

Learning to Capture Rocks using an Excavator: A Reinforcement Learning Approach with Guiding Reward Formulation

Amirmasoud Molaei^{a,*}, Mohammad Heravi^a, Reza Ghabcheloo^a

^a*Faculty of Engineering and Natural Sciences, Tampere University, Tampere, 33720, Finland*

Abstract

Rock capturing with standard excavator buckets is a challenging task typically requiring the expertise of skilled operators. Unlike soil digging, it involves manipulating large, irregular rocks in unstructured environments where complex contact interactions with granular material make model-based control impractical. Existing autonomous excavation methods focus mainly on continuous media or rely on specialized grippers, limiting their applicability to real-world construction sites. This paper introduces a fully data-driven control framework for rock capturing that eliminates the need for explicit modeling of rock or soil properties. A model-free reinforcement learning agent is trained in the AGX Dynamics® simulator using the Proximal Policy Optimization (PPO) algorithm and a guiding reward formulation. The learned policy outputs joint velocity commands directly to the boom, arm, and bucket of a CAT®365 excavator model. Robustness is enhanced through extensive domain randomization of rock geometry, density, and mass, as well as the initial configurations of the bucket, rock, and goal position. To the best of our knowledge, this is the first study to develop and evaluate an RL-based controller for the rock capturing task. Experimental results show that the policy generalizes well to unseen rocks and varying soil conditions, achieving high success rates comparable to those of human participants while maintaining machine stability. These findings demonstrate the feasibility of learning-based excavation strategies for discrete object manipulation without

*Corresponding author

Email address: amirmasoud.molaei@tuni.fi (Amirmasoud Molaei)

requiring specialized hardware or detailed material models.

Keywords: Excavators, Automatic rock capturing, Reinforcement learning, High-fidelity simulation, Guiding Reward Formulation, Non-prehensile manipulation

1. Introduction

Autonomous excavation holds a great promise in addressing increasing demands of the mining and construction industries, two of the largest and most essential sectors worldwide. The excavator is one of the most widely used and versatile heavy-duty mobile machines (HDMMs), which is typically operated through a hydraulic system. Excavators are utilized for a wide range of earth-moving tasks, including digging, trenching, grading, and in particular material handling. Despite their versatility, traditional manual operation of excavators can result in low efficiency, increased physical strain on operators, and exposure to hazardous environments like open-pit mines. These challenges underscore the need for automation to enhance safety and productivity. An excavator is primarily composed of three major components, the traveling body, swing body, and the front digging manipulator. The digging manipulator, includes three main parts, boom, arm, and bucket, which are actuated by hydraulic cylinders. Additionally, joints connect the swing body, boom, arm, and bucket, allowing for flexible and precise motion [1, 2, 3, 4].

Bulk material handling is effectively performed by HDMMs, particularly excavators. In large-scale open-pit mining, where ledges are blasted, or on construction sites where rocks are blocked the working area or buried in soil, excavators often face the challenge of picking individual large rocks mixed with finer gravel and soil. Collecting these large rocks and transporting them to dump trucks or other destinations requires high degree of expertise, typically only experienced operators have [5, 1, 6, 7].

Autonomous rock capturing presents unique challenges due to the different behavior of large rocks compared to homogeneous materials. While homogeneous materials can often be excavated in a predefined scooping motion, rocks must be individually and precisely captured using the bucket, requiring a different control strategy. Figure 1 illustrate the rock capturing task. A major difficulty lies in the complex and poorly understood interactions between the bucket and surrounding materials, especially when both soil and rock are present [8]. Modeling these interactions using physical models,

such as terra-mechanics models [9], is highly challenging due to unknown and variable factors such as rock mass, geometry, friction, and material properties. Moreover, the terrain may be uneven, and critical parameters related to the material are often unavailable or difficult to estimate accurately [10, 11]. Even with accurate knowledge of these parameters, designing controllers in the traditional way remains highly challenging. Therefore, machine learning algorithms, such as Reinforcement Learning (RL), can be a promising solution to this complex task.

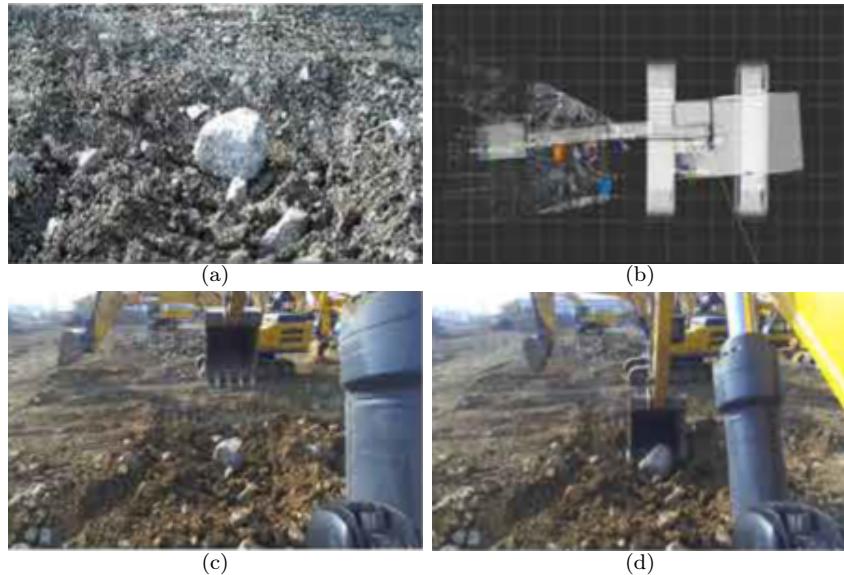


Figure 1: Rock capturing task using an excavator [7].

Identifying the similarities between robotic manipulation and rock excavation offers valuable insight into the rock capturing task, indicating that techniques developed in industrial manipulator literature can be applied to this task. Capturing a rock using a bucket has similarities to a non-prehensile manipulation task in the presence of granular material [1]. In this task, the bucket interacts with the environment to guide the rock along a desired trajectory. Research in non-prehensile manipulation has made significant progress in recent years, with several studies extensively investigating RL approaches for acquiring such skills [12, 13, 14].

RL techniques are based on the concept of an agent, a program that interacts with the environment by performing actions aimed at maximizing a predefined cumulative reward. Through continuous exploration, the agent

gathers information from observations and receives feedback in the form of rewards, which guide its learning process. Positive rewards reinforce desirable behavior, while negative rewards discourage incorrect actions. Moreover, RL agents can be retrained for new patterns and enabling further improvements [15]. Training a control policy using RL demands a large number of interaction samples, which are often impractical to collect using physical robots. To address this, physics-based simulators, such as Bullet Physics [16] and MuJoCo [17], are commonly used as training environments for RL agents. However, these simulators generally face challenges in accurately representing granular materials [18]. It can cause a significant challenge regarding training an efficient control policy for the rock capturing task, since the interactions of granular materials with the bucket and rock are highly essential.

In this paper, a learning-based approach is proposed to automatically capture large rocks using an excavator bucket, without relying on any explicit knowledge of the rock geometry or mass or surrounding material properties. A control policy is trained using model-free RL within the high-fidelity AGX Dynamics[®] simulation environment. Prior research has shown that AGX Dynamics[®] offers the realism and performance necessary for simulating complex earth-moving tasks involving granular media and rigid body interactions [19, 20, 21]. In our framework, a Proximal Policy Optimization (PPO) algorithm is employed alongside guiding reward formulation that encourages successful rock manipulation while maintaining machine stability. To enhance the robustness and generalizability of the learned policy, key parameters such as rock geometry, density, mass, initial positions of the rock and bucket, and goal position are extensively randomized during training. The trained policy is evaluated under various conditions, including scenarios with previously unseen rock shapes and different material properties, and demonstrates generalization capabilities. Results indicate that the controller can adapt its strategy effectively to novel situations and achieves faster task completion compared to human participants.

The rest of this paper is structured as follows. Section 2 provides a review of the literature. Section 3 outlines the proposed approach, and the results are presented in Section 4. Section 5 discusses the characteristics of the proposed method, and the conclusions are provided in Section 6.

2. Literature Review

Over the past several years, many studies have explored the automation of excavators, with particular attention given to the bucket-filling task. Nevertheless, autonomous excavators still fall short of matching the performance and flexibility of skilled human operators, which continues to limit their widespread adoption in industries [22, 2]. In addition, the rock capturing task, a challenging and safety-critical operation, has largely been overlooked in the literature [1].

A straightforward approach to automate the bucket-filling task involves designing a predefined bucket trajectory through the soil [23, 24, 25]. While this method performs well under uniform soil conditions, it tends to fail when those conditions vary [2]. The excavation process can be divided into several steps, such as penetration, dragging, scooping, and lifting. To minimize manual adjustments, in [26], a controller is proposed that automatically transitions from dragging to scooping for more accurate bucket filling. However, essential parameters, especially the excavation depth of the trajectory, still require manual tuning for a specific soil type. To address it, in [27], a method is suggested to adjust the bucket height during the dragging phase to apply maximum power to the soil, improving efficiency and preventing stalling. Despite this, other parts of the trajectory still require manual calibration. In [28], a Model Predictive Control (MPC) approach is introduced to follow the desired shape. This approach utilizes a soil-bucket model based on the Koopman theory, trained on data collected from excavation experiments involving a single soil type. However, this controller requires precise force control, which demands costly modifications to the hydraulic system [29]. Furthermore, the extension of this method to handle various soil types remains an open question for future research. In [30, 31], by defining a bucket-force trajectory for each excavation phase, a soil-adaptive algorithm is obtained across different soils. It could accurately excavate a target shape by switching to position control once the bucket reaches the desired height. Accurate force control is obtained by replacing the standard main-stage valves with high-performance servo valves. In [32], an impedance control is employed to track the desired bucket force to prevent stalling while following a predefined bucket trajectory. However, this results in incomplete bucket filling when the soil is harder than expected. This issue is addressed in [33] by proposing an iterative learning controller with a disturbance observer. The method assumes similar soil responses, which does not always hold true. In [34], an

Echo-State Network (ESN) is used to learn an inverse model of an excavator for trajectory tracking in repetitive digging tasks. The ESN is initially pretrained with a conventional Proportional-Derivative (PD) controller and then updated online during operation to improve performance under varying conditions. While the controller is able to adapt over time, it requires multiple digging cycles and assumes consistent soil conditions.

Rule-based approaches are another strategy to prevent stalling, typically by implementing predefined corrective actions when interaction forces become excessive [35] or when tracking errors grow too large [36]. Some studies have investigated this concept by creating libraries of motion primitives, which are then selected or combined using rule-based logic to perform autonomous excavation tasks [37, 38, 39, 40, 41]. Although such methods have been successfully implemented on real machines, developing the motion primitives demands significant engineering work and deep domain expertise, particularly for handling more complex operations [2].

To move beyond manually designed bucket trajectories, Trajectory Optimization (TO) methods have been widely explored for autonomous excavation. Generally, kinematic-based approaches optimize objective functions, including the volume of scooped soil, time, and motion smoothness [42, 43]. Although these methods can consider both the current and target terrain elevations and are suitable for real-time applications, they fall short in ensuring trajectory feasibility since they neglect soil reaction forces. To address it, a dynamics-aware MPC framework with a disturbance observer is introduced in [44] to follow a kinematically optimized trajectory and demonstrated in simulation. However, when disturbances arise, such as encountering soil that is harder than expected, the system diverges from the intended path, leading to incomplete bucket filling, as the trajectory is not updated in real time. Other TO methods incorporate soil properties by embedding soil models into the optimization process, which significantly increases computational complexity, making real-time execution impractical [45, 46, 47, 48]. To reduce the computational time in deployment, in [46], an approach is suggested distilling TO results into a neural network, which enables fast inference within milliseconds. A common limitation of TO approaches, that rely on soil models, is the need for prior knowledge of soil parameters. These parameters can be obtained through time-consuming geological site inspection [48], or by optimizing model parameters to minimize the error between predicted and actual soil forces during either manual operation [45], or laboratory experiments [49]. In [50], a supervised learning approach is used to estimate soil

parameters directly from measured resistance data while excavating known soil types. However, the accuracy of these methods depends heavily on the quantity and quality of the training data, which are time-intensive to gather. Additionally, it is still uncertain whether these models can generalize across different machines.

