This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

SAND2021-15812C

UC San Diego
Structural Engineering
JACOBS SCHOOL OF ENGINEERING

ATA ENGINEERING, INC.

ENSEMBLE OF NUMERICS-INFORMED NEURAL NETWORKS WITH EMBEDDED HAMILTONIAN CONSTRAINTS FOR MODELING NONLINEAR STRUCTURAL DYNAMICS

David A. Najera[1,2], and **Dr. Michael D. Todd**[1]

1. Dept. of Structural Engineering, University of California San Diego
2. ATA Engineering, Inc.

IMAC XL – Orlando, FL
February 2022

*This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned su[…]ct DE-NA0003525

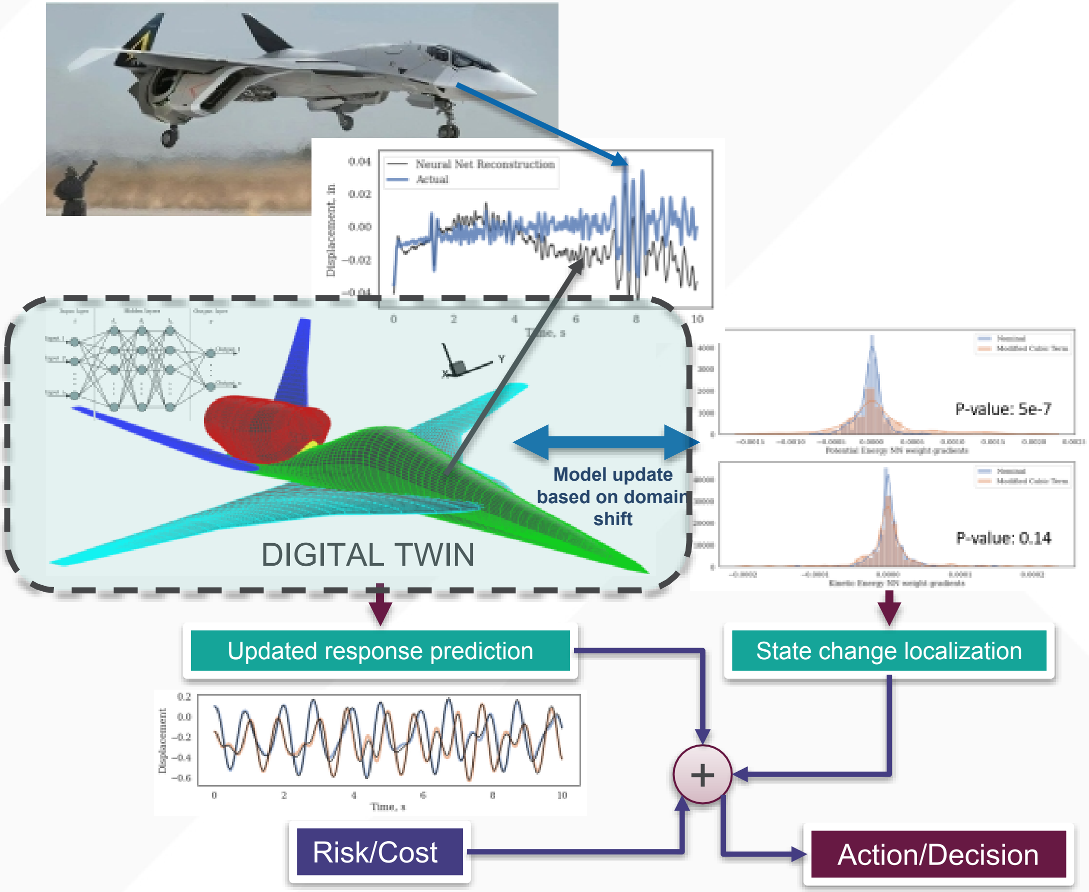
Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525.

RESEARCH OBJECTIVES – A PHYSICS-CONSTRAINED DIGITAL TWIN MODEL

Main objective is to develop a **physics-constrained** digital twin of the system of interest that is a hybrid data-driven, physics-based reduced-order model.

Desired properties of the digital twin:

- Data-driven: no need to specify model parameters or model structure.
- Physics-constrained: network architecture and data flow follows a physics-based structure provided by Hamiltonian mechanics.
- Self-aware: trained model is able to recognize domain shifts in new inputs



PHYSICS-CONSTRAINED ML FRAMEWORK

Why ML? Speed, Differentiability, Learn Unknown Physics, Distill Reusable Modules

The physical constraints chosen are based on Hamiltonian mechanics. This framework was chosen to allow flexibility in the model so that it can handle nonlinearities, and Rayleigh dissipation models (i.e., proportional to velocity).

Because the Euler-Lagrange equations of motion are based on the energy of the system, which are scalar fields, this approach is more computationally efficient than having to construct full state matrices.

A consequence of choosing Hamiltonian mechanics is that the system has to be solved in generalized coordinates. In general, order reduction methods do not result in a generalized coordinate set, so an autoencoder is used to perform coordinate transformation.

Newtonian form of equations of motion

$$\mathrm{M}\ddot{\boldsymbol{x}} + \mathrm{C}\dot{\boldsymbol{x}} + (\mathrm{K} + \mathrm{Knl})\boldsymbol{x} + \boldsymbol{f}_{NL} = \boldsymbol{f}_{ext}$$

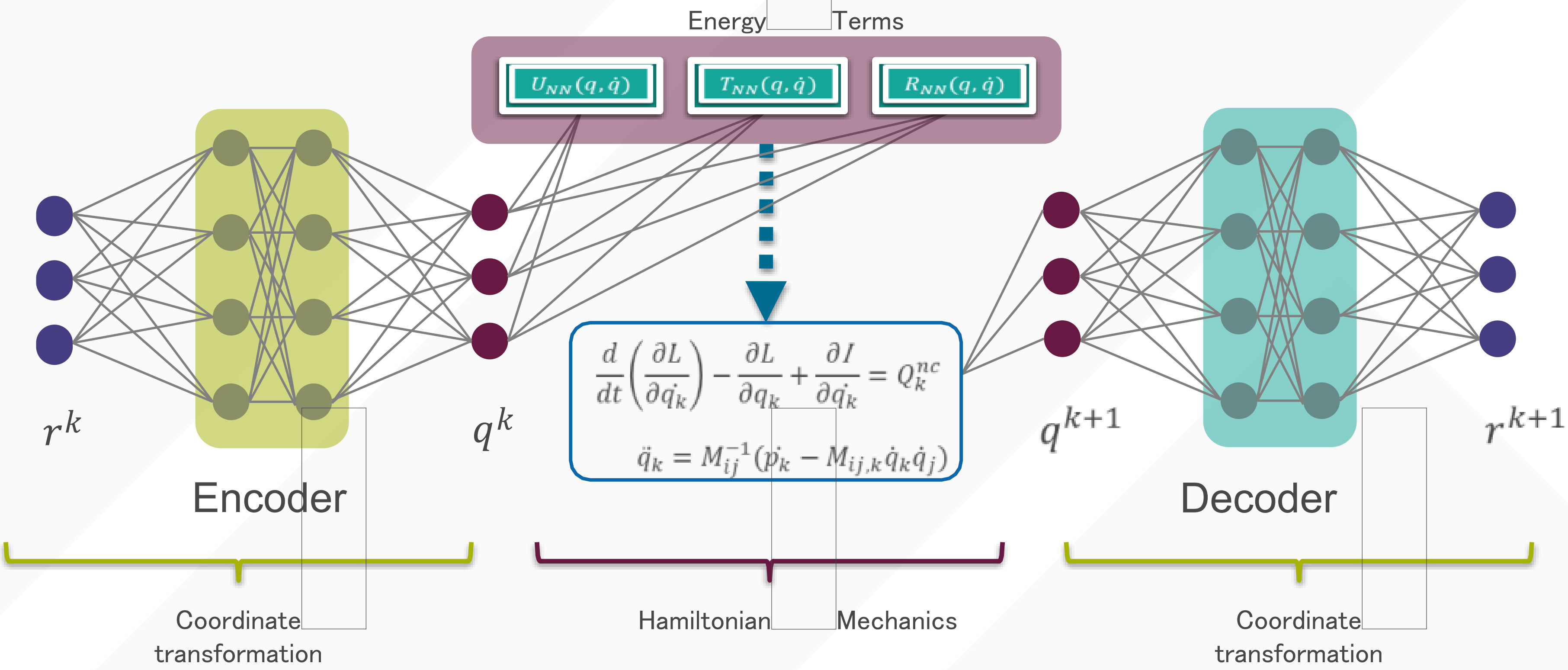
Equivalent

Euler-Lagrange equations of motion

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_k}\right) - \frac{\partial L}{\partial q_k} + \frac{\partial I}{\partial \dot{q}_k} = Q_k^{nc}$$

Other physics-based pROMs can be integrated as inductive bias kernels to the network.

