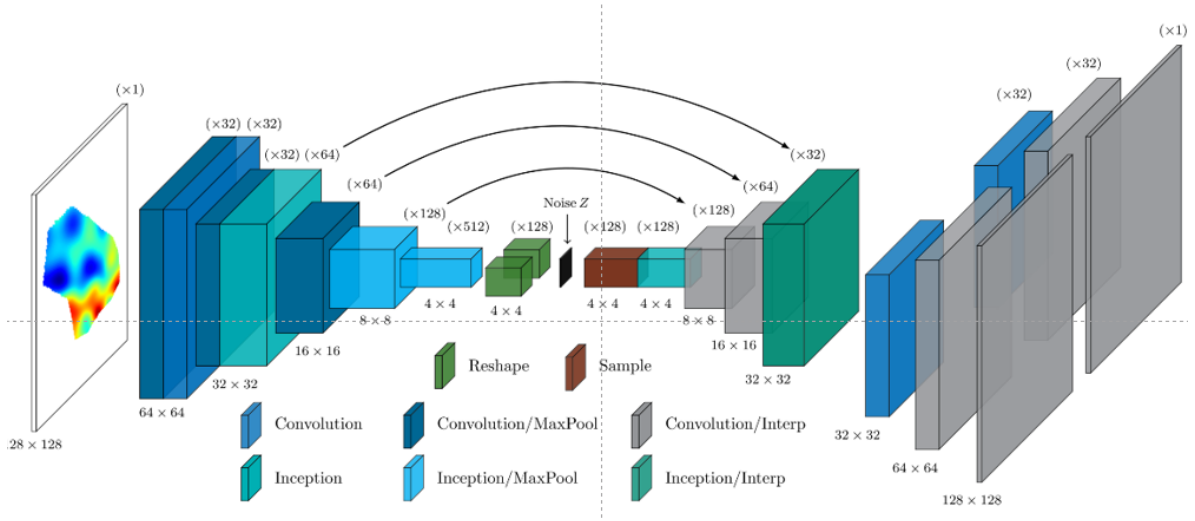
Towards Third Wave AI: Interpretable, Robust Trustworthy Machine Learning for Diverse Applications in Science and Engineering

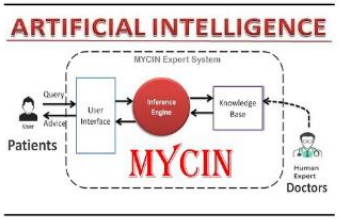
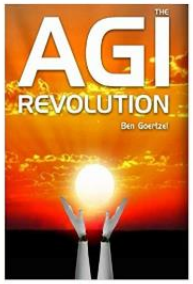
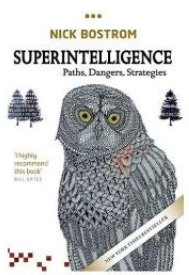
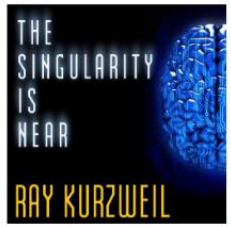
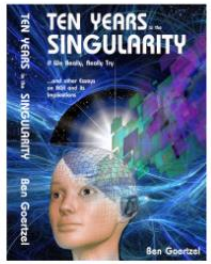
Guang Lin,

Director, Data Science Consulting Service, Full Professor of Departments of Mathematics, Statistics, & Mechanical Engineering, Purdue University



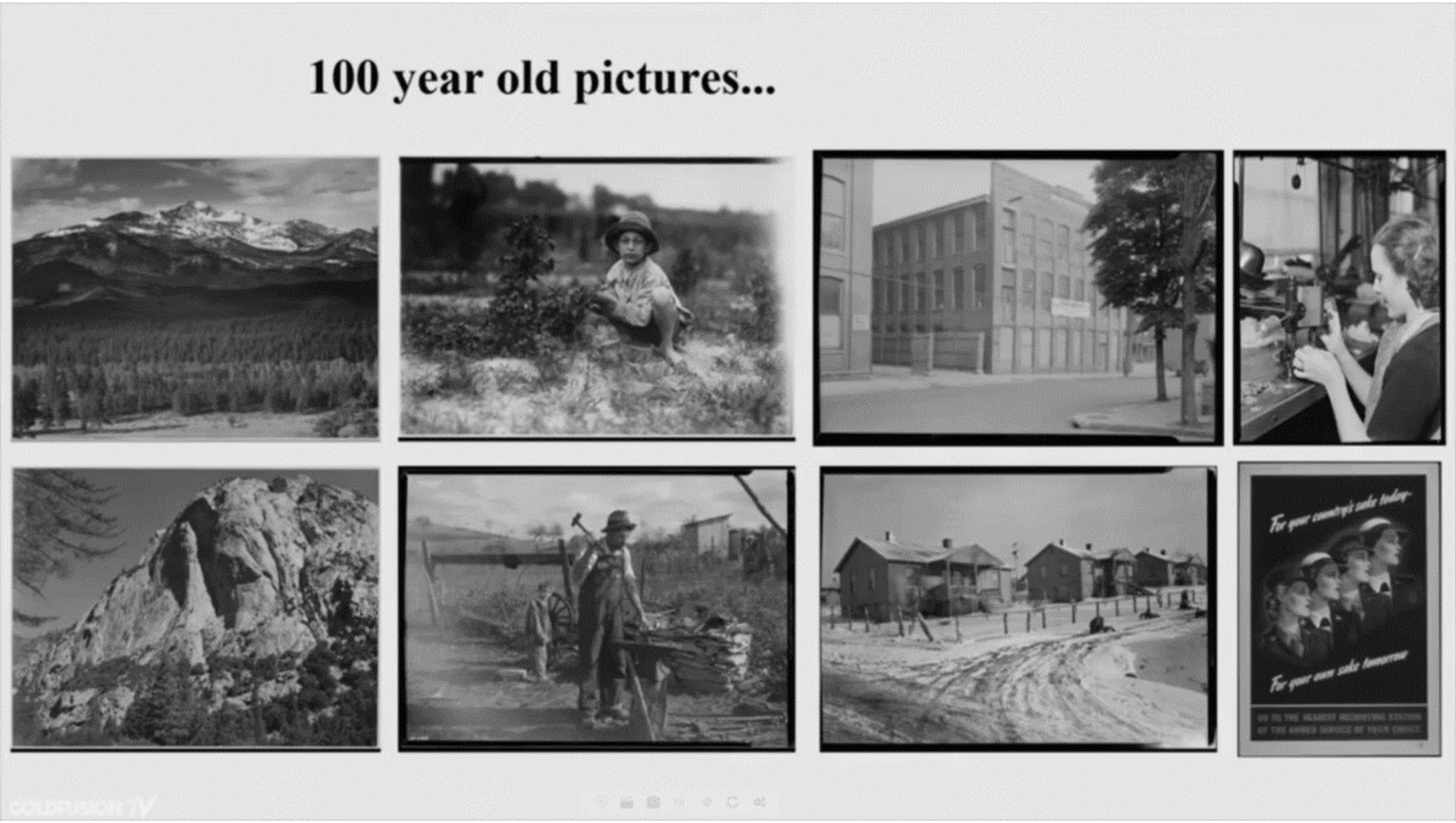
MATH-IMS Joint Applied Mathematics Colloquium, Chinese University of Hong Kong, May 20, 2022

The Four Waves of AI

First Wave	Second Wave	Third Wave	Fourth Wave
c. 1970s - 1990s	*c. 2000s - present*	*est. 2020s - 2030s*	*est. 2030s →*
Good at reasoning, but no ability to learn or generalize. • GOFAI - "Good Old Fashioned AI." • Symbolic, heuristic, rule based. • Handcrafted knowledge, "expert systems."	Good at learning and perceiving, but minimal ability to reason or generalize. • Statistical learning, "deep" neural nets, CNNs, RNNs. • Advanced text, speech, language and vision processing.	Excellent at perceiving, learning and reasoning, and able to generalize. • Contextual adaptation, able to explain decisions. • Can converse in natural language. • Requires far fewer data samples for training. • Able to learn and function with minimal supervision.	Able to perform any intellectual task that a human can. • AGI (Artificial General Intelligence), possibly leading to ASI (Artificial Superintelligence) and the "Technological Singularity."
	 	SingularityNET aigo.ai Pandai	   
			

Six Kin Development (adapted from DARPA's "Three Waves of AI")

Artificial Intelligence: Image Coloring



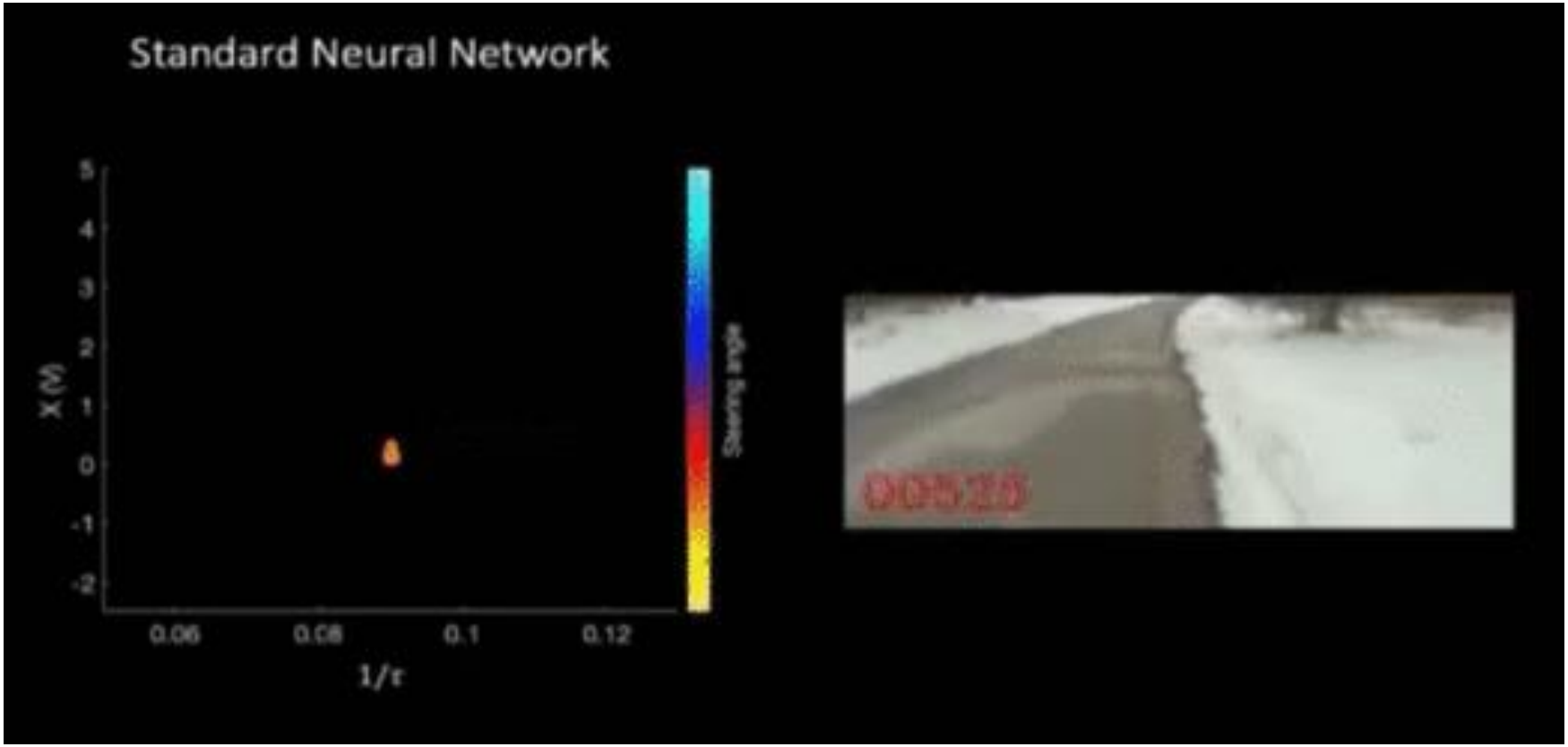
Credit to ColdFusion

Artificial Intelligence: AlphaGo

Reinforcement Learning in AlphaGo

Credit to DeepMind

Artificial Intelligence: Autonomous Driving

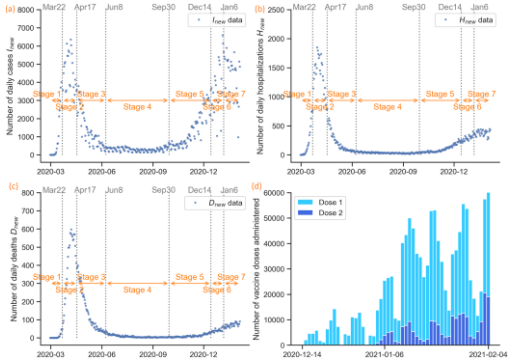


Credit to Ramin Hasani, MIT.