Rather than relying on TO to generate excavation trajectory, another line of research utilizes expert demonstrations. These demonstrations are either reparameterized [51] or optimized using stochastic methods [52] to achieve kinematic goals such as speed or smoothness. However, since these approaches neglect soil reaction forces, they risk causing the excavator to stall. One way to address it is by imposing a force threshold and restricting the excavation area that can result in inefficient excavation [53]. In [54], a demonstration-based method is introduced to learn dynamic motion primitive parameters for replicating human excavation trajectories, featuring online adaptation to adjust excavation depth and prevent excessive forces or stalling. However, because the trajectory endpoint is fixed, the approach frequently leads to insufficient bucket filling, especially in harder soils where interaction forces are greater [55]. The proposed methods in [56, 57] involve training visual prediction models of the excavation scene using Convolutional Neural Network (CNN) and data collected from real machines and simulations. The model is then used in sampling-based optimization to determine suitable actions for reaching a desired state of scene without accounting for interaction forces. In [58], a method is proposed using imitation learning to automate excavation by classifying and adapting expert-demonstrated motions. However, the approach depends on a large database of excavation, which is challenging to gather in real-world. In [59], an imitation learning approach is developed capable of utilizing non-optimal demonstrations. The method is applied to an excavator model operating in granular soil. A persistent challenge across these techniques is their heavy reliance on the quantity, quality, and diversity of demonstration data, raising questions about their transferability and effectiveness across different excavators [2].

To eliminate the need for costly and time-consuming data collection on real machines, RL in simulation offers a promising solution. In [13], an offline RL algorithm is applied on a Franka robot, introducing a cost function to discourage large reaction forces in the optimization of expert demonstrations. While this method generally results in trajectories requiring less torque, it does not guarantee the feasibility of the trajectory. Similarly, in [60] an offline RL is employed to minimize forces while maximizing bucket depth in the

penetration phase. In [61], an RL policy is trained in simulation for excavating rigid objects using a Franka Panda arm, relying on visual input from the excavation scene. However, due to the absence of soil interaction forces, the policy frequently failed during deployment because of discrepancies between simulation and real-world conditions. In [62], an RL controller is trained to push against a rock and activate an excavator-mounted hammer tool for rock breaking. The sim-to-real gap is addressed by incorporating actuation delays in the simulation and applying filtering to sensor measurements during deployment. In [63], RL is also applied to real-time energy management in a hybrid excavator, where it regulates energy flow and enhance efficiency. In [64], the PPO algorithm is used to train a neural network agent to perform a surface leveling task. The simulation is performed using the Dynasty simulation engine (proprietary software by Caterpillar Inc.), while the transfer to the real world is left for future work. In [15], RL is used to train a neural network controller using multibody excavator model. In [6], RL is used to precisely control an excavator manipulator for the grading task. To reduce the sim-to-real gap and reach competitive accuracy, the simulation accuracy is enhanced with an actuator model trained using real data. In [55], a simple analytical soil model based on the Fundamental Equation of Earth-moving (FEE) [65] is employed to train an RL controller.

Given the substantial sample requirements for RL training, these controllers are typically trained in simulation before being transferred to physical hardware. Moreover, testing with HDMMs, such as excavators, is both costly and potentially hazardous, making simulation-based experiments a crucial preliminary step. Popular examples of these simulators include Bullet Physics, MuJoCo, Open Dynamics Engine (ODE), NVIDIA PhysX, and Havok. While simulators like PhysX, Havok, and ODE often lack adequate precision for high-accuracy applications [66], Bullet Physics and MuJoCo also face limitations when modeling interactions involving soft or granular materials [18]. Due to the difficulty of achieving sufficiently accurate simulations, successful sim-to-real transfer for conventional construction machinery remains rare and has not yet been demonstrated in studies such as [15, 67, 68]. One potential solution to these challenges is the use of high-fidelity physics-engine simulators. Tools like AGX Dynamics[®] can simulate the excavation process with high accuracy. In [69], a multi-scale terrain simulation method based on AGX Dynamics[®] is shown to replicate excavation forces and soil displacement with an error margin of 10-25%, while still achieving real-time performance. In [20], a simulation platform called TERA, built with Unity3D

and AGX Dynamics[®], is developed to accurately model excavator–terrain interaction and enable scalable simulations. In [21], a simulation-to-reality discrepancy of roughly 10% has been reported in bucket-filling tasks. Also, in [70], AGX Dynamics[®]-based simulations integrated with world models is used to optimize sequential loading operations. Collectively, these results have demonstrated that AGX Dynamics[®] delivers the precision required for simulating excavation processes.

Capturing a rock using a bucket of an excavator is considered a non-prehensile manipulation task in the presence of granular material within the robotics literature [71]. In this task, the bucket, which is functioning as a non-prehensile end-effector, must move the rock by interacting with the surrounding environment. Actually, the interaction between the bucket and the rock can be transmitted indirectly through the surrounding material. In addition, the behavior of materials introduces considerable complexity and is inherently challenging to model. Moreover, the motion of the rock does not follow a flat ground plane due to the uneven and dynamically changing soil conditions [1]. Although the excavation of individual rocks using excavators is recognized as a challenging task in [10, 8, 11], only two approaches have been introduced in [37, 1]. In [37], a trial-and-error method is proposed, where the rock is removed by digging at progressively deeper depths. The approach relies on a set of if-then rules designed to mimic skilled operators and is calibrated on a one-fifth-scale LUCIE prototype operating mostly in homogeneous soil. Rule-based controllers often lack robustness and adaptability to varying conditions, and the influence of different soil types is not explored. In [1], an optimal control approach is proposed to minimize the distance between the rock and the bucket. A Gaussian Process (GP) modeling technique combined with an Unscented Kalman Filter is used to model the rock’s motion dynamics. The approach is implemented on a laboratory-scale UR10e manipulator equipped with a 3D-printed bucket to emulate an excavator equipped with a bucket, and its robustness to varying material properties is not examined.

2.1. Contributions

This work introduces a RL-based control strategy for automating the rock capturing task with an excavator. Conventional model-based approaches are hindered by the complexity of unstructured environments and the highly dynamic interactions between the bucket, rock, and surrounding granular material. In contrast, our method leverages model-free learning and extensive

randomization to achieve robust, generalizable, and efficient control. The main contributions are as follows:

Data-driven control without explicit material modeling: The proposed approach eliminates the need for analytical models of rock or soil properties. A PPO agent is trained entirely in simulation using excavator state variables and rock position information, directly generating joint speed commands for the excavator.

Robustness through extensive domain randomization: Key parameters—including rock geometry, density, and mass, as well as the initial configurations of the rock, bucket, and goal—are randomized during training. This allows the controller to adapt effectively to unseen rock shapes and material properties during evaluation.

Stable and efficient performance: The learned controller achieves a success rate of 0.8, comparable to human participants, while maintaining machine stability and avoiding excessive tilting. Once trained, the policy executes through a lightweight neural network forward pass, enabling real-time deployment in excavation tasks.

3. Methodology

In this section, an approach is described to learn a control policy entirely in simulation for automatic rock capturing using an excavator. In our framework, the term agent refers to the control policy, while the environment represents the excavator, rock, and terrain. A simple schematic of the proposed method is shown in Fig. 2. First, the basic of RL is explained, then the modeling of the problem as a Markov Decision Process is elaborated.

3.1. Goal-Conditioned Reinforcement Learning

The problem is framed within the context of goal-conditioned RL, specifically as a finite horizon goal-conditioned Partially Observable Markov Decision Process (POMDP), represented by the tuple $(\mathcal{S}, \Omega, \mathcal{G}, \mathcal{A}, O, P, r, H, \rho_0, \rho_g)$. At each time step t , the environment is in a state $s_t \in \mathcal{S}$, an observation $o_t \in \Omega$ is obtained, a goal $g_t \in \mathcal{G}$ is considered, and an action $a_t \in \mathcal{A}$ is executed. It is important to note that the goal remains fixed throughout the duration of an episode. Additionally, the observation model $O : \mathcal{S} \times \mathcal{A} \rightarrow \text{Pr}(\Omega)$ defines the probability distribution over observations, the transition function $P : \mathcal{S} \times \mathcal{A} \rightarrow \text{Pr}(\mathcal{S})$ characterizes the system’s dynamics, and the reward function $r : \mathcal{S} \times \mathcal{A} \times \mathcal{G} \rightarrow \mathbb{R}$ specifies the feedback signal based on the

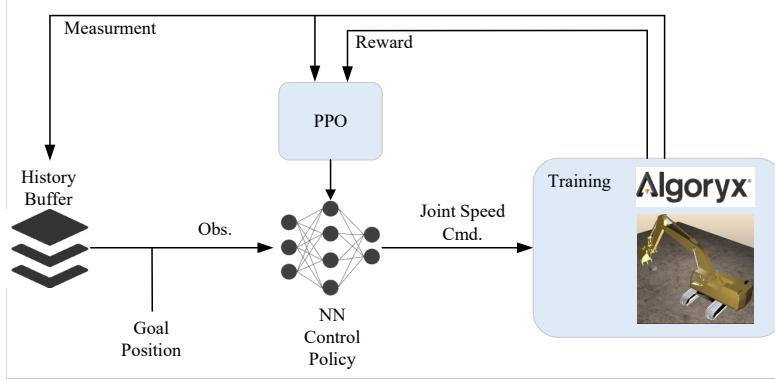


Figure 2: The pipeline for training of the control policy.

state, action, and goal. Each episode is limited to a maximum length of H time steps. The initial state and goal for each episode are sampled from the distributions ρ_0 and ρ_g , respectively [72, 73, 12].

The objective is to learn a policy $\pi_\theta : \Omega \times \mathcal{G} \rightarrow \text{Pr}(\mathcal{A})$ that maximizes the expected sum of discounted rewards $\mathbb{E}_\pi \left[\sum_{t=0}^{H-1} \gamma^t r_t \right]$ where $\gamma \in (0, 1)$ is a discount factor which trades off between current and future rewards. Our approach employs PPO to train a stochastic policy. PPO is a widely used on-policy RL algorithm, known for its effectiveness across diverse control tasks, such as locomotion [74] and in-hand manipulation [75]. PPO aims to update the policy by maximizing a surrogate objective function that balances between improving the policy and maintaining stability. The key idea is to ensure that the new policy does not deviate too much from the old policy, preventing large, destabilizing updates [76].

3.2. Simulation Setup

The simulation environment is built using AGX Dynamics[®], a high-fidelity physics engine designed for simulating complex mechanical systems. AGX Dynamics[®] provides the accuracy and stability for simulating the intricate interactions between the excavator and granular terrain, enabling a reliable platform for developing autonomous excavation strategies [69, 21]. The machine, based on the excavator CAT[®]365, is a large excavator weighing 65,960 kg, with a maximum digging depth of 9.64 m and a bucket capacity of up to 3.8 m³. The excavator model in our simulation includes three linear actuators, which are actuated as motor-driven joints.

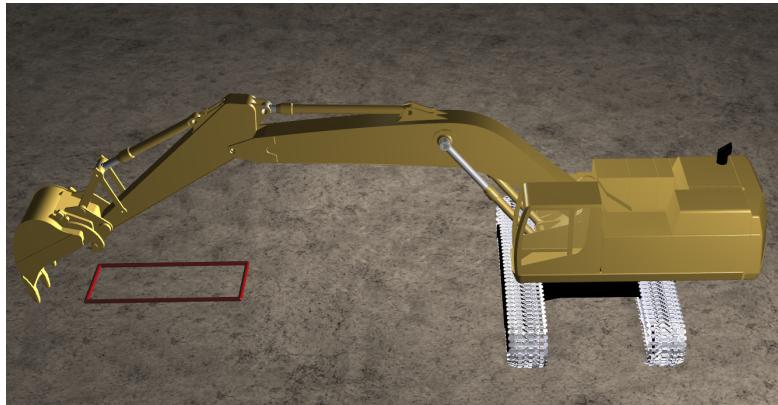
The terrain is represented using a deformable 3D grid-based model, where properties such as mass, compaction, and soil type are discretely stored. Soil is deformed and moved via shovel objects, which interact with the terrain to simulate realistic digging and earthmoving behavior. These shovels convert static soil into dynamic mass aggregates, generating feedback forces based on soil mechanics theory, and allow for excavation, compaction, and soil redistribution [19].

Moreover, a rock is introduced by creating its mesh, scaling the vertices to the desired size, and defining its collision geometry. The rock is assigned a material and then is added to the simulation along with the appropriate contact to ensure realistic interactions with the bucket, the terrain, and its particles.

