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

Gotta Scoop 'Em All: Sim-and-Real Co-Training of Graph-based Neural Dynamics for Long-Horizon Scooping

Kaiwen Hong[1], Haonan Chen[1,2*], Jiaming Xu[1*], Runxuan Wang[1*], Kaylan Wang[1], Mingtong Zhang[1], Shuijing Liu[4], Yifan Zhu[5], Yunzhu Li[3], Katherine Driggs-Campbell[1]

***Abstract*— We present a system to address a new problem of removing all granular objects from a container using a scoop, a task that requires long-horizon reasoning over the complex interactions between granular objects, the scoop, and the container. To tackle this challenge, we adopt a graph-based neural dynamics (GBND) model trained through sim-and-real co-training that leverages both rich synthetic data and a small amount of real-world data. Using the learned model, we plan long-horizon scooping sequences of behavior primitives with Monte Carlo Tree Search (MCTS). Extensive experimental evaluations demonstrate that our system with the co-training strategy is capable of removing all materials in a container with fewer than 20 scoops, significantly outperforming strategies that use only simulation or real data. Our approach can also generalize in a zero-shot manner to new materials, and quickly adapt to new containers with few additional data points. More videos and analyses are available on our project website https://scoopthemall.github.io/.**

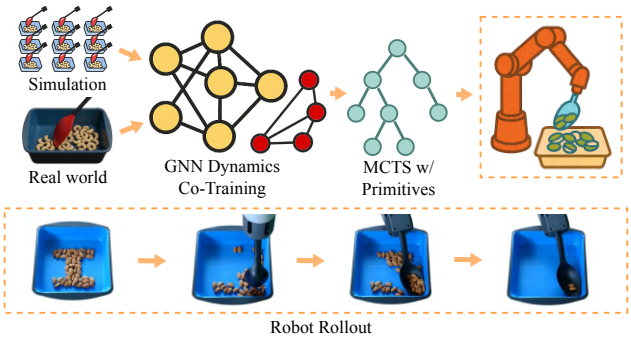

Fig. 1: Overview of our SAGO task and our approach. We do a co-training of graph-based neural dynamics model using simulation and real-world data and use an MCTS planner to decide primitives and execute robot actions.

I. INTRODUCTION

Robotic manipulation of granular objects is fundamental for various applications, spanning healthcare [1]–[3], industrial assembly [4], construction [5], collaborative robotics [6], and scientific exploration [7]. In this work, we study the problem of scooping *all* granular objects in a container (SAGO). While scooping granular terrains and in open space can often succeed with simple heuristic motions [5], [7]–[9], SAGO goes beyond simple heuristics and requires strategic, long-horizon planning. The robot must carefully coordinate its actions to ensure full coverage and recover from errors, because single-scoop mistakes can compound, leaving objects in unrecoverable states and causing task failure. In addition, SAGO is different from tasks such as scooping and excavation in open spaces [5], [8]–[10] and planar manipulation of granular materials [11] that mainly involve interactions between the scoop and granular particles. Instead, it requires the robot to also reason about the interactions among the scoop, granular particles, and the container.

To address these challenges, we adopt a model-based planning approach that employs graph-based neural dynamics (GBND) to learn the system dynamics [11], [12], which implements a graph neural network (GNN) for dynamics learning and has shown good performance on learning the dynamics of granular media. Different from prior works [11], [12], where a small amount of real-world data is sufficient for their tasks, using only real-world data performs poorly on SAGO. Instead, we develop a co-training strategy that combines a small amount of real-world data with large-scale, low-cost simulation data, which we refer to as sim-and-real co-training. Although simulation exhibits a notable sim-to-real gap for granular objects, we find that this strategy significantly improves the accuracy of the learned dynamics model. In particular, simulation provides privileged access to full particle states, while real-world data offers grounding in real-world sensor noise and dynamics. Together, these complementary sources enable accurate and transferable dynamics learning. Our experiments demonstrate that the dynamics models co-trained with simulation and real data significantly outperform those trained solely on simulation or real-world data. We also find that the dynamics models trained with real-world data plateau in their performance as the amount of data increases, significantly underperforming those co-trained. This highlights that simulation data is not only helpful, but also **necessary** for accurately modeling our task. Once the model is learned, we employ Monte Carlo tree search (MCTS) with an action primitive library to plan long-horizon actions that efficiently remove all materials.

In summary, our contributions are threefold: 1) We introduce a challenging multi-step task SAGO, that requires scooping all granular objects from a storage container; 2) We present a scooping system that leverages co-training with simulation and real data for learning a GNN model and a MCTS-based planner for completing the task at high success rates; 3) We perform extensive experiments that demonstrate the effectiveness of our approach across various types of

* Equal contribution.
Corresponding author: kaiwen2@illinois.edu
[1]University of Illinois Urbana-Champaign, [2]Harvard University, [3]Columbia University, [4]The University of Texas at Austin, [5]University of Illinois Chicago

