Deep Field Dynamics Model for Granular Object Piles Manipulation

Shangjie Xue, Shuo Cheng, Pujith Kachana, Danfei Xu
Georgia Institute of Technology

I. Introduction

Granular objects such as beans, nuts, and ball bearings are ubiquitous in daily life and industry [1], making accurate granular modeling and manipulation essential for various robotic tasks. However, such modeling is challenging due to the large number of particles and complex dynamics, as well as the properties of granular materials such as size, shape, friction, and contact mechanics. Overcoming these challenges is crucial to developing accurate and efficient learning-based models that enable robots to perform tasks involving granular materials.

To address these challenges, recent efforts have been made to model granular pile dynamics using deep latent dynamics models directly from pixel input [2]. However, such models have shown limitations in capturing the dynamics, underperforming a linear dynamics model due to a lack of inductive biases [2]. Alternatively, physics-inspired concepts such as particles offer strong inductive biases for deep dynamics models. A long line of existing works approximate a system as a collection of particles and accurately model inter-particle dynamics [3]. However, particle-based techniques pose scalability challenges for granular manipulation as their memory and computational costs grow superlinearly with the number of particles [4] [5]. Moreover, these methods assume that all particles have identical properties and that each particle can be tracked, which is not practical for granular manipulation.

We propose that field-based representations are ideal for modeling granular piles as they avoid challenges with particle interactions, process input in pixel space, and account for sparsity and spatial equivariance. We introduce the Deep Field Dynamics Model, which efficiently models granular material dynamics based on fully convolutional networks. Our model is also fully differentiable and amenable to gradient-based trajectory optimization methods for complex motion planning.

We evaluate our approach in different simulation and real-world scenarios and demonstrate that field-based models are more efficient and accurate compared to existing latent dynamics and object-centric models [2]. Our model is also capable of performing complex, unseen planning tasks such as pushing piles and avoiding obstacles. Additionally, we show the generality of our field-based representation by demonstrating transferability to different environments with varying pusher and object shapes in a zero-shot setting.

II. Related Works

Learned Dynamics Model with Inductive Bias. There has been growing interest in learning dynamics models with a strong inductive bias for robot manipulation tasks. Particle-based methods [3], [6]–[8] have been developed for expressing the structure of objects and manipulating deformable objects. However, particle-based approaches, face scalability issues as the number of particles increases, thereby making them computationally expensive and challenging to use in practical planning tasks [5].

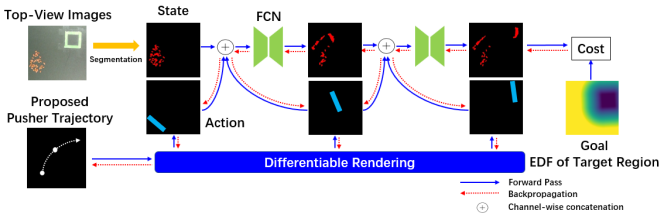

Figure 1. **Illustration of our framework which enables a robot to move object piles into the target region over multiple pushes. Using an initial observation and hypothesized action sequence, our model generates future states and calculates the cost of the final task goal. We extract the optimal action sequence by minimizing cost and backpropagating the gradient.**

Pile Manipulation, specifically pushing piles of objects to target regions, is a challenging task due to the complex dynamics and the large number of objects involved. Suh et al. proposed a dense latent dynamics model for pile manipulation [2]. However, they found that the neural network-based dynamics model under-performed both a linear least squares model and an object-centric baseline. Additionally, model-free reinforcement learning approaches [9], [10] are proposed to predict the start and end poses of pushes. However, training such models require task-specific oracle demonstrations, making it hard to generalize across different environments.

III. Method

Developing dynamics models for granular materials manipulation is challenging due to the complex and non-convex nature of their dynamics, as well as the need for scalability and compatibility with planning algorithms. To address these challenges, we propose a learning-based model based on Fully Convolutional Network (FCN) that captures the complexity of granular materials dynamics. By representing objects as images using the Eulerian approach, our FCN model is scalable and well-suited for real-world scenarios. In this section, after formulating the problem, we show how our proposed model better captures the complex dynamics of granular materials and integrates with trajectory optimization algorithms.