NEURAL NETWORK ENSEMBLE ARCHITECTURE



PHYSICS-CONSTRAINED ML FRAMEWORK

Start with $(r, \dot{r}) \rightarrow E_{NN}(r, \dot{r}) \rightarrow (q, \dot{q})$

$$U(q) = U_{NN}(q, \dot{q}) \qquad T(q) = T_{NN}(q, \dot{q}) \qquad I(q) = R_{NN}(q, \dot{q})$$

$$L(q, \dot{q}) = T_{NN}(q, \dot{q}) - U_{NN}(q, \dot{q})$$

$$p_k = \frac{\partial L(\mathbf{q}, \dot{\mathbf{q}})}{\partial \dot{q}_k}$$

$$H(\mathbf{p}, \dot{\mathbf{q}}) = \mathbf{p} \cdot \dot{\mathbf{q}} - L(\mathbf{q}, \dot{\mathbf{q}})$$

$$\dot{p_k} = -\frac{\partial H}{\partial q_k} + Q_k^{nc} - \frac{\partial I}{\partial \dot{q}_k}$$

$$\ddot{q}_k = M_{ij}^{-1}(\dot{p_k} - M_{ij,k}\dot{q}_k\dot{q}_j)$$

$$Q_k^{nc} = \sum_{i=1}^{N} F_i \cdot \frac{\partial D_{NN\,i}}{\partial q_k} \qquad M_{ij} = \frac{\partial T_{NN}}{\partial \dot{q}_i \partial \dot{q}_j}$$

Pseudo-inverse

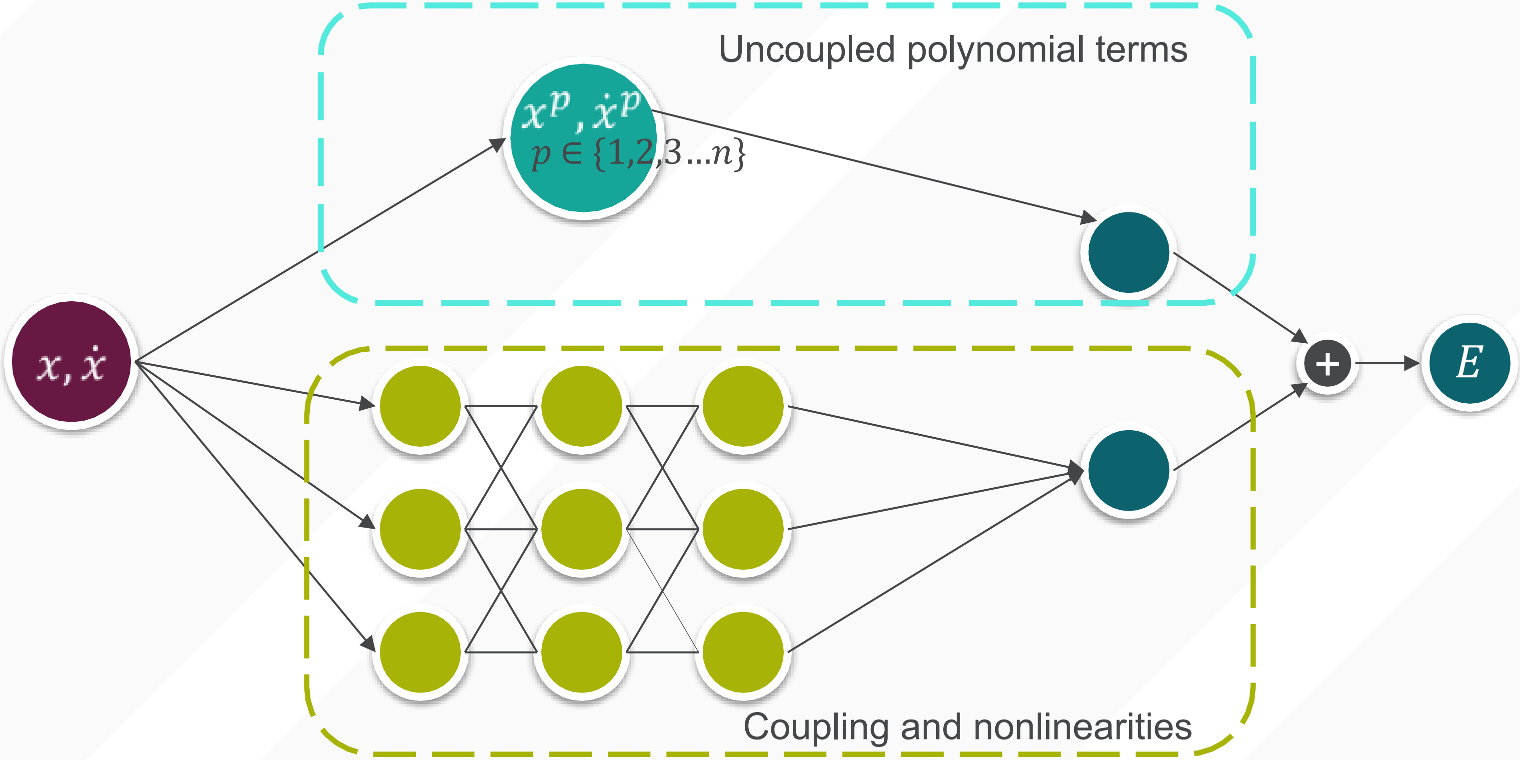
$$Loss_{EL} = \frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}_k}\right) - \frac{\partial L}{\partial q_k} + \frac{\partial I}{\partial \dot{q}_k} - Q_k^{nc}$$

$$Loss_F = \sum_{j=0}^{s} \alpha_j \begin{pmatrix} \mathbf{q}_{k+j} \\ \dot{\mathbf{q}}_{k+j} \end{pmatrix} - h \sum_{j=0}^{s} \beta_j \begin{pmatrix} \dot{\mathbf{q}}_{k+j}(\mathbf{t}_{k+j}) \\ \ddot{\mathbf{q}}_{k+j}(\mathbf{t}_{k+j}) \end{pmatrix}$$

Numerics-informed

$(\dot{q}, \ddot{q}) \rightarrow D_{NN}(q, \dot{q}) \rightarrow (\dot{r}, \ddot{r})$

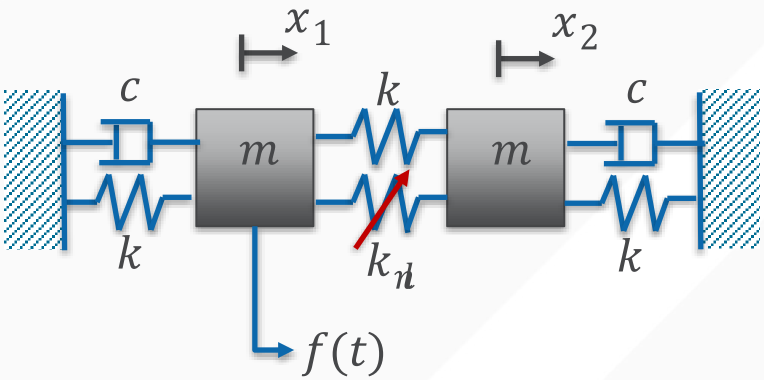
MULTILAYER PERCEPTRON NETWORKS FOR ENERGY