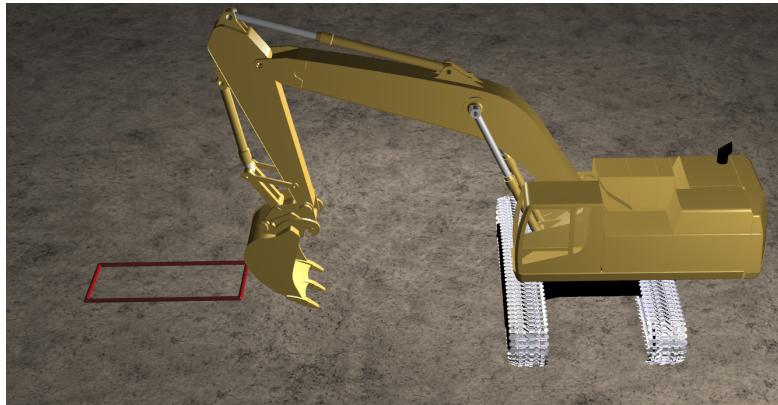
3.3. Task Description

The primary objective of the RL agent is to capture the rock using the bucket and move it to a designated goal location as quickly as possible. The task begins with the rock initialized at a location within the manipulator’s effective workspace. Although the excavator’s bucket can physically reach areas beyond the effective workspace, both closer to and farther from the machine, these regions are not considered part of the effective workspace because the bucket cannot perform the rock capturing task effectively there. In real-world scenarios, placing the rock too close or too far from the excavator is unrealistic. For successful operation, the rock must be positioned within the machine’s effective workspace. The effective workspace of the boom, arm, and bucket is illustrated by the red boundary in Fig. 3.

The excavator is controlled by generating speed commands to the bucket, arm, and boom joints. Although the excavator model includes components such as the cabin hinge joint and tracks, these are not actuated in this task, as three joints of manipulator are sufficient for the rock capturing. In this task, maintaining stability is critical, as improper interaction between the bucket and the terrain can cause the bucket to get stuck, potentially leading to tilting of the excavator. Another critical aspect of the task is the interaction between the bucket and the rock. If the bucket approaches the rock from an incorrect Point of Attack (PoA) or with excessive speed, the rock may be thrown, which is undesirable and potentially dangerous. At the end of the task, the rock should be securely inside the bucket and positioned close enough to the goal position.



(a) Maximum reach at ground level.



(b) Minimum reach at ground level.

Figure 3: Illustration of the maximum and minimum reach capabilities of the manipulator at ground level. The red boundary indicates the excavator's effective workspace where the rock capturing task should be performed.

3.4. Episode Initialization

At the beginning of each training episode, the environment is initialized in a randomized state. The randomized parameters are listed in Table 1.

Table 1: Randomized parameters at the beginning of each training episode.

Parameter	Distribution
Goal position	$\mathcal{N}(\boldsymbol{\mu} = \begin{bmatrix} -7.0 \\ 1.5 \end{bmatrix}, \boldsymbol{\Sigma} = \begin{bmatrix} 0.1^2 & 0 \\ 0 & 0.1^2 \end{bmatrix})$
Rock density	$\mathcal{N}(\mu = 2000.0, \sigma^2 = 85.0^2)$
Rock geometry	$\mathcal{U}\{I, II\}$
Rock position (x -axis)	$\mathcal{U}(-11.5, -8.0)$

The goal position is sampled from a 2D normal distribution (also known as a bivariate normal distribution) $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where the mean $\boldsymbol{\mu}$ and standard deviation $\boldsymbol{\Sigma}$ values are presented in Table 1. To ensure feasibility, the sampled position is constrained within a circle of radius 0.3 m. If a sample falls outside this boundary, it is projected onto the circle's edge. For better understanding, a visualization of randomly sampled goal positions is shown in Fig. 4.

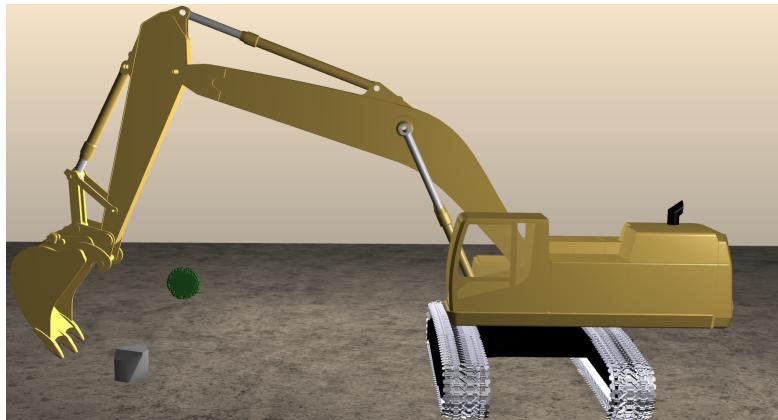


Figure 4: Illustration of randomly sampled goal positions during training. The samples are drawn from the bivariate normal distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and constrained within a circular region of radius 0.3 m.

A rock is added to the environment with a randomly selected geometry from two different mesh models, as illustrated in Fig. 5. The randomization of geometry and density results in significant randomization of the rock

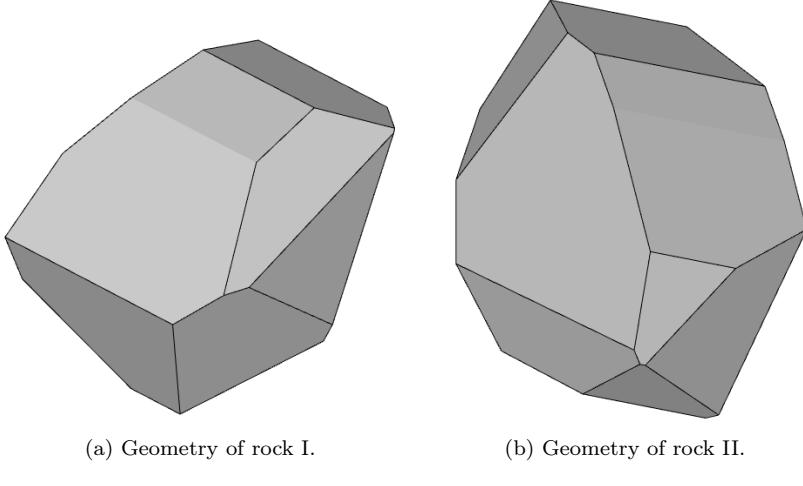


Figure 5: The geometries of the rocks used during training. Each episode randomly selects one of the two rock meshes.

mass. Further implementation details and parameter specifications for the environment initialization are provided in Appendix A.

3.5. Observation

At each time step t , the policy receives an observation o_t that includes the position, speed, and force of the three prismatic joints; the x - and z -coordinates of the goal position, the position of the rock’s Center of Mass (CoM), and the center position of the bucket. All quantities are expressed in the base frame. The base frame is located beneath the cabin at the excavator’s swing pin. Its x -axis points toward the rear of the machine (i.e., opposite the direction the excavator faces), its z -axis points upward, and the y -axis is defined according to the right-hand rule. The motion of the excavator’s manipulator is confined to the $x-z$ plane of the base frame. The observation also contains the roll ϕ and pitch θ angles of the base frame. To maintain the stability and avoid tilting of the excavator during the operation, the roll ϕ and pitch θ angles of the base frame play a key role. However, these observations do not capture other aspects of the environment state, such as the frictional contact forces, the contact points, or the mass and geometric properties of the rock. These unobserved variables can affect the dynamics of interaction and task success. The observations are summarized in Table 2. In addition, to accelerate convergence, the observations are normalized to the range $[-1, 1]$ using their minimum and maximum values.

Observations o_t	Notation	Unit
Joint positions	$(q_{\text{boom}}, q_{\text{arm}}, q_{\text{bucket}})$	m
Joint speeds	$(v_{\text{boom}}, v_{\text{arm}}, v_{\text{bucket}})$	m/s
Joint forces	$(f_{\text{boom}}, f_{\text{arm}}, f_{\text{bucket}})$	kN
Bucket position	$(x_{\text{bucket}}, z_{\text{bucket}})$	m
Rock position	$(x_{\text{rock}}, z_{\text{rock}})$	m
Goal position	$(x_{\text{goal}}, z_{\text{goal}})$	m
Cabin pitch and roll angles	(θ, ϕ)	rad

Table 2: Observations used by the policy at each time step. The observation vector o_t has a total dimension of 17. Units and notation for each observation are shown in the table.

3.6. Goal and Action

The policy is trained to control the excavator to capture a rock and move it toward a designated goal position in the base frame. The rock is considered close enough to the goal if the horizontal and vertical distances between the rock’s CoM and the goal position are each less than proximity threshold δ_{prox} , meaning the proximity condition defined in Eq. (1) is satisfied:

$$\mathbb{C}_{\text{proximity}} = \begin{cases} 1, & \text{if } |x_{\text{rock}} - x_{\text{goal}}| < \delta_{\text{prox}} \text{ and} \\ & |z_{\text{rock}} - z_{\text{goal}}| < \delta_{\text{prox}} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

In addition, another condition regarding the tilting of the excavator is required to obtain the goal condition. The roll ϕ and pitch θ angles of the cabin should be less than tilt threshold δ_{tilt} , meaning the tilting condition defined in Eq. (2) is satisfied:

$$\mathbb{C}_{\text{tilting}} = \begin{cases} 1, & \text{if } |\phi| < \delta_{\text{tilt}} \text{ and } |\theta| < \delta_{\text{tilt}} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

The combination of proximity condition $\mathbb{C}_{\text{proximity}}$ and tilting condition $\mathbb{C}_{\text{tilting}}$ is expressed as the goal condition \mathbb{C}_{goal} :

$$\mathbb{C}_{\text{goal}} = \begin{cases} 1, & \text{if } \mathbb{C}_{\text{proximity}} \text{ and } \mathbb{C}_{\text{tilting}} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Given a goal g_t and an observation o_t , the policy takes an action $a_t = [v_{\text{boom}}, v_{\text{arm}}, v_{\text{bucket}}]^T$, which consists of the joint speed of the boom, arm, and bucket. The trained policy generates normalized speed commands in the range of $[-1, 1]$, which are then scaled by the maximum speed limits of each joint. The maximum speed commands for the boom, arm, and bucket joints are 0.3 m/s , 0.3 m/s , and 0.2 m/s , respectively.

3.7. Reward

Designing an effective reward function is crucial for enabling fast and stable learning, especially in environments with sparse rewards, where the agent must explore extensively before discovering rewarding behaviors. To address this challenge, a guiding reward formulation is adopted, inspired by prior works [77, 78]. Unlike sparse rewards, which provide feedback only upon satisfying the goal conditions, the guiding reward offers intermediate feedback by encouraging the agent to move toward the target and to maintain the desired state once reached. In our case, the agent receives a reward r_t at each time step, composed of two components: a guidance reward r_{guidance} , which incentivizes progress toward the goal state, and a goal reward r_{goal} , which encourages the agent to remain within the goal region once it is reached:

$$r_t = r_{\text{guidance}} + r_{\text{goal}} \quad (4)$$

This reward shaping approach facilitates both exploration and convergence. However, it is important to note that reward functions designed through trial-and-error can lead to reward overfitting, where the learned behavior becomes overly tailored to a specific algorithm or training scenario [77]. To mitigate this risk, the guiding reward are deliberately kept abstract and general, motivating the agent to reach and maintain the goal state, without relying on detailed task-specific heuristics. A fixed episode length is long enough to ensure sufficient time for goal achievement while promoting efficient learning.

The guidance reward r_{guidance} is formulated as follows:

$$\begin{aligned}
r_{\text{guidance}} = & -\frac{1}{w_1} (x_{\text{rock}} - x_{\text{goal}})^2 \\
& -\frac{1}{w_2} (z_{\text{rock}} - z_{\text{goal}})^2 \\
& -\frac{1}{w_3} \|a_t \odot f_t\|_2^2 \\
& -\frac{1}{w_4} \|a_t - a_{t-1}\|_2^2 \\
& -\frac{1}{w_5} (\theta^2 + \phi^2)
\end{aligned} \tag{5}$$

where $f_t = [f_{\text{boom}}, f_{\text{arm}}, f_{\text{bucket}}]^T$ denotes the joint forces. The first and second terms penalize the distance between the rock's CoM and the goal in the x - and z -axes. The third term penalizes energy usage by minimizing the product of joint forces and speeds, the fourth term penalizes unnecessary or excessive variations in control input (joint speeds) to promote smooth movements, and the last term penalizes cabin tilt to ensure the excavator remains stable. The notation $\|\cdot\|_2$ denotes the Euclidean norm (2-norm), and \odot indicates the element-wise product. The weights w_i , $i \in \{1, 2, 3, 4, 5\}$ serve to normalize the associated penalty terms and control the trade-offs among the different components. The goal reward r_{goal} is defined as:

$$r_{\text{goal}} = \begin{cases} 5, & \text{if } \mathbb{C}_{\text{goal}} \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Moreover, an episode truncates if the rock becomes inaccessible to the bucket. The truncate condition is defined as:

$$\mathbb{C}_{\text{truncate}} = \begin{cases} 1, & \text{if } |y_{\text{rock}}| > y_{\text{truncate}} \text{ or } x_{\text{rock}} < x_{\text{truncate}} \\ 0, & \text{otherwise} \end{cases} \tag{7}$$

where y_{rock} is the position of the rock's CoM in y -axis in the base frame. The $\mathbb{C}_{\text{truncate}}$ indicates whether the rock lies outside the excavator's effective workspace: $x_{\text{rock}} < x_{\text{truncate}}$ means it is too far, and $|y_{\text{rock}}| > y_{\text{truncate}}$ means it is outside the $x - z$ plane of manipulator motion. The parameters used in the reward function and conditions are listed in Table 3. Finally, an episode terminates if the episode length reaches to the maximum episode length H .