Guang Lin's Group's Main Research

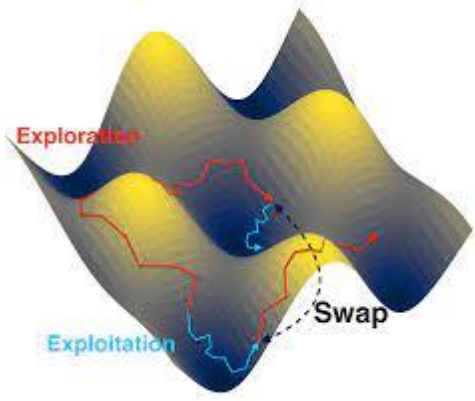
COVID 19 Pandemic Prediction

1. Nature Computational Science, 1-10, 2021
2. PlOS Computational Biology, 2021



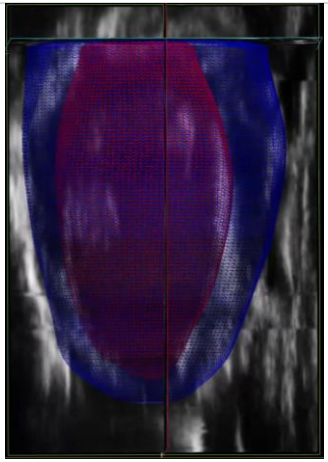
Uncertainty Quantification for Deep Learning

- Tier 1 AI conference: ICLR21, ICML21, WSDM21



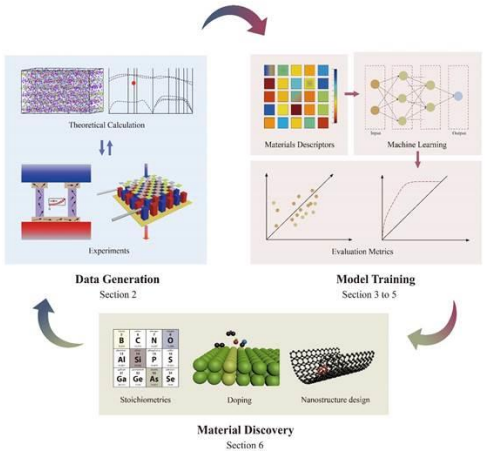
AI for Cardiac 4D Ultrasound

- Received Trask Innovation Award
- Patent submitted
- Applied Sciences 21



Machine Learning for Material Discovery

- ES Materials & Manufacturing 21



Outline:

- **Incorporate Physics Knowledge and AI to design new interpretable models**
- Incorporate Physics Knowledge into AI to predict multiscale problems: NH-PINN
- Interpretable AI enables data-driven scientific discovery with uncertainty quantification capability – ALZHEIMER's Disease Prediction
- Sparse Neural Architecture Design with quantified uncertainties
- Scalable training large-scale Deep Neural Network

How to incorporate Physics Knowledge and AI to design new interpretable models? - Interpretable AI for Science

1. Ehsan Kharazmi, Min Cai, Xiaoning Zheng, Guang Lin, George Em Karniadakis, **Identifiability and predictability of integer- and fractional-order epidemiological models using physics-informed neural networks, Nature Computational Science, 1, 744-753, 2021**
2. Sheng Zhang, Joan Ponce, Zhen Zhang, Guang Lin, George Karniadakis, **An integrated framework for building trustworthy data-driven epidemiological models: Application to the COVID-19 outbreak in New York City, *PLoS Computional Biology*** 17(9): e1009334. https://doi.org/10.1371/journal.pcbi.1009334

Predicting the COVID-19 pandemic with uncertainties using trustworthy data-driven epidemiological models

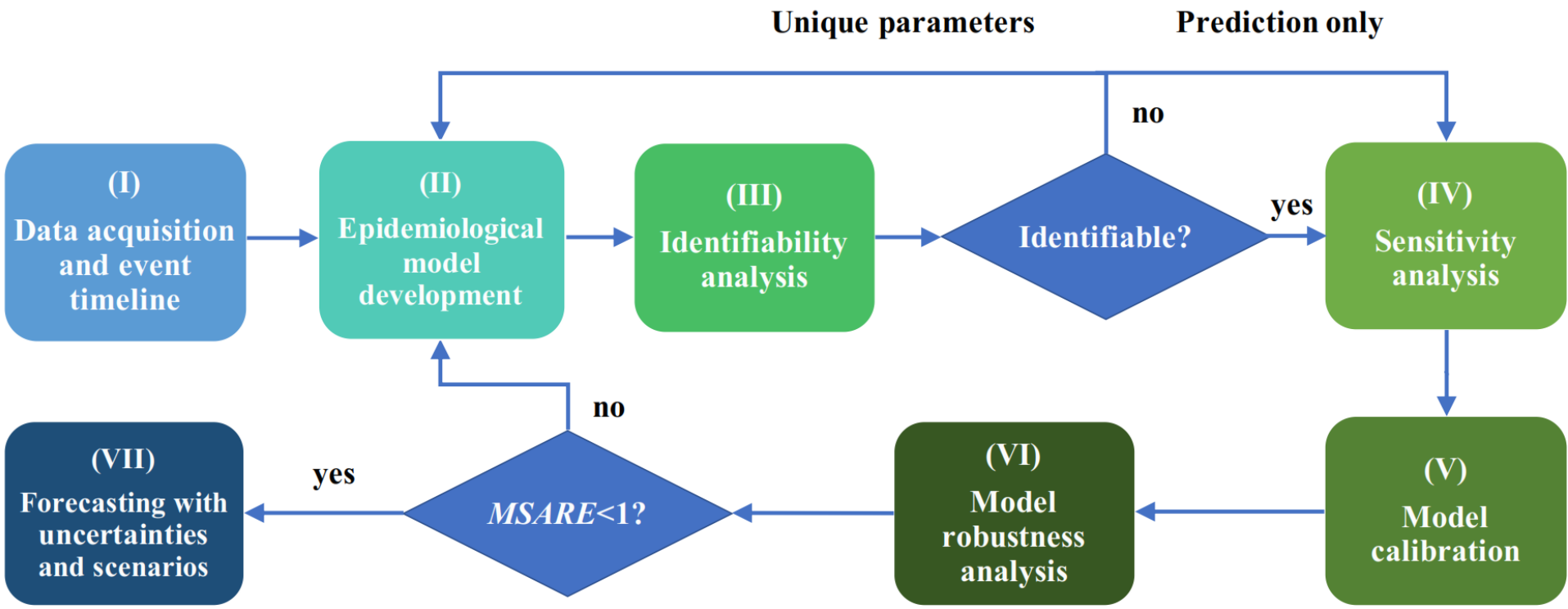
1. Ehsan Kharazmi, Min Cai, Xiaoning Zheng, Guang Lin, George Em Karniadakis, **Identifiability and predictability of integer- and fractional-order epidemiological models using physics-informed neural networks, Nature Computational Science, 1, 744-753, 2021**

2. Sheng Zhang, Joan Ponce, Zhen Zhang, Guang Lin, George Karniadakis, **An integrated framework for building trustworthy data-driven epidemiological models: Application to the COVID-19 outbreak in New York City, *PLoS Computional Biology*** 17(9): e1009334. https://doi.org/10.1371/journal.pcbi.1009334

BROWN

PURDUE UNIVERSITY

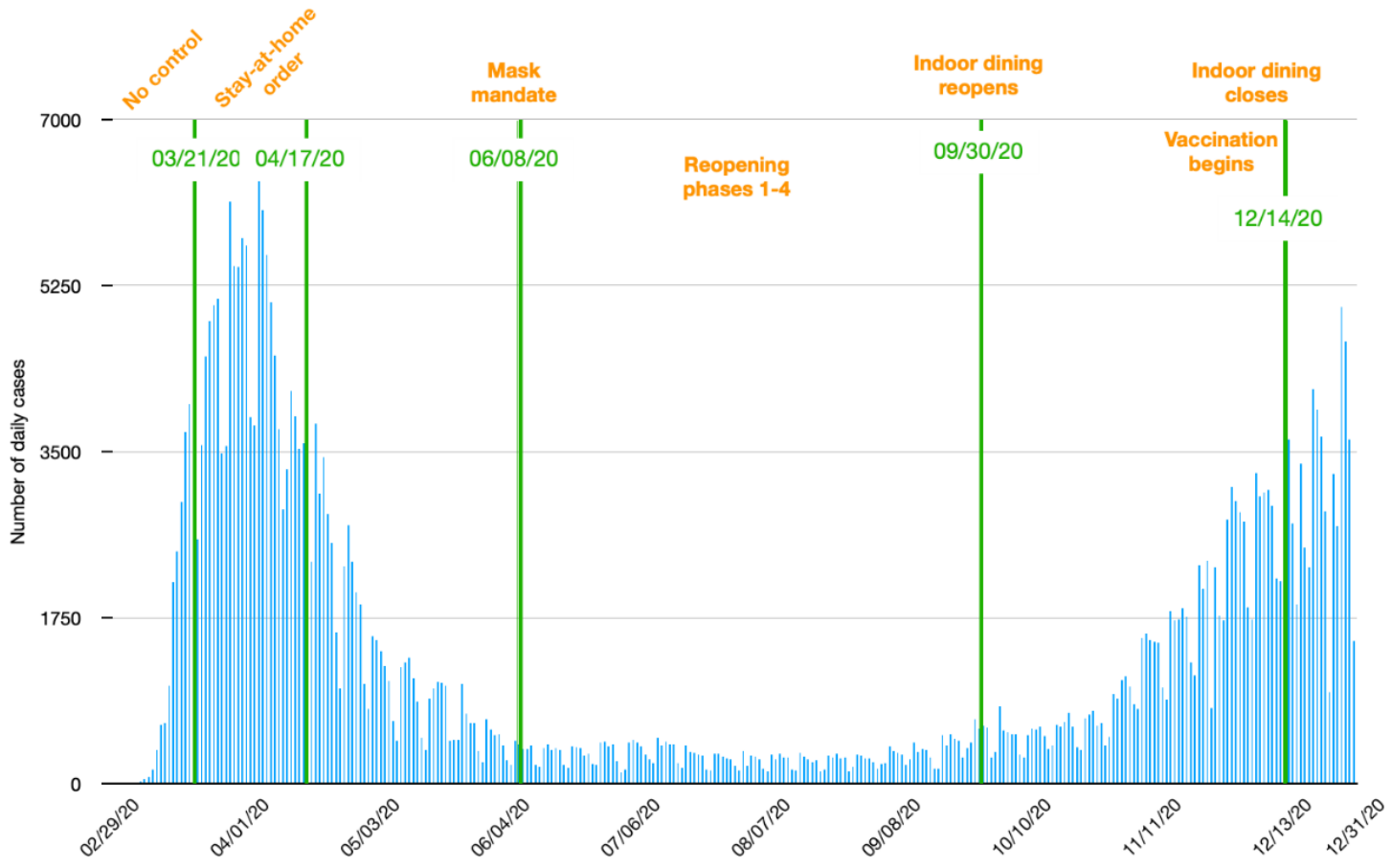
A general framework for building a trustworthy data-driven epidemiological model



Sheng Zhang, Joan Ponce, Zhen Zhang, Guang Lin, George Karniadakis, **An integrated framework for building trustworthy data-driven epidemiological models: Application to the COVID-19 outbreak in New York City,** ***PLoS Computional Biology*** 17(9): e1009334.
https://doi.org/10.1371/journal.pcbi.1009334

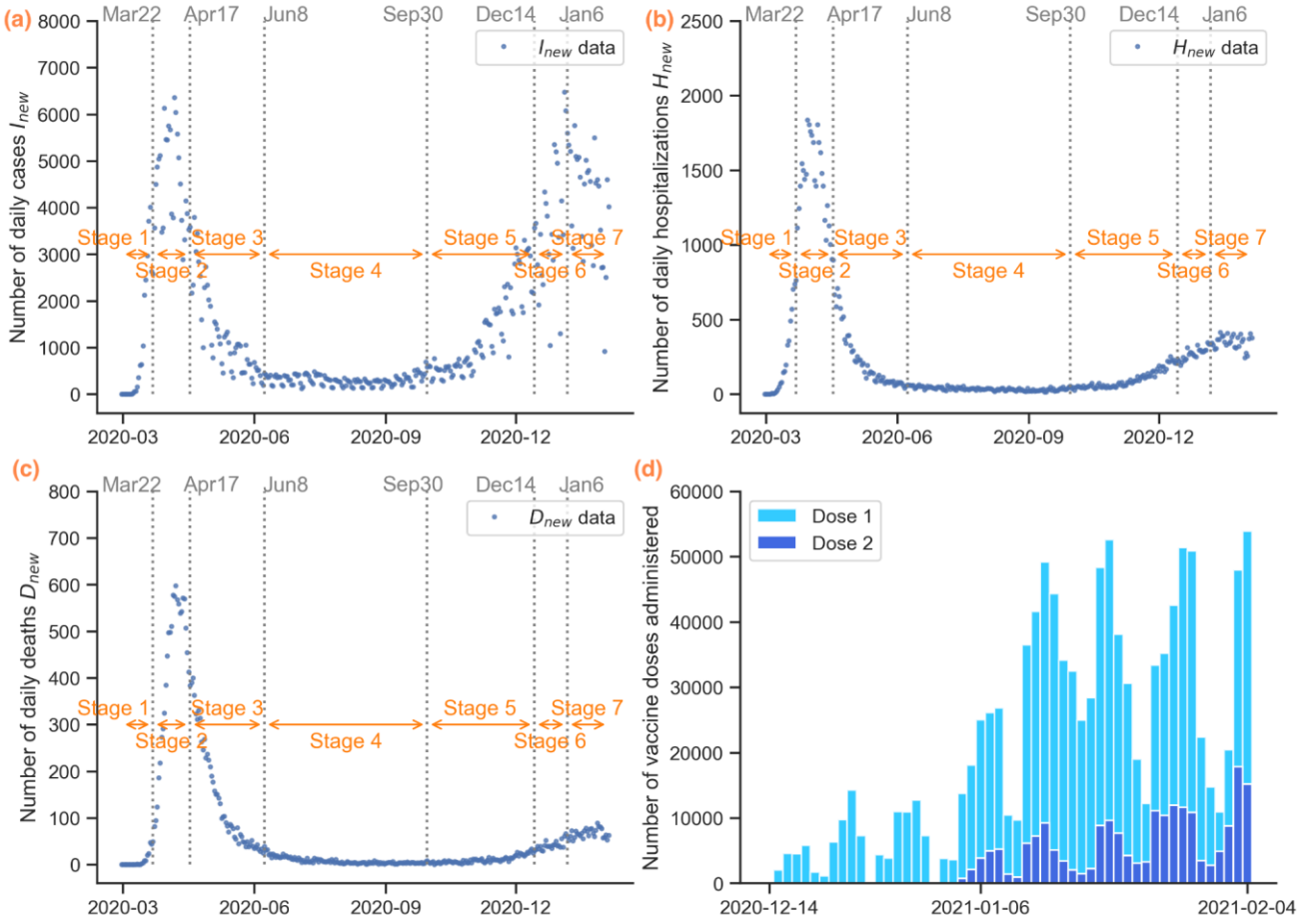
New York City COVID-19 related Event Timeline

Calibrate piecewise-constant model parameters to capture local epidemiological dynamics



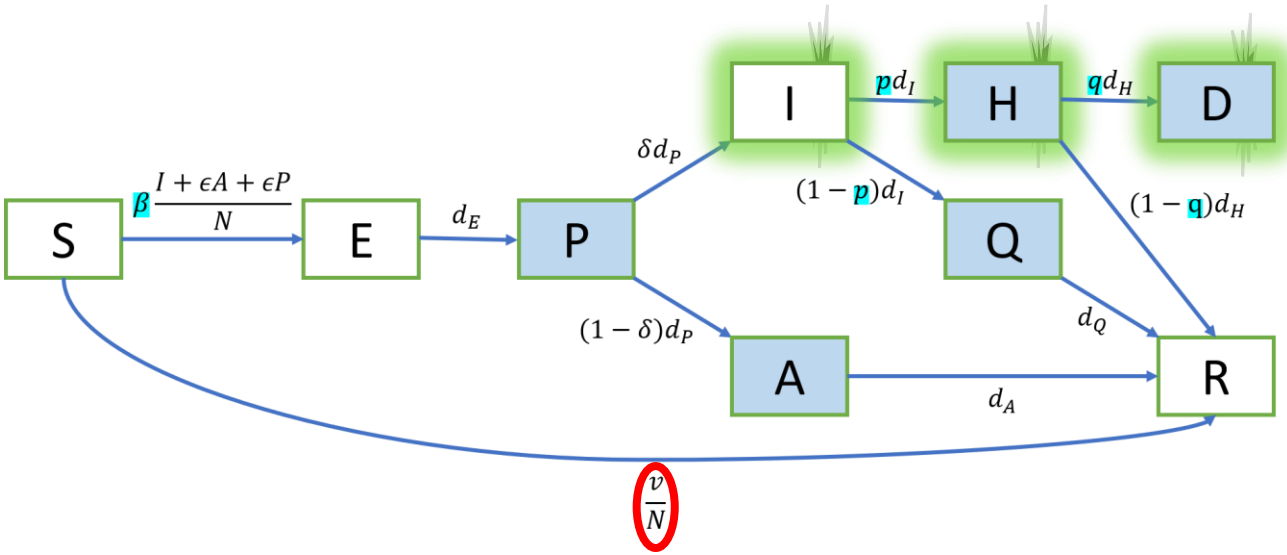
New York City COVID-19 related Event Timeline

