ParticleFormer: A 3D Point Cloud World Model for Multi-Object, Multi-Material Robotic Manipulation

Suning Huang†
Stanford University
suning@stanford.edu

Qianzhong Chen
Stanford University
qchen23@stanford.edu

Xiaohan Zhang
RAI Institute
xzhang@rai-inst.com

Jiankai Sun
Stanford University
jksun@stanford.edu

Mac Schwager†
Stanford University
schwager@stanford.edu

Abstract: 3D world models (i.e., learning-based 3D dynamics models) offer a promising approach to generalizable robotic manipulation by capturing the underlying physics of environment evolution conditioned on robot actions. However, existing 3D world models are primarily limited to single-material dynamics using a particle-based Graph Neural Network model, and often require time-consuming 3D scene reconstruction to obtain 3D particle tracks for training. In this work, we present `ParticleFormer`, a Transformer-based point cloud world model trained with a hybrid point cloud reconstruction loss, supervising both global and local dynamics features in multi-material, multi-object robot interactions. `ParticleFormer` captures fine-grained multi-object interactions between rigid, deformable, and flexible materials, trained directly from real-world robot perception data without an elaborate scene reconstruction. We demonstrate the model's effectiveness both in 3D scene forecasting tasks, and in downstream manipulation tasks using a Model Predictive Control (MPC) policy. In addition, we extend existing dynamics learning benchmarks to include diverse multi-material, multi-object interaction scenarios. We validate our method on six simulation and three real-world experiments, where it consistently outperforms leading baselines by achieving superior dynamics prediction accuracy and less rollout error in downstream visuomotor tasks. Experimental videos are available at particleformer.

Keywords: Robotic Manipulation, Learning-based Dynamics Modeling, Model-based Planning and Control

1 Introduction

Recent advances in reinforcement learning (RL) [1, 2, 3, 4, 5] and imitation learning (IL) [6, 7, 8, 9, 10] have enabled robots to acquire sophisticated manipulation skills across diverse tasks. However, most existing approaches directly map observations to actions, inherently coupling environmental dynamics with task-specific objectives. This tight coupling hinders generalization, as agents must learn solutions for every task-environment combination [11, 12, 13]. In contrast, humans generalize effectively by forming flexible mental models of physical interactions and intuitively decoupling task goals from environment dynamics through visual perception [14, 15, 16]. Inspired by this insight, developing world models [17] that capture underlying physical interaction dynamics is crucial for improving generalization in robotic manipulation.

These world models have a long history in robotics and control [18, 19]. Such models have been proposed with 2D (pixel space) or 3D (metric space) representations. For 3D world models, re-

†Corresponding author

9th Conference on Robot Learning (CoRL 2025), Seoul, Korea.

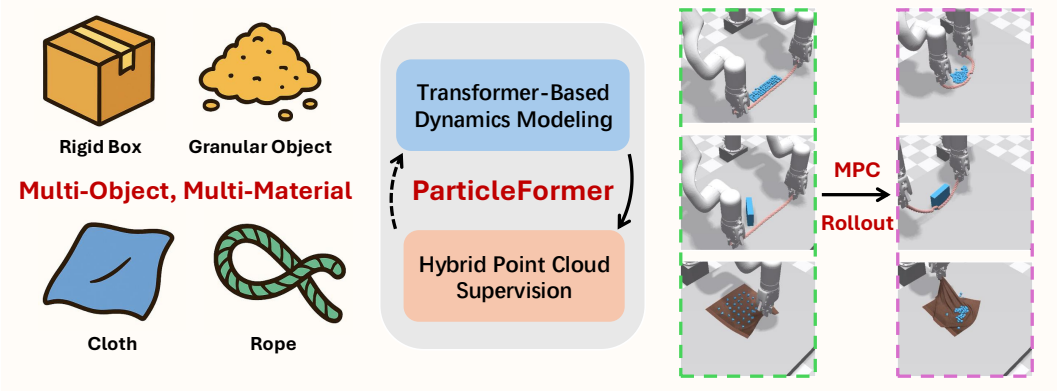

Figure 1: **Motivation.** Modeling dynamics in multi-object, multi-material scenarios is challenging due to complex and heterogeneous interactions. In this paper, we propose `ParticleFormer`, a Transformer-based point cloud world model trained with hybrid supervision, enabling accurate prediction and model-based control in robotic manipulation tasks.

searchers have shown that by parameterizing objects as compositions of basic particles in 3D space and applying a graph-based neural dynamics (GBND) framework to learn the interaction laws among these particles, the model can effectively represent scene dynamics for downstream planning and control [20, 21, 22, 23, 24, 25, 26]. However, we observe that these methods constrain particle interaction learning to the topology defined by the graph, making the model sensitive to hyperparameters such as the maximum distance for establishing particle adjacency and the maximum number of neighbors. This sensitivity becomes particularly problematic in environments involving multi-material, multi-object interactions, which feature heterogeneous and complex contact dynamics. These models are also difficult to train with real-world data, often requiring time-consuming dynamic Gaussian Splatting (GSplat) [27, 21] reconstruction to obtain 3D particle tracks, which are then used to supervise training of the particle dynamics model through an MSE loss. 2D world models bypass these issues by directly learning task-agnostic representations based on images, capturing dynamics within unified compact latent embeddings directly from pixel space [11, 28, 29, 1, 30]. However, these methods typically encode the 2D appearance of the scene into a single latent vector, which can fail to capture underlying 3D environmental structure, limiting their applicability to compositional task scenes and fine-grained dynamics modeling.

To address these limitations, we present **ParticleFormer**, a Transformer-based [31] 3D point cloud world model for multi-material robot manipulation, which achieves state-of-the-art performance in dynamics prediction and downstream visuomotor control tasks. Specifically, `ParticleFormer`: (1) introduces a Transformer-based neural dynamics backbone that enables the model to automatically infer detailed 3D particle interactions among environment particles without manual hyperparameter tuning; and (2) leverages a hybrid point cloud loss that jointly optimizes the Chamfer Distance and a differentiable approximation of the Hausdorff Distance [32] to capture both local accuracy and global structural consistency. Crucially, the loss is applied directly to point cloud sensor streams thereby avoiding costly dynamic GSplat scene reconstruction, while providing fine-grained supervision for accurate and robust dynamics modeling, even in multi-material, multi-object scenarios. Our simple yet effective framework consistently outperforms existing baselines across both synthetic and real-world benchmarks. Owing to the high quality of the dynamics predictions produced by `ParticleFormer`, we can directly apply model predictive control (MPC) [33, 34] to achieve goal-directed behavior during evaluation.

Our key contributions are threefold. (i) First, we introduce a Transformer-based neural dynamics backbone to forecast the motion of point clouds representing the scene geometry. This improves both the accuracy and robustness of the dynamics prediction process. (ii) Second, we supervise the Transformer world model with Chamfer distance and Hausdorff distance loss terms applied directly to RGB sensor streams. This loss jointly captures local geometric accuracy and global structural

consistency without requiring dense correspondence or explicit 3D reconstruction. (iii) Finally, we extend the current 3D dynamics modeling benchmarks to include multi-material, multi-object robot interactions, broadening the applicability to more diverse and practical scenarios. `ParticleFormer` is experimentally evaluated on six simulation tasks and three real-world experiments, where results on both dynamics prediction accuracy and downstream visuomotor control demonstrate that our method consistently outperforms competitive baselines across all settings.

