IEEE-RAS 24th International Conference on Humanoid Robots (Humanoids), Seoul, Korea, 2025.

Interactive Shaping of Granular Media Using Reinforcement Learning

Benedikt Kreis[1,3,4] Malte Mosbach[2,4] Anny Ripke[1] M. Ehsan Ullah[1] Sven Behnke[2,3,4] Maren Bennewitz[1,3,4]

***Abstract*— Autonomous manipulation of granular media, such as sand, is crucial for applications in construction, excavation, and additive manufacturing. However, shaping granular materials presents unique challenges due to their high-dimensional configuration space and complex dynamics, where traditional rule-based approaches struggle without extensive engineering efforts. Reinforcement learning (RL) offers a promising alternative by enabling agents to learn adaptive manipulation strategies through trial and error. In this work, we present an RL framework that enables a robotic arm with a cubic end-effector and a stereo camera to shape granular media into desired target structures. We show the importance of compact observations and concise reward formulations for the large configuration space, validating our design choices with an ablation study. Our results demonstrate the effectiveness of the proposed approach for the training of visual policies that manipulate granular media including their real-world deployment, outperforming two baseline approaches.**

I. Introduction

The ability to manipulate granular media such as sand has many applications in robotics, ranging from construction and excavation [1]–[8] to additive manufacturing [9]. Unlike the manipulation of rigid bodies, the shaping of granular media is accompanied by unique challenges due to their particle nature. Accurate modeling and control of such media, requires accounting for complex interactions that may vary depending on the material composition. To successfully shape such media, an agent must continuously adapt its manipulation strategy in response to material deformation. Applying traditional modeling approaches requires extensive engineering efforts due to the large configuration space of deformable objects and media [10].

Reinforcement learning (RL) provides an alternative forgoing the need to predict the precise consequences of manipulations to the granular media, allowing the agent instead to adaptively react to the way the manipulation unfolds by learning optimal strategies through trial and error. This allows the robot to interact with the state of the medium in a closed loop (see Fig. 1). Although RL has been successfully applied to the dexterous manipulation of rigid objects [11], recent research suggests its potential for manipulating deformable objects, including cloth handling [10] and fluid control [12]. However, the application of RL to the interactive manipulation of granular media has not been explored sufficiently, likely due to two key challenges. First, *finding a compact observation space* for granular media is difficult. Rigid objects can often be represented efficiently using their poses, but granular media exhibits an effectively infinite configuration space, requiring a high-dimensional representation. Second, *designing an effective reward function* is challenging. While object manipulation tasks often leverage distance-based rewards to guide learning, shaping rewards this way for granular media results in very sparse rewards since most random manipulation actions lead to configurations further away from the desired goal, which can result in an agent that avoids interactions with the granular media altogether.

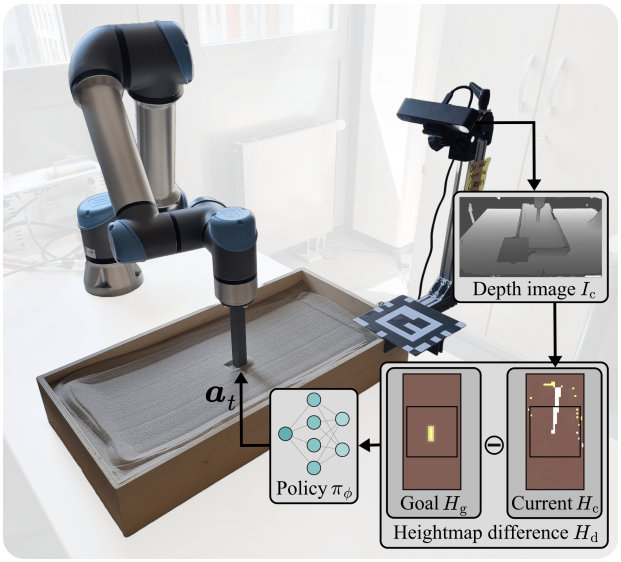


Fig. 1: The robot's task is to manipulate the granular media with its cubic end-effector to shape it as close as possible to desired goal configurations. The configurations are abstracted as height maps and the robot reconstructs the height map corresponding to the current configuration from depth observations. Our approach closes the loop between partial visual observations and goal-oriented manipulation, accounting for the dynamics of collapsing granular media during the manipulation.

In this work, we address these challenges by studying how the Markov decision process (MDP) of granular media manipulation can be defined to make RL algorithms sucessfully applicable. To this end, we make the following contributions:

- We develop a novel reward formulation that fosters fast and stable convergence of RL training towards functional granular media manipulation behaviors.
- We demonstrate that RL policies can be learned from visual observations by converting high-dimensional depth images to compact height map representations.
- We demonstrate that the resulting formulation allows for zero-shot transfer of trained policies to a real robot.

[1]Humanoid Robots Lab, University of Bonn, Germany.
[2]Autonomous Intelligent Systems Lab, University of Bonn, Germany.
[3]Center for Robotics, University of Bonn, Germany.
[4]Lamarr Institute for ML and AI, Bonn, Germany.
This work has been partially funded by the EC, grant No. 964854 RePAIR H2020-FETOPEN-2018-2020 and by the BMBF within the Robotics Institute Germany, grant No. 16ME0999.
The corresponding author is Benedikt Kreis: kreis@cs.uni-bonn.de

II. Related Work

Manipulating granular media like sand is an active research field in robotics tackling challenges in simulating contacts for locomotion [13]–[15], retrieving objects [16], grading [17], sweeping [18], [19], trenching [20], and excavating [1]–[8], [21]. The scale of the proposed pipelines varies from sweeping a limited amount of grit stones [18] to shaping entire landscapes [22].