Calibrate piecewise-constant model parameters to capture local epidemiological dynamics



Epidemiological Model Development

$$
\begin{cases}
\frac{dS}{dt} = -\beta \frac{I + \epsilon A + \epsilon P}{N} S - \frac{v}{N} S \\
\frac{dE}{dt} = \beta \frac{I + \epsilon A + \epsilon P}{N} S - d_E E \\
\frac{dP}{dt} = d_E E - d_P P \\
\frac{dI}{dt} = \delta d_P P - d_I I \\
\frac{dA}{dt} = (1-\delta) d_P P - d_A A \\
\frac{dH}{dt} = p d_I I - d_H H \\
\frac{dQ}{dt} = (1-p) d_I I - d_Q Q \\
\frac{dD}{dt} = q d_H H \\
\frac{dR}{dt} = d_A A + (1-q) d_H H + d_Q Q + \frac{v}{N} S.
\end{cases}
$$

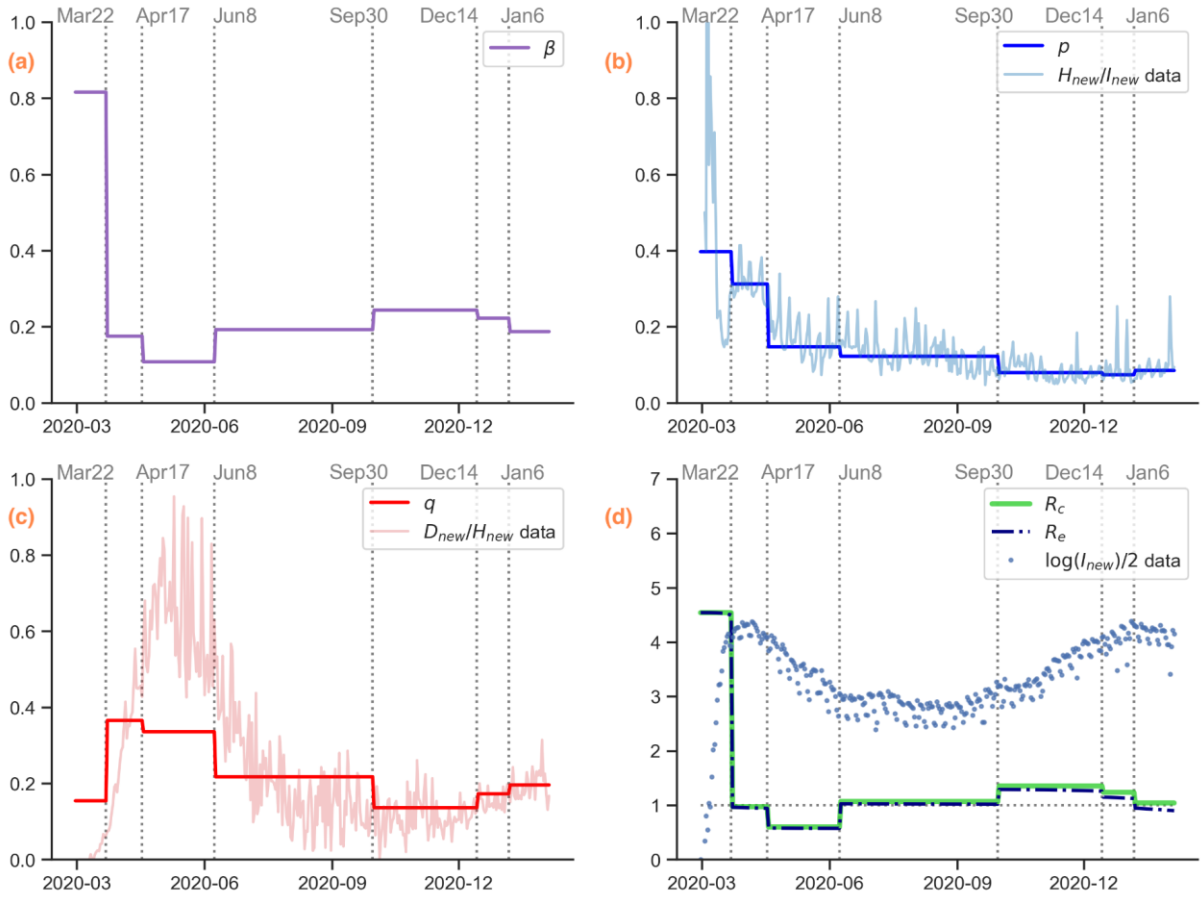


Fixed parameters:
eps = 0.75
delta = 0.6
d_E = 1/2.9
d_P = 1/2.3
d_I = 1/2.9
d_A = 1/7
d_H = 1/6.9
d_Q = 1/10

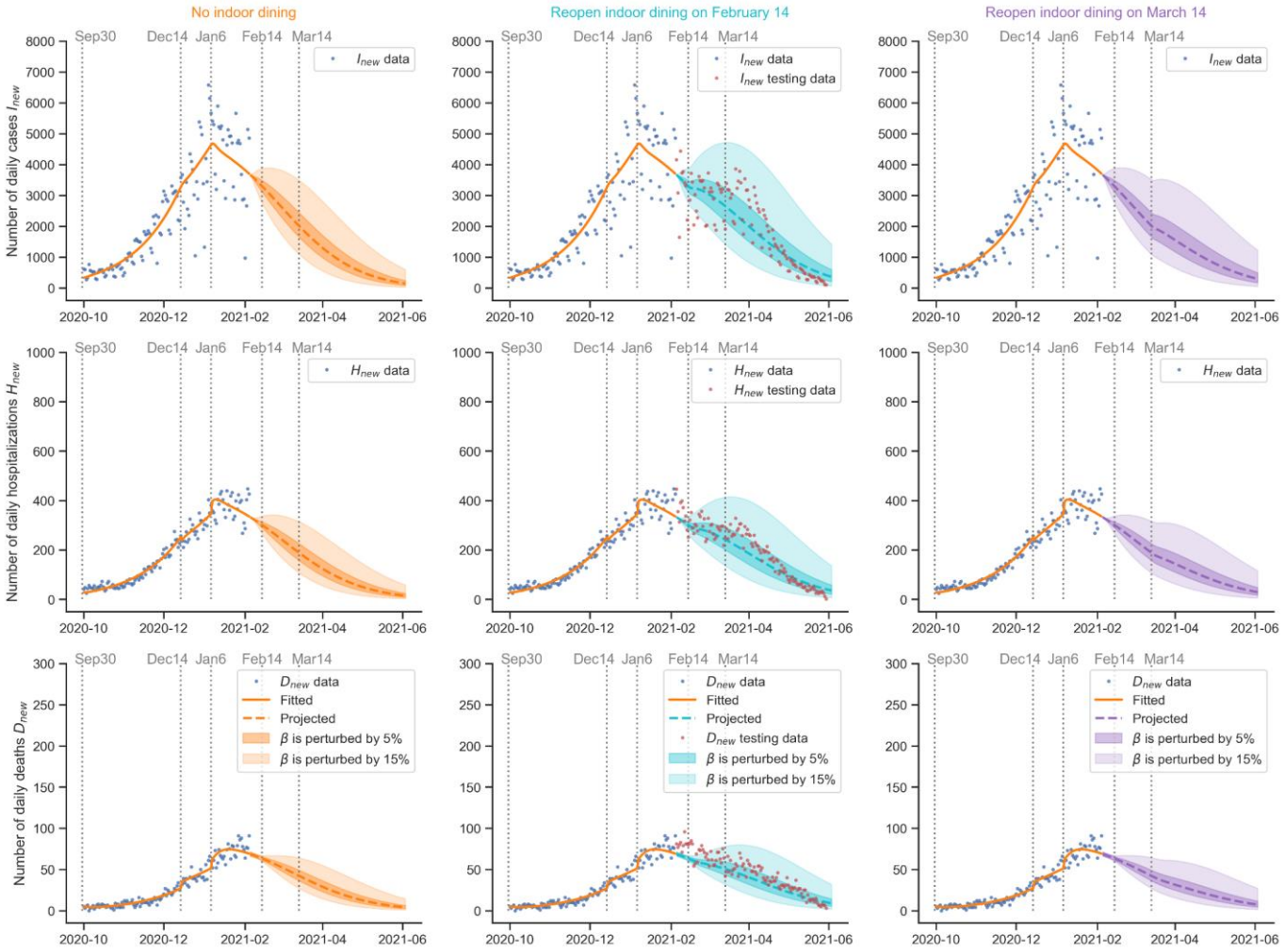
β: Transmission rate
p: Hospitalization rate
q: Death from hospital rate

Calibrated COVID-19 Transmission Rate for New York City

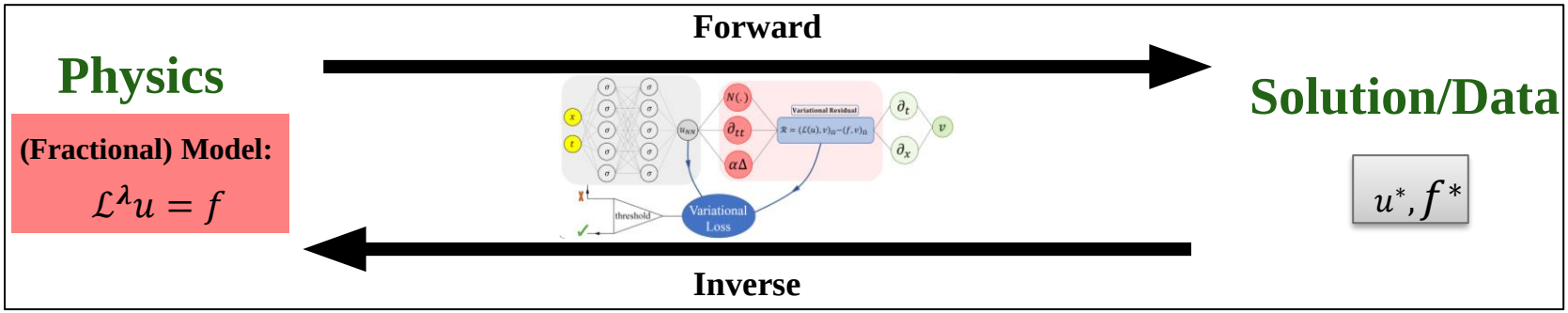
Calibrate piecewise-constant model parameters to capture local epidemiological dynamics



Forecasting with Uncertainties and Scenarios



Physics Informed Neural Networks (PINNs)



- $\mathcal{L}^{\lambda}$ A (non-local) differential operator with parameters $\boldsymbol{\lambda}$

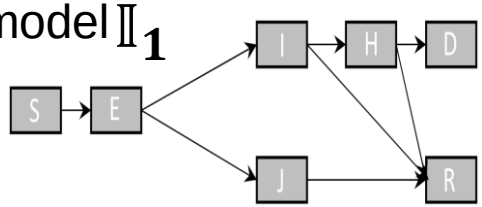
A flexible **computational** tool to study **model uncertainty**
Incorporate **data** and **different models**
Accurate **fitting** to data
Inferring model parameters and **discovering** unobserved dynamics

1. Ehsan Kharazmi, Min Cai, Xiaoning Zheng, Guang Lin, George Em Karniadakis, **Identifiability and predictability of integer- and fractional-order epidemiological models using physics-informed neural networks, Nature Computational Science, 1, 744-753, 2021**

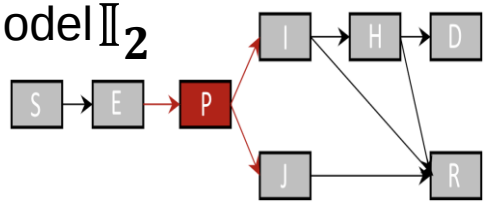
Different Epidemiological Models

Integer-Order Models (simple to complex models)

