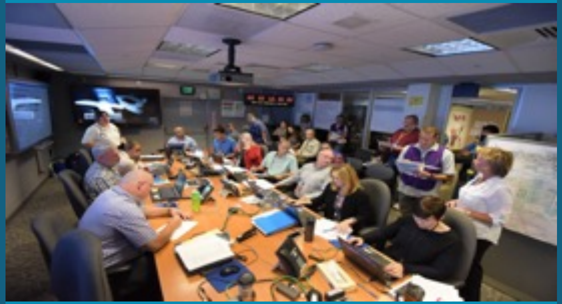
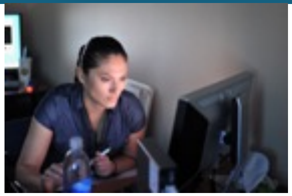
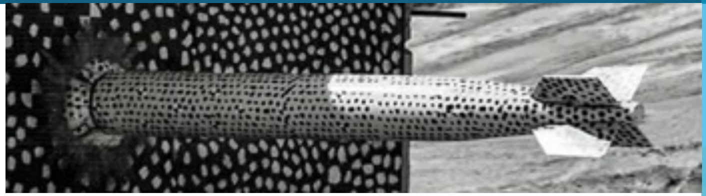
Analysis of Neural Networks as Random Dynamical Systems

Project # 21-0528







PI: Khachik Sargsyan, org. 8351
PM: Janine Bennett, org. 8739

Team: Joshua Hudson (8351),
Marta D'Elia (8734), Habib Najm (8300)

Continuation review: May 12, 2022

FY21-24, $500K/yr

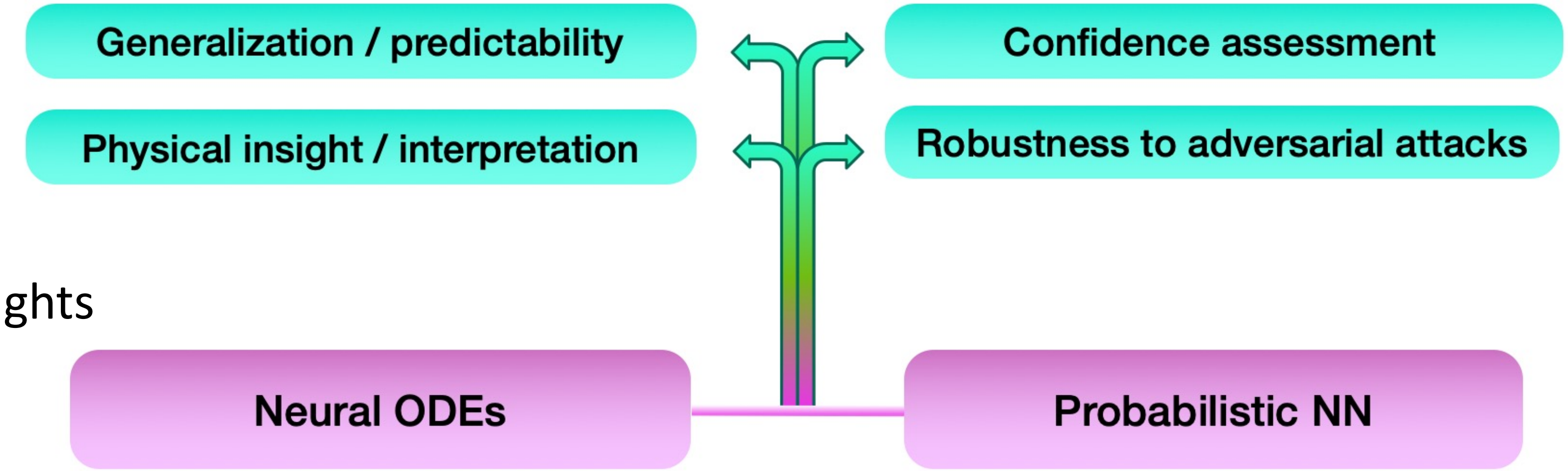
IDEA SUMMARY: Overview

No major deviations from the plan

What: Analyze the performance of neural networks (NNs)
[training, generalization, predictive confidence] from dynamical and probabilistic viewpoints.

State of the art: Despite all the success, there are many recognized challenges and unknowns in NN behavior.

Why now:
There is a lot of accumulated knowledge from ODE and UQ; prime time to build on these insights



Take advantage of 'continuous' limit of residual neural networks

Probabilistic viewpoint opens even more opportunities

Encouragement: a few recent papers at the intersection of Bayesian and Neural ODEs

TECHNICAL ACCOMPLISHMENTS: OVERVIEW

M1 (09/2021): Reduced NODE in deterministic setting — done

Reduction by removing the fast dynamics via regularizing with stiffness

- *Pres. At RAMSES (int. conf.)*
- *Paper to be subm. to JMLMC in June*

Integral NODEs

Paper in review in JCP

M2 (09/2022): Training of regularized probabilistic NODEs — in progress

ResNet regularization via weight parameterization

Loss surface analysis, probabilistic augmentation

Collaboration with Emory U

Current postdoc, + new postdoc

M3 (09/2023): Quantifiable improvement on exemplar applications — in progress

Climate modeling, E3SM Land Model — *BER-funded interest*

Materials science — *FES-funded interest*

Catalytic chemistry — *BES-funded interest*

ResNet and NODE in a regression setting (supervised ML)

ResNet (discrete)

$$\begin{cases} \boldsymbol{x}_1 = \boldsymbol{x} + \boldsymbol{\alpha}_0 \boldsymbol{\sigma}(\boldsymbol{W}_0 \boldsymbol{x}_0 + \boldsymbol{b}_0) \\ \vdots \\ \boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \boldsymbol{\alpha}_n \boldsymbol{\sigma}(\boldsymbol{W}_n \boldsymbol{x}_n + \boldsymbol{b}_n) \\ \vdots \\ \boldsymbol{y} = \boldsymbol{x}_{L-1} + \boldsymbol{\alpha}_{L-1} \boldsymbol{\sigma}(\boldsymbol{W}_{L-1} \boldsymbol{x}_{L-1} + \boldsymbol{b}_{L-1}) \end{cases}$$