Table 3: Parameters used in the guidance reward r_{guidance} and task conditions.

Parameter	Value	Unit
w_1	13.0	—
w_2	8.0	—
w_3	3.0×200.0^2	—
w_4	12.0	—
w_5	1.0	—
δ_{prox}	0.2	m
δ_{tilt}	0.1	rad
x_{truncate}	-13.0	m
y_{truncate}	1.0	m

3.8. Training

The agent is trained using the PPO algorithm [76] implemented in the Stable-Baselines3 library [79]. Separate neural networks are used to represent the policy and value functions, each receiving identical inputs and having a linear output layer. The training hyperparameters are provided in Table 4. All training and experience collection are performed on a Linux workstation

Table 4: PPO training hyperparameters.

Parameter	Value
Policy hidden layers	128×128 , tanh
Value hidden layers	128×128 , tanh
Discount factor γ	0.99
Max. episode length	$500(@60Hz) \approx 8.3\ s$
Entropy coefficient	3×10^{-4}
Learning rate	3×10^{-4}
Value function coefficient	0.5
Max. grad norm	0.5
GAE λ	0.95
Mini-batches	128
Optimization epochs	4
Clip range	0.2

running Ubuntu 22.04 LTS. The system is equipped with an Intel Xeon E5-

1650 v2 CPU (6 physical cores, 12 threads @ 3.5 GHz), an NVIDIA GeForce RTX 4070 GPU with 12 GB of VRAM, and 64 GB of system RAM. The policy is trained for 15×10^6 time steps, which lasts around 17 h.

4. Results

In this section, the results of the proposed method are presented. First, the cumulative reward obtained during policy training is illustrated. The training setup is extensively described in Sections 3.2 and 3.3, where the rock’s geometry (shown in Fig. 5) and density are randomized, its initial position is sampled within the excavator’s effective workspace, and the goal position is drawn from a normal distribution. The soil is modeled as cohesive dirt, whose physical parameters are listed in Table A.7. Then, the trained policy is evaluated across four distinct scenarios: (I) under conditions similar to the training environment, (II) with rocks different from those used during training, (III) with soil properties that differ from the training conditions, and (IV) in comparison with human participants using a game-like control interface.

The cumulative reward curve, shown in Fig. 6, represents the mean episodic cumulative reward, averaged over the most recent 100 episodes. To improve interpretability, the cumulative reward is smoothed using Eq. (8):

$$s(t) = \begin{cases} x(0), & \text{if } t = 0 \\ w_s s(t-1) + (1 - w_s)x(t), & \text{if } t > 0 \end{cases} \quad (8)$$

where $x(t)$ is the original data at time step t , $s(t)$ is the smoothed value at time step t , and $w_s \in [0, 1]$ is the smoothing weight. Larger values of w_s result in greater smoothing, making the output less sensitive to recent changes in the data. Lower values of w_s provide less smoothing and give greater weight to recent observations. As shown in Fig. 6, the cumulative reward increases until approximately 8 million time steps, after which it plateaus, indicating that the policy has converged. The success rate is presented in Fig. 7. This value represents the fraction of episodes during a rollout (a window of recent 100 episodes) in which the agent successfully reaches the goal. An episode is considered successful if the goal condition \mathbb{C}_{goal} defined in Eq. (3) is satisfied. At the end of the training, the success rate is around 0.8 which shows satisfactory performance for this complex task.

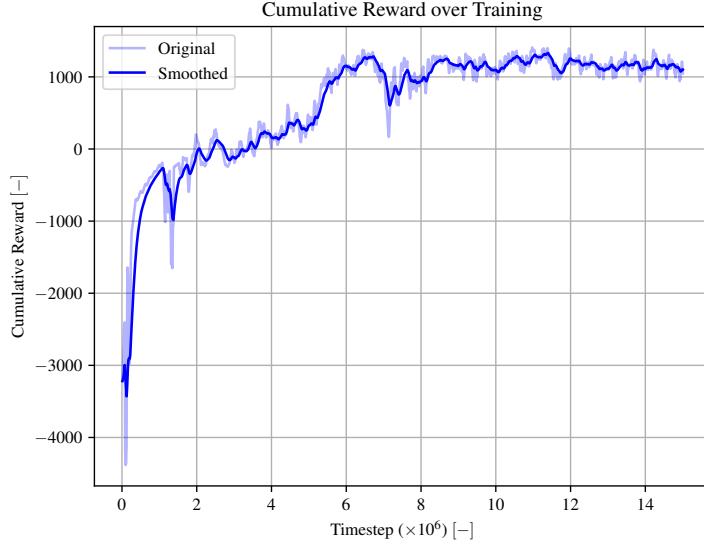


Figure 6: Cumulative reward obtained during training. To improve interpretability, a smoothing function (Eq. (8)) with $w_s = 0.9$ is applied to the cumulative reward. The original values are shown with reduced opacity for visual reference.

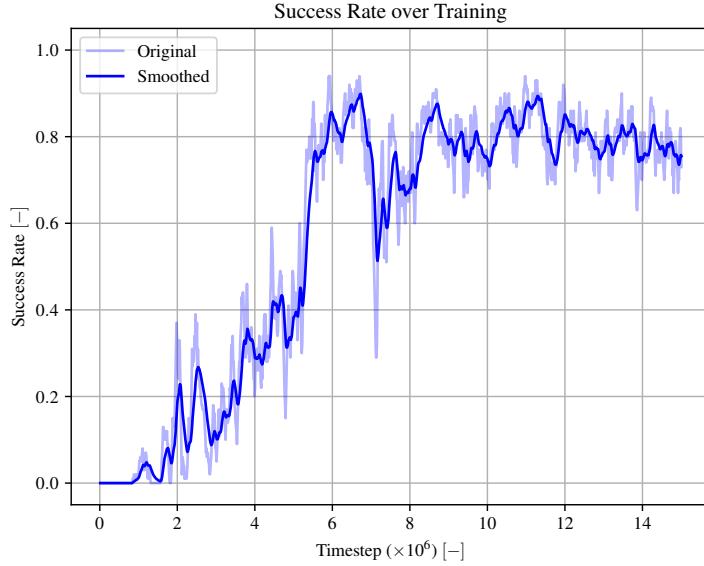


Figure 7: Success rate obtained during training. To improve interpretability, a smoothing function (Eq. (8)) with $w_s = 0.9$ is applied to the success rate. The original values are shown with reduced opacity for visual reference.

4.1. Evaluation Under Training Conditions

In this section, the learned policy is evaluated over 10 episodes under conditions similar to those used during training. The success rate, along with the mean and standard deviation of the cumulative reward of the agent across different evaluation scenarios, is summarized in Table 5.

Table 5: Success rates and mean and standard deviation of cumulative reward over 10 episodes for each evaluation scenario.

Evaluation Scenario	Success Rate	Cumulative Reward
Training condition	0.9	1428.47 ± 611.13
Unseen rocks	0.8	1195.84 ± 745.17
Unseen materials	0.7	952.04 ± 779.10

To facilitate deeper analysis, one successful episode is randomly selected for detailed examination. The trajectories of the rock and the bucket are illustrated in Fig. 8, showing that the rock successfully reaches the goal and remains within its proximity. The control inputs, defined as the joint speeds

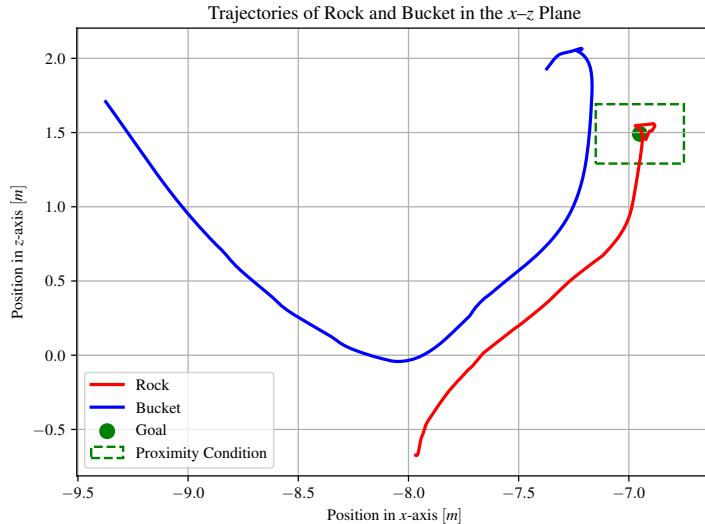


Figure 8: Trajectories of the rock and the bucket in the x - z plane during a successful episode under training conditions. The green square denotes the proximity condition.

of the boom, arm, and bucket, are shown in Fig. 9. In the simulation, the

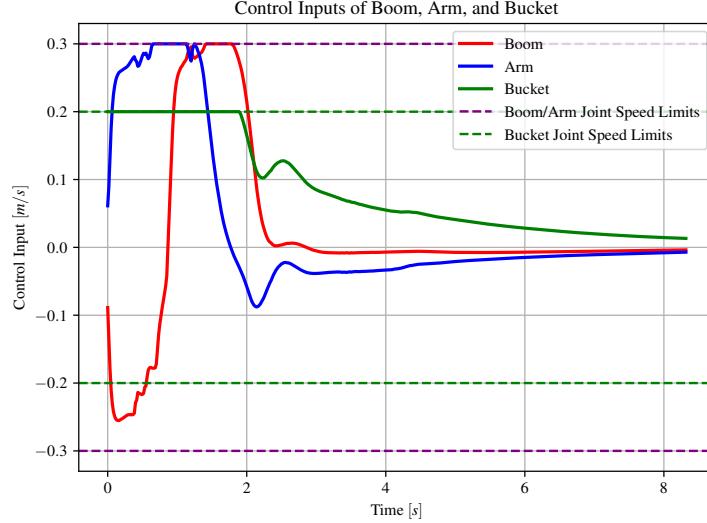


Figure 9: Joint speed commands for the bucket, arm, and boom throughout a successful episode under training conditions.

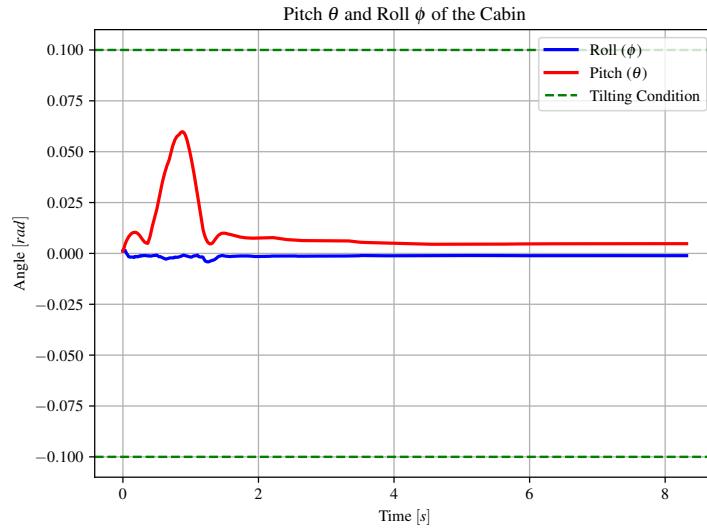


Figure 10: Pitch θ and roll ϕ angles of the cabin during a successful episode under training conditions. The valid range defined by the tilting condition C_{tilting} is indicated by green dashed lines.

excavator immediately executes the commanded joint speeds, so the control inputs directly match the actual joint speeds. It is worth noting that, unlike the simulation, a real excavator exhibits delays between commanded and measured joint speeds, which introduces a sim-to-real gap that must be considered for deployment. Toward the end of the episode, the joint speeds gradually converge to zero, which helps stabilize the bucket and the rock near the goal location. A smooth deceleration is crucial, since abruptly driving the joints to zero could cause the rock to be thrown out of the bucket due to inertial effects, potentially falling to the ground or even being projected toward the cabin, which would be unsafe in real-world operation. As observed, the signals exhibit no jerky or excessive actions, suggesting that the generated control inputs are feasible for real-world operations. Finally, the cabin’s pitch θ and roll ϕ angles are shown in Fig. 10. Both angles remain within the valid range defined by the tilting condition C_{tilting} .