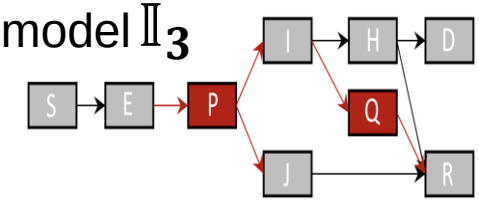
model $\mathbb{I}_1$



model $\mathbb{I}_2$



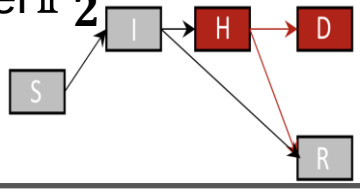
model $\mathbb{I}_3$



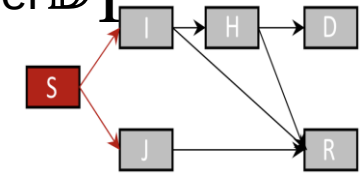
Fractional-Order and Time-Delay Models (add memory effects)

model $\mathbb{F}_1$

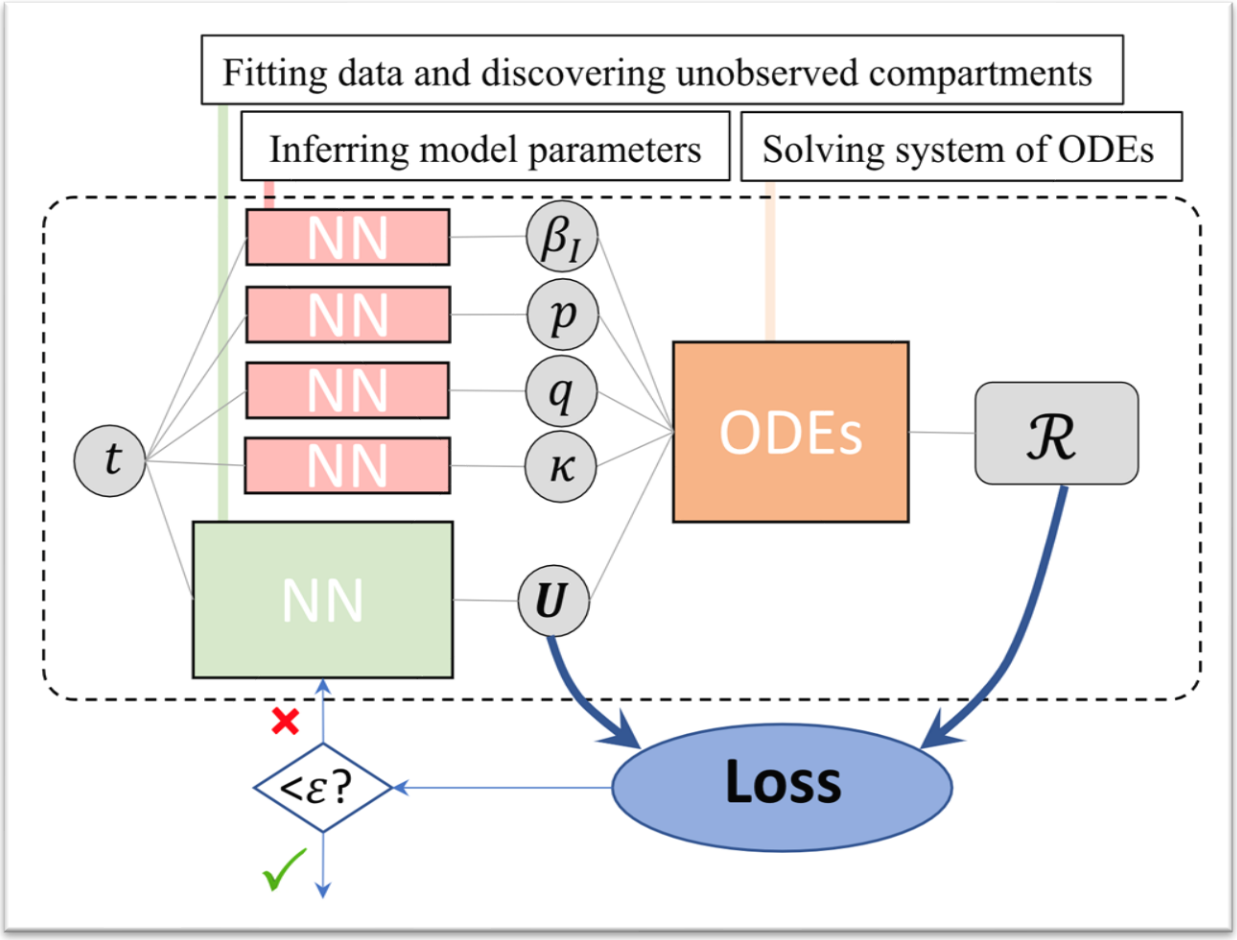
model $\mathbb{F}_2$



model $\mathbb{D}_1$



PINNs for (Fractional) Epidemiological Models



$$loss = \frac{1}{N_u}\sum_{i=1}^{N_u}|\boldsymbol{U}(t_i;\boldsymbol{\theta}) - data(t_i)|^2$$

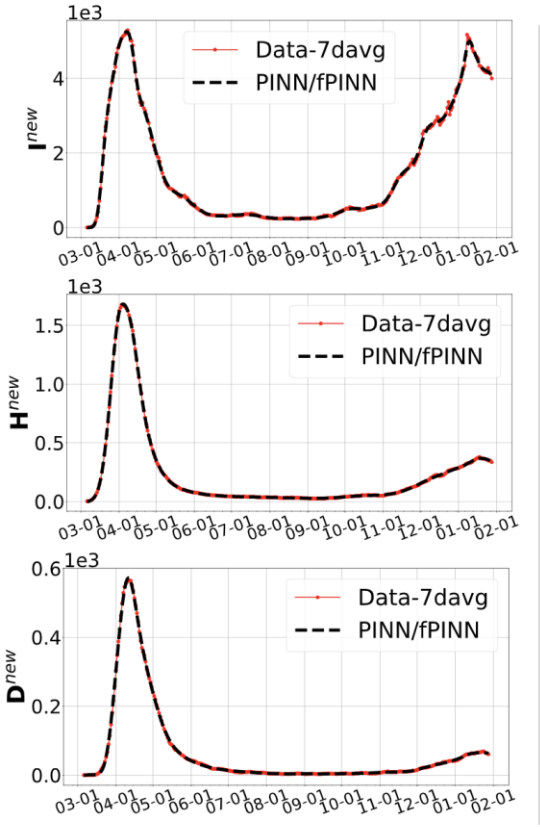
Loss u (data/IC points)

$$+\frac{1}{N_r}\sum_{j=1}^{N_r}\left|\mathcal{R}\left(t_j;\boldsymbol{\theta},\boldsymbol{\lambda}\right)\right|^2$$

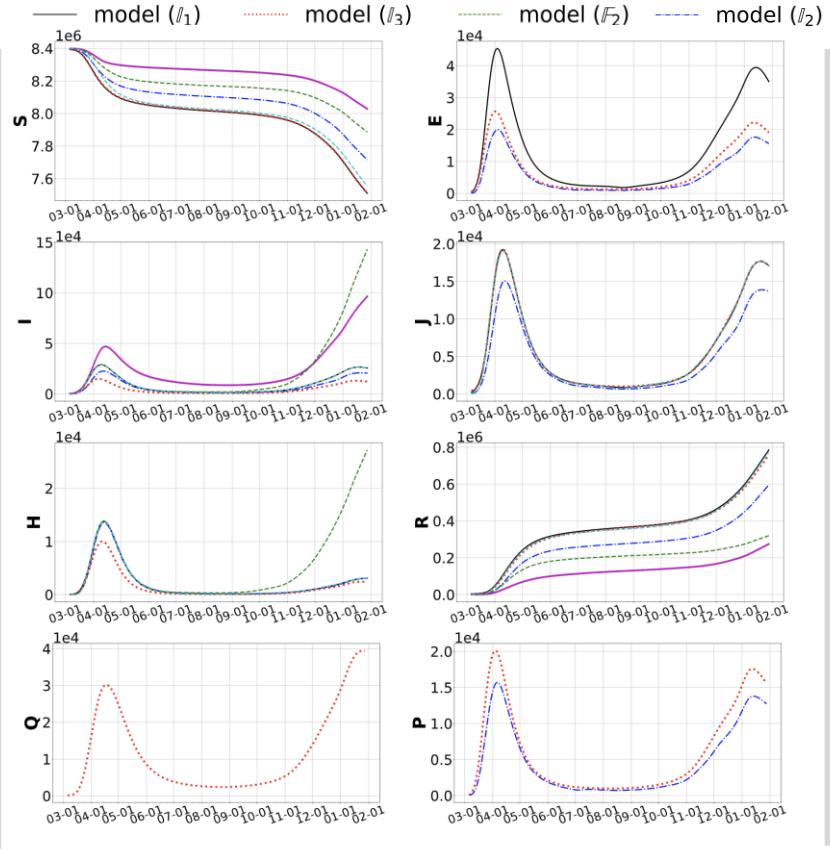
Loss ODE (residual points)

PINN Results: Model Uncertainty based on NYC dataset

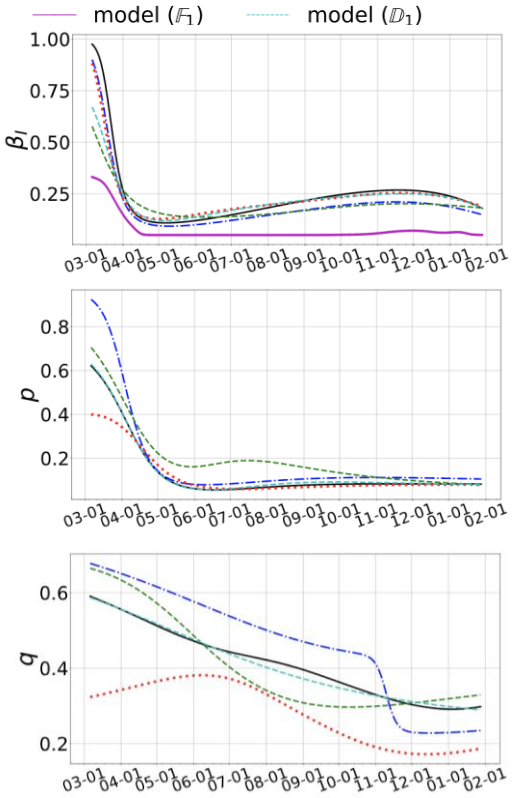
Fitting the data accurately



Discovering unobserved dynamics



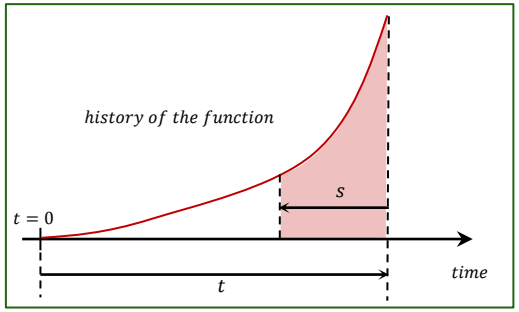
Inferring model parameters



Fractional Order Models Introduce Memory in the Dynamics

Caputo fractional derivative of order $\kappa \in (0,1)$: a **convolution** type **integro-differential** operator

$$\frac{\partial^\kappa}{\partial t^\kappa} u(t) = {}_0^C\mathcal{D}_t^\kappa\, u(t) = \frac{1}{\Gamma(1-\kappa)} \int_0^t \frac{1}{(t-s)^\kappa} \frac{du(s)}{ds}\, ds$$

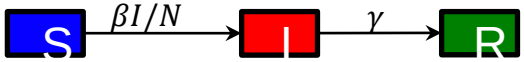


Memory: The derivative at time t depends on the weighted values of the function **from initial point $t = 0$ up to current time t.**

- Fractional order κ is the notion of memory effect
- Smaller κ can induce a delay in the dynamics
- $\kappa = \kappa(t)$ can be time varying

Different Compartments May Have Different Memory Effects!

model $\mathbb{F}_1$

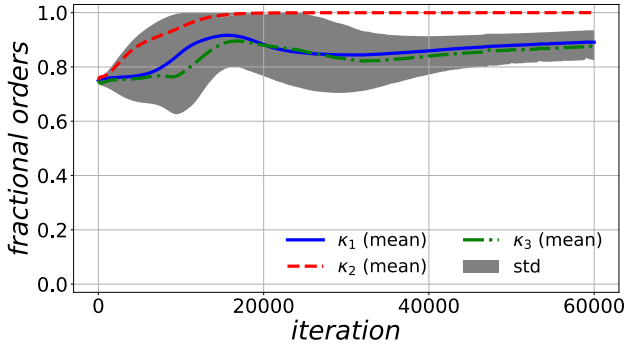


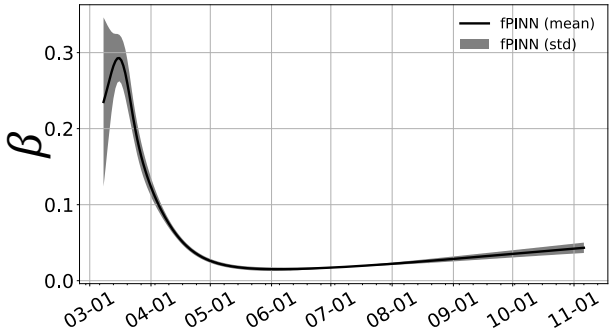
$$ {}_0^C\mathcal{D}_t^{\kappa_1}\mathbf{S}(t) = -\frac{\beta}{N}\mathbf{I}(t)\mathbf{S}(t), $$

$$ {}_0^C\mathcal{D}_t^{\kappa_2}\mathbf{I}(t) = \frac{\beta}{N}\mathbf{I}(t)\mathbf{S}(t) - \gamma\mathbf{I}(t), $$

$$ {}_0^C\mathcal{D}_t^{\kappa_3}\mathbf{R}(t) = \gamma\mathbf{I}(t), $$

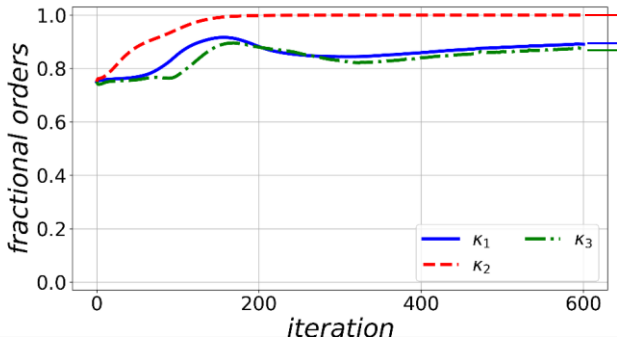
$$ {}_0^C\mathcal{D}_t^{\kappa_2}\mathbf{I}^c(t) = \frac{\beta}{N}\mathbf{I}(t)\mathbf{S}(t). $$



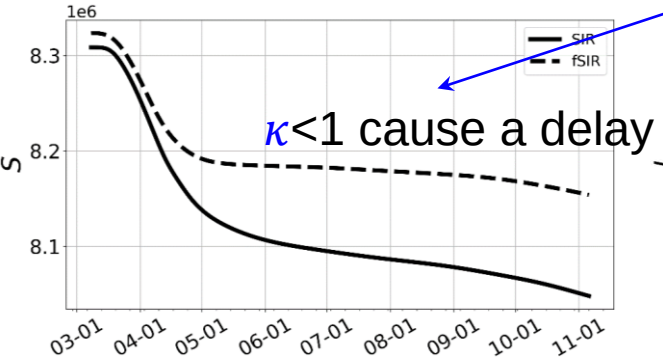


model $\mathbb{F}_1$

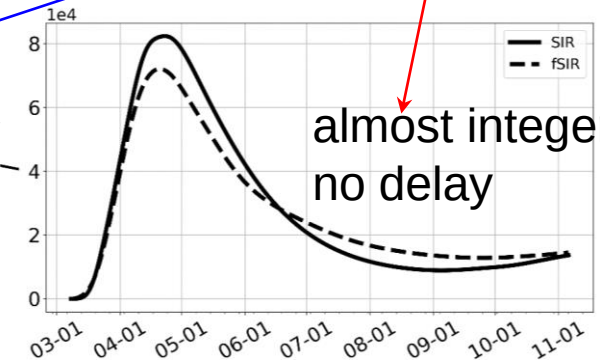
Fractional Order SIR V.S. Integer Order SIR



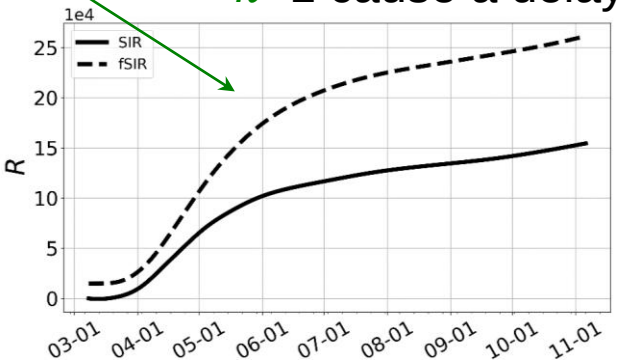
$\kappa_1 = 0.89$ $\kappa_2 = 0.99$ $\kappa_3 = 0.87$



κ<1 cause a delay



almost integer, no delay



κ<1 cause a delay

Summary

This is the first work to employ structural and practical identifiability tools to study COVID-19 model identifiability based on the available data.

