Celebi's Choice: Causality-Guided Skill Optimisation for Granular Manipulation via Differentiable Simulation[†]

Minglun Wei[1,*], Xintong Yang[1,*], Junyu Yan[2], Yu-Kun Lai[3], Ze Ji[1]

I. INTRODUCTION

Robotic manipulation of deformable objects remains challenging due to their nonlinear and unpredictable responses under force. Granular materials like soil further complicate control by exhibiting both particulate and continuum properties. As a representative case, soil manipulation is central to automated container-based agriculture, where robots must perform accurate excavation and levelling. However, physical trials are costly, and standard reinforcement learning (RL) methods often suffer from sample inefficiency and unstable dynamics in such settings.

Differentiable physics (DP) offers an efficient gradient-based approach to robotic skill and trajectory optimisation in contact-rich environments [1]. Yet, standard DP methods typically update all control parameters uniformly, regardless of their actual influence on outcomes, leading to inefficient learning, increased computational cost, and instability in high-dimensional control spaces.

To address this, we propose **Celebi**—causality-enhanced soil levelling and excavation skill optimisation via backpropagation with physical information. Inspired by Celebi's mythical foresight, our method leverages causal analysis to identify how skill parameters *causally* influence task-relevant outcomes, enabling adaptive gradient step sizes and direction correction.

Celebi integrates a differentiable simulator to model granular dynamics and defines skill parameters mapped to control inputs. During optimisation, structured features are extracted from height maps, and their causal effects guide parameter updates. Experiments in both simulation and the real world show that Celebi achieves faster convergence, improved stability, and precise soil manipulation, with strong sim-to-real transfer.

II. METHOD: DIFFERENTIABLE OPTIMISATION

A. Problem Formulation

We formulate skill optimisation for robotic soil manipulation as a differentiable trajectory optimisation problem over parameterised skills Θ. These parameters define robot motion primitives and are optimised via gradient descent through differentiable simulation and causal guidance. At each epoch j, we minimise the task loss between the final observation $\mathbf{o}_T^{(j)}$ and the target state $\mathbf{o}^{\text{target}}$:

$$\min_{\Theta} \mathcal{L}(\mathbf{o}_T^{(j)}, \mathbf{o}^{\text{target}}) \tag{1}$$

The control sequence $\mathcal{U}^{(j)}$ is generated by mapping $\Theta^{(j)}$ through a differentiable function $\mathbf{g}(\cdot)$, and the resulting states $\mathcal{X}^{(j)}$ are obtained via simulation function $\mathbf{f}(\cdot)$. The skill parameters are updated as:

$$\Theta^{(j)} = \Theta^{(j-1)} - \boldsymbol{\alpha}^{(j)} \odot \mathcal{G}\left(\nabla_{\Theta^{(j-1)}}\mathcal{L}\right) \tag{2}$$

where $\mathcal{G}$ corrects gradient directions, and $\boldsymbol{\alpha}^{(j)} = \mathcal{C}(\mathbf{o}_T^{(j-1)}, \mathbf{o}^{\text{target}})$ scales the step sizes based on causal effects.

B. Task Specification

We consider two soil manipulation tasks: excavating planting pits and levelling soil to cover seeds. The system state $\mathbf{x}_i$ includes full particle-level data, while the robot only observes partial information $\mathbf{o}_i$ from a depth camera. These observations are converted into heightmaps for loss computation.