Problem Statement. This study aims to use a flat-surface pusher, such as a spatula, to interact with piles of granular

materials, in order to push all the particles to a target region. The model takes the state of objects, represented as an image $\boldsymbol{s} \in \mathcal{R}^{H \times W}$ (see Sec. III State and Action Representation), and the target region $\boldsymbol{G} \in \mathcal{R}^{H \times W}$ as input. Unlike previous methods that only generate start and end positions assuming straight-line pushes, our method generates curvilinear trajectories that are represented by a sequence of poses of the pusher, for more flexible behaviors (more detail in Sec. III Trajectory Optimization with Learned Dynamics), allowing the robot to execute continuous pushes to avoid obstacles.

State and Action Representation. To develop a scalable model that is independent of the number of particles, we propose to represent the system state as a grid-based density field of the granular materials. This approach, known as the Eulerian approach, avoids the explicit modeling and perception of each particle and enables our model to scale effectively in complex real-world scenarios. Specifically, we capture the density field state $\boldsymbol{s}$ by segmenting an RGB image into a one-channel occupancy grid after an orthographic projection. We also propose to represent the action as the rendered image of the pusher, which could also be viewed as the density field of the pusher. Moreover, to bridge the gap between the learning-based dynamics model operating on image space and gradient-based trajectory optimization operating on robot poses, we utilize a differentiable renderer that maps gripper poses to images (see figure 1). This allows the error to be backpropagated to the poses through the neural network-based dynamics model during optimization. In particular, the action is represented as $\boldsymbol{a}_t \coloneqq [r(\boldsymbol{x}_t), r(\boldsymbol{x}_{t+1})]$, where $\boldsymbol{x}_t \in SE(2)$ is the proposed pusher pose at time t, we assume during a short time interval $[t, t+1)$, the pusher move linearly, therefore we can use $\boldsymbol{x}_t, \boldsymbol{x}_{t+1}$ to represent a small segment of the trajectory during t to $t+1$. In addition, $r : SE(2) \rightarrow \mathcal{R}^{H \times W}$ is a differentiable rendering function that rasterizes a pusher pose into a one-channel image representing the density field of the pusher in the plane. (see Fig. 1).

Learning Dynamics. To overcome the lack of inductive bias that limits the accuracy of previous latent-dynamics models, we proposed using FCNs as the dynamics model. This approach draws inspiration from computational physics, which models contact dynamics under an Eulerian representation to solve the PDEs that describe physical systems [11]–[13], and also leverages the analogy between convolutional neural networks and discretizations of PDEs [14]. Moreover, FCN-based dynamics model are translational equivariant by nature [15], as $f_\theta(\omega * \boldsymbol{s}, \omega * \boldsymbol{a}) = \omega * f_\theta(\boldsymbol{s}, \boldsymbol{a})$, where f_θ is the FCN with parameter θ, and ω is a translation. Such translational equivariance provides better learning efficiency. Our dynamics model adopts a shallow U-Net [16] architecture to ensure computational efficiency. Given the current state $\boldsymbol{s}_t$ and the proposed action $\boldsymbol{a}_t$ as input, the dynamics model predicts the next state $\boldsymbol{s}_{t+1}$, As is shown in Fig 1, the dynamics model could be written as: $\hat{\boldsymbol{s}}_{t+1} = f_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t)$. The network is trained in a self-supervised manner, using randomized pushing data in environments to predict subsequent steps following an action, minimizing the loss function $\mathcal{L}_{train} = ||f_\theta(\boldsymbol{s}_t, \boldsymbol{a}_t) - \boldsymbol{s}_{t+1}||_F^2$

