Spatio-Temporal Super-Resolution of Dynamical Systems using Physics-Informed Deep-Learning

Rajat Arora[*], **Ankit Shrivastava**[†]

Abstract

This work presents a physics-informed deep learning-based super-resolution framework to enhance the spatio-temporal resolution of the solution of time-dependent partial differential equations (PDE). Prior works on deep learning-based super-resolution models have shown promise in accelerating engineering design by reducing the computational expense of traditional numerical schemes. However, these models heavily rely on the availability of high-resolution (HR) labeled data needed during training.

In this work, we propose a physics-informed deep learning-based framework to enhance the spatial and temporal resolution of coarse-scale (both in space and time) PDE solutions without requiring any HR data. The framework consists of two trainable modules independently super-resolving the PDE solution, first in spatial and then in temporal direction. The physics based losses are implemented in a novel way to ensure tight coupling between the spatio-temporally refined outputs at different times and improve framework accuracy. We analyze the capability of the developed framework by investigating its performance on an elastodynamics problem. It is observed that the proposed framework can successfully super-resolve (both in space and time) the low-resolution PDE solutions while satisfying physics-based constraints and yielding high accuracy. Furthermore, the analysis and obtained speed-up show that the proposed framework is well-suited for integration with traditional numerical methods to reduce computational complexity during engineering design.

1 Introduction

Accurate modeling of the dynamic behavior of nonlinear systems is crucial for many industrial applications ranging from microscale MEMS sensors to large-scale structural systems. Therefore, significant research is being done to understand and resolve the complex physical phenomena occurring within these dynamical systems at extremely small spatial and temporal scales. This scientific pursuit of capturing complex physical phenomena occurring at widely varying spatio-temporal scales has led to the ever-increasing

---

[*]Senior member of technical staff at Advanced Micro Devices, Inc. (AMD).

[†]Post Doctoral associate at Sandia National Laboratories.

sophistication of the physical system's governing Partial Differential Equations (PDEs). For example, a PDE-based model (Arora and Acharya 2020a,b; Joshi et al. 2020; Arora, Zhang, and Acharya 2020; Arora 2019; Arora, Arora, and Acharya 2022) capturing defects evolution in materials at the nanoscale has been shown superior to conventional theories for a wide range of applications. However, the massive data storage and computational expense requirements to simulate such multi-physics coupled PDEs with high-fidelity bring traditional numerical solvers to their limits. Hence, fast and accurate techniques to perform these multi-physics simulations at multi-scale are of utmost importance.

On the other hand, the recent advances in Machine Learning (ML) have led to the development of several data-driven and Physics Informed ML models to solve PDEs occurring in fluid (Sun et al. 2020; Rao, Sun, and Liu 2020; Jin et al. 2021) and solid mechanics (Frankel, Tachida, and Jones 2020; Arora et al. 2022; Shrivastava et al. 2022; Zhu, Liu, and Yan 2021). However, issues ranging from its theoretical considerations (such as convergence, stability, accuracy, and generalizability) to issues related to boundary conditions, neural network architecture design, or optimization aspects still need to be fully resolved (Markidis 2021; Cuomo et al. 2022). Therefore, hybrid strategies integrating physics-informed ML with traditional approaches are emerging as a promising option to tackle this computational challenge of solving complex multi-physics PDEs (Arora 2021, 2022; Gao, Sun, and Wang 2021).

To this end, in this research, we aim to investigate a two-stage hybrid approach integrating ML and traditional approaches to obtain (reconstruct) solutions to spatio-temporal PDEs. 1) In the first stage, low-resolution (LR) PDE solutions are obtained by doing numerical simulations on a coarse scale both in space and time (using large grid size and timestep). This low-resolution solution with satisfactory accuracy can be generated with a huge reduction in computational expense compared to solving PDE on a fine scale. 2) In the second stage, the spatio-temporal resolution of this coarse-scale solution is enhanced using a physics-formed deep learning-based framework. A significant advantage of such 'physics-guided resolution enhancement' approach is the reduced computational expense and data storage requirements during the scientific exploration phase, which will significantly accelerate the process of scientific investiga-

tion and engineering design. This enhancement in resolution will also be referred to as the upsampling or super-resolution (SR) in this work.