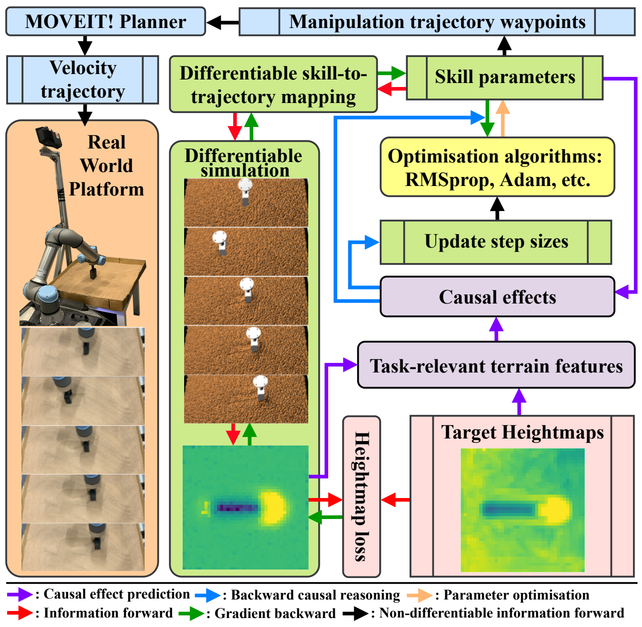

MOVEIT! Planner
Manipulation trajectory waypoints
Velocity trajectory
Differentiable skill-to-trajectory mapping
Skill parameters
Real World Platform
Differentiable simulation
Optimisation algorithms: RMSprop, Adam, etc.
Update step sizes
Causal effects
Task-relevant terrain features
Heightmap loss
Target Heightmaps
: Causal effect prediction
: Backward causal reasoning
: Parameter optimisation
: Information forward
: Gradient backward
: Non-differentiable information forward

Fig. 1: Overall workflow of our method. Orange: real-world manipulation system. Green: differentiable simulation, skill parameters, and skill-to-trajectory mapping. Blue: ROS/MOVEIT!-based motion planning. Purple: causal reasoning. Pink: manipulation target and loss function. Colour-coded arrows represent information directions.

*Equal contribution, order determined by alphabetical order of last names.

[†]This paper has been accepted by the 2025 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2025). Corresponding author: Ze Ji. JiZ1@cardiff.ac.uk

[1]School of Engineering, Cardiff University, Cardiff, CF24 3AA, United Kingdom.

[2]School of Engineering, The University of Edinburgh, Edinburgh, EH9 3FB, United Kingdom.

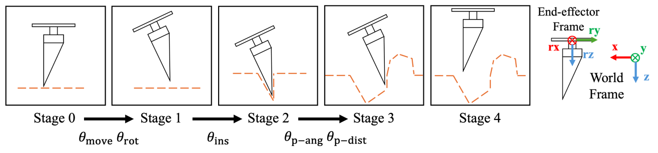
[3]School of Computer Science and Informatics, Cardiff University, Cardiff, CF24 4AG, United Kingdom.

Excavation starts from a flat soil surface, with targets derived from demonstrations. Levelling begins from the excavation results and aims to restore flatness using a reconstruction pipeline [2]. Robot actions are 6D Cartesian displacements $\mathbf{u} \in \mathbb{R}^6$, derived from low-dimensional skill parameters Θ via a differentiable mapping $\mathbf{g}(\cdot)$. These are executed using MOVEIT! [3] in the real world. Task performance is evaluated via pixel-wise L_1 loss between the target and observed heightmaps.

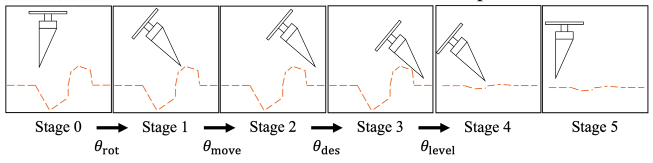
C. Skill Parameters

Long-horizon manipulation in 6D space is challenging due to the high dimensionality of the control sequence. To abstract this, we derive skill parameters Θ from human demonstrations, with each parameter bounded by $\theta^{\min}$ and $\theta^{\max}$. Each Θ is mapped via $\mathbf{g}(\cdot)$ to a control sequence $\mathcal{U}$ over horizon T (see Fig. 2 and Appendix V-A).

Given fixed translational and rotational velocities, we compute the number of steps per stage by dividing the displacement in each control dimension by the corresponding velocity, rounding to the nearest integer. The resulting per-step actions form $\mathbf{u}$ across time, yielding the full control sequence $\mathcal{U}$.



(a) Excavation Task Process and Frame Representation.



(b) Levelling Task Process Visualisation.

Fig. 2: Visualisation of the Excavation and Levelling Tasks.

D. Differentiable Simulation

We employ the Moving Least Squares Material Point Method (MLS-MPM) [4] combined with the St. Venant-Kirchhoff elastic energy and the Drucker-Prager yield criterion [5] to approximate the granular material dynamics in our simulator. A simplified process is detailed in Appendix V-B.

We enable end-to-end optimisation of skill parameters via differentiable physics, computing gradients of the task loss with respect to parameters through the full pipeline, including perception, simulation, and control. All components are differentiable, and non-smooth operations such as rounding in the skill-to-action mapping are handled to preserve gradient flow.

III. METHOD: CAUSALITY GUIDANCE

We propose a causality-guided optimisation method that improves trajectory learning by selectively adjusting parameter updates based on their causal influence on task-relevant features. Our approach includes structured feature extraction, causal effect estimation, and causality-informed gradient descent.