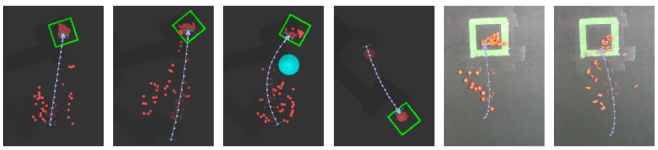

Figure 2. Qualitative Results: (1) straight line pushing (2) curved pushing (3) Obstacle avoidance (4) pushing different objects without retraining (5-6) real-world experiments on pushing blocks and beans without retraining.

between the predicted and observed states, where $||\cdot||_F$ is the Frobenius norm. To perform sequential prediction for long-horizon, the predicted state $\hat{\boldsymbol{s}}_{t+1}$ are fed again into the dynamics model along with the proposed action $\boldsymbol{a}_{t+1}$. Noticing that $\boldsymbol{a}_t$, $\boldsymbol{a}_{t+1}$ are not necessarily along the same direction, and hence allows us to model curvilinear trajectory.

Trajectory Optimization with Learned Dynamics. To effectively apply our model in complex scenarios, where obstacles may exist within the scene, we integrate the learning-based dynamics model with gradient-based trajectory optimization. The objective of such optimization is to maximize the amount of material that is pushed to the designated target region, while simultaneously avoiding collisions with obstacles. The optimization problem is defined as follows:

$$\begin{aligned} &\min_{\boldsymbol{x}_t \in SE(2)} \ell_{goal}(\boldsymbol{s}_T; \boldsymbol{G}) + \sum_{i=0}^{T} \ell_{action}(\boldsymbol{x}_i; \boldsymbol{O}) + \sum_{i=1}^{T} \ell_{state}(s_i; \boldsymbol{O}) \\ &\text{s.t.} \quad \boldsymbol{s}_{t+1} = f_\theta(\boldsymbol{x}_t, [r(\boldsymbol{x}_t), r(\boldsymbol{x}_{t+1})]) \ \forall t \in [0, T-1] \end{aligned} \tag{1}$$

where $\boldsymbol{G} \in \mathcal{R}^{H \times W}$ is the mask of the target zone and it is assigned 0 inside the zone and 1 out of the zone, moreover,

$$\begin{aligned} \ell_{goal}(\boldsymbol{s}_T; \boldsymbol{G}) &\coloneqq \alpha_1 \langle EDF(\boldsymbol{G}), \boldsymbol{s}_T \rangle_F - \alpha_2 \langle \boldsymbol{G}, \boldsymbol{s}_T \rangle_F \\ \ell_{state}(\boldsymbol{s}_i; \boldsymbol{O}) &\coloneqq \alpha_3 \langle \boldsymbol{O}, \boldsymbol{s}_T \rangle_F \\ \ell_{action}(\boldsymbol{x}_i; \boldsymbol{O}) &\coloneqq \alpha_4 \langle \boldsymbol{O}, r(\boldsymbol{x}_i) \rangle_F \end{aligned} \tag{2}$$

where $EDF(\boldsymbol{G})$ computes the Euclidean Distance Field (EDF) of the target zone. $\langle , \rangle_F$ is the Frobenius inner product (i.e. the sum of element-wise product), $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ are positive weight parameters. ℓ_{action} and ℓ_{state} are loss functions utilized to address geometric constraints such as obstacles avoidance, $\boldsymbol{O} \in \mathcal{R}^{H \times W}$ is the mask of the obstacle objects.

To enhance the efficiency of manipulation in scenarios where obstacles hinder the direct path of motion, and achieve a more human-like behavior, curved pushing is required as opposed to straight-line pushing. As in such cases, optimal straight-line pushes may collide with obstacles, making it necessary to adopt a curvilinear motion to optimally achieve the task. In our experiments, in order to ensure the smoothness of generated trajectory, B-spline curves are applied to parametrize the trajectory due to their flexibility and ability to smoothly interpolate data points. The parametrized curves could be viewed as an additional constraint in 1. To further account for the non-convex nature of the dynamics model and mitigate the risk of local minima, an initial random sampling of the control points for the spline curves is performed. Subsequently, the optimization is conducted in batches, and the resulting optimal solution that minimizes the cost function is chosen as the output of the planning model.

