Unveiling Debris Flow Dynamics: Physics-Informed Machine Learning via MPM Wave Simulations

Trey A. Gower[1] and Justin Bonus[2]

[1]Undergraduate student, Department of Aerospace Engineering, The University of Texas at Austin.
Email: gowertrey@utexas.edu

[2]NHERI SimCenter, The University of California at Berkeley.
Email: bonus@berkeley.edu

ABSTRACT

Tsunamis, landslides, and storm-surges can mobilize debris, compounding the hazards faced by the built environments. Predicting the dynamics of entrained debris in these complex flow events is challenging due to the varying materials, phases, and the uncertainty associated with their respective properties. This study presents a novel approach to address this challenge using a combination of machine learning (ML) and high-performance numerical simulation. Our understanding of the dynamics of these hazards is underpinned by experimental facilities, which we aim to augment with a digital twin surrogate modeling workflow. In this manuscript, we introduce a prototype digital twin of the Hinsdale Wave Research Facility's Large Wave Flume at Oregon State University (OSU LWF) for studying wave-debris dynamics. The first-phase of our two-stage high-performance digital twin uses the Material Point Method (MPM), implemented within the Taichi programming language, to support classical numerical simulation of large-deformation, multi-material dynamics. For the creation of lightweight surrogate models in the second-phase, we leverage the highly flexible Graph Network Simulator (GNS), a graph neural network software package (GNN), to enable application of our workflow to a plethora of unexplored experimental facilities which may or may not center on wave-debris studies. By utilizing high-performance computing (HPC) and deep learning

networks, this approach enables the efficient representation of complex physics and facilitates uncertainty quantification in debris hazard events. With this, the ongoing integration of these technologies within the NHERI SimCenter's engineering workflow, specifically into HydroUQ, has great potential to provide a robust framework to run high-fidelity simulations. This work demonstrates the potential of ML-driven surrogate models to enhance the predictive capabilities and efficiency of simulation, paving the way for improved debris flow hazard mitigation strategies.

INTRODUCTION

The integration of computing, simulation, fluid dynamics, and machine learning in scientific research holds transformative potential, particularly in the modeling of natural hazards. Leveraging these technologies can lead to significant societal benefits. This study presents a novel approach to advancing our understanding of debris flow dynamics by integrating the Graph Network Simulator (GNS) Machine Learning (ML) model with the high-performance computation framework, Taichi. Within the NHERI network this coupling will aid research efforts in employing digital twins to DesignSafe/TACC/SimCenter infrastructure.

Several key papers have laid the groundwork for this approach by addressing the complexities of modeling debris flow fields. Notably, Trujillo-Vela and et al. (2022) expands on the classification of debris flows by defining the parameters that distinguish them. These flows typically consist of a mixture of fluids (e.g., water and air) and solids (e.g., clastic sediments, heavy metals, and other man-made objects). Given the diverse composition and size variations of the debris, debris flows are primarily governed by the interactions between solid, fluid, and solid-fluid forces, such as particle shearing, adhesion, collision, viscous shear, turbulence, and drag. By studying these interactions and the material size variations, the destructive capabilities of debris flows become evident. This relevant approach is demonstrated later with the use of the Material Point Method (MPM) in Taichi.

Further contributions by Kumar and Vantassel (2023) were instrumental in developing surrogate models for natural hazard events, focusing on Graph Neural Networks (GNN). Various datasets were used in the initial establishment of the novel, surrogate model workflow (Figure 1). They primarily center on granular materials (Kumar and Choi 2023d, 2023e, 2023c, 2024) and fluid flow (Kumar and Choi 2023b; Vantassel and Kumar 2022) as these are most applicable to natural hazards ranging from tsunamis to landslides.

The seminal work by Brunton et al. (2020) provides a foundation for the physics-informed machine learning applied in this article. This text outlines essential machine learning methodologies, optimizations, strengths, and limitations relevant to modern physics-based ML developed over the past decade. Although not used in this article, Brunton et al. (2016) introduced the Sparse Identification of Nonlinear Dynamics (SINDy) software, which could serve as a valuable alternative or even an addition to the existing workflow (Figure 1). SINDy offers a programmatic approach to characterizing complex physical systems, which is crucial for modeling debris-fluid events.

Another foundational work was experiments conducted by Arduino et al. (2018) at the National Science Foundation's (NSF) O.H. Hinsdale Wave Research Laboratory, which also inspired the development of this simulation and machine learning approach. Building on the foundation of rigorous and well-established physical models, such as these, reduces uncertainty when developing a numerical tool. Consequently, these experiments serve as both a guide and a potential validation for recreating experiments in simulation.

Presented here were the essential elements needed to efficiently develop a high-fidelity numerical tool. By integrating Taichi's advanced simulation capabilities with the user-friendly nature of GNS, this approach has reduced computational costs while maintaining accuracy in predicting complex physical phenomena and ensuring rapid model outputs. Although not yet a fully developed digital twin of the NSF experimental wave flume facility, it shows significant potential for accuracy across multiple use cases. This integrated approach, within the DesignSafe/TACC/SimCenter infrastructure, marks a step forward in applying machine learning to geophysical modeling, offering a tool that balances sophisticated simulation capabilities with practical usability and efficiency.