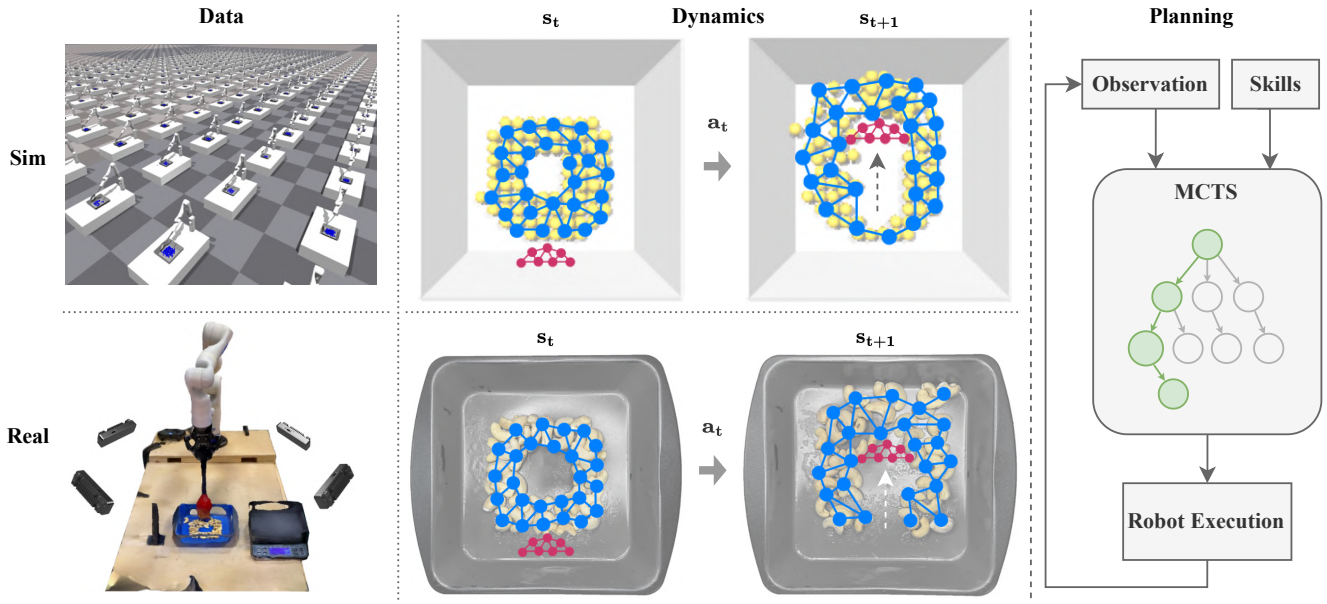

Fig. 2: Overview of our framework. *(Left and Middle)* The first row shows parallel simulation environments and simulation data are collected; the second row shows the real-world environment and some cashews are swept by a spoon. *(Right)* The proposed planning pipeline, where RGB-D observations are passed to the MCTS planner to select the type and parameters of a skill, and then re-plan based on new observation.

granular objects and containers. Our sim-and-real co-training strategy is the only approach that is capable of scooping all materials out of the container.

II. Related Works

In this section, we review related works in granular material manipulation tasks. Then we review closely related works in the technical approaches for leveraging sim-and-real co-training in robotics, and long-horizon planners for non-prehensile manipulation tasks.

A. Granular Material Manipulation

Many granular material manipulation tasks have been explored, such as scooping and excavation from open spaces [7], [13], grasping [14], and pushing [12], [15]. The most closely related works study pushing and shape-forming of granular materials on planar surfaces [12], [15], which can be seen as special cases of our more general setting. Our task requires reasoning about interactions with containers of varying shapes and planning over long horizons, introducing additional complexity beyond planar pushing. While these works also leverage GNNs to train the dynamics model, they train the models only on real-world data, whereas we demonstrate that co-training on simulation data is not only helpful but also necessary to achieve good manipulation performance on this task. In addition, we leverage MCTS to plan long manipulation sequences with complex behavior primitives.

B. Sim-and-Real Co-Training in Robotics

While real-world data is expensive to collect, various works have demonstrated that mixing model training with simulation and real data boosts the performance of the trained model in robotic manipulation tasks [16]–[19]. We similarly adopt this idea for training GNNs to model the interaction dynamics of granular objects with rigid tools and environments. As we will detail in Sec. IV-A, one interesting finding is that in this task, models only trained with increasing amounts of real-world data would start to plateau in their performance, significantly underperforming those co-trained with sim and real data. This is likely due to the fact that simulation data provides privileged state information of all the particles in the scene and boosts the accuracy of the trained model.

C. Long-horizon Planning for Non-Prehensile Manipulation Tasks

Long-horizon manipulation is a longstanding challenge in robotics [20], [21]. Integrated task and motion planning (ITMP) combines discrete task planning with continuous motion planning, enabling robots to holistically consider actions and their long-term consequences [22]. Other prior works explored table-top non-prehensile rearrangement tasks using a planar push [23]–[26] or MCTS [27] algorithms. MCTS can search in high-dimensional state spaces with an adaptive sampling strategy. However, these works assume negligible sim-to-real gap and thus employ physics-based simulators such as box2d [28] and pybullet [29] to predict the effect of a robot action. In this work, we adopt an MCTS framework because of its ability to search in high-dimensional spaces efficiently, which is well suited for long-horizon tasks.

III. Problem Statement

We formulate the task of scooping all granular materials out from a container using a spoon as a sequential decision-making problem. In each episode, the robot observes a

colored point cloud of the scene with semantic labels $l \in \{\text{container}, \text{granule}, \text{scoop}\}$ for each point, uses a policy to apply an action a from a discrete set of parametrized behavior primitives $\mathcal{A}$, and receives a binary reward of whether the container is empty. Each skill a consists of a sequence of atomic actions $(u_1, u_2, \cdots, u_t), u_i \in \mathbb{R}^{d_{\text{control}}}$, generated from semantically meaningful parameters $\theta_a \in \mathbb{R}^{d_a}$ where d_a denotes the number of parameters for skill a. When presented with a task T, characterized by the container type, and the granular object type, the goal for the robot is to empty the container in as few actions as possible.