Energy networks

$$E = \sum_p \alpha_p x^p + \beta_p \dot{x}^p + \sum_k \theta_k g^{(k)}(x, \dot{x})$$

EXAMPLE: 2DOF OSCILLATOR WITH CUBIC NONLINEARITY

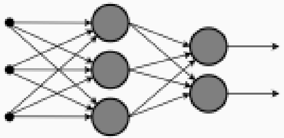


$$T = \frac{1}{2}m(\dot{x}_1^2 + \dot{x}_2^2)$$

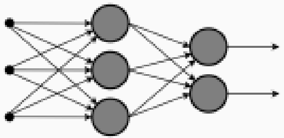
$$I = \frac{1}{2}c(\dot{x}_1^2 + \dot{x}_2^2)$$

$$U = \frac{1}{2}k(x_1^2 + x_2^2) + \frac{1}{4}k_{nl}(x_2 - x_1)^4 + \frac{1}{2}k(x_2 - x_1)^2$$

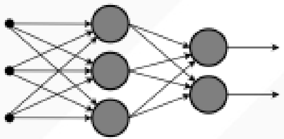
$$T(x, \dot{x}) = T_{NN}(x, \dot{x})$$



$$I(x, \dot{x}) = I_{NN}(x, \dot{x})$$



$$U(x, \dot{x}) = U_{NN}(x, \dot{x})$$

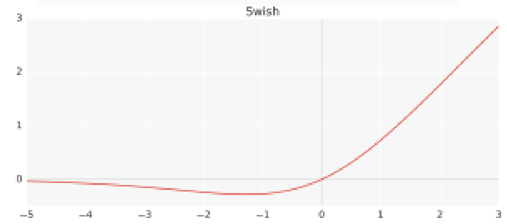


MLP Architecture:

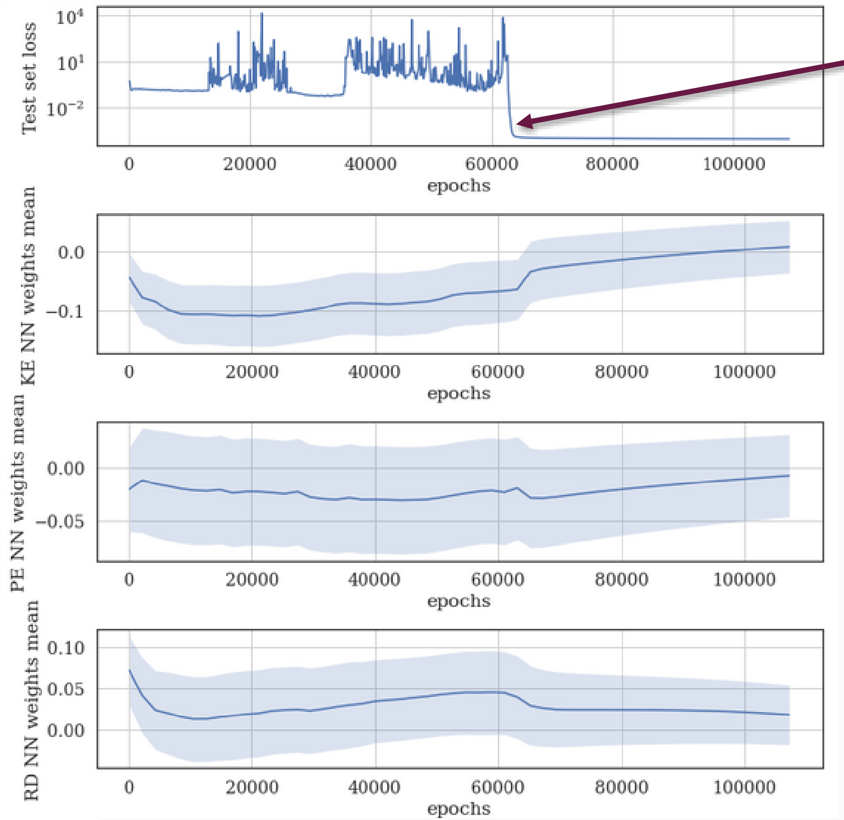
- 4 layers
- 8 neurons
- Swish activation

Training:

- Adam optimizer, ~80,000 epochs



NEURAL NETWORK TRAINING DETAILS

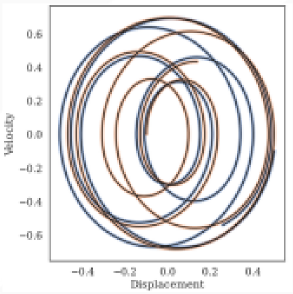
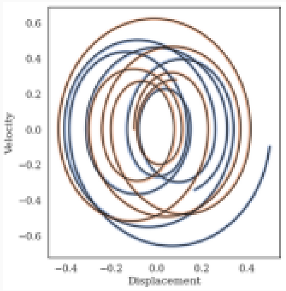
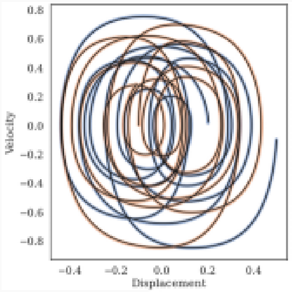
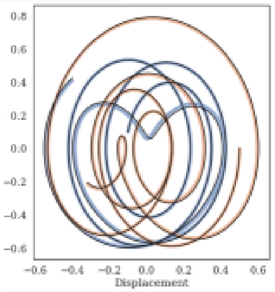
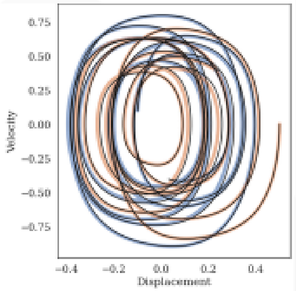


Typical training behavior observed

Trained with a single realization and tested with different initial conditions and/or loads.

Training took, on average, 60 minutes on 1 CPU, including code compilation.

RESULTS OF A SIMPLE 2DOF OSCILLATOR

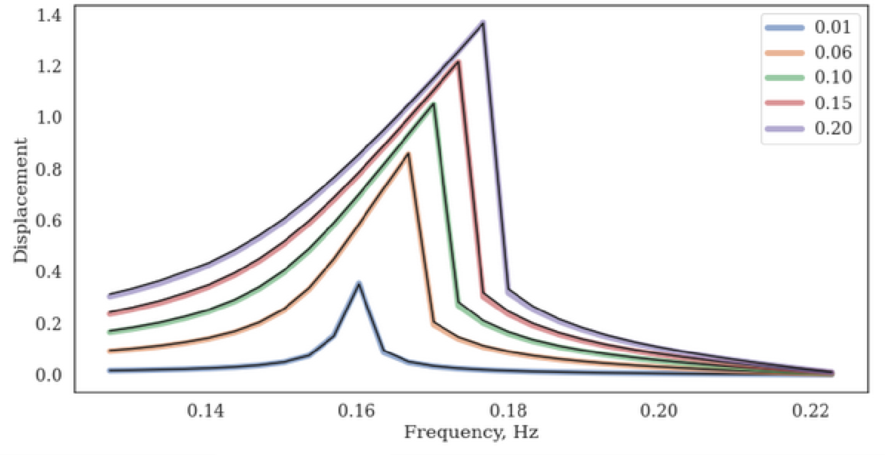
	Linear Undamped Unforced	Linear Damped Unforced	Nonlinear k Nonlinear c Unforced	Linear Damped Forced	Nonlinear Damped Forced
Runtime per evaluation (s)	3.9E-5	4.1E-5	1.9E-4	1.7E-4	2.3E-4
Displacement MSE	1.0E-9	4.2E-8	2.4E-11	5.6E-7	1.5E-6
Phase space					

Color: Analytical | Black: Neural Net

*Trained with a single realization and tested with different initial conditions and/or loads. Training took, on average, ~60 minutes on 1 CPU.

