
Graph Network Simulators and its Applications

Shizheng Wen¹

Abstract

Graph neural network (GNN)-based simulators have gained significant attention in science and engineering due to their computational efficiency and physical differentiability compared to traditional numerical simulators. In this paper, we provide a review of the development for GNN-based simulators, starting from the initial interaction network models to the recent model improvements for MeshGraphNet. Furthermore, we explore the promising applications of GNN-based simulators in robotics control and inverse design tasks. We believe that graph network simulators hold great promise in addressing long-standing challenges in scientific computing. [Repository](#).

1. Introduction

Data-driven surrogate models have gained significant interest within the machine learning community as a means to overcome challenges encountered by conventional numerical simulators in various fields such as robotics (Li et al., 2018), aerospace (Brunton et al., 2020) and computer graphics (Prantl et al., 2022). Compared with traditional simulators, machine learning surrogate models offer a promising solution including higher inference speeds (Lam et al., 2022; Pathak et al., 2022), superior generalization capability (Li et al., 2018), physical differentiability (Li et al., 2018; Allen et al., 2022a) and etc. When utilizing a machine learning model to approximate a simulator, there is often uncertainty regarding whether the model truly learns the underlying physics or merely fits and interpolates a curve. If the model only captures the curve, it will struggle with extrapolation and handling domain shifts in the input distribution. Given the impossibility of building a dataset that covers the entire input space of a general-purpose simulator, it becomes essential to introduce inductive biases into the learning setup (Battaglia et al., 2018) for constructing

more generalizable model. These biases enable the learning algorithm to prioritize specific solutions with better physical interpretations. Graph Neural Networks (GNNs) based on message-passing framework emerge as a promising choice due to its permutation equivariance, pairwise message, local interaction and superposition principle for learning the genuine physics—reusable knowledge akin to physical laws—and can be implemented beyond the training data (Battaglia et al., 2016).

This paper provides a retrospective analysis of the advancements made in GNN-based simulators in recent years. It also explores the application of GNN surrogate solvers in various downstream tasks, such as control and inverse design. By examining the progress of GNN-based simulators and their utilization as surrogate solvers, this paper sheds light on their significance and potential impact in the field.

2. GNN-based simulators

2.1. Model Architecture

The prevailing framework in this field is the next-step prediction model (Wu, 2022), also known as an autoregressive model, as depicted in Fig. 1.

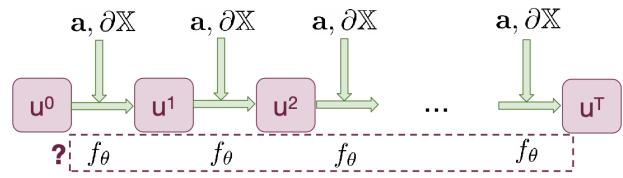


Figure 1. Architecture of autoregressive GNN model. u^t indicates state of the system at time t , which can be represented as a graph. \mathbf{a} and $\partial \mathbb{X}$ suggest parameters and boundary condition of the system, respectively. f_θ represents the graph neural network with trainable parameters θ .

Among various GNN architectures, the message passing framework (Gilmer et al., 2017) has gained widespread adoption due to its utilization of a shared update function and superposition for nodes and edges, resembling the universal local interactions in physics. The core concept underlying the message passing framework involves aggregating all messages from neighboring nodes, as illustrated in Fig. 2.

¹Department of Mathematics, ETH Zurich, Zurich, Switzerland.
Correspondence to: Shizheng Wen <shiwenzw@student.ethz.ch>.

The update function for edge and node can be modeled by the multilayer perceptron (MLP). Through multiple layers of message passing, the state of each node within the graph can be effectively updated.

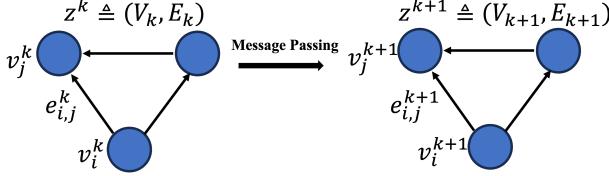


Figure 2. The process of learned message passing over the latent graph z^k , composed of vertex features V_k and edge features E_k . These features can be updated by edge function $e_{ij}^k = f_{Edge}(v_j^k, j_i)$ and node function $v_j^{k+1} = f_{Node}(v_j^k, \sum_{k \in N(v_j^k)} e^k)$

