2026 IEEE International Conference on Robotics and Automation (ICRA 2026)
June 1-5, 2026. Vienna, Austria

FLIP: Flowability-Informed Powder Weighing

Nikola Radulov[1], Alex Wright[1], Thomas Little[1], Andrew I. Cooper[2] and Gabriella Pizzuto[1,2]

Abstract— **Autonomous manipulation of powders remains a significant challenge for robotic automation in scientific laboratories. The inherent variability and complex physical interactions of powders in flow, coupled with variability in laboratory conditions necessitates adaptive automation. This work introduces *FLIP*, a flowability-informed powder weighing framework designed to enhance robotic policy learning for granular material handling. Our key contribution lies in using material flowability, quantified by the angle of repose, to optimise physics-based simulations through Bayesian inference. This yields material-specific simulation environments capable of generating accurate training data, which reflects diverse powder behaviours, for training 'robot chemists'. Building on this, *FLIP* integrates quantified flowability into a curriculum learning strategy, fostering efficient acquisition of robust robotic policies by gradually introducing more challenging, less flowable powders. We validate the efficacy of our method on a robotic powder weighing task under real-world laboratory conditions. Experimental results show that *FLIP* with a curriculum strategy achieves a low dispensing error of $2.12 \pm 1.53\,\mathrm{mg}$, outperforming methods that do not leverage flowability data, such as domain randomisation ($6.11 \pm 3.92\,\mathrm{mg}$). These results demonstrate *FLIP*'s improved ability to generalise to previously unseen, more cohesive powders and to new target masses.**

I. INTRODUCTION

Robotic systems are increasingly used in laboratories to relieve human researchers of tedious and repetitive tasks, allowing them to focus on higher-level scientific reasoning. These intelligent systems have already achieved success in different chemistry domains [1], such as photocatalysis [2] and solid-state powder x-ray diffraction [3]. While effective and reliable for sample transportation, robotic methods have yet to demonstrate robust and adaptive scalability for more complex tasks in this domain. Consequently, more dexterous lab tasks still require the skill and domain knowledge of human researchers.

Developing robotic chemists that are capable of handling different materials with a wide range of properties, such as granular and powdery materials, would allow us to bring autonomous materials discovery to a new level. For example, robots could carry out end-to-end chemical synthesis, from sample preparation and reaction execution to transfer of samples for characterisation. Despite their central role in both materials chemistry and organic synthesis, powders remain difficult to manipulate reliably with robotic systems. This difficulty stems from the complex, non-linear dynamics of powders that are highly sensitive to material properties in real-world conditions. For example, many powders are hygroscopic, meaning that their properties can change significantly under humid conditions. A fundamental task that underpins chemistry and materials science is powder weighing. The central challenge in automating this task is that properties of a given sample can range from highly cohesive and abrasive powders, to dry, free-flowing powders that behave more predictably. Existing powder weighing systems [4], while highly effective in specific domains, often struggle to generalise across the diverse material behaviours present in materials research. Approaches that have used general purpose robots to address this include using dual-arm manipulators with fuzzy logic control [4], or employing a single manipulator guided by a deep reinforcement learning algorithm [5]. While the method by Jiang et al. [4] used a rule-based system with binned particle size input, neither method addressed how material properties can be embedded into the learning framework to generalise on new materials.

Our work proposes a method to enhance the reliability of robotic powder weighing within automated experimental workflows by explicitly leveraging knowledge of material properties, specifically powder flowability. This property defines how readily a powder flows under gravity or applied stress [6], influencing its handling characteristics (*e.g.,* during dispensing and dosing). First, we construct a framework to reduce the simulation-to-reality gap for powder manipulation. This involves using automated Angle of Repose (AoR) measurements from real powders to optimise key simulation parameters (*e.g.*, friction, cohesion, adhesion). The resulting flowability-informed simulation environment provides a training dataset representative of various powders, enabling the training of a Reinforcement Learning (RL) policy for precise powder dispensing. We explore curriculum learning [7], increasing material cohesiveness as training progresses. Finally, we zero-shot transfer this flowability-informed policy to a physical robot and evaluate its performance on real powders.

In summary, the contributions of this work are:

1) The design and implementation of ***FLIP***, a novel framework integrating material property measurement, flowability-informed granular materials simulation, and policy learning for improving robotic powder weighing in laboratory automation.
2) A robotic system for accurate AoR measurement, validated against manual methods.
3) A simulation calibration framework that leverages measured material flowability data to improve granular material dynamics within robotic simulators.
4) Empirical validation demonstrating an improved robotic policy performance and generalisation for the challenging task of powder weighing, validated through simulation and real-world experiments.

[1] School of Computer Science and Informatics, University of Liverpool, UK.
[2] Department of Chemistry, University of Liverpool, UK.