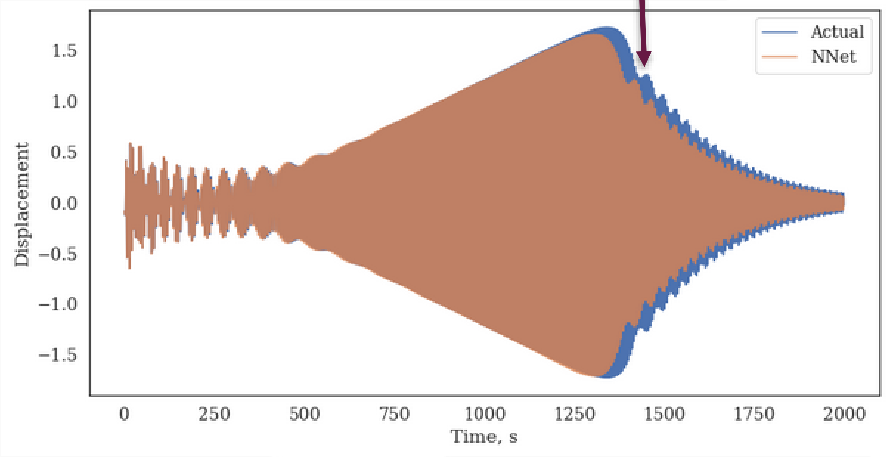
NONLINEAR TRAINED NEURAL NETWORK CAPTURES BIFURCATION

*trained with 0.08 forcing amplitude



Color: Analytical | Black: Neural Net

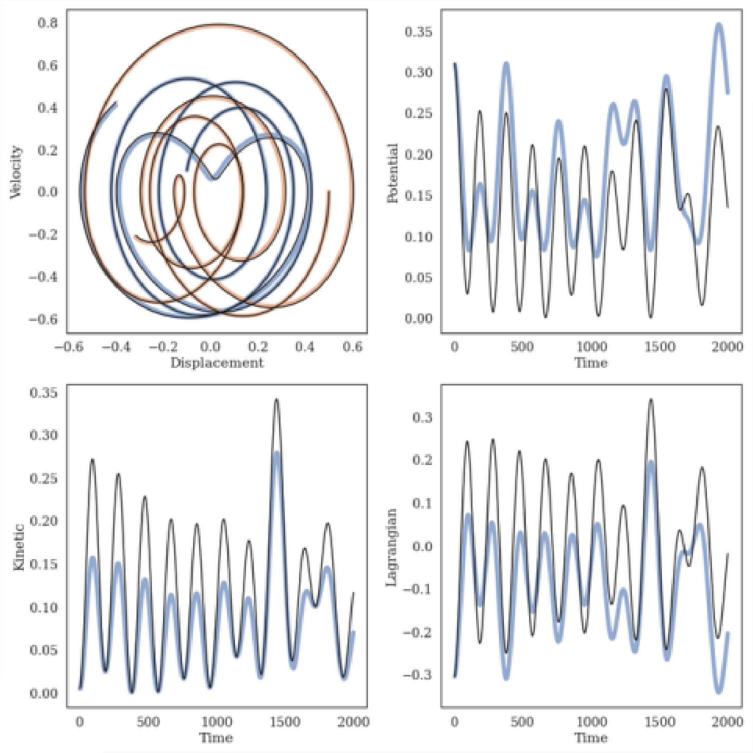
Still some discrepancies present due to this region not being explored at all during training.



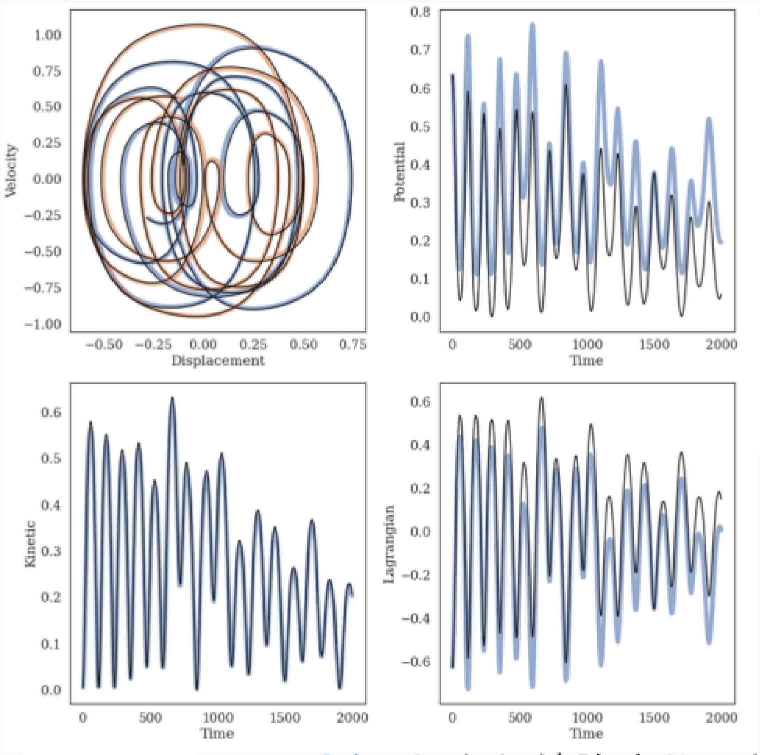
Model was trained with a single example from random vibration response. The trained model was able to capture nonlinear behavior at larger amplitudes than the training data.

DISCREPANCIES IN ENERGY PREDICTIONS

Linear Damped Forced



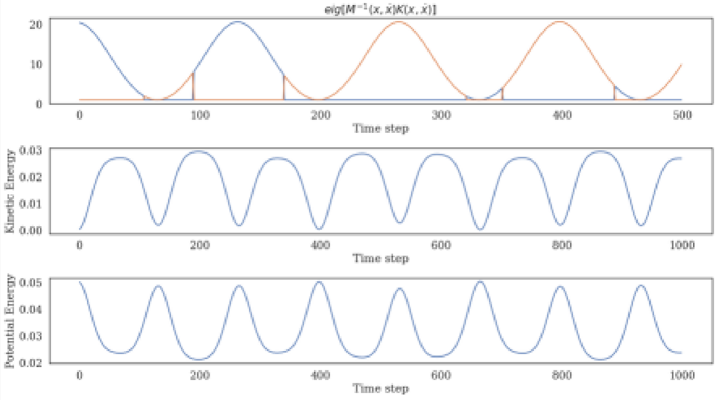
Nonlinear Damped Forced



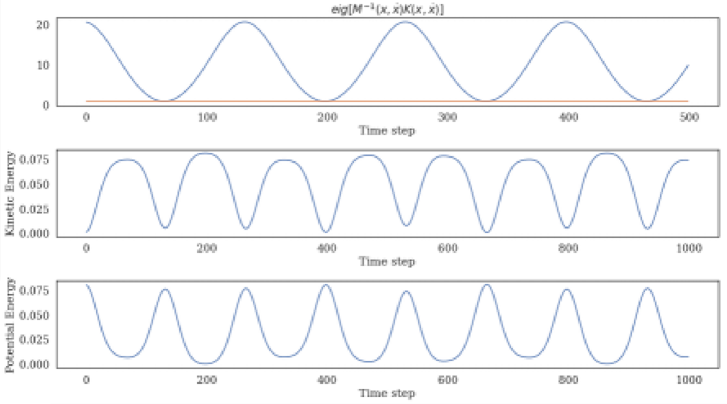
Color: Analytical | Black: Neural Net

WHAT IS THE NETWORK LEARNING?