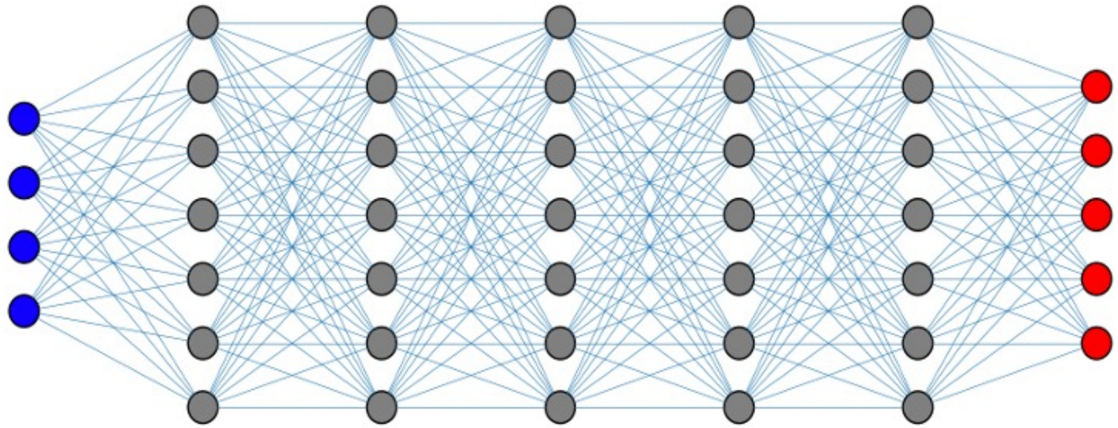
Neural ODE (continuous)

$$\frac{d\boldsymbol{x}}{dt} = \boldsymbol{\sigma}(\boldsymbol{W}(t)\boldsymbol{x} + \boldsymbol{b}(t))$$

$$\boldsymbol{x}(0) = \boldsymbol{x} \qquad \boldsymbol{x}(T) = \boldsymbol{y}$$

$\boldsymbol{x} = \boldsymbol{x}_0$

Input



$\boldsymbol{y} = \boldsymbol{x}_T$

Output

gitlab-ex.sandia.gov/ksargsy/nnrds

In-house codes for both discrete and continuous cases

TA: Foundational capabilities impacting multiple applications

Predictive capability of Neural Networks (NNs) hinges on **generalization** (ability to predict well outside training data).

Regularization of NNs as a way to achieve generalization.

- ✓ Stiffness Penalization
- ✓ Weight Parameterization
- ✓ Probabilistic Weights

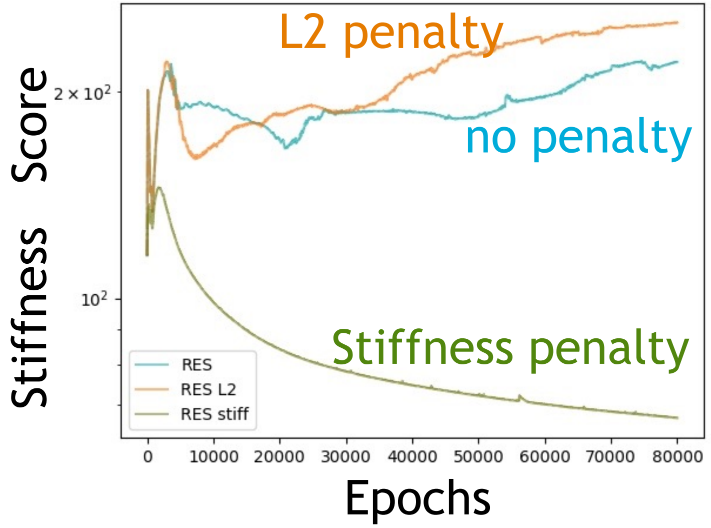
Methods

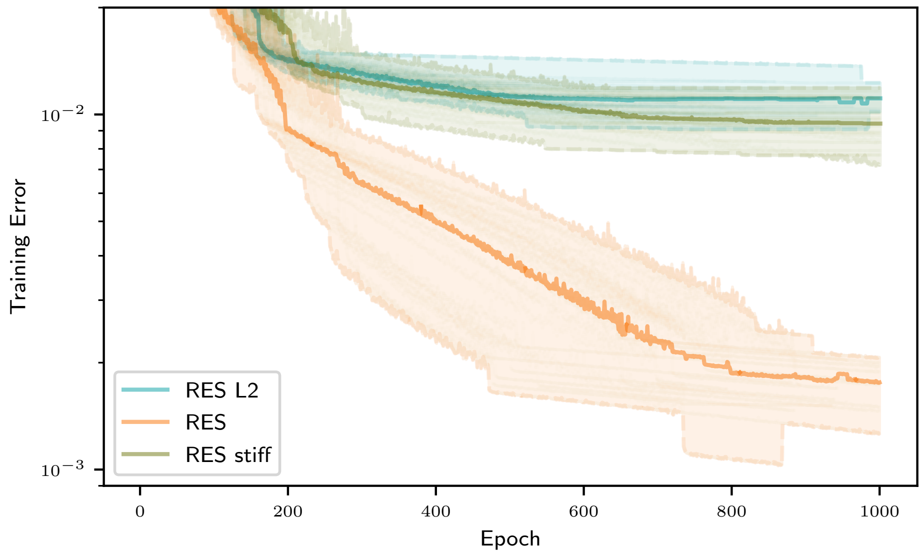
- ✓ Climate Land Modeling
- ✓ Catalytic Chemistry
- ✓ Materials Science

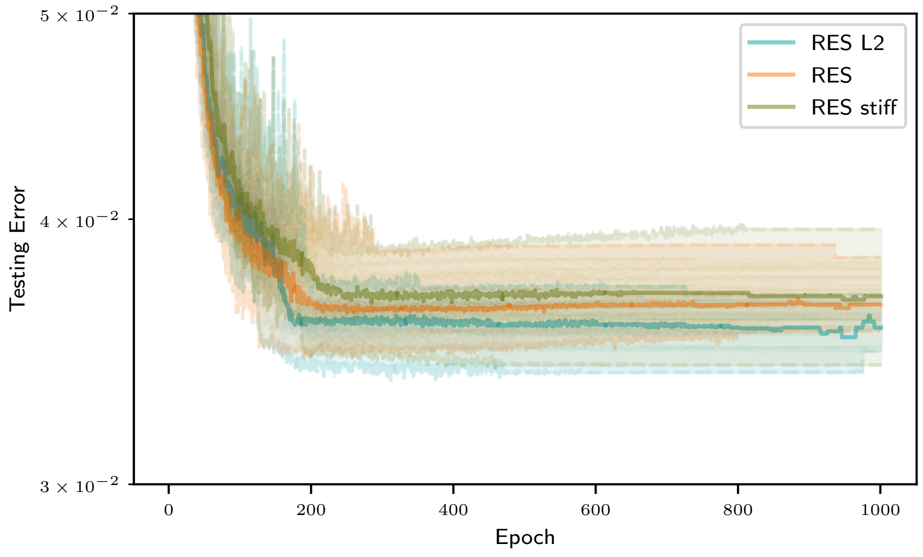
Applications

TA: Regularization #1: Stiffness penalty

- We defined a stiffness metric for ResNets
- Stiffness penalty regularizes the training and reduces generalization gap
- Demo on climate land model data
- Paper nearing submission to JMLMC