The recent works involving spatio-temporal super-resolution of physical systems (Ren et al. 2022; Esmaeilzadeh et al. 2020; Fukami, Fukagata, and Taira 2021) use labeled high-resolution (HR) ground truth data for model training. Fukami et. al (Fukami, Fukagata, and Taira 2021) presents a purely data-driven SR framework, and therefore the super-resolved fields may not satisfy the physics-based constraints accurately. The works of Ren et al. (Ren et al. 2022) and Jiang et al. (Esmaeilzadeh et al. 2020) are 'almost data-driven' in that the scaling coefficient of prediction/data loss is chosen to be 20 times the coefficient of physics loss in the total loss for optimal accuracy. In fact, the errors are huge when HR-labeled data is not taken into account (purely physics-driven) (Esmaeilzadeh et al. 2020, see Table 1). Moreover, the HR-labeled data is computationally expensive to obtain. Therefore, its use during training completely negates the massive benefit of accelerating scientific computing that ML models aim to achieve. In this research, we present an end-to-end physics-informed deep learning-based framework to enhance the spatial and temporal resolution of coarse scale (both in space and time) PDE solutions without requiring any HR-labeled data. In summary, our main contributions are as follows:

1. We propose a novel and efficient physics-informed deep learning-based spatio-temporal resolution enhancement framework.

2. The framework consists of two trainable deep learning modules independently responsible for spatial and temporal upscaling of the coarse-scale PDE solution.

3. The physics based losses are implemented in a novel way to ensure tight coupling between the spatio-temporally refined outputs at different times and improve framework accuracy.

4. Unlike other works (Esmaeilzadeh et al. 2020; Fukami, Fukagata, and Taira 2021; Ren et al. 2022), the proposed framework does not rely on the availability of any high-resolution labeled data.

5. The effectiveness of the framework is tested by using the low-resolution coarse grid simulation data as input as opposed to using downsampled high-resolution labeled data.

Organization. The remainder of this paper is organized as follows: Section 2 briefly recalls the governing equations of a general spatio-temporal PDE. Section 3 present the details of the framework architecture, data setup, and composite loss function. Towards the end, Section 4 presents the results that validate the developed framework and demonstrate its effectiveness in super-resolving the solution fields for the test problem discussed. Finally, conclusions and avenues for further research are briefly discussed in Section 5.

2 Background

2.1 Governing equations

A typical spatio-temporal PDE governing the dynamical systems can be written in the following form:

$$\dot{z} - \mathcal{F}(z, x, t; x \in \Omega) = 0, \tag{1}$$

subjected to the initial and boundary conditions

$$\begin{aligned} \mathcal{I}(z;\ t = 0, x \in \Omega) &= 0, \\ \mathcal{B}(z;\ x \in \partial\Omega) &= 0. \end{aligned} \tag{2}$$

In equations 1, 2, z denotes the system solution comprised of m state variables, and $\dot{z}$ denotes its time derivative. $\mathcal{F}$ is the nonlinear functional of the polynomial and derivatives terms of its arguments. Ω and $\partial\Omega$ denotes the physical domain and its boundary, respectively. Any additional constraints which are inherently present in the system or required because of numerical methods, such as mixed finite element methods, can be assembled into $\mathcal{C}(z) = 0$.

Given a solution z_c which is obtained by solving the system of equations (1)-(2) at a coarse scale (large mesh size and timestep), our objective is to enhance the spatial and temporal resolution of the solution by using a physics-informed deep-learning based method. The schematic of our proposed framework is shown in Figure 1.

3 Methodology

Section 3.1 briefly outlines the physics-informed composite loss (objective function) used during the framework's training. We then present an overview of the "end-to-end" spatio-temporal resolution enhancement framework in Section 3.3.

3.1 Objective function

We note that the framework proposed in this work is unsupervised and therefore the composite loss function is obtained only from the governing equations of the system - Initial conditions, boundary conditions, and PDEs. Following (Arora 2022, Sec. IV), we impose the boundary conditions in the 'hard' manner (exactly), thus eliminating the boundary condition loss contribution from the composite loss. The physics-informed objective function is then written as follows:

$$\mathcal{L} = \lambda_1 \left(\underbrace{||\dot{z} - \mathcal{F}(z, x, t)||_1}_{\text{PDE}} \right) + \\ \lambda_2 \underbrace{||\mathcal{C}(z)||_1}_{\text{Constraints}} + \lambda_3 \underbrace{||\mathcal{I}(z)||_1}_{\text{I.Cs.}}, \tag{3}$$

where $||A||_1$ denotes the mean absolute error (MAE) between each element in the quantity A and target 0.