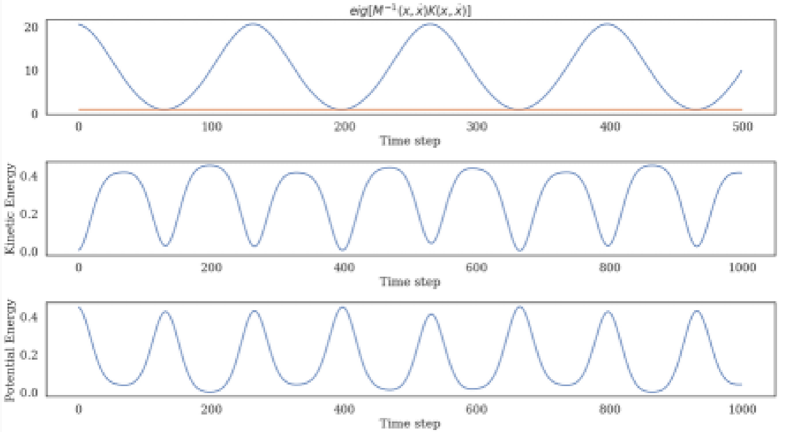
A. Free-form



B. General form of the energy is prescribed

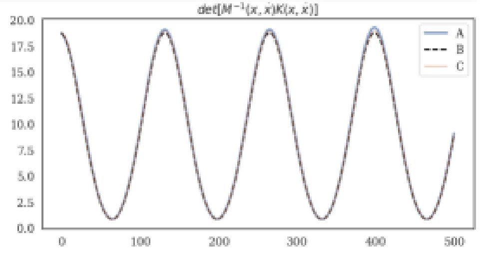


C. General form of the energy is prescribed + linear stiffness parameter is prescribed



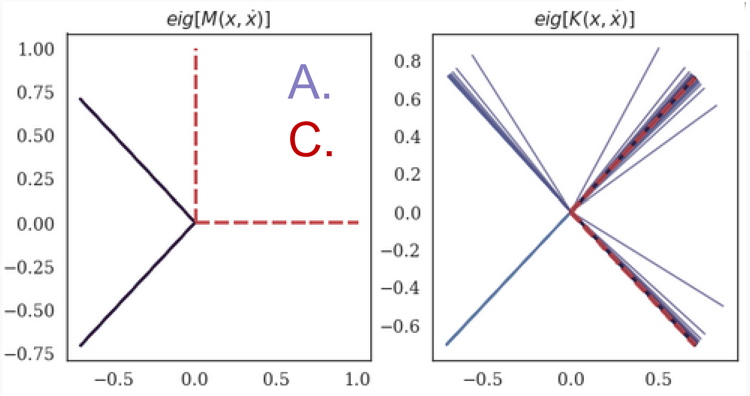
This case reproduces energies and eigenvalues exactly.

Even though energy terms are not reconstructed exactly, the state space representation ($M^{-1}K$) learned is the same for all systems.

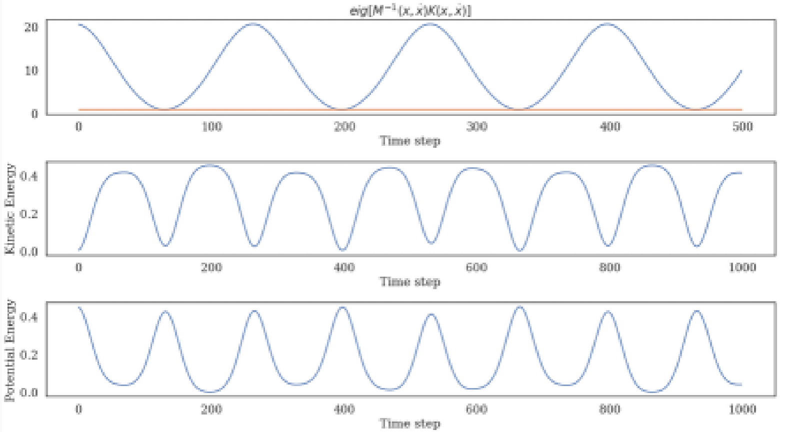


WHAT IS THE NETWORK LEARNING?

Visualization of **M** and **K** rotation

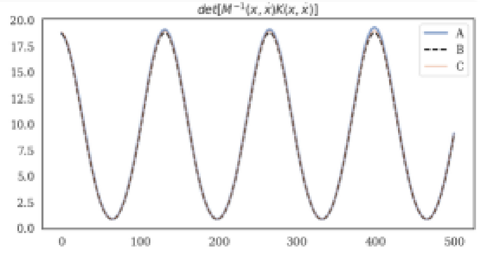


C. General form of the energy is prescribed + linear stiffness parameter is prescribed



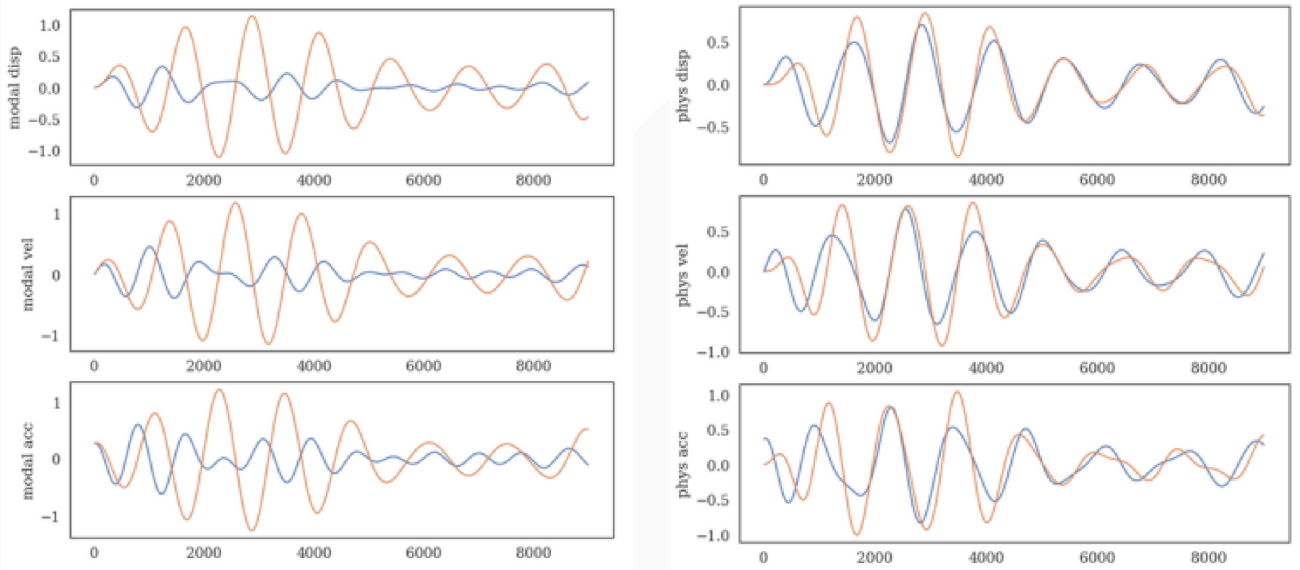
This case reproduces energies and eigenvalues exactly.

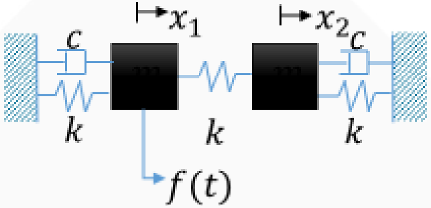
Even though energy terms are not reconstructed exactly, the state space representation ($M^{-1}K$) learned is the same for all systems.



WHAT IF SYSTEM IS NOT IN GENERALIZED COORDINATES?

Start with Modal Coordinates, and find transformation to another set of generalized coordinates.