A. Simulating Granular Media

Modeling contact interactions between robots and granular media is challenging due to the amount of individual, simulated particles [23]. Exact methods like finite element methods (FEMs) or discrete element methods (DEMs) are suitable to simulate material deformations and physically accurate behavior of granular media. While FEM is less computationally expensive than DEM, it does not capture the granularity of particle interactions, as it treats materials as continua. Although both simulate a larger variety of phenomena than rigid body simulations, their required extensive computations can hinder their application in robotics [13]. Therefore, Xu *et al.* [16] use relatively big and coarse particles to reduce their total amount. This minimizes the computational cost, but it makes the model less accurate.

Another way to save computational cost is to model particles as a large graph and selectively activate small subgraphs to predict how local robot-terrain interactions deform the granular medium, as proposed by Liu *et al.* [21].

Kim *et al.* [24] as well as Pavlov and Johnson [20] go one step further by focusing on deformations on the surface of granular media, rather than computing the behavior of each particle. They proposed a sand model using a height map that mimics the collapse of sand piles based on the angle of repose. If the angle of repose between two height map cells is surpassed, the excess sand is distributed to adjacent cells until the angle condition is met again. As their model is both computationally efficient and physically realistic, we base our granular media simulation on the same model.

B. Robotic Manipulation of Granular Media

A common construction task is the grading of sand, e.g., to build roads. Miron *et al.* [17] leverage imitation learning to control a bulldozer to level sand piles modeled as two-dimensional multivariate Gaussian distributions.

Instead of leveling piles, Alatur *et al.* [18] proposed to form them by teaching a robot to sweep grit stones and wooden chips on a table. By extending a classical motion planner with optimal transport, their robot is able to arrange predetermined pile shapes.

Another well-researched construction task is excavating granular media. Schenck *et al.* [1] learned predictive models with highly-tailored convolutional neural network (CNN) architectures based on an experimental data set. This allows to predict the dynamics of scooping and dumping actions. Jin *et al.* [2] tackled the excavation task with the offline RL algorithm Implicit Q-Learning (IQL) and trained on a prerecorded data set with six different terrain types. IQL allows them to outperform sub-optimal demonstrations in the data set, but it requires new data to adapt the policy to new terrain types.

Several authors have used an autonomous walking excavator called HEAP [25]. Using this heavy machinery, they proposed multiple approaches ranging from classical path planning focusing on the overall landscaping system [4], [22], [26] to an RL-based approach [5]. In the latter, they use Proximal Policy Optimization (PPO) with general advantage estimation (GAE) to train a controller to adaptively dig in different soil types. Further RL-based excavation approaches for heavy machinery have been proposed by Kurinov *et al.* [7] using Proximal Policy Optimization with Covariance Matrix Adaptation (PPO-CMA), as well as by Osa and Aizawa [6] using Qt-Opt, a sample efficient variant of Q-learning trained from depth images.

For extraterrestrial missions, being able to reuse existing equipment is a crucial advantage due to the saved weight. Therefore, Pavlov and Johnson [20] proposed the idea to dig trenches with rover wheels instead of a dedicated end-effector. Kim *et al.* [24] built upon Pavlov's experimental results and they proposed three methods to dig trenches of constant depths. Among them, a classical A^*-based planner performing single strokes and a learned approach based on Deep Deterministic Policy Gradient (DDPG) combined with Hindsight Experience Replay (HER) performing multiple strokes.

Apart from weight restrictions, extraterrestrial systems can suffer domain-shift issues due to partially known environmental conditions when training on earth and then deploying to outer space. In particular, vision-based systems struggle with unforeseen conditions. To overcome this problem, Zhu *et al.* [3] proposed an adaptive, vision-based scooping strategy leveraging meta-learning on a deep Gaussian process and show that despite the domain shift, depth images allow to learn the scooping task.

Yet another challenge is to move the robot itself within granular media, especially on steep, inclined surfaces. That is why, Karsai *et al.* [14] and Kerimoglu *et al.* [15] explored gaits to manipulate local granular terrain to improve the climbing and turning performance of a wheeled-legged robot using Bayesian optimization. In addition to direct terrain manipulation, Hu *et al.* [27] proposed to indirectly manipulate objects by creating small avalanches within granular media. They actuated a robot's leg to trigger the collapse of piles, while a vision transformer is trained to capture the avalanche dynamics.

Using Kim *et al.*'s [24] efficient sand model allows us to train an RL agent in an online manner overcoming shortcomings of suboptimal demonstrations in the data set as in [2]. We adopt the idea of using height maps as observations, but instead of a stroke-based approach, where end-effector motions are limited to straight lines between two x-y coordinates at a fixed z height in the granular medium, we let the agent freely decide which motions to take in any Cartesian direction at each step. Also, we do not limit the agent to binary heights so that our agent can shape structures

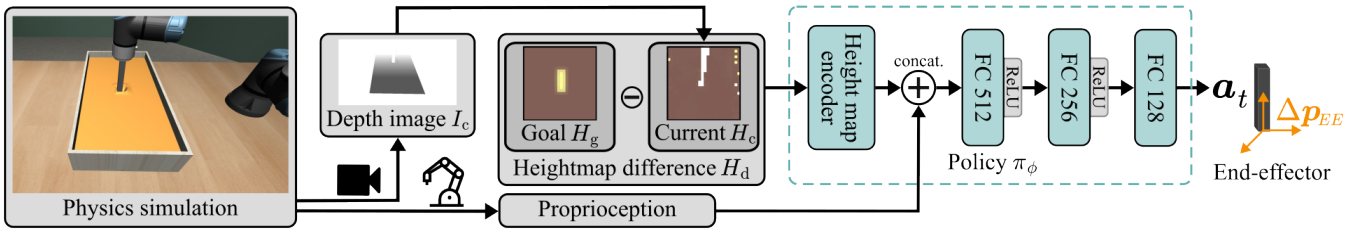


Fig. 2: Overview of our approach: We employ a training process to enable agents to manipulate granular media using sensory inputs. We train a visual policy via reinforcement learning to realize goal shape configurations using the difference between the current and the desired goal height map. The current height map is reconstructed from depth images. The height map difference is fed into our height map encoder and concatenated with the robot's proprioception. The resulting policy controls the end-effector to shape the granular media.

of varying depth. To perceive the state of the granular media, we make use of depth images like [3], [6], but we extract the relevant state information by representing them as height maps, which is computationally more efficient.