We use a fourth-order finite difference scheme to evaluate the spatial derivatives of the solution over the grid. For time discretization, we use the Crank-Nicholson algorithm, which has the virtues of being unconditionally stable and is also second-order accurate in both space and time dimensions.
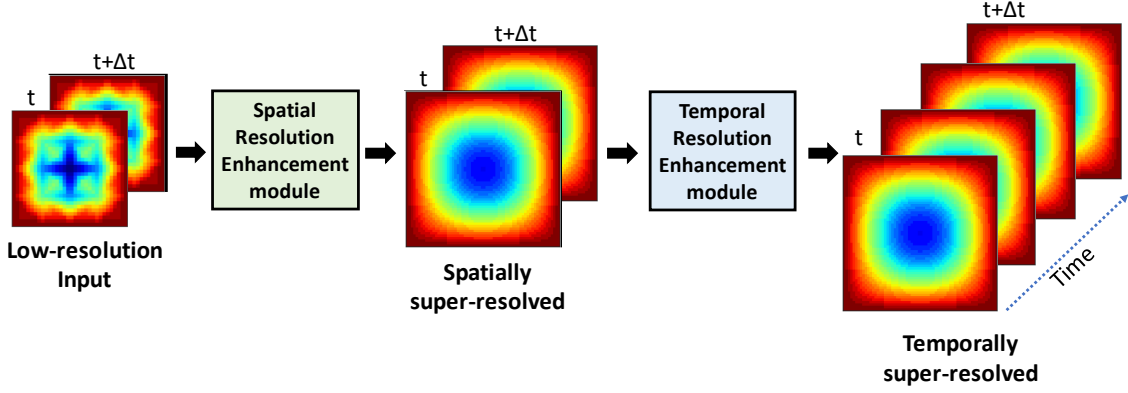
Figure 1: The figure shows the structure of the proposed framework. The input to the framework is a two-channel image of the coarse-scale solution, and the final output of the framework is a $(k+1)$ channel image of the spatio-temporally resolved solution. The framework consists of a spatial and a temporal resolution enhancement module. The former performs super-resolution in space, whereas the latter super-resolves in the temporal direction. The modules are trained independently.

3.2 Input and output for the framework

The input to the framework consists of tuples of LR coarse-scale PDE solution $\mathcal{I}_c^t = \{z_c^t, \ z_c^{t+\Delta t}\}$ at the consecutive timesteps t and $t + \Delta t$, respectively. The outputs from the framework consist of the spatially upscaled PDE solution at the same timesteps $\{\hat{z}^t, \hat{z}^{t+\Delta t}\}$ along with the synthesis of the HR snapshots at $(k-1)$ intermediate timesteps $\{\hat{z}^{t+\frac{\Delta t}{k}}, \hat{z}^{t+2\frac{\Delta t}{k}}, ..., \hat{z}^{t+(k-1)\frac{\Delta t}{k}}\}$. Therefore, the framework produces spatially upscaled PDE solution at $(k+1)$ timesteps referred to $\mathcal{O}^t$. We refer to the scalar k as the temporal upscaling factor and is given as the ratio of the coarse-scale timestep to the fine-scale timestep i.e. $k = \frac{\Delta t_c}{\Delta t_f}$. Similarly, the upscaling factor in the spatial direction s is given as the ratio of the coarse to fine grid resolution, i.e. $s = \frac{\Delta x_c}{\Delta x_f}$.

3.3 Framework Architecture

The framework is composed of two trainable modules: the Spatial resolution enhancement module and the Temporal resolution enhancement module, as shown in Figure 1. These two modules independently perform the super-resolution in space and time, respectively. We observed that this dual module approach which first performs super-resolution in space and subsequently increases temporal resolution leads to better convergence and accuracy of the super-resolved fields as is also observed in the data-driven approach of Fukami et. al (Fukami, Fukagata, and Taira 2021).

A typical solution z of any dynamical system consists of m state variables. Therefore, each module in the framework consists of m deep learning models (with the same architecture) that individually reconstruct each state variable; refer to figure 2. These models, however, are coupled during the training through the objective function (loss). Sections 3.3 and 3.3 discuss these spatial and temporal super-resolution modules in greater detail, respectively.

Spatial resolution enhancement module Given a tuple of LR snapshots at two consecutive time steps $\mathcal{I}^t = \{z_c^t, z_c^{t+\Delta t}\}$ for the state vector z, the spatial upscaling module outputs the corresponding HR frames $\{\hat{z}^t, \hat{z}^{t+\Delta t}\}$ representing the enhanced spatial resolution of the input state at the original time steps. Therefore, the input (output) to this module consists of a two-channel image representing the values of the LR (HR) state variables at time steps t and $t + \Delta t$.

During training, we observe that for the successful evolution of solution from initial conditions in both the output channels, the PDE loss in (3) has to be implemented both within an output and across outputs as highlighted in Figure 3. This coupling in the loss helps in mitigating the *propagation failure* mode (Daw et al. 2022) for this module.