2 Related work

World Model for Robotic Manipulation. Dynamics models are increasingly essential in robotics for predicting environment evolution from current states and actions [17, 35, 36, 37, 38]. Broadly speaking, there are two main lines of work:

- **Object-centric Modeling:** This approach represents objects as particles or meshes and uses GNNs to model local interactions among neighboring units [23, 21, 39, 40, 25, 41]. While effective in capturing spatial relations, its performance heavily depends on hyperparameters such as adjacency size and connection thresholds, limiting scalability to complex scenes.
- **Scene-centric Modeling:** This paradigm avoids explicit topology construction by directly learning latent dynamics from pixel-based observations. An encoder maps input frames to a unified latent space, where a dynamics model predicts future latent states [28, 11, 42, 1, 43]. However, compressing the entire scene into a single vector often overlooks structural details, impairing generalization in compositional or fine-grained settings.

Unlike prior approaches, `ParticleFormer` learns environment dynamics in an object-centric manner, preserving structural details while addressing the limitations of graph-based methods. It achieves more accurate predictions and removes the need for extensive manual hyperparameter tuning, resulting in more robust and generalizable dynamics modeling.

Multi-Material Robot-Object Interactions. Prior works have explored the application of learned world models to a variety of material types, including rigid bodies [44, 45, 46], deformable fabrics [37, 47, 48, 49, 50, 51], plasticine [52, 41], fluid [53] and granular materials [54, 21]. However, most of these approaches focus on single-material robot-object scenarios and overlook multi-material interactions, limiting their applicability in real-world settings. In contrast, our work explicitly addresses this gap by extending the benchmark to multi-material robot-object interactions, requiring world models to handle heterogeneous physical properties within a unified framework. Our experiments show that `ParticleFormer` effectively generalizes in this setting, outperforming existing baselines across diverse material compositions.

3 Method

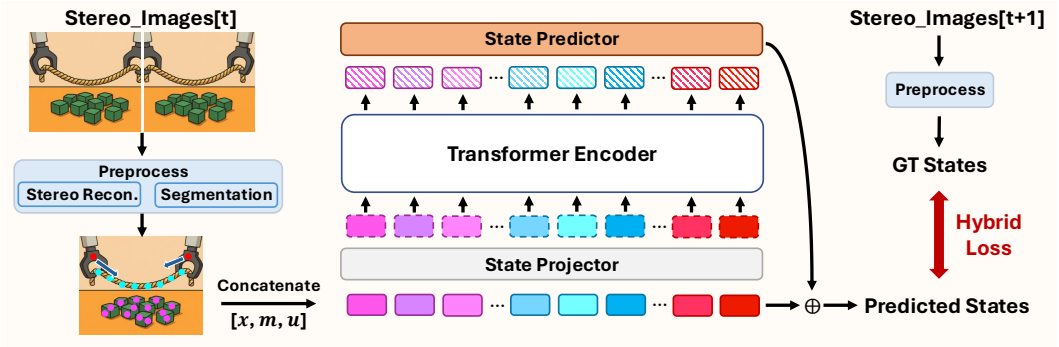

Figure 2: **Overview.** `ParticleFormer` reconstructs particle-level states from stereo image inputs via stereo matching and segmentation. A Transformer encoder models interaction-aware dynamics over particle features concatenating position, material, and motion cues. The model is trained using a hybrid loss computed against future ground-truth states extracted from the next stereo frames.

3.1 Problem Formulation

We consider a robotic manipulation setting where the environment is partially observed at each timestep t as a task-relevant point cloud: $x_t^{\text{obj}} \in \mathbb{R}^{N\times 3}$ for the object state and $x_t^{\text{ee}} \in \mathbb{R}^{M\times 3}$ for the end-effector state, both extracted from the full scene observation. Here, N is the number of downsampled object particles, and M is the number of end-effector points representing its position and shape. Our objective is to learn a neural world model f that predicts the environment's evolution in response to robot actions, with the goal of minimizing the accumulated prediction error over time.

The model takes as input the current environment state $x_t = \{x_t^{\text{obj}}, x_t^{\text{ee}}\}$, the externally applied motion $u_t = \{u_t^{\text{obj}}, u_t^{\text{ee}}\}$, and a material encoding $M = \{m(i)\}$. Specifically, $u_t^{\text{obj}} = 0$ for object particles, and $u_t^{\text{ee}} = x_t^{\text{ee}} - x_{t-1}^{\text{ee}}$ for the end-effector, computed via forward kinematics. The material encoding M contains one-hot vectors, where each $m(i)$ represents the material type of the i-th particle (e.g., rigid, granular, rope, or cloth), and $m(i) = 0$ if the i-th particle belongs to the end-effector. The world model then predicts the next state as:

$$x_{t+1} = f(x_t, u_t, M). \tag{1}$$

Conditioning on material type M allows the model to capture material-dependent physical properties and behaviors, which are essential for accurately modeling multi-material interactions.

3.2 Transformer-Based Neural Dynamics Learning

Our proposed dynamics modeling framework (as shown in Figure 2) is designed to learn particle-level interactions in physical systems using a Transformer-based architecture [31] that incorporates multi-head self-attention. At each timestep t, the architecture comprises three main components: (i) observation embedding, (ii) dynamics transition, and (iii) motion prediction.

Observation Embedding. This component extracts the state of task-relevant objects and converts pixel-based perception into feature embeddings for downstream dynamics learning. Given a pair of stereo images from the task scene, we first apply FoundationStereo [55] to reconstruct a dense point cloud. Then, we use GroundingDINO [56] and Segment Anything [57] to segment the specific object of interest being manipulated, yielding a clean point cloud x_t^{obj} that excludes irrelevant objects. For each particle i at time t, we concatenate its position $x_t(i)$, material encoding $m(i)$, and motion $u_t(i)$, and encode the combined feature using an projector f_{proj} to obtain its latent representation $z_t(i)$:

$$z_t(i) = f_{\text{proj}}([x_t(i), m(i), u_t(i)]). \tag{2}$$

Dynamics Transition. At each timestep t, we obtain particle-wise embeddings $z_t \in \mathbb{R}^{(N+M)\times d}$, where d is the embedding dimension and $N+M$ denotes the total number of object and end-effector particles. Since the environment is parameterized as a set of particles, learning dynamics reduces to modeling their pairwise interactions. To capture these interactions, we apply a stack of $L = 3$ multi-head self-attention layers, implemented as a dynamics transition module f_{dyn}. The particle embeddings are updated by applying f_{dyn} to the entire set of latent embeddings:

$$z'_{t+1} = f_{\text{dyn}}(z_t), \tag{3}$$

where $z'_{t+1} \in \mathbb{R}^{(N+M)\times d}$ denotes the predicted latent representation at the next timestep. The module f_{dyn} consists of multi-head Transformer encoder layers that allow each particle to attend to all others and infer interaction-aware dynamics. Note that z_t already incorporates spatial and motion information, so no additional positional encodings are needed. The attention mechanism directly operates on these embeddings to model physical interactions across particles. In contrast to prior methods that rely on predefined interaction graphs or distance-based adjacency thresholds, our approach learns interaction structures implicitly through attention, reducing sensitivity to such hyperparameters and enabling more flexible modeling in complex, multi-material environments.