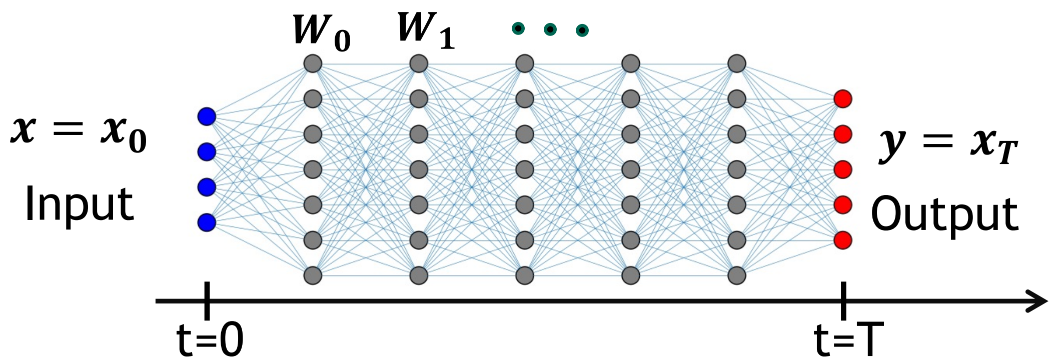






TA: Regularization #2: Weight parameterization

ResNet: $\boldsymbol{x_{n+1}} = \boldsymbol{x_n} + \boldsymbol{\alpha_n \sigma}(\boldsymbol{W_n x_n} + \boldsymbol{b_n})$



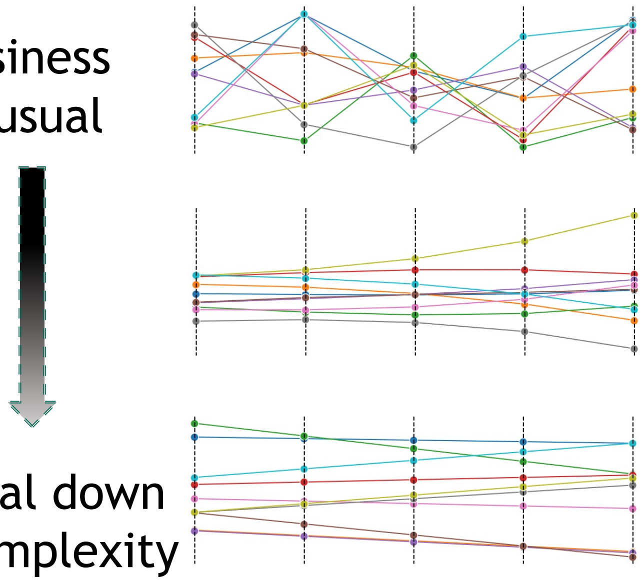
Training for weight matrices $\boldsymbol{W_0}, \boldsymbol{W_1}, \ldots$

Heavily overparameterized, does not generalize well

Parameterize $\boldsymbol{W}(t; \boldsymbol{\alpha})$ and train for $\boldsymbol{\alpha}$'s.

Parameterization of weight functions reduces capacity and improves generalization

Business as usual



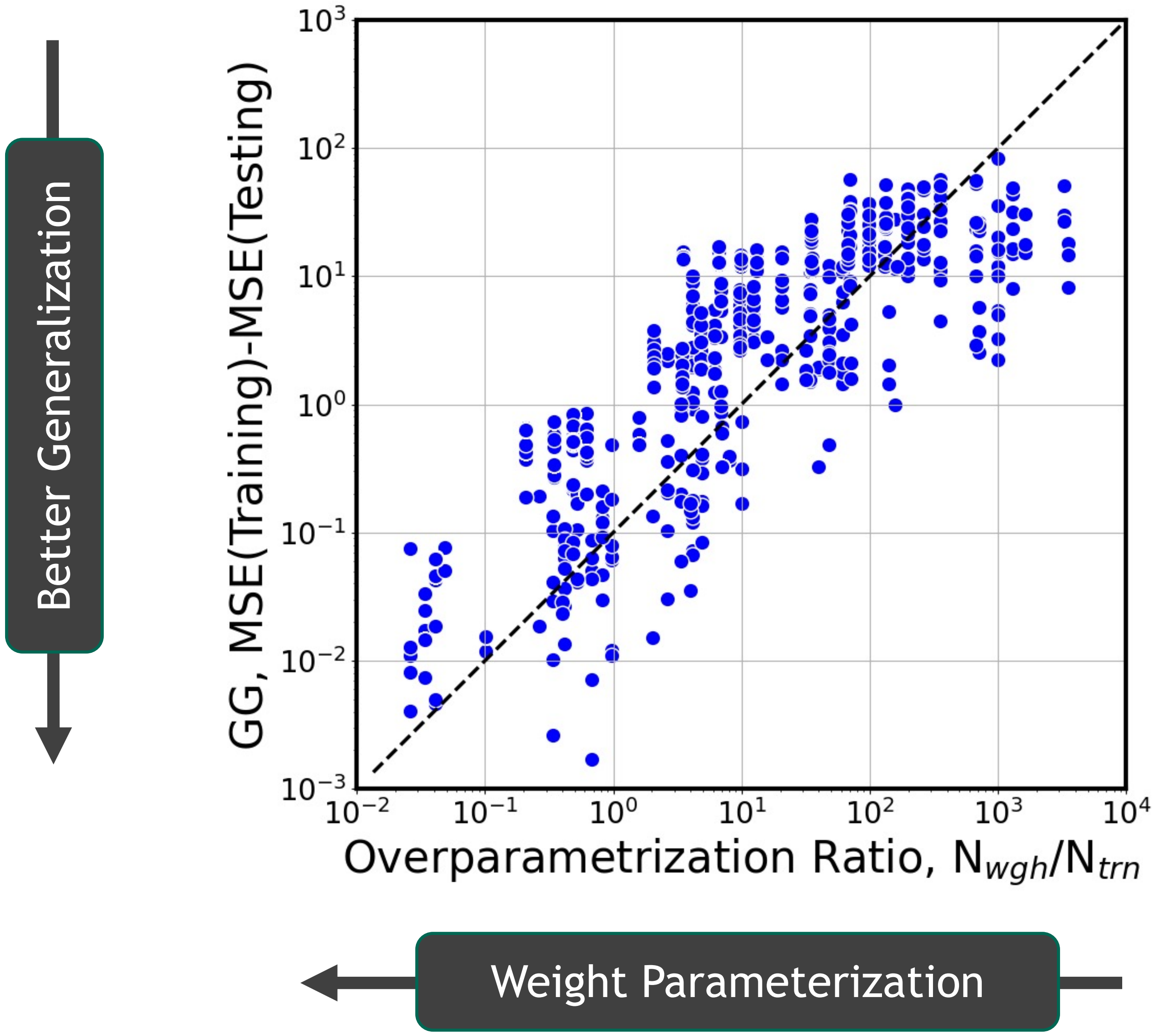
Dial down complexity

NonPar $\boldsymbol{W}(t; \boldsymbol{\alpha})$ = $\boldsymbol{W_{tL/T}}$

Cubic $\boldsymbol{W}(t; \boldsymbol{\alpha})$ = $\boldsymbol{\alpha t^3} + \boldsymbol{\beta t^2} + ..$

Linear $\boldsymbol{W}(t; \boldsymbol{\alpha})$ = $\boldsymbol{\alpha} t + \boldsymbol{\beta}$

TA: Weight parameterization improves generalization



- Generalization Gap correlates with overparameterization

- Weight-parameterized ResNets reduce Generalization Gap

Each dot is a training run with varying weight parameterization functions

TA: Regularization #3: Probabilistic weights

- Conventional NN: training for *deterministic* weight matrices $\boldsymbol{W_0}, \boldsymbol{W_1}, \ldots$
- *Probabilistic* approach: training for probability distributions $\boldsymbol{p}(\boldsymbol{W_0}), \boldsymbol{p}(\boldsymbol{W_1}), \ldots$
- Three classes of options:

Full Bayesian → Approximate Bayesian → Ensemble methods

Full Bayesian

- ➢ Markov chain Monte Carlo (MCMC)
- Typically infeasible for overparameterized NNs
- With weight parameterization loss functions are better behaved (lower-dimensional, fewer symmetries), hence MCMC path more feasible

Approximate Bayesian

- ➢ Variational methods
- Typically underestimates extrapolative predictions

Ensemble methods

- ➢ Heuristic, but works
- Many recent papers viewing deep ensembles as Bayesian approximation

Methods are being added to in-house code **QUiNN**, as a probabilistic wrapper to deterministic ResNets

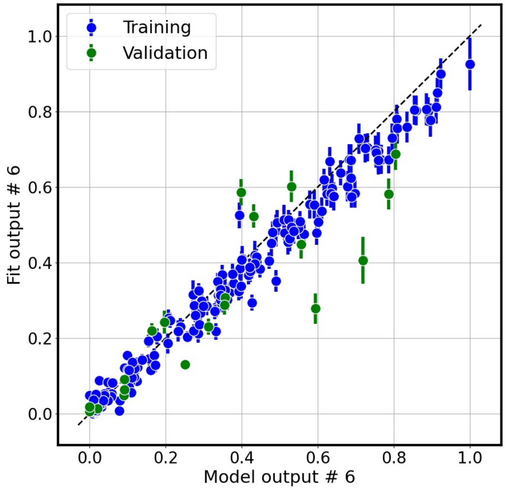
Apps:

- Multiple applications are informing the development of foundational research
- None of these applications have been previously exposed to NN prediction uncertainties, particularly in the context of ResNets and weight parameterization

E3SM Vegetation Dynamics

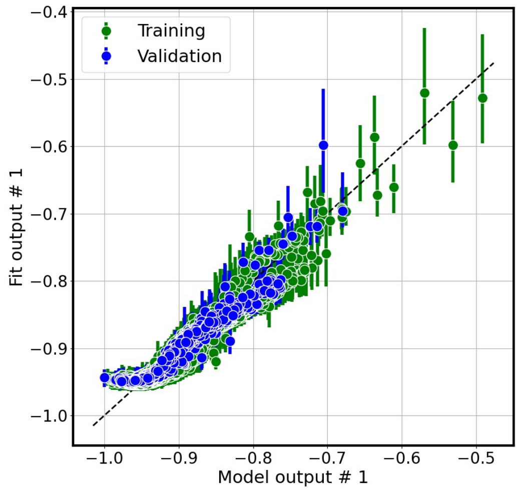
- 15 input parameters
- 10 static output QoIs
- 2K training simulations

E3SM
Energy Exascale
Earth System Model



FitSNAP Entropy Dataset

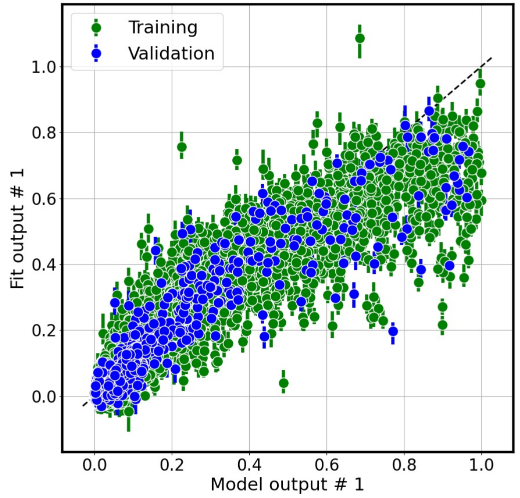
- 30 input bases
- 1 output (Energy/Force/Stress)
- 20K training DFT simulations



CO-on-Pt(111) Adsorbate

- 6 input d.o.f.
- 1 output (Energy)
- 10K training DFT simulations

Exascale Catalytic Chemistry



PROJECT STATUS: No adjustments needed

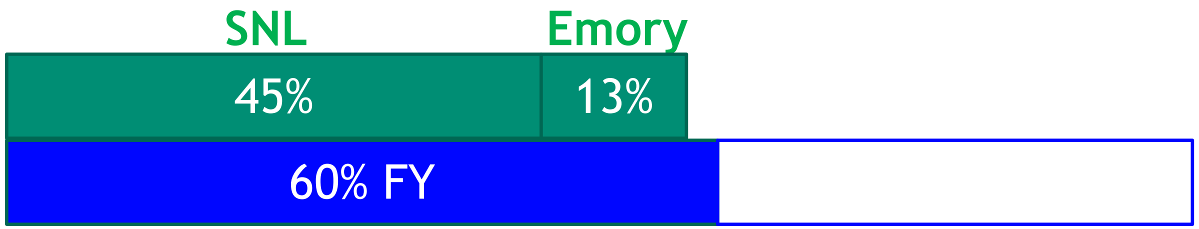
- Y1 (deterministic) milestone is met,
- Y2 (probabilistic) milestone in progress,
- Y3 (application impact) milestone in progress.

- Hired Joshua Hudson in November 2020 via CSRI postdoc announcement,
- Offer pending for the second postdoc, expected to start in summer 2022,
- Involving Oscar Diaz-Ibarra (current postdoc on another project) part time .

- Marta D'Elia leaving SNL in June 2022.

- Subcontract with Prof. Lars Ruthotto (Emory U), who is a leader in the field. Graduate student Haley Rosso started working. Summer visit planned.

Programmatic / Spend plan:



PROJECT PLAN FOR REMAINDER OF FY22 and FY23

No technical risks, only staffing

- Dynamical analysis/stiffness: *[Joshua Hudson]*
  - Finalize/publish stiffness-regularized ResNets
  - ResNet->NODE continuous analogy, applications

- Weight parameterization: *[Emory U, Oscar Diaz-Ibarra]*
  - demonstrate regularization with weight parameterization
  - incorporate sparsity, and/or complex parameterization
  - loss surface and NN model capacity analysis

- Probabilistic NODEs: *[Ushnish Sengupta, postdoc to start in Aug]*
  - demonstrate probabilistic reformulation – e.g. Bayesian NN made feasible by simplified parameterization
  - publishable results with applications from climate, chemistry, mat. sci.

ANTICIPATED OUTPUTS AND TIE TO INVESTMENT AREA CALL

How did this project contribute to the CIS IA strategic goals and objectives?

- CIS Trusted AI RC Thrusts: at least 2/3 directly relevant: Math Foundations and Usability/Trust
- Mission relevance through impact on DOE-wide efforts (BER, FES, BES-funded applications)
- Collaboration with Emory U (Lars Ruthotto) will enhance our capabilities and lay the ground for external proposals

What are the key results from this research that will be useful to other current and future projects?