A. Feature Extraction

Directly establishing causal relationships between skill parameters and raw height maps is challenging due to their high dimensionality and unstructured nature. To address this, we extract lower-dimensional features using morphological operations to construct hole and peak maps. Binary masks and connected regions are further processed to define task-specific features. We define task-specific feature sets for causal analysis. For excavation, $\Lambda_e = \lambda_d, \lambda_s, \lambda_l$ captures the depth, start location, and length of the largest hole. For levelling, $\Lambda_l = \lambda_{ha}, \lambda_{hs}, \lambda_{ps}, \lambda_{pe}$ includes the area and initial point of the first hole, as well as the start and end points of the first peak, providing a compact representation of surface flatness.

B. Causal Effect Estimation

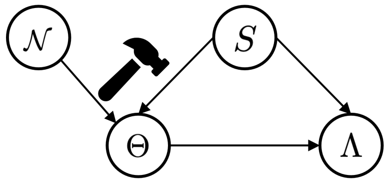


Fig. 3: SCM in our method. The solid arrow indicates that the parent node causes the child node; the hammer represents the do operator that removes the parents of the node. S is the environment bias variable and $\mathcal{N}$ denotes the noise variable.

We model the causal relationship between skill parameters Θ and extracted features Λ using a structural causal model (SCM) [6], as shown in Fig. 3. To isolate direct causal effects, we block confounding paths through environmental variables by applying *do*-interventions. We quantify influence using a normalised form of Average Causal Effect (ACE), extended from binary to continuous variables. The normalised ACE of a skill parameter θ_n at value β on feature λ_m is defined as:

$$ACE_{do(\theta_n=\beta)} = \frac{\mathbb{E}[\lambda_m \mid do(\theta_n = \beta)]}{\mathbb{E}[\lambda_m \mid do(\theta_n = 0)]} - 1 \qquad (3)$$

By sampling β within the valid range of θ_n and evaluating corresponding feature responses, we compute directional ACE scores that indicate the strength and polarity of each parameter-feature relationship. The computed causal effect maps are provided in Appendix V-C.

C. Causality-Guided Gradient Descent

Excavation: Causal effects are used to adapt both the step size and direction of parameter updates. Step sizes are computed based on feature differences from target values, scaled using a sigmoid function. For parameters with strong causal influence, gradients are corrected in the direction indicated by ACE to ensure progress toward desired outcomes.

Levelling: Since the task involves returning to a flat surface, we define discrete surface phases based on current feature values. Parameters are selectively updated depending

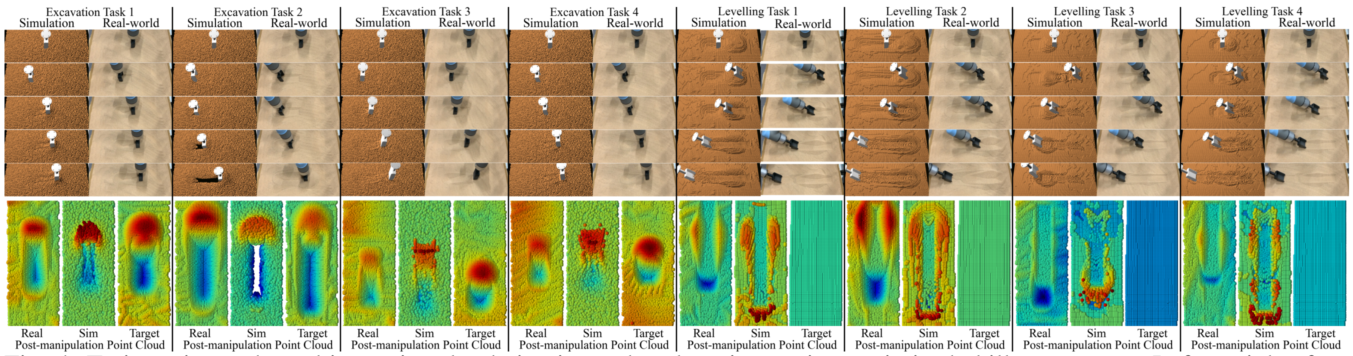
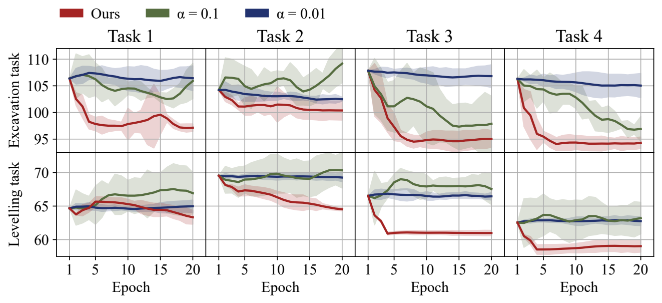


Fig. 4: Trajectories and resulting point clouds in sim and real settings using optimised skill parameters. Left to right: four excavation and four levelling tasks; time flows top to bottom. Levelling starts from excavation end states in the real world.