Motion Prediction. After the dynamics transition, we obtain updated latent embeddings $z'_{t+1} \in \mathbb{R}^{(N+M)\times d}$, which encode the predicted interaction-aware state at the next timestep. To recover

particle-level motion, we apply a shared decoder f_{pred} to each latent embedding to predict the displacement of each particle:

$$\Delta\hat{x}_{t+1}(i) = f_{\text{pred}}(z'_{t+1}(i)), \quad \text{for } i = 1, \ldots, N + M, \tag{4}$$

where $\Delta\hat{x}_{t+1}(i) \in \mathbb{R}^3$ denotes the predicted positional change of the i-th particle in 3D space.

The predicted next-step particle position is then computed as:

$$\hat{x}_{t+1}(i) = x_t(i) + \Delta\hat{x}_{t+1}(i). \tag{5}$$

This decoding scheme ensures that the predicted dynamics remain physically grounded in particle-level motion, while maintaining modularity between interaction modeling and spatial state recovery.

3.3 Hybrid Supervision

To supervise dynamics learning, we propose a hybrid geometric loss that combines Chamfer Distance (CD) and a differentiable approximation of Hausdorff Distance (HD) [32]:

$$\mathcal{L}_{\text{hybrid}} = \alpha \cdot \mathcal{L}_{\text{CD}} + (1 - \alpha) \cdot \mathcal{L}_{\text{HD}}, \tag{6}$$

where $\alpha \in [0, 1]$ balances fine-grained local alignment and global structural coverage.

Chamfer Distance provides local supervision by minimizing the average nearest-neighbor distance between predicted and ground-truth particles. However, in multi-material scenes with sparse or uneven motion, CD tends to bias the learning toward regions with prominent motion, often neglecting particles affected indirectly through contacts or deformations.

Hausdorff Distance complements this by penalizing large worst-case deviations, helping the model account for subtle but semantically meaningful motions. This hybrid formulation encourages accurate predictions across both dominant and passive regions without requiring point-level correspondences, offering a practical balance between precision and scalability.

Training. We train the world model using the hybrid loss described in Section 3.3, applied over a rollout horizon of k future steps. At each training iteration, we randomly sample a time step t from the dataset as the rollout starting frame, and autoregressively unroll the model for k steps using its own predictions as inputs. In all experiments, we set $k = 5$ to balance long-term supervision and computational efficiency.

3.4 Model-Based Planning and Control

The learned world model f is deployed for downstream robotic manipulation via model-based control. Given the state space $\mathbb{X}$, action space $\mathbb{U}$, and a cost function $c : \mathbb{X} \times \mathbb{X} \times \mathbb{U} \to \mathbb{R}$, our objective is to optimize a sequence of actions that minimizes the total cost with respect to the current object state x_t^{obj} and a target state $x_{\text{target}}^{\text{obj}}$. At each control step, we sample action sequences $\{u_i\}_{i=1}^{H}$ over a predefined horizon H, roll out predicted trajectories using the world model f, and apply the Model Predictive Path Integral (MPPI) algorithm [34] to find the sequence with the lowest expected cost. The cost function includes a task-specific term measuring the distance to the target, as well as penalty terms for infeasible actions and collisions.

4 Experiment

We empirically evaluate `ParticleFormer` to address the following questions: (1) Can it generate visually plausible and physically coherent dynamics, especially in complex multi-material interactions? (2) How does its prediction accuracy compare to existing dynamics modeling baselines? (3) To what extent does it enhance visuomotor control performance in downstream manipulation tasks?

4.1 Experiment Setup and Task Design

We extend existing dynamics learning benchmarks and evaluate our method on six simulation tasks (using Nvidia FleX [58]) and three real-world tasks, with extensive coverage of multi-material,

multi-object interactions. Below, we introduce the real-world task designs. Each task involves one or two UFACTORY xArm-6 robot arms and a ZED-2i stereo camera. We collect training data by capturing stereo RGB images and recording end-effector poses at 10 FPS during random robot-object interactions. Details of the experimental setup and benchmarks are provided in Appendix A.

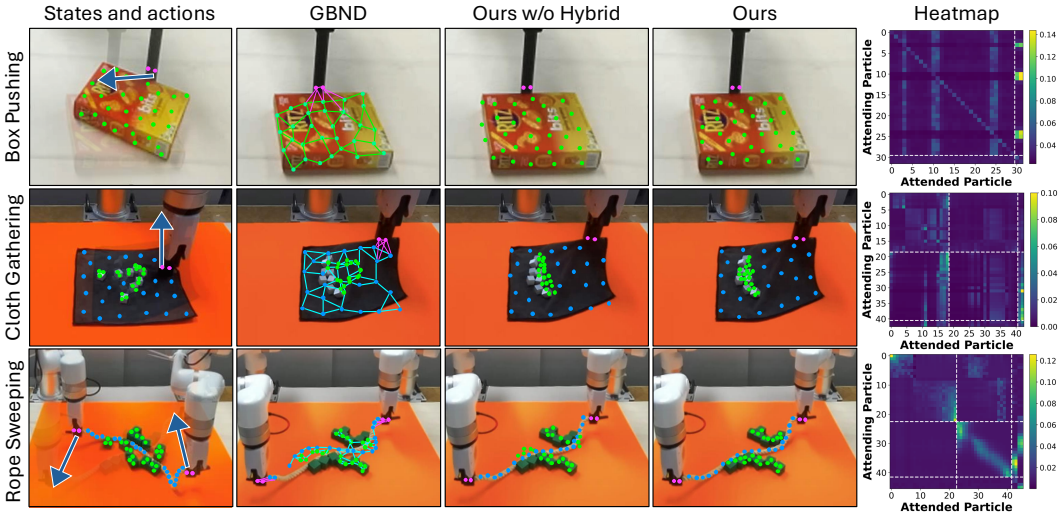

Figure 3: **Qualitative Results for Dynamics Prediction.** We compare one-step dynamics predictions from `ParticleFormer` and baseline methods. `ParticleFormer` demonstrates superior capability in capturing object dynamics and multi-material interactions. The rightmost column shows block-wise attention heatmaps from our method (attention weights from particle i(row) to particle j (column)), revealing the learned interaction structures across both intra- and inter-material particles.

Box Pushing. In this task, the robot end-effector moves horizontally along the table surface, equipped with a cylindrical pusher at its tip. The goal is to push a rigid box continuously toward a desired pose. Successfully completing this task requires `ParticleFormer` to capture accurate rigid-body dynamics, including both translational and rotational motion, to align the pushing direction with the box's inertial response.

Cloth Gathering. The robot is tasked with gathering granular particles scattered on a cloth. It samples grasping points and vertically lifts the cloth using a two-finger gripper, causing the granular objects to fall and slide due to the induced motion of the fabric. This task demands `ParticleFormer` to model both the deformable dynamics of the cloth and the passive flow of granular media, enabling accurate prediction of granular motion patterns under localized fabric manipulation.

Rope Sweeping. This task involves using two robot arms to sweep scattered granular particles into a designated area with a nylon rope held at both ends. The model is only trained on scenes where the rope is surrounded by granular objects. However, during testing, we place the rope on one side and the granular objects on the other, and ask the model to perform model-based planning and control in this unseen configuration. The robots must coordinate to manipulate the shape and trajectory of the rope to surround and sweep granular material. Success in this task requires `ParticleFormer` to capture fine-grained rope dynamics and indirect contact-based interactions.