III. Method

In this section, we present our RL framework that learns how to move a robotic end-effector (EE) in granular media to create diverse goal shapes.

A. Overview

Fig. 2 provides an overview of our learning framework. Using observations from a physics simulation, we train a visual policy to manipulate granular media using the difference between the reconstructed current height map and the height map representing the desired goal configuration.

B. Task Description

The goal of the agent during one episode is to form the granular media according to a physically viable goal shape. We focus on shapes that deepen an initially flat surface. The shape is given in the form of a height map as shown in Fig. 2. When a training episode starts the robot is initialized to a random start configuration with the EE laying within a virtual cuboid of $30 \times 30 \times 5\,\mathrm{cm}^3$, which is located $2\,\mathrm{cm}$ above the granular media. The granular media is initialized as a flat bed at a height of $6\,\mathrm{cm}$. From the starting position, the agent has to move the EE through the media to shape it to resemble the goal height map as much as possible. An episode terminates after a fixed number of time steps N_{ep}. At the end of an episode, we reset the environment and the agent receives a new goal height map.

C. Architecture

The underlying idea of RL is to model and optimize transitions from one state to the next $s_t \rightarrow s_{t+1}$ as an MDP. In this process, the reward $r_t = r(s_t, a_t)$ incentivizes the RL agent to take actions $a_t = \pi_\phi(s_t)$ at the time step t with respect to a policy π_ϕ. Commonly, pairs of tuples (s_t, a_t, r_t, s_{t+1}) summarize states and actions. The agent's goal is to maximize the cumulative return $R = \sum_{i=t}^{T} \gamma^{(i-t)} r_t$ of the γ-discounted rewards.

RL Algorithm: In this work, we use the off-policy algorithm Truncated Quantile Critics (TQC) [28] with two critics and 25 quantiles, which showed the best training convergence among three tested RL algorithms (see Sec. IV-C). To assure the agent's exploration, we add Gaussian noise from a process $\mathcal{N}$ with a standard deviation σ_{ϵ_π} to the actions, so that $a_t = \pi_\phi(s_t) + \mathcal{N}(0, \sigma_{\epsilon_\pi})$.

Action Space: Per time step, our agent moves the robot's EE in continuous action increments $(\Delta x, \Delta y, \Delta z)$ using an operational space controller (OSC) [29]. At each step the agent can move the EE in increments of up to $4\,\mathrm{cm}$ in each direction or it can keep it still at the current position. We normalize all actions to values in the interval $[-1, +1]$.

Observation Space: The task-relevant information included in the observation space is shown in Tab. I. We normalize all observations to values in the interval $[-1, +1]$. For the EE, we include its current $(x_c, y_c, z_c)_{\text{EE}}$ and previous position $(x_p, y_p, z_p)_{\text{EE}}$ so that the agent can adequately scale the inputs for the OSC. Furthermore, we use the difference height map $H_{\text{d}} = H_{\text{g}} - H_{\text{c}}$, obtained from the goal H_{g} and the current height map H_{c}. During the reconstruction process of the current height map, we convert the current depth image I_{c} into 3D points and fit a grid on top of the ones within the granular media. The mean of all points that fall into one grid cell define the resulting cell elevation. To indicate the current EE position and the goal area in the same compact representation as the current height map, we create two Boolean masks of the same shape. The true values in the EE mask M_{EE} represent the two-dimensional projection to the xy-plane of the EE's base shape, while the true values in the goal mask M_{g} represent the grid cells that are unequal to the initialization height of the goal height map. To reduce the observation size, we combine the difference height map with the two masks using our height map encoder (see Fig. 2). It ensures that the information of the EE's current position and the goal area is available in the same

TABLE I: Observation space.

Observation	Notation	Size
Current EE position	$(x_c, y_c, z_c)_{\text{EE}}$	3
Previous EE position	$(x_p, y_p, z_p)_{\text{EE}}$	3
Difference height map	H_{d}	64
EE mask	M_{EE}	
Goal mask	M_{g}	
	Sum	70

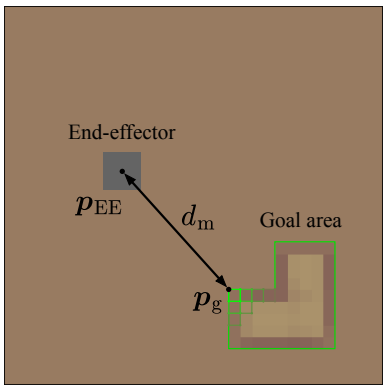


Fig. 3: We employ the distance d_{m} of the EE position $\boldsymbol{p}_{\text{EE}}$ to the closest point belonging to the goal area $\boldsymbol{p}_{\text{g}}$ to guide the agent towards regions that require manipulation through a dense reward signal (see Sec. III-D).

space as the height map elevation data. To achieve that, we first stack the difference height map with the EE mask. Then we extract feature vectors of size 64 from both, the stacked observation and the goal mask using a CNN and amplify the stacked features within the goal area by multiplying them with the sigmoid function of the goal mask, which works like a gating mechanism. The resulting feature vector of size 64 is concatenated with a vector of all other EE observations. The height map encoder is designed for inputs of size 32×32. Depending on the available manipulation area and goal area size, we reduce the input size by padding the outer cells with zero. In our implementation, all height maps and masks have a grid cell size of $1 \times 1\,\text{cm}^2$ and we limit the height map elevation values to $[0\,\text{cm}, 20\,\text{cm}]$.