Several notable studies have employed this framework for graph-based simulation. Battaglia et al. (Battaglia et al., 2016) firstly introduced a general-purpose learnable physics engine called the interaction network, which enables reasoning about object interactions in complex systems such as n-body problems, rigid-body collisions, and non-rigid dynamics. Chang et al. (Chang et al., 2016) proposed a neural physics engine (NPE) for learning simulators of intuitive physics, demonstrating its generality across varying object counts and scene configurations. Sanchez-Gonzalez et al. (Sanchez-Gonzalez et al., 2018) not only built a GNN simulator for the Mujoco system but also used it for downstream tasks such as system identification and control. These works represent scenes as graphs and model interactions using GNN, shown in Fig. 3. However, they focused on relatively small domains with no more than 15 bodies and 100 time steps. The scalability of graph net simulators to real-world complexity with a large number of particles and time steps remains an open problem.

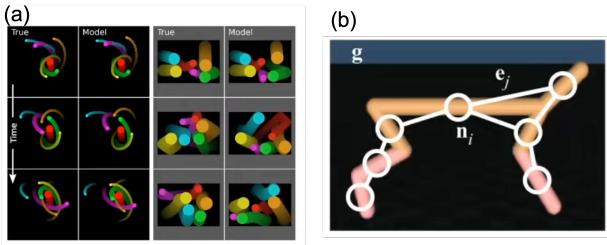


Figure 3. Graph representation and GN-based models. (a) n-body prediction based on GNN (Battaglia et al., 2016). (b) A physical system's bodies and joints can be represented by a graph's nodes and edges, respectively (Sanchez-Gonzalez et al., 2018).

In 2019, Li et al. (Li et al., 2018) studied larger systems and achieved closed-loop feedback control using a learned GNN model. To account for potential inaccuracies in the model, they applied model-predictive control (MPC) optimized using gradient descent. Their real-world experiments successfully demonstrated the adaptability of the learned model and control algorithm to environments with unknown physical parameters, manipulating a deformable foam into various target shapes. Subsequently, researchers from DeepMind developed the high-quality general-purpose simulator (GNS) (Sanchez-Gonzalez et al., 2020) capable of simulating fluid dynamics, rigid solids, and deformable materials, shown in Fig. 4. The simulated systems consisted of up to 85k particles and remained stable for thousands of time steps. Their work also showcased strong generalization capabilities across larger domains, varying initial conditions, and unseen data. However, the stability of elastic materials posed limitations for GNS. While the particle-based model constructed the graph based on proximity criteria, systems without well-established particle representations raised questions about learning from simulations with connectivity, such as finite element method (FEM). This served as the motivation behind MeshGraphNet (Pfaff et al., 2020), proposed by DeepMind in 2021, where the connectivity is already built into the dataset. This work also introduced dual space message passing in order to pass information between nodes that are spatially close, but distant in mesh space. In comparison to GNS, MeshGraphNet exhibits stable error accumulation for elastic materials (e.g., flagsimple) without simulation explosions.

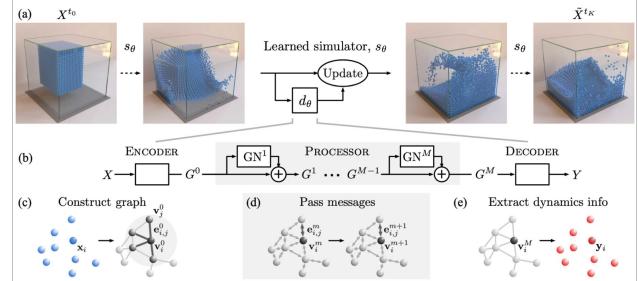


Figure 4. Architecture of the graph network-based simulators (GNS). Graph is constructed from particles based on proximity criteria (Sanchez-Gonzalez et al., 2020).

GNS and MeshGraphNet serve as the cornerstones of the graph-based simulator field. Subsequently, the community has primarily focused on improving models, addressing error accumulation, enabling large-scale simulations, achieving higher resolution, and incorporating more physics priors into the GNN framework.