Each model in the spatial upscaling module is built upon the Residual Dense Network (RDN) proposed in (Zhang et al. 2018), which has unique advantages for image SR over other networks (Zhang et al. 2018, Sec. 4). We use 4 residual blocks with 8 layers in each block and a feature channel size of 32. The kernel size for convolution is set to be 3.

Temporal resolution enhancement module The temporal module enhances the resolution of the state variables in time. The output of the spatial module serves as an input to the temporal module. Therefore, the input is a two-channel image representing the values of the HR state variables at time steps, t, and $t + \Delta t$. The module outputs an image with $(k+1)$ channels, representing the state variables at time steps $\{t, t + \frac{\Delta t}{k}, \ldots, t + \Delta t\}$. We note here that the inputs to the temporal module still have the temporal discretization error of the order $O(\Delta t^2)$. Therefore, the temporal module also reconstructs the solution at the initial and the final times t and $t + \Delta t$, respectively, to reduce this error to $O\left(\left(\frac{\Delta t}{k}\right)^2\right)$. We make the following changes to the composite loss (3) for the training of this module:

- Similar to the implementation of PDE loss in the spatial module, the PDE loss for the temporal module is also
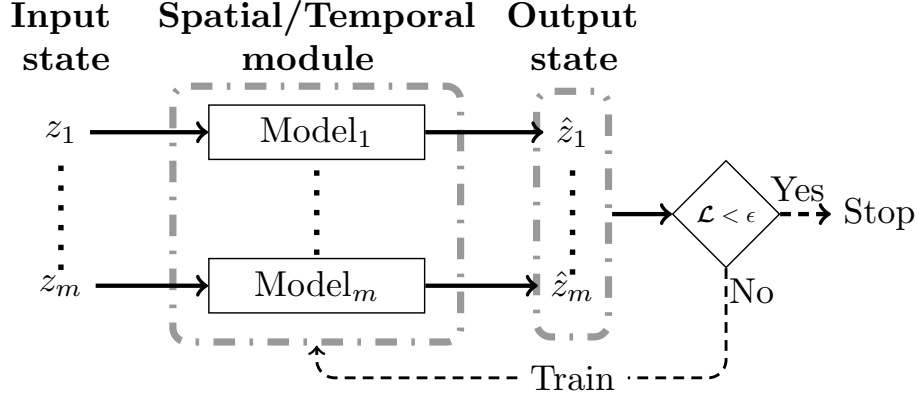
Figure 2: The figure shows the structure of the spatial/temporal module. Each module consists of m deep-learning models independently super-resolving the state variables $\{z_1, z_2, \ldots, z_m\}$. The models are, however, trained using a coupled loss function.
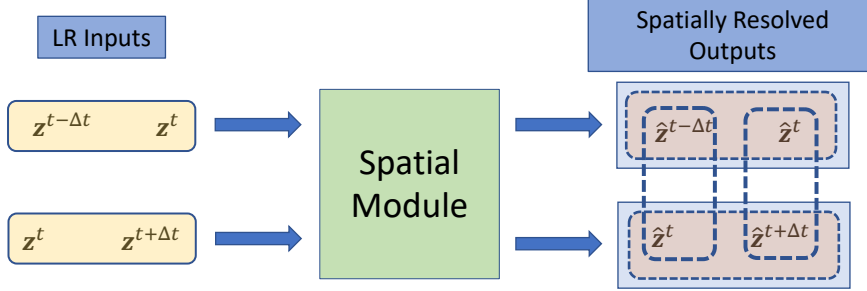

Figure 3: The figure shows the coupling between the predictions from the spatial resolution enhancement module while calculating the time derivative in the physics-based PDE loss. The two dashed vertical boxes show state variables from different outputs to calculate the time derivatives. Similarly, the two dashed horizontal boxes show the state variables within the same outputs that are used to calculate the time derivative.

implemented both within an output and across outputs, as highlighted in Figure 4.

- Since we reconstruct the outputs at initial and final timesteps (t and $t+\Delta t$), we add a constraint loss between these inputs and outputs, which helps in faster convergence of the module.

For the sake of simplicity, the model architecture is similar to the spatial module except for the number of channels in the output and the elimination of upsampling layer.

4 Experiment

In this section, we briefly discuss the problem setup, dataset generation, and evaluation metric. We then evaluate the performance of the developed framework by investigating its effectiveness in enhancing the spatio-temporal resolution of the coarse-scale solutions to an elastodynamics problem. The results presented herein demonstrate the remarkable accuracy of the network without requiring any HR-labeled data, unlike all previous works.