4.2. Evaluation Using Unseen Rock Geometries

In this section, the learned policy is evaluated over 10 episodes using two unseen rock geometries to assess the policy’s generalization capabilities. The shapes of these rocks are shown in Fig. 11. The success rate, along with the

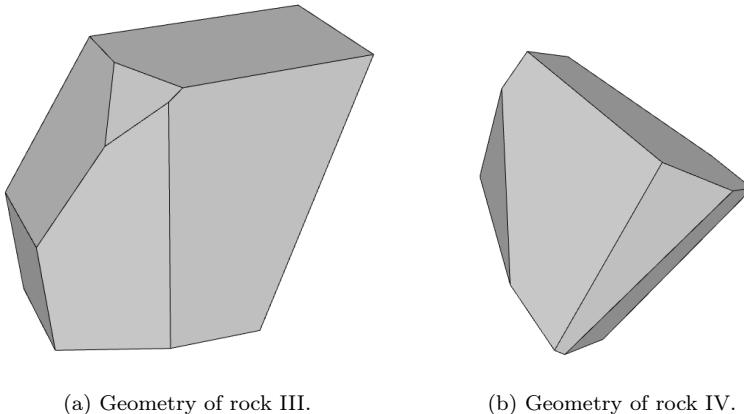


Figure 11: The geometries of the rocks used in the unseen rock evaluation scenario. Each episode randomly selects one of the two rock meshes.

mean and standard deviation of the cumulative reward for this scenario, is summarized in Table 5. A significant reduction in both metrics compared to the training conditions is not observed, demonstrating that the policy does not overfit to specific rock geometries, densities, or masses.

To provide further insight, one successful episode is randomly selected for detailed analysis. The trajectories of the rock and bucket in the x - z plane are illustrated in Fig. 12. The rock clearly reaches and remains within the goal region (highlighted by the green square). Notably, the movement appears near-optimal, with an almost direct trajectory and minimal unnecessary motion. Figure 13 displays the control inputs (joint speed commands) for the

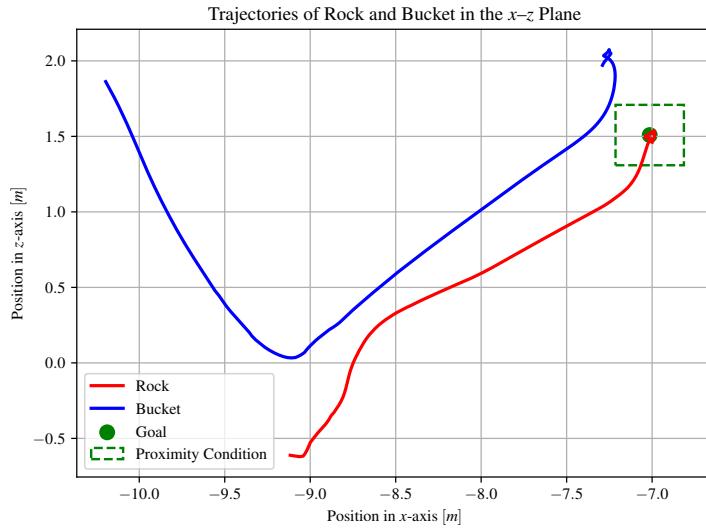


Figure 12: Trajectories of the rock and bucket in the x - z plane during a successful episode in unseen rock evaluation scenario. The green square denotes the proximity condition.

bucket, arm, and boom. After approximately 2 s, the inputs reduce to stabilize the rock near the goal position. The absence of abrupt or excessive commands supports the real-world feasibility of the learned policy. Finally, the cabin’s pitch θ and roll ϕ angles are plotted in Fig. 14. Only minor variations are observed, all of which remain within the allowable tilting range defined by $\mathbb{C}_{\text{tilting}}$.

4.3. Evaluation Using Unseen Material Properties

In this section, the controller is evaluated over 10 episodes using sand as material to assess the policy’s generalization to varying material properties. The key physical parameters defining the sand material are provided in Appendix B. The success rate, as well as the mean and standard deviation of the cumulative reward for this scenario, are reported in Table 5. The controller’s performance in this setting is comparable to that observed under

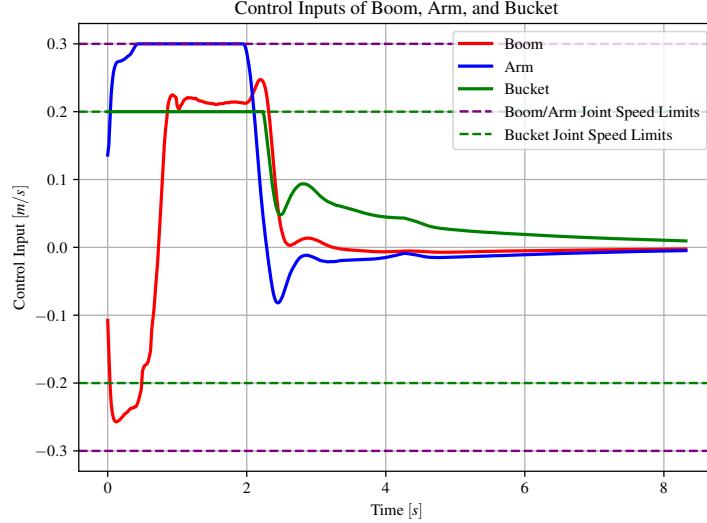


Figure 13: Joint speed commands for the bucket, arm, and boom throughout a successful episode in unseen rock evaluation scenario.

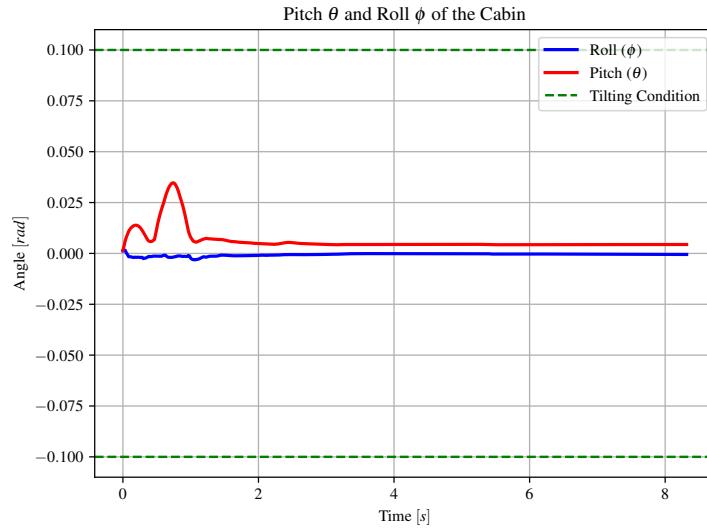


Figure 14: Pitch θ and roll ϕ angles of the cabin during a successful episode in unseen rock evaluation scenario. The valid range defined by the tilting condition C_{tilting} is indicated by green dashed lines.

the training conditions and with unseen rock geometries, indicating that the learned policy is not sensitive to changes in material properties.

To gain deeper insight into the agent's behavior, one successful episode is randomly selected for detailed analysis. The trajectories of the rock and the bucket in the x - z plane are illustrated in Fig. 15. Initially, the rock reaches the proximity of the goal, but it later drifts slightly outside the goal square due to motion along the x -axis. The bucket trajectory indicates that the agent adjusts its movement strategy to return the rock to the vicinity of the goal position. Figure 16 presents the joint speed commands for the bucket,

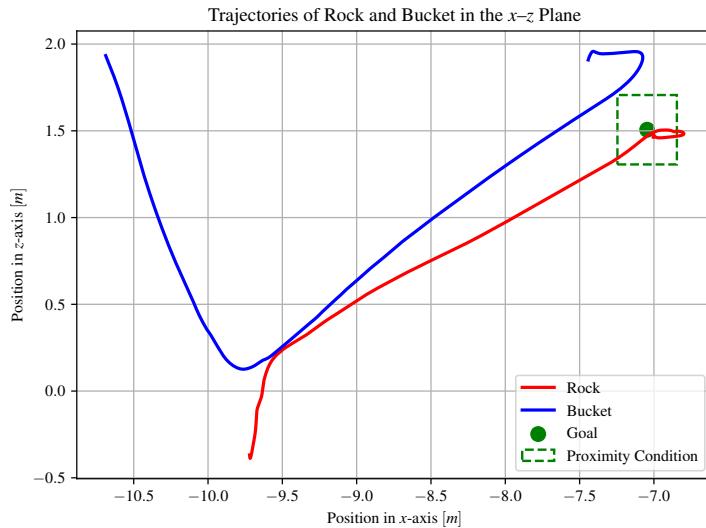


Figure 15: Trajectories of the rock and bucket in the x - z plane during a successful episode in unseen material evaluation scenario. The green square denotes the proximity condition.

arm, and boom during a successful episode in the unseen material evaluation scenario. Unlike the previous scenarios, around $t = 3$ s, the bucket joint moves in the negative direction before returning to a positive value. Such a pattern in the bucket speed command was not observed in two previous scenarios, suggesting that the agent adjusts its strategy to reposition the rock and maintain it within the goal proximity. The cabin's pitch θ and roll ϕ angles during a successful episode in the unseen material evaluation scenario are shown in Fig. 17.

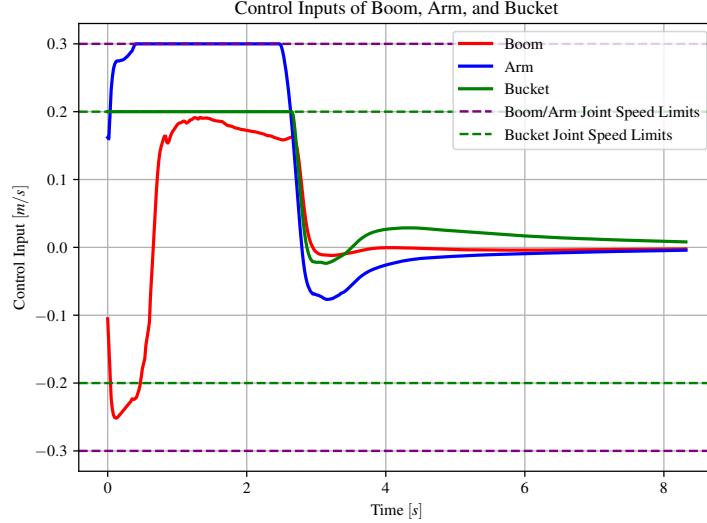


Figure 16: Joint speed commands for the bucket, arm, and boom throughout a successful episode in unseen material evaluation scenario.

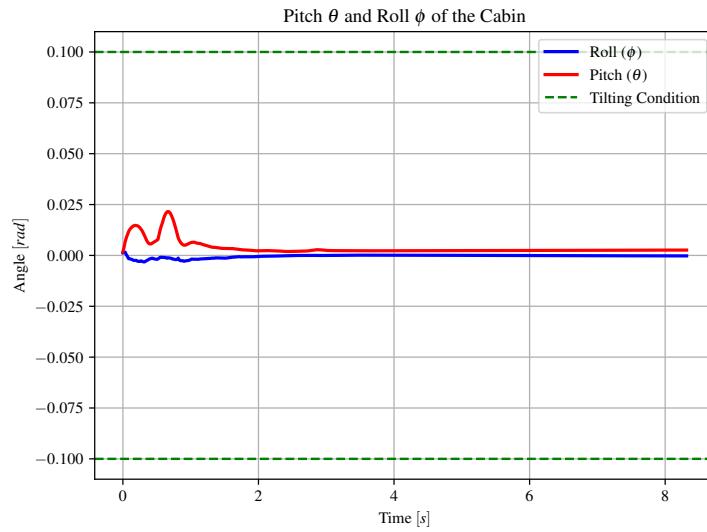


Figure 17: Pitch θ and roll ϕ angles of the cabin during a successful episode in unseen material evaluation scenario. The valid range defined by the tilting condition C_{tilting} is indicated by green dashed lines.

