task execution, identifies failure states, and generates missing conditions or skill templates to recover from failures. The BT is then updated using a reactive planner [7] to handle similar future occurrences.

Main Contributions

- We propose a novel integration of VLMs with BTs for monitoring, failure detection, identification and recovery in robotic systems.
- We use VLMs to generate missing preconditions or skill templates to address failures and update the BT policy.
- We conduct experiments to demonstrate the effectiveness of the approach.

2   Background

This section provides essential background concepts to our proposed approach, focusing on behavior trees, reactive planner and vision-based language models.

2.1   Behavior Trees (BT)

Behavior Trees (BTs) are hierarchical models for task execution, known for their modularity and flexibility [8]. A BT organizes task execution through nodes that receive tick signals, indicating readiness for execution. BTs consist of two types of nodes: control-flow nodes and execution nodes. *Control-flow nodes* manage execution flow and return statuses of success, failure, or running; examples include Sequence (AND) and Fallback/Selector (OR). *Execution nodes*, which are leaf nodes, represent either skills ("!") or conditions ("?"). Skills perform specific tasks, while conditions evaluate the environment, returning only success or failure. BTs offer modularity and clarity, making them ideal for robotics applications, particularly in dynamic environments where flexibility is required [9].

2.2   Reactive Planner

Reactive planners generate BTs dynamically, using a backchaining approach to select skills that satisfy goal conditions [10]. The process iteratively selects skills based on their preconditions and postconditions, constructing a BT that satisfies the specified goal. This recursive process continues until a leaf node is reached or a predefined depth is attained. The planner ensures adaptability by dynamically responding to changes in the environment without requiring extensive re-planning. This planner has been extended for various applications in [11,12].

2.3 Vision Language Models (VLM) in Robotics

Vision Language Models (VLMs) are powerful tools in robotics, enabling a deeper understanding of complex environments by combining visual data with language inputs. VLMs excel at tasks such as scene interpretation, object recognition, and generating control skills based on visual cues and task descriptions [13, 14]. Recent applications of VLMs in robotics include failure recovery, task planning, and multimodal reasoning, with systems like ReplanVLM [15] and AHA [16] demonstrating their ability to reason over failures and generate dynamic solutions.

3   Approach

We extend the existing framework [3] to handle unknown failures by integrating VLMs for failure detection, identification and recovery and generate missing preconditions or skill templates to be incorporated into the BT.
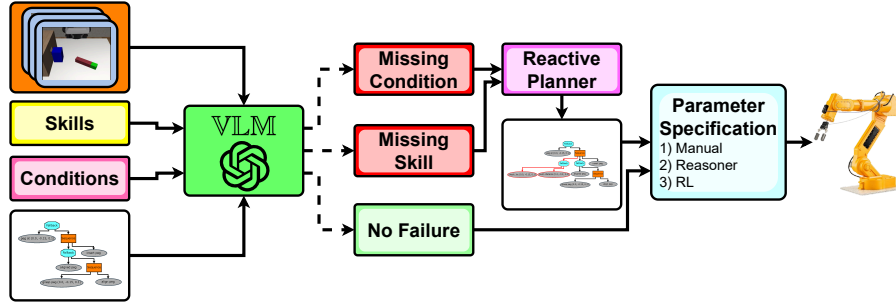

Fig. 1. Overview of the proposed approach, where the VLM takes a set of images, skills, conditions, and a BT as input. The VLM uses this information to provide missing conditions or skills, which are then used to update the BT through a planner.

3.1   Failure Detection and Idenitification

Failure detection refers to the system's ability to recognize when a task cannot be completed due to unforeseen errors, such as hardware malfunctions or environmental disturbances. This can be achieved by using sensor data, such as from cameras or force-torque sensors, and comparing it against the expected postconditions of skills. For example, in a peg-in-hole task, if an object blocks the hole, the system detects this failure when the postcondition of the "insert" skill (peg inserted) is not met [1] (see Figure 3).