As shown in Fig. 2, the scooping system consists of a robot arm equipped with a spoon at the end-effector, along with 4 calibrated RGB-D cameras that generate merged point clouds of the entire scene. The task is to transfer all granular materials from one container to the other. We assume that the container in the scene and the robot scoop are known. The raw point cloud of the scene is first segmented to remove all points except for the container, the granular materials, and the scoop, given the knowledge of the scoop and container poses obtained from robot proprioception and the estimation of container pose with colored ICP [30]. All points above the container bottom are treated as points belonging to the granular particles. Since only the surface of the granular particles is observed in the raw point cloud, we fill the space between the surface point cloud and the container bottom with additional points. Finally, we downsample the point cloud with furthest point sampling [31] with a varying maximum number to be 125 for the computational efficiency of GNNs and assign semantic labels to all points in the scene. We denote the positions of the downsampled point cloud as X.

IV. Methodology

Our method adopts a GNN to model the interactions between the particles, the container, and the spoon in contact. Compared to black-box neural networks, GNNs take advantage of the local interactions between granular particles to achieve better generalizations [11]. The dynamics model is co-trained on both rich simulation data and a small amount of real-world data. During task execution, we adopt an MCTS planner to plan a long action sequence in each episode to remove as many particles as possible, until all particles are scooped out or the time budget is exceeded. In the rest of this section, we first present the dynamics model and the training process, and then detail the planner along with the behavior primitives.

A. Learned Dynamics Model

Inspired by [11], [32], we model the scene as a GNN and learn the dynamics of how X evolves given a robot action. The state of the GNN at time t is formed by $(\mathcal{O}_t, \mathcal{E}_t)$, which consists of a set of GNN vertices and edges. Each vertex $o_{i,t} = \langle x_{i,t}, c_{i,t} \rangle$ consists of the 3D positional information $x_{i,t}$ and vertex attribute $c_{i,t}$ that is the semantic label of the vertex. To represent robot control commands, for vertices belonging to the scoop, we additionally augment the vertex attribute $c_{i,t}$ with the *intended motion* $v_{i,t}$, encoding how the robot is expected to move at that point in time. The graph vertices $\mathcal{O}_t$ are set to be equal to X_t. Each edge $e_{k,t} = (u_k, v_k, c_k)$ denotes the relations between the vertices and can represent both interactions between granular particles, and those with the scoop and the container. Here k is the edge index, u_k and v_k denote the receiver and sender vertex index, and c_k denotes the relationship type of the two vertices (e.g., between granular particles). The edges are formed between the vertices based on a predefined distance metric and dynamically evolve over time as the vertices move, hence the time subscript t on the edges.

To learn the dynamics of the entire scene given a robot action with a GNN $X_{t+1} = f(X_t, u_t)$, the vertices and edges are assigned with latent features $h^O_{i,t}$ and $h^E_{k,t}$, generated by Multi-Layer Perceptron (MLP) encoders. Then to predict the positions of the vertices at time $t+1$, an object function f^O and an edge function f^E are used:

$$h^E_{k,t+1} = f^E(h^E_{k,t}, h^O_{u_k,t}, h^O_{v_k,t}) \quad \forall e_{k,t} \in \mathcal{E}_t, \tag{1}$$

$$h^O_{u_k,t+1} = f^O(h^O_{u_k,t}, \sum_{v_k \in \mathcal{N}_{u_k}} h^E_{u_k,v_k,t}) \quad \forall o_{u_k,t} \in \mathcal{O}_t, \tag{2}$$

where $\mathcal{N}_i$ is the set of edges with particle i as the receiver. The functions f^O and f^E are both instantiated with MLPs, and multiple iterations of message passing are employed to propagate the signals across the graph. We denote the model parameters of the GNN by ϕ, which consists of the weights of the MLPs.

B. Sim-and-Real Co-Training

To enable model generalization and real-world transfer, we adopt a co-training strategy that leverages both synthetic data that are agnostic to specific granular material types, and real-world data collected for materials of the target task. The same set of synthetic data is reused across all tasks.

To generate diverse synthetic data, we randomize the initial scene configurations in simulation using the domain randomization parameters such as friction coefficients, object radius, number of objects, and object initial positions. Each episode involves sampling a behavior primitive type a and its parameters, and executing the corresponding action sequence. The simulator uses square-shaped containers with randomly sampled side lengths to introduce geometric variability across episodes. Since the simulator provides full particle-level state information, we directly downsample the resulting point cloud for each timestep to form training triplets $d_i = (X_i, u_i, X_{i+1})$, yielding a synthetic dataset D_{sim} with N_{sim} samples.

Real-world data D_{real} is generated by randomly sampling and executing behavior primitives on the physical system. After each execution, the scooped particles are returned to the container to reset the scene. We record point cloud observations before and after each action and process using the perception module, resulting in N_{real} data points of the same form $d_i = (X_i, u_i, X_{i+1})$.

Since the simulator provides accurate particle-to-particle correspondence, we can adopt the mean squared error (MSE) loss defined as:

$$\mathcal{L}_{\text{MSE}}(\phi; D_{\text{sim}}) = \frac{1}{N_{\text{sim}}} \sum_{d_i \in D_{\text{sim}}} \|X_{i+1} - f_\phi(X_i, u_i)\|_2 , \tag{3}$$

where f_ϕ denotes the learned GNN dynamics model parameterized by ϕ.