- Evident practical benefits of ResNets/NODEs in terms of training efficiency, accuracy and generalization
- Augmenting NN predictions with uncertainty is sought after both in climate (ongoing BER work) and chemistry (ongoing BES and FES work)
- Connections to these efforts exist and the team is working with application data to hone the methods

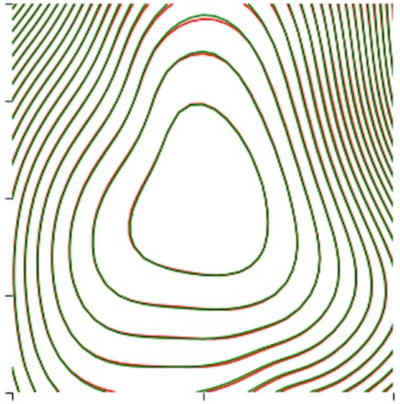
Technology insertion and follow-on funding for potential and realized ROI

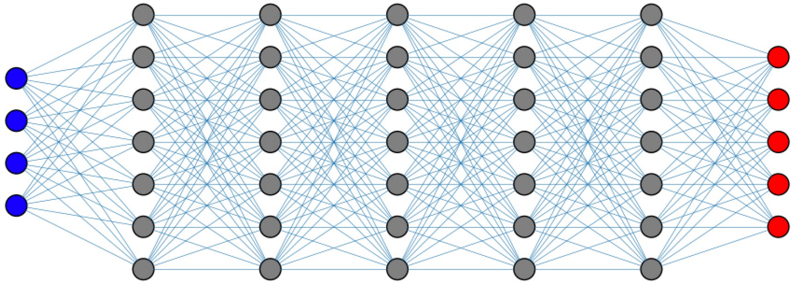
- Fundamental mathematical work, with both code development and impact on real applications
- Strong potential to help Sandia establish leadership in the ML/UQ landscape
- Weight-parameterized ResNets as well as probabilistic NN wrapper software can serve as a base for future funding, both for ASCR (theory) and for application-specific proposals DOE-wide.

Analysis of Neural Networks as Random Dynamical Systems
PI: Khachik Sargsyan (8351) PM: Janine Bennett (8739)

Project Goal(s)

- Develop methods for analysis and regularization of neural networks (NNs)
- Merge probabilistic and ODE viewpoints to improve NN training and accuracy
- Demonstrate on exemplar applications: climate, material science, catalytic chemistry





FY23 Technical Milestones

- Extend dynamical analysis under uncertainty
- Regularize via sparse, probabilistic weight functions
- M3: Tangible impact on application exemplars

Mission Impact

- Bring together theory, modeling, computation, and data, under potentially noisy and adversarial conditions
- Improved NN performance can be key to many mission apps
- Probabilistic NODEs will be a unique capability and will remain mission-relevant for years to come

Transition Plan

- Unique and risky capability: if successful, can lead to further ASCR funding
- Follow-on funding (BER, FES, BES) for applications of interest highly likely
- Software not a direct target but a bi-product which will serve us well for future funding

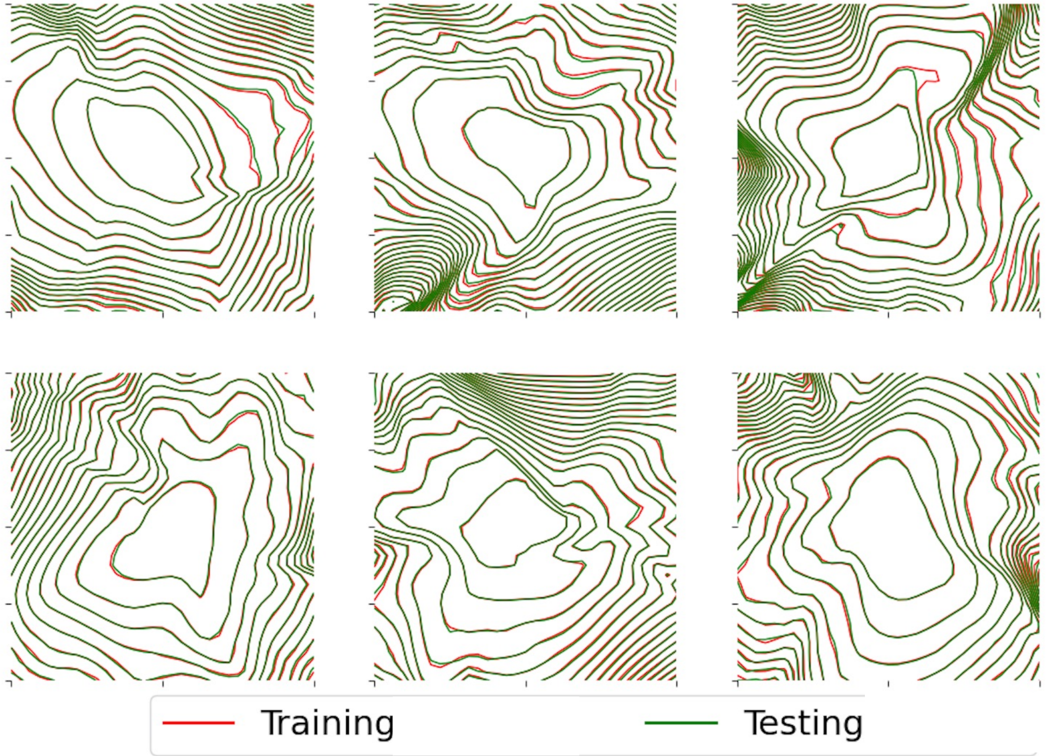
Extra Materials

		Task (T) / Milestone(M)	Period
FY21	T1.1	Dynamical analysis in deterministic setting†	10/2020-06/2021
	T1.2	Development of deterministic, sparse weight representations	01/2021-09/2021
	M1	Demonstrate reduced NODE in deterministic setting	by 09/2021
	T1.3	Formulation of Bayesian inference of weights	04/2021-09/2021
FY22	T2.1	Formulation of stability conditions under uncertainty†	10/2021-03/2022
	T2.2	Extension of dynamical analysis under uncertainty	10/2021-06/2022
	T2.3	Formulation of fractional PNODE construction*	01/2022-09/2022
	T2.4	Regularization via weight representations in PNODEs	01/2022-06/2022
	T2.5	Inference/training of weight functions in PNODEs	04/2022-09/2022
	M2	Demonstrate training of regularized PNODEs	by 09/2022
FY23	T3.1	Explore the scaling and performance of PNODEs	10/2022-03/2023
	T3.2	Construction of NNs as PNODE discretization with desired features	10/2022-09/2023
	M3	Demonstrate quantifiable improvement on exemplar applications	by 09/2023
	T3.3	Final SAND report	07/2023-09/2023

TA: With weight parameterization, ResNets regularize loss landscape compared to MLPs

