Differentiable Skill Optimisation for Powder Manipulation in Laboratory Automation

Minglun Wei[1], Xintong Yang[1], Yu-Kun Lai[2] and Ze Ji[1,†]

***Abstract*— Robotic automation is accelerating scientific discovery by reducing manual effort in laboratory workflows. However, precise manipulation of powders remains challenging, particularly in tasks such as transport that demand accuracy and stability. We propose a trajectory optimisation framework for powder transport in laboratory settings, which integrates differentiable physics simulation for accurate modelling of granular dynamics, low-dimensional skill-space parameterisation to reduce optimisation complexity, and a curriculum-based strategy that progressively refines task competence over long horizons. This formulation enables end-to-end optimisation of contact-rich robot trajectories while maintaining stability and convergence efficiency. Experimental results demonstrate that the proposed method achieves superior task success rates and stability compared to the reinforcement learning baseline.**

I. INTRODUCTION

Robotic automation is accelerating scientific discovery by streamlining workflows in areas such as photocatalysis [1] and synthetic chemistry [2]. These systems reduce manual workload and allow scientists to focus on higher-level reasoning and design [3]. However, while macroscale processes have seen rapid automation, challenges persist at the microscale, particularly in the precise handling of powders. Powders play a critical role in pharmaceuticals and materials science, where precise transport is essential for reproducibility and system stability. Despite some efforts in weighing [4], [3] and grinding [5], [6], powder transport is often treated as secondary, and its reliable optimisation remains underexplored.

This research gap is not incidental. The dynamic behavior of powders during motion and interaction is highly nonlinear and sensitive to environmental variability, making it difficult to address with conventional learning or control techniques. Recent advances in differentiable physics and high-performance parallel computation [7], [8] present a promising direction for addressing this challenge. By optimising within high-fidelity, real-world-consistent differentiable simulation environments, it becomes possible to obtain precise and reliable powder transport trajectories in a safe manner. While prior studies have applied such methods to the manipulation of materials like elastoplastic solids [9], [10], [11] and fluids [12], [13], no existing work has systematically addressed the problem of powder transport in laboratory settings using skill or trajectory optimisation.

[†]Corresponding author: Ze Ji. JiZ1@cardiff.ac.uk

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

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

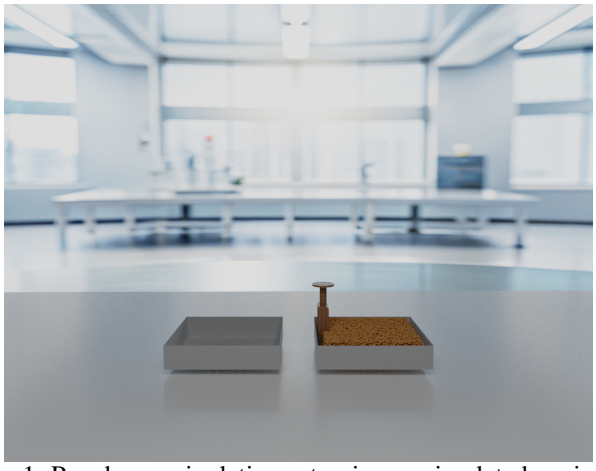

Fig. 1: Powder manipulation setup in our simulated environment. Our renderer is built based on LuisaRender [14].

To address this gap, we propose a trajectory optimisation framework for powder transport in a laboratory setting. Specifically, we build a differentiable physics simulator using Taichi to model powder dynamics and define differentiable skill parameters mapped to control inputs. To tackle the challenges of long-horizon manipulation, we further adopt a curriculum optimisation strategy that first focuses on scooping-related parameters before optimising the full parameter set. These parameters are optimised via gradient backpropagation through a task-specific loss. Comparative experiments with standard reinforcement learning (RL) methods demonstrate the effectiveness and efficiency of our approach.

II. RELATED WORK

A. Powder Manipulation in Lab Automation

Powder manipulation plays a critical role in lab automation, particularly in pharmaceuticals and materials science, where precise transport is key to maintaining accuracy and material integrity. Prior work has explored powder-related tasks such as grinding [5], [6], weighing [4], [3], scooping [15], scraping [16], and dispensing [17]. Notably, [4] proposed a sim-to-real RL framework for weighing, while [3] improved simulation fidelity using flowability-aware Bayesian inference. Although [15] tackles scooping, its focus is on hand design rather than control strategy. To date, no existing work systematically addresses trajectory-level optimisation for powder transport in lab settings.