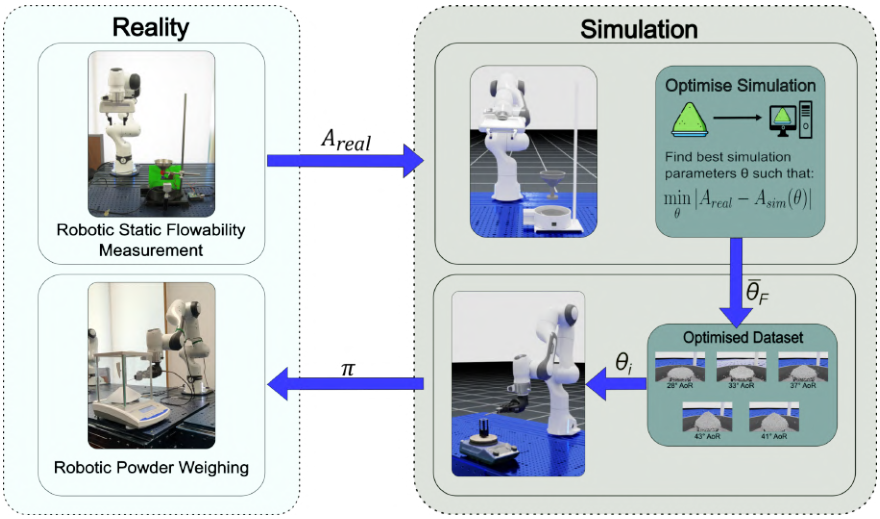


Fig. 1: The ***FLIP*** framework leverages Bayesian optimisation to calibrate simulation parameters using real-world powder flowability data. This flowability-informed powder simulation then serves as a high-fidelity environment for training the robot control policy for powder weighing.

II. Related Work

A. Robotics for Laboratory Automation

The rapid progress in robotics for a multitude of everyday tasks has led to increased interest in the adoption of robotic platforms for chemistry laboratory environments [1]. While these efforts can automate routine tasks, 'robotic chemists' still lack the dexterity and skill of human chemists. Recent works have focused on introducing sensory feedback for glassware manipulation [8] and microplate insertion [9]. Beyond glassware manipulation, human chemists can handle heterogeneous materials with varying physical properties. Some works have looked at material manipulation in 'robot chemists' across different lab tasks, for example, sample scraping [10], grinding [11] and powder weighing [4], [5]. Here, the authors focused on automating the tasks using either data-driven (model-free, off-policy deep reinforcement learning) or rule-based methods. While some approaches may have used physical properties as inputs, they did not embed them directly within the learning framework itself. Here, we concentrate our research endeavours on adding material properties directly into the learning paradigm towards improving robotic manipulation in laboratory environments.

B. Sim-to-Real Gap in Robotic Manipulation of Materials

Simulators are vital for developing learning-based 'robotic chemists' and generating cost-effective training data. However, their effectiveness in chemistry lab automation hinges on accurately modelling real-world interactions, such as contact-rich manipulation and particle-based simulation, since the inherent differences between simulated and real-world physics (sim-to-real) pose a major hurdle for robotics. A common strategy to mitigate this is domain randomisation [12], where simulation parameters or visuals are varied to improve policy robustness. For instance, Kadokawa et al. [5] used a domain-randomised simulator to train an RL agent for powder weighing, achieving sub-milligram accuracy but lacking explicit material modelling for broader generalisation across powders with different properties. Sim-to-real transfer methods have also enabled policy deployment in other high-precision tasks such as deformable object manipulation [13]. While domain randomisation can improve robustness, achieving accurate simulation of complex physical materials is also critical for reliable skill transfer, particularly for tasks involving sensitive interactions with materials such as powders. Recent works calibrate simulation parameters using real-world observations to improve physics fidelity. Notably, Matl et al. [14] inferred granular properties (*e.g.*, friction, restitution) from visual data of formations for tasks like pouring, and Lopez-Guevara et al. [15] inferred liquid properties through interaction-based probing for pouring optimisation. In contrast to prior sim-to-real methods primarily relying on DR, our approach explicitly uses real-world material flowability data to inform simulation parameters. This enables the creation of material physics-informed environments, which directly embed knowledge into the training loop to enhance robotic powder manipulation performance and generalisation.

III. Methodology

Our method improves powder weighing by leveraging material-specific knowledge, specifically powder flowability. To address the sim-to-real challenge, we develop a framework (Section III-A) that optimises simulation parameters (*e.g.,* adhesion, cohesion, friction, etc.) using real-world flowability data, thereby creating realistic training environments for robots to learn how to handle diverse powders. This process yields the realistic training environments that form the core of ***FLIP*** (Section III-B), our powder weighing framework. We define the task of powder weighing as follows: the robot is equipped with a spatula in its end effector, an analytical balance is used for weight feedback, and the target powder is collected in a container (typically

using lab glassware).

A. Closing the Sim-to-Real Gap For Powders Using Flowability Data

1) Problem Formulation: Let the dynamics of the simulation engine be described by a function p parameterised by θ, which gives the probability distribution over the state s_t at time t, given the state s_{t-1} and action a_{t-1} at time $t-1$. Let A be a task, or a set of tasks, that can give a quantifiable measure of the physical property to optimise. Let A_{real} be the measure obtained from executing task(s) A in the real world and $A_{sim}(\theta)$ be the corresponding measure obtained by executing A in the simulation environment. The reality gap, or error E, with respect to the physical property under optimisation is defined as: $E = |A_{real} - A_{sim}(\theta)|$. The objective is to find the set of simulation parameters θ such that E is minimised:

$$\min_{\theta} |A_{real} - A_{sim}(\theta)| \tag{1}$$

To optimise the simulation framework with respect to our physical property, powder flowability, the task set A corresponds to the measurement of static flowability [16]. This property is quantified by the AoR, which inversely correlates with static flowability (*i.e.*, a higher AoR results in a lower flowability). This is measured experimentally using a powder tester following the procedure outlined in the ISO 8398:1989 standard (Fig. 2). The process involves loading approximately 50 g of powder into a funnel, allowing it to form a pile on a fixed base and using vibration to assist flow through a sieve. This is followed by measuring the height h of the resulting pile. The AoR is then calculated from the measured height and the known base diameter (d_{base}) using Equation 2.

$$AoR = \frac{2h}{d_{base}} \tag{2}$$

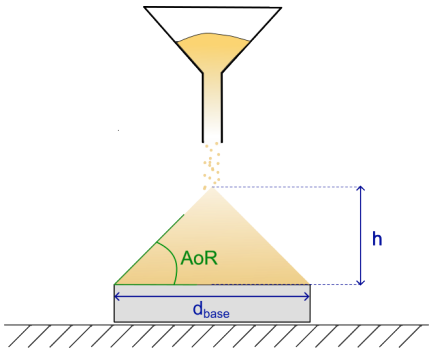


Fig. 2: Measurement of static powder flowability using the AoR method. The figure depicts the experimental setup, the formation of the powder pile after flowing through a funnel, and indicates the height measurement used for AoR calculation.