D. Reward Function

Unlike rigid-body tasks, where distances to goal positions are well-defined and straightforward to compute, the high-dimensional configuration space of granular materials make reward shaping significantly more challenging. A random manipulation is often more likely to increase the distance to the goal configuration than to reduce it, resulting in an unbalanced reward signal where most actions are penalized, ultimately discouraging exploration. To mitigate this, we propose two complementary reward components that together provide informative and balanced feedback to guide the agent's learning, ultimately shaping the cells in the goal area, which we define as the set of grid cells where the target height map differs from the initial (flat) configuration.

Granular Media Shaping: The first component directly incentivizes reducing the discrepancy between the current and the goal state of the granular media. We consider two formulations for this reward.

The *delta* reward provides feedback proportional to the reduction in distance between the current and goal configuration. Reducing this distance leads to positive rewards. The reward is given by:

$$r_{delta} = \alpha_{\text{c}} \cdot (\hat{d}_{t\text{-}1} - \hat{d}_t), \tag{1}$$

where α_{c} is a scaling factor and $\hat{d}$ is the mean absolute difference between the goal height map and the truncated current height map. It is defined as:

$$\hat{d} = \frac{1}{N_{\text{cell}}} \sum_{i=1}^{N_{\text{cell}}} |h_{\text{g},i} - \min(h_0, h_{\text{c},i})|, \tag{2}$$

where $h_{\text{g},i}$ and $h_{\text{c},i}$ denote the height at a grid cell i in H_{g} and H_{c}, respectively, h_0 is the initial height level of the granular medium, and N_{cell} is the number of grid cells. Note that for $\hat{d}$, the height values of the current height map H_{c} are cut off at the initial flat height of the granular media as the goal height map H_{g} only contains negative shape imprints. In other words, piled-up granular medium is disregarded.

While r_{delta} is intuitive, its formulation can discourage exploration by penalizing intermediate actions that happen to increase the distance to the goal, but lead to it on the long run. To address this challenge, we draw inspiration from reward shaping techniques that have been employed for 6D object reposing [30]. The so-called *progressive* reward, rewards the agent for making progress relative to the best configuration reached so far within the current episode, rather than relative to the immediately preceding time step. In this case, the agent receives a reward when it changes the granular medium into a configuration that is closer to the target configuration than all other ones that have been reached before within the episode. Likewise, we penalize the agent for reaching configurations further from the target than the furthest one reached so far within the same episode. The progressive reward is defined as:

$$r_{prog} = \alpha_{\text{c}} \cdot \max(\hat{d}_{\text{closest}} - \hat{d}, 0) - \alpha_{\text{f}} \cdot \min(\hat{d}^o - \hat{d}^o_{\text{furthest}}, 0), \tag{3}$$

where $\hat{d}^o$ is the mean absolute difference of the current and the goal height map outside the goal area, $\cdot_{\text{closest}}$ and $\cdot_{\text{furthest}}$ denote the closest and furthest reached distances inside the current episode, respectively. This formulation avoids cycles of positive return.

For both proposed rewards r_{delta} and r_{prog}, we use the scaling factors $\alpha_{\text{c}} = 5,000$ and $\alpha_{\text{f}} = 1,000$.

Goal Area Movement: To encourage the agent to move its EE towards the region of interest and to stay within this area, we add a further reward term using the distance of the EE to the goal area. The agent is incentivized to bring its EE close to this area using a distance-based penalty, with a binary bonus for reaching the region. Formally, we define the reward as:

$$r_m = -\tanh(\alpha_{\text{m}} \cdot d_{\text{m}}) + \mathbb{1}_{\text{reached}}, \tag{4}$$

where d_{m} is the minimum Euclidean distance between the EE and the goal area, computed as visualized in Fig. 3 and $\mathbb{1}_{\text{reached}}$ is an indicator function that returns 1 when the EE is inside the goal area and 0 otherwise. We use a value of $\alpha_{\text{m}} = 10$ to control the steepness of the distance-based penalty.

The total reward is computed as the sum of the goal movement reward and the shaping reward:

$$r = r_{\text{m}} + r_{\text{s}}, \tag{5}$$

where r_{s} $\{r_{delta}, r_{prog}\}$.

IV. Experimental Evaluation

To demonstrate the performance of our RL approach to shape granular media compared to two baselines, we performed experiments with different goal height maps. We further tested different reward formulations and environment state observability modes. Additionally, we conducted an ablation study of the feature extractor and the choice of RL algorithm. More details of our method, the supplementary video, and our code are available on the paper website[1].

A. Baselines

We implemented two different baseline approaches for comparison, which we detail in this section.

Random Baseline: For the random baseline **(RAND)**, at the start of each evaluation episode we place the robot's EE at a random start position uniformly sampled within the goal mask M_g and at the surface of the granular medium. The robot then executes random actions for N_ep time steps, after which the episode strictly terminates.

Boustrophedon Coverage Path Planning Baseline: For this baseline **(B-CPP)**, we utilize Boustrophedon decomposition [31] combined with Coverage Path Planning (CPP), following the approach of Terenzi and Hutter [26]. First, we perform a flood fill on the Boolean goal mask M_g to extract connected regions. We then compute the centroid of each region and solve the traveling salesman problem on these centroids by performing a greedy nearest-neighbor search [32]. Within each region, we generate parallel sweep lines spaced by the EE's footprint and alternate direction on each pass to minimize the traversal overhead. Note that if the footprint partially covers a cell, it is rounded to a whole cell. For each grid coordinate in a sweep, we extract the target height and assemble a sequential list of waypoints. During an evaluation episode, the robot moves its EE to each waypoint following the order of the list, terminating once the full plan is executed. Hence, the length of an episode is defined by the number of waypoints.