Herein we explore the methodology, focusing on the integration of simulation, machine learning models, and surrogate model development as presented in the workflow (Figure 1). We, then, expand on the workflow, detailing the validation of simulation and model performance. Moreover, from an

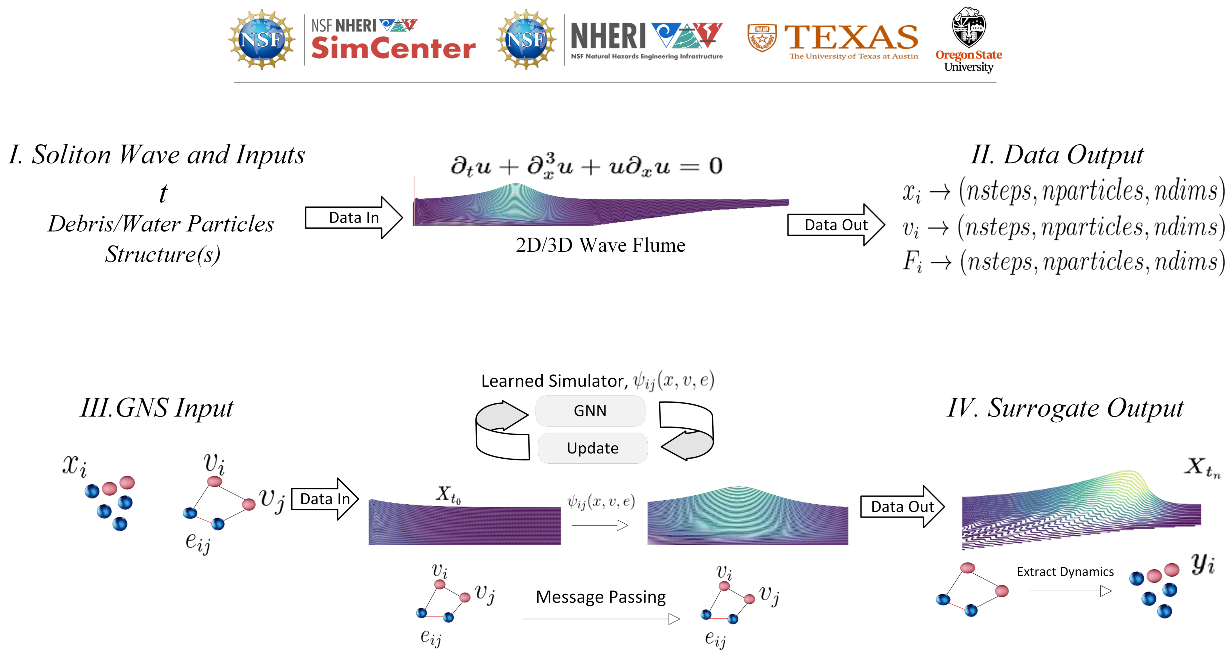

Fig. 1. Flow-chart of the standard workflow using the Taichi Wave-Flume Simulation with the GNS surrogate model for training and evaluation.

analysis of outcomes, future advancements in debris flow fields, simulation, and new applications for machine learning can be developed from the novel workflow.

BACKGROUND AND SIGNIFICANCE

Starting in the early 2000s, machine learning models were applied to weather models for calibration and processing of large datasets. Models of natural hazards began to transition from solely scaled, physical replications to a computer-augmented approach. Now, the combination of large natural hazards datasets, machine learning, and digital twin simulation environments is a rapidly advancing field (Figure 2).

Recent advancements in high-performance computing (HPC) and deep-learning networks (DNN) suggest prior limitations of physical experiments may be alleviated with digital twin augmentation. Using near-real-time (NRT) numerical simulations and surrogate models there is an opportunity to reduce facility operating costs while amplifying the research impact of experimental findings.

Along with this, natural disasters often lead to cascading effects; for instance, wildfires can trigger landslides, and earthquakes can cause tsunamis. Developing flexible simulation software and workflows to enhance data augmentation for these scenarios offers significant use cases to mitigate the effects of such event. Integration of HPC and GNN technology with facilities like the OSU LWF is an intriguing proposition.

METHODOLOGY

Rapidly training an ML model and bypassing expensive numerical interactions of full-order multi-material models is a desired outcome for this physics-informed machine learning process. Producing and acquiring quality data has high computational, monetary, and time costs associated with its collection. For rapid generation of quality surrogate models, the data needs to be complex enough to accurately capture the physics associated with a dynamic system, but simple enough that training any given model can take some 6-48 hours with 1-8 GPUs. In the pursuit of cheaper data, developed in this article were 2D and 3D simulations using the MPM within the Taichi framework (Hu et al. 2019, 2020, and 2021) (Figure 2). Along with this, the choice of an ML model that is fit for modeling the physics from a relatively sparse dataset was taken under consideration. GNN's and the GNS were a valid choice for their favorability in HPC environments and parallel GPU deployment (i.e., strong and weak scaling). This choice of model also ensured that usage within the DesignSafe/TACC/SimCenter infrastructure would be compatible and optimized with current software features.