A general data-driven epidemiological modeling framework is developed, which seamlessly integrates model identifiability, model sensitivity analysis, model calibration, model prediction with confidence intervals, and evaluating control strategies under uncertainties.

We treat beta (transmission rate), p (proportion of isolated individuals), and q (proportion of disease-related deaths) as time-dependent piece-wise model parameters and calibrate them using the available New York City COVID-19 dataset.

The developed COVID-19 model is employed to evaluate the effects of vaccination deployment scenarios.

We developed a flexible computational framework using physics-informed neural networks (PINNs) to study model uncertainty and discover time-dependent parameters.

Outline:

- Incorporate Physics Knowledge and AI to design new interpretable models
- **Incorporate Physics Knowledge into AI to predict multiscale problems: NH-PINN**
- Interpretable AI enables data-driven scientific discovery with uncertainty quantification capability – ALZHEIMER's Disease Prediction
- Sparse Neural Architecture Design with quantified uncertainties
- Scalable training large-scale Deep Neural Network

NH-PINN: Neural homogenizationbased the physics-informed neural network for the multiscale problems

Wing Tat Leung, Guang Lin, Zecheng Zhang, NH-PINN: Neural homogenization based Physics-informed Neural Network for Multiscale Problems, 2022, https://arxiv.org/abs/2108.12942

Content

1. Physics-informed nerual network (PINN)
2. Homogenization
3. Neural homogenization based PINN (NH-PINN)
4. Numerical examples

Wing Tat Leung, Guang Lin, Zecheng Zhang. NH-PINN: Neural homogenization based the physics-informed neural network for the multiscale problems. arXiv:2108.12942.

Physics-informed nerual network (PINN)

$$\mathcal{L}(u) = f \text{ in } \Omega$$
$$\mathcal{B}(u) = b \text{ on } \partial\Omega$$

where $\mathcal{L}$ is a differential operator and $\mathcal{B}$ is the boundary condition operator. f is the given source term, b is the given boundary condition.

$$\min_{\beta} \frac{w_1}{N_f} \sum_{i=1}^{N_f} |\mathcal{L}(\mathcal{F}_\beta(p_i)) - f(p_i)|^2 + \frac{w_2}{N_b} \sum_{i=1}^{N_b} |\mathcal{B}(\mathcal{F}_\beta(q_i)) - b(q_i)|^2, \tag{1}$$

where $w_1 + w_2 = 1$ are the positive weights; $\{p_i\} \subset \Omega$, $\{q_i\} \subset \partial\Omega$ and N_f, N_b are the number of points used in discretizing the domain and boundary respectively, $\mathcal{F}_\beta(\cdot)$ is the network and β is the parameters associated with the network.

Motivation

$$-\nabla \cdot (\kappa \nabla u(x)) = f, x \in \Omega,$$
$$u = 0, x \in \partial\Omega,$$

where $\Omega = [0, \pi]$ and $f = \sin(x)$, $\kappa(x) = 0.5 \sin(2\pi x/\epsilon) + 2$, where $\epsilon = \frac{1}{8}$.

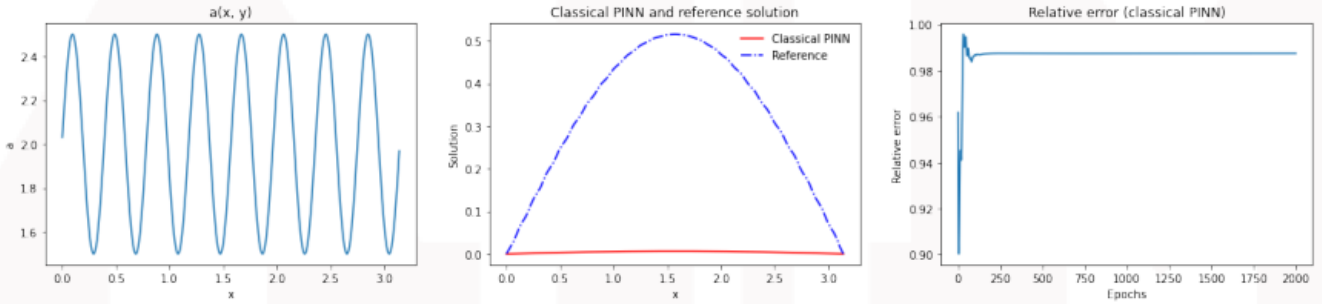


Figure: Left: demonstration of the permeability $\kappa(x) = 0.5 \sin(2\pi x/\epsilon) + 2$, where $\epsilon = 1/8$. Middle: learnt solution by the classical PINN vs the reference solution. Right: relative error as a function of the training epochs. The average error of the last 500 epochs is 0.987471.

Homogenization

Consider we are solving:

$$-\frac{\partial}{\partial x_i}\left(a_{ij}(x/\epsilon)\frac{\partial}{\partial x_j}u_\epsilon(x)\right) = f(x), x \in \Omega \tag{2}$$

with $u_\epsilon(x) = 0$ on $\partial\Omega$. The asymptotic expansion:

$$u_\epsilon = u_0 + \epsilon u_1 + \epsilon^2 u_2 + ... \tag{3}$$

1. Solve the cell problem:

$$-\frac{\partial}{\partial y_i}\left(a_{ij}(y)\frac{\partial}{\partial y_j}\right)\chi_j = \frac{\partial}{\partial y_i}a_{ij}(y), \tag{4}$$

$$\chi_j \text{ is periodic in } y \text{ with mean } 0. \tag{5}$$

2. Solve the homogenized equation:

$$-\frac{\partial}{\partial x_i}\left(a^*_{ij}\frac{\partial}{\partial x_j}\right)u_0 = f, \tag{6}$$

where $a^*_{ij} = \int_Y (a_{ij} + a_{ik}\frac{\chi_j}{\partial y_k})dy$

Neural homogenization based PINN (NH-PINN)

Proposed NH-PINN:

1. Solve the cell problems using PINN.
2. Evaluate the homogenized coefficients.
3. Solve the homogenized equation using PINN.

Oversampling (to improve PINN accuracy for solving periodic problems):

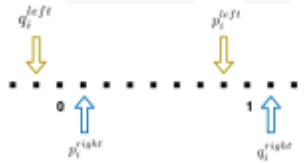


Figure: Oversampling mesh demonstration.

Suppose $w_1 + w_2 + w_3 = 1$ are positive constants, the new loss function:

$$\min_{\beta} \left\{ \frac{w_1}{N_f} \sum_{i=1}^{N_f} |\mathcal{L}(\mathcal{F}_\beta(p_i)) - f(p_i)|^2 + w_2 |(\mathcal{F}_\beta(q_1) - \mathcal{F}_\beta(q_2)|^2 \right.$$

$$\left. + \frac{w_3}{N_o} \sum_{i=1}^{N_0} \left(|(\mathcal{F}_\beta(q_i^{left}) - \mathcal{F}_\beta(p_i^{left})|^2 + |(\mathcal{F}_\beta(q_i^{right}) - \mathcal{F}_\beta(p_i^{right})|^2 \right) \right\}.$$

Notations for the numerical examples

Solution notation	Cell problems solver	Homogenized equation solver
$p(x)$	PINN	PINN
$v(x)$	FEM	FEM
$w(x)$	PINN	FEM

Table: Notations of the solutions. u_ϵ which is not listed here is the reference solution which solves the multiscale PDE directly by fine scale finite element methods.

Relative errors:

$$e_1 = \frac{\|p(x) - u_\epsilon(x)\|}{\|u_\epsilon\|}, e_2 = \frac{\|w(x) - u_\epsilon(x)\|}{\|u_\epsilon(x)\|},$$
$$e_3 = \frac{\|p(x) - w(x)\|}{\|w(x)\|}, e_4 = \frac{\|v(x) - u_\epsilon(x)\|}{\|u_\epsilon\|},$$

where $\|.\|$ is the L_2 norm.

Numerical examples

We consider the following $2D$ elliptic equation:

$$-\frac{\partial}{\partial x_i}\left(a(\frac{x}{\epsilon})\frac{\partial}{\partial x_i}u_\epsilon(x)\right) = f(x), x \in \Omega, \tag{7}$$

$$u_\epsilon(x) = 0, x \in \partial\Omega. \tag{8}$$

In our examples, $\Omega = [0,1]^2$ and the permeability $a(x/\epsilon) = 2 + \sin(2\pi x_1/\epsilon)\cos(2\pi x_2/\epsilon)$ and $\epsilon = \frac{1}{8}$.

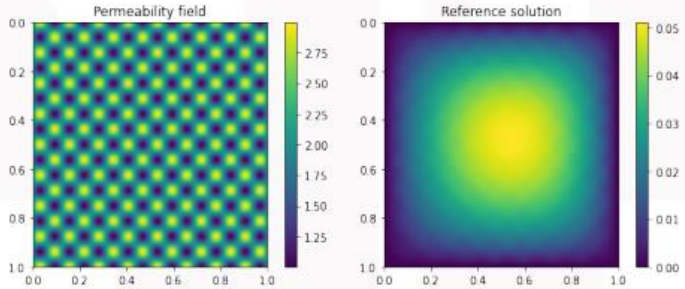


Figure: Left: permeability of the 2D elliptic problem. Note that $\epsilon = 1/8$. Right: the reference solution.

Oversampling and cell problems

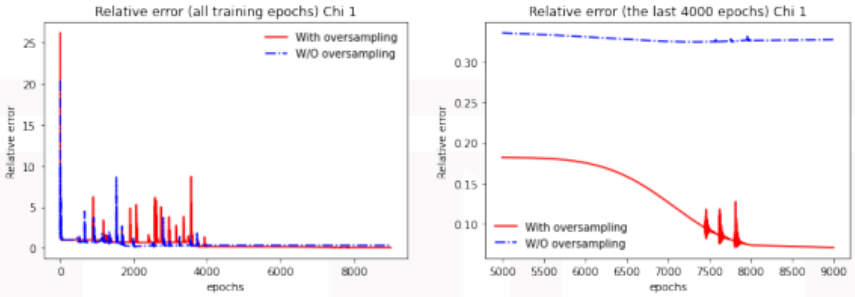


Figure: $2D$ elliptic cell problem χ_1. Relative error as a function of the training epochs for χ_1. Left: all training epochs; right: the last 4000 epochs. The average relative errors of the last 300 epochs for the oversampling and without oversampling are 0.072034 and 0.326963 respectively.

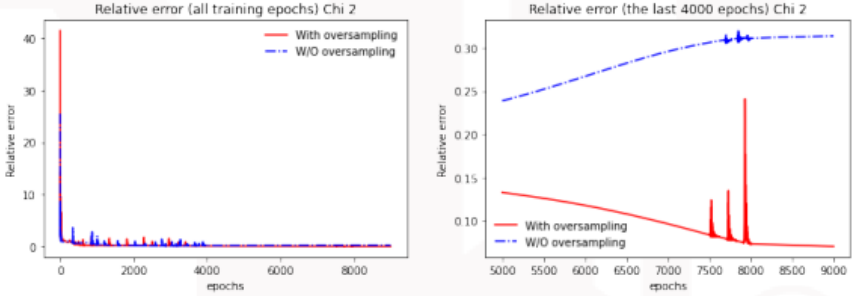


Figure: $2D$ elliptic cell problem χ_2. Relative error as a function of the training epochs for χ_2. Left: all training epochs; right: the last 4000 epochs. The average relative errors of the last 300 epochs for the oversampling and without oversampling are 0.071396 and 0.312961 respectively.

Convergence results

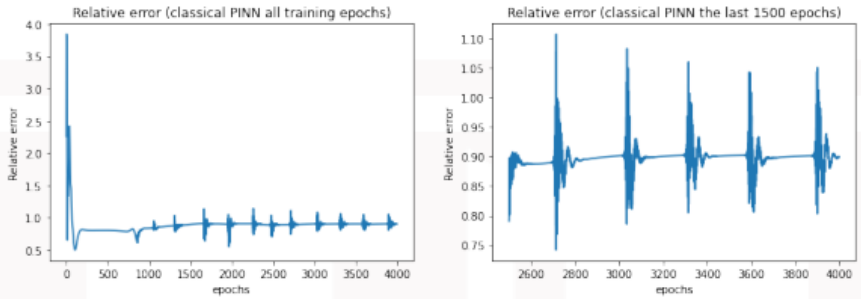


Figure: $2D$ elliptic problem by the classical PINN. Relative error as a function of the training epochs, the entire history (left), the last 1500 epochs (right).

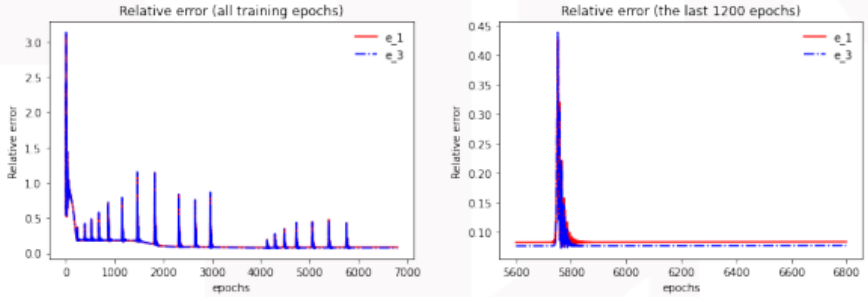


Figure: $2D$ elliptic cell problem e_1 and e_3 relative errors (NH-PINN) with respective to the training epochs. Left: history of all training epochs; right: history of the last 1200 epochs.

Analysis and transfer learning

Errors:

e_1	e_2	e_3	e_4
0.082994	0.0212534	0.076604	0.021316

Table: Relative errors for the $2D$ elliptic problem.

Transfer learning: We use the trained NH-PINN network to initialize the PINN. The failure shows that the minimizers of two loss functions are not closed. Modify the PINN loss function should be the future.

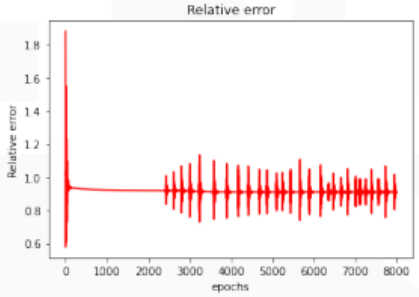


Figure: 2D elliptic transfer learning of classical PINN.