Baselines. To demonstrate the effectiveness of `ParticleFormer`, we compare against three baselines in our main experiments: (1) *GBND*, a particle-based GNN world model following Zhang et al. [21], retrained with hybrid loss; (2) *Ours w/o Hybrid*, an ablated version of `ParticleFormer` trained using only Chamfer Distance supervision; (3) *DINO-WM* [11], a scene-centric world model that builds world models directly from 2D pixel observations using a pretrained vision encoder.

4.2 Qualitative Evaluation on Dynamics Prediction

We first evaluate whether `ParticleFormer` produces visually accurate and physically plausible dynamics predictions. Figure 3 compares one-step predictions from `ParticleFormer`, *GBND*, and *Ours w/o Hybrid* across three real-world tasks (blue arrows indicate the robot's motion direction).

Table 1: **Quantitative Results for Dynamics Prediction.** We report prediction errors across three multi-material simulation tasks (please refer to Appendix A for detailed task descriptions), where ground-truth object state information is available for computing evaluation metrics. Since *DINO-WM* does not provide point-to-point particle correspondences, we evaluate it using point cloud-based metrics only. Results for additional tasks are provided in Appendix B.

Metric	Method	Cloth Gathering	Rope Maneuver	Rope Sweeping
MSE ↓	Ours	**0.0023**	**0.0013**	**0.0008**
	Ours w/o Hybrid	0.0042	0.0037	0.0013
	GBND [21]	0.0076	0.0069	0.0029
CD ↓	Ours	0.096	**0.071**	0.067
	Ours w/o Hybrid	**0.082**	0.074	**0.061**
	GBND [21]	0.152	0.183	0.107
	DINO-WM [11]	0.187	0.233	0.126
CD+HD ↓	Ours	**0.484**	**0.216**	**0.328**
	Ours w/o Hybrid	0.749	0.673	0.415
	GBND [21]	0.886	0.628	0.473
	DINO-WM [11]	1.053	0.937	0.798

In the **Box Pushing** task, `ParticleFormer` captures both translational and rotational dynamics, while *GBND* produces unrealistic local distortions due to its limited receptive field, and *Ours w/o Hybrid* struggles to align orientation. In the **Cloth Gathering** task, `ParticleFormer` accurately models the chain of interaction from cloth lifting to granular flow, whereas *GBND* fails to propagate forces across the cloth, and *Ours w/o Hybrid* underestimates the resulting granular movement. In the **Rope Sweeping** task, `ParticleFormer` correctly predicts both rope deformation and induced granular displacement, while *GBND* fails to propagate motion through the rope, resulting in a split rope graph and unrealistic prediction; *Ours w/o Hybrid* again underpredicts the motion of indirectly affected granular media, underscoring the value of hybrid supervision.

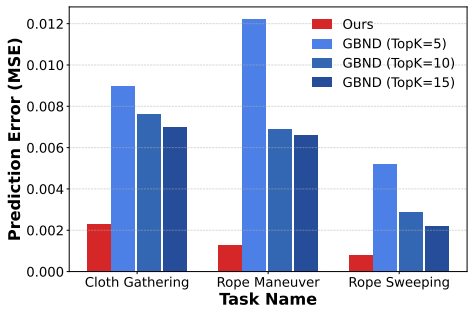

Figure 4: **Effect of `TopK` on *GBND* Dynamics Accuracy.** Increasing the number of allowed neighbors improves *GBND*'s prediction accuracy but still falls short of `ParticleFormer`, which achieves lower error without requiring hyperparameter tuning.

The final column shows attention heatmaps from `ParticleFormer`. For instance, in the Rope Sweeping task, distinct structures emerge: rope-rope interactions form a narrow linear band along the diagonal, consistent with the rope's sequential connectivity; granular-granular interactions appear as two broader blocks, reflecting spatially separated granular clusters; strong granular-to-rope attention highlights the force transfer from the rope to the granular particles; and rope particles attend asymmetrically to different robot end-effectors at either end. These patterns indicate that our method captures fine-grained material interactions directly from data without predefined topologies.

4.3 Quantitative Evaluation on Dynamics Prediction

The quantitative results in Table 1 show that `ParticleFormer` consistently achieves the best performance across all metrics, particularly in MSE and CD+HD, indicating superior accuracy in both local details and global structure. To enable a fair comparison with *DINO-WM*, which produces only 2D scene predictions rather than 3D particle states, we additionally use Depth Anything V2 [59] to reconstruct particle point clouds from its predicted images in order to compute 3D metrics.

Additionally, *GBND* exhibits strong sensitivity to hyperparameter settings. Specifically, we evaluate its dynamics prediction performance under varying values of a crucial variable: the maximum

number of allowed adjacent nodes (i.e., `TopK`), as shown in Figure 4. As this parameter increases from 5 to 15, the prediction error decreases significantly. However, the computational cost also increase rapidly (Appendix B), due to the need to store and process larger adjacency matrices—making it impractical to increase this value indefinitely. In contrast, `ParticleFormer` leverages the soft-attention mechanism through its Transformer-based architecture, which eliminates the need for manually tuning such graph-related hyperparameters. It achieves lower prediction error with less cost, offering benefits in both performance and scalability.

4.4 Model-Based Planning and Control

Finally, we evaluate whether the learned world model enables effective downstream robot manipulation via model-based control. We deploy `ParticleFormer` with the MPPI [34] algorithm to achieve goal-directed behaviors toward novel target configurations unseen during training, using the normalized CD+HD as the cost function for planning. Figure 5 presents both qualitative and quantitative results across the three real-world tasks introduced in Section 4.1. Compared to baselines, `ParticleFormer` demonstrates superior capability in leveraging accurate dynamics prediction, consistently achieving lower final-state errors and producing more fine-grained manipulation behaviors, even in complex multi-material scenarios. Please refer to Appendix C for more results.

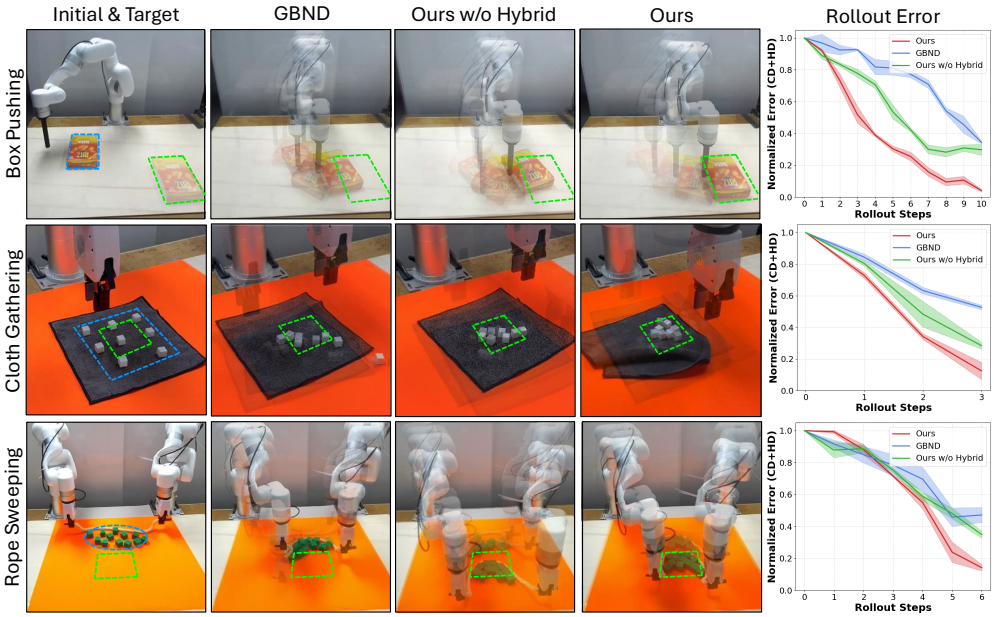

Figure 5: **Experimental Results for MPC Rollout.** The robot is tasked with using the learned world model to perform closed-loop feedback control toward novel target states unseen during training. The initial state is indicated by blue dashed boxes, and the target state is indicated by green dashed boxes. Compared to baselines, `ParticleFormer` achieves more accurate planning and control, exhibiting the lowest final-state mismatch over three rollout trials.