B. Experimental Setup

We evaluate the effectiveness of our policies, by randomly selecting out of 400 distinct goal height maps in the beginning of each episode. We distinguish between heuristic baselines, policies trained in a privileged setting with full state observability, and policies using observations that are obtainable in the real world, consequently suffering, e.g., from occlusions. Tab. II lists all relevant RL training parameters.

For all experiments, we use the 6-DoF robotic arm UR5e, equipped with a custom, cubic end-effector (with dimensions of $2\times2\times15\,\mathrm{cm}^3$), and an external ZED 2i camera, as shown in Fig. 1. However, our approach is applicable to any robotic arm, since our policies learn EE movement increments instead of robot joint positions. Furthermore, we rely on the RL algorithm implementations of Stable-Baselines3 [33], and the OpenAI Gym toolkit [34], as well as robosuite [35] that is based on MuJoCo [36] for simulation, and the Robotics Toolbox for Python [37].

[1]Paper website: https://humanoidsbonn.github.io/granular_rl/

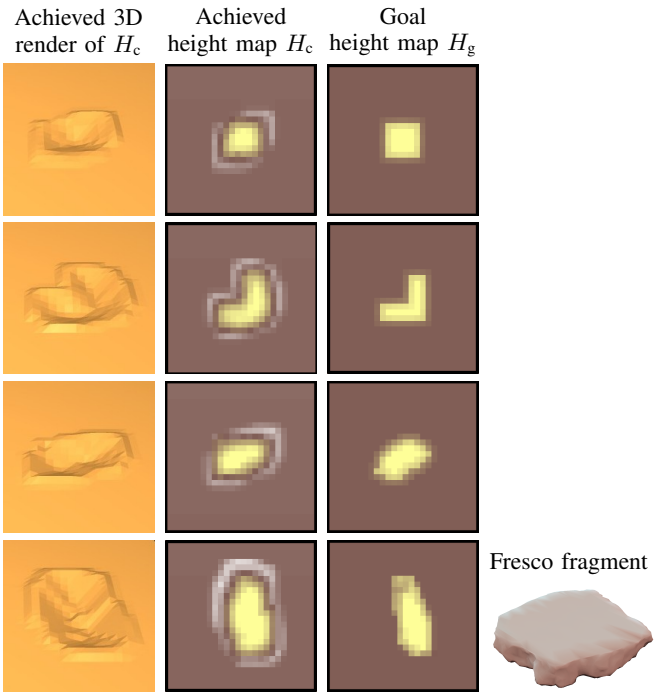


Fig. 4: From left to right, we show the reconstructed 3D scene in simulation, the reconstructed height map after the manipulation, and the goal height map the agent aims to achieve. From top to bottom, the given goal shapes are an exemplary rectangle, an L, a polygon, and a fresco fragment's negative.

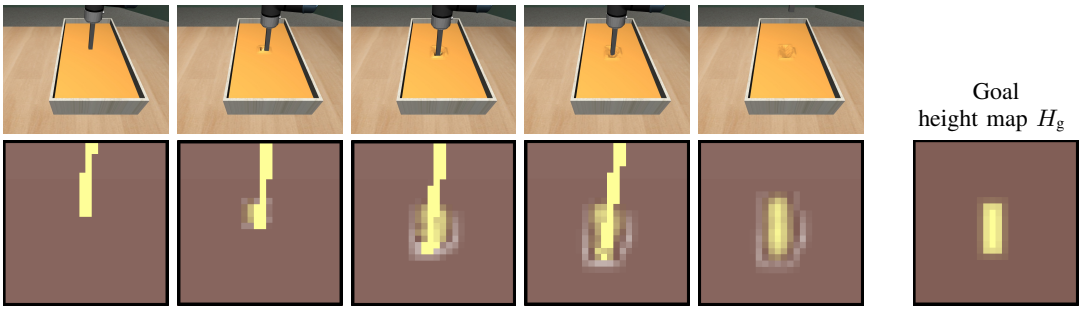
Goal Height Maps: We evaluate on a wide range of different shapes represented as goal height maps: 100 rectangles, 100 L-shapes, 100 polygons, and 100 negatives of archaeological fresco fragments. The goal area of the shapes are up to $10\times10\,\mathrm{cm}$ with varying heights of up to $3\,\mathrm{cm}$. We randomly place the shapes within the granular media, resulting in various configurations of each shape. All target shapes are designed such that they are not achievable by the agent executing a single stroke through the granular media.

To generate the fresco fragment goals, we place three-dimensional scans of fresco fragments from the RePAIR dataset [38] into granular media such that their painted surfaces are parallel and $5\,\mathrm{mm}$ above the surface of the granular media. Fig. 4 shows an examplary 3D render of an utilized fresco fragment, which was scanned by archaeologists.

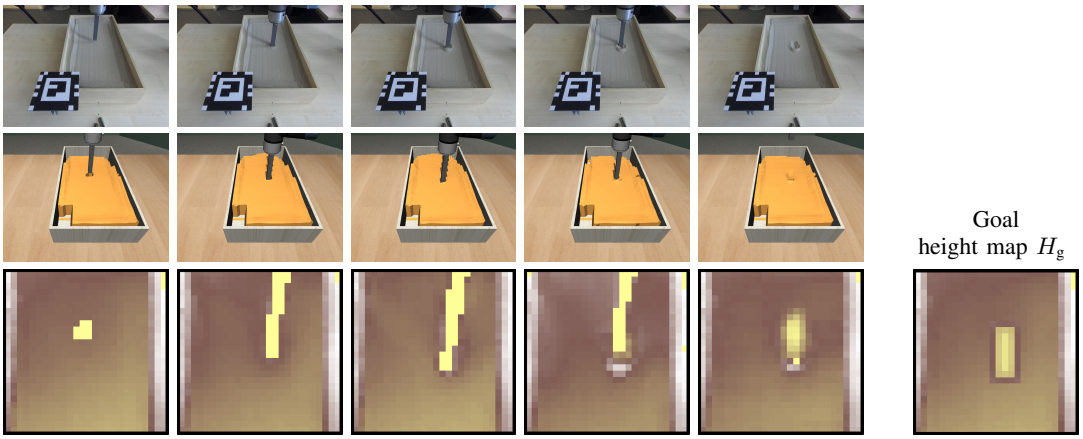
On the real robotic system, we initially measure the mean height of the goal area and fit the goal shape onto the surface of the granular media. This results in an adjusted goal height

TABLE II: RL notations and training settings.