Let the simulation parameters θ for simulator p be such that $A_{sim}(\theta)$ is the simulated task measurement for static powder flowability in our case. A_{real} represents the desired measurement derived from the actual system ($\mathcal{M}_{real}$). When optimising a material, calculating A_{real} involves a singular physical expenditure: our work indicates that conducting 5 experimental trials is adequate to establish a consistent average AoR. To account for the complex, non-linear sensitivities of the simulator, the objective function (Eq. 1) maps multiple parameters to a single AoR. This intentional redundancy mirrors physical reality, where disparate material properties often yield identical flowabilities. To navigate this multi-dimensional solution space, we employ a sequential Bayesian Optimization (BO) strategy [17], as implemented in [18]. It uses a Gaussian process with a Matérn kernel as the proxy model. The search was guided by an Upper Confidence Bound function. As illustrated in Algorithm 1 the process begins by initialising ($\mathcal{I}_B$) the search bounds for θ based on simulator constraints (Table I). Let $\mathcal{H} = \{(\theta_i, E_i)\}$ be the history of evaluated parameters and their errors, where $i \in [0, |\mathcal{H}|]$. At each iteration k, $\mathcal{O}(\theta, E)$ is invoked to register previous evaluation results with the optimiser and propose a new set of parameters θ_{k+1} ($\mathcal{O}$). E_{k+1} is evaluated through $A_{sim}(\theta_{k+1})$, and $\mathcal{H}$ is updated. Each simulation evaluation on our system (IV-A) averaged 2 minutes and 30 seconds. If progress stagnates after $T_{stagnation}$ iterations, we adaptively refine the search bounds, $\mathcal{B}_h$. The new bounds are defined by the minimum and maximum values of each parameter found within the top 10% best-performing parameter sets in $\mathcal{H}$. The process continues until termination conditions $\mathcal{E}_S$ are met, aiming to collect N acceptable parameter sets in Θ_{opt}.

Algorithm 1 Sim-to-Real Optimisation Using Powder Flowability

Require: $N \geq 1$ (Number of solutions desired)
$\Theta_{opt} \leftarrow \emptyset$
$\mathcal{H} \leftarrow \emptyset$
$A_{real} \leftarrow \mathcal{M}_{real}()$
$k \leftarrow 0$
$\mathcal{I}_B()$
$\theta \leftarrow \emptyset$
$E \leftarrow \infty$
while $N > 0$ **do**
  $\theta \leftarrow \mathcal{O}(\theta, E)$
  $A_{sim} \leftarrow p(\theta)$
  $E \leftarrow |A_{real} - A_{sim}|$
  $\mathcal{H} \leftarrow \mathcal{H} \cup \{(\theta, E)\}$
  if $E < T_{AoR}$ **then**
    $\Theta_{opt} \leftarrow \Theta_{opt} \cup \{\theta\}$
    $N \leftarrow N - 1$
  else if $k \geq T_{stagnation}$ **then**
    $\mathcal{B}_h(\mathcal{H})$
    if $\mathcal{E}_S()$ **then**
      return Θ_{opt}
    end if
    $k \leftarrow 0$
  end if
  $k \leftarrow k + 1$
end while
return Θ_{opt}

2) Simulation Assumptions: Due to the complexity of particle physics, certain assumptions and simplifications were

TABLE I: Physical parameter bounds for optimisation of powder simulation in NVIDIA Isaac Sim.

Physical Parameter	Minimum	Maximum
Particle Diameter [mm]	0.22	0.32
Adhesion [·]	0.0	1.3
Particle Adhesion Scale [·]	0.0	1.3
Adhesion Offset Scale [·]	0.0	0.00005
Friction [·]	0.0	1.3
Gravity Scale [·]	0.3	1.1
Damping [·]	0.0	1.0
Cohesion [·]	0.0	1.2
Particle Mass [μg]	13	16.5

adopted in our simulation. First, particles are assumed to be perfectly spherical. While real powders exhibit complex geometries, this simplification accelerates collision detection computations, making the simulation practical for efficient training of robotic manipulation policies. Second, we assumed that all particles within a given simulation instance possess identical, uniform size and mass, thus neglecting the inherent polydispersity found in most real-world powders. The particle diameter is constrained to lie within a range that is computationally tractable. Consequently, the permissible particle mass range is determined based on the particle diameter range and the densities of the specific materials being simulated. Third, particle dynamics are simulated using the Position-Based Dynamics (PBD) solver [19]. Although this framework simulates the dynamics of both inter-particle and external forces applied to the system, it satisfies constraints such as attachments and contacts using a position-level approach that prioritises simulation speed at the expense of physical fidelity. The physical parameters governing these interactions, listed in Table I, are bounded by physically plausible real-world values and the inherent limits of the simulator (NVIDIA Isaac Sim [20]). Given the simplifying assumptions employed, the optimal values identified for these parameters may not directly correspond to their real-world physical counterparts. Our objective is not to estimate exact micro-dynamic coefficients but rather to calibrate the simulation parameters to accurately reproduce the macroscopic powder behaviour.