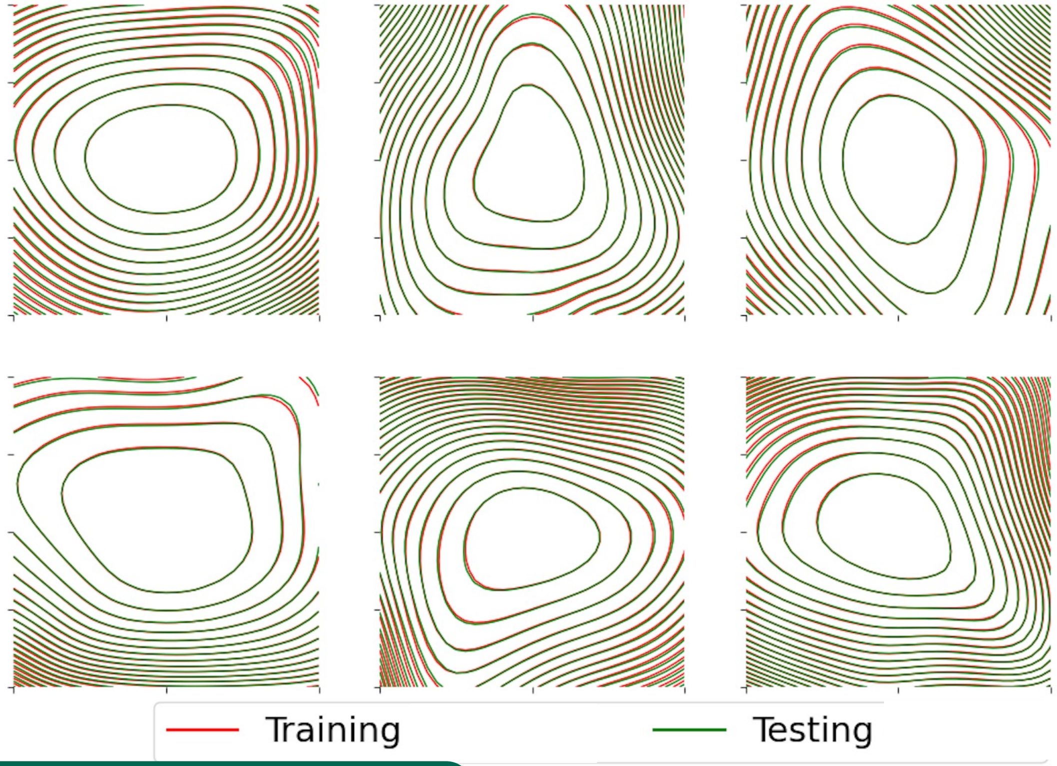
MLP NN: $x_{n+1} = \sigma(W_n x_n + b_n)$

Multilayer Perceptron (learning the layer)



ResNet: $x_{n+1} = x_n + \alpha_n \sigma(W_n x_n + b_n)$

ResNets (learning the layer diff.)



This is with E3SM model data – and this feature helps both the training and the accuracy of model surrogate

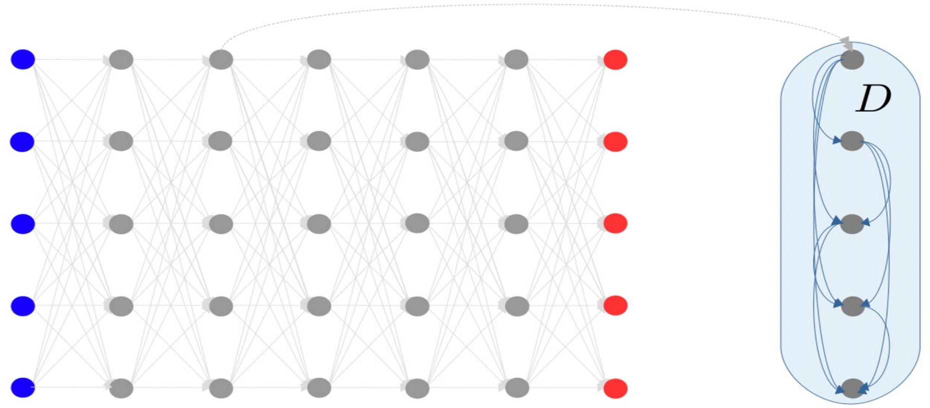
TA: Integral NODEs for PDE learning outperforms conventional operator learning methods

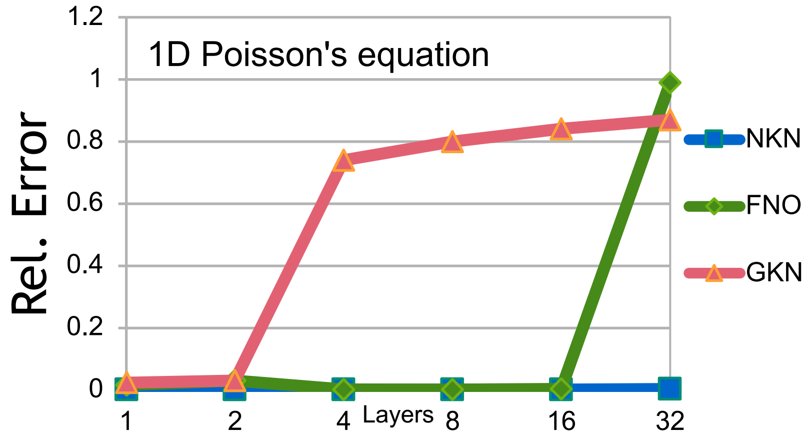
$$\frac{d\mathbf{h}}{dt} = \sigma(W_t\mathbf{h}_t + \beta_t)$$

NODE

$$\frac{\partial \mathbf{h}(\mathbf{x},t)}{\partial t} = \int_D k_{NN}(\mathbf{x},\mathbf{y};\mathbf{v})(\mathbf{h}(\mathbf{y},t)-\mathbf{h}(\mathbf{x},t))d\mathbf{y} - R_{NN}(\mathbf{x})\mathbf{h}(\mathbf{x},t)$$

Nonlocal Kernel Network





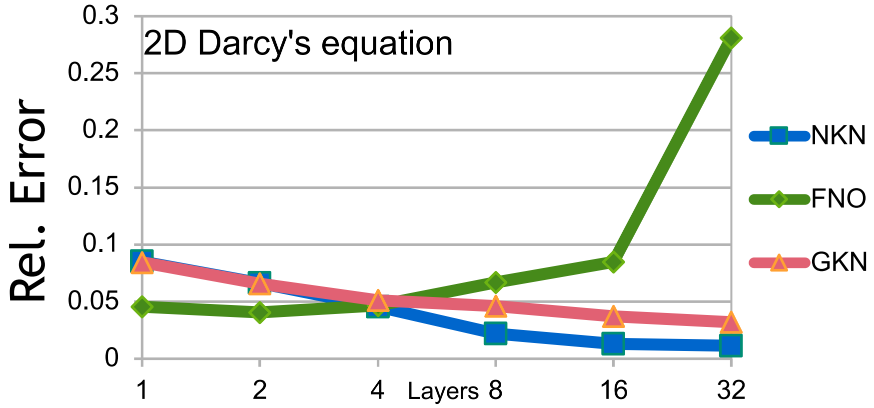
Nonlocal Kernel Network

Fourier Neural Operator

Graph Kernel Network

*You, Yu, **D'Elia**, Gao, Silling, Nonlocal Kernel Network (NKN): A Stable and Resolution-Independent Deep Neural Network. To be Submitted.*

- Resolution-independent
- Stable operator learning
- **NKN beats leading competitors FNO and GKN**



TA: Analytical work done to establishing equivalence

Task 0.0

ResNet

$$x_{n+1} = x_n + \alpha_n \sigma(W_n x_n + b_n)$$

NODE

$$\frac{dx}{dt} = \sigma(W(t)x + b(t))$$

Forward equivalence:

Neural ODE discretized using explicit Euler and ResNet produce identical outputs choosing time step: $\Delta t = \frac{T}{L}$, $\alpha_n := \Delta t, W_n := W(n\Delta t)$ and $b_n := b(n\Delta t)$ for all n.