In contrast, the real-world data lack pointwise correspondences due to independent downsampling at each timestep. We therefore use the 3D Earth Mover's Distance (EMD) [33] to measure the distributional difference between the predicted and observed point clouds. The EMD loss is defined as:

$$\mathcal{L}_{\text{EMD}}(\phi; D_{\text{real}}) = \frac{1}{N_{\text{real}}} \sum_{d_i \in D_{\text{real}}} \min_{\mu:\, f_\phi(X_i,u_i)\to X_{i+1}} \|\mu(f_\phi(X_i, u_i)) - X_{i+1}\|_2 , \tag{4}$$

where μ is the optimal bijection minimizing transport cost between predicted and ground-truth point sets. While the resolution of the point cloud downsampling affects the EMD loss calculation, we find that it is overall insensitive to the resolution of the downsampling. The final training loss combines both objectives:

$$\mathcal{L} = \alpha \mathcal{L}_{\text{MSE}}(\phi; D_{\text{sim}}) + (1 - \alpha)\mathcal{L}_{\text{EMD}}(\phi; D_{\text{real}}), \tag{5}$$

where α is a constant, and we have set it to 0.98, which works well empirically.

C. Behavior Primitives and Planner

We adopt a model predictive controller to complete the task. The controller iteratively plans a sequence of actions to minimize some objective and executes the first planned action. Here, we first detail the behavior primitives, and then present the MCTS-based planner.

To scoop objects out of the container, one common practice is to adopt parameterized scooping action from past excavation literature [34], but in practice, it is hard to scoop anything when the number of object pieces in the container is low. For humans to accomplish this task, it is intuitive to use the corner edge of the container and leverage counter forces to scoop objects out. Therefore, we define four primitives as defined in Table 1.

To plan sequences of actions over a long horizon for the container clearing task, we utilize Monte Carlo Tree Search (MCTS). MCTS constructs a lookahead search tree to evaluate action sequences from the current state X_0. We enhance the standard MCTS algorithm by incorporating learned neural policy (p_θ) and value (v_θ) networks, inspired by AlphaZero [35], to effectively guide the search. Furthermore, our learned graph-based dynamics model f_ϕ (Section IV-A) is used within the search to predict the outcomes of simulated actions. MCTS builds a search tree through selection, expansion, and backpropagation. At each node, the planner predicts the next state via $X_{i+1} = f_\phi(X_i, a)$ to evaluate and expand the tree. After a fixed number of simulations, the most promising action $a^* \in \mathcal{A}$ is selected based on visit counts $N(X_0, a)$.

Although a natural choice for the reward function for planning is the amount of materials scooped using `Scoop-Corner` and `Scoop-Side`, it is extremely challenging to train the dynamics function to accurately predict particles lifted by the scoop. Instead, we use grid-search to optimize the scooping angle of `Scoop-Corner` and `Scoop-Side` according to a small-scale pilot experiment, and fix the parameters for planning on one container. As a result, we simply use a reward function that encourages the materials to be gathered to the corner of the container, before being scooped up. The reward is defined by the EMD between the predicted state and a manually defined target state, where all particles are gathered in the corner.

V. Experiments

Fig. 3: Different containers (from left to right: Container 1, Container 2, Container 3) and granular materials (from left to right: Almonds, Pistachios, Cashews, Peanuts) considered in this work.

We conducted experiments using materials and containers shown in Fig. 3. As illustrated in Figure 2, each episode begins with a pile of these materials placed in a container on the left. The robot's goal is to transfer all materials to the right container. Using Isaac Gym PhysX [36], we set up a simulation environment designed to closely replicate real-world conditions, including a square storage container, a spoon of similar shape to the one used in the real-world setup, a 7-DOF robot arm, and spherical objects to simulate granular objects. The setting of action primitives is the same for both the simulation and real-world experiments.

We seek to address three key questions in the experiments: 1) Do dynamics models co-trained with both simulated and real data outperform those trained solely on simulated or

Name	Description	Params
`Sweep-Left`	gathers particles towards the left side	x_init, x_end
`Sweep-Front`	gathers particles towards the front side	y_init, y_end
`Scoop-Corner`	scoops particles from a corner region	x_init, y_init, rotation angle
`Scoop-Side`	scoops particles accumulated along a container wall	x_init, twist angle

TABLE I: Description and parameters of behavior primitives

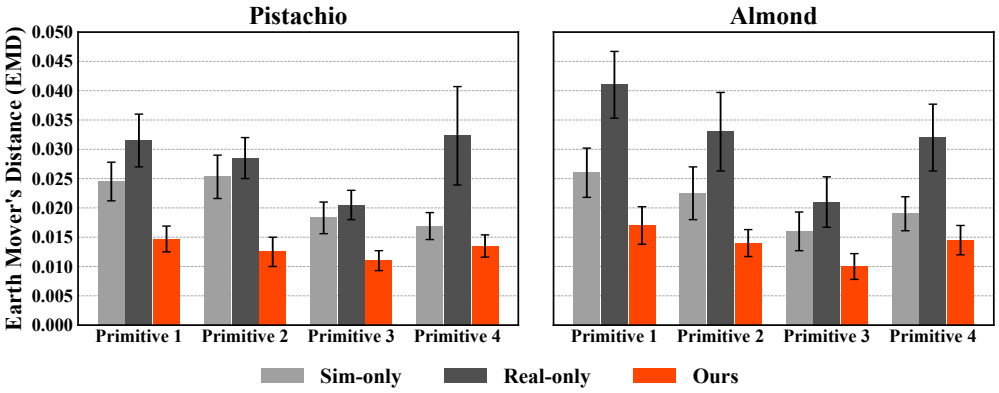

Fig. 4: Dynamics prediction results in terms of 3D EMD using the Almond and Pistachio materials in Container 1 with the four primitives.