B. Flowability-Informed Powder Weighing

1) Problem Formulation: We use the notion of a Markov Decision Process (MDP) [21] to formulate the powder weighing task as a reinforcement learning problem. The MDP is represented as a tuple $\mathcal{M} = \langle \mathcal{X}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma \rangle$ of a set of states ($\mathcal{X}$), actions ($\mathcal{A}$), transition probabilities ($\mathcal{P}$), and immediate rewards ($\mathcal{R}$), with γ being a scalar discount factor in $[0, 1]$, that determines the emphasis given to immediate rewards relative to future ones. We use parametrised shake and incline robot motions, similar to the method in [5], to improve sample efficiency and accelerate policy convergence. Each episode consists of 10 sequential steps; at each step, the agent executes first the inclination motion and then the shake action, before it receives an observation and an immediate reward based on the weighing error. We define the observation o as the triplet $o = (w_{current}, w_{target}, \theta_{spoon})$, w_{target} represents the target amount of powder, $w_{current}$ the quantity of powder that has been dumped so far, θ_{spoon} the current pitch of the spoon. The action $\mathcal{A}$ space is continuous and two-dimensional, $a = (a_{shake}, a_{incline}) \in \mathcal{A}$. Here, a_{shake} specifies the amplitude of the shaking motion (backward displacement before returning to the initial position) and $a_{incline}$ dictates the target inclination angle (pitch) for the spoon tilt. The reward function is computed from the error between the target weight w_{target} and the dispensed weight $w_{current}$ calculated as follows:

$$\mathcal{R} = \begin{cases} \Delta + 1, & \Delta > -1 \\ \Delta, & otherwise \end{cases}, \tag{3}$$

where $\Delta = -|w_{target} - w_{current}|$.

2) Flowability-Optimised Data: The static and dynamic flow characteristics of a powder impact the precision achievable during robotic weighing. Materials with high flowability allow more controlled dispensing, whereas cohesive powders are prone to adhesion, clumping, and erratic bulk discharge, which increases the task complexity. Since flowability is an emergent property—that is, different combinations of micro-parameters like friction, cohesion, and particle geometry can yield similar flow behaviour—it serves as a crucial indicator of manipulation difficulty. Therefore, we structure our agent training around this macroscopic property. Instead of broadly randomising individual simulation particle parameters, we focus the training distribution by sampling particle parameters conditioned on achieving a diverse range of target flowability values. This flowability-conditioned training enhances data efficiency by grouping physically distinct powders that present similar control challenges. We hypothesise that policies trained under this regime will generalise effectively to novel powders based on their flowability, rather than requiring precise knowledge of the underlying micro-parameters.

To this aim, let $F_r : [AoR_{min}, AoR_{max}]$ be the representative flowability range over which we wish to train our policy. We assume that for each flowability level F sampled from this range, there exists a physics dynamics model represented by the state transition function $p_F(s_t|s_{t-1}, a_{t-1})$. Within a parametrised simulation, this function becomes $p(s_t|s_{t-1}, at-1, \theta_F)$, for which the parameter set θ_F can be derived by the method in Section III-A.1. Constructing a flowability-informed dataset then simply involves finding a set:

$$\overline{\theta}_{\mathcal{F}} = \{\theta_{F_1}, \theta_{F_2}, \theta_{F_3}, ..., \theta_{F_{N-1}}, \theta_{F_N}\} \tag{4}$$

where $\mathcal{F} = \{F_1, F_2, F_3, ..., F_N\}$ represents the sampled flowability levels from our range R, which could be obtained through random uniform sampling or curriculum sampling (Section III-B.3).

3) Flowability-Informed Curriculum Learning: We employ a curriculum learning strategy [7] to accelerate policy training within simulation environments configured using flowability-optimised parameters (derived via Algorithm 1). The curriculum is structured based on powder flowability,

ordered from highest flowability (corresponding to the lowest AoR) to lowest flowability (highest AoR). Formally, this corresponds to starting from simulation parameters $\theta \in \Theta^*_{\mathcal{F}}$ associated with the highest flowability regime within our target set $\mathcal{F}$. The curriculum then progressively transitions to parameter sets representing lower flowability levels, thereby systematically increasing the task difficulty for the learning agent. This easy-to-hard progression is motivated by the empirical observation that low-flowability powders, often exhibiting significant cohesion and adhesion, present greater challenges for precise robotic manipulation and mass control during the weighing task.

As detailed in Algorithm 2, this curriculum is implemented by first indexing the optimised parameter sets $\Theta^*_{\mathcal{F}}$ by their associated flowability metric (*e.g.*, AoR). At each episode, the training environment is configured with a randomly selected parameter set (θ), representing a powder of a specific flowability level. After each episode, the error (e_i) ,between the target amount and the amount dumped, is evaluated. The curriculum advances to the subsequent level (lower flowability) only when the policy achieves a predetermined performance criterion, assessed by the error over the last K episodes, at the present level (line 22). To mitigate potential stagnation on a particularly challenging flowability class, a timeout mechanism is incorporated: the curriculum advances to the subsequent parameter set if the performance criteria are not met within a predefined maximum of M episodes for the current level, regardless of the achieved error (line 27).