4.4. Comparison with Human Participants

In this evaluation, two human participants were invited to perform the rock capturing task under conditions identical to the training environment. The setup resembled a game-like interface, aligning with recent trends where simulators are commonly used to both train and assess the performance of human operators. Each participant was given the opportunity to practice in the AGX Dynamics® simulator for 100 trials. Following the practice phase, they completed 10 evaluation trials each, resulting in a total of 20 human-operated episodes. The first participant achieved a success rate of 0.8, while the second participant reached 0.4, yielding an overall success rate of 0.6 for this scenario. These results highlight the difficulty of the task, even for human operators. It is important to note that comparing the RL agent with human operators is not straightforward. While the agent is explicitly trained to maximize the designed reward function, human operators pursue their own implicit objectives during task execution, which may not fully align with the agent’s reward structure. Despite this difference, humans are still relatively successful at completing the task, as reflected in their success rate.

To better understand the behavior of human participants, one successful episode is randomly selected for detailed analysis. The trajectories of the rock and the bucket in x - z plane are shown in Fig. 18. It is important to note that unlike the learned agent, the human can see the goal location and its proximity area but does not have access to the exact position of the rock’s CoM. As a result, aligning the CoM precisely with the goal position is a challenging task. The human participant visually compares the entire rock and its geometry against the goal proximity area instead of focusing on a single reference point like the rock’s CoM. Moreover, the operator interacts with the task through a 2D view of the simulator, which limits depth perception and spatial awareness. This restricted perspective further increases the difficulty of achieving precise alignment and can negatively affect overall performance compared to the agent. In this evaluation scenario, similar to a game-like simulator, human operators used a keyboard to send commands to control the speed of the bucket, arm, and boom joints. To enable comparison with other evaluation scenarios, the resulting joint speed derived from the keyboard inputs are visualized in Fig. 19. It is evident that the human control strategy differs significantly from that of the learned policy, exhibiting a bang–bang-like behavior in the joint space. This behavior can be attributed to two main factors: first, the limited sensitivity and resolution of the keyboard input device, and second, the lack of haptic feedback in the simulator,

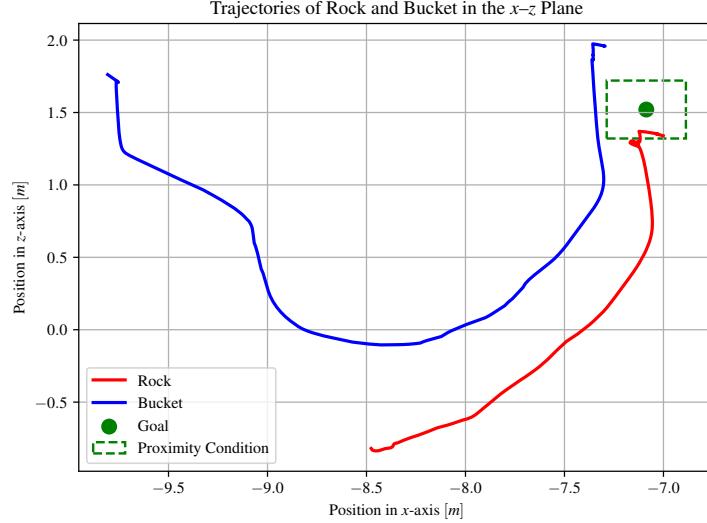


Figure 18: Trajectories of the rock and bucket in the x - z plane during a successful episode in human participant evaluation scenario. The green square denotes the proximity condition.

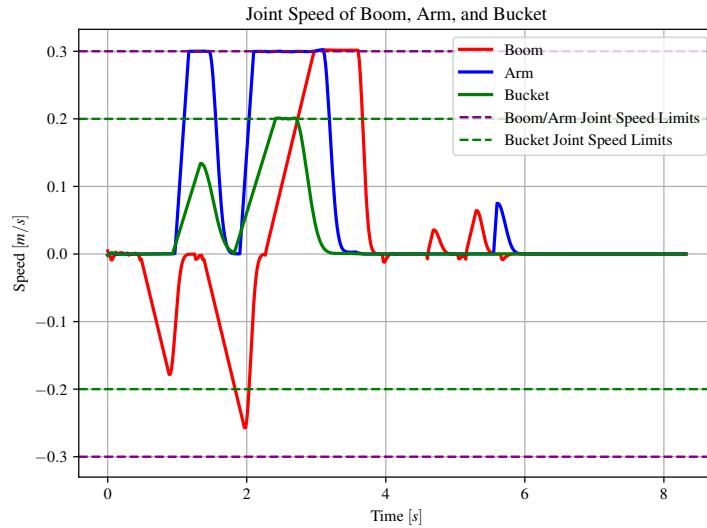


Figure 19: Joint speed the bucket, arm, and boom throughout a successful episode in human participant evaluation scenario.

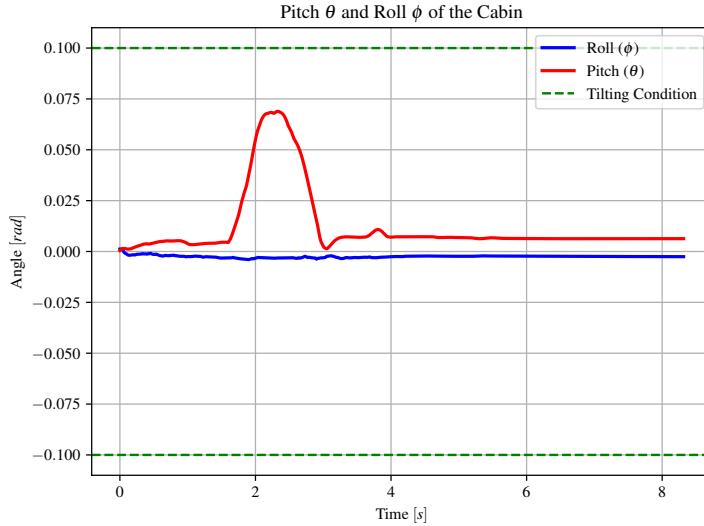


Figure 20: Pitch θ and roll ϕ angles of the cabin during a successful episode in human participant evaluation scenario. The valid range defined by the tilting condition C_{tilting} is indicated by green dashed lines.

meaning that the operator does not feel the machine’s motion, vibrations, or contact forces. These factors make it challenging for human participants to apply smooth and continuous control actions, in contrast to the fine-grained control achievable by the RL policy. Finally, the cabin’s pitch θ and roll ϕ angles are shown in Fig. 20. Compared to the learned policy across different evaluation scenarios, greater variation is observed in the pitch angle θ , indicating less stable control by the human operator. However, the values remain within the acceptable tilting range.

5. Discussion

This paper presented a RL-based controller for automatic rock capturing using an excavator. As demonstrated in the result section, the trained policy is capable of effectively capturing and moving the rock to the proximity of a desired goal location, even when the rock geometry or material properties differ from those used during training. As shown in Fig. 7, the controller achieves a success rate of 0.8, comparable to that of human participants.

Despite these promising results, several limitations and areas for future improvement remain. The policy has only been evaluated in simulation. De-

ployment on a real machine is required to assess the sim-to-real transferability. While zero-shot transfer may be possible, fine-tuning on real-world data will likely be required. First, applying domain randomization to parameters related to the homogeneous material, rock properties, and contact interactions can reduce the sim-to-real gap. Additionally, reducing the particle size of the material could improve simulation realism and may inadvertently make the task easier, as smaller particles enable smoother interactions with the rock. However, this comes at the cost of increased computational complexity, which can slow down the training process. Second, actuation delays can be explicitly modeled in the simulator to better reflect the latency present in real machine. Third, injecting noise into observations during training can help develop a more robust policy capable of handling real-world sensor inaccuracies. Forth, the simulation currently models the excavator’s joints as linear actuator with closed chain mechanisms, while real machines operate using hydraulic cylinders. Incorporating a more realistic hydraulic model would contribute to reducing the sim-to-real gap. Fifth, in the current setup, the rock’s CoM position is directly available from the simulator. However, in real-world applications, this information must be inferred using a vision-based system, which introduces uncertainty and noise. These discussions outline potential directions that will be further investigated in future work to enhance realism and facilitate real-world deployment.

From a computational perspective, the RL-based controller offers key advantages at deployment. Unlike online optimization-based methods, which often require solving computationally intensive problems at each step, the RL policy uses a simple neural network forward pass to generate control commands. This makes inference extremely fast and suitable for real-time applications.

From a learning perspective, the current reward function does not penalize episode failure or timeouts. Introducing a negative reward in such cases could accelerate the learning process and encourage more efficient behaviors. Beyond success rate, energy consumption could also serve as a valuable secondary metric for evaluating the controller’s overall efficiency. The reward function also does not penalize the amount of homogeneous soil scooped along with the rock. Including such a penalty could encourage more precise and efficient manipulation strategies; however, it may also make the learning process more challenging and slower. Training efficiency could benefit from curriculum learning. Starting with relaxed goal conditions and gradually tightening them as training progresses would likely speed up convergence

and improve performance. Currently, the agent relies solely on the most recent observation. Incorporating a short history of past observations, either through observation stacking or by using recurrent neural networks such as Long Short-Term Memory (LSTM), could improve temporal reasoning and enhance performance.

Safety is a critical consideration in the rock capturing task. Unsafe behaviors, such as aggressive or rapid bucket movements, can result in the rock being unintentionally thrown, posing serious risks to the machine and nearby personnel. Integrating safe RL techniques could enhance the stability of the system and reduce the likelihood of hazardous actions. Finally, exploring alternative RL algorithms, such as Soft Actor-Critic (SAC), could potentially improve sample efficiency and performance. Additionally, integrating model-based RL approaches may further enhance learning speed and policy effectiveness by leveraging predictive models of the environment.

6. Conclusion

This paper presented a fully data-driven, RL-based control strategy for automating the rock capturing task using a standard excavator. Capturing and moving large rocks with a bucket is a highly skill-intensive task due to the unstructured environment and complex, dynamic interactions between the rock, granular material, and manipulator. Traditional control approaches struggle with these challenges because accurate analytical models are difficult to obtain and require extensive tuning. To address this, a model-free RL policy was trained entirely in the high-fidelity AGX Dynamics® simulator using the PPO algorithm. The training employed domain randomization of rock geometry, density, mass, and initial configurations of the rock, bucket, and goal, ensuring robustness and generalization. The learned controller directly outputs joint speed commands for the boom, arm, and bucket, without relying on explicit knowledge of rock or material properties. Evaluation across multiple scenarios, including unseen rocks and varying material properties, demonstrates that the policy achieves a high success rate of 0.8, comparable to human participants, while maintaining stable machine behavior and avoiding unsafe bucket motions. These results indicate that complex rock manipulation tasks can be effectively automated using standard excavator buckets through learning-based control, without requiring specialized end-effectors. Future work will focus on bridging the sim-to-real gap by incorporating hydraulic actuator modeling, actuation delays, vision-based rock

tracking, and sensor noise injection. Safety-critical aspects, such as preventing aggressive bucket actions that could lead to hazardous outcomes, also warrant further exploration, potentially via safe RL techniques. To the best of our knowledge, this study is the first to propose and evaluate an RL-based controller for rock capturing with an excavator, highlighting the potential of learning-based methods for automating challenging earth-moving tasks in construction and mining operations.

Acknowledgement

The work was funded in part by Horizon Europe Project XSCAVE under Grant 101189836, and in part by the Research Council of Finland through the PROFI 7 grant.

Declaration of Generative AI and AI-Assisted Technologies in the Writing Process

During the preparation of this work the authors used ChatGPT in order to improve language clarity. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. Episode Initialization Details

The rock is assigned a material, and its density is sampled from a normal distribution with a mean of 2000 kg/m^3 and a standard deviation of 85 kg/m^3 . The rock is initialized along the x -axis of the base frame, with its y -coordinate set to 0 and its z -coordinate elevated to 0.5 m above the ground to prevent initial collision with the terrain. The x -coordinate is sampled uniformly from the range $[-11.5, -8.0]$.

The initial configuration of the excavator’s bucket, arm, and boom is set such that the bucket starts behind and above the rock, positioned appropriately based on the rock’s position. Details of the initial manipulator

Table A.6: Initial configurations of the excavator’s boom, arm, and bucket based on the initial position of the rock in x -axis.