4.1 Setup

We consider a mixed-variable elastodynamics system widely used in structural engineering and seismologic applications. The governing equations of the system (in the absence of inertia) under 2-d antiplane strain conditions are given as follows:

$$v = \dot{u}$$
$$\frac{\partial \sigma_{xz}}{\partial x} + \frac{\partial \sigma_{yz}}{\partial y} = \frac{\rho}{\mu}\dot{v} \qquad (4)$$
$$\sigma_{xz} = \mu\frac{\partial u}{\partial x} \;\; ; \;\; \sigma_{yz} = \mu\frac{\partial u}{\partial y},$$

where ρ is the material density, b is the body force per unit volume, μ denotes the shear modulus of the material, and u and v denote the displacement and material velocity (both in the z direction), respectively. σ_{xz} and σ_{yz} denote the nonzero components of the (symmetric) stress tensor $\boldsymbol{\sigma}$. The equations 4, along with the boundary and initial conditions in the equations 5, define the governing equations of the sys-
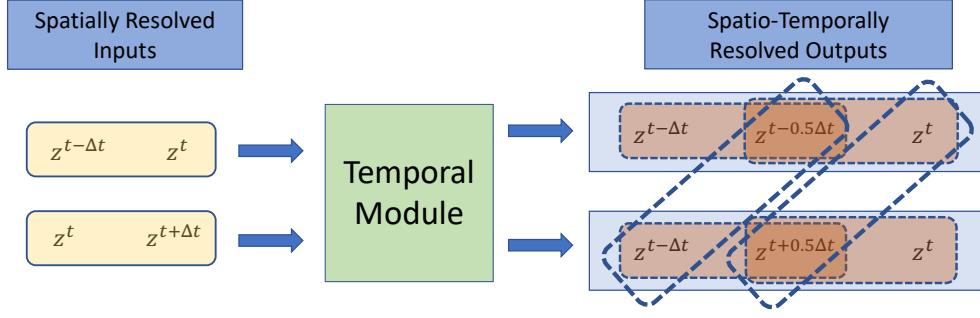
Figure 4: The figure shows the coupling between the predictions from the temporal resolution enhancement module while calculating the time derivative in the physics-based PDE loss. The (two) dashed inclined boxes show state variables from different outputs to calculate the time derivatives. Similarly, the four dashed horizontal boxes show the state variables within the same outputs that are used to calculate the time derivative.

tem.

$$u(t) = u_{bc} \text{ on } \partial\Omega,$$
$$u(t = 0) = u_0 \text{ and } v(t = 0) = v_0. \qquad (5)$$

In the above, u_0 and v_0 denote the known initial conditions on u and v, respectively. u_{bc} denote the known displacement on the domain boundary $\partial\Omega$. Without loss of generality, we take $v_0 = u_{bc} = 0$ in this work. The initial condition for u is $u_0 = \sin(\pi x)\sin(\pi y)$.

4.2 Generation of low-resolution input data

To evaluate the performance of the framework, we generate the low-resolution input data for the problem setup presented above in Section 4.1. We solve the governing system of equations (4) for the non-dimensional values of stress, displacement, and velocity, which amounts to setting $\frac{\rho}{\mu} = 1$ in (4). The equations are solved on a coarse mesh of 64 triangular elements (41 nodes and $\Delta x_c \approx 0.176$) and coarse-scale timestep $\Delta t_c = 0.005$ with finite element method (FEM) for 48 timesteps. The obtained coarse grid solution is then interpolated to a structured 32×32 grid using the FEM interpolation and is used as an input to the super-resolution framework. For comparison, the HR ground truth data is obtained by solving the same equations on a fine mesh of 64×64 nodes and $\Delta x_f = .0158$ with $\Delta t_f = \frac{\Delta t_c}{2}$. This sets the spatial and temporal upscaling factors to be $s = 11$ and $k = 2$, respectively, for the test case discussed.

4.3 Evaluation metric

For any state variable α (one of u, v, σ_{xz}, or σ_{yz}), we define a full field error measure e to quantitatively measure the discrepancy between the HR ground truth data α^{HR} and the framework predictions $\hat{\alpha}$ as follows:

$$e = \frac{||\alpha^{HR} - \hat{\alpha}||_{L^2}}{||\alpha^{HR}||_{L^2}} \times 100. \qquad (6)$$

We note here that the HR ground truth data is only used for comparison with the predicted outputs of the framework.

4.4 Training

The framework is implemented and trained using PyTorch. The training strategy used in this work consists of two stages: a) The spatial module is trained in the first stage, which amounts to enhancing spatial resolution. b) The outputs from the trained spatial module are then used as input to the temporal module during its training. The weights for the spatial module are frozen during the second stage.