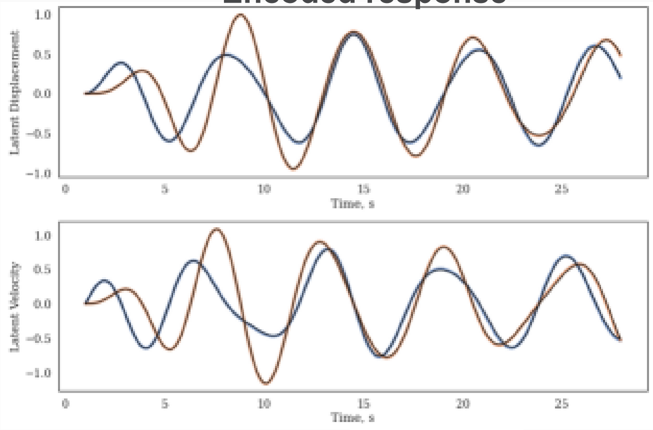


Color: Analytical | Black: Neural Net

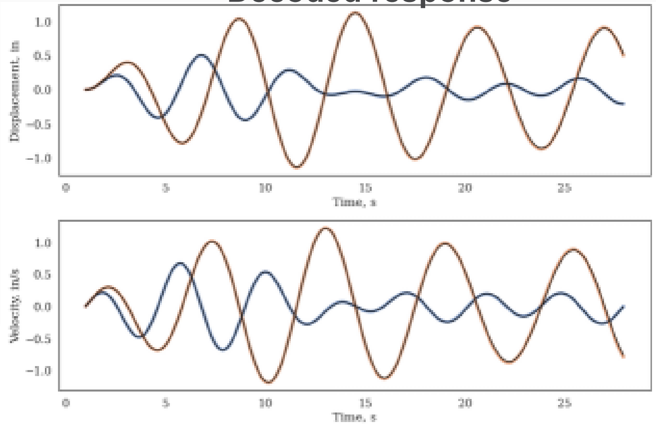
AUTOENCODER LEARNS MODAL TRANSFORMATION

Color: Analytical | Black: Neural Net

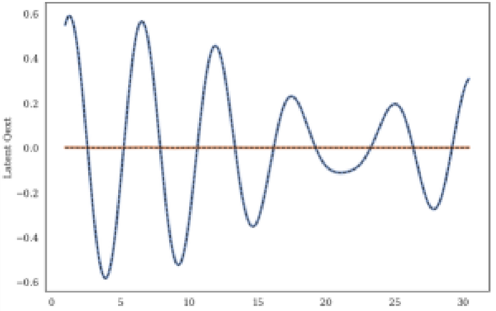
Encoded response

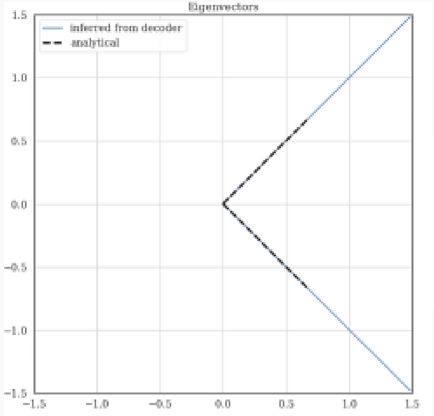


Decoded response

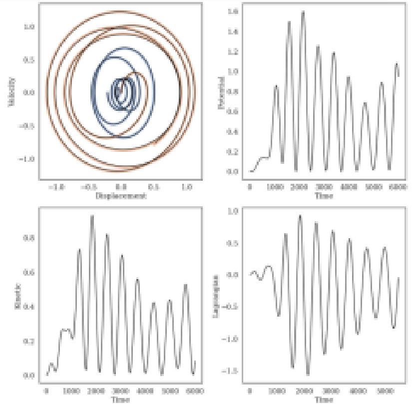


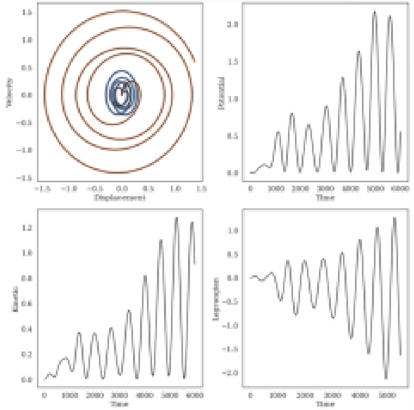
Encoded external force



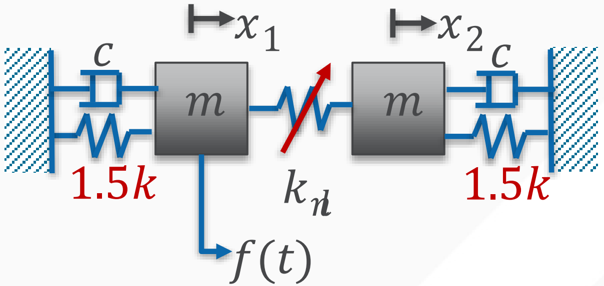


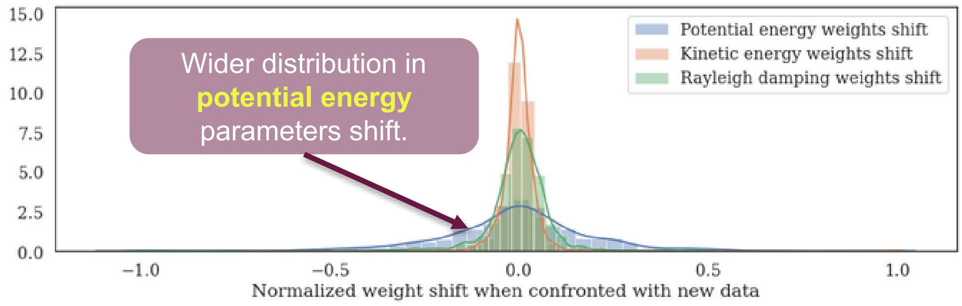
Network is robust to different forcing functions

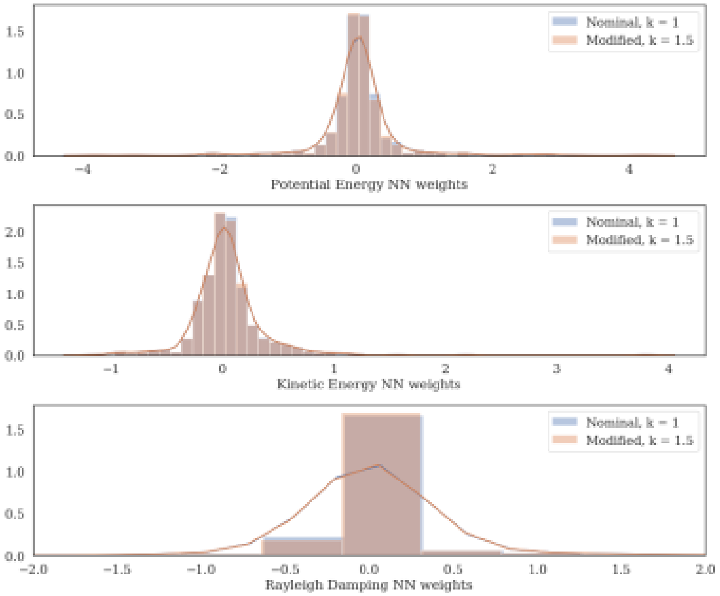


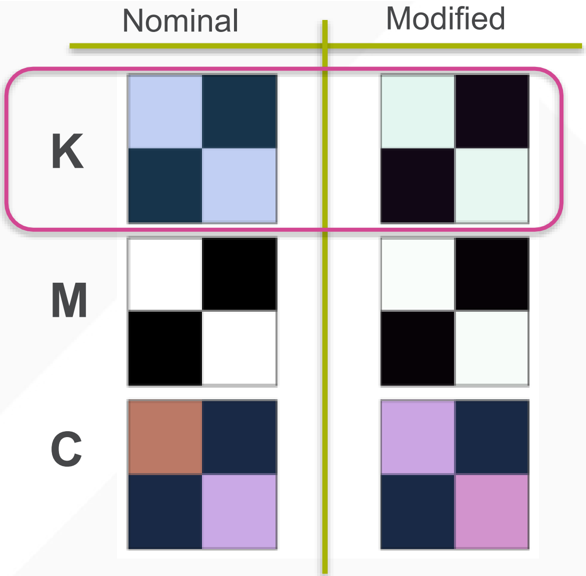


LEVERAGING THE NETWORK TO IDENTIFY SYSTEM CHANGES









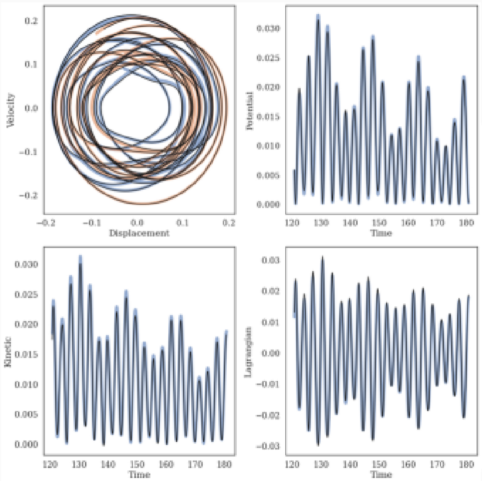
CONCLUSION AND FUTURE WORK

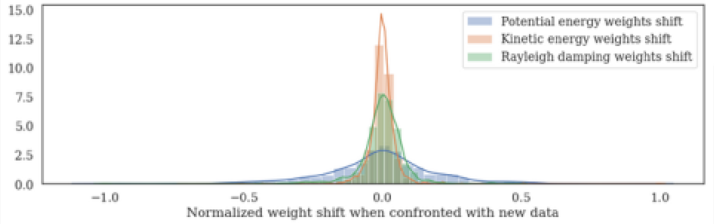
A framework for data-driven, physics-constrained, numerics-informed neural networks was established based on Hamiltonian mechanics.