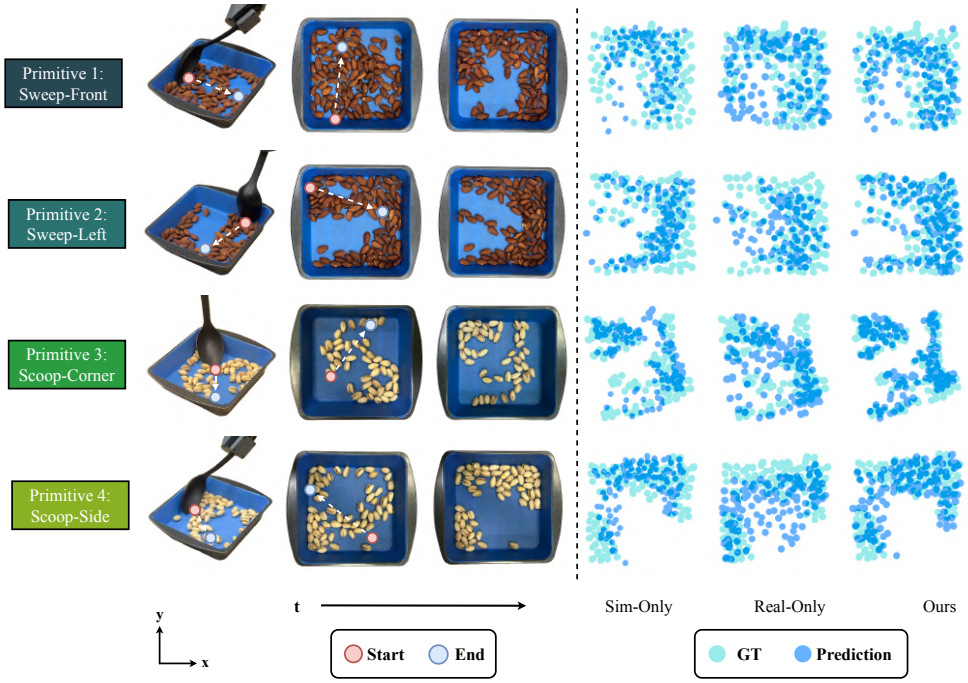

Fig. 5: Qualitative result of different dynamics model prediction for different materials and skills. From left to right: side view of the primitive start; top-down view of the object state change before and after the action; corresponding dynamics prediction and ground truth from three different models. The red and blue points indicate the starting and ending point of the action. For the scoop primitive, we only consider the first half (dragging) as the goal is to gather as many particles to the corner as possible.

real data in terms of prediction accuracy? 2) Does this lead to better manipulation policies? 3) Can the approach generalize to different materials and containers?

A. *Effectiveness of Dynamics Models*

To answer the first question, we compare the prediction error measured by 3D EMD loss across three different dynamics models: a GNN dynamics model trained from simulation data (denoted as sim-only), a GNN dynamics model trained on real-world data (denoted as real-only), and a GNN dynamics model co-trained with sim and real data (denoted as ours). For these experiments, we use the Almond and Pistachio materials in Container 1. We use 2500 samples from simulation and 50 samples from the real-world data for training, with the training procedure described in Section IV-B.

The quantitative results of the experiments are demonstrated in Fig. 4, and some qualitative results are shown in Fig. 5. The model co-trained with sim and real data (ours) outperforms those trained on pure simulation data (sim-only) and pure real-world data (real-only) across all skills and materials. Our empirical analysis reveals distinct limitations of sim-only or real-only training approaches. The sim-only model captures general motion trends but exhibits large sim-to-real gaps, consistently overestimating manipulation effects. In contrast, the real-only model, constrained by limited training samples, fails to generalize effectively—particularly