In both stages, we use Adam optimizer with a learning rate of 4×10^{-4} for around 2000 epochs with a batch size of 8 samples. As the training progresses, the learning rate is adjusted using `ReduceLROnPlateau` scheduler with `patience` set to 40. We use a sequential training data sampler to respect the causal structure inherently present in the spatio-temporal PDEs, which has been shown to improve the accuracy of the physics-informed neural networks significantly, refer (Wang, Sankaran, and Perdikaris 2022). We use the scaling coefficients $\lambda_1 = 5$, $\lambda_2 = 1$, and $\lambda_3 = 10$ in this work.

4.5 Performance Evaluation

This section focuses on the performance evaluation of the proposed framework. The low-resolution coarse-scale PDE solution generated in Section 4.2 is used as an input to the framework. In what follows, we compare the error measure e for $t \geq 0.02$ because the initial velocity condition is zero in this work.

Figure 5a shows the error measure e for all the HR state variables obtained from the framework. We note that the % error e is less than 4% for all the state variables at all times $t \leq 0.24$. We also note that the framework exhibits poor extrapolation capabilities as the errors become quite large after $t > 0.24$. Figure 5b shows the error measure e for the state variables when a simple bi-linear interpolation of the low-resolution data is performed to upscale the solution. We can see that the errors are relatively large for all of the state variables.

Figure 6 shows the super-resolved state variables obtained from the framework at a particular time $t = 0.14$. We can notice that the super-resolved fields are indistinguishable
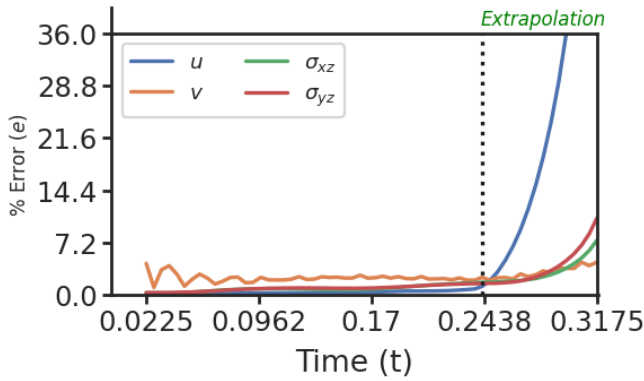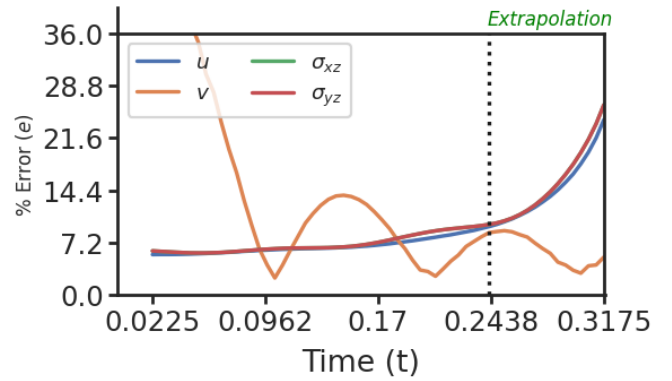
(a) Proposed framework. (b) Bi-linear interpolation.

Figure 5: The figure shows the error for the solution fields of elastodynamics at different time steps after super-resolution using the proposed framework, Figure 5a, and using bi-linear interpolation, Figure 5b. The super-resolution framework is highly accurate as compared to simple bi-linear interpolation. The framework's accuracy suffers beyond time, $t = 0.2438$ (extrapolation region), as the framework is trained till $t = 0.2438$. The high error in the extrapolation region suggests that the framework is capable of producing high-accuracy results for super-resolution within the convex hull of training set.

from the HR ground truth data. At the same time, the simple bi-linear upscaling of the LR input data can be seen to significantly differ from the ground truth reference data.

Therefore, we conclude that the proposed framework successfully enhanced the spatial and temporal resolution of the solution by a factor of 11 and 2, respectively.

4.6 Speed-up

Next, we calculate the speed-up obtained using the proposed framework to upsample the coarse-scale solution compared to obtaining the fine-scale solution using FEM. The FEM calculations are performed on a single-core AMD EPYC 7742 Processor. For the same simulation end time, the coarse-scale simulation ($\Delta x_c \approx 0.176$ and $\Delta t_c = 0.005$) takes 0.033 seconds, whereas it takes 4.27 seconds to run the fine-scale simulation ($\Delta x_f \approx 0.0158$ and $\Delta t_f = 0.0025$). The above numbers are averaged over 10 runs of the entire simulation. They do not include the time taken for mesh generation, node numbering, memory allocation, data I/O, or any other bookkeeping required by FEM.