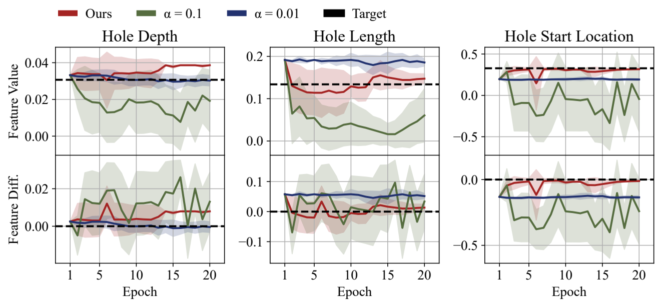
on their causal relevance to the active phase, with fixed step sizes. Gradient directions are similarly corrected to drive the system out of undesirable phases. The definition of these phases is detailed in Appendix V-C.

IV. EXPERIMENTS AND RESULTS

Our method is evaluated on real and simulated setups with a UR5e arm and 3D-printed shovel. Appendix V-D provides details. A high-resolution Zivid depth camera [7] captures the surface point cloud. In simulation, we replicate the same setup in the real world, including robot, container, and tool models. Physical parameters are identified using our previous DPSI framework [2] to ensure alignment between simulated and real-world dynamics. We optimise skill parameters within $[-1, 1]$, using RMSprop [8] with $\beta_r = 0.9$. The simulation runs at $\Delta t = 0.01$ s with 20 substeps.



(a) Optimisation loss curves of excavation (top) and levelling (bottom).



(b) Feature values and distances to targets for excavation task 1.

Fig. 5: Optimisation performances for both tasks and case study for excavation task 1.

We design four excavation tasks, each followed by a corresponding levelling task, yielding eight total scenarios. Each is run with five random seeds. Our method is compared to baselines with fixed step sizes ($\alpha = 0.1$ and 0.01).

As shown in Fig. 5a (top), our method consistently converges within 10 epochs for excavation, achieving lower loss and variance than baselines. In-depth feature analysis (Fig. 5b bottom) on Excavation Task 1 shows that our method aligns hole shape and location more closely with targets, while maintaining stable convergence. Although depth is slightly underestimated in some cases, this corresponds to a negligible causal effect, which intentionally suppresses its update.

For levelling, Fig. 5a (bottom) shows faster convergence and lower loss across most tasks. Despite the absence of explicit feature tracking in this phase, our method reduces loss efficiently and maintains stability throughout.

Trajectory visualisations in Fig. 4 highlight successful sim-to-real transfer. Excavation trajectories produce consistent hole shapes, with minor sim-to-real differences attributed to material adhesion in simulation. Levelling skills flatten the terrain effectively, though large holes may remain partially unfilled due to one-shot execution in the real world. Overall, our method demonstrates robust convergence, low variance, and strong transfer across tasks.

V. CONCLUSIONS

This paper proposes an optimisation method that integrates differentiable physics simulation with causality-based adaptive step-size adjustment and gradient direction correction by modelling the causal effects between manipulation results and skill parameters. Experimental results demonstrate notably improved optimisation stability, accuracy, and convergence efficiency in excavation and levelling tasks in simulation, and high transferability to real-world environments.

ACKNOWLEDGMENT

Minglun Wei was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) through a Doctoral Training Partnership (No. EP/W524682/1). This work was also partially supported by the UK EPSRC grant No. EP/X018962/1.

REFERENCES

[1] Y. Hu, L. Anderson, T.-M. Li, *et al.*, "Difftaichi: Differentiable programming for physical simulation," *arXiv preprint arXiv:1910.00935*, 2019.

[2] X. Yang, Z. Ji, and Y.-K. Lai, "Differentiable physics-based system identification for robotic manipulation of elastoplastic materials," *The International Journal of Robotics Research*, 2025. DOI: 10.1177/02783649251334661.