B. Differentiable Simulation-based Manipulation

Differentiable simulation enables gradient-based reasoning over physical processes, making it a compelling tool for learning control policies. Early examples like DeLaN [18], [19] applied this to rigid-body systems, though they lack support for complex material interaction. Recent frameworks such as Taichi [8] have enabled scalable differentiable simulation across domains like deformable solids [9], [10], [11], fluids [12], [13], granular materials [20], and thin-shell structures [21]. Among these, [20] guided RL with differentiable trajectory optimisation in granular media. However, most methods optimise low-level actions per timestep, which can suffer from instability. Our approach instead introduces skill-level parameterisation to improve stability and reduce dimensionality.

III. METHOD

A. Problem Formulation

This study focuses on optimising powder transport trajectories in laboratory settings. We aim to find an optimal set of skill parameters Θ that generate control sequences of horizon T for the dynamic system. We formulate a time-discretised trajectory optimisation problem with state trajectory $\mathcal{S} = (\mathbf{s}_0, ..., \mathbf{s}_T)$, observations $\mathcal{O} = (\mathbf{o}_0, ..., \mathbf{o}_T)$, and control inputs $\mathcal{U} = (\mathbf{u}_0, ..., \mathbf{u}_{T-1})$. Starting from an initial state $\mathbf{s}_0$, the optimisation is defined as:

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

$$\text{s.t. } \mathbf{s}_{i+1} = \mathbf{f}(\mathbf{s}_i, \mathbf{u}_i) \tag{2}$$

$$\mathbf{u}_i = \mathbf{g}(\Theta)[i] \quad \forall i = 0, \ldots, T-1 \tag{3}$$

$$\Theta = \begin{cases} \Theta^{\text{init}} & j = 0 \\ \Theta^{-} - \alpha \cdot \nabla_{\Theta^-} \mathcal{L}(\mathcal{X}^-, \Theta^-) & j > 0 \end{cases} \tag{4}$$

Here, $\mathcal{L}(\cdot)$ measures the distance between the final observation $\mathbf{o}_T$ and the target position $\mathbf{p}^{\text{target}}$, $\mathbf{f}(\cdot)$ represents system dynamics, and $\mathbf{g}(\cdot)$ maps skill parameters to control inputs. The parameter update follows a gradient descent rule with learning rate α, using the gradient of the loss evaluated on the previous trajectory and parameters.

B. Task Specification

We consider a powder transport task in a laboratory setting, where the objective is to move as much powder as possible from a source container to an initially empty target container. The simulated environment replicates the real-world configuration: one container is filled with powder, while the other is empty. The system state $\mathbf{s}$ contains full particle-level information (e.g., positions, velocities, accelerations), while the robot has access only to partial observations $\mathbf{o}$, consisting of particle positions.

The robot action is represented as a 6-dimensional Cartesian displacement $\mathbf{u} \in \mathbb{R}^6$, encompassing translation and rotation. These control inputs are generated from a low-dimensional skill parameter set Θ via a differentiable mapping $\mathbf{g}(\cdot)$. To guide the optimisation, we define a task-level loss function $\mathcal{L}$ based on spatial proximity, rather than matching a fixed target configuration. Specifically, the loss is computed as the sum of absolute distances between the final observed particle positions $\mathbf{o}_T$ and a designated goal position $\mathbf{p}^{\text{target}}$:

$$\mathcal{L} = \sum_{\mathbf{p}_j \in \mathbf{o}_T} |\mathbf{p}_j - \mathbf{p}^{\text{target}}| \tag{5}$$

where j indexes the observed particles. This formulation not only ensures efficient powder delivery to the target while accommodating variations in the final distribution.

C. Skill Parameters

To reduce the complexity of long-horizon 6D control, we introduce a compact representation of robot behaviour through a set of bounded skill parameters $\Theta \in [\theta^{\min}, \theta^{\max}]$. These parameters, derived from human demonstrations, define temporally abstracted behaviours and are mapped to control sequences $\mathcal{U}$ over a time horizon T via a differentiable function $\mathbf{g}(\cdot)$.

In our implementation, five parameters govern key aspects of the motion: scooping depth θ_{d}, scooping angle θ_{s}, post-scoop lifting height θ_{l}, transport displacement θ_{t}, and pouring angle θ_{p}. As shown in Fig. 2, these parameters collectively define the spatiotemporal structure of the robot's trajectory.