The framework combines physical and mathematical structure to regularize the network and provide a physically meaningful parameterization.

This work demonstrates that the framework can be used to recover the general system state dynamics from data and feasibility of using ML model weight shifting for domain shift detection

This research represents the first step towards a predictive ML digital twin model that can be incorporated in a general structural health monitoring system.

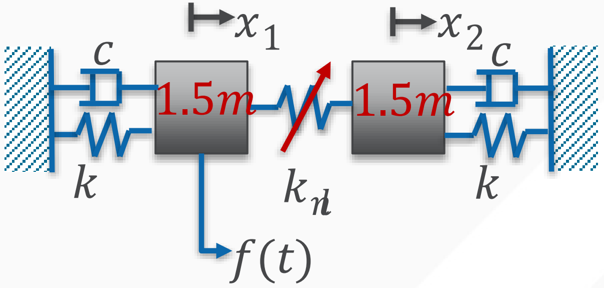


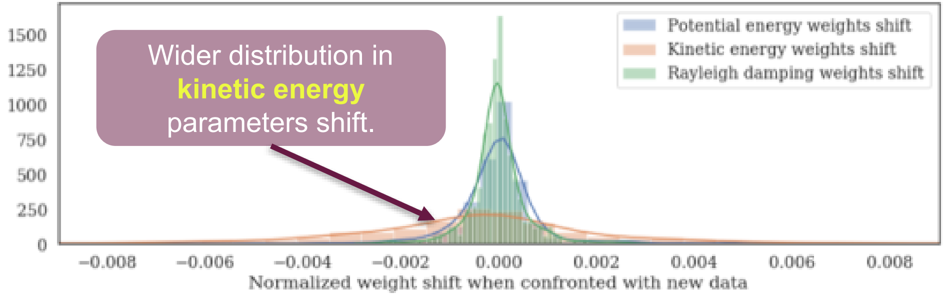


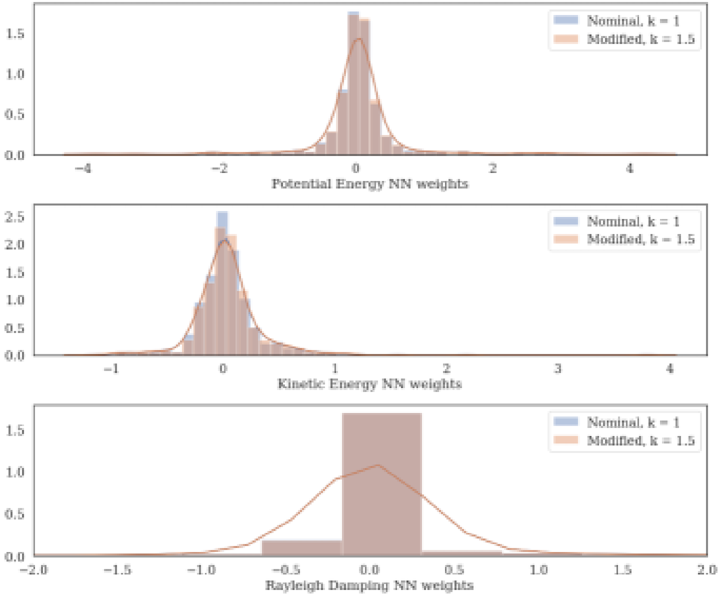
*This paper describes objective technical results and analysis. Any subjective views or opinions that might be expressed in the paper do not necessarily represent the views of the U.S. Department of Energy or the United States Government. Sandia National Laboratories is a multimission laboratory managed and operated by National Technology & Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International Inc., for the U.S. Department of Energy’s National Nuclear Security Administration under contract DE-NA0003525