2.2. Model Improvement

2.2.1. CONTROL THE ERROR ACCUMULATION

One key challenge of the autoregressive model lies in the potential for simulation explosions caused by error accumulation. This issue can be attributed to overfitting to the one-step training distribution, which leads to poor generalization when the input deviates from it, as often occurs after a few rollout steps. In GNS and MeshGraphNet, the authors addressed this problem by introducing noise into the training data, enabling better learning of distribution shifts. However, this approach is not always stable and may encounter breakdowns in certain cases. To develop stable rollouts, two main techniques have been proposed. Firstly, an encoder-decoder structure can be designed to incorporate temporal modeling and capture long-term dependencies. In Fig. 5(a), Han et al. (Han et al., 2022b) introduced an attention-based sequence model that summarizes features and prioritizes error accumulation in fluid dynamics prediction tasks. Equer et al. (Equer et al., 2023) introduced the long-expressive memory (LEM) and gating mechanism to learn multi-scale features in time and space. Secondly, different training strategies can be employed. Rather than minimizing the Mean Squared Error (MSE) on one-step predictions, Wu et al. (Wu et al., 2022) developed multi-step rollout objectives that minimize the MSE on predictions spanning multiple steps, as shown in Fig. 5(b). DeepMind researchers further introduced an annealed multi-step learning objective (Lam et al., 2022), successfully applying it to large-scale weather forecasting. Brandstetter et al. (Brandstetter et al., 2022) designed the adversarial-style loss, which predicts N steps into the future, but only backprops on the last step to mitigate the distribution shift problem during the iteration. Also, they applied the temporal bundling techniques for reducing the number of solver calls such that reduce the error propagation speed.

2.2.2. DEPLOY IN SUPER LARGE-SCALE SYSTEM

Another promising avenue is the exploration of learning super large-scale simulations using graph network simulators. However, two key challenges arise: graph storage and efficient message passing. When dealing with extremely large graphs, it becomes infeasible to fit the entire graph within a single GPU, necessitating the use of distributed computing. Additionally, propagating information efficiently between distant nodes in the graph poses a significant challenge. To address the first challenge, Wu et al. (Wu et al., 2022) introduced sector-based training, where the full grid is partitioned into overlapping sectors (subgraphs), and random combinations of sectors are used as training examples. This approach allowed them to successfully train realistic 3D scenarios of surface fluid flow with up to 1.1 million cells per time step, surpassing prior models by two orders of magnitude. For the second challenge, DeepMind researchers intro-

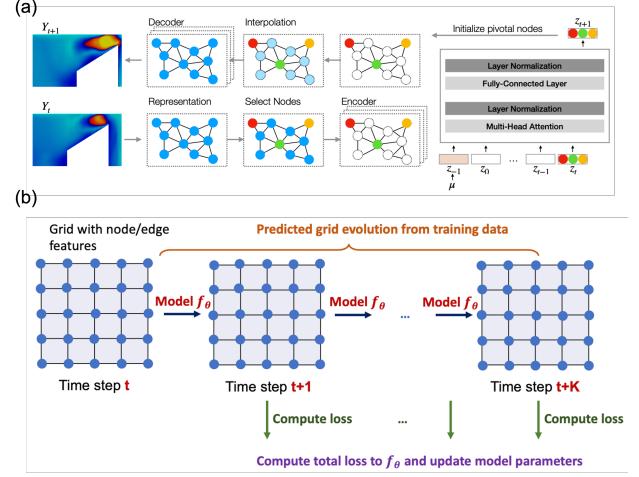


Figure 5. (a) Attention-based sequence MeshGraphNet (Han et al., 2022b). (b) Loss function based on Multi-step rollout (Wu et al., 2022).

duced hierarchical multiscale MeshGraphNets (Lam et al., 2022; Fortunato et al., 2022), as shown in Fig. 6, which employ message passing on multiple resolutions (finer and coarser) to capture interactions at different scales. In their highly successful weather forecasting model, each graph consists of approximately 1 million nodes, with each node encompassing 227 variables. Overall, the state representation encompasses an impressive 235 million values.

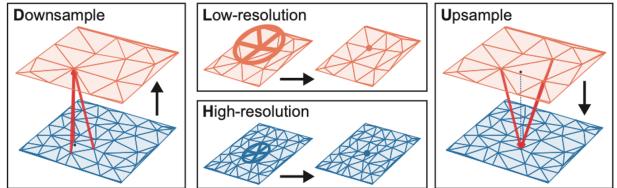


Figure 6. Multi-scale MeshGraphNet. Message passing in the two graphs with different resolutions (Fortunato et al., 2022).