5 Conclusion

In this paper, we present `ParticleFormer`, a state-of-the-art world model that achieves superior performance in challenging robotic tasks. `ParticleFormer` improves dynamics modeling accuracy and robustness through two key designs: a Transformer-based particle-parametrization framework that flexibly models soft interaction dependencies among environmental particles, and a hybrid point cloud supervision loss that captures both global structure and local motion details. Furthermore, we extend evaluation benchmarks from single-material to multi-material scenarios and demonstrate the effectiveness of `ParticleFormer` across both synthetic and real-world settings. We believe `ParticleFormer` provides a capable and easy-to-use dynamics modeling framework, with potential to push the boundaries of world model applications in robotic manipulation tasks.

Limitations. While `ParticleFormer` has demonstrated its efficacy, it is currently trained per scene and has not yet been shown to generalize across a wide variety of environments and robots. In addition, the pipeline depends on externally generated object masks obtained with off-the-shelf foundation models; segmentation failures can propagate to the dynamics prediction. Future work could explore training a single, scene-agnostic model on broader 3D robotic datasets and embedding semantic codes directly into the world model without explicit masking.

Acknowledgment. This work was supported in part by the U.S. National Science Foundation under Grant No. FRR-2342246.

References

[1] D. Hafner, J. Pasukonis, J. Ba, and T. Lillicrap. Mastering diverse domains through world models. *arXiv preprint arXiv:2301.04104*, 2023.

[2] S. Huang, B. Chen, H. Xu, and V. Sitzmann. Dittogym: Learning to control soft shape-shifting robots. *arXiv preprint arXiv:2401.13231*, 2024.

[3] Z. Fu, Q. Zhao, Q. Wu, G. Wetzstein, and C. Finn. Humanplus: Humanoid shadowing and imitation from humans. *arXiv preprint arXiv:2406.10454*, 2024.

[4] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa. Amp: Adversarial motion priors for stylized physics-based character control. *ACM Transactions on Graphics (ToG)*, 40(4): 1–20, 2021.

[5] S. Huang, Z. Zhang, T. Liang, Y. Xu, Z. Kou, C. Lu, G. Xu, Z. Xue, and H. Xu. Mentor: Mixture-of-experts network with task-oriented perturbation for visual reinforcement learning. *arXiv preprint arXiv:2410.14972*, 2024.

[6] S. Lee, Y. Wang, H. Etukuru, H. J. Kim, N. M. M. Shafiullah, and L. Pinto. Behavior generation with latent actions. *arXiv preprint arXiv:2403.03181*, 2024.

[7] T. Z. Zhao, V. Kumar, S. Levine, and C. Finn. Learning fine-grained bimanual manipulation with low-cost hardware. *arXiv preprint arXiv:2304.13705*, 2023.

[8] C. Chi, Z. Xu, S. Feng, E. Cousineau, Y. Du, B. Burchfiel, R. Tedrake, and S. Song. Diffusion policy: Visuomotor policy learning via action diffusion. *The International Journal of Robotics Research*, page 02783649241273668, 2023.

[9] Z. Fu, T. Z. Zhao, and C. Finn. Mobile aloha: Learning bimanual mobile manipulation with low-cost whole-body teleoperation. *arXiv preprint arXiv:2401.02117*, 2024.

[10] M. J. Kim, K. Pertsch, S. Karamcheti, T. Xiao, A. Balakrishna, S. Nair, R. Rafailov, E. Foster, G. Lam, P. Sanketi, et al. Openvla: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[11] G. Zhou, H. Pan, Y. LeCun, and L. Pinto. Dino-wm: World models on pre-trained visual features enable zero-shot planning. *arXiv preprint arXiv:2411.04983*, 2024.

[12] H. Etukuru, N. Naka, Z. Hu, S. Lee, J. Mehu, A. Edsinger, C. Paxton, S. Chintala, L. Pinto, and N. M. M. Shafiullah. Robot utility models: General policies for zero-shot deployment in new environments. *arXiv preprint arXiv:2409.05865*, 2024.

[13] A. Brohan, N. Brown, J. Carbajal, Y. Chebotar, J. Dabis, C. Finn, K. Gopalakrishnan, K. Hausman, A. Herzog, J. Hsu, et al. Rt-1: Robotics transformer for real-world control at scale. *arXiv preprint arXiv:2212.06817*, 2022.

[14] R. Baillargeon, E. S. Spelke, and S. Wasserman. Object permanence in five-month-old infants. *Cognition*, 20(3):191–208, 1985.

[15] P. W. Battaglia, J. B. Hamrick, and J. B. Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013.

[16] S. Watzl. *Structuring mind: The nature of attention and how it shapes consciousness*. Oxford University Press, 2017.

[17] D. Ha and J. Schmidhuber. World models. *arXiv preprint arXiv:1803.10122*, 2018.

[18] R. S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. *ACM Sigart Bulletin*, 2(4):160–163, 1991.

[19] E. Todorov and W. Li. A generalized iterative lqg method for locally-optimal feedback control of constrained nonlinear stochastic systems. In *Proceedings of the 2005, American Control Conference, 2005.*, pages 300–306. IEEE, 2005.

[20] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.

[21] M. Zhang, K. Zhang, and Y. Li. Dynamic 3d gaussian tracking for graph-based neural dynamics modeling. *arXiv preprint arXiv:2410.18912*, 2024.

[22] J. Sun, L. Yu, P. Dong, B. Lu, and B. Zhou. Adversarial inverse reinforcement learning with self-attention dynamics model. *IEEE Robotics and Automation Letters*, 6(2):1880–1886, 2021.

[23] 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.

[24] H. Xue, A. Torralba, J. Tenenbaum, D. Yamins, Y. Li, and H.-Y. Tung. 3d-intphys: Towards more generalized 3d-grounded visual intuitive physics under challenging scenes. *Advances in Neural Information Processing Systems*, 36:7116–7136, 2023.

[25] 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.

[26] J. Sun, D.-A. Huang, B. Lu, Y.-H. Liu, B. Zhou, and A. Garg. Plate: Visually-grounded planning with transformers in procedural tasks. *IEEE Robotics and Automation Letters*, 7(2): 4924–4930, 2022.

[27] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023.

[28] T. Li, M. Slavcheva, M. Zollhoefer, S. Green, C. Lassner, C. Kim, T. Schmidt, S. Lovegrove, M. Goesele, R. Newcombe, et al. Neural 3d video synthesis from multi-view video. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5521–5531, 2022.

[29] T. Brooks, B. Peebles, C. Holmes, W. DePue, Y. Guo, L. Jing, D. Schnurr, J. Taylor, T. Luhman, E. Luhman, et al. Video generation models as world simulators. *OpenAI Blog*, 1:8, 2024.