Algorithm 2 Flowability-Informed Powder Weighing ***(FLIP)***

Require: Flowability indexed parameter sets $\Theta^*_{\mathcal{F}} = \{\theta^{(0)}, \ldots, \theta^{(L-1)}\}$; Max episodes N_{max}, Max steps T_{max}; Error thresholds T_{mean}, T_{max_err}; Error queue size K, Max episodes per level M

1: $\pi \leftarrow \pi_0$
2: $\mathcal{D} \leftarrow \emptyset$
3: $\mathcal{Q}_E \leftarrow$ FIFOQueue(K)
4: $level \leftarrow 0$
5: $m \leftarrow 0$
6: **for** $i \leftarrow 1$ to N_{max} **do**
7: $\quad\theta \leftarrow$ SampleConfig$(\Theta^*_{\mathcal{F}}, level)$
8: $\quad$SetConfig(θ)
9: $\quad s \leftarrow$ EnvReset()
10: $\quad$**for** $t \leftarrow 1$ to T_{max} **do**
11: $\quad\quad a \leftarrow \pi(s)$
12: $\quad\quad s', r, d \leftarrow$ EnvStep(a)
13: $\quad\quad \mathcal{D} \leftarrow \mathcal{D} \cup \{(s, a, r, s', d)\}$
14: $\quad\quad \pi \leftarrow$ UpdatePolicy$(\mathcal{D})$
15: $\quad\quad s \leftarrow s'$
16: $\quad$**end for**
17: $\quad e_i \leftarrow$ GetEpisodeError(s)
18: $\quad\mathcal{Q}_E$.push(e_i)
19: $\quad m \leftarrow m + 1$
20: $\quad$**if** $|\mathcal{Q}_E| = K$ **then**
21: $\quad\quad$**if** Mean$(\mathcal{Q}_E) < T_{mean}$ **or** Max$(\mathcal{Q}_E) < T_{max_err}$ **then**
22: $\quad\quad\quad level \leftarrow \min(level + 1, L - 1)$
23: $\quad\quad\quad\mathcal{Q}_E$.clear()
24: $\quad\quad\quad m \leftarrow 0$
25: $\quad\quad$**end if**
26: $\quad$**else if** $m \geq M$ **then**
27: $\quad\quad level \leftarrow \min(level + 1, L - 1)$
28: $\quad\quad\mathcal{Q}_E$.clear()
29: $\quad\quad m \leftarrow 0$
30: $\quad$**end if**
31: **end for**
32: **return** π

IV. Experimental Evaluation

We evaluated ***FLIP*** both in simulation and in real world experiments designed to answer the following questions: (1) Can an automated system accurately collect powder flowability data comparable to manual methods? (2) Can real-world flowability data be integrated into simulation to model diverse powders accurately? (3) Does our ***FLIP*** method improve robot policy learning for autonomous powder weighing in simulation and reality? Our initial experiments validated simulation fidelity by comparing simulated powder dynamics to empirical flowability data, essential for successful sim-to-real transfer. Subsequent experiments compared ***FLIP***-trained robot policies (using different data ordering strategies) against those trained without flowability information evaluating weighing accuracy.

A. Experimental Setup

Our simulation environments were developed in MatteriX [22] using IsaacLab [23] and are outlined below (Section IV-A.1 and IV-A.2). These were run on a system featuring a 13th Gen Intel(R) Core(TM) i7-13700F processor, 64GB of RAM and an NVIDIA GeForce RTX 4090 GPU, operating on Ubuntu 20.04. A Franka Research 3 (FR3) robotic manipulator was used in all simulation and real-world setups.

1) Autonomous Robotic Flowability Measurement: We automated the static powder flowability measurement for reliable execution in both real-world and simulation environments (Fig. 1). The robot grasps and inverts the powder platform, pouring the contents into a closed stainless steel funnel with a $14\,\mathrm{mm}$ exit diameter. It then centres the platform beneath the funnel. Powder release is controlled by an Arduino Uno-actuated servo-motor, which opens a slide gate at the funnel's bottom. An RGB-D camera (*Intel RealSense D405*) is used to measure the height of the pile that forms on the platform. The RGB image is processed to locate the funnel's tip (p_{tip}) and the powder pile's apex (p_{apex}) (Fig. 2), and their 3D coordinates are extracted using the corresponding depth data. The height of the pile (h) is computed by subtracting the measured distance between these two points from the known distance between the funnel and the platform (D) as detailed in Equation 5.

$$h = D - \|\vec{p}_{\text{apex}} - \vec{p}_{\text{tip}}\| \tag{5}$$

We use Equation 2 to compute the AoR, where the diameter of the pile is the diameter of its base. In the simulated

environment the height of the pile can directly be computed using the 3D coordinates of each independent particle.

2) Robotic Powder Weighing: As illustrated in Fig. 1, our experimental setup incorporates a FR3 robot arm equipped with a Robotiq 85F Gripper. The gripper is mounted parallel to the working surface to manoeuvre the dispensing spoon more easily. The spoon operates above a 20ml vial, which serves as the target container. To gauge the powder amount in the vial, a precision scale (Fisherbrand FPRS223) is used, feeding the measured weight directly into the ***FLIP*** framework. While the real-world setup is largely replicated in simulation, known simulator constraints with the vial's intricate CAD mesh led to collision issues. Consequently, the model was simplified to a comparable-sized cuboid for improved collision detection and computational efficiency.

B. Powder Flowability Optimisation Results

To evaluate the performance of our robotic system for automatic AoR measurement (Section IV-A.1), a comparative study was conducted against manual measurements performed according to the ISO 8398:1989 standard. We tested a range of powders with AoRs (*i.e.,* different flowabilities) spanning $27°$ to $43°$. For each powder sample, five manual measurements and five automated system measurements were recorded. The differences between the measured AoR values obtained by each method are reported in Table II. The largest discrepancy between manual and automated measurements was observed for semolina, with an absolute error of $1.8°$. Across all powders, the average absolute error was $0.84°$, demonstrating the reliability and repeatability of our automated measurement process.