The Material Point Method In Taichi

The Material Point Method (MPM) is a hybrid Lagrangian/Eulerian technique that uses a background grid to store particle velocities, positions, and other relevant physical data within the simulation. In the OSU LWFS, a solitary-like wave is generated using an error function (Eq. 1), which includes comprehensive wave-debris interaction, crucial for achieving results that closely match those observed in a physical wave flume (Figure 2). This approach replicates the experiments conducted by Mascarenas (2022) and utilizes the high-performance debris-fluid-structure interaction method described by Bonus (2023).

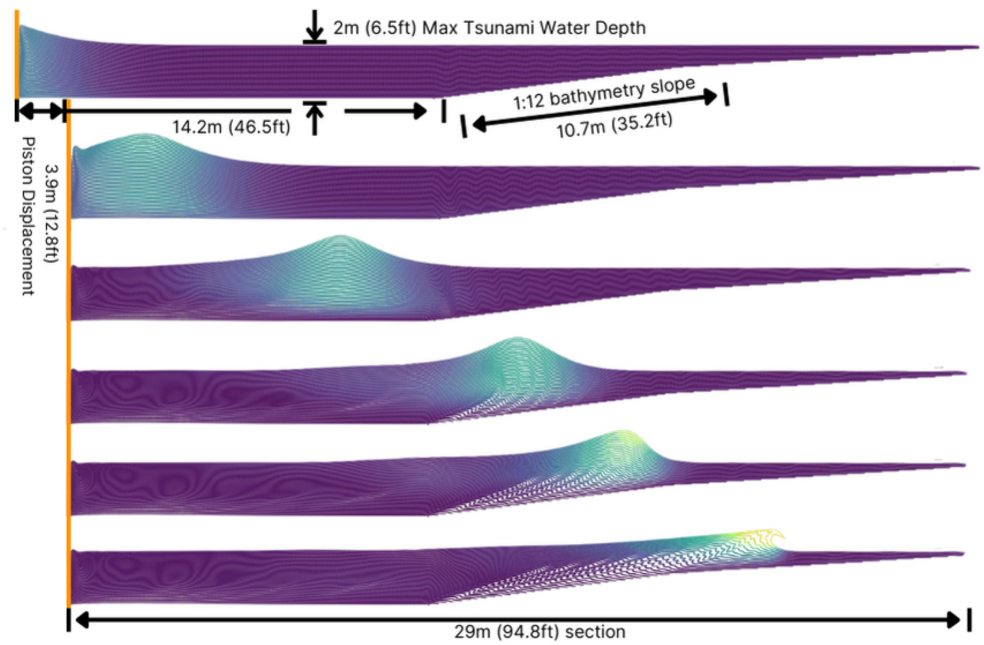

Fig. 2. Digital twin of Oregon State University's Large Wave Flume (OSU LWF) in the MPM, developed for surrogate training. Key-frames are sequential from $t = 0s$ to $t = 10s$ from top-to-bottom. Water, debris, the piston (far left in orange), with key dimensions shown.

$$\frac{1}{2} * A_{piston} * x_{piston} * (\text{erf}(\frac{t_{current} - \bar{t}}{\sigma_t} * 0.707) + 1.0) \tag{1}$$

The error function erf introduces a smooth, Gaussian-like transition in the wave profile, ensuring that the wave starts gradually and reaches its peak amplitude as time progresses. The multiplication by 0.707 in the argument of the error function normalizes the temporal width to create a solitary-like wave form. The overall expression scales the amplitude and displacement of the piston to simulate the wave's behavior accurately.

The Graph Network Simulator

GNNs have recently shown strong capabilities for physics-based surrogate modeling (Kumar and Choi 2023a). A graph (i.e., edges and nodes forming a network) visually represents the connections between objects, mapping directly to any geometric relationships present in the underlying system of study. The connections within these graphs often represent symmetries and structural patterns within the data. Yet, in cases where symmetries are absent, a GNN leverages its understanding of symmetries to interpolate missing or incomplete data. This process aims to fill in gaps and enhance the overall characterization of complex systems and datasets (Sanchez-Lengeling et al. 2021).

The Graph Network Simulator (GNS) is a generalized, efficient, and potentially accurate GNN for surrogate simulations in particulate and fluid systems. GNS code is viable for the MPM, Computational Fluid dynamics (CFD), and other simulation methodologies used in the field to produce high-quality surrogate models. This Simulator was the applied ML model to produce high-quality surrogates of The OSU Large Wave Flume Simulation (OSU LWFS).

GNS can be broken into encoder, processor, decoder, and updater. The architecture of GNS is designed to simulate particle dynamics. The model begins with vertex and edge encoders that transform particle features and their interactions into vertex and edge feature vectors, which are embedded into an initial latent graph. To improve performance, the vertex encoder excludes particle positions, focusing instead on features like velocity. Particle interactions are considered only within a defined connectivity radius, a critical hyperparameter that balances local interaction and global dynamics. The model then iteratively updates the graph through message passing using multi-layer perceptrons (MLPs), refining the representation of particle dynamics. The decoder extracts these dynamics from the updated graph to predict the particles' future velocities and positions. Finally, an updater uses Euler integration to update the physical state of the particles, ensuring efficient learning by focusing on dynamic interactions while simplifying static and inertial motions. The model is designed to generalize across different geometries, as particle dynamics are treated independently of their absolute positions. This process is summarized in the intermediate step of the workflow between steps three and four (Figure 1).