[30] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine. Stochastic variational video prediction. *arXiv preprint arXiv:1710.11252*, 2017.

[31] A. Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[32] G. Charpiat, O. Faugeras, and R. Keriven. Approximations of shape metrics and application to shape warping and empirical shape statistics. *Foundations of Computational Mathematics*, 5(1):1–58, 2005.

[33] B. Kouvaritakis and M. Cannon. Model predictive control. *Switzerland: Springer International Publishing*, 38:13–56, 2016.

[34] G. Williams, A. Aldrich, and E. A. Theodorou. Model predictive path integral control: From theory to parallel computation. *Journal of Guidance, Control, and Dynamics*, 40(2):344–357, 2017.

[35] H. T. Suh and R. Tedrake. The surprising effectiveness of linear models for visual foresight in object pile manipulation. In *Algorithmic Foundations of Robotics XIV: Proceedings of the Fourteenth Workshop on the Algorithmic Foundations of Robotics 14*, pages 347–363. Springer, 2021.

[36] P. Mitrano, D. McConachie, and D. Berenson. Learning where to trust unreliable models in an unstructured world for deformable object manipulation. *Science Robotics*, 6(54):eabd8170, 2021.

[37] X. Lin, Y. Wang, Z. Huang, and D. Held. Learning visible connectivity dynamics for cloth smoothing. In *Conference on Robot Learning*, pages 256–266. PMLR, 2022.

[38] Z. Huang, X. Lin, and D. Held. Mesh-based dynamics with occlusion reasoning for cloth manipulation. *arXiv preprint arXiv:2206.02881*, 2022.

[39] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba. 3d neural scene representations for visuomotor control. In *Conference on Robot Learning*, pages 112–123. PMLR, 2022.

[40] 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 *7th Annual Conference on Robot Learning*, 2023.

[41] H. Shi, H. Xu, S. Clarke, Y. Li, and J. Wu. Robocook: Long-horizon elasto-plastic object manipulation with diverse tools. *arXiv preprint arXiv:2306.14447*, 2023.

[42] D. Hafner, T. Lillicrap, J. Ba, and M. Norouzi. Dream to control: Learning behaviors by latent imagination. *arXiv preprint arXiv:1912.01603*, 2019.

[43] V. Micheli, E. Alonso, and F. Fleuret. Transformers are sample-efficient world models. *arXiv preprint arXiv:2209.00588*, 2022.

[44] Y. Li, J. Wu, J.-Y. Zhu, J. B. Tenenbaum, A. Torralba, and R. Tedrake. Propagation networks for model-based control under partial observation. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 1205–1211. IEEE, 2019.

[45] I. Huang, Y. Narang, R. Bajcsy, F. Ramos, T. Hermans, and D. Fox. Defgraspnets: Grasp planning on 3d fields with graph neural nets. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5894–5901. IEEE, 2023.

[46] Z. Liu, G. Zhou, J. He, T. Marcucci, F.-F. Li, J. Wu, and Y. Li. Model-based control with sparse neural dynamics. *Advances in Neural Information Processing Systems*, 36:6280–6296, 2023.

[47] A. Longhini, M. Moletta, A. Reichlin, M. C. Welle, D. Held, Z. Erickson, and D. Kragic. Edo-net: Learning elastic properties of deformable objects from graph dynamics. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3875–3881. IEEE, 2023.

[48] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. Battaglia. Learning mesh-based simulation with graph networks. In *International conference on learning representations*, 2020.

[49] X. Lin, Y. Wang, J. Olkin, and D. Held. Softgym: Benchmarking deep reinforcement learning for deformable object manipulation. In *Conference on Robot Learning*, pages 432–448. PMLR, 2021.

[50] K. Puthuveetil, S. Wald, A. Pusalkar, P. Karnati, and Z. Erickson. Robust body exposure (robe): A graph-based dynamics modeling approach to manipulating blankets over people. *IEEE Robotics and Automation Letters*, 8(10):6299–6306, 2023.

[51] Z. Huang, X. Lin, and D. Held. Self-supervised cloth reconstruction via action-conditioned cloth tracking. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7111–7118. IEEE, 2023.

[52] H. Shi, H. Xu, Z. Huang, Y. Li, and J. Wu. Robocraft: Learning to see, simulate, and shape elasto-plastic objects in 3d with graph networks. *The International Journal of Robotics Research*, 43(4):533–549, 2024.

[53] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. Battaglia. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pages 8459–8468. PMLR, 2020.

[54] 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.

[55] B. Wen, M. Trepte, J. Aribido, J. Kautz, O. Gallo, and S. Birchfield. Foundationstereo: Zero-shot stereo matching. *arXiv preprint arXiv:2501.09898*, 2025.

[56] S. Liu, Z. Zeng, T. Ren, F. Li, H. Zhang, J. Yang, Q. Jiang, C. Li, J. Yang, H. Su, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. In *European Conference on Computer Vision*, pages 38–55. Springer, 2024.

[57] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[58] M. Macklin, M. Müller, N. Chentanez, and T.-Y. Kim. Unified particle physics for real-time applications. *ACM Transactions on Graphics (TOG)*, 33(4):1–12, 2014.

[59] L. Yang, B. Kang, Z. Huang, Z. Zhao, X. Xu, J. Feng, and H. Zhao. Depth anything v2. *Advances in Neural Information Processing Systems*, 37:21875–21911, 2024.

Appendix

A Experiment Implementation Details

A.1 Setup

Our experiments focus on robotic manipulation tasks where `ParticleFormer` must model the dynamics of diverse objects, including complex multi-material interactions. The tasks span both single-material and multi-material robot-object interactions, and are used to evaluate dynamics modeling accuracy as well as downstream visuomotor control performance. We run all dynamics learning experiments on an NVIDIA A100 GPU and perform real-world model-based visuomotor control on an NVIDIA RTX 3090 GPU.

Simulation. As shown in Figure 6, we extend the existing dynamics modeling benchmark created by Zhang et al. [23] based on the NVIDIA FleX simulator [58]. The original tasks, which involve only single-material robot-object interactions—Rope, Granular Object, and Cloth—are used solely for evaluating dynamics modeling performance. We additionally design three tasks involving multi-material robot-object interactions, used to evaluate both dynamics modeling and downstream visuomotor control. All tasks use the UFACTORY xArm-6 robot, and the end-effectors include a cylindrical pusher, a planar pusher, and a parallel gripper. We collect 1,000 episodes of data for each task as the training dataset.