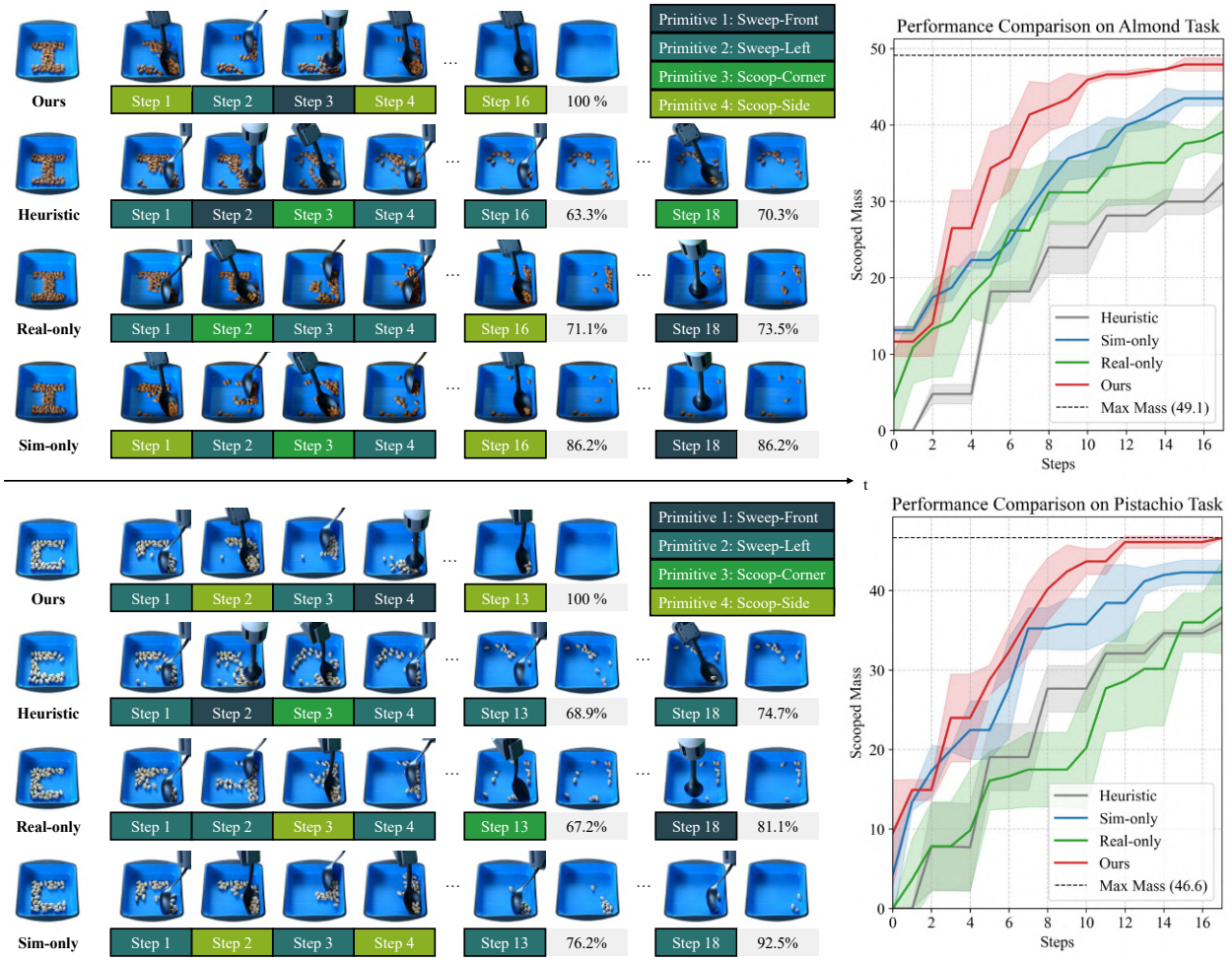

Fig. 6: Left: Manipulation rollouts of Ours, heuristic, sim-only, and real-only approaches for Almond and Pistachio. Right: Scooped mass over action steps. Our method achieves 100% success rate within 18 action steps, outperforming baselines by 29.7%, 26.5%, and 13.8% (heuristic, real-only, sim-only) for almond scooping, and by 25.3%, 18.9%, and 7.5% for pistachio scooping at the final state. Primitive actions (Sweep-Front, Sweep-Left, Scoop, Scoop-Side) are color-coded beneath each rollout visualization. Performance curves show scooped mass trajectories with variance bands, where our approach demonstrates superior convergence and consistency compared to baseline methods, scooping out maximum capacity earlier with lower variance.

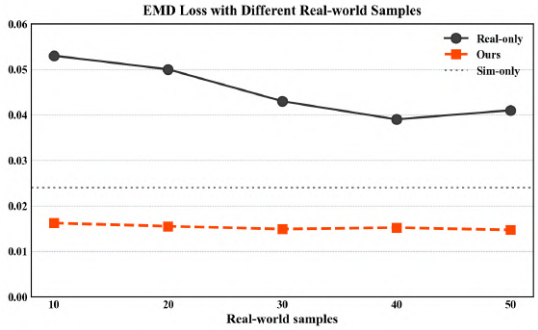

Fig. 7: 3D EMD loss for `Sweep-Front` with Almond in Container 1 with different numbers of real samples.

evident in its inability to predict pushing and sweeping dynamics accurately. By leveraging both simulation priors and real-world interactions, our co-training approach achieves better dynamics prediction that closely matches ground truth distributions while maintaining consistency across diverse manipulation scenarios.

We also evaluated the effect of increasing the amount of real-world data on model performance, as shown in Fig. 7. Although the model trained exclusively on real-world data improved with more samples, its prediction performance plateaued quickly despite the increased data. In contrast, models trained using our hybrid framework required fewer real-world samples yet achieved higher accuracy than models trained on simulation or real-world data alone. This highlights the importance of using simulated data, which is **not only helpful but also necessary to the quality of the learned granular dynamics model**.

B. Manipulation Results

To demonstrate the effectiveness of the dynamics model on the scooping task, we conducted comprehensive SAGO experiments on Pistachio and Almond. The baseline methods include using the same MCTS planner but with the sim-

Fig. 8: Rollouts of our method on unseen materials: Peanuts and Cashews. Our approach, trained only on almonds and pistachios, generalizes to materials with similar granular properties. The method also adapts to different containers with minimal data: using only 10 interaction samples per skill, our approach completes scooping in 18 and 9 steps for pistachios and almonds respectively in new containers, and achieves 100% success on peanuts (13 steps) and cashews (11 steps).

only and real-only dynamics models, and an additional heuristic baseline with fixed primitive order (`Sweep-Left`, `Sweep-Front`, and `Scoop-Corner`). For all the rollouts, we set an action budget of 18. The results are shown in Fig. 6. Our method is able to outperform all the baselines. Notably, our model could accomplish the task with an average number of actions of 14.7 on the Pistachio task, which none of the other baselines could finish.