Loss Function of The Graph Network Simulator

Loss functions are imperative for accurately evaluating an algorithms performance in replicating certain critical aspects of datasets taken to be a "ground-truth" (i.e. lower loss is indicative of a better surrogate model for a given dataset). Typically, crafting loss functions is one of the most arduous and taxing steps in the machine learning process, requiring extensive testing of the many popular loss functions and evaluating efficacy of newly proposed ones specific to a problem domain. For The Graph Network Simulator, the loss function (Eq. 2) measures the difference between accelerations of GNS graph-nodes and the ground-truth particle trajectories from the MPM. In this article, only this standard loss function was taken into account:

$$\theta = \frac{1}{n}\sum_{i=1}^{n}((\ddot{x}_{t^i})_{GNS} - (\ddot{x}_{t^i})_{actual}) \tag{2}$$

where n and θ are the particle, which is interchangeable with graph-node and material point, and the parameter(s) to be learned in the GNS. Thus, this loss function tracks errors in the GNN models predictive capabilities of particle trajectories.

Aggregation of The MPM and GNS

The standard workflow is designed to outline the sequence of operations, as illustrated in Figure 1. The process begins with specifying parameters, including simulation duration, the number of water and debris particles (optimized for different machines), and, in the future, any structures to be analyzed. Steps two and three describe the exchange of inputs and outputs between the OSU LWFS and GNS, which will be discussed in greater detail later. Finally, step four presents the simplified model outputs from GNS and the insights derived from the augmented data.

Following this standard workflow, the integration of the MPM and GNS facilitates the generation of high-quality representations of physical systems. The OSU LWFS Taichi MPM model and the GNS model provide various renderings that emphasize the forces and dynamics within the OSU LWFS. Additionally, these models produce valuable data for further, more detailed analysis, including wave height, wave velocity, pressures, particle positions, and potential future insights into forces acting on bodies within the flume.

VALIDATION OF THE SIMULATION

The development of a simulation requires that numerical data represent a larger or more complex physical system to model. In creating the OSU LWFS, considerations included material law, boundary conditions, forcing functions, and, most importantly, validation of results via The

Korteweg-De Vries (KdV) shallow water partial differential equation (PDE). These elements were introduced to model debris flow of various types in scenarios such as tsunamis, landslides, and similar events.

The Korteweg-De Vries One-Soliton Solution

Offering a large number of explicit solutions and infinite conserved quantities, the KdV PDE (Eq. 3) serves as a main mathematical model in which we can validate the wave flume simulation. This shallow water model can be solved for multiple infinitely propagated soliton waves or, in the case of the OSU LWFS, a single-soliton solution for waters of constant depth (2m for the tsunami case). It is important to note that the KdV equations do not account for the bathymetry ramp in the OSU LWFS. That said, we have first-principal reasoning (fundamental truths) as validation for deviations caused by the bathymetry ramp. Found below is the PDE along with the single-soliton solution from Madsen et al. (2008) used for validation (Eq. 4 and 5).

$$\partial_t u + \partial_x^3 u + u\partial_x u = 0 \tag{3}$$

$$u(x,t) = H \,\mathrm{sech}^2(K_s\chi), \quad \chi = (x - ct) \tag{4}$$

where

$$K_s = \frac{1}{h}\sqrt{\frac{3H}{4h}}, \quad c_s = \sqrt{g(h+H)} \tag{5}$$

Validation Via The Single-Soliton KdV Solution

Plotting wave characteristics over time and space, e.q., Fig. 4 gives insight into the accuracy of the OSU LWFS using MPM. In the case of the bathymetry ramp, wave depth may be too large relative to the water depth for the wave to maintain stability. This discrepancy can lead to phenomena such as wave breaking. Variations in the depth and contours of the flumes floor can also cause waves to refract, reflect, or change in amplitude and speed. One can see that the amplitudes match up until reaching the bathymetry ramp where the wave begins to break. As noted above, a one-hundred percent match for KdV isn't necessary for the wave break. And although further validation of the OSU LWFS still needs to be conducted, initial findings have shown that the path taken is leading to promising results.