Notation	TQC / SAC / TD3	Description
N_ep	40	Number of steps per episode
N_crit	2	Number of critics
B_E	$1\cdot10^5$	Experience replay buffer size
B_G	256	Minibatch for each gradient update
l_T	$3\cdot10^{-4}$	Learning rate
γ	0.99	Discount factor
σ_{ϵ_π}	0.2	Std. deviation of exploration noise ϵ_π
τ	0.005	Soft update coefficient



a) The agent is forming the given goal height map in the **simulated environment** (top). The resulting current reconstructed height maps (bottom) include occlusions of the end-effector. At the start of the episode (left) the granular medium is perfectly flat and in the end the desired goal shape is visible in it (right).



b) We deploy the agent to the **real robot** in a zero-shot manner, without any additional training. From top to bottom, we show the raw camera view of the box filled with the granular medium, the reconstructed 3D scene in simulation, and the reconstructed height map that the agent observes, while it manipulates the granular medium to match the desired goal configuration. Note that the granular medium is not perfectly flat like in the simulated scenario. Despite noisy observations from the depth camera, the agent is able to create the desired rectangle shape in the real world.

Fig. 5: Manipulation of the granular medium by a visual, goal-conditioned RL agent in simulation (top) and on the real robotic system (bottom).

map that considers the unevenness of the media's initial surface.

Note that the larger the task, the completion time scales in proportion to the dimensions of the goal area. Hence, while larger shapes are trainable, the chosen dimensions are a trade-off between training time and showcasing the agent's capabilities.

***Metrics**:* Based on the difference between the goal height map and the current height map at the end of 100 evaluation episodes, we calculate the absolute mean cell height difference $\hat{d}$ **(Height Diff.)** within the goal area. Furthermore, we determine the percentage of grid cells within the goal area that have been changed **(Changed)** by the EE.

C. Experimental Results

The policies optimize the reward components discussed in Sec. III-D, iteratively refining its manipulation strategy to shape the granular medium into the desired configuration. The results presented in the first column of Tab. III, show that our best policy (DELTA) achieves a remaining absolute mean cell height difference $\hat{d}$ of up to $3.4\,\mathrm{mm}$ over all goal area cells, which is close to the privileged setting, and outperforming two baselines. This indicates that the trained policy exhibits proficient manipulation behaviors, given that the agent has to learn the dynamics of collapsing cells to reach the desired goal configuration.

***Qualitative Results**:* For a qualitative assessment, we visualize rollouts of the policy, displaying the resulting height maps in Fig. 5. The rollouts show the robot forming a rectangle shape in simulation and in the real world. Furthermore, Fig. 4 shows the qualitative results of several goal shapes ranging from a simple rectangular shape to a complex shape resulting from the negative of an archaeological fragment.

***Quantitative Results**:* To understand the effectiveness of our proposed reward design, we compare the performance of our trained policy using the delta reward **(DELTA)** against the progressive reward **(PROG)** and the reward ablation where we remove the goal area movement reward **(NO-M)**. First, removing the goal area movement reward leads to a policy that entirely avoids manipulation behaviors, since discovering strategies that shape the granular medium to match the desired configuration are challenging to discover without any guidance. As a result, the policies perform no better than a random baseline **(RAND)**. Second, even though the Boustrophedon Coverage Path Planning baseline **(B-CPP)** changes all goal cells, our approach has $31\,\%$ higher accuracy in achieving the target heights than the B-CPP baseline. Third, using the reconstructed height map H^{R}

TABLE III: Quantitative evaluation results.

Metric	Reconstructed Height Map H^{R}			Privileged Height Map H^{P}			Baselines	
	DELTA (OUR)	PROG	NO-M	DELTA	PROG	NO-M	RAND	B-CPP
a) Height Diff. [mm] ↓	$\mathbf{3.4 \pm 1.1}$	4.5 ± 1.9	6.0 ± 1.8	$\mathbf{3.3 \pm 1.0}$	4.5 ± 1.8	6.1 ± 1.9	7.2 ± 2.5	$\mathbf{4.8 \pm 1.0}$
b) Changed [%] ↑	97.4 ± 10.4	$\mathbf{98.2 \pm 7.7}$	3.4 ± 12.8	95.5 ± 14.9	$\mathbf{98.7 \pm 5.6}$	1.0 ± 6.7	53.6 ± 33.8	$\mathbf{100.0 \pm 0.0}$

The metrics are based on the manipulated cells within the goal area during 100 evaluation episodes. We distinguish between an evaluation setting using the the reconstructed height map H^{R} (left) containing occlusions and the privileged height map H^{P} (center). On average DELTA achieves the lowest values for the Height Diff. metric compared to both baselines (right). While B-CPP changes all goal cells, the trained RL agents relying on either one of the two reward formulations (DELTA and PROG) change more cells to the correct height. When removing the goal area movement reward (NO-M), the agents learn to completely avoid manipulations, which leads to the second highest Height Diff. mean values, lower than the random baseline (RAND), but higher than the B-CPP baseline.

TABLE IV: Feature extractor ablation results.