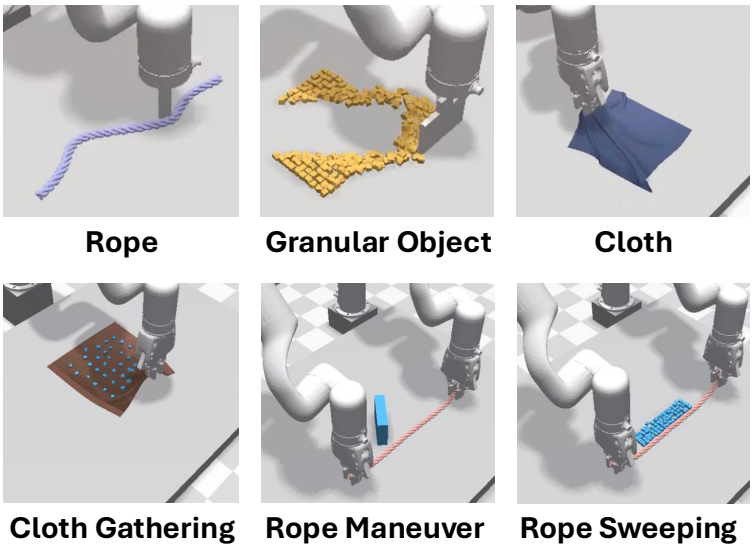

Figure 6: **Simulation Experiment Setup.**

Real World. In our real-world experiments, as shown in Figure 7, we utilize one UFACTORY xArm-6 robotic arm (for the Box Pushing and Cloth Gathering tasks) or two arms (for the Rope Sweeping task), each with 6 DoF. A 3D-printed cylindrical pusher is attached for the Box Pushing task, while a parallel gripper is used for the Cloth Gathering task and two grippers for the Rope Sweeping task. A ZED-2i camera is positioned in front of the workspace to capture stereo RGB images at 10 Hz with a resolution of 1280×720. The robot manipulation workspace measures approximately 70 cm × 55 cm. We record 500-second episodes of random robot-object interactions for each task to construct the training dataset.

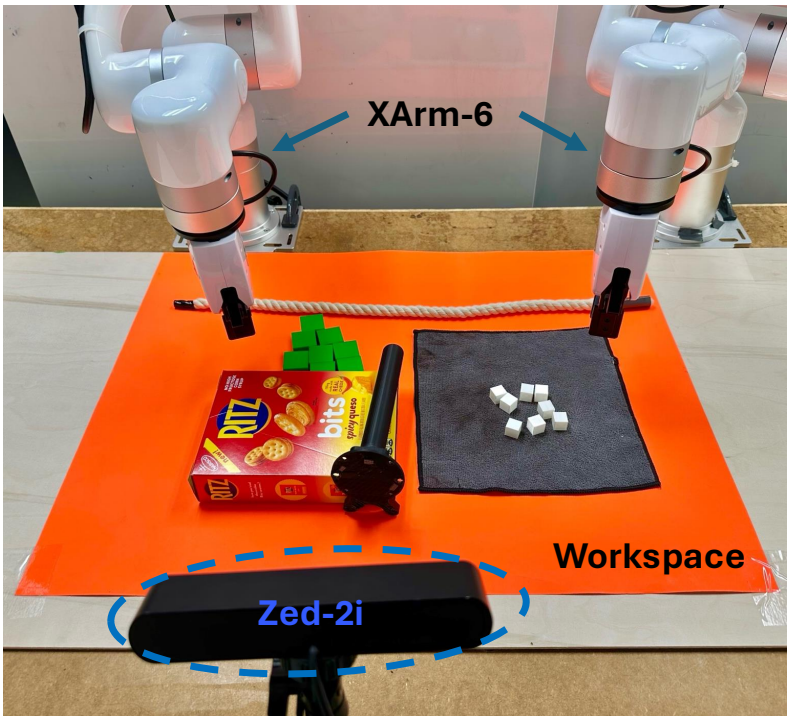

Figure 7: **Real-World Experiment Setup.**

A.2 Task Design

We introduce the tasks used to evaluate the performance of `ParticleFormer` compared to baselines, spanning both single-material and multi-material interactions.

Rope. The robot uses a cylindrical pusher to randomly push a rope placed in the planar workspace. This task evaluates whether the world model can accurately capture the deformable dynamics of the rope and its interactions with the robot.

Granular Object. The robot uses a flat pusher to randomly push piles of granular material scattered across the workspace. This task tests the model's ability to predict the dynamics of disjoint granular particles under direct physical contact.

Cloth. The robot randomly grasps the corner of a cloth using a parallel gripper and moves it along different trajectories. The goal is to evaluate the model's ability to capture the deformable dynamics of the fabric under manipulation.

Box Pushing. In this task, the robot end-effector moves horizontally along the table surface, equipped with a cylindrical pusher at its tip. The goal is to push a rigid box continuously toward a desired pose. Successfully completing this task requires `ParticleFormer` to capture accurate rigid-body dynamics, including both translational and rotational motion, to align the pushing direction with the box's inertial response.

Cloth Gathering. The robot is tasked with gathering granular particles scattered on a cloth. It samples grasping points and vertically lifts the cloth using a two-finger gripper, causing the granular objects to fall and slide due to the induced motion of the fabric. This task demands `ParticleFormer` to model both the deformable dynamics of the cloth and the passive flow of granular media, enabling accurate prediction of granular motion patterns under localized fabric manipulation.

Rope Maneuver. In this task, two robot arms grasp and manipulate the ends of a rope to indirectly move a rigid box placed on the ground. The objective is to maneuver the box to a target pose without tipping it over. This setting features rich multi-object, multi-material interactions and requires `ParticleFormer` to reason about indirect force transmission between deformable and rigid bodies.

Rope Sweeping. This task involves using two robot arms to sweep scattered granular particles into a designated area with a nylon rope held at both ends. The robots must coordinate to manipulate the shape and trajectory of the rope to surround and sweep granular material. Success in this task requires `ParticleFormer` to capture fine-grained rope dynamics and indirect contact-based interactions, as the rope deforms and transmits force to the granular particles through friction and sweeping motion.

In this paper, we evaluate `ParticleFormer` on six simulation tasks—**Rope**, **Granular Object**, **Cloth**, **Cloth Gathering (Sim)**, **Rope Maneuver**, and **Rope Sweeping (Sim)**—and three real-world tasks—**Box Pushing**, **Cloth Gathering (Real)**, and **Rope Sweeping (Real)**—to extensively assess its performance in dynamics modeling under complex scenarios, in comparison with leading baselines.

B Additional Results for Dynamics Prediction

This section provides additional results for dynamics modeling comparison in the simulation.

B.1 Dynamics Prediction Comparison

Table 2 presents the dynamics prediction performance comparison on three single-material tasks. The results demonstrate the superior performance of `ParticleFormer` in terms of both MSE and CD+HD metrics, highlighting its ability to model dynamics accurately at both the global structural level and in fine-grained local movements. Results for multi-material tasks are provided in Table 1.

Table 2: **Addition Quantitative Results for Dynamics Prediction.** We report prediction errors across the three single-material simulation tasks, where ground-truth object state information is available for computing evaluation metrics. Since *DINO-WM* does not provide point-to-point particle correspondences, we evaluate it using point cloud-based metrics only.

Metric	Method	Rope	Granular Object	Cloth
MSE ↓	Ours	**0.00017**	**0.00152**	**0.00160**
	Ours w/o Hybrid	0.00025	0.00173	0.00182
	GBND [21]	0.00043	0.00196	0.00194
CD ↓	Ours	0.016	0.062	0.063
	Ours w/o Hybrid	**0.013**	**0.056**	**0.059**
	GBND [21]	0.045	0.074	0.076
	DINO-WM [11]	0.052	0.125	0.130
CD+HD ↓	Ours	**0.149**	**0.413**	**0.488**
	Ours w/o Hybrid	0.178	0.433	0.527
	GBND [21]	0.291	0.470	0.664
	DINO-WM [11]	0.304	0.527	1.425

B.2 Hyperparameter Analysis

As a complementary result to Figure 4, we analyze the effect of the `TopK` hyperparameter in *GBND* [21] on GPU memory usage during training (All models are evaluated using input clusters of 200 scene particles). Figure 8 shows that increasing `TopK` leads to significantly higher memory consumption per epoch, reflecting the scalability challenges of graph-based dynamics models.