Conclusion

1. We find that PINN accuracy on solving multiscale problems degenerates. We propose a 3-step approach, neural homogenization based PINN (NH-PINN). We first apply PINN to solve the cell problems which are used to derive the homogenized equation; the homogenized equation can then be easily solved by PINN.
2. We propose an oversampling strategy to solve the periodic PDE by PINN; this method greatly improves the accuracy of the PINN when solving the high dimensional periodic problems.
3. We also observe that NH-PINN can improve the homogenization accuracy. If we apply PINN to implement homogenization, the solution may be more accurate than the traditional numerical methods. PINN may be a potential alternative of implementing the homogenization.

Outline:

- Incorporate Physics Knowledge and AI to design new interpretable models
- Incorporate Physics Knowledge into AI to predict multiscale problems: NH-PINN
- **Interpretable AI enables data-driven scientific discovery with uncertainty quantification capability – ALZHEIMER's Disease Prediction**
- Sparse Neural Architecture Design with quantified uncertainties
- Scalable training large-scale Deep Neural Network

Interpretable AI:

Question: Can we use available observation data to discover the physical laws?

Goal: Enable Data-driven Scientific Discovery?

S. Zhang, **G. Lin**, Robust data-driven discovery of governing physical laws with error bars, Proceedings of the Royal Society of London. Series A, mathematical, physical and engineering sciences, in press, 2018.

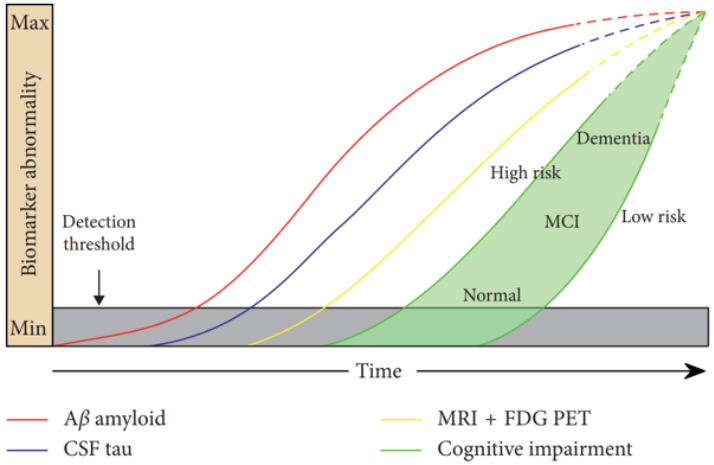
Jiuhai Chen, Lulu Kang, Guang Lin, Gaussian process assisted active learning of physical laws, Technometrics, in press, 2020.
https://doi.org/10.1080/00401706.2020.1817790

Sheng Zhang, Guang Lin, Robust subsampling-based threshold sparse Bayesian regression to tackle high noise and outliers for data-driven discovery of differential equations, Journal of Computational Physics, 428: 109962, 2021.

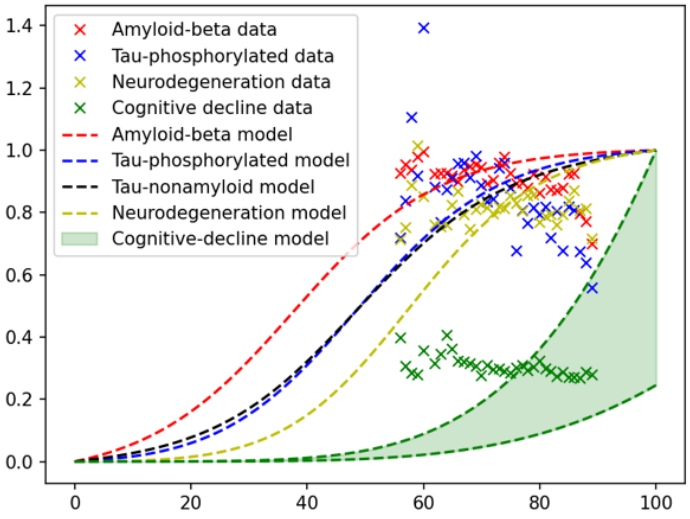
ALZHEIMER'S DISEASE PREDICTION

Haoyang Zheng, Jeffrey Petrella, P.Murali Doraiswamy, Guang Lin, Wenrui Hao, Nature Medicine, in review, 2022

AD prediction

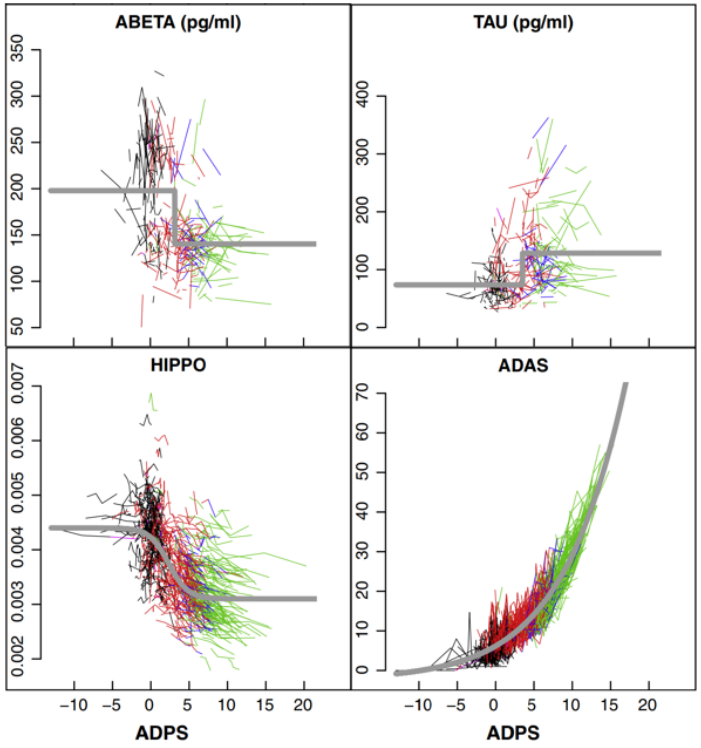


How to apply patient data (ADNI dataset) to optimize ODEs?



PURDUE UNIVERSITY | School of Mechanical Engineering

Dataset



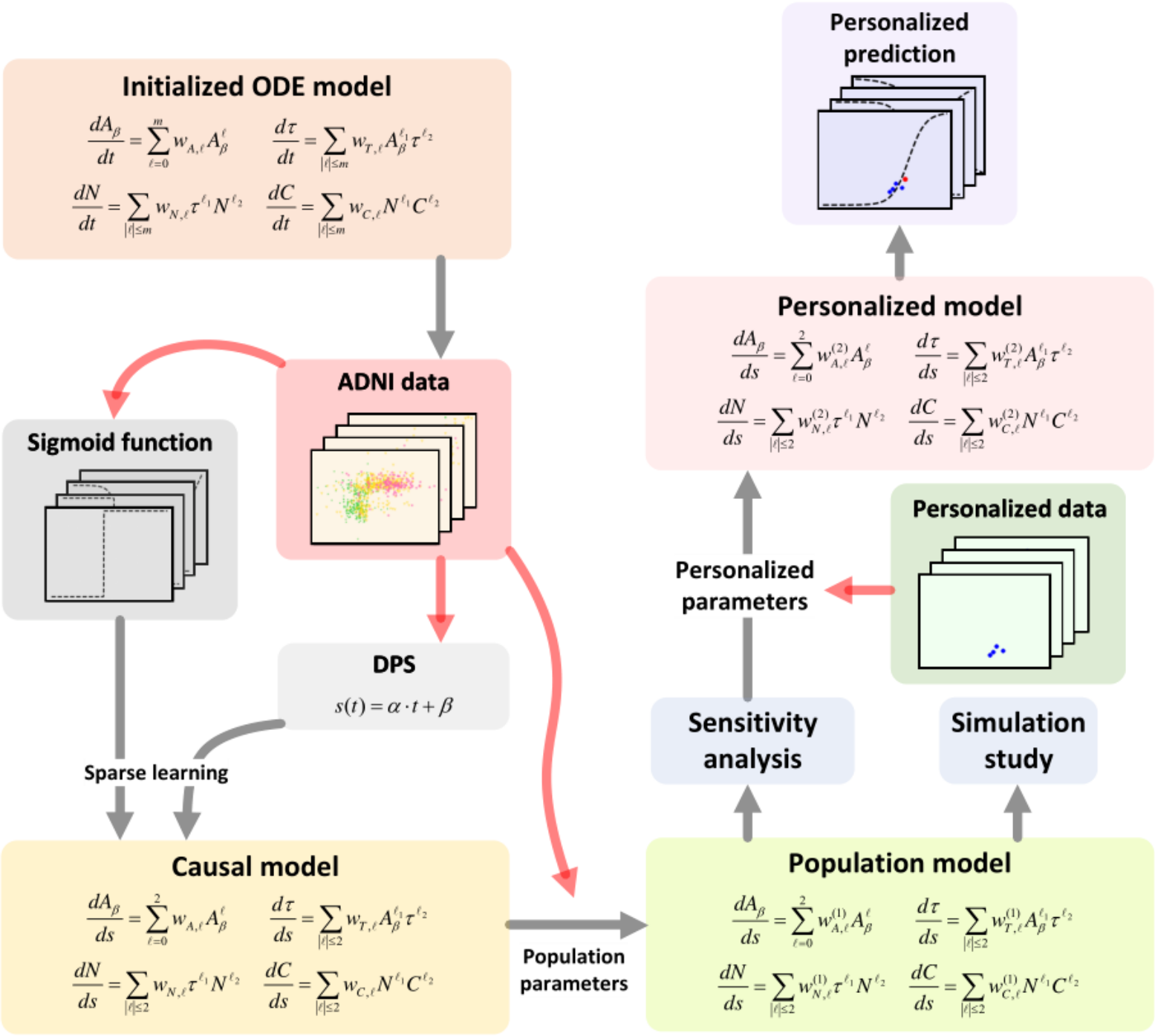
Progression of ADNI biomarkers as function of the Alzheimer's Disease Progression Score (ADPS)

- Normal–Normal
- MCI–MCI
- AD–AD
- MCI–AD
- MCI–N
- sigmoids

Disease progression score (DPS)

- Patients' disease progression differs in their age of onset and rate of progression.
- Each biomarker follows a sigmoid shaped curve.

Jedynak, Bruno M., et al. "A computational neurodegenerative disease progression score: method and results with the Alzheimer's disease neuroimaging initiative cohort." *Neuroimage* 63.3 (2012): 1478-1486.


Initialized ODE model
$\frac{dA_\beta}{dt} = \sum_{\ell=0}^{m} w_{A,\ell} A_\beta^\ell$
$\frac{d\tau}{dt} = \sum_{|\ell|\le m} w_{T,\ell} A_\beta^{\ell_1} \tau^{\ell_2}$
$\frac{dN}{dt} = \sum_{|\ell|\le m} w_{N,\ell} \tau^{\ell_1} N^{\ell_2}$
$\frac{dC}{dt} = \sum_{|\ell|\le m} w_{C,\ell} N^{\ell_1} C^{\ell_2}$
ADNI data
Sigmoid function
DPS
$s(t) = \alpha \cdot t + \beta$
Sparse learning
Causal model
$\frac{dA_\beta}{ds} = \sum_{\ell=0}^{2} w_{A,\ell} A_\beta^\ell$
$\frac{d\tau}{ds} = \sum_{|\ell|\le 2} w_{T,\ell} A_\beta^{\ell_1} \tau^{\ell_2}$
$\frac{dN}{ds} = \sum_{|\ell|\le 2} w_{N,\ell} \tau^{\ell_1} N^{\ell_2}$
$\frac{dC}{ds} = \sum_{|\ell|\le 2} w_{C,\ell} N^{\ell_1} C^{\ell_2}$
Population parameters
Population model
$\frac{dA_\beta}{ds} = \sum_{\ell=0}^{2} w_{A,\ell}^{(1)} A_\beta^\ell$
$\frac{d\tau}{ds} = \sum_{|\ell|\le 2} w_{T,\ell}^{(1)} A_\beta^{\ell_1} \tau^{\ell_2}$
$\frac{dN}{ds} = \sum_{|\ell|\le 2} w_{N,\ell}^{(1)} \tau^{\ell_1} N^{\ell_2}$
$\frac{dC}{ds} = \sum_{|\ell|\le 2} w_{C,\ell}^{(1)} N^{\ell_1} C^{\ell_2}$
Sensitivity analysis
Simulation study
Personalized parameters
Personalized data
Personalized model
$\frac{dA_\beta}{ds} = \sum_{\ell=0}^{2} w_{A,\ell}^{(2)} A_\beta^\ell$
$\frac{d\tau}{ds} = \sum_{|\ell|\le 2} w_{T,\ell}^{(2)} A_\beta^{\ell_1} \tau^{\ell_2}$
$\frac{dN}{ds} = \sum_{|\ell|\le 2} w_{N,\ell}^{(2)} \tau^{\ell_1} N^{\ell_2}$
$\frac{dC}{ds} = \sum_{|\ell|\le 2} w_{C,\ell}^{(2)} N^{\ell_1} C^{\ell_2}$
Personalized prediction

Fit ODEs with the use of patient data

Results

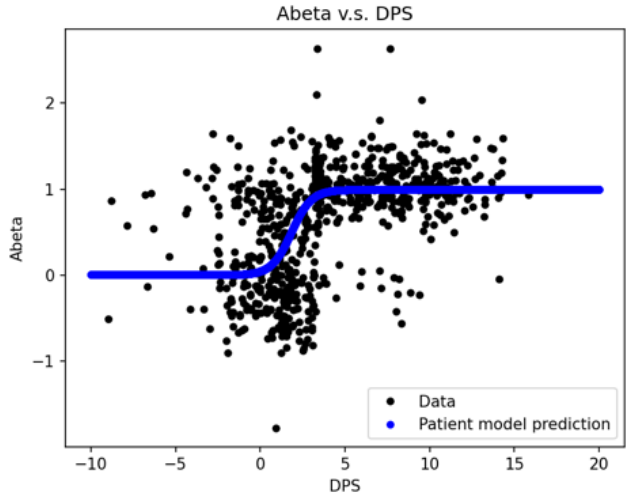
$$\frac{dA_\beta(t)}{dt} = -1.7112 \cdot (A_\beta(t) + 0.005) \cdot (A_\beta(t) - 1) + 0.0045;$$