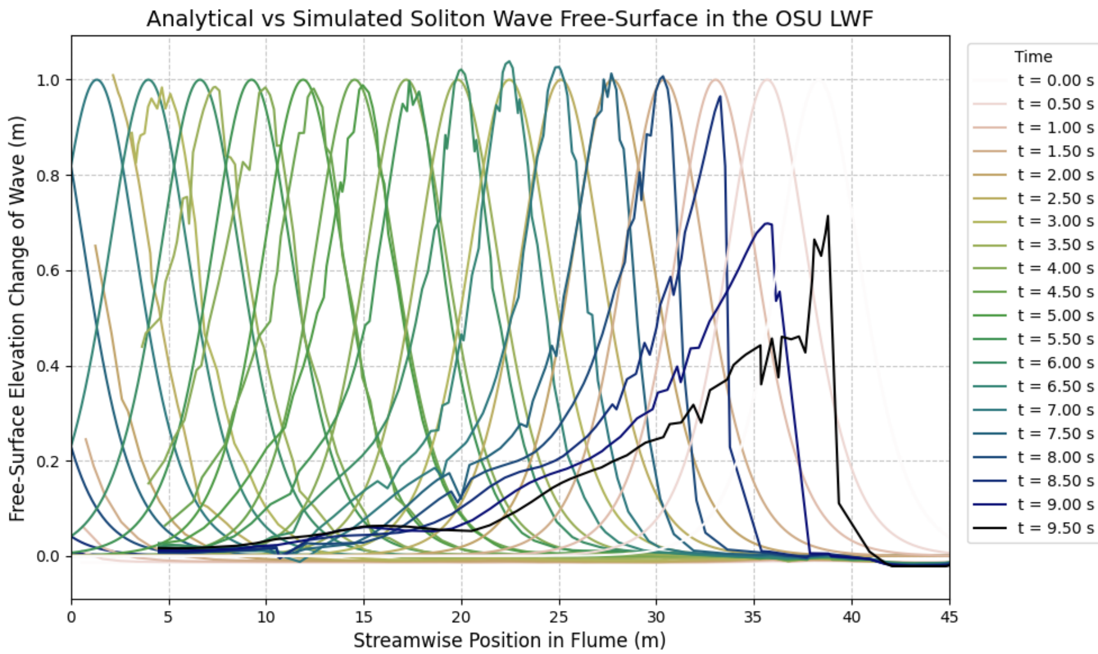

Fig. 3. Analytical vs. numerical wave amplitudes for times t = 3 to t = 8 (fully formed wave to the breaking of the wave over the bathymetry ramp).

GNS SURROGATE MODELS

Using data collected from the OSU LWFS and following the standard workflow (as outlined in steps two and three of Figure 1), GNS takes particle data inputs—such as position, velocity, and forces—and processes them through a series of neural network layers designed to capture the complex, nonlinear dynamics of the fluid system. This process generates a surrogate model that aims to effectively emulate the physical system (illustrated in Figures 5 and 6). For input data into GNS, the save metadata function and the save simulation function within the OSU LWFS script generate three files (and optionally a fourth validation dataset) of specific format (Figure 4).

The surrogate model, trained on the underlying physics learned from the dataset, replicates the specific conditions observed and may generalize to predict the behavior of the system under

• **Metadata** file with dataset information (sequence length, dimensionality, box bounds, default connectivity radius, statistics for normalization, ...) :

```
{
  "bounds": [[0.1, 0.9], [0.1, 0.9]],
  "sequence_length": 320,
  "default_connectivity_radius": 0.015,
  "dim": 2,
  "dt": 0.0025,
  "vel_mean": [5.123277536458455e-06, -0.00099652059
  "vel_std": [0.0021978993231675805, 0.0026653552458
  "acc_mean": [5.237611158734309e-07, 2.363302798885
  "acc_std": [0.0002582944917306106, 0.0002955453166
}
```

• Two **npz** files (training and validation datasets) containing all simulation trajectories (ground-truth/outputs):

```
simulation_data = {
    "simulation_0": (
        positions,  shape=(ntimestep, nparticles, ndims)
        particle_types,  shape=(nparticles)
        material_properties  # Optional
    )
    "simulation_1": (
        ...
    ),

    ...

    "simulation_n": (
    ...
    )
}
```

Fig. 4. GNS input files for surrogate model creation.

different scenarios. This could possibly include forecasting the response of the system to new, untested wave conditions, making it a powerful tool for exploring a wide range of wave cases and leading to replication with physical testing methods. The ability to simulate and predict such conditions with high fidelity allows for more efficient design and analysis in applications ranging from coastal engineering to disaster preparedness. In essence, this methodology allows for the proper allotment of funds to testing cases.

DISCUSSION AND FUTURE USE CASES

Developing simulation software to bridge data gaps in natural hazards is crucial for understanding and mitigating these impacts. Such tools are invaluable to research organizations, local communities, and others who need to plan for disaster resilience. Additionally, this software can reduce costs at NSF testing sites by offering insights into necessary physical model tests. The future of OSU LWFS and the integration of GNS technology holds significant promise.