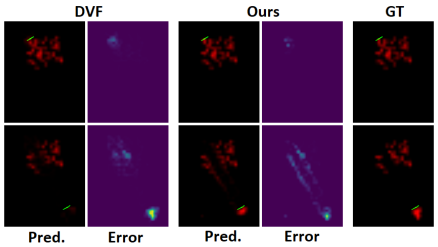

Figure 3. Visualizing model predictions and errors.

IV. Experiments

We performed evaluations in simulation and real-world environments. All methods are trained with simulated data generated from a PyBullet-based simulator. We measure success rate, which is the fraction of objects that are inside the goal region at the end of an episode, and cost function $\ell_{eval}(\boldsymbol{s_t}; \boldsymbol{G}) = \langle EDF(\boldsymbol{G}), \boldsymbol{s_t} \rangle_F$ which could also be viewed as the Control Lyapunov Function as has been pointed out by Suh et al [2]. 20 scenes that are randomly initialized are utilized to evaluate each model, with object positions and target regions being randomly sampled. Performance evaluation is conducted for each instance following 10 pushes, which corresponds to the optimal demonstration's push count and reflects our objective of effectively pushing piles into the target region with minimal pushes. We evaluate different planning methods for our proposed model: (1) straight line pushing using the shooting method [17] with only forward pass (Forward), (2) trajectory optimization with straight line pushing (Opt-Line), and (3) trajectory optimization with curved pushing (Opt-Curve). For Forward and Opt-Line, we compare both single-step prediction as well as sequential prediction with DVF [2].

Our method outperforms the baseline methods in speed and accuracy. The evaluation of prediction accuracy is performed between our method and DVF [2] which is a dense latent dynamics model. We improved the performance of the original DVF model by augmenting the data with intermediate states from sequential pushes. Mean Squared Error (MSE) between the predicted states and the ground truth is used to evaluate the prediction performance. Quantitative results(table I) and qualitative results demonstrate (figure 3) that our method outperforms the DVF model in both single-push and sequential-push scenarios in prediction. Additionally, our method exhibits superior efficiency, as it utilizes fewer parameters and requires fewer computational resources compared to the DVF model. Furthermore, The rollout evaluation is performed among our methods, DVF, object-centric method proposed in [2], and Transporter Network [9]. Our sequential prediction-based approach outperforms all baselines in both metrics (see table II and III). Notably, despite our method not being trained on expert data, it achieves better performance than the Transporter Network which was trained by 10,000 expert pushes. As illustrated in Figure 4, our method reaches the goal significantly faster than other methods, indicating the optimality of our approach. Moreover, our method closely approximates expert behavior, even in the absence of expert demonstration data. However, in the second half of the horizon, our method's convergence rate is slower than that of expert demonstration. This is because our randomly generated dataset rarely involves pushing one particle to a pile of particles once most particles are already in the target region, which results in less accurate predictions.

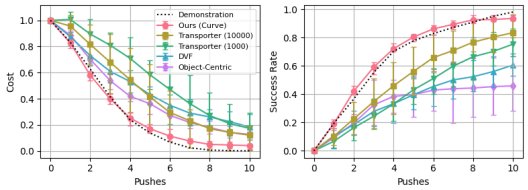

Figure 4. Rollout performance

Table I
Prediction Performance

	Params	GFLOP	Error	
			Single Pred.	Sequential Pred.
DVF [2]	6.9E+7	1.6E-1	7.8E-4	2.1E-3
Ours	**1.5E+4**	**7.4E-3**	**5.7E-4**	**4.4E-4**

Table II
Rollout Performance

Model		Single Prediction		Sequential Prediction	
		Cost	Success	Cost	Success
Object Centric [2]	Forward	0.1265	0.459	-	-
DVF	Forward	0.1838	0.606	1.103	0.014
DVF	Opt-Line	0.1498	0.768	1.126	0.015
Ours	Forward	0.1677	0.744	0.0819	0.8
Ours	Opt-Line	**0.0717**	**0.899**	**0.0197**	**0.966**
Ours	Opt-Curve	-	-	0.0416	0.936