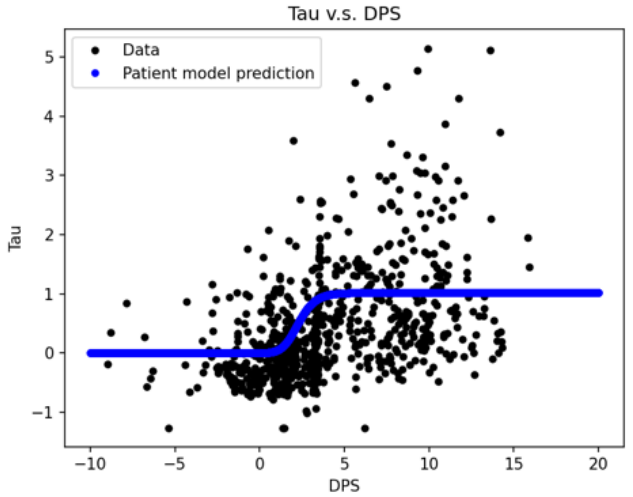
$$\frac{d\tau_\rho(t)}{dt} = -0.0276 \cdot A_\beta(t) + 3.6918 \cdot A_\beta^2(t) - 0.2498 \cdot \tau_\rho(t)[\tau_\rho(t) - 1 + 14.2932 \cdot A_\beta(t)];$$

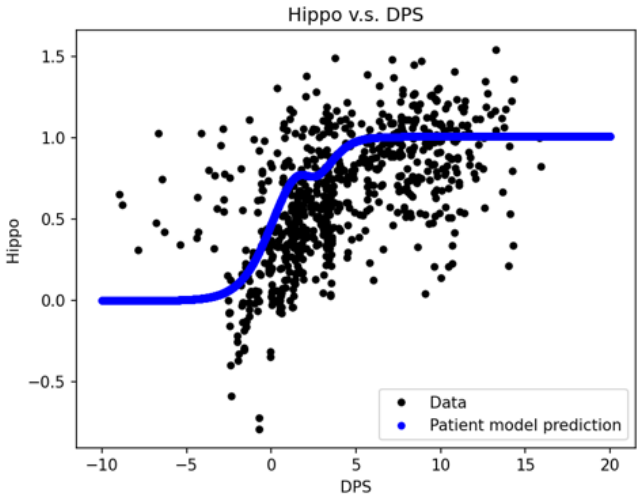
$$\frac{dN(t)}{dt} = -0.5106 \cdot \tau_\rho(t) + 1.0708 \cdot \tau_\rho^2(t) - 1.0257 \cdot N(t)[N(t) - 1 + 0.5533 \cdot \tau_\rho(t)]$$

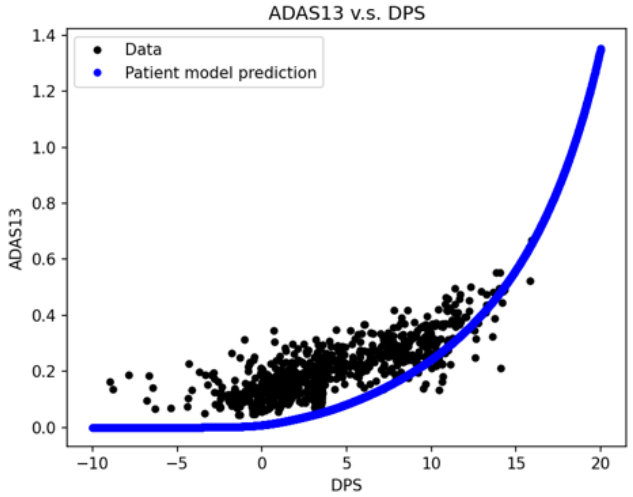
$$\frac{dC(t)}{dt} = -0.0017 \cdot N(t) + 5.9 \cdot 10^{-7} \cdot N^2(t) - 0.0372 \cdot C(t)[C(t) - 2 - 0.0118 \cdot N(t)]$$

Results


Abeta v.s. DPS
Abeta
DPS
Data
Patient model prediction


Tau v.s. DPS
Tau
DPS
Data
Patient model prediction


Hippo v.s. DPS
Hippo
DPS
Data
Patient model prediction


ADAS13 v.s. DPS
ADAS13
DPS
Data
Patient model prediction

Outline:

- Incorporate Physics Knowledge and AI to design new interpretable models
- Incorporate Physics Knowledge into AI to predict multiscale problems: NH-PINN
- Interpretable AI enables data-driven scientific discovery with uncertainty quantification capability – ALZHEIMER's Disease Prediction
- **Sparse Neural Architecture Design with quantified uncertainties**
- Scalable training large-scale Deep Neural Network

Sparse Neural Architecture Design with quantified uncertainties:

Question: Can we develop a fast, small & accurate deep neural network with better interpretability and less demanding on the computational resource?

Goal: Enable Fast Interpretable Nonlinear Data-driven Scientific Discovery.

W. Deng, X. Zhang, F. Liang, **G. Lin**, An adaptive empirical Bayesian method for sparse deep learning, **2019 Conference on Neural Information Processing Systems (NIPS)**, Dec. 8 – Dec. 14, 2019, Vancouver, Canada.

NeurIPS'19, NeurIPS'20, ICML'20, ICLR'21, JCP'20, JCP'21a, JCP'21b

Bayesian Sparse learning with preconditioned stochastic gradient MCMC and its applications

Guang Lin[1]

Joint work with Yating Wang, Wei Deng, Xiao Zhang, Faming Liang

[1]Departments of Mathematics, Statistics & School of Mechanical Engineering

Purdue University

NeurIPS'19, JCP'20

Outline

Motivation and Objective

DNN challenges:

- Over-parameterized DNN requires heavy memory and computation power, and may cause overfitting
- Cost function is difficult to optimize and a good local minima is hard to obtain

Objective:

- Sparse learning to enhance efficiency, robustness and interpretability
- Bayesian approach to capture uncertainty
- Compute posterior expectation to obtain more robust result
- Escape "shallow" local optimas and saddle points to achieve better point estimate

Notations

- The entire data: $\mathcal{D} = \{d_i\}_{i=1}^N$, a mini-batch of data $\mathcal{B}$
- Model parameters: $\boldsymbol{\beta} \in \mathbb{R}^D$
- Log posterior density $L(\boldsymbol{\beta}) = \log(p(\boldsymbol{\beta}|\mathcal{D}))$ and true gradient

$$\nabla_{\boldsymbol{\beta}} L(\boldsymbol{\beta}) = \nabla_{\boldsymbol{\beta}} \log p(\boldsymbol{\beta}) + \sum_{i=1}^{N} \nabla_{\boldsymbol{\beta}} \log p(d_i|\boldsymbol{\beta})$$

- Stochastic gradient using a mini-batch is $\nabla \tilde{L}(\boldsymbol{\beta})$.

$$\nabla_{\boldsymbol{\beta}} \tilde{L}(\boldsymbol{\beta}) = \nabla_{\boldsymbol{\beta}} \log p(\boldsymbol{\beta}) + \frac{N}{n} \sum_{i=1}^{n} \nabla_{\boldsymbol{\beta}} \log p(d_i|\boldsymbol{\beta})$$

Key Components:

- **Sparse learning** - Enhance efficiency, robustness and interpretability
- **Bayesian approach** - Capture uncertainty
- **Empirical Bayesian method** - Learn a class of hierarchical Bayes models, yield data-driven adaptive penalties
- **Adaptive Stochastic Gradient Langevin Dynamics (SGLD) or SG-MCMC** - Capture parameter uncertainty and avoid overfitting, escaping "shallow" local optima and saddle points, and compute posterior expectation to obtain more robust result
- **Preconditioned SGLD (PSGLD)** - Update parameters with different step size, adaptive to local geometric and resulting in faster convergence for components of β have different scales
- **Stochastic Approximation (SA)** - Optimize latent variables in prior (SGLD-SA, PSGLD-SA) to converge to the asymptotically correct distribution with a controllable bias
- **Pruning Strategy** - Enable Sparse Deep Neural Network; Sparsity is ensured, resulting in less usage in memory and computational power

SGD

SGD update the parameters using

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \epsilon_k \nabla_{\boldsymbol{\beta}} \tilde{L}(\boldsymbol{\beta}_k)$$

- Find MAP for model parameter through stochastic optimization
- Do not capture parameter uncertainty and can potentially overfit data.

Langevin Dynamics and SGLD

- A Langevin diffusion with stationary distribution $p(\boldsymbol{\beta})$ can be described by the SDE

$$d\boldsymbol{\beta}(t) = \nabla_{\boldsymbol{\beta}} L(\boldsymbol{\beta})dt + \sqrt{2}dW(t) \tag{1}$$

  where $W(t)$ is a Brownian motion.
- Euler discretization of (1), and approximate the true gradient by the stochastic gradient:

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k + \epsilon_k \nabla_{\boldsymbol{\beta}} \tilde{L}(\boldsymbol{\beta}_k) + N(0, 2\epsilon_k \tau^{-1} I) \tag{2}$$

  (2) asymptotically converges to $\pi(\boldsymbol{\beta}|\mathcal{D}) \propto e^{\tau L(\boldsymbol{\beta})}$. As τ increases, ϵ_k decreases, the solution tends to the global optima with a higher probability.

Advantages of SGLD

- Sample DNN posterior to model uncertainty
- Compute posterior expectation to obtain more robust result
- Escape local optimas to achieve better point estimate

SGLD updates all parameters with the same step size, this may cause slow mixing when components of β have different scales.

A class of Adaptive Stochastic Gradient MCMC

Stochastic gradient Langevin Dynamics (SGLD), the first order SG-MCMC algorithm, is a sampling algorithm in DNN which asymptotically converges to a stationary distribution of $e^{\tau L(\boldsymbol{\beta})}$:

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + \epsilon^{(k)} \nabla_{\boldsymbol{\beta}} \tilde{L}(\boldsymbol{\beta}^{(k)}) + \mathcal{N}(0, 2\epsilon^{(k)} \tau^{-1}), \tag{4}$$

For the adaptive SGLD, we introduce an auxiliary variable $\boldsymbol{\theta}$ and formulate an inhomogeneous Markov Chain as follows:

$$\begin{aligned} \boldsymbol{\beta}^{(k+1)} &= \boldsymbol{\beta}^{(k)} + \epsilon^{(k)} \nabla_{\boldsymbol{\beta}} \tilde{L}(\boldsymbol{\beta}^{(k)}, \boldsymbol{\theta}^{(k)}) + \tilde{\boldsymbol{\eta}}^{(k)}, \\ \boldsymbol{\theta}^{(k+1)} &= \boldsymbol{\theta}^{(k)} + \omega^{(k+1)} H(\boldsymbol{\theta}^{(k)}, \boldsymbol{\beta}^{(k+1)}), \end{aligned} \tag{5}$$

where $\tilde{\boldsymbol{\eta}}^{(k)} \sim \mathcal{N}(0, 2\epsilon^{(k)}/\tau \boldsymbol{I})$, $H(\boldsymbol{\theta}, \boldsymbol{\beta})$ is a **biased but aymptotically unbiased estimator** of the mean field function $h(\boldsymbol{\theta})$. The interpretation of this algorithm is that we sample $\boldsymbol{\beta}^{(k+1)}$ from $\tilde{L}(\boldsymbol{\beta}^{(k)}, \boldsymbol{\theta}^{(k)})$ and adaptively optimize $\boldsymbol{\theta}$ such that $\langle \boldsymbol{\theta} - \boldsymbol{\theta}^*, h(\boldsymbol{\theta}) \rangle = 0$.

Pruning Strategy - Enable Sparse Deep Neural Network

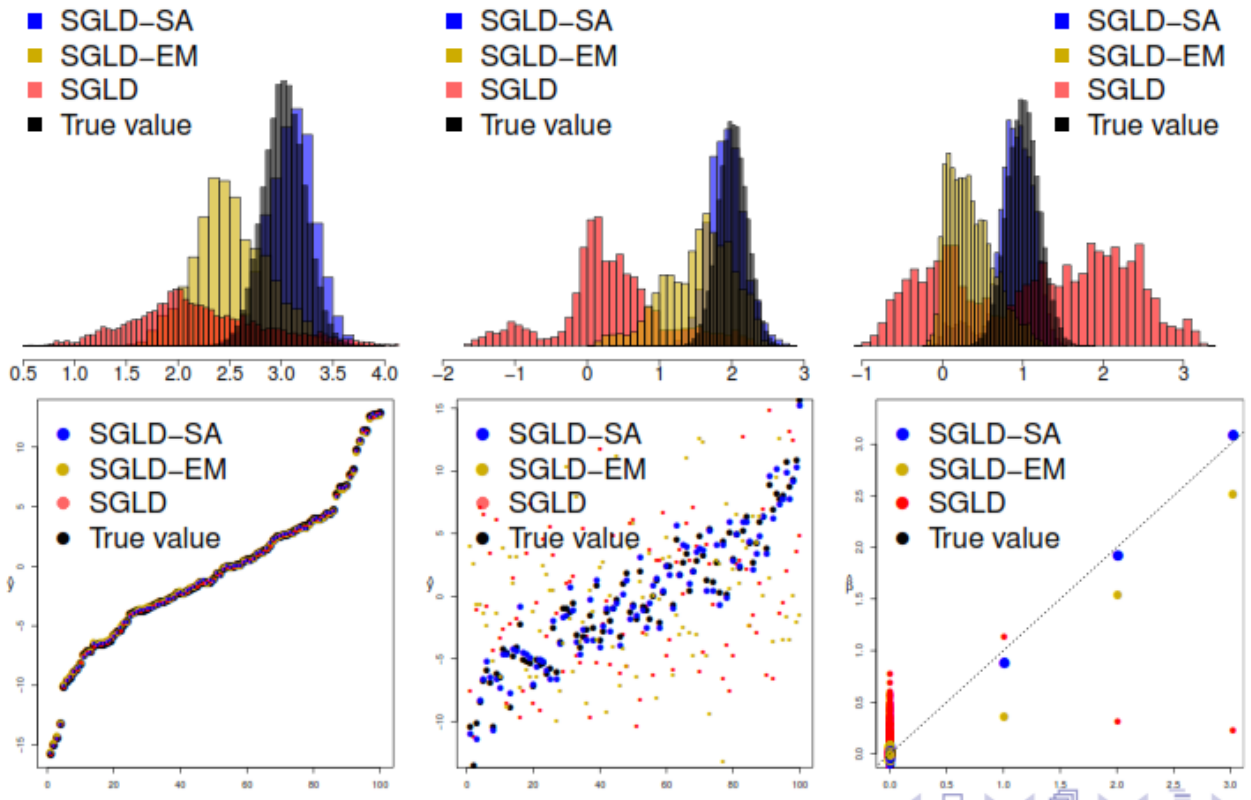
Although the magnitude-based unit pruning shows more computational savings, it doesn't demonstrate robustness under coarser pruning. We instead apply the magnitude-based weight-pruning to our compression experiments. The weight pruning can also be viewed as a greedy L^0 regularization in optimization.