Backward not so much:

Gradient computations differ!

Consider $W(t) \equiv W$ and $b \equiv 0$:

Discretized Neural ODE with adjoint method:

$$\nabla loss = 2\big((1 + \delta t\, W)^L x - y\big)(1 + \delta t\, W)^{\boxed{L}} x$$

ResNet with backpropagation:

$$\nabla loss = 2\big((1 + \delta t\, W)^L x - y\big)(1 + \delta t\, W)^{\boxed{L-1}} x$$

- Gradients converge as $L \to \infty$ but differences can be large for small L,
- Optimize then discretize (adjoint method) $\neq$ discretize then optimize (backpropagation).

TECHNICAL ACCOMPLISHMENTS: OVERVIEW

		Task (T) / Milestone(M)	Period
FY21	T1.1	Dynamical analysis in deterministic setting†	10/2020-06/2021
	T1.2	Development of deterministic, sparse weight representations	01/2021-09/2021
	M1	Demonstrate reduced NODE in deterministic setting	by 09/2021
	T1.3	Formulation of Bayesian inference of weights	04/2021-09/2021

Task 0.0

Ground work done: we have PyTorch codes developed from scratch for both ResNet (discrete) and Neural ODE (continuous)

Task 1.1

- Dynamical analysis
- Stiffness penalization

Task 1.2

- Weight parameterization by depth

Task 1.3

- Bayesian inference of weights
- Both Markov chain Monte Carlo and variational inference

M 1

- … with a slight re-focus: not removing the fast dynamics, but regularizing

Analysis of Neural Networks as Random Dynamical Systems

FY21-0528: K. Sargsyan (8351), J. Bennett (8759), 3Yr, Total $1600K

Prior Work

First ingredient: Neural ODEs

- **Neural ODEs** been around a while (few papers in 90's), but revived in ML community recently
  - *Chen, Duvenaud, 2018+*: clever trick with adjoints; *Ruthotto et al, 2018+*: more fundamental, discovery; *Weinan E, 2017*: dynamical system context; training as control
- Many extensions followed
  - SDE context [*Liu et al, 2019; Tzen et al, 2019*]; PDE context: [*Ruthotto et al, 2018; Long et al, 2018*]
  - Inspires new NN architectures [*Lu et al, 2018*]
  - Fractional/nonlocal DNN [*Antil, 2020; Pang, 2020; D'Elia, 2020*]
- Challenges with NODEs, active area of research: we will inherit some of the issues, but also enhancements
- Not restricted to dynamical, time-resolved physical models
- Good balance of optimism and skepticism in literature

Analysis of Neural Networks as Random Dynamical Systems

FY21-0528: K. Sargsyan (8351), J. Bennett (8759), 3Yr, Total $1600K

Prior Work

Second ingredient: Probabilistic formulation

- **Probabilistic NN** have been around since 90s [*MacKay, 1992; Neal, 1997*]
  - Full probabilistic treatment was infeasible back then (and still is, generally)
  - Recent work showed avenues via variational methods with clever tricks:

    Bayes by Backprop [*Blundell, 2015*]; Probabilistic backprop [*Hernandez-Lobato 2015*]
  - *Ghahramani, "Probabilistic Machine Learning and Artificial Intelligence". Nature, 2015*
    - *"Nearly all approaches to probabilistic programming are Bayesian since*

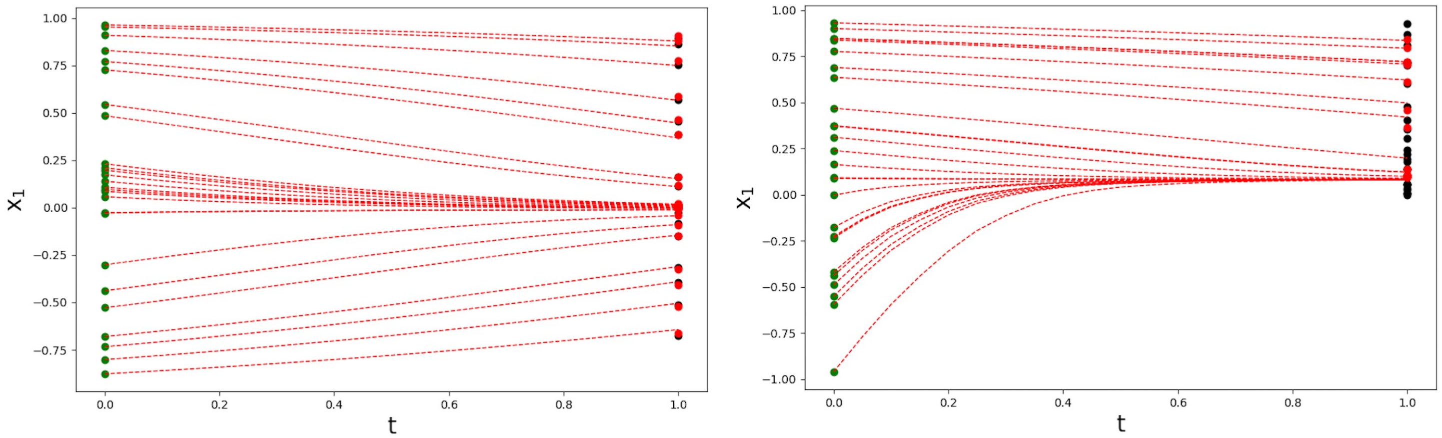
      it is hard to create other coherent frameworks for automated reasoning about uncertainty"
  - Industry ***is*** catching up: Bayesflow at Google, infer.NET at Microsoft, Uber has shown interest
  - Still not industry-standard: expensive, not well understood.

