APPENDIX

A. Simulated Dataset

For the simulation experiments, we utilize the environmental setup from Ravens [35]. Each task is executed over 100 trials to calculate success rates and state performance. We set up 8 cameras for data collection.

B. Gaussian Splatting Reconstruction

For Gaussian splatting reconstruction, we follow the implementation provided in this repository. The framework is optimized using the Adam optimizer [38] with a learning rate of 0.001 over 2000 epochs. The loss function used for training Gaussian splatting is:

$$\mathscr{L} = \mathscr{L}_{L1} + 0.25 \cdot (1 - \mathscr{L}_{SSIM}) \tag{17}$$

, where $\mathscr{L}_{L1}$ and $\mathscr{L}_{SSIM}$ are L1 loss and structure similarity metrics between the reconstructed image and ground-truth image.

C. Dynamic Model Training

Our dynamic model f consists of three components: f_{enc}, f_{mp}, and f_{dec}. The encoder, f_{enc}, is composed of two SAGE layers [30] with a hidden dimension of 256 and ReLU activation functions. The message-passing module, f_{mp}, includes two SAGE layers with two recursive message-passing steps. The decoder, f_{dec}, is composed of a single SAGE layer. The dynamic model is optimized using the Adam optimizer [38] with a learning rate of 0.001, without applying any learning rate scheduler.

For the graph forming part, the distance threshold ω used was 0.1.

D. Baseline Implementation

In this section, we provide implementation details for each baseline.

Dynamic resolution [16]. We adapt the official implementation from this link. To ensure a fair comparison, we convert our dataset into their format and use the hyperparameters provided by the authors in the appendix.

NeRF-dy [37]. We implemented this approach using the source code provided by the authors. To maintain fairness, we converted our dataset into their format and applied the hyperparameters provided in their appendix.

NFD [28]. Since the official implementation is not available, we re-implemented this method based on the hyperparameters and network architecture described in the paper and supplementary materials. We verified the validity of our implementation by comparing its performance to the results reported in [28].

DVF [17]. Since the official implementation is not available, we re-implemented this method based on the hyperparameters and network architecture described in the paper and supplementary materials. We verified the validity of our implementation by comparing its performance to the results reported in [17].

E. Real-World Experiments

Franka Panda manipulator and four Intel RealSense D415. Each task is executed over 20 trials to calculate success rates.

REFERENCES

[1] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," in *ECCV*, 2020.

[2] V. Sitzmann, M. Zollhöfer, and G. Wetzstein, "Scene representation networks: Continuous 3d-structure-aware neural scene representations," in *Advances in Neural Information Processing Systems*, 2019.

[3] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] B. Kerbl, G. Kopanas, T. Leimkühler, and G. Drettakis, "3d gaussian splatting for real-time radiance field rendering," *ACM Transactions on Graphics*, vol. 42, no. 4, July 2023. [Online]. Available: https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/

[5] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[6] W.-C. Tseng, H.-J. Liao, L. Yen-Chen, and M. Sun, "Cla-nerf: Category-level articulated neural radiance field," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE Press, 2022, p. 8454–8460. [Online]. Available: https://doi.org/10.1109/ICRA46639.2022.9812272

[7] Q. Dai, Y. Zhu, Y. Geng, C. Ruan, J. Zhang, and H. Wang, "GraspNeRF: Multiview-based 6-DoF grasp detection for transparent and specular objects using generalizable NeRF," *arXiv [cs.RO]*, Oct. 2022.

[8] Z. Jiang, Y. Zhu, M. Svetlik, K. Fang, and Y. Zhu, "Synergies between affordance and geometry: 6-dof grasp detection via implicit representations," *Robotics: science and systems*, 2021.

[9] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," in *ICLR*, 2019.

[10] 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, 2017, pp. 239–248.

[11] N. Tuomainen, D. B. Mulero, and V. Kyrki, "Manipulation of granular materials by learning particle interactions," *CoRR*, vol. abs/2111.02274, 2021. [Online]. Available: https://arxiv.org/abs/2111.02274

[12] Y. Li, J. Wu, R. Tedrake, J. B. Tenenbaum, and A. Torralba, "Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids," in *ICLR*, 2019.

[13] W. F. Whitney, T. Lopez-Guevara, T. Pfaff, Y. Rubanova, T. Kipf, K. Stachenfeld, and K. R. Allen, "Learning 3d particle-based simulators from rgb-d videos," in *ICLR*, 2024.