[3] M. Görner, R. Haschke, H. Ritter, and J. Zhang, "Moveit! task constructor for task-level motion planning," in *2019 International Conference on Robotics and Automation (ICRA)*, IEEE, 2019, pp. 190–196.

[4] Y. Hu, Y. Fang, Z. Ge, *et al.*, "A moving least squares material point method with displacement discontinuity and two-way rigid body coupling," *ACM Transactions on Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.

[5] G. Klár, T. Gast, A. Pradhana, *et al.*, "Drucker-prager elastoplasticity for sand animation," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–12, 2016.

[6] S. Bongers, P. Forré, J. Peters, and J. M. Mooij, "Foundations of structural causal models with cycles and latent variables," *The Annals of Statistics*, vol. 49, no. 5, pp. 2885–2915, 2021.

[7] Zivid, *Zivid one plus 3d cameras*, https://www.zivid.com/zivid-one-plus, Accessed: 2024-01-15, 2024.

[8] T. Tieleman and G. Hinton, "Rmsprop: Divide the gradient by a running average of its recent magnitude. coursera: Neural networks for machine learning," *COURSERA Neural Networks Mach. Learn*, vol. 17, 2012.

APPENDIX

A. Skill Parameter Definitions

Excavate: The excavation skill is parameterised by $\Theta_e = \{\theta_{\text{move}}, \theta_{\text{rot}}, \theta_{\text{ins}}, \theta_{\text{p-ang}}, \theta_{\text{p-dist}}\}$, with each parameter governing a specific motion primitive. As illustrated in Fig. 2a, this process has four stages. It begins with translating the shovel along the world's x-axis by d_{move} while simultaneously rotating it about its own x-axis by an angle ϕ_{rot}. Next, the tool is inserted into the material along its pointing direction to a depth of d_{ins}. This is followed by a pushing motion at an angle ϕ_{push} over a distance d_{push}. Finally, the tool is returned to its neutral angle and lifted by d_{lift}.

Level: The levelling skill is parameterised by $\Theta_l = \{\theta_{\text{rot}}, \theta_{\text{move}}, \theta_{\text{des}}, \theta_{\text{level}}\}$. As shown in Fig. 2b, this task consists of five consecutive stages. It begins with rotating the tool around the x-axis by an angle ϕ_{rot}. Next, the tool translates along the world's x-axis by d_{move}. Following this, it descends along the z-axis by d_{des}. The tool then moves along the x-axis by d_{level} to smooth the surface. Finally, it lifts and resets as in excavation.

B. Substep Execution in Simulation

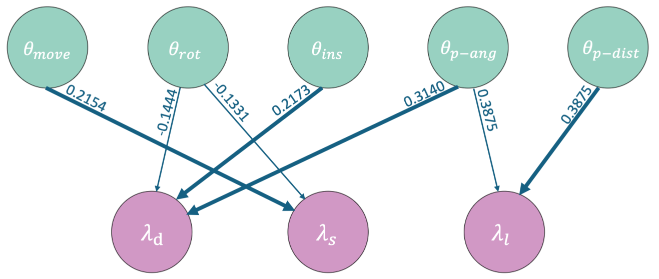
For the i-th time step with a time step size Δt, the update process is refined through N_{sub} substeps, each with a duration of $\Delta t_{\text{sub}} = \Delta t / N_{\text{sub}}$. Thus, the control input for each substep is $\mathbf{u}^{\text{sub}} = \mathbf{u}/N_{\text{sub}}$, and $\mathbf{f}$ is further divided into N_{sub} sub-processes f. Algorithm 1 presents a simplified simulation process f for a single substep.