Failure identification involves describing the failure using existing system conditions and understanding why the task could not be completed. For instance, in the peg-in-hole task, the missing precondition when an obstacle is blocking the hole could be "Not any obstacle at hole" for the insert skill. This allows the system to formulate strategies for dealing with similar failures in the future.
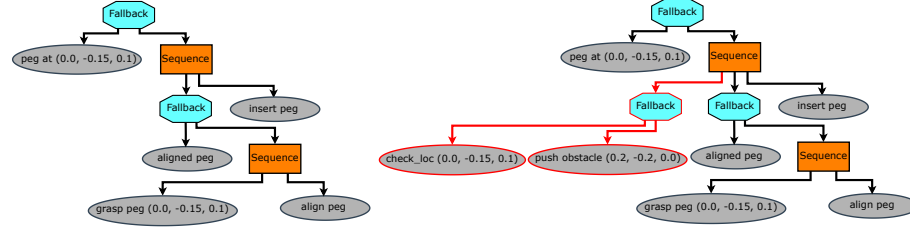
Fig. 2. Comparison of Behavior Trees (BTs). The left side shows the initial BT, while the right side illustrates the updated BT, with the changes highlighted in red connections.
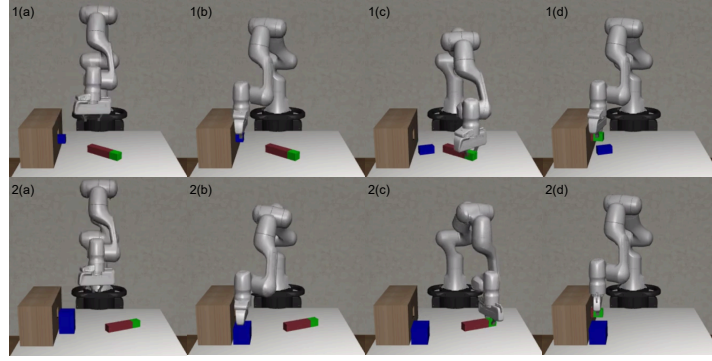
Fig. 3. Scenes showing peg-in-hole task execution with obstacles. The first row (1(a)–1(d)) illustrates the task with a small obstacle, while the second row (2(a)–2(d)) depicts the task with a large obstacle.

3.2   Monitoring using VLM

We use VLMs to enable failure detection, identification and recovery (Figure 1). Before task execution, the VLM is queried with images of the task environment, the BT structure, and the skills and conditions involved. The VLM then assesses whether the task will succeed and, if not, identifies the cause of failure (detection). It also suggests the missing condition (identification) that could prevent the failure. If the system lacks the required skill to recover from a failure (recovery), the VLM suggests an appropriate recovery skill based on the provided library of existing skills (see Figure 2).

3.3   Condition and Skill Template Generation

When the VLM identifies a missing condition or skill, it updates the BT accordingly. For example, if a small obstacle blocks the peg, the condition "hole free" is generated and added as a precondition to the "insert" skill. The reactive planner regenerates the BT by incorporating this condition, ensuring similar failures are handled in the future [7]. If a recovery skill is missing, the VLM generates an

skill template that follows a structured format and requires some manual inputs to complete. For instance, if a large object blocks the peg hole and the gripper cannot grasp it, the VLM suggests a "push" skill template to remove the obstacle (see Figure 3). This template is added to the BT, and in the future, this process could be fully automated, allowing the system to autonomously recover from failures.

4 Experiments

We validated the proposed approach using simulations in robosuite [17] and OpenAI's GPT-4. The experiments were designed around three tasks, each involving unknown failures:

- *Peg-in-Hole Task:* Two scenarios—(A) a small obstacle inside the hole, and (B) a large obstacle positioned in front of the hole.
- *Lift Task:* An additional cube is placed on top of the target object, creating an unforeseen failure.
- *Door Opening Task:* The robot attempts to open a door but lacks the precondition that the handle must be turned first.