Metric	DELTA (OUR)	ABL-CNN	ABL-IMG
a) Height Diff. [mm] ↓	$\mathbf{3.4 \pm 1.1}$	3.7 ± 1.2	4.6 ± 1.7
b) Changed [%] ↑	97.4 ± 10.4	$\mathbf{97.7 \pm 6.2}$	95.0 ± 12.5

Our visual feature extractor (DELTA) shows the best performance for the Height Diff. metric compared to two ablated versions using CNN-based extractors without the gating mechanism (ABL-CNN) and relying on pure depth images (ABL-IMG) instead of the reconstructed height map H^{R}.

in simulation leads to similar performance as relying on the privileged height map H^{P}, while only the former is deployable to the real world. In summary, our results indicate that the full reward formulation significantly accelerates learning and improves the final performance compared to the ablated version (NO-M).

Feature Extractor Ablation: To analyze the effectiveness of our feature extractor described in Sec. III-C, we compare its performance to two ablated versions: The first one (**ABL-CNN**) does not contain the gating mechanism and uses a three-channel CNN for the height map and the two mask observations instead. The second ablated version (**ABL-IMG**) also uses a CNN-based encoder, but instead of the reconstructed difference height map it directly uses the depth image from the camera together with the two mask observations. The quantitative results in Tab. IV show that using our feature extractor with the delta reward achieves the best performance compared to both ablated versions, with a mean height difference of $3.4\,\mathrm{mm}$. Relying on pure depth images (ABL-IMG) prevents the trained agent from learning suitable features, such that it is not able to correctly lower the goal cells, resulting in a mean height difference of $4.6\,\mathrm{mm}$.

RL Algorithm Ablation: Furthermore, we compare the performance of the TQC algorithm with two other commonly used off-policy RL algorithms, namely Soft Actor-Critic (SAC) [39] and Twin Delayed Deep Deterministic Policy Gradient (TD3) [40], using the same training parameters (see Tab. II) and three different seeds. Fig. 6 shows the mean rewards with three different training seeds. Note that the mean reward values are based on 25 evaluation episodes. Training with TQC leads to a fast convergence and it outperforms SAC in terms of mean rewards. Using TD3 does not converge to positive rewards, validating our design choice to utilize TQC.

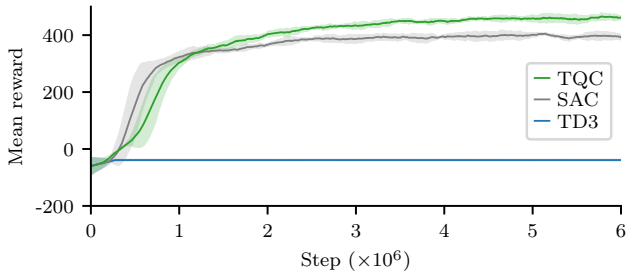


Fig. 6: Using three different seeds, the RL training with TQC [28] results in higher mean rewards compared to SAC [39], while TD3 [40] does not converge showing poor performance.

Real World Transfer: Finally, we zero-shot deployed the agent trained in simulation to the real robotic system. Fig. 5 b shows one qualitative episode with an exemplary rectangle shape. The performance of our agent in the real world is similar to the one in simulation (compare Fig. 5 a and Fig. 5 b), which demonstrates that our agent can successfully be deployed on a real robot.

V. Conclusion

Manipulating granular media requires interactive behaviors that adapt to changes in the medium's state in a closed-loop fashion. However, traditional modeling approaches require extensive engineering efforts to shape granular media due to its high-dimensional configuration space and its deformable nature. Overcoming these shortcomings with reinforcement learning has remained a challenge. In this work, we addressed this by developing suitable observation representations and reward functions that together enable a stable and efficient training.

Our approach outperforms two baselines in terms of target shape accuracy, achieving the lowest difference between the desired and final medium configuration. Furthermore, we demonstrated that policies trained entirely in simulation using depth-based observations can transfer zero-shot to real-world robotic systems, underscoring the applicability and robustness of our method.

References

[1] C. Schenck, J. Tompson, S. Levine, and D. Fox, "Learning robotic manipulation of granular media," in *Proc. of Conf. on Robot Learning (CoRL)*, 2017.

[2] S. Jin, Z. Ye, and L. Zhang, "Learning excavation of rigid objects with offline reinforcement learning," *arXiv preprint*, 2023.

[3] Y. Zhu, P. Thangeda, M. Ornik, and K. Hauser, "Few-shot adaptation for manipulating granular materials under domain shift," in *Proc. of Robotics: Science and Systems (RSS)*, 2023.

[4] D. Jud, I. Hurkxkens, C. Girot, and M. Hutter, "Robotic embankment: Free-form autonomous formation in terrain with heap," *Construction Robotics*, vol. 5, no. 2, 2021.

[5] P. Egli, D. Gaschen, S. Kerscher, D. Jud, and M. Hutter, "Soil-adaptive excavation using reinforcement learning," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 4, 2022.

[6] T. Osa and M. Aizawa, "Deep reinforcement learning with adversarial training for automated excavation using depth images," *IEEE Access*, vol. 10, 2022.

[7] I. Kurinov, G. Orzechowski, P. Hamalainen, and A. Mikkola, "Automated excavator based on reinforcement learning and multibody system dynamics," *IEEE Access*, vol. 8, 2020.

[8] Q. Lu, Y. Zhu, and L. Zhang, "Excavation reinforcement learning using geometric representation," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, 2022.

[9] A. Cherubini, V. Ortenzi, A. Cosgun, R. Lee, and P. Corke, "Model-free vision-based shaping of deformable plastic materials," *Intl. Journal of Robotics Research (IJRR)*, vol. 39, no. 14, 2020.

[10] J. Matas, S. James, and A. J. Davison, "Sim-to-real reinforcement learning for deformable object manipulation," in *Proc. of Conf. on Robot Learning (CoRL)*, 2018.