TABLE II: Comparison of manual and automated AoR static powder flowability measurements.

Material	**Automated AoR (°)**	**Manual AoR (°)**
SiO_2	28.55 ± 1.94	27.69 ± 2.54
Sugar	31.41 ± 1.42	30.93 ± 0.87
$NaCl$	36.03 ± 2.29	35.8 ± 1.71
Semolina	42.762 ± 1.97	40.91 ± 1.85
$NaHCO_3$	42.38 ± 2.48	43.14 ± 1.0

These automated AoR values are then used as target flowability levels in a simulation parameter optimisation process. For each powder, we perform a BO search to identify simulation parameters that accurately replicate the real-life AoR. The objective is to find configurations yielding a simulated AoR within a tolerance of $T_{\text{AoR}} = 1.5°$ of the measured value. To initiate the optimiser, we employ 5 points selected at random and determine that the number of configurations to discover is $N = 15$ (refer to Algorithm 1). The process is capped at 200 evaluations, beyond which optimisation is halted, and we assign $T_{stagnation} = 75$. The leading 10 parameter configurations that adhere to the AoR constraint are preserved.

For each powder, Table III presents visual comparisons of the simulated and real pile formations, and includes the mean and standard deviation of the absolute AoR error derived from the 10 highest-ranked solutions.

C. Simulation Powder Weighing Results

We evaluate our ***FLIP*** method using three different orderings of flowability-optimised data: random ordering, curriculum ordering (*i.e.,* most to least flowable) and reverse curriculum ordering (*i.e.,* least to most flowable). These are compared against a baseline method employing domain randomisation across all physical parameters, comparable to previous works [5]. We define the training flowability range as $F_r \in [28°, 37°]$ and sample three flowability levels: $28°$, $33°$, and $37°$, corresponding to SiO_2, sugar, and $NaCl$. This range purposely excludes highly cohesive, flour-like materials such as semolina and $NaHCO_3$. We use a model-free, off-policy DRL algorithm, Soft Actor-Critic (SAC) [24] to train our agent. We adopted the same neural network architecture and implementation, including a training termination criterion of 4000 episodes, as in related works [5] for a fair comparison. For curriculum ordering, training for each flowability level had a cut-off of 1330 episodes. A transition to the next level occurred earlier if the mean final error over the last 20 episodes fell below $0.8\,\text{mg}$ or if the maximum final error fell below $1\,\text{mg}$. We treated these thresholds as hyperparameters and manually tuned them for the task. Other hyperparameters included the number of optimised data points per flowability level (set to 7, a value found to balance learning quality and stability), and the choice of flowability levels themselves (based on real-world data availability). Powder weighing error (in mg) is the primary evaluation metric, averaged over 5 independent runs with different random seeds.

Our results are presented in Fig. 3. ***FLIP*** with random and curriculum data ordering methods are the only ones achieving close to $1\,\text{mg}$ accuracy, with the curriculum-based method converging faster. It is worth noting that the curriculum agent reaches mg-level accuracy after approximately 600 episodes at the initial flowability level. The total reward curve (Fig. 3a) shows a dip when transitioning from $28°$ to $33°$, although the final error remains below $1.5\,\text{mg}$. This likely results from increased cohesion at higher flowability levels, which reduces powder displacement per action, requiring more steps to dispense the desired amount. The fact that the reverse curriculum method performed worse than domain randomization strengthens our hypothesis that ordering materials based on their flowability levels is crucial. By starting with the most cohesive powders, which require strong, cumulative actions to dispense, the agent faces a significant delayed reward problem and develops an overly aggressive policy. This leads to negative transfer when the curriculum switches to more flowable powders (step 2660, Fig. 3): the agent overshoots the target, causing the absolute error to increase even as the total reward rises. This outcome confirms that a progressive learning strategy is essential for developing robust policies. Additionally, the domain randomisation method shows high instability (large standard deviation in final powder weighing errors).

TABLE III: Real-world data and the resulting optimised powder simulation. The sim-to-real error measured for the five different powder samples demonstrates the fidelity achieved by parameter optimisation.

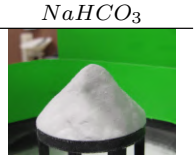
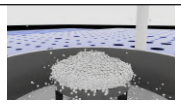
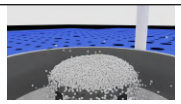
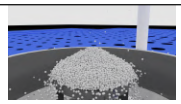
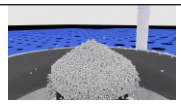
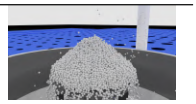
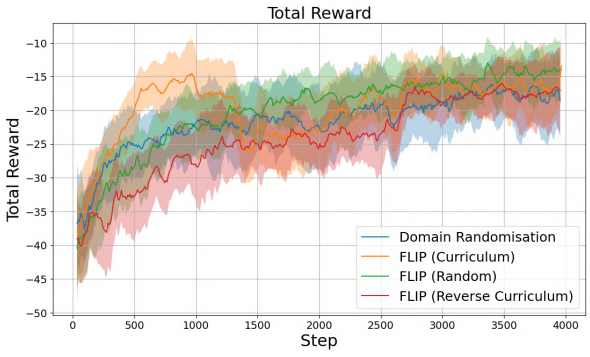
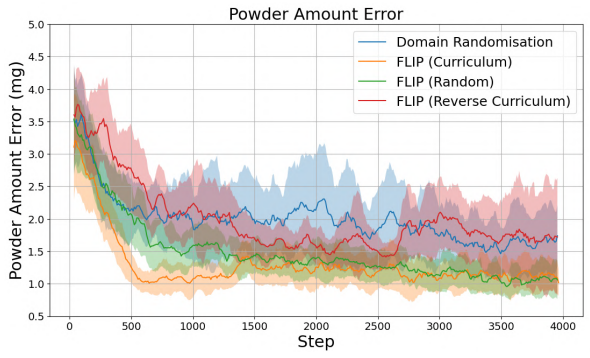
	SiO_2	**Sugar**	$NaCl$	**Semolina**	$NaHCO_3$
Real powder pile					
Simulated powder pile					
Sim-to-real error (°)	0.25 ± 0.13	0.22 ± 0.12	1.09 ± 0.15	0.40 ± 0.14	1.37 ± 0.43