The next logical step for the OSU LWFS and workflow is to implement structural objects into the flume for analysis, enabling the collection of stress, strain, force, and deformation data for structural analysis purposes in the wake of debris flows. From the boundary conditions already implemented in the digital twin it would be possible to apply to rigid structures. This could involve simply adding

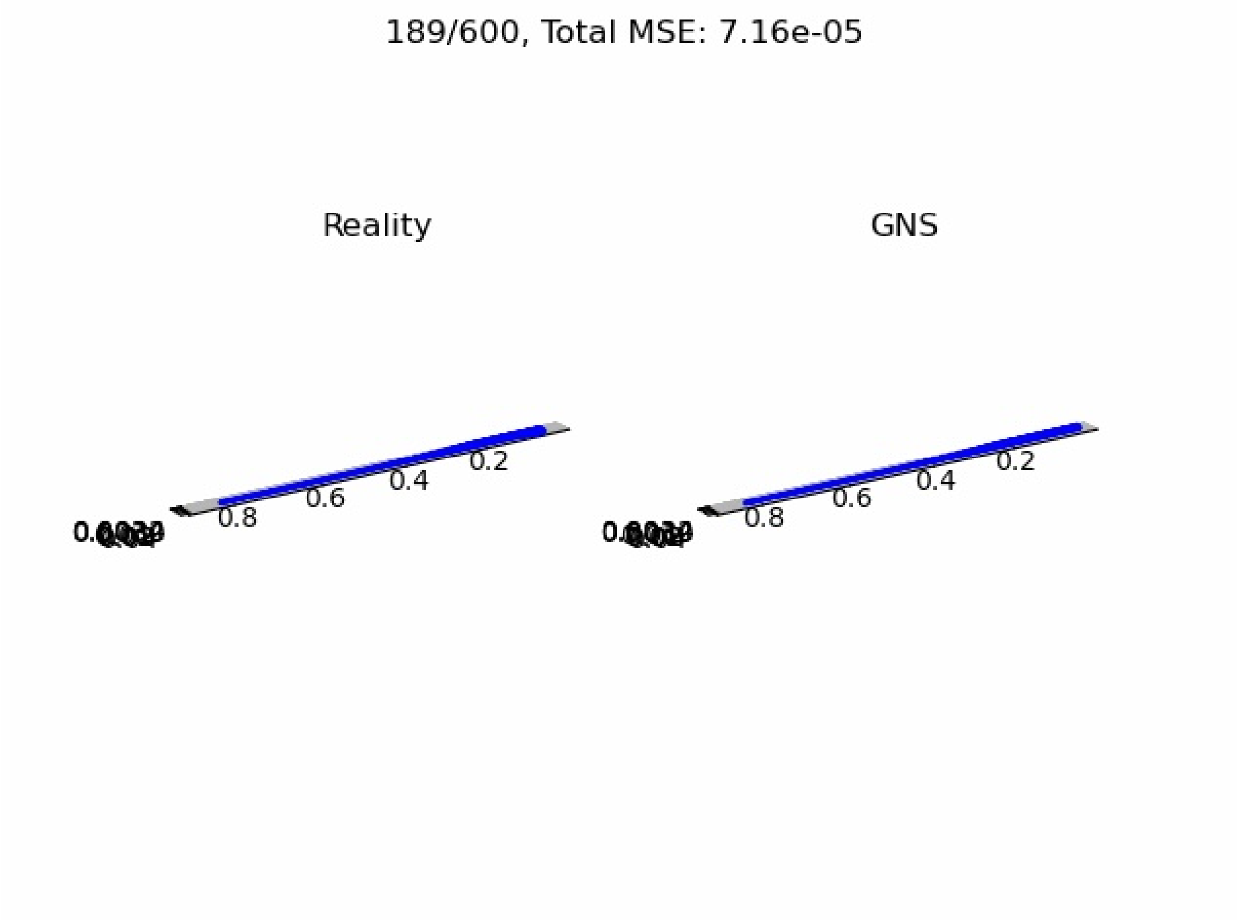

Fig. 5. GNS learned surrogate model of the OSU LWFS for t = 0s to t = 10s, 5000 water particles, 17 debris particles (100x down-sampling) and trained with one million steps with a 7e-5 loss metric.

Fig. 6. GNS learned surrogate model rendered in Houdini

seawalls to the existing wave run-up implemented by the bathymetry ramp or, more innovatively, using volumetric 3D point clouds (Figure 7). These point clouds, generated by technologies like OpenAI's Point-E Model or lidar, would significantly reduce time cost for other structures generated to be put into the OSU LWFS. Although, work on GNS still needs to be done to represent structural materials and material law for the production of a characteristic surrogate.