Initial position of the rock in x -axis [m]	Joint positions [m]		
	Boom	Arm	Bucket
$-8.0 \leq x_{\text{rock}}$	+0.13	+0.24	-0.88
$-8.5 \leq x_{\text{rock}} < -8.0$	+0.08	+0.11	-0.80
$-9.0 \leq x_{\text{rock}} < -8.5$	+0.06	-0.03	-0.74
$-9.5 \leq x_{\text{rock}} < -9.0$	+0.03	-0.15	-0.74
$-10.0 \leq x_{\text{rock}} < -9.5$	-0.01	-0.33	-0.70
$-10.5 \leq x_{\text{rock}} < -10.0$	-0.03	-0.39	-0.70
$-11.0 \leq x_{\text{rock}} < -10.5$	-0.10	-0.57	-0.70
$-11.5 \leq x_{\text{rock}} < -11.0$	-0.10	-0.70	-0.70
$x_{\text{rock}} < -11.5$	-0.16	-0.80	-0.78

configuration are provided in Table A.6. Note that moving the bucket to this initial position is not part of the RL agent’s task; the agent begins from this pre-positioned state. A simple schematic of initial configurations of the rock and bucket is illustrated in Fig. A.21.

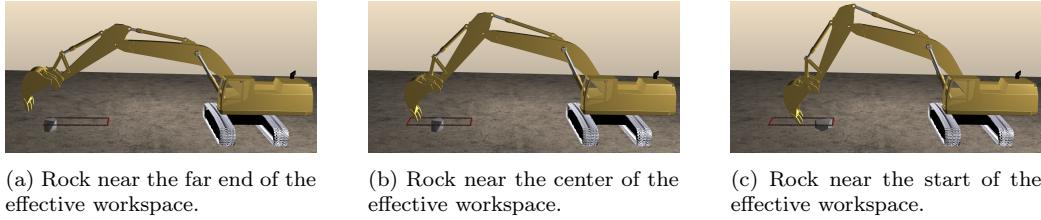


Figure A.21: Schematic illustrations of three example initial configurations of the rock and the excavator’s bucket. The bucket is pre-positioned above and behind the rock before each episode begins. These configurations are determined based on the rock’s initial position along the x -axis, as detailed in Table A.6.

It should be noted the terrain, which is made of dirt material with zero slope, is not subject to randomization. The dirt material represents a cohesive, moderately stiff soil commonly found in natural excavation environments. The key parameters used to define the dirt material are summarized in Table A.7.

Table A.7: Key parameters of dirt material used in simulation.

Parameter	Value	Unit
Cohesion (bulk)	2100.0	Pa
Density	1474.0	kg/m^3
Maximum Density	2000.0	kg/m^3
Young's Modulus	10^6	Pa
Internal Friction Angle	0.70	rad
Dilatancy Angle	0.23	rad
Swell Factor	1.10	—
Angle of Repose Compaction Rate	24.0	—

Appendix B. Sand Material Properties

The sand material used for the third evaluation scenario was defined to assess the generalization capability of the trained policy under different terrain conditions. The key physical parameters of the sand are summarized in Table B.8.

Table B.8: Key parameters of sand material used in third evaluation scenario.

Parameter	Value	Unit
Cohesion (bulk)	0.0	Pa
Density	1474.0	kg/m^3
Maximum Density	1800.0	kg/m^3
Young's Modulus	4.5×10^6	Pa
Internal Friction Angle	0.68	rad
Dilatancy Angle	0.16	rad
Swell Factor	1.0	—
Angle of Repose Compaction Rate	1.0	—

References

- [1] F. E. Sotiropoulos, H. H. Asada, Autonomous excavation of rocks using a gaussian process model and unscented kalman filter, *IEEE Robotics and Automation Letters* 5 (2) (2020) 2491–2497. doi:10.1109/LRA.2020.2972891.

- [2] P. Egli, L. Terenzi, M. Hutter, Reinforcement learning-based bucket filling for autonomous excavation, *IEEE Transactions on Field Robotics* 1 (2024) 170–191. [doi:10.1109/TFR.2024.3432508](https://doi.org/10.1109/TFR.2024.3432508).
- [3] C. Ishmatuka, I. Soesanti, A. Ataka, Autonomous pick-and-place using excavator based on deep reinforcement learning, in: 2023 15th International Conference on Information Technology and Electrical Engineering (ICITEE), 2023, pp. 19–24. [doi:10.1109/ICITEE59582.2023.10317662](https://doi.org/10.1109/ICITEE59582.2023.10317662).
- [4] A. Molaei, A. Kolu, K. Lahtinen, M. Geimer, Automatic estimation of excavator actual and relative cycle times in loading operations, *Automation in Construction* 156 (2023) 105080. [doi:10.1016/j.autcon.2023.105080](https://doi.org/10.1016/j.autcon.2023.105080).
- [5] L. Werner, F. Nan, P. Eyschen, F. A. Spinelli, H. Yang, M. Hutter, Dynamic throwing with robotic material handling machines, in: 2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2024, pp. 98–104. [doi:10.1109/IROS58592.2024.10802743](https://doi.org/10.1109/IROS58592.2024.10802743).
- [6] P. Egli, M. Hutter, A general approach for the automation of hydraulic excavator arms using reinforcement learning, *IEEE Robotics and Automation Letters* 7 (2) (2022) 5679–5686. [doi:10.1109/LRA.2022.3152865](https://doi.org/10.1109/LRA.2022.3152865).
- [7] L. Zhang, J. Zhao, P. Long, L. Wang, L. Qian, F. Lu, X. Song, D. Manocha, An autonomous excavator system for material loading tasks, *Science Robotics* 6 (55) (2021) eabc3164. [doi:10.1126/scirobotics.abc3164](https://doi.org/10.1126/scirobotics.abc3164).
- [8] X. Huang, L. E. Bernold, Toward an adaptive control model for robotic backhoe excavation, *Transportation Research Record* (1406) (1993) 20–24.
- [9] S. Blouin, A. Hemami, M. Lipsett, Review of resistive force models for earthmoving processes, *Journal of Aerospace Engineering* 14 (3) (2001) 102–111. [doi:10.1061/\(ASCE\)0893-1321\(2001\)14:3\(102\)](https://doi.org/10.1061/(ASCE)0893-1321(2001)14:3(102)).
- [10] X. D. Huang, L. E. Bernold, Robotic rock handling during backhoe excavation, in: *Proceedings of the 10th International Symposium on*

- Automation and Robotics in Construction (ISARC), International Association for Automation and Robotics in Construction (IAARC), 1993, pp. 355–362. doi:10.22260/ISARC1993/0046.
- [11] C. McKinnon, J. A. Marshall, Automatic identification of large fragments in a pile of broken rock using a time-of-flight camera, *IEEE Transactions on Automation Science and Engineering* 11 (3) (2014) 935–942. doi:10.1109/TASE.2014.2308011.
 - [12] J. Del Aguila Ferrandis, J. Moura, S. Vijayakumar, Nonprehensile planar manipulation through reinforcement learning with multimodal categorical exploration, in: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2023, pp. 5606–5613. doi:10.1109/IROS55552.2023.10341629.
 - [13] Y. Zhu, L. Wang, L. Zhang, Excavation of fragmented rocks with multimodal model-based reinforcement learning, in: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022, pp. 6523–6530. doi:10.1109/IROS47612.2022.9981537.
 - [14] D. Wang, C. Liu, F. Chang, H. Huan, K. Cheng, Multi-stage reinforcement learning for non-prehensile manipulation, *IEEE Robotics and Automation Letters* 9 (7) (2024) 6712–6719. doi:10.1109/LRA.2024.3412630.
 - [15] I. Kurinov, G. Orzechowski, P. Hämäläinen, A. Mikkola, Automated excavator based on reinforcement learning and multibody system dynamics, *IEEE Access* 8 (2020) 213998–214006. doi:10.1109/ACCESS.2020.3040246.
 - [16] E. Coumans, Bullet physics simulation, in: ACM SIGGRAPH 2015 Courses, SIGGRAPH ’15, Association for Computing Machinery, New York, NY, USA, 2015. doi:10.1145/2776880.2792704.
URL <https://doi.org/10.1145/2776880.2792704>
 - [17] E. Todorov, T. Erez, Y. Tassa, Mujoco: A physics engine for model-based control, in: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2012, pp. 5026–5033. doi:10.1109/IROS.2012.6386109.

- [18] J. Collins, S. Chand, A. Vanderkop, D. Howard, A review of physics simulators for robotic applications, *IEEE Access* 9 (2021) 51416–51431. [doi:10.1109/ACCESS.2021.3068769](https://doi.org/10.1109/ACCESS.2021.3068769).
- [19] M. Servin, T. Berglund, S. Nystedt, A multiscale model of terrain dynamics for real-time earthmoving simulation, *Advanced Modeling and Simulation in Engineering Sciences* 8 (1) (2021) 11. [doi:10.1186/s40323-021-00196-3](https://doi.org/10.1186/s40323-021-00196-3).
- [20] C. Aluckal, R. V. K. Lal, S. Courtney, Y. Turkar, Y. Dighe, Y.-J. Kim, J. Gemerek, K. Dantu, TERA: A simulation environment for terrain excavation robot autonomy (dec 2024). [doi:10.48550/arXiv.2501.01430](https://doi.org/10.48550/arXiv.2501.01430).
- [21] K. Aoshima, M. Servin, Examining the simulation-to-reality gap of a wheel loader digging in deformable terrain, *Multibody System Dynamics* (2024). [doi:10.1007/s11044-024-10005-5](https://doi.org/10.1007/s11044-024-10005-5)
URL <https://doi.org/10.1007/s11044-024-10005-5>
- [22] K. Johnson, The elusive dream of fully autonomous construction vehicles, Technical report (2023).
- [23] A. J. Koivo, M. Thoma, E. Kocaoglan, J. Andrade-Cetto, Modeling and control of excavator dynamics during digging operation, *Journal of Aerospace Engineering* 9 (1) (1996) 10–18. [doi:10.1061/\(ASCE\)0893-1321\(1996\)9:1\(10\)](https://doi.org/10.1061/(ASCE)0893-1321(1996)9:1(10)).
- [24] H. Yoshida, T. Yoshimoto, D. Umino, N. Mori, Practical full automation of excavation and loading for hydraulic excavators in indoor environments, in: 2021 IEEE 17th International Conference on Automation Science and Engineering (CASE), 2021, pp. 2153–2160. [doi:10.1109/CASE49439.2021.9551504](https://doi.org/10.1109/CASE49439.2021.9551504).
- [25] H. Shao, H. Yamamoto, Y. Sakaida, T. Yamaguchi, Y. Yanagisawa, A. Nozue, Automatic excavation planning of hydraulic excavator, in: C. Xiong, H. Liu, Y. Huang, Y. Xiong (Eds.), *Intelligent Robotics and Applications*, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 1201–1211.
- [26] R. J. Sandzimier, H. H. Asada, A data-driven approach to prediction and optimal bucket-filling control for autonomous excavators, *IEEE Robotics*

and Automation Letters 5 (2) (2020) 2682–2689. doi:10.1109/LRA.2020.2969944.

- [27] F. E. Sotiropoulos, H. H. Asada, A model-free extremum-seeking approach to autonomous excavator control based on output power maximization, IEEE Robotics and Automation Letters 4 (2) (2019) 1005–1012. doi:10.1109/LRA.2019.2893690.
- [28] F. E. Sotiropoulos, H. H. Asada, Dynamic modeling of bucket-soil interactions using koopman-dfl lifting linearization for model predictive contouring control of autonomous excavators, IEEE Robotics and Automation Letters 7 (1) (2022) 151–158. doi:10.1109/LRA.2021.3121136.
- [29] M. Hutter, P. Leemann, S. Stevsic, A. Michel, D. Jud, M. Hoepflinger, R. Siegwart, R. Figi, C. Caduff, M. Loher, S. Tagmann, Towards optimal force distribution for walking excavators, in: 2015 International Conference on Advanced Robotics (ICAR), 2015, pp. 295–301. doi:10.1109/ICAR.2015.7251471.
- [30] D. Jud, G. Hottiger, P. Leemann, M. Hutter, Planning and control for autonomous excavation, IEEE Robotics and Automation Letters 2 (4) (2017) 2151–2158. doi:10.1109/LRA.2017.2721551.
- [31] S. K. D. Jud, P. Leemann, M. Hutter, Autonomous free-form trenching using a walking excavator, IEEE Robotics and Automation Letters 4 (4) (2019) 3208–3215. doi:10.1109/LRA.2019.2925758.
- [32] N. Reginald, J. Seo, M. Cha, Integrative tracking control strategy for robotic excavation, International Journal of Control, Automation and Systems 19 (2021) 3435–3450. doi:10.1007/s12555-020-0595-2.
- [33] G. J. Maeda, I. R. Manchester, D. C. Rye, Combined ilc and disturbance observer for the rejection of near-repetitive disturbances, with application to excavation, IEEE Transactions on Control Systems Technology 23 (5) (2015) 1754–1769. doi:10.1109/TCST.2014.2382579.
- [34] J. Park, B. Lee, S. Kang, P. Y. Kim, H. J. Kim, Online learning control of hydraulic excavators based on echo-state networks, IEEE Transactions on Automation Science and Engineering 14 (1) (2017) 249–259. doi:10.1109/TASE.2016.2582213.