TABLE IV: Real-world performance comparison for the powder weighing task. For each powder sample, results per method represent the mean absolute error and the corresponding standard deviation, calculated over 5 runs using the best-performing policy. Sodium bicarbonate and semolina represent out-of-distribution materials; the policy was not trained on these specific powders or materials with similar flow characteristics. The 20 mg target weight was also untrained.

Method	**Powder Weighing Error (mg)**									
	$NaCl$		Sugar		SiO_2		$NaHCO_3$		Semolina	
	15 mg	20 mg	15 mg	20 mg	15 mg	20 mg	15 mg	20 mg	15 mg	20 mg
Domain Randomisation	3.2 ± 1.93	16.5 ± 5.1	2.83 ± 1.3	4.0 ± 3.0	2.33 ± 1.4	1.83 ± 0.6	7.8 ± 4.83	7.0 ± 6.89	6.4 ± 4.17	9.2 ± 4.79
FLIP (Random)	2.6 ± 2.33	4.8 ± 3.05	2.0 ± 1.09	2.66 ± 1.24	$\mathbf{1.33 \pm 0.47}$	$\mathbf{1.81 \pm 1.83}$	9.4 ± 6.52	4.8 ± 3.31	4.0 ± 1.78	6.0 ± 3.79
FLIP (Curriculum)	$\mathbf{2.4 \pm 1.86}$	$\mathbf{2.4 \pm 1.74}$	$\mathbf{1.6 \pm 0.48}$	$\mathbf{1.2 \pm 0.97}$	1.6 ± 0.8	2.2 ± 1.32	$\mathbf{3.6 \pm 2.3}$	$\mathbf{1.6 \pm 1.35}$	$\mathbf{1.8 \pm 0.97}$	$\mathbf{2.8 \pm 2.31}$
FLIP (Reverse Curriculum)	13.6 ± 7.9	11.0 ± 8.87	11.4 ± 5.35	1.8 ± 1.16	3.8 ± 2.22	5.8 ± 3.13	12.2 ± 3.34	8.6 ± 6.71	9.5 ± 5.52	11.2 ± 7.65



(a)



(b)

Fig. 3: Total reward (a) and final powder weighing error (b), averaged over 5 random seeds.

D. Real-world Powder Weighing Results

We conduct sim-to-real experiments to evaluate the performance of our method ***FLIP*** in the context of an autonomous powder weighing task. For each evaluation, we select the policy corresponding to the best-performing run in simulation, where performance is measured by the lowest final error over the evaluation window. The trained network is transferred zero-shot to the real robotic system, with the robot speed and shake amplitude calibrated to match those used in the simulator. Each selected policy is tested across a set of five powders (two of which have out-of-distribution flowability levels) with varying flowabilities (see Table II) and for two target weights: 15 mg and 20 mg; the latter target weight was not seen previously during training. For each configuration, we repeat the task five times and report the mean and standard deviation of the absolute error. Results are presented in Table IV, where the best-performing policy for each condition is highlighted. Our method, ***FLIP***, outperforms both the baseline method and the ablation using reverse curriculum policy across all tested materials. The curriculum ***FLIP*** policy achieves the lowest average powder weighing error across all trials (2.12 ± 1.53 mg), followed by the random ***FLIP*** policy (3.94 ± 2.55), the DR policy without flowability data (6.11 ± 3.92) and finally, the reverse curriculum ***FLIP*** policy (8.19 ± 4.88). The curriculum-based policy demonstrates strong generalisation to previously unseen, more cohesive materials (out-of-distribution flowability levels). For example, $NaHCO_3$, with an AoR of 42°, lies outside the training range of 28° to 37°, yet is handled effectively. This suggests that simulating microscopic behaviour like particle shape is not essential for developing effective manipulation strategies. Instead, a macro-scale approach avoids the computational burden of micro-level modelling by calibrating simplified physics to a representative behavioural envelope. By using flowability as an umbrella property to capture this behavioural range,

the model generalises to novel materials without requiring parameter re-identification.

V. Conclusions

This paper presented a new approach to robotic powder weighing in autonomous laboratories through the first successful application of flowability-informed reinforcement learning. Our core contribution is ***FLIP***, a method that strategically integrates powder flowability information into both simulation design and curriculum learning. This is supported by a successful simulation-to-reality transfer pipeline that uses real-world flowability metrics, including automated AoR measurements, to create high-fidelity simulation training environments for improved robotic manipulation of powders. Both simulation and real-world experiments demonstrated that policies trained with ***FLIP*** outperform baselines lacking flowability information, achieving low powder weighing errors and generalisation to more cohesive, unseen materials and target weights. Future work will focus on novel adaptations to the controller to better accommodate cohesive powders ($AoR > 45$), where current displacements are insufficient. Furthermore, we plan on enhancing the policy's observation space by incorporating a real-time estimate of the material remaining in the tool, such that the policy could adapt its actions based on this. Ultimately, extending this approach to complementary tasks like powder scooping tasks could establish a comprehensive and robust end-to-end autonomous powder weighing pipeline but also fundamentally transform how novel, uncharacterised materials are handled in chemical research labs. More generally, this approach undoubtedly has analogous uses in other lab automation domains where approximate physical models are available.