2.2.3. ENCODE MORE PHYSICS PRIORS

Pure data-driven deep learning methods often suffer from error accumulation and limited generalizability when predicting complex physical processes. A more promising approach is to incorporate prior physics knowledge into deep learning models, leveraging our understanding of fundamental laws and principles (Hao et al., 2022), illustrated in Fig. 7. This integration of physical priors can encompass a range of biases, from strong to weak, including partial differential equations (PDEs), symmetry, and intuitive physical

constraints. In the context of message passing GNN solvers, intuitive physical constraints such as Newton's laws are utilized to learn local interactions between particles (Battaglia et al., 2016; Sanchez-Gonzalez et al., 2020). However, for accurate particle dynamics simulation, the conservation of momentum plays a crucial role. Prantl et al. (Prantl et al., 2022) further proposed a novel method that guarantees linear momentum in a learned physics engine by introducing antisymmetrical continuous convolutional layers. They demonstrated that this algorithm significantly improves accuracy and generalization, even for new scenarios involving up to one million particles. Regarding symmetry (Han et al., 2022a), Satorras et al. (Satorras et al., 2021) introduced Equivariant Graph Neural Networks (EGNNs) to ensure translations, reflections, and permutations. For addressing PDEs, the physical informed neural network (PINN) architecture (Karniadakis et al., 2021) is widely adopted, utilizing PDEs as regularization terms in the loss function. Gao et al. (Gao et al., 2022) presented a novel discrete PINN framework based on graph convolutional networks (GCNs) and the variational structure of PDEs to solve forward and inverse stationary partial differential equations (PDEs) in a unified manner.

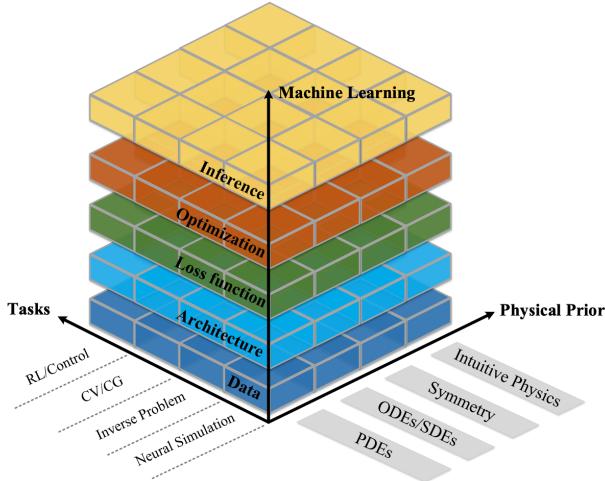


Figure 7. Various methods of incorporating physical prior knowledge into machine learning models, ranging from strong to weak forms. These physical priors can be incorporated into different aspects of machine learning models, such as data, model architecture, loss function, etc. And Applications of physical-informed ML in various fields (Hao et al., 2022).

3. Application of GNN-based simulators

3.1. Robotics control (Sim-to-Real)

Physical simulations that accurately model reality are crucial for robotic motion planning. However, transferring models from simulation to the real world presents challenges in bridging the "reality gap," which refers to the mismatch between simulated and real-world environments. Narrowing this gap has been a subject of intense interest in robotics. In Fig. 8, Shi et al. (Shi et al., 2022) directly learned the model from observed data captured by cameras in the real world. With just 10 minutes of real-world interaction training data, their model accurately predicts object deformations under planned actions and enables manipulation tasks with reasonable accuracy. In contrast, Linkerhagener et al. (Linkerhagener et al., 2023) employed an imputation-based model that incorporates sensory information as additional data for a standard Graph Network Simulator.

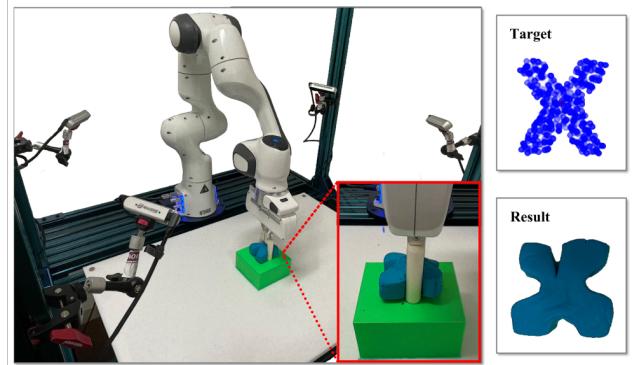


Figure 8. Experiment setup for directly learning the GNN-based simulator from the real world (Shi et al., 2022).