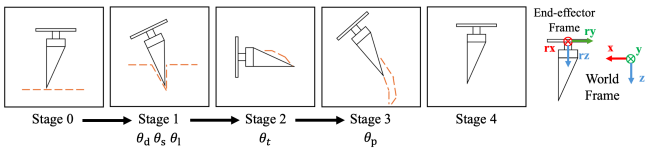

Fig. 2: Transport Task Process Visualisation.

Given fixed translational and rotational velocities, the number of steps for each control segment is computed by dividing the desired displacement in each control dimension by its corresponding velocity and rounding the result. These per-step increments are then distributed across time to form the complete control sequence $\mathcal{U}$.

D. Differentiable Simulation

We simulate granular dynamics using the Moving Least Squares Material Point Method (MLS-MPM) [22], combined with St. Venant-Kirchhoff elasticity and the Drucker-Prager plasticity model [23]. Each global time step Δt is divided into N_{sub} substeps of size $\Delta t_{\text{sub}} = \Delta t / N_{\text{sub}}$, with control input scaled as $\mathbf{u}^{\text{sub}} = \mathbf{u}/N_{\text{sub}}$. The global simulation update $\mathbf{f}$ is decomposed into N_{sub} sub-processes f, where each substep is computed as Algorithm 1.

Here, F, C, σ, $\mathbf{g}$, and I denote the per-particle deformation gradient, affine velocity field, Cauchy stress, gravity, and identity matrix, respectively. Each F is decomposed as $F = USV^\top$ via singular value decomposition. Material properties are included in κ. $\mathbf{x}$ and $\mathbf{v}$ are particle positions and velocities, while $\mathbf{s}_{\text{agent}}$ and $\mathbf{s}_{\text{con}}$ denote the robot and container states. Sub-functions include f_{svd} (SVD), f_{con} (constitutive model), f_{p2g} and f_{g2p} (particle-grid transfers), f_{move} (robot actuation), and f_{col} (collision handling). Superscript $'$ indicates updated values at the next substep.

Algorithm 1: Simplified Sub-step Procedure f

Grid Reset and Initialisation
$F_{\text{tmp}} = (I + \Delta t_{\text{sub}} C)F$
$U, S, V = f_{\text{svd}}(F_{\text{tmp}})$
$F', \sigma = f_{\text{con}}(U, S, V, \kappa)$
$\mathbf{v}_{\text{grid}} = f_{\text{p2g}}(\mathbf{x}, \mathbf{v}, \sigma, C, \Delta t_{\text{sub}})$
$\mathbf{s}'_{\text{agent}} = f_{\text{move}}(\mathbf{s}_{\text{agent}}, \mathbf{u}^{\text{sub}}, \Delta t_{\text{sub}})$
$\mathbf{v}'_{\text{grid}} = \mathbf{v}_{\text{grid}} + \Delta t_{\text{sub}} \mathbf{g}$
$\mathbf{v}'_{\text{grid}} = f_{\text{col}}(\mathbf{s}'_{\text{agent}}, \mathbf{s}_{\text{con}}, \mathbf{v}'_{\text{grid}}, \Delta t_{\text{sub}})$
$\mathbf{v}', C' = f_{\text{g2p}}(\mathbf{v}'_{\text{grid}}, \Delta t_{\text{sub}})$
$\mathbf{v}' = f_{\text{col}}(\mathbf{s}'_{\text{agent}}, \mathbf{s}_{\text{con}}, \mathbf{x}, \mathbf{v}', \Delta t_{\text{sub}})$
$\mathbf{x}' = \mathbf{x} + \Delta t_{\text{sub}} \mathbf{v}'$

Our framework enables end-to-end optimisation of skill parameters using differentiable physics. Gradients of the task loss are propagated through the entire pipeline, including perception, simulation, and control. All components are differentiable, and gradient flow is carefully maintained across non-smooth operations (e.g., rounding) to ensure end-to-end differentiability.

E. Curriculum Optimisation

We formulate powder transport as a long-horizon manipulation task. Prior approaches often divide such tasks into sequential stages such as scooping, transporting, and dumping, with each trained separately and executed in a chained manner [20]. In contrast, our framework optimises the entire process in an end-to-end fashion, which significantly reduces training overhead. However, both the quantity of material scooped and the accuracy of its deposition are critical to overall task success. In particular, the quality of the initial scooping action directly affects the feasibility of the subsequent transport and pouring stages. To address this, we adopt a curriculum-based optimisation strategy. Specifically, training initially focuses on the first three skill parameters associated with scooping (θ_d, θ_s, θ_l), and gradually expands to include the full parameter set. This progressive approach encourages the agent to acquire reliable foundational behaviours before learning full-sequence coordination, resulting in improved convergence and final performance.