5 Evaluation Metrics and Results

We evaluated the VLM's performance using three key metrics: consistency in failure detection and recovery, the importance of vision input, and skill feasibility considerations (ensuring that suggested skills, such as a "grasp" skill, are feasible based on the gripper's affordance and object location). For all experiments, we used model parameters of temperature and top_p set to 0.1, which resulted in more deterministic and focused outputs, reducing randomness and ensuring that the model consistently chose the most likely responses.

- **Consistency of Failure Detection and Recovery:** The VLM's reliability was tested across 10 trials per scenario, consistently detecting and recovering from failures in all tasks, achieving 100% consistency.
- **Vision Importance Ablation Study:** To assess the impact of visual input, we compared VLM (with visual input) and LLM (without visual input). In the *Peg-in-Hole (small obstacle)*, *Lift*, and *Door Opening* tasks, both models achieved 100% accuracy. However, in the *Peg-in-Hole (large obstacle)* task, the VLM achieved 100% accuracy, while the LLM reached 30% accuracy without skill feasibility considerations and 60% with feasibility checks.
- **Skill Feasibility Considerations:** When skill feasibility is considered, LLM performance improved but still fell short of the VLM. The VLM excelled in complex scenarios like the *Peg-in-Hole (large obstacle)* task, generating feasible recovery skills autonomously.

6   Conclusion and Future Work

In this paper, we introduced a method for integrating Vision Language Models (VLMs) with Behavior Trees (BTs) to autonomously detect, identify, and recover from failures in robotic systems. By generating missing conditions and skill templates, the system can effectively handle unknown failures and adapt its execution policy for future tasks. Future work will focus on several key areas: expanding the range of failure scenarios to include more complex and dynamic environments, improving the skill generation mechanism to move from generating skill templates to directly producing feasible skills, thereby reducing manual input. Additionally, we aim to fine-tune an open-source model to further enhance the system's performance and adaptability across diverse robotic tasks.

References

1. R. Wu et al. Automated behavior tree error recovery framework for robotic systems. In *IEEE ICRA 2021*.
2. F. Jusuf et al. Review on defenses against common cause failures on digital safety systems. In *AIP Conference Proceedings 2021*.
3. F. Ahmad et al. Adaptable recovery behaviors in robotics: A behavior trees and motion generators (btmg) approach for failure management. In *IEEE CASE 2024*.
4. F. Rovida et al. Extended behavior trees for quick definition of flexible robotic tasks. In *IEEE/RSJ IROS 2017*.
5. M. Mayr et al. Skill-based multi-objective reinforcement learning of industrial robot tasks with planning and knowledge integration. In *IEEE ROBIO 2022*.
6. F. Ahmad et al. Learning to adapt the parameters of behavior trees and motion generators (btmgs) to task variations. In *IEEE/RSJ IROS 2023*.
7. J. Styrud et al. Automatic behavior tree expansion with llms for robotic manipulation. *arXiv preprint arXiv:2409.13356*, 2024.
8. M. Colledanchise and P. Ögren. *Behavior Trees in Robotics and AI: An Introduction*. CRC Press 2018.
9. M. Iovino et al. A survey of behavior trees in robotics and ai. *Robotics and Autonomous Systems 2022*.
10. M. Colledanchise et al. Towards blended reactive planning and acting using behavior trees. In *IEEE ICRA 2019*.
11. M. Iovino et al. A framework for learning behavior trees in collaborative robotic applications. In *IEEE CASE 2023*.
12. J. Styrud et al. Bebop-combining reactive planning and bayesian optimization to solve robotic manipulation tasks. In *IEEE ICRA 2024*.
13. H. Chen et al. Automating robot failure recovery using vision-language models with optimized prompts. *arXiv preprint arXiv:2409.03966*, 2024.
14. M. J. Kim et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.
15. A. Mei et al. Replanvlm: Replanning robotic tasks with visual language models, 2024. Available at: https://arxiv.org/abs/2407.21762.
16. J. Duan et al. Aha: A vision-language-model for detecting and reasoning over failures in robotic manipulation, 2024. Available at: https://arxiv.org/abs/2410.00371.
17. Robosuite: A framework for robot learning, 2023.