3.2. Inverse design

Another important application of learned GNN models is in inverse problems. The physical differentiability of GNN models enables gradient-based design optimization for inverse design tasks. In Fig. 9(a), Allen et al. (Allen et al., 2022b) designed surfaces and tools to manipulate fluid flows and optimized the shape of an airfoil to minimize drag. In Fig. 9(b), Zhao et al. (Zhao et al., 2022) inferred unknown system states or parameters by optimizing the mean squared error (MSE) between predicted ($\hat{u}^t(x)$) and ground truth ($u^t(x)$) values with respect to the initial state u^0 or parameter vector \mathbf{a} , given sparse observations of $u^t(x)$ at specific locations $x \in \Omega$.

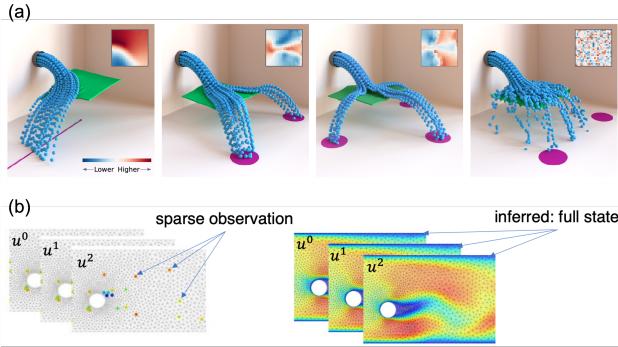


Figure 9. (a) Manipulate fluid flows into the hole with GNN-designed surfaces (Allen et al., 2022b). (b) Infer the full flow field based on some sparse observations and GNN based simulator (Zhao et al., 2022).

4. Conclusions

In this review, we have provided a systematic summary of the field of GNN-based simulators. We initially discussed the motivations for adopting GNNs in simulation learning and then presented the development of the prevailing frameworks in this field. We also highlighted the existing challenges in current model architectures and discussed potential paths for addressing these issues. Furthermore, we summarized the applications of GNN-based simulators in various domains. It is important to note that the high computational cost for training and inference of GNNs remains a significant challenge. Additionally, finding efficient ways to incorporate physical priors into existing architectures remains an important research direction. In conclusion, we affirm the potential of GNN-based simulators as valuable tools in scientific simulation, while acknowledging the need for further improvements and advancements in this exciting field.

References

- Allen, K., Lopez-Guevara, T., Stachenfeld, K. L., Sanchez Gonzalez, A., Battaglia, P., Hamrick, J. B., and Pfaff, T. Inverse design for fluid-structure interactions using graph network simulators. *Advances in Neural Information Processing Systems*, 35:13759–13774, 2022a.
- Allen, K. R., Lopez-Guevara, T., Stachenfeld, K., Sanchez-Gonzalez, A., Battaglia, P., Hamrick, J., and Pfaff, T. Physical design using differentiable learned simulators. *arXiv preprint arXiv:2202.00728*, 2022b.
- Battaglia, P., Pascanu, R., Lai, M., Jimenez Rezende, D., et al. Interaction networks for learning about objects, relations and physics. *Advances in neural information processing systems*, 29, 2016.
- Battaglia, P. W., Hamrick, J. B., Bapst, V., Sanchez-Gonzalez, A., Zambaldi, V., Malinowski, M., Tacchetti, A., Raposo, D., Santoro, A., Faulkner, R., et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018.
- Brandstetter, J., Worrall, D., and Welling, M. Message passing neural pde solvers. *arXiv preprint arXiv:2202.03376*, 2022.
- Brunton, S. L., Noack, B. R., and Koumoutsakos, P. Machine learning for fluid mechanics. *Annual review of fluid mechanics*, 52:477–508, 2020.
- Chang, M. B., Ullman, T., Torralba, A., and Tenenbaum, J. B. A compositional object-based approach to learning physical dynamics. *arXiv preprint arXiv:1612.00341*, 2016.
- Equer, L., Rusch, T. K., and Mishra, S. Multi-scale message passing neural pde solvers. *arXiv preprint arXiv:2302.03580*, 2023.
- Fortunato, M., Pfaff, T., Wirnsberger, P., Pritzel, A., and Battaglia, P. Multiscale meshgraphnets. *arXiv preprint arXiv:2210.00612*, 2022.
- Gao, H., Zahr, M. J., and Wang, J.-X. Physics-informed graph neural galerkin networks: A unified framework for solving pde-governed forward and inverse problems. *Computer Methods in Applied Mechanics and Engineering*, 390:114502, 2022.
- Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. PMLR, 2017.
- Han, J., Rong, Y., Xu, T., and Huang, W. Geometrically equivariant graph neural networks: A survey. *arXiv preprint arXiv:2202.07230*, 2022a.
- Han, X., Gao, H., Pfaff, T., Wang, J.-X., and Liu, L.-P. Predicting physics in mesh-reduced space with temporal attention. *arXiv preprint arXiv:2201.09113*, 2022b.
- Hao, Z., Liu, S., Zhang, Y., Ying, C., Feng, Y., Su, H., and Zhu, J. Physics-informed machine learning: A survey on problems, methods and applications. *arXiv preprint arXiv:2211.08064*, 2022.
- Karniadakis, G. E., Kevrekidis, I. G., Lu, L., Perdikaris, P., Wang, S., and Yang, L. Physics-informed machine learning. *Nature Reviews Physics*, 3(6):422–440, 2021.
- Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger, P., Fortunato, M., Pritzel, A., Ravuri, S., Ewalds, T., Alet, F., Eaton-Rosen, Z., et al. Graphcast: Learning skillful