On the other hand, after training, the inference time (averaged over 100 inferences) for the spatial and temporal modules are 0.015 and 0.017 seconds, respectively. Therefore, using the proposed framework, it takes around 1.569 seconds (including time taken for the coarse-scale simulation) to obtain the fine-scale PDE solution for the same simulation end time. Thus, we obtain a speed-up factor of around 2.72 for the (relatively) simple test case discussed here. These inferences are performed on an NVIDIA Tesla V100-SXM2 GPU with 32 GB RAM.

We emphasize here that the speed-up factor strongly depends on the complexity of the problem (linear vs. nonlinear) and the framework's spatial and temporal upscaling factors.

5 Conclusion

In this work, we presented a novel unsupervised physics-informed machine learning framework, a first in the literature, that:

- enables spatial and temporal upscaling (resolution enhancement) of coarse-grained solutions to spatio-temporal PDEs while ensuring that the (upscaled) outputs satisfy the governing laws of the system.

- easily allows imposition of any additional constraints (PDE or algebraic) in the framework.

- is generalizable to non-rectangular domains by using elliptic coordinate transformation as outlined in (Gao, Sun, and Wang 2020).

- is amenable to scalability to clusters with multi-gpu nodes using `DistributedDataParallel` functionality in PyTorch (Li et al. 2020) functionality for workloads that require substantial computational resources.

We demonstrated the framework's application to an elastodynamics problem under anti-plane strain conditions. The framework successfully enhanced the spatial and temporal resolutions of the coarse-scale input fields by a factor of $s = 11$ and $k = 2$ in the space and time directions, respectively, while satisfying the physics-based constraints and yielding great accuracy (error $\leq 4\%$).

In the future, we aim to study the framework's application on a wide range of physical applications in fluid mechanics and compare the performance of the current framework with other super-resolution works (although supervised) in the literature (Ren et al. 2022; Esmaeilzadeh et al. 2020). Another interesting line of research to pursue would be the addition of ConvLSTM (Shi et al. 2015) in the temporal module to improve the predicting capabilities of the framework beyond the convex hull of the traning set.
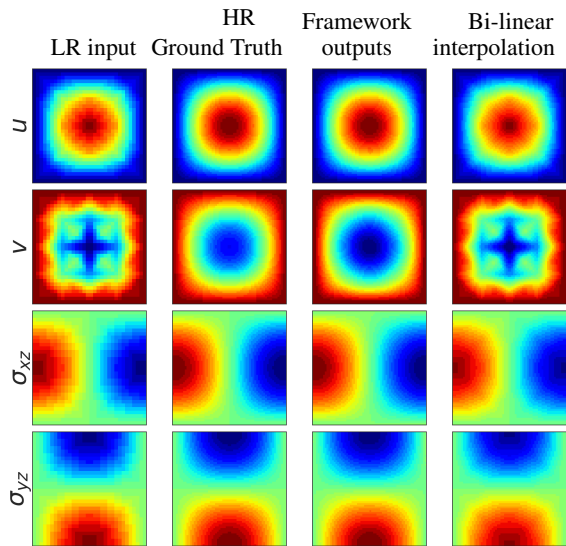
Figure 6: The figure shows the snapshot of the elastodynamics' solution fields at a time step. The LR input and HR Ground-truth are the coarse-scale and fine-scale solution fields obtained using the finite element method. For a given LR input, the reconstructed solution from the framework is highly accurate compared to the bi-linear interpolation method.

Research Data

Acknowledgment

References

Alnæs, M.; Blechta, J.; Hake, J.; Johansson, A.; Kehlet, B.; Logg, A.; Richardson, C.; Ring, J.; Rognes, M. E.; and Wells, G. N. 2015. The FEniCS project version 1.5. *Archive of Numerical Software*, 3(100).

Arora, A.; Arora, R.; and Acharya, A. 2022. Mechanics of micropillar confined thin film plasticity. *arXiv preprint arXiv:2202.06410*.

Arora, R. 2019. *Computational Approximation of Mesoscale Field Dislocation Mechanics at Finite Deformation*. Ph.D. thesis, Carnegie Mellon University.

Arora, R. 2021. Machine learning-accelerated computational solid mechanics: Application to linear elasticity. *arXiv preprint arXiv:2112.08676*.

Arora, R. 2022. PhySRNet: Physics informed super-resolution network for application in computational solid mechanics. *arXiv preprint arXiv:2206.15457*.

Arora, R.; and Acharya, A. 2020a. Dislocation pattern formation in finite deformation crystal plasticity. *International Journal of Solids and Structures*, 184: 114–135.