IV. EXPERIMENTS AND RESULTS

We compare our trajectory optimisation method with a representative RL baseline, Soft Actor-Critic (SAC) [24], on the powder transport task in a simulated laboratory environment. The simulation replicates real-world conditions, including robot, container, and tool models. Physical parameters are calibrated using our prior DPSI framework [9] to ensure sim-to-real consistency. The simulator runs at $\Delta t = 0.01$ s with 20 substeps per step.

Skill parameters are optimised within $[-1, 1]$ using RMSprop [25] with $\beta_r = 0.9$ and a learning rate of 0.05, for 30 epochs. In the first 15 epochs, only scoop-related parameters are optimised, and the full parameter set is optimised thereafter. SAC is trained for the same number of episodes, with $\gamma = 0.99$, batch size 8, and actor/critic learning rates of 0.001. The task loss (Eq. 5) is negated and used as the reward for SAC to ensure consistent objectives. For evaluation, we report an indicator metric defined as the difference between the target value and the number of particles transported.

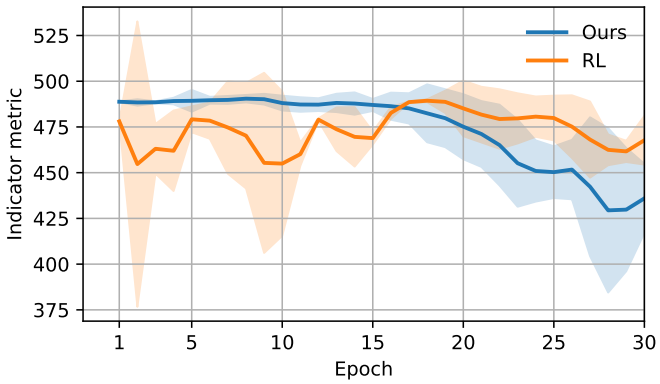

Fig. 3: Indicator metric curves for the powder transport task.

To evaluate the performance of our method, we conducted experiments under three different random seeds. The indicator metric curves for our approach and the RL baseline are presented in Fig. 3. In the initial stage of training (the first 15 epochs), the indicator metric for our method remains relatively flat. This behaviour is expected, as the full set of parameters is not yet being optimised at this stage. Instead, the robot is primarily acquiring the prerequisite skills required for powder scooping. Once this foundational skill is sufficiently acquired, the optimisation process shifts toward the entire skill set, leading to a more noticeable and stable decrease in indicator metric. Compared to the RL baseline, our method demonstrates a more consistent and eventually better indicator metric trajectory, indicating better convergence and overall performance on the powder transport task.

V. CONCLUSIONS

This work presents a trajectory optimisation framework for powder transport in laboratory settings. By constructing a differentiable physics simulator, designing differentiable skill mappings, and incorporating a curriculum optimisation strategy, our method addresses the challenges of long-horizon, contact-rich powder manipulation. Experimental results demonstrate that our approach achieves higher task success rates and improved stability compared to the representative RL baseline, underscoring its effectiveness for precision powder handling in laboratory automation scenarios.

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 and the UK Biotechnology and Biological Sciences Research Council (BBSRC) grant No. BB/Y008537/1.

References

[1] B. Burger, P. M. Maffettone, V. V. Gusev, C. M. Aitchison, Y. Bai, X. Wang, X. Li, B. M. Alston, B. Li, R. Clowes *et al.*, "A mobile robotic chemist," *Nature*, vol. 583, no. 7815, pp. 237–241, 2020.

[2] T. Dai, S. Vijayakrishnan, F. T. Szczypiński, J.-F. Ayme, E. Simaei, T. Fellowes, R. Clowes, L. Kotopanov, C. E. Shields, Z. Zhou *et al.*, "Autonomous mobile robots for exploratory synthetic chemistry," *Nature*, vol. 635, no. 8040, pp. 890–897, 2024.

[3] N. Radulov, A. Wright, A. I. Cooper, G. Pizzuto *et al.*, "Flip: Flowability-informed powder weighing," *arXiv preprint arXiv:2506.03896*, 2025.

[4] Y. Kadokawa, M. Hamaya, and K. Tanaka, "Learning robotic powder weighing from simulation for laboratory automation," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 2932–2939.