- [35] H.-S. Park, D.-V. Dang, T.-T. Nguyen, N.-T. Le, Implementation of a virtual autonomous excavator, *Transactions of FAMENA* 41 (3) (2017) 65–80. doi:[10.21278/TOF.41306](https://doi.org/10.21278/TOF.41306).
- [36] M. Dunbabin, P. Corke, Autonomous excavation using a rope shovel, *Journal of Field Robotics* 23 (6-7) (2006) 379–394. doi:<https://doi.org/10.1002/rob.20132>.
- [37] D. A. Bradley, D. W. Seward, The development, control and operation of an autonomous robotic excavator, *Journal of Intelligent and Robotic Systems* 21 (1998) 73–97. doi:[10.1023/A:1007932011161](https://doi.org/10.1023/A:1007932011161).
- [38] T. Groll, S. Hemer, T. Ropertz, K. Berns, Autonomous trenching with hierarchically organized primitives, *Automation in construction* 98 (2019) 214–224. doi:[10.1016/j.autcon.2018.11.016](https://doi.org/10.1016/j.autcon.2018.11.016).
- [39] Q. Ha, M. Santos, Q. Nguyen, D. Rye, H. Durrant-Whyte, Robotic excavation in construction automation, *IEEE Robotics & Automation Magazine* 9 (1) (2002) 20–28. doi:[10.1109/100.993151](https://doi.org/10.1109/100.993151).
- [40] D. Schmidt, M. Proetzsch, K. Berns, Simulation and control of an autonomous bucket excavator for landscaping tasks, in: 2010 IEEE International Conference on Robotics and Automation, 2010, pp. 5108–5113. doi:[10.1109/ROBOT.2010.5509546](https://doi.org/10.1109/ROBOT.2010.5509546).
- [41] X. Shi, P. Lever, F.-Y. Wang, Experimental robotic excavation with fuzzy logic and neural networks, in: Proceedings of IEEE International Conference on Robotics and Automation, Vol. 1, 1996, pp. 957–962 vol.1. doi:[10.1109/ROBOT.1996.503896](https://doi.org/10.1109/ROBOT.1996.503896).
- [42] Y. Yang, P. Long, X. Song, J. Pan, L. Zhang, Optimization-based framework for excavation trajectory generation, *IEEE Robotics and Automation Letters* 6 (2) (2021) 1479–1486. doi:[10.1109/LRA.2021.3058071](https://doi.org/10.1109/LRA.2021.3058071).
- [43] Y. Zhang, Z. Sun, Q. Sun, Y. Wang, X. Li, J. Yang, Time-jerk optimal trajectory planning of hydraulic robotic excavator, *Advances in Mechanical Engineering* 13 (7) (2021) 16878140211034611. doi:[10.1177/16878140211034611](https://doi.org/10.1177/16878140211034611).
- [44] D. Lee, I. Jang, J. Byun, H. Seo, H. J. Kim, Real-time motion planning of a hydraulic excavator using trajectory optimization and model

- predictive control, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021, pp. 2135–2142. doi: [10.1109/IROS51168.2021.9635965](https://doi.org/10.1109/IROS51168.2021.9635965).
- [45] Y. B. Kim, J. Ha, H. Kang, P. Y. Kim, J. Park, F. Park, Dynamically optimal trajectories for earthmoving excavators, *Automation in Construction* 35 (2013) 568–578. doi:[10.1016/j.autcon.2013.01.007](https://doi.org/10.1016/j.autcon.2013.01.007).
 - [46] Y. Yang, J. Pan, P. Long, X. Song, L. Zhang, Time variable minimum torque trajectory optimization for autonomous excavator, CoRR abs/2006.00811 (2020).
 - [47] T. Yoshida, T. Koizumi, N. Tsujiuchi, Z. Jiang, et al., Digging trajectory optimization by soil models and dynamics models of excavator, *SAE International Journal of Commercial Vehicles* 6 (2) (2013) 429–440. doi: [10.4271/2013-01-2411](https://doi.org/10.4271/2013-01-2411).
 - [48] Z. Zou, J. Chen, X. Pang, Task space-based dynamic trajectory planning for digging process of a hydraulic excavator with the integration of soil–bucket interaction, *Proceedings of the Institution of Mechanical Engineers, Part K* 233 (3) (2019) 598–616. doi:[10.1177/1464419318812589](https://doi.org/10.1177/1464419318812589).
 - [49] K. Althoefer, C. P. Tan, Y. H. Zweiri, L. D. Seneviratne, Hybrid soil parameter measurement and estimation scheme for excavation automation, *IEEE Transactions on Instrumentation and Measurement* 58 (10) (2009) 3633–3641. doi:[10.1109/TIM.2009.2018699](https://doi.org/10.1109/TIM.2009.2018699).
 - [50] Y. Zhao, J. Wang, Y. Zhang, C. Luo, A novel method of soil parameter identification and force prediction for automatic excavation, *IEEE Access* 8 (2020) 11197–11207. doi:[10.1109/ACCESS.2020.2965214](https://doi.org/10.1109/ACCESS.2020.2965214).
 - [51] J. Zhao, Y. Hu, C. Liu, M. Tian, X. Xia, Spline-based optimal trajectory generation for autonomous excavator, *Machines* 10 (7) (2022). doi: [10.3390/machines10070538](https://doi.org/10.3390/machines10070538).
 - [52] Q. Guo, Z. Ye, L. Wang, L. Zhang, Imitation learning and model integrated excavator trajectory planning, in: 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2022, pp. 5737–5743. doi:[10.1109/IROS47612.2022.9981220](https://doi.org/10.1109/IROS47612.2022.9981220).

- [53] J. Huh, J. Bae, D. Lee, J. Kwak, C. Moon, C. Im, Y. Ko, T. K. Kang, D. Hong, Deep learning-based autonomous excavation: A bucket-trajectory planning algorithm, *IEEE Access* 11 (2023) 38047–38060. doi:10.1109/ACCESS.2023.3267120.
- [54] B. Son, C. Kim, C. Kim, D. Lee, Expert-emulating excavation trajectory planning for autonomous robotic industrial excavator, in: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2020, pp. 2656–2662. doi:10.1109/IROS45743.2020.9341036.
- [55] P. Egli, D. Gaschen, S. Kerscher, D. Jud, M. Hutter, Soil-adaptive excavation using reinforcement learning, *IEEE Robotics and Automation Letters* 7 (4) (2022) 9778–9785. doi:10.1109/LRA.2022.3189834.
- [56] C. Schenck, J. Tompson, S. Levine, D. Fox, Learning robotic manipulation of granular media, in: Proceedings of the 1st Annual Conference on Robot Learning, Vol. 78 of Proceedings of Machine Learning Research, PMLR, 2017, pp. 239–248.
- [57] Q. Lu, L. Zhang, Excavation learning for rigid objects in clutter, *IEEE Robotics and Automation Letters* 6 (4) (2021) 7373–7380. doi:10.1109/LRA.2021.3097264.
- [58] R. Fukui, T. Niho, M. Nakao, M. U. and, Imitation-based control of automated ore excavator: improvement of autonomous excavation database quality using clustering and association analysis processes, *Advanced Robotics* 31 (11) (2017) 595–606. doi:10.1080/01691864.2017.1297735.
- [59] H. Tahara, H. Sasaki, H. Oh, B. Michael, T. Matsubara, Disturbance-injected robust imitation learning with task achievement, in: 2022 International Conference on Robotics and Automation (ICRA), 2022, pp. 2466–2472. doi:10.1109/ICRA46639.2022.9812376.
- [60] S. Jin, Z. Ye, L. Zhang, Learning excavation of rigid objects with offline reinforcement learning (2023). arXiv:2303.16427.
URL <https://arxiv.org/abs/2303.16427>
- [61] Q. Lu, Y. Zhu, L. Zhang, Excavation reinforcement learning using geometric representation, *IEEE Robotics and Automation Letters* 7 (2) (2022) 4472–4479. doi:10.1109/LRA.2022.3150511.

- [62] P. Samtani, F. Leiva, J. Ruiz-del Solar, Learning to break rocks with deep reinforcement learning, *IEEE Robotics and Automation Letters* 8 (2) (2023) 1077–1084. doi:10.1109/LRA.2023.3236562.
- [63] Q. Zhu, Q.-F. Wang, Real-time energy management controller design for a hybrid excavator using reinforcement learning, *Journal of Zhejiang University - Science A* 18 (11) (2017) 855–870. doi:10.1631/jzus.A1600650.
URL <https://doi.org/10.1631/jzus.A1600650>
- [64] B. J. Hodel, Learning to operate an excavator via policy optimization, *Procedia Computer Science* 140 (2018) 376–382, cyber Physical Systems and Deep Learning Chicago, Illinois November 5-7, 2018. doi:10.1016/j.procs.2018.10.301.
- [65] A. R. Reece, Paper 2: The fundamental equation of earth-moving mechanics, *Proceedings of the Institution of Mechanical Engineers, Conference Proceedings* 179 (6) (1964) 16–22. doi:10.1243/PIME_CONF_1964_179_134_02.
- [66] T. Erez, Y. Tassa, E. Todorov, Simulation tools for model-based robotics: Comparison of bullet, havok, mujoco, ode and physx, in: 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 4397–4404. doi:10.1109/ICRA.2015.7139807.
- [67] K. Matsumoto, A. Yamaguchi, T. Oka, M. Yasumoto, S. Hara, M. Iida, M. Teichmann, Simulation-based reinforcement learning approach towards construction machine automation (2020).
- [68] T. Osa, M. Aizawa, Deep reinforcement learning with adversarial training for automated excavation using depth images, *IEEE Access* 10 (2022) 4523–4535. doi:10.1109/ACCESS.2022.3140781.
- [69] M. Servin, T. Berglund, S. Nystedt, A multiscale model of terrain dynamics for real-time earthmoving simulation, *Advanced Modeling and Simulation in Engineering Sciences* 8 (1) (2021) 11. doi:10.1186/s40323-021-00196-3.
- [70] K. Aoshima, High-performance autonomous wheel loading : a computational approach, Ph.D. thesis, Umeå University, Department of Physics (2025).

- [71] M. T. Mason, Progress in nonprehensile manipulation, *The International Journal of Robotics Research* 18 (11) (1999) 1129–1141. doi:10.1177/02783649922067762.
- [72] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, M. Hutter, Learning robust perceptive locomotion for quadrupedal robots in the wild, *Science Robotics* 7 (62) (2022) eabk2822. doi:10.1126/scirobotics.abk2822.
- [73] L. Cong, H. Liang, P. Ruppel, Y. Shi, M. Görner, N. Hendrich, J. Zhang, Reinforcement learning with vision-proprioception model for robot planar pushing, *Frontiers in Neurorobotics Volume 16 - 2022* (2022). doi:10.3389/fnbot.2022.829437.
URL <https://www.frontiersin.org/journals/neurorobotics/articles/10.3389/fnbot.2022.829437>
- [74] N. Heess, D. TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. M. A. Eslami, M. A. Riedmiller, D. Silver, Emergence of locomotion behaviours in rich environments, CoRR abs/1707.02286 (2017). arXiv:1707.02286.
URL <http://arxiv.org/abs/1707.02286>
- [75] O. M. Andrychowicz, B. Baker, M. Chociej, R. Józefowicz, B. McGrew, J. Pachocki, A. Petron, M. Plappert, G. Powell, A. Ray, J. Schneider, S. Sidor, J. Tobin, P. Welinder, L. Weng, W. Zaremba, Learning dexterous in-hand manipulation, *The International Journal of Robotics Research* 39 (1) (2020) 3–20. doi:10.1177/0278364919887447.
- [76] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, O. Klimov, Proximal policy optimization algorithms, CoRR abs/1707.06347 (2017). arXiv:1707.06347.
URL <http://arxiv.org/abs/1707.06347>
- [77] G. Vasan, Y. Wang, F. Shahriar, J. Bergstra, M. Jagersand, A. R. Mahmood, Revisiting sparse rewards for goal-reaching reinforcement learning (2024).
URL <https://arxiv.org/abs/2407.00324>
- [78] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, W. Zaremba, Openai gym, CoRR abs/1606.01540 (2016).
URL <http://arxiv.org/abs/1606.01540>

- [79] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, N. Dormann, Stable-baselines3: Reliable reinforcement learning implementations, *Journal of Machine Learning Research* 22 (268) (2021) 1–8.
URL <http://jmlr.org/papers/v22/20-1364.html>