TA: Simple 1d and 2d demos highlight challenges ahead

Task 0.0

Path crossing issue, best seen in 1d

Function $y = x^3$ Function $y = x^2$

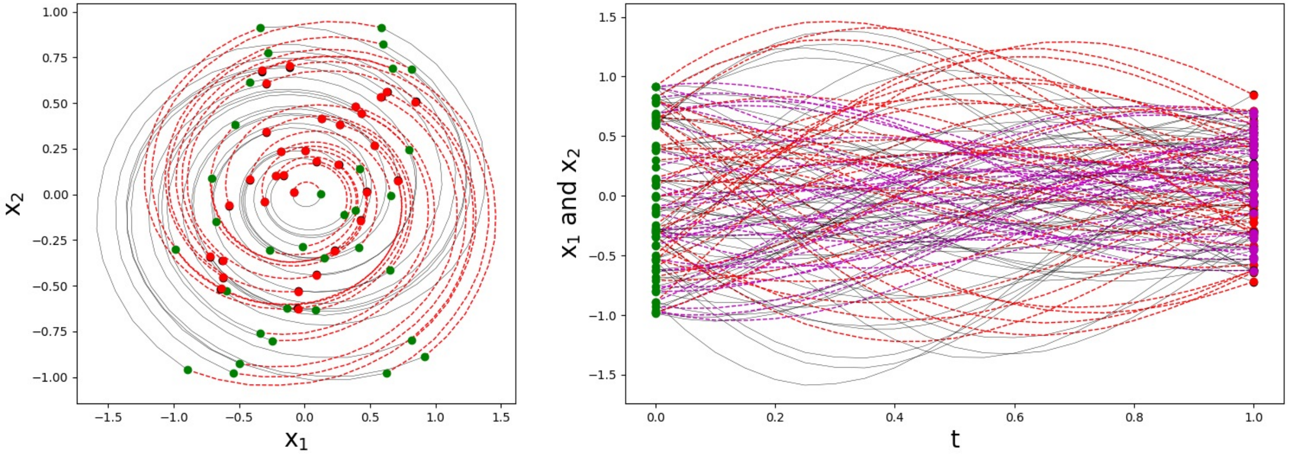


Linear, autonomous NODE $\frac{d\boldsymbol{x}}{dt} = \boldsymbol{W}\boldsymbol{x}$

Exact solution $\boldsymbol{x}(T) = \boldsymbol{x}(0)e^{\boldsymbol{W}T}$

Matrix logarithm is not unique!

$e^{\boldsymbol{W}_1} = e^{\boldsymbol{W}_2}$ but $\boldsymbol{W}_1 \neq \boldsymbol{W}_2$



CIS LDRD Key Performance Indicators and Metrics

Enhance technology base at Sandia: New, differentiated, utilized internal tools and capabilities

R&D technologies (math, software, hardware, knowledge, skills, techniques, design, etc.) deployed resulting in mission achievement, cost savings and/or **follow-on funding**

Journal and conference publications, conference presentations, invited talks, citations, journal impact factor

Career growth: Professional awards/recognition, best papers

Intellectual Property: Technical advances, patents, commercial license, copyrights, **open source software assertions**, government use notices, royalties

Capabilities and knowledge picked up by MF IA's or sponsors to advance TRL

External collaborations: Academia, industry, other Labs and gov. agencies

Software reproducibility and reuse across Sandia

Neuromorphic Graph Algorithms
PI: Ojas Parekh PM: Kevin Dixon

Project Goal(s)

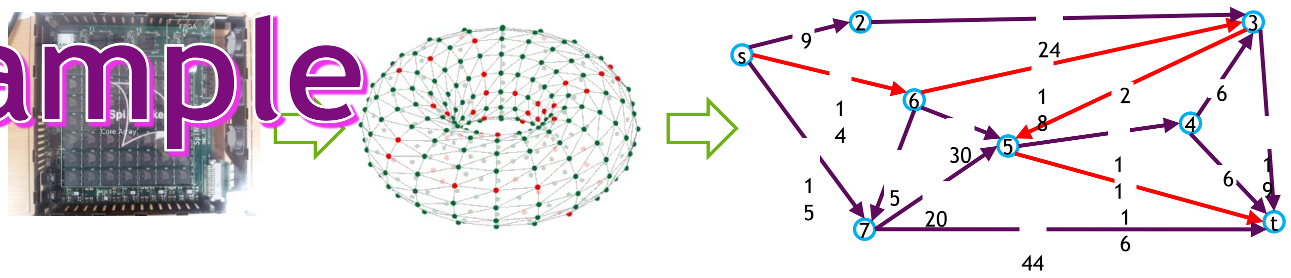
Leverage energy-efficient massive parallelism of near-term neuromorphic systems in designing scalable beyond-Moore Neuromorphic Graph Algorithms (NGAs)

Provide one of world's first empirical and rigorous proofs of neuromorphic advantages, to validate and focus neuromorphic hardware development

Mission Impact

Graph analysis for, e.g., applications to cybersecurity, counter-terrorism, rare event detection, and social network analysis are a significant mission driver. The anticipated scalability and energy-efficiency of neuromorphic graph algorithms would provide a massive leap in our capabilities.

Example



FY20 Technical Milestones

Milestone	Status
Theoretical performance models for neuromorphic computing devices (NCDs)	Achieved
NGA that outperforms best conventional algorithms (empirically or asymptotically)	Achieved (Go/No-Go)
Shortest path implementation on an NCD	In progress

Transition Plan

More fundamental algorithmic work is expected to transition to the the DOE Office of Science Advanced Scientific Computing Research (ASCR) program, while identified near-term neuromorphic graph algorithms wins will be further developed to benefit current sponsors of graph algorithms work.

Linear Programming in Strongly Polynomial Time

PI: Mohamed Ebeida, PM: Randy Smith

Project goal(s)

Develop algorithms and software to solve linear programs (LP) in strongly linear time (this is *9th on Smale's list of greatest unsolved math problems of the 21st century*!)

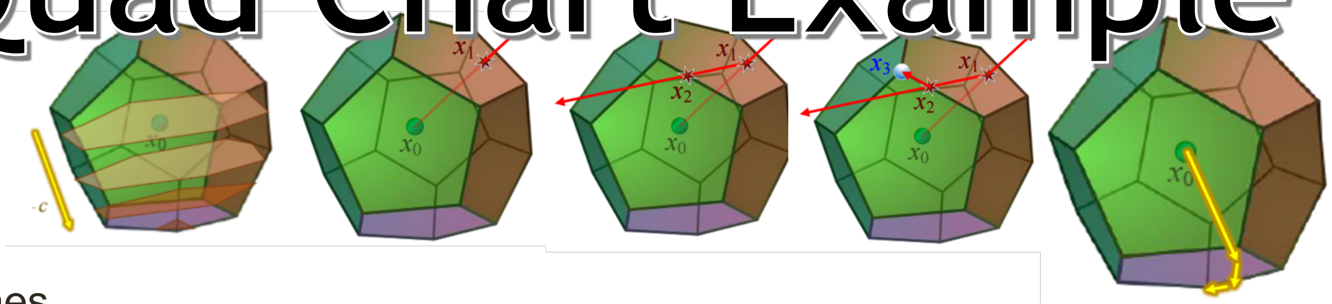
Provide rigorous proof of performance bounds

Computer certificate of correctness by simultaneously solving the LP's dual problem

Mission Impact

This capability would open up the design space for a wide variety of national security missions. It would enable the efficient solution of inverse problems arising from non-destructive testing, and solving scheduling and NW logistics problems in real-time.

Quad Chart Example



FY18 Technical Milestones

Serial implementation of practical LP solver	3/2018
Prove strongly-polynomial linear program for the exponential-time simplex cases or serial implementation outperforms open-source simplex method	6/2018
Tailor LP algorithm for discrete optimization	9/2018

Transition Plan

Algorithms and software developed in this project will transition into the NNSA NA-10 funded Advanced Simulation and Computing (ASC) program and/or the DOE Office of Science Advanced Scientific Computing Research (ASCR) program.