Interestingly, the model trained purely on simulation data (sim-only) achieves better manipulation results than the model trained on real-world data (real-only), which aligns with the dynamics prediction result. We hypothesize that this is because the sim-only model captures general object movements more effectively, while the real-only model, limited by the number of real-world samples, struggles to identify the effects of certain actions.

C. Generalization to New Materials

To evaluate our approach's generalization capabilities to novel materials with similar properties (Peanuts and Cashews), we deploy the dynamics model co-trained with simulation data and Pistachio data without finetuning in the planner. As shown in Fig. 8, our approach achieves successful scooping with a 100% completion rate, requiring only 13 and 11 action steps, respectively. This zero-shot transfer demonstrates that our sim-and-real co-training framework has the potential to learn generalizable manipulation strategies that transfer to materials with similar granular properties, without requiring further task-specific training for each new material. On the other hand, the planner that uses a real-only dynamics suffers in this zero-shot deployment.

D. Adaptation to Novel Container

To evaluate adaptability to different container geometries, we tested whether minimal real-world samples enabled transfer to unseen containers. Our approach, trained in simulation and real-world data from container 1, is augmented with only 10 data points each from novel Container 2 and Container 3. As shown in Fig. 8, with these minimal extra data, our system successfully completes the tasks in 18 steps for Pistachios and 9 steps for Almonds in the new containers. This demonstrates that our sim-and-real co-training framework has the potential to effectively adapt to novel container geometries as well.

VI. Conclusions

In this work, we address the problem of scooping all granular materials from containers, which requires long-term reasoning about the dynamics of granular particles interacting with the scoop and the container. We develop a framework combining a graph-based neural dynamics model with an MCTS planner for long-horizon decision-making. We demonstrated the importance of co-training the GNN on both simulated and real-world data, which leads to efficient scooping and significantly enhances our pipeline's performance over those with models trained solely on simulation or real-world data.

Our work exhibits several limitations: (1) the system requires high-quality primitives, which are sometimes infeasible and inflexible in an open-world setting; (2) we have not

considered very small grains of objects like corn and rice, which may be a future direction for broader use cases.

VII. ACKNOWLEDGMENTS

The authors would like to thank Haochen Shi for his guidance on the GNN implementation, Jun Lv for his insightful discussions, and Zhe Huang for his valuable feedback on an early draft of this paper.

REFERENCES

[1] J. Grannen, Y. Wu, S. Belkhale, and D. Sadigh, "Learning bimanual scooping policies for food acquisition," in *Conference on Robot Learning (CoRL)*, 2022.

[2] D. Park, Y. Hoshi, H. P. Mahajan, H. K. Kim, Z. Erickson, W. A. Rogers, and C. C. Kemp, "Active robot-assisted feeding with a general-purpose mobile manipulator: Design, evaluation, and lessons learned," *Robotics and Autonomous Systems*, vol. 124, p. 103344, 2020.

[3] D. Park, Y. K. Kim, Z. M. Erickson, and C. C. Kemp, "Towards assistive feeding with a general-purpose mobile manipulator," *arXiv preprint arXiv:1605.07996*, 2016.

[4] C. Schlette, A. G. Buch, F. Hagelskjær, I. Iturrate, D. Kraft, A. Kramberger, A. P. Lindvig, S. Mathiesen, H. G. Petersen, M. H. Rasmussen *et al.*, "Towards robot cell matrices for agile production–sdu robotics' assembly cell at the wrc 2018," *Advanced Robotics*, vol. 34, no. 7-8, pp. 422–438, 2020.

[5] Q. Lu, Y. Zhu, and L. Zhang, "Excavation reinforcement learning using geometric representation," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 4472–4479, 2022.

[6] H. Chen, Y.-J. Mun, Z. Huang, Y. Niu, Y. Xie, D. L. McPherson, and K. Driggs-Campbell, "Learning task skills and goals simultaneously from physical interaction," *arXiv preprint arXiv:2309.04596*, 2023.

[7] Y. Zhu, P. Thangeda, M. Ornik, and K. Hauser, "Few-shot adaptation for manipulating granular materials under domain shift," *arXiv preprint arXiv:2303.02893*, 2023.

[8] Y. Niu, S. Jin, Z. Zhang, J. Zhu, D. Zhao, and L. Zhang, "Goats: Goal sampling adaptation for scooping with curriculum reinforcement learning," *arXiv preprint arXiv:2303.05193*, 2023.

[9] C. Schenck, J. Tompson, S. Levine, and D. Fox, "Learning robotic manipulation of granular media," in *Proceedings of the 1st Annual Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, S. Levine, V. Vanhoucke, and K. Goldberg, Eds., vol. 78. PMLR, 13–15 Nov 2017, pp. 239–248. [Online]. Available: https://proceedings.mlr.press/v78/schenck17a.html

[10] Y. Zhu, L. Wang, and L. Zhang, "Excavation of fragmented rocks with multi-modal model-based reinforcement learning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022.

[11] H. Shi, H. Xu, Z. Huang, Y. Li, and J. Wu, "Robocraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks," *arXiv preprint arXiv:2205.02909*, 2022.