[11] T. Chen, J. Xu, and P. Agrawal, "A system for general in-hand object re-orientation," in *Proc. of Conf. on Robot Learning (CoRL)*, 2022.

[12] B. Font, F. Alcántara-Ávila, J. Rabault, R. Vinuesa, and O. Lehmkuhl, "Deep reinforcement learning for active flow control in a turbulent separation bubble," *Nature Communications*, vol. 16, no. 1, 2025.

[13] Y. Zhu, L. Abdulmajeid, and K. Hauser, "A data-driven approach for fast simulation of robot locomotion on granular media," in *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[14] A. Karsai, D. Kerimoglu, D. Soto, S. Ha, T. Zhang, and D. I. Goldman, "Real-time remodeling of granular terrain for robot locomotion," *Advanced Intelligent Systems*, vol. 4, no. 12, 2022.

[15] D. Kerimoglu, D. Soto, M. L. Hemsley, J. Brunner, S. Ha, T. Zhang, and D. I. Goldman, "Learning manipulation of steep granular slopes for fast mini rover turning," in *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2024.

[16] J. Xu, Y. Jia, D. Yang, P. Meng, X. Zhu, Z. Guo, S. Song, and M. Ciocarlie, "Tactile-based object retrieval from granular media," *arXiv preprint*, 2024.

[17] Y. Miron, C. Ross, Y. Goldfracht, C. Tessler, and D. Di Castro, "Towards autonomous grading in the real world," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2022.

[18] N. Alatur, O. Andersson, R. Siegwart, and L. Ott, "Material-agnostic shaping of granular materials with optimal transport," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2023.

[19] S. Xue, S. Cheng, P. Kachana, and D. Xu, "Neural field dynamics model for granular object piles manipulation," in *Proc. of Conf. on Robot Learning (CoRL)*, 2023.

[20] C. Pavlov and A. M. Johnson, "Soil displacement terramechanics for wheel-based trenching with a planetary rover," in *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2019.

[21] C. Liu, Y. Li, and K. Hauser, "Localized graph-based neural dynamics models for terrain manipulation," *arXiv preprint*, 2025.

[22] I. Hurkxkens, A. Mirjan, F. Gramazio, M. Kohler, and C. Girot, "Robotic landscapes: Designing formation processes for large scale autonomous earth moving," in *Impact: Design With All Senses*, 2020.

[23] N. Tuomainen, D. Blanco-Mulero, and V. Kyrki, "Manipulation of granular materials by learning particle interactions," *IEEE Robotics and Automation Letters (RA-L)*, vol. 7, no. 2, 2022.

[24] W. Kim, C. Pavlov, and A. M. Johnson, "Developing a simple model for sand-tool interaction and autonomously shaping sand," *arXiv preprint*, 2019.

[25] D. Jud, S. Kerscher, M. Wermelinger, E. Jelavic, P. Egli, P. Leemann, G. Hottiger, and M. Hutter, "Heap - the autonomous walking excavator," *Automation in Construction*, vol. 129, 2021.

[26] L. Terenzi and M. Hutter, "Toward autonomous excavation planning," *IEEE Transactions on Field Robotics (T-FR)*, vol. 1, 2024.

[27] H. Hu, F. Qian, and D. Seita, "Learning granular media avalanche behavior for indirectly manipulating obstacles on a granular slope," in *Proc. of Conf. on Robot Learning (CoRL)*, 2025.

[28] A. Kuznetsov, P. Shvechikov, A. Grishin, and D. P. Vetrov, "Controlling overestimation bias with truncated mixture of continuous distributional quantile critics," in *Proc. of the Intl. Conf. on Machine Learning*, 2020.

[29] O. Khatib, "A unified approach for motion and force control of robot manipulators: The operational space formulation," *IEEE Journal of Robotics and Automation*, vol. 3, no. 1, 1987.

[30] A. Petrenko, A. Allshire, G. State, A. Handa, and V. Makoviychuk, "Dexpbt: Scaling up dexterous manipulation for hand-arm systems with population based training," in *Proc. of Robotics: Science and Systems (RSS)*, 2023.

[31] H. Choset, "Coverage of Known Spaces: The Boustrophedon Cellular Decomposition," *Autonomous Robots*, vol. 9, no. 3, 2000.

[32] D. J. Rosenkrantz, R. E. Stearns, and P. M. Lewis, "Approximate algorithms for the traveling salesperson problem," in *Proc. of the Annual Symposion on Switching and Automata Theory (SWAT)*, 1974.

[33] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann, "Stable-baselines3: Reliable reinforcement learning implementations," *Journal of Machine Learning Research (JMLR)*, vol. 22, no. 268, 2021.

[34] G. Brockman, V. Cheung, L. Pettersson, J. Schneider, J. Schulman, J. Tang, and W. Zaremba, "Openai gym," *arXiv preprint*, 2016.

[35] Y. Zhu, J. Wong, A. Mandlekar, R. Martín-Martín, A. Joshi, K. Lin, A. Maddukuri, S. Nasiriany, and Y. Zhu, "robosuite: A modular simulation framework and benchmark for robot learning," *arXiv preprint*, 2020.

[36] E. Todorov, T. Erez, and Y. Tassa, "Mujoco: A physics engine for model-based control," in *Proc. of the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, 2012.

[37] P. Corke and J. Haviland, "Not your grandmother's toolbox - the robotics toolbox reinvented for python," in *Proc. of the IEEE Intl. Conf. on Robotics & Automation (ICRA)*, 2021.

[38] T. Tsesmelis *et al.*, "Re-assembling the past: The repair dataset and benchmark for real world 2d and 3d puzzle solving," in *Proc. of the Conf. on Neural Information Processing Systems (NIPS)*, 2025.

[39] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. of the Intl. Conf. on Machine Learning*, 2018.

[40] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," in *Proc. of the Intl. Conf. on Machine Learning*, 2018.