Algorithm 1: Simplified Simulation Process f

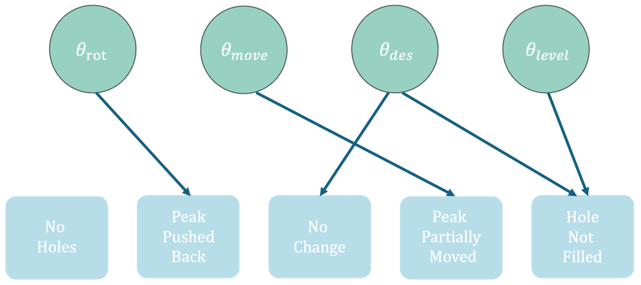
```
1  Grid Reset and Initialisation
2  $F_{\text{tmp}} = (I + \Delta t_{\text{sub}} C) F$
3  $U, S, V = f_{\text{svd}}(F_{\text{tmp}})$
4  $F', \sigma = f_{\text{con}}(U, S, V, \kappa)$
5  $\mathbf{v}_{\text{grid}} = f_{\text{p2g}}(\mathbf{p}, \mathbf{v}, \sigma, C, \Delta t_{\text{sub}})$
6  $\mathbf{s}'_{\text{agent}} = f_{\text{move}}(\mathbf{s}_{\text{agent}}, \mathbf{u}^{\text{sub}}, \Delta t_{\text{sub}})$
7  $\mathbf{v}'_{\text{grid}} = \mathbf{v}_{\text{grid}} + \Delta t_{\text{sub}} G$
8  $\mathbf{v}'_{\text{grid}} = f_{\text{col}}(\mathbf{s}'_{\text{agent}}, \mathbf{s}_{\text{con}}, \mathbf{v}'_{\text{grid}}, \Delta t_{\text{sub}})$
9  $\mathbf{v}', C' = f_{\text{g2p}}(\mathbf{v}'_{\text{grid}}, \Delta t_{\text{sub}})$
10 $\mathbf{v}' = f_{\text{col}}(\mathbf{s}'_{\text{agent}}, \mathbf{s}_{\text{con}}, \mathbf{p}, \mathbf{v}', \Delta t_{\text{sub}})$
11 $\mathbf{p}' = \mathbf{p} + \Delta t_{\text{sub}} \mathbf{v}'$
```

Here, $F \in \mathbb{R}^{3\times3\times N}, C \in \mathbb{R}^{3\times3\times N}, \sigma \in \mathbb{R}^{3\times3\times N}, I \in \mathbb{R}^{3\times3\times N}$, and $G \in \mathbb{R}^{3\times N}$ denote the deformation gradient, affine matrix, Cauchy stress, identity matrix, and gravity matrix for all particles in $\mathbf{x}$. U, S, V are the singular value decomposition (SVD) results of F for each particle. κ represents material parameters. $\mathbf{p}$ and $\mathbf{v}$ are the position and velocity components in $\mathbf{x}$. $\mathbf{s}_{\text{agent}}$ and $\mathbf{s}_{\text{con}}$ are the state of the robot and static container. $f_{\text{svd}}, f_{\text{con}}, f_{\text{p2g}}, f_{\text{move}}, f_{\text{col}}$, and f_{g2p} represent the SVD, elastoplastic processing, particle-to-grid, robot motion, collision handling, and grid-to-particle in f, respectively. The symbol $'$ indicates the next sub-step. For more details on the MLS-MPM procedure, refer to [4].

C. Causal Effects and Phase Specification



(a) Causal effects of skill parameters and features in the excavation task, where the numbers are *ACE* values, and the bold arrows denote strong causal effects.



(b) Causal effects of skill parameters and different phases in the levelling task.

Fig. 6: The causal effect in our tasks.

TABLE I: Phase Determination in Levelling Tasks. τ_a: Threshold for hole area. $\lambda_{hai}, \lambda_{hsi}, \lambda_{psi}, \lambda_{pei}$: Features of the initial material state.

Conditions	Phases
$\lambda_{h_a} < \tau_a$	No Holes
$\lambda_{psi} > \lambda_{ps}$	Peak Pushed Back
$\lambda_{psi} \approx \lambda_{ps}$ and $\lambda_{pei} \approx \lambda_{pe}$	No Change
$\lambda_{psi} \approx \lambda_{ps}$ and $\lambda_{pei} > \lambda_{pe}$	Peak Partially Moved
$\lambda_{hsi} > \lambda_{hs}$	Hole Not Filled

D. Experiment Setup

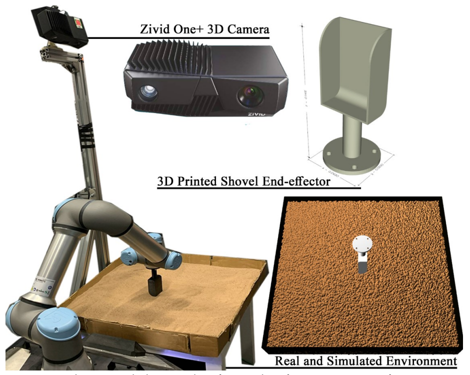

Zivid One+ 3D Camera
3D Printed Shovel End-effector
Real and Simulated Environment

Fig. 7: Real-world and simulation experiment setup.