[12] Y. Wang, Y. Li, K. Driggs-Campbell, L. Fei-Fei, and J. Wu, "Dynamic-resolution model learning for object pile manipulation," *arXiv preprint arXiv:2306.16700*, 2023.

[13] S. Dadhich, U. Bodin, and U. Andersson, "Key challenges in automation of earth-moving machines," *Automation in Construction*, vol. 68, pp. 212–222, 2016.

[14] K. Takahashi, W. Ko, A. Ummadisingu, and S.-i. Maeda, "Uncertainty-aware self-supervised target-mass grasping of granular foods," in *IEEE International Conference on Robotics and Automation*, 2021, pp. 2620–2626.

[15] K. Zhang, B. Li, K. Hauser, and Y. Li, "Adaptigraph: Material-adaptive graph-based neural dynamics for robotic manipulation," *arXiv preprint arXiv:2407.07889*, 2024.

[16] J. Bjorck, F. Castañeda, N. Cherniadev, X. Da, R. Ding, L. Fan, Y. Fang, D. Fox, F. Hu, S. Huang *et al.*, "Gr00t n1: An open foundation model for generalist humanoid robots," *arXiv preprint arXiv:2503.14734*, 2025.

[17] K. Bousmalis, A. Irpan, P. Wohlhart, Y. Bai, M. Kelcey, M. Kalakrishnan, L. Downs, J. Ibarz, P. Pastor, K. Konolige *et al.*, "Using simulation and domain adaptation to improve efficiency of deep robotic grasping," in *2018 IEEE international conference on robotics and automation (ICRA)*. IEEE, 2018, pp. 4243–4250.

[18] A. Maddukuri, Z. Jiang, L. Y. Chen, S. Nasiriany, Y. Xie, Y. Fang, W. Huang, Z. Wang, Z. Xu, N. Chernyadev *et al.*, "Sim-and-real co-training: A simple recipe for vision-based robotic manipulation," *arXiv preprint arXiv:2503.24361*, 2025.

[19] S. Luo, Q. Peng, J. Lv, K. Hong, K. R. Driggs-Campbell, C. Lu, and Y.-L. Li, "Human-agent joint learning for efficient robot manipulation skill acquisition," in *2025 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2025, pp. 1370–1377.

[20] A. Curtis, X. Fang, L. P. Kaelbling, T. Lozano-Pérez, and C. R. Garrett, "Long-horizon manipulation of unknown objects via task and motion planning with estimated affordances," in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 1940–1946.

[21] A. Simeonov, Y. Du, B. Kim, F. Hogan, J. Tenenbaum, P. Agrawal, and A. Rodriguez, "A long horizon planning framework for manipulating rigid pointcloud objects," in *Proceedings of the 2020 Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, J. Kober, F. Ramos, and C. Tomlin, Eds., vol. 155, 16–18 Nov 2021, pp. 1582–1601.

[22] C. R. Garrett, R. Chitnis, R. Holladay, B. Kim, T. Silver, L. P. Kaelbling, and T. Lozano-Pérez, "Integrated task and motion planning," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, no. 1, pp. 265–293, 2021. [Online]. Available: https://doi.org/10.1146/annurev-control-091420-084139

[23] J. E. King, V. Ranganeni, and S. S. Srinivasa, "Unobservable monte carlo planning for nonprehensile rearrangement tasks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2017, pp. 4681–4688.

[24] H. Song, J. A. Haustein, W. Yuan, K. Hang, M. Y. Wang, D. Kragic, and J. A. Stork, "Multi-object rearrangement with monte carlo tree search: A case study on planar nonprehensile sorting," in *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE, 2020, pp. 9433–9440.

[25] B. Huang, T. Guo, A. Boularias, and J. Yu, "Interleaving monte carlo tree search and self-supervised learning for object retrieval in clutter," in *2022 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2022.

[26] B. Huang, X. Zhang, and J. Yu, "Toward optimal tabletop rearrangement with multiple manipulation primitives," *arXiv preprint arXiv:2310.00167*, 2023.

[27] C. B. Browne, E. Powley, D. Whitehouse, S. M. Lucas, P. I. Cowling, P. Rohlfshagen, S. Tavener, D. Perez, S. Samothrakis, and S. Colton, "A survey of monte carlo tree search methods," *IEEE Transactions on Computational Intelligence and AI in games*, vol. 4, no. 1, pp. 1–43, 2012.

[28] I. Parberry, *Introduction to Game Physics with Box2D*. CRC Press, 2017.

[29] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," 2016.

[30] J. Park, Q.-Y. Zhou, and V. Koltun, "Colored point cloud registration revisited," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 143–152.

[31] C. Moenning and N. A. Dodgson, "Fast marching farthest point sampling," University of Cambridge, Computer Laboratory, Tech. Rep., 2003.

[32] H. Chen, Y. Niu, K. Hong, S. Liu, Y. Wang, Y. Li, and K. R. Driggs-Campbell, "Predicting object interactions with behavior primitives: An application in stowing tasks," in *Conference on Robot Learning (CoRL)*, 2023.

[33] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *International journal of computer vision*, vol. 40, pp. 99–121, 2000.

[34] Q. Lu and L. Zhang, "Excavation learning for rigid objects in clutter," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7373–7380, 2021.

[35] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel *et al.*, "Mastering chess and shogi by self-play with a general reinforcement learning algorithm," *arXiv preprint arXiv:1712.01815*, 2017.

[36] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.