[5] Y. Nakajima, M. Hamaya, Y. Suzuki, T. Hawai, F. von Drigalski, K. Tanaka, Y. Ushiku, and K. Ono, "Robotic powder grinding with a soft jig for laboratory automation in material science," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2320–2326.

[6] Y. Nakajima, M. Hamaya, K. Tanaka, T. Hawai, F. von Drigalski, Y. Takeichi, Y. Ushiku, and K. Ono, "Robotic powder grinding with audio-visual feedback for laboratory automation in materials science," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 8283–8290.

[7] Y. Hu, T.-M. Li, L. Anderson, J. Ragan-Kelley, and F. Durand, "Taichi: a language for high-performance computation on spatially sparse data structures," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 6, pp. 1–16, 2019.

[8] Y. Hu, L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand, "Difftaichi: Differentiable programming for physical simulation," *arXiv preprint arXiv:1910.00935*, 2019.

[9] 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.

[10] S. Chen, Y. Liu, S. W. Yao, J. Li, T. Fan, and J. Pan, "Diffsrl: Learning dynamical state representation for deformable object manipulation with differentiable simulation," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 9533–9540, 2022.

[11] Z. Huang, Y. Hu, T. Du, S. Zhou, H. Su, J. B. Tenenbaum, and C. Gan, "Plasticinelab: A soft-body manipulation benchmark with differentiable physics," *arXiv preprint arXiv:2104.03311*, 2021.

[12] Z. Xian, B. Zhu, Z. Xu, H.-Y. Tung, A. Torralba, K. Fragkiadaki, and C. Gan, "Fluidlab: A differentiable environment for benchmarking complex fluid manipulation," *arXiv preprint arXiv:2303.02346*, 2023.

[13] Z. Li, Q. Xu, X. Ye, B. Ren, and L. Liu, "Difffr: Differentiable sph-based fluid-rigid coupling for rigid body control," *ACM Transactions on Graphics (TOG)*, vol. 42, no. 6, pp. 1–17, 2023.

[14] S. Zheng, Z. Zhou, X. Chen, D. Yan, C. Zhang, Y. Geng, Y. Gu, and K. Xu, "Luisarender: A high-performance rendering framework with layered and unified interfaces on stream architectures," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 6, pp. 1–19, 2022.

[15] T. Takahashi, C. C. Beltran-Hernandez, Y. Kuroda, K. Tanaka, M. Hamaya, and Y. Ushiku, "Scu-hand: Soft conical universal robotic hand for scooping granular media from containers of various sizes," *arXiv preprint arXiv:2505.04162*, 2025.

[16] G. Pizzuto, H. Wang, H. Fakhruldeen, B. Peng, K. S. Luck, and A. I. Cooper, "Accelerating laboratory automation through robot skill learning for sample scraping," in *2024 IEEE 20th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2024, pp. 2103–2110.

[17] Y. Jiang, H. Fakhruldeen, G. Pizzuto, L. Longley, A. He, T. Dai, R. Clowes, N. Rankin, and A. I. Cooper, "Autonomous biomimetic solid dispensing using a dual-arm robotic manipulator," *Digital Discovery*, vol. 2, no. 6, pp. 1733–1744, 2023.

[18] M. Lutter, C. Ritter, and J. Peters, "Deep lagrangian networks: Using physics as model prior for deep learning," in *International Conference on Learning Representations (ICLR)*, 2019.

[19] M. Lutter and J. Peters, "Combining physics and deep learning to learn continuous-time dynamics models," *The International Journal of Robotics Research*, vol. 42, no. 3, pp. 83–107, 2023.

[20] M. Wei, X. Yang, Y.-K. Lai, S. A. Tafrishi, and Z. Ji, "Automachef: A physics-informed demonstration-guided learning framework for granular material manipulation," *arXiv preprint arXiv:2406.09178*, 2024.

[21] Y. Wang, J. Zheng, Z. Chen, Z. Xian, G. Zhang, C. Liu, and C. Gan, "Thin-shell object manipulations with differentiable physics simulations," *arXiv preprint arXiv:2404.00451*, 2024.

[22] Y. Hu, Y. Fang, Z. Ge, Z. Qu, Y. Zhu, A. Pradhana, and C. Jiang, "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.

[23] G. Klár, T. Gast, A. Pradhana, C. Fu, C. Schroeder, C. Jiang, and J. Teran, "Drucker-prager elastoplasticity for sand animation," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–12, 2016.

[24] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870.

[25] 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.