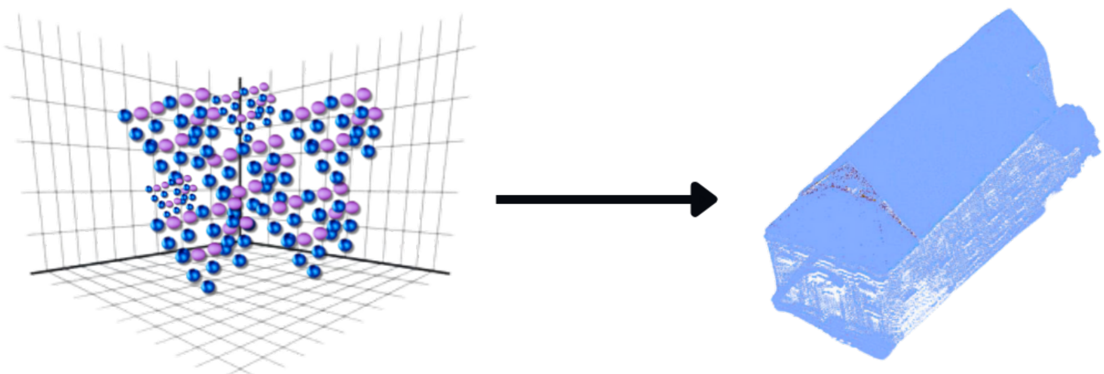

Fig. 7. 3D point cloud forming a structure to be placed into the OSU LWFS

Another application of the workflow could involve first using the OSU LWFS, along with datasets from realistic environments, to train the GNS on a wide range of wave dynamics, structural loading scenarios, and material properties. The GNS could then be integrated with a large language model (LLM) or an image classifier to process user input and generate point data for structures, which would serve as input into the GNS. This integrated system could subsequently produce useful and potentially viable data for more comprehensive structural analysis (Figure 8).

A well-crafted loss function needs to be developed and implemented to ensure the surrogate model adheres to first-principles, physics-based guidance and performs as intended for the problem at hand, as validated by ground-truth datasets. With this approach, there is the potential to explore alternative loss function strategies that could improve the prediction of dynamic behaviors of the GNS model.

The additions presented above, while basic, are powerful enhancements to the technologies discussed. Beyond these, there are numerous other applications for the MPM and the GNS that warrant exploration in the future.

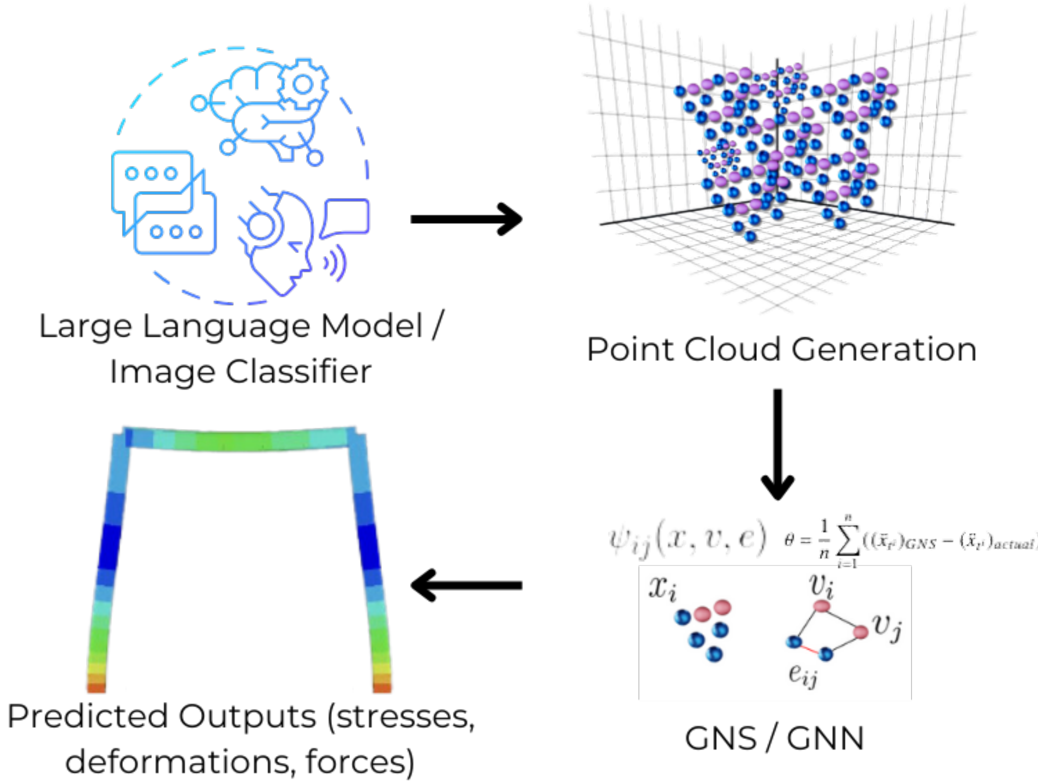

Fig. 8. LLM/image classifier coupled with point clouds and GNS workflow

CONCLUSION

Applying a surrogate modeling approach using the Material Point Method allows for the rapid accumulation of experience and bypasses the expensive interactions of full-order simulated models. The data must be complex enough to accurately capture the physics of a dynamic system, yet simple enough that training a machine learning model is manageable. With this objective in mind, over the past ten weeks, we developed a highly accurate digital twin of the OSU LWF (Figures 2 and 3), providing cost-effective and timely data inputs for the GNS. This digital twin not only captures the complex dynamics of debris flow with remarkable precision but also demonstrates the potential of using multifidelity simulations in conjunction with machine learning for real-time applications.

In pairing the highly flexible GNS software to easily train GNN models the desired workflow was achieved (Figure 1). This workflow underscores the importance of balancing data complexity with computational efficiency, showing that even with a surrogate modeling approach, it is possible to achieve high-fidelity simulations that are computationally feasible for practical deployment in the field of natural disasters.