- medium-range global weather forecasting. *arXiv preprint arXiv:2212.12794*, 2022.
- Li, Y., Wu, J., Tedrake, R., Tenenbaum, J. B., and Torralba, A. Learning particle dynamics for manipulating rigid bodies, deformable objects, and fluids. *arXiv preprint arXiv:1810.01566*, 2018.
- Linkerhagener, J., Freymuth, N., Scheikl, P. M., Mathis-Ullrich, F., and Neumann, G. Grounding graph network simulators using physical sensor observations. *arXiv preprint arXiv:2302.11864*, 2023.
- Pathak, J., Subramanian, S., Harrington, P., Raja, S., Chattopadhyay, A., Mardani, M., Kurth, T., Hall, D., Li, Z., Azizzadenesheli, K., et al. Fourcastnet: A global data-driven high-resolution weather model using adaptive fourier neural operators. *arXiv preprint arXiv:2202.11214*, 2022.
- Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., and Battaglia, P. W. Learning mesh-based simulation with graph networks. *arXiv preprint arXiv:2010.03409*, 2020.
- Prantl, L., Ummenhofer, B., Koltun, V., and Thuerey, N. Guaranteed conservation of momentum for learning particle-based fluid dynamics. *arXiv preprint arXiv:2210.06036*, 2022.
- Sanchez-Gonzalez, A., Heess, N., Springenberg, J. T., Merel, J., Riedmiller, M., Hadsell, R., and Battaglia, P. Graph networks as learnable physics engines for inference and control. In *International Conference on Machine Learning*, pp. 4470–4479. PMLR, 2018.
- Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., and Battaglia, P. Learning to simulate complex physics with graph networks. In *International conference on machine learning*, pp. 8459–8468. PMLR, 2020.
- Satorras, V. G., Hoogeboom, E., and Welling, M. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.
- Shi, H., Xu, H., Huang, Z., Li, Y., and Wu, J. Robocraft: Learning to see, simulate, and shape elasto-plastic objects with graph networks. *arXiv preprint arXiv:2205.02909*, 2022.
- Wu, T. Ai physicist machine learning for simulations. <https://drive.google.com/file/d/1cYFChLUVJZakZhnF81kN1S14WBV2HnQL/view>, 2022.
- Wu, T., Wang, Q., Zhang, Y., Ying, R., Cao, K., Sosic, R., Jalali, R., Hamam, H., Maucce, M., and Leskovec, J. Learning large-scale subsurface simulations with a hybrid graph network simulator. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4184–4194, 2022.
- Zhao, Q., Lindell, D. B., and Wetzstein, G. Learning to solve pde-constrained inverse problems with graph networks. *arXiv preprint arXiv:2206.00711*, 2022.