Table III
Comparison with Model-free Method

	# Expert Pushes	w/o Obstacle		w/ Obstacle	
		Cost	Success	Cost	Success
Transporter [9]	1000	0.1731	0.754	0.575	0.049
	10000	0.1211	0.835	0.5331	0.088
Ours (Opt-Curve)	0	**0.0361**	**0.949**	**0.1318**	**0.838**

By using the sequential prediction model, we are able to generate flexible trajectory in complex environments. Notably, our approach can effectively plan in the presence of obstacles between the initial objects and the target region. As illustrated in Table III, our method achieves high performance despite not being explicitly trained for such scenarios. This highlights the generalizability of our approach, especially compared to model-free methods like the Transporter.

Our method can generalize to diverse environments with different dynamics without learning. As we varied the shapes and sizes of objects and pushers, despite using unseen pusher lengths (50% longer and 20% shorter) and unseen objects (one big circular block instead of small square particles), our method achieved excellent performance, with costs of 0.016, 0.108, and 0.002 respectively. Moreover, in real-world experiments, our model performs well with unseen objects, such as beans, without retraining (see figure 2). This emphasizes the generality of our method and its ability to generalize well across different environments.

REFERENCES

[1] P. Richard, M. Nicodemi, R. Delannay, P. Ribiere, and D. Bideau, "Slow relaxation and compaction of granular systems," *Nature materials*, vol. 4, no. 2, pp. 121–128, 2005.

[2] H. Suh and R. Tedrake, "The surprising effectiveness of linear models for visual foresight in object pile manipulation," in *International Workshop on the Algorithmic Foundations of Robotics*. Springer, 2021, pp. 347–363.

[3] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," *arXiv preprint arXiv:1810.01566*, 2018.

[4] D. E. Stewart, "Rigid-body dynamics with friction and impact," *SIAM review*, vol. 42, no. 1, pp. 3–39, 2000.

[5] Z. Pan, A. Zeng, Y. Li, J. Yu, and K. Hauser, "Algorithms and systems for manipulating multiple objects," *IEEE Transactions on Robotics*, 2022.

[6] P. Battaglia, R. Pascanu, M. Lai, D. Jimenez Rezende *et al.*, "Interaction networks for learning about objects, relations and physics," *Advances in neural information processing systems*, vol. 29, 2016.

[7] 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.

[8] Z. Xu, Z. He, J. Wu, and S. Song, "Learning 3d dynamic scene representations for robot manipulation," *arXiv preprint arXiv:2011.01968*, 2020.

[9] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani *et al.*, "Transporter networks: Rearranging the visual world for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2021, pp. 726–747.

[10] M. Shridhar, L. Manuelli, and D. Fox, "Cliport: What and where pathways for robotic manipulation," in *Conference on Robot Learning*. PMLR, 2022, pp. 894–906.

[11] D. J. Benson and S. Okazawa, "Contact in a multi-material eulerian finite element formulation," *Computer methods in applied mechanics and engineering*, vol. 193, no. 39-41, pp. 4277–4298, 2004.

[12] G. Falkovich, *Fluid mechanics: A short course for physicists*. Cambridge University Press, 2011.

[13] K.-S. Ku, C.-H. An, K.-C. Li, and M.-I. Kim, "An eulerian model for the motion of granular material with a large stokes number in fluid flow," *International Journal of Multiphase Flow*, vol. 92, pp. 140–149, 2017.

[14] C. Rackauckas, "Parallel computing and scientific machine learning (sciml): Methods and applications," 2022. [Online]. Available: https://github.com/SciML/SciMLBook

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.

[16] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[17] R. Tedrake, *Underactuated Robotics*, 2023. [Online]. Available: https://underactuated.csail.mit.edu