DATA AVAILABILITY STATEMENT

Some or all data, models, or code generated or used during the study are available in a repository or online in accordance with funder data retention policies.

The original Graph Network Simulator (GNS) and the Taich-To-GNS-Wave-Flume workflow example presented in this manuscript may be found at:

https://github.com/geoelements/gns/tree/main

https://github.com/TreyGower7/Taichi-To-GNS-Wave-Flume

ACKNOWLEDGEMENTS

This research is supported by the National Science Foundation under the Natural Hazards Engineering Research Infrastructure Network Coordination Office Grant No. 2129782 and the National Science Foundation University of California Berkeley Simulation and Computational Modeling Center Grant No. 2131111.

I extend my deepest thanks to Dr. Justin Bonus and Dr. Matthew Schoettler for their outstanding mentorship, support, and guidance. I also wish to express my gratitude to Dr. Robin Nelson, who, over the past ten weeks, guided me through the writing and stylization process that culminated in this paper.

REFERENCES

Arduino, P., M. Motley, M. Eberhard, D. Cox, A. Barbosa, and P. Lomonaco. 2018. “Nheri debris impact experiments.” *DesignSafe-CI*. https://doi.org/10.17603/ds2-2y9x-qm74.

Bonus, J. 2023. “Evaluation of fluid-driven debris impacts in a high-performance multi-gpu material point method.” Ph.D. Thesis, University of Washington, Seattle, WA, United States.

Brunton, S., B. Noack, and P. Koumoutsakos. 2020. “Machine learning for fluid mechanics.” *Annu. Rev. Fluid Mech.*, 52, 477–508. https://doi.org/10.1146/annurev-fluid-010719-060214.

Brunton, S., J. Proctor, and N. Kutz. 2016. “Discovering governing equations from data by sparse identification of nonlinear dynamical systems..” *Proc. Natl. Acad. Sci. U.S.A.*, 113(15). https://doi.org/10.1073/pnas.1517384113.

Hu, Y., L. Anderson, T.-M. Li, Q. Sun, N. Carr, J. Ragan-Kelley, and F. Durand. 2020. “Difftaichi: Differentiable programming for physical simulation.” https://doi.org/10.48550/arXiv.1910.00935.

Hu, Y., T.-M. Li, L. Anderson, J. Ragan-Kelley, and F. Durand. 2019. “Taichi: A language for high-performance computation on spatially sparse data structures.” *ACM Transactions on Graphics (TOG)*, 38(6), 201. https://doi.org/10.1145/3355089.3356506.

Hu, Y., J. Liu, X. Yang, M. Xu, Y. Kuang, W. Xu, Q. Dai, W. T. Freeman, and F. Durand. 2021. “Quantaichi: A compiler for quantized simulations.” *ACM Transactions on Graphics (TOG)*, 40(4). https://doi.org/10.1145/3450626.3459671.

Kumar, K. and Y. Choi. 2023a. “Accelerating particle and fluid simulations with differentiable graph networks for solving forward and inverse problems.”

Kumar, K. and Y. Choi. 2023b. “Cylinder flow with graph neural network-based simulator.” *Designsafe-CI*. https://doi.org/10.17603/DS2-FZG7-1719.

Kumar, K. and Y. Choi. 2023c. “Granular column collapse with graph neural network-based simulator.” *Designsafe-CI*. https://doi.org/10.17603/DS2-GVVW-GT60.

Kumar, K. and Y. Choi. 2023d. “Training, testing data, and trained model.” *Designsafe-CI*. https://doi.org/10.17603/DS2-H3ZZ-GQ43.

Kumar, K. and Y. Choi. 2023e. “Training, validation, testing data, and trained model.” *Designsafe-CI*. https://doi.org/10.17603/DS2-4NQZ-S548.

Kumar, K. and Y. Choi. 2024. “Solving inverse problems using differentiable graph neural network simulator.” *Designsafe-CI*. https://doi.org/10.17603/DS2-0WJQ-0J84.

Kumar, K. and J. Vantassel. 2023. “Gns: A generalizable graph neural network-based simulator for particulate and fluid modeling.” *Journal of Open Source Software*, 8(88), 5025. https://doi.org/10.21105/joss.05025.

Madsen, P. A., D. Fuhrman, and H. Schäffer. 2008. "On the solitary wave paradigm for tsunamis." *Journal of Geophysical Research: Oceans*, 113. https://doi.org/10.1029/2008JC004932.

Mascarenas, D. 2022. "Experimental evaluation of loads from inundation-driven debris fields." Ph.D. Thesis, University of Washington, Seattle, WA.

Sanchez-Lengeling, B., E. Reif, A. Pearce, and A. B. Wiltschko. 2021. "A gentle introduction to graph neural networks." https://doi.org/10.23915/distill.00033.

Trujillo-Vela, M. and et al.. 2022. "An overview of debris-flow mathematical modelling.." *Earth Sci. Rev.*, 232(104135). https://doi.org/10.1016/j.earscirev.2022.104135.

Vantassel, J. and K. Kumar. 2022. "Graph network simulator datasets." *Designsafe-CI*. https://doi.org/10.17603/DS2-0PHB-DG64.