[14] X. Li, Y. Cao, M. Li, Y. Yang, C. Schroeder, and C. Jiang, "Plasticitynet: Learning to simulate metal, sand, and snow for optimization time integration," in *Advances in Neural Information Processing Systems*, A. H. Oh, A. Agarwal, D. Belgrave, and K. Cho, Eds., 2022. [Online]. Available: https://openreview.net/forum?id=_WqHmwoE7Ud

[15] C. Jiang, C. Schroeder, J. Teran, A. Stomakhin, and A. Selle, "The material point method for simulating continuum materials," in *ACM SIGGRAPH 2016 Courses*, ser. SIGGRAPH '16. New York, NY, USA: Association for Computing Machinery, 2016. [Online]. Available: https://doi.org/10.1145/2897826.2927348

[16] Y. Wang, Y. Li, K. Driggs-Campbell, L. Fei-Fei, and J. Wu, "Dynamic-resolution model learning for object pile manipulation," in *Robotics: Science and Systems*, 2023.

[17] H. J. T. Suh and R. Tedrake, "The surprising effectiveness of linear models for visual foresight in object pile manipulation," in *Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2020.

[18] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," *arXiv preprint arXiv:1612.00593*, 2016.

[19] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *arXiv preprint arXiv:1706.02413*, 2017.

[20] M. Liu, X. Li, Z. Ling, Y. Li, and H. Su, "Frame mining: a free lunch for learning robotic manipulation from 3d point clouds," in *6th Annual Conference on Robot Learning*, 2022.

[21] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *IROS*, 2015, pp. 922–928.

[22] S. James, K. Wada, T. Laidlow, and A. J. Davison, "Coarse-to-fine q-attention: Efficient learning for visual robotic manipulation via discretisation," *CoRR*, vol. abs/2106.12534, 2021. [Online]. Available: https://arxiv.org/abs/2106.12534

[23] S. James and A. J. Davison, "Q-attention: Enabling efficient learning for vision-based robotic manipulation," *CoRR*, vol. abs/2105.14829, 2021. [Online]. Available: https://arxiv.org/abs/2105.14829

[24] W. Shen, G. Yang, A. Yu, J. Wong, L. P. Kaelbling, and P. Isola, "Distilled feature fields enable few-shot language-guided manipulation," in *7th Annual Conference on Robot Learning*, 2023.

[25] L. Yen-Chen, P. Florence, J. T. Barron, T.-Y. Lin, A. Rodriguez, and P. Isola, "NeRF-Supervision: Learning dense object descriptors from neural radiance fields," in *IEEE Conference on Robotics and Automation (ICRA)*, 2022.

[26] A. Sanchez-Gonzalez, J. Godwin, T. Pfaff, R. Ying, J. Leskovec, and P. W. Battaglia, "Learning to simulate complex physics with graph networks," *CoRR*, vol. abs/2002.09405, 2020. [Online]. Available: https://arxiv.org/abs/2002.09405

[27] K. Kumar and J. Vantassel, "Gns: A generalizable graph neural network-based simulator for particulate and fluid modeling," 2022. [Online]. Available: https://arxiv.org/abs/2211.10228

[28] S. Xue, S. Cheng, P. Kachana, and D. Xu, "Neural field dynamics model for granular object piles manipulation," in *CoRL*, 2023.

[29] T. Xie, Z. Zong, Y. Qiu, X. Li, Y. Feng, Y. Yang, and C. Jiang, "Physgaussian: Physics-integrated 3d gaussians for generative dynamics," *arXiv preprint arXiv:2311.12198*, 2023.

[30] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *CoRR*, vol. abs/1706.02216, 2017. [Online]. Available: http://arxiv.org/abs/1706.02216

[31] J. Tang, J. Ren, H. Zhou, Z. Liu, and G. Zeng, "Dreamgaussian: Generative gaussian splatting for efficient 3d content creation," *arXiv preprint arXiv:2309.16653*, 2023.

[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[33] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," in *ICLR Workshop on Representation Learning on Graphs and Manifolds*, 2019.

[34] E. Coumans and Y. Bai, "Pybullet, a python module for physics simulation for games, robotics and machine learning," http://pybullet.org, 2016–2021.

[35] A. Zeng, P. Florence, J. Tompson, S. Welker, J. Chien, M. Attarian, T. Armstrong, I. Krasin, D. Duong, V. Sindhwani, and J. Lee, "Transporter networks: Rearranging the visual world for robotic manipulation," *Conference on Robot Learning (CoRL)*, 2020.

[36] K. Zhang, M. Sharma, J. Liang, and O. Kroemer, "A modular robotic arm control stack for research: Franka-interface and frankapy," *arXiv preprint arXiv:2011.02398*, 2020.

[37] Y. Li, S. Li, V. Sitzmann, P. Agrawal, and A. Torralba, "3d neural scene representations for visuomotor control," in *5th Annual Conference on Robot Learning*, 2021. [Online]. Available: https://openreview.net/forum?id=zv3NYgRZ7Qo

[38] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: https://api.semanticscholar.org/CorpusID:6628106