Arora, R.; and Acharya, A. 2020b. A unification of finite deformation J2 Von-Mises plasticity and quantitative dislocation mechanics. *Journal of the Mechanics and Physics of Solids*, 143: 104050.

Arora, R.; Kakkar, P.; Dey, B.; and Chakraborty, A. 2022. Physics-informed neural networks for modeling rate-and temperature-dependent plasticity. *arXiv preprint arXiv:2201.08363*.

Arora, R.; Zhang, X.; and Acharya, A. 2020. Finite element approximation of finite deformation dislocation mechanics. *Computer Methods in Applied Mechanics and Engineering*, 367: 113076.

Cuomo, S.; Di Cola, V. S.; Giampaolo, F.; Rozza, G.; Raissi, M.; and Piccialli, F. 2022. Scientific Machine Learning through Physics-Informed Neural Networks: Where we are and What's next. *arXiv preprint arXiv:2201.05624*.

Daw, A.; Bu, J.; Wang, S.; Perdikaris, P.; and Karpatne, A. 2022. Rethinking the importance of sampling in physics-informed neural networks. *arXiv preprint arXiv:2207.02338*.

Esmaeilzadeh, S.; Azizzadenesheli, K.; Kashinath, K.; Mustafa, M.; Tchelepi, H. A.; Marcus, P.; Prabhat, M.; Anandkumar, A.; et al. 2020. MeshfreeFlowNet: a physics-constrained deep continuous space-time super-resolution framework. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, 1–15. IEEE.

Frankel, A.; Tachida, K.; and Jones, R. 2020. Prediction of the evolution of the stress field of polycrystals undergoing elastic-plastic deformation with a hybrid neural network model. *Machine Learning: Science and Technology*, 1(3).

Fukami, K.; Fukagata, K.; and Taira, K. 2021. Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows. *Journal of Fluid Mechanics*, 909.

Gao, H.; Sun, L.; and Wang, J.-X. 2020. PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parametric PDEs on irregular domain. *arXiv e-prints*, arXiv–2004.

Gao, H.; Sun, L.; and Wang, J.-X. 2021. Super-resolution and denoising of fluid flow using physics-informed convolutional neural networks without high-resolution labels. *Physics of Fluids*, 33(7): 073603.

Jin, X.; Cai, S.; Li, H.; and Karniadakis, G. E. 2021. NSFnets (Navier-Stokes flow nets): Physics-informed neural networks for the incompressible Navier-Stokes equations. *Journal of Computational Physics*, 426: 109951.

Joshi, T.; Arora, R.; Basak, A.; and Gupta, A. 2020. Equilibrium shape of misfitting precipitates with anisotropic elasticity and anisotropic interfacial energy. *Modelling and Simulation in Materials Science and Engineering*, 28(7): 075009.

Li, S.; Zhao, Y.; Varma, R.; Salpekar, O.; Noordhuis, P.; Li, T.; Paszke, A.; Smith, J.; Vaughan, B.; Damania, P.; et al. 2020. Pytorch distributed: Experiences on accelerating data parallel training. *arXiv preprint arXiv:2006.15704*.

Markidis, S. 2021. The old and the new: Can physics-informed deep-learning replace traditional linear solvers? *Frontiers in big Data*, 92.

Rao, C.; Sun, H.; and Liu, Y. 2020. Physics informed deep learning for computational elastodynamics without labeled data. *arXiv preprint arXiv:2006.08472*.

Ren, P.; Rao, C.; Liu, Y.; Ma, Z.; Wang, Q.; Wang, J.-X.; and Sun, H. 2022. Physics-informed Deep Super-resolution for Spatiotemporal Data. *arXiv preprint arXiv:2208.01462*.

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Advances in neural information processing systems*, 28.

Shrivastava, A.; Liu, J.; Dayal, K.; and Noh, H. Y. 2022. Predicting peak stresses in microstructured materials using convolutional encoder–decoder learning. *Mathematics and Mechanics of Solids*, 10812865211055504.

Sun, L.; Gao, H.; Pan, S.; and Wang, J.-X. 2020. Surrogate modeling for fluid flows based on physics-constrained deep learning without simulation data. *Computer Methods in Applied Mechanics and Engineering*, 361: 112732.

Wang, S.; Sankaran, S.; and Perdikaris, P. 2022. Respecting causality is all you need for training physics-informed neural networks. *arXiv preprint arXiv:2203.07404*.

Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; and Fu, Y. 2018. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2472–2481.

Zhu, Q.; Liu, Z.; and Yan, J. 2021. Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks. *Computational Mechanics*, 67(2): 619–635.