Acknowledgements

This work was supported by the Leverhulme Trust through the Leverhulme Research Centre for Functional Materials Design, the Royal Academy of Engineering under the Research Fellowship Scheme and EPSRC through the AI for chemistry: AIchemy hub (EPSRC grant EP/Y028775/1 and EP/Y028759/1). A.I.C. thanks the Royal Society for a Research Professorship (RSRP\S2\232003). All authors would like to thank Dr. Ben Alston and Dr. Charlotte Boott for their input on the materials chemistry aspect of the project.

References

[1] G. Tom, S. P. Schmid, S. G. Baird, Y. Cao, K. Darvish, H. Hao, S. Lo, S. Pablo-García, E. M. Rajaonson, M. Skreta, and et al., "Self-driving laboratories for chemistry and materials science," *Chemical Reviews*, 2024.

[2] B. Burger, P. M. Maffettone, V. V. Gusev, C. M. Aitchison, Y. Bai, X. yan Wang, X. Li, B. M. Alston, B. Li, R. Clowes, N. Rankin, B. Harris, R. S. Sprick, and A. I. Cooper, "A mobile robotic chemist," *Nature*, vol. 583, pp. 237 – 241, 2020.

[3] A. M. Lunt, H. Fakhruldeen, G. Pizzuto, L. Longley, A. White, N. Rankin, R. Clowes, B. Alston, L. Gigli, G. M. Day, A. I. Cooper, and S. Y. Chong, "Modular, multi-robot integration of laboratories: an autonomous workflow for solid-state chemistry," *Chem. Sci.*, vol. 15, pp. 2456–2463, 2024.

[4] 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, pp. 1733–1744, 2023.

[5] 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)*, 2023, pp. 2932–2939.

[6] J. Prescott and R. Barnum, "On powder flowability," *Pharmaceutical Technology*, vol. 24, pp. 60–84+236, 01 2000.

[7] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: Association for Computing Machinery, 2009, p. 41–48.

[8] A. Butterworth, G. Pizzuto, L. Pecyna, A. I. Cooper, and S. Luo, "Leveraging multi-modal sensing for robotic insertion tasks in r&d laboratories," in *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, 2023, pp. 1–8.

[9] V. Scamarcio, J. Tan, F. Stellacci, and J. Hughes, "Reliable and robust robotic handling of microplates via computer vision and touch feedback," *Frontiers in Robotics and AI*, vol. 11, 2025.

[10] 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)*, 2024, pp. 2103–2110.

[11] 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," 10 2023.

[12] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," 2017.

[13] P. M. Scheikl, E. Tagliabue, B. Gyenes, M. Wagner, D. Dall'Alba, P. Fiorini, and F. Mathis-Ullrich, "Sim-to-real transfer for visual reinforcement learning of deformable object manipulation for robot-assisted surgery," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 560–567, 2023.

[14] C. Matl, Y. S. Narang, R. Bajcsy, F. Ramos, and D. Fox, "Inferring the material properties of granular media for robotic tasks," *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 2770–2777, 2020.

[15] T. Lopez-Guevara, R. Pucci, N. K. Taylor, M. U. Gutmann, S. Ramamoorthy, and K. Suhr, "Stir to pour: Efficient calibration of liquid properties for pouring actions," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 5351–5357.

[16] W. Liu, Z. Deng, Y. Zhang, X. Zhu, J. Huang, H. Zhang, and J. Zhu, "Decoding powder flowability: Machine learning pioneers the analysis of particle-size distribution effects," *Powder Technology*, vol. 435, p. 119407, 2024.

[17] J. Snoek, H. Larochelle, and R. P. Adams, "Practical bayesian optimization of machine learning algorithms," in *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'12. Red Hook, NY, USA: Curran Associates Inc., 2012, p. 2951–2959.

[18] F. Nogueira, "Bayesian Optimization: Open source constrained global optimization tool for Python," 2014–.

[19] M. Müller, B. Heidelberger, M. Hennix, and J. Ratcliff, "Position based dynamics," *J. Vis. Comun. Image Represent.*, vol. 18, no. 2, p. 109–118, Apr. 2007.

[20] NVIDIA, "NVIDIA Isaac Sim," NVIDIA, 2025.

[21] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: A Bradford Book, 2018.

[22] K. Darvish, A. Sohal, A. Mandal, H. Fakhruldeen, N. Radulov, Z. Zhou, S. Veeramani, J. Choi, S. Han, B. Zhang, J. Chae, A. Wright, Y. Wang, H. Darvish, Y. Zhao, G. Tom, H. Hao, M. Bogdanovic, G. Pizzuto, A. I. Cooper, A. A. Guzik, F. Shkurti, and A. Garg, "Matterix: toward a digital twin for robotics-assisted chemistry laboratory automation," *Nature Computational Science*, vol. 6, pp. 67 – 82, 2025.

[23] M. Mittal, C. Yu, Q. Yu, J. Liu, N. Rudin, D. Hoeller, J. L. Yuan, P. P. Tehrani, R. Singh, Y. Guo, H. Mazhar, A. Mandlekar, B. Babich, G. State, M. Hutter, and A. Garg, "Orbit: A unified simulation framework for interactive robot learning environments," 2023.

[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 *ICML*, 2018.