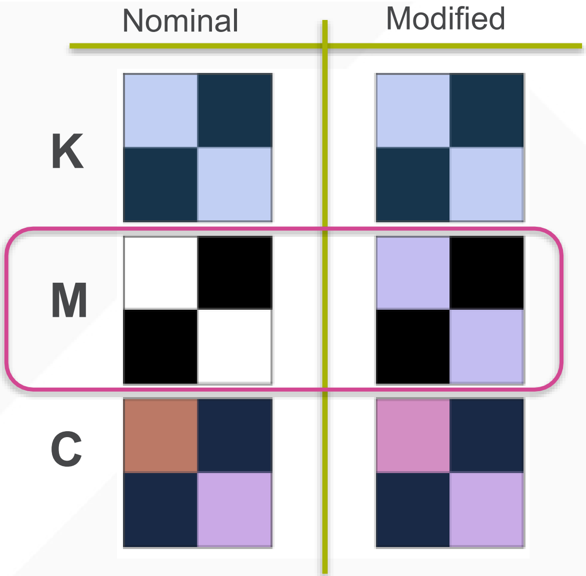
BONUS SLIDES

LEVERAGING THE NETWORK TO IDENTIFY SYSTEM CHANGES

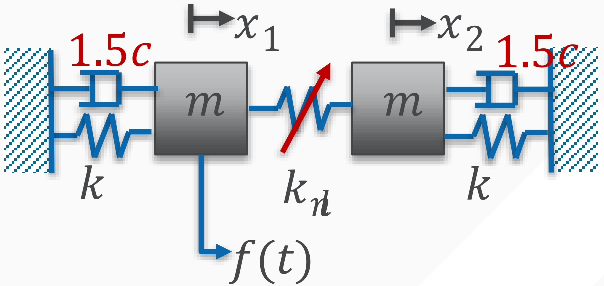


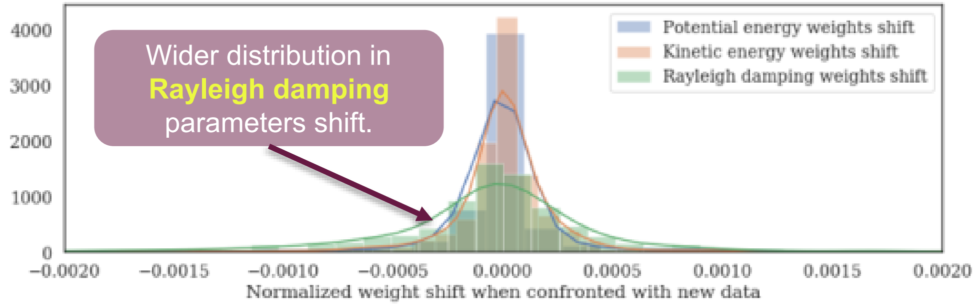


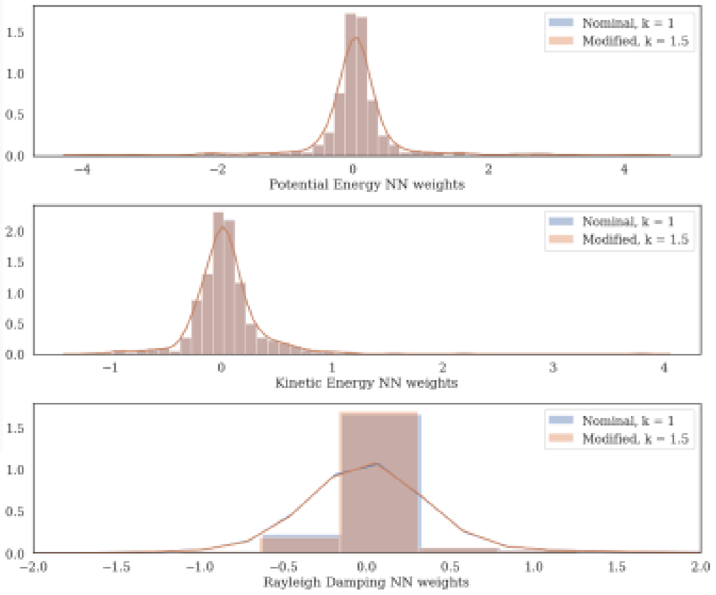


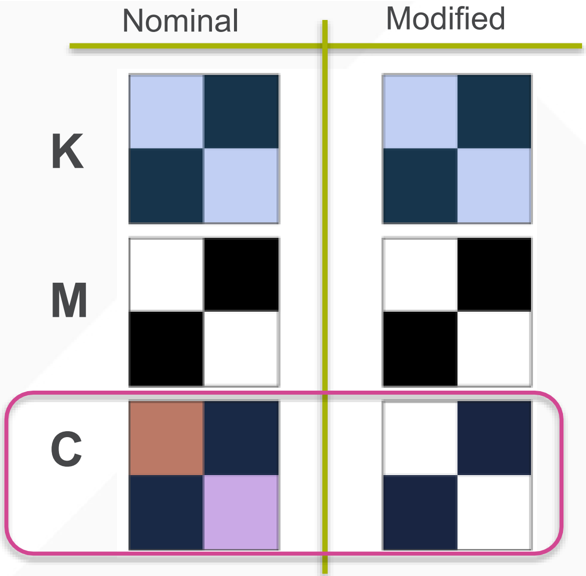


LEVERAGING THE NETWORK TO IDENTIFY SYSTEM CHANGES









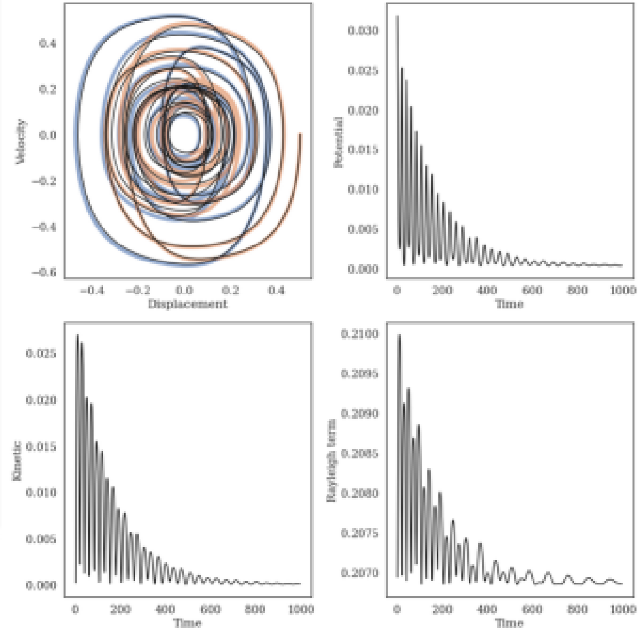
ADDING EXPLICIT PARAMETRIC DEPENDENCE TO THE NETWORKS

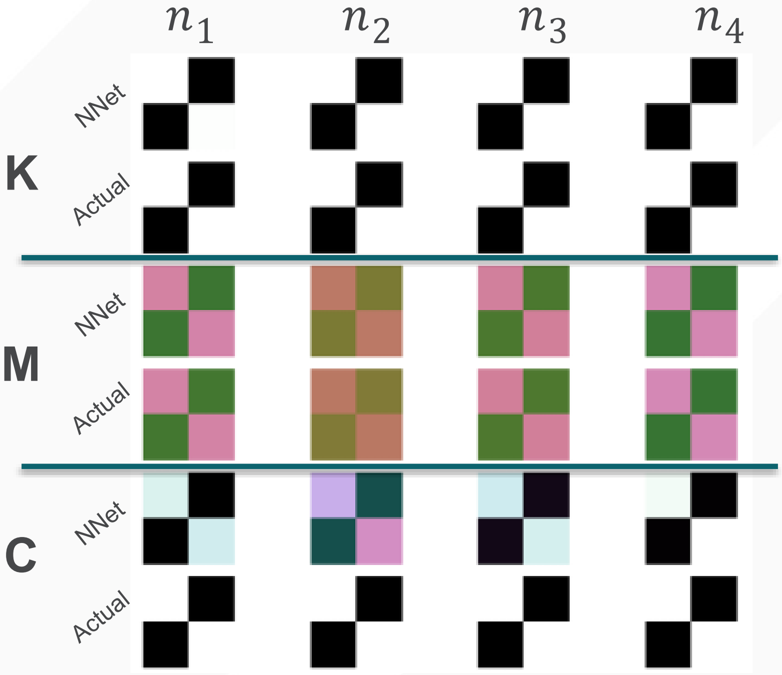
Color: Analytical | Black: Neural Net

$$U(q) = U_{NN}(q, \dot{q}, \theta) \quad T(q) = T_{NN}(q, \dot{q}, \theta) \quad I(q) = R_{NN}(q, \dot{q}, \theta)$$

$$\theta = [m k c k_{nl}] \qquad m = (0.7,1.5), k = (0.7,1.5), c = (0.01,0.1), k_{nl} = (5,15)$$

Test on new random parameter set

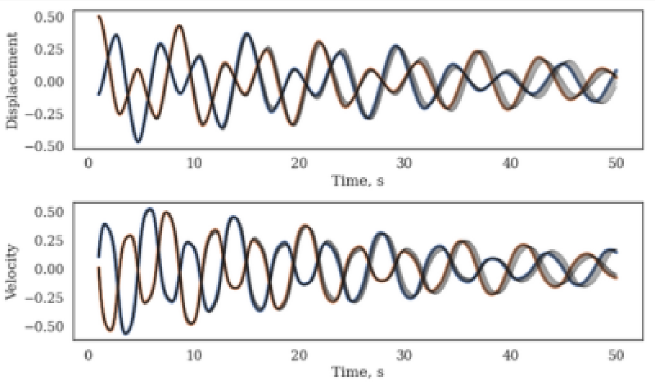




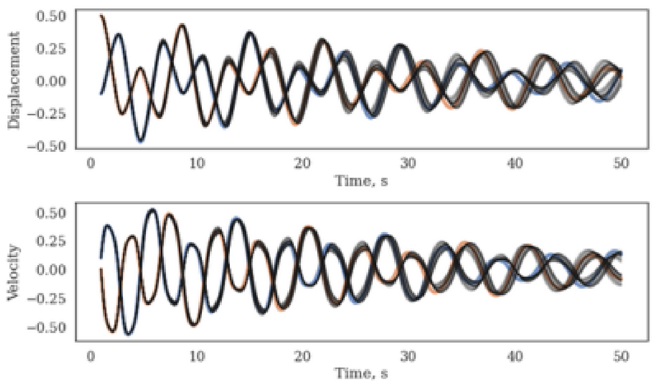
DO THE NETWORKS PROPAGATE PARAMETRIC UNCERTAINTY?

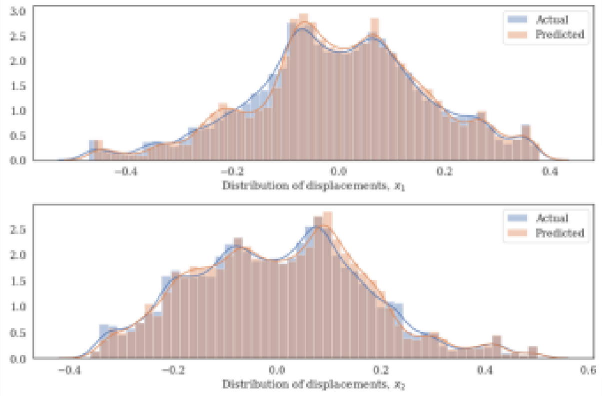
Perturbations of k

Actual



NNet





Self-organizing map weights for random sample collection