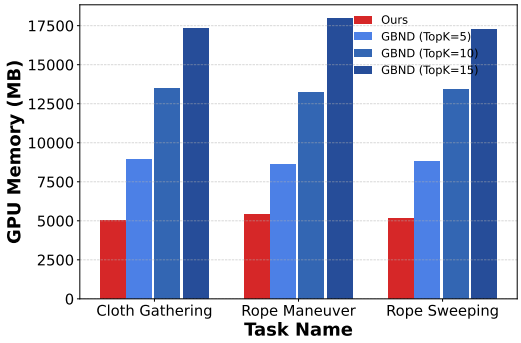

Figure 8: **Effect of `TopK` on *GBND* GPU Usage.** As the maximum number of allowed adjacent nodes increases, *GBND*'s GPU memory usage grows significantly. This highlights the scalability bottleneck in GNN-based methods. In contrast, `ParticleFormer` avoids this issue by using soft attention without explicit neighbor selection.

Additionally, we observe that *GBND* exhibits strong sensitivity to another key hyperparameter: the maximum distance threshold used to construct edges between particles, denoted as `MaxDist`. This parameter directly affects the range over which information can propagate in a single message-passing step within the graph. We conduct ablation experiments on the Cloth task to analyze its impact, as shown in Figure 9. Specifically, we compare the dynamics prediction error (MSE) under three conditions: (1) when the end-effector particle is connected to all cloth particles, (2) when only short-range connections are allowed (`MaxDist` = 0.3), and (3) when longer-range connections are allowed (`MaxDist` = 0.7). The results reveal that the dynamics accuracy of *GBND* varies significantly depending on this setting, highlighting its vulnerability to edge construction heuristics. In contrast, `ParticleFormer` leverages a Transformer encoder with soft attention to bypass these limitations, achieving more stable and accurate performance.

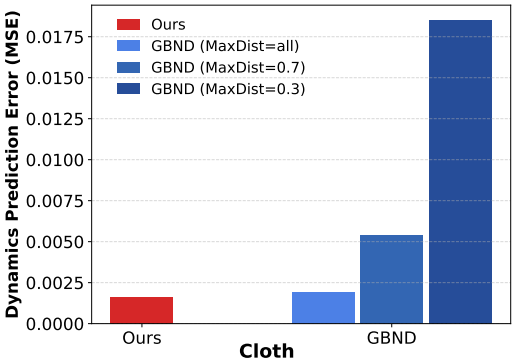

Figure 9: **Effect of `MaxDist` on *GBND* Dynamics Accuracy.** Since information propagation in *GBND* relies on graph topology, its performance is highly sensitive to the maximum distance threshold for edge construction. In contrast, `ParticleFormer` avoids this sensitivity through attention-based interactions.

C Additional Results for Model-Based Planning and Control

In this section, we provide additional results on model-based visuomotor control using world models trained with `ParticleFormer` and baseline methods across three multi-object, multi-material simulation tasks. The qualitative results are presented in Figure 10, demonstrating that `ParticleFormer` enables more fine-grained and higher-quality control behavior in downstream manipulation tasks compared to prior approaches.

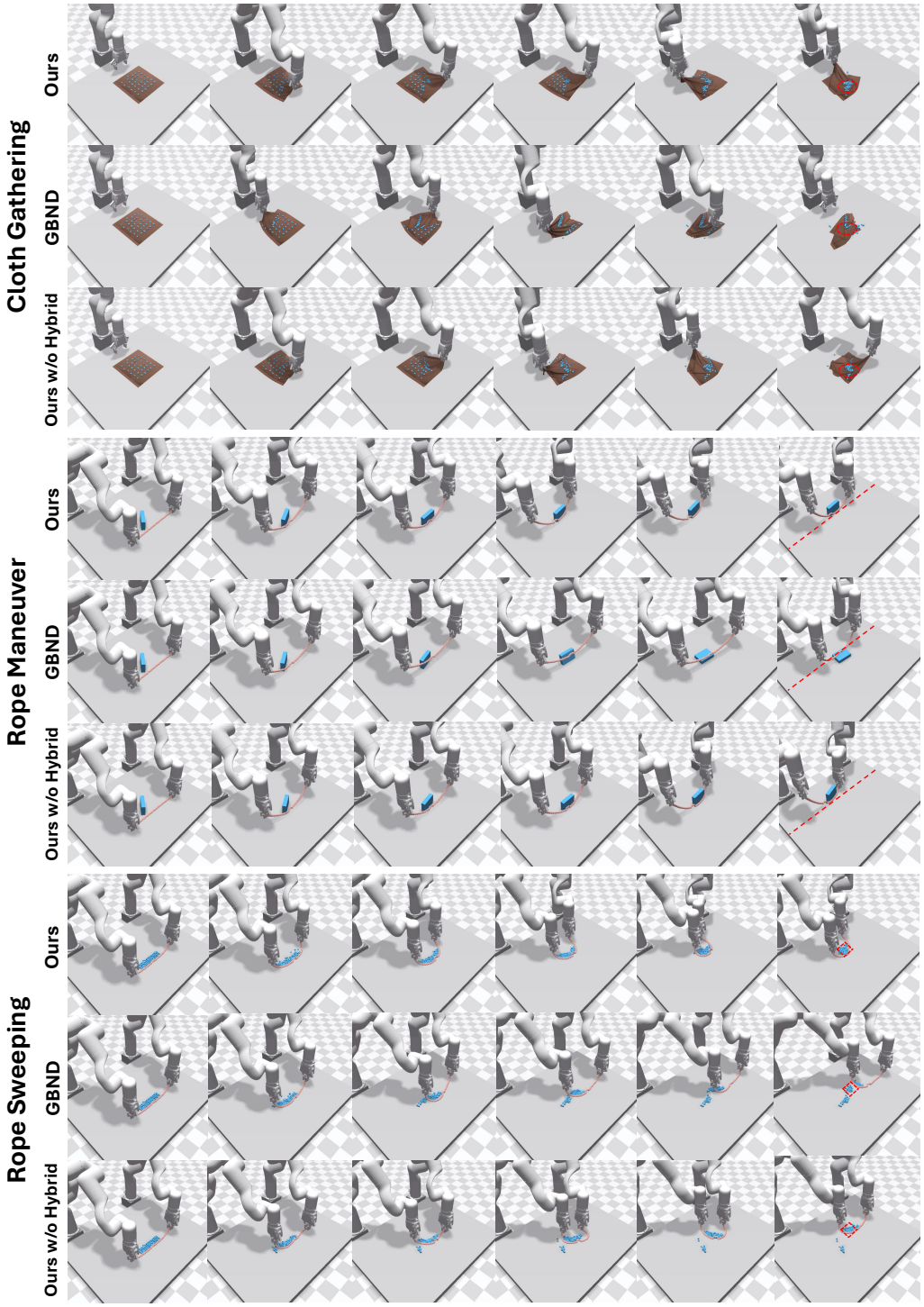

Figure 10: **MPC Rollout Results in Multi-Material Simulation Tasks.** The robot is tasked with using the learned world model to perform closed-loop feedback control toward novel target states unseen during training. Target states are indicated by red dashed boxes or lines. Compared to baselines, `ParticleFormer` achieves more accurate planning and lower final-state mismatch.