Preconditioned SGLD (PSGLD)

- Updating parameters with different step size, adaptive to local geometric and resulting in faster convergence for components of β have different scales

Simulation of Large-p-Small-n Regression

Dataset: $n = 100$ and $p = 1000$. $\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\eta}$ where $\boldsymbol{X} \sim \mathcal{N}_p(0, \boldsymbol{\Sigma})$, $\boldsymbol{\beta} = (\beta_1, \beta_2, \beta_3, 0, 0, ..., 0)'$, $\boldsymbol{\eta} \sim \mathcal{N}_n(\boldsymbol{0}, 3\boldsymbol{I}_n)$, $\beta_1 \sim \mathcal{N}(3, \sigma_c^2)$, $\beta_2 \sim \mathcal{N}(2, \sigma_c^2)$, $\beta_3 \sim \mathcal{N}(1, \sigma_c^2)$, $\sigma_c = 0.2$.



Residual Network Compression

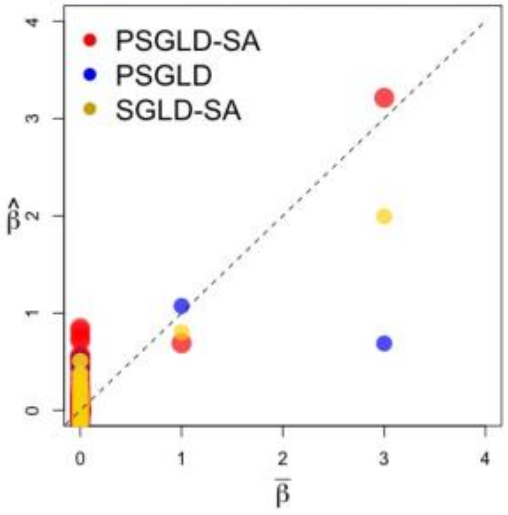
Table 1: Resnet20 Compression on CIFAR10. $L_2 = 1 \times 10^{-4}$ means we apply weight decay 1×10^{-4} to the sparse layers with target sparse rate $\mathbb{S}$.

PENALTY \ $\mathbb{S}$	30%	50%	70%	90%
$L_1 = 1 \times 10^{-3}$	92.88	92.75	92.62	89.95
$L_1 = 1 \times 10^{-2}$	89.50	89.79	90.07	89.83
$L_2 = 1 \times 10^{-2}$	94.17	93.82	93.17	90.11
$L_2 = 1 \times 10^{-1}$	94.02	93.96	93.50	91.20
SGHMC-SA	**94.23**	**94.27**	**93.74**	**91.68**

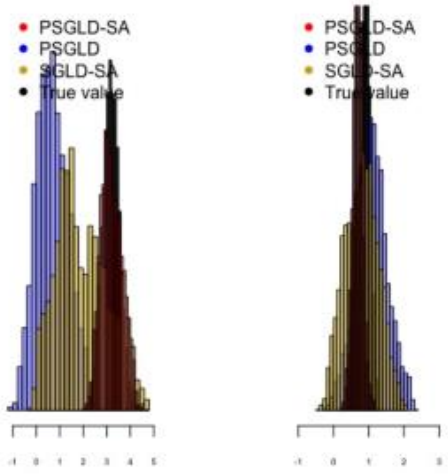
Most notably, **91.68% accuracy based on 27K parameters (90% sparsity) in Resnet20 is the besting existing result**. By contrast, targeted dropout (2018) achieved 91.48% accuracy based on 47K parameters (90% sparsity) of Resnet32, BC-GHS (2017) achieved 91.0% accuracy based on 8M parameters (94.5% sparsity) of VGG models.

Small n large p problem

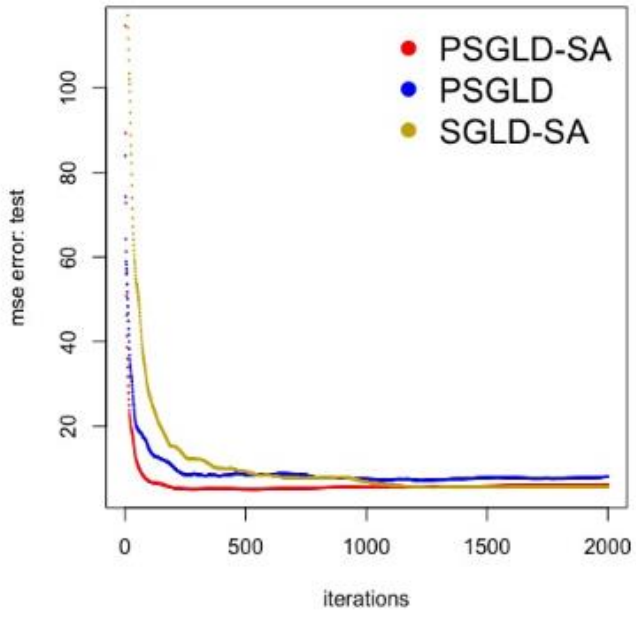
Number of observations $n = 100$, Number of predictors $p = 200$.
Predictors: $X \in \mathcal{N}(0, \Sigma)$, $\Sigma_{ij} = 0.6^{|i-j|}$, $X[:, 1] * 0.3$.
Model parameters: $\beta_1 = 3, \beta_2 = 1, \beta_j = 0$, for $j = 1, \cdots, p$.
Responses: $y = X\boldsymbol{\beta} + \epsilon$, $\epsilon \sim \mathcal{N}_n(0, 3I_n)$.



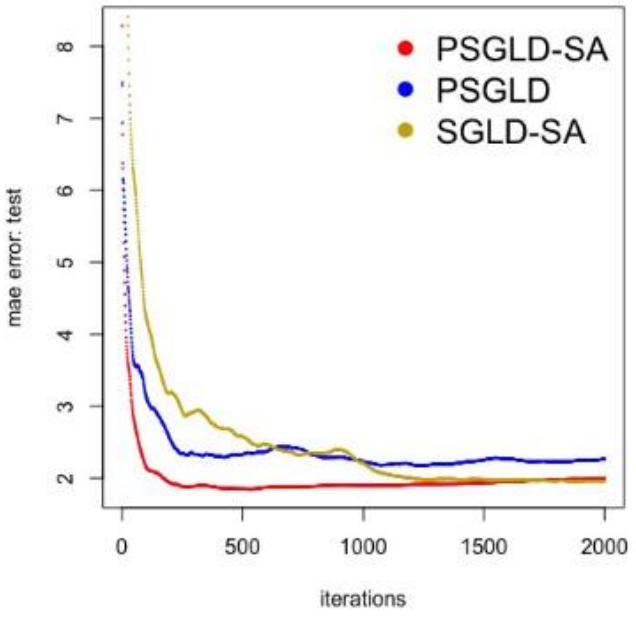
(a) Posterior mean vs true



(b) Posterior estimation of β_1, β_2



(c) Testing MSE error history



(d) Testing MAE error history

Multiscale flow problem

Flow equation:

$$\begin{aligned} \kappa^{-1}u + \nabla p &= 0 \quad \text{in} \quad D \\ \text{div}(u) &= f \quad \text{in} \quad D \\ u \cdot n &= 0 \quad \text{on} \quad \partial D \end{aligned}$$

κ: heterogeneous permeability field.
On fine grid $\mathcal{T}_h$, using Mixed finite element method:

$$\begin{bmatrix} A_h(\kappa) & B_h^T \\ B_h & 0 \end{bmatrix} \begin{bmatrix} u_h \\ p_h \end{bmatrix} = \begin{bmatrix} 0 \\ -F \end{bmatrix} \tag{9}$$

Velocity approximation: RT_0 (the lowest order Raviart-Thomas element) .
Pressure approximation: P_0 (piecewise constant element).
Difficulties: $A_h(\kappa)$ is large, and depends on κ.

Network architecture

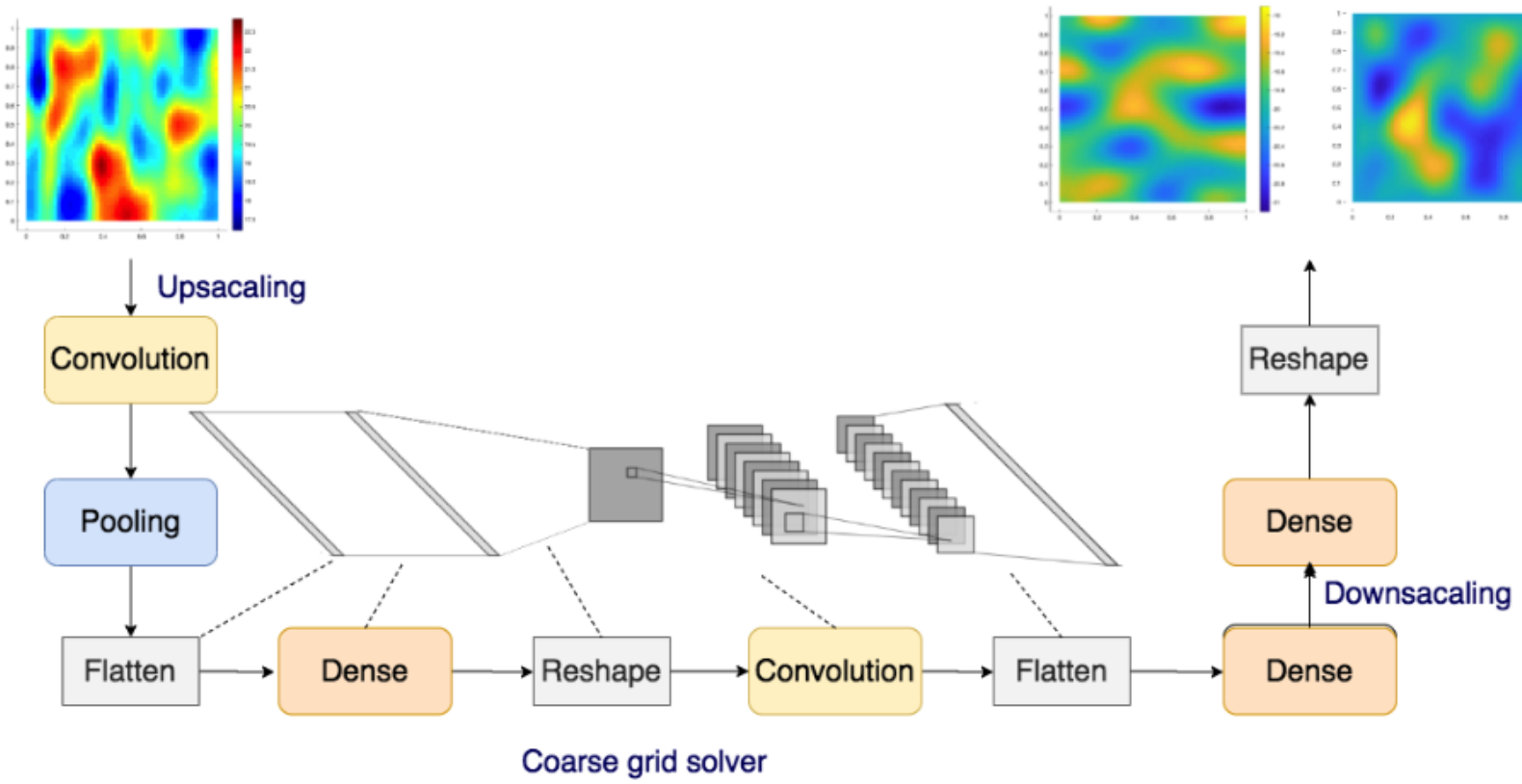
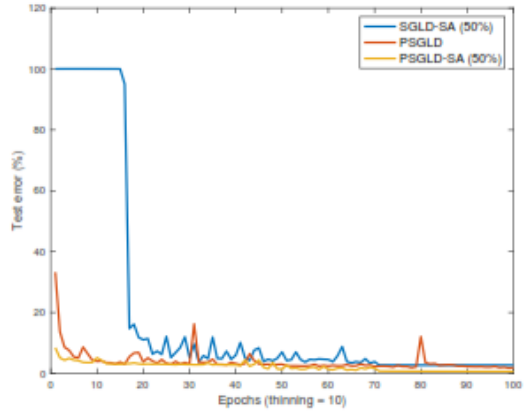


Figure 2: An illustration of the network architecture for flow approximation.

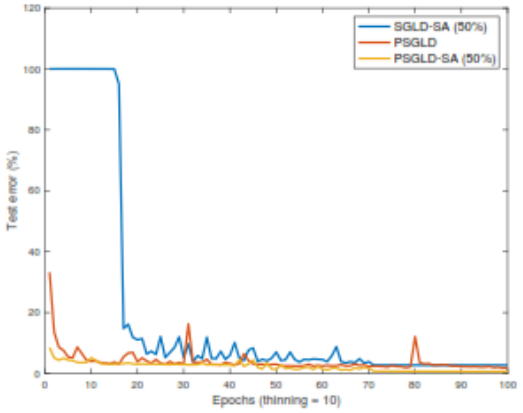
Numerical results

Dense (6,110,624 weight parameters)		
	PSGLD (e_1/ e_2 %)	SGLD(e_1/ e_2 %)
KLE32	0.75/0.57	2.37 /2.17
KLE64	0.82/0.63	2.38 /2.25
KLE128	2.13 /1.93	2.90 /2.60
	PSGLD-SA (e_1/ e_2 %)	SGLD-SA (e_1/ e_2 %)
Sparse rate 50% (2,326,049 weight parameters)		
KLE32	0.59/0.56	2.67 /2.35
KLE64	0.78 /0.58	2.68 /2.41
KLE128	1.60 /1.31	3.47 /3.00
Sparse rate 70% (758,738 weight parameters)		
KLE32	0.58/ 0.51	2.28 /2.10
KLE64	0.76 /0.61	2.40 /2.97
KLE128	1.79 /1.60	3.51/3.02

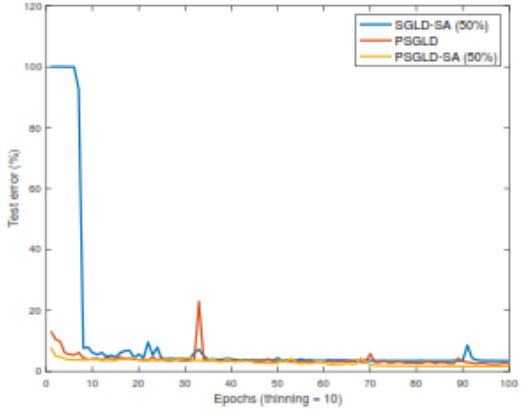
Table 2: Errors between the true velocity and predicted velocity (from trained neural networks) using SGLD, PSGLD, SGLD-SA, and proposed PSGLD-SA. Mean errors of 300 testing cases.



(a) KLE 32, learning curves



(b) KLE 64, learning curves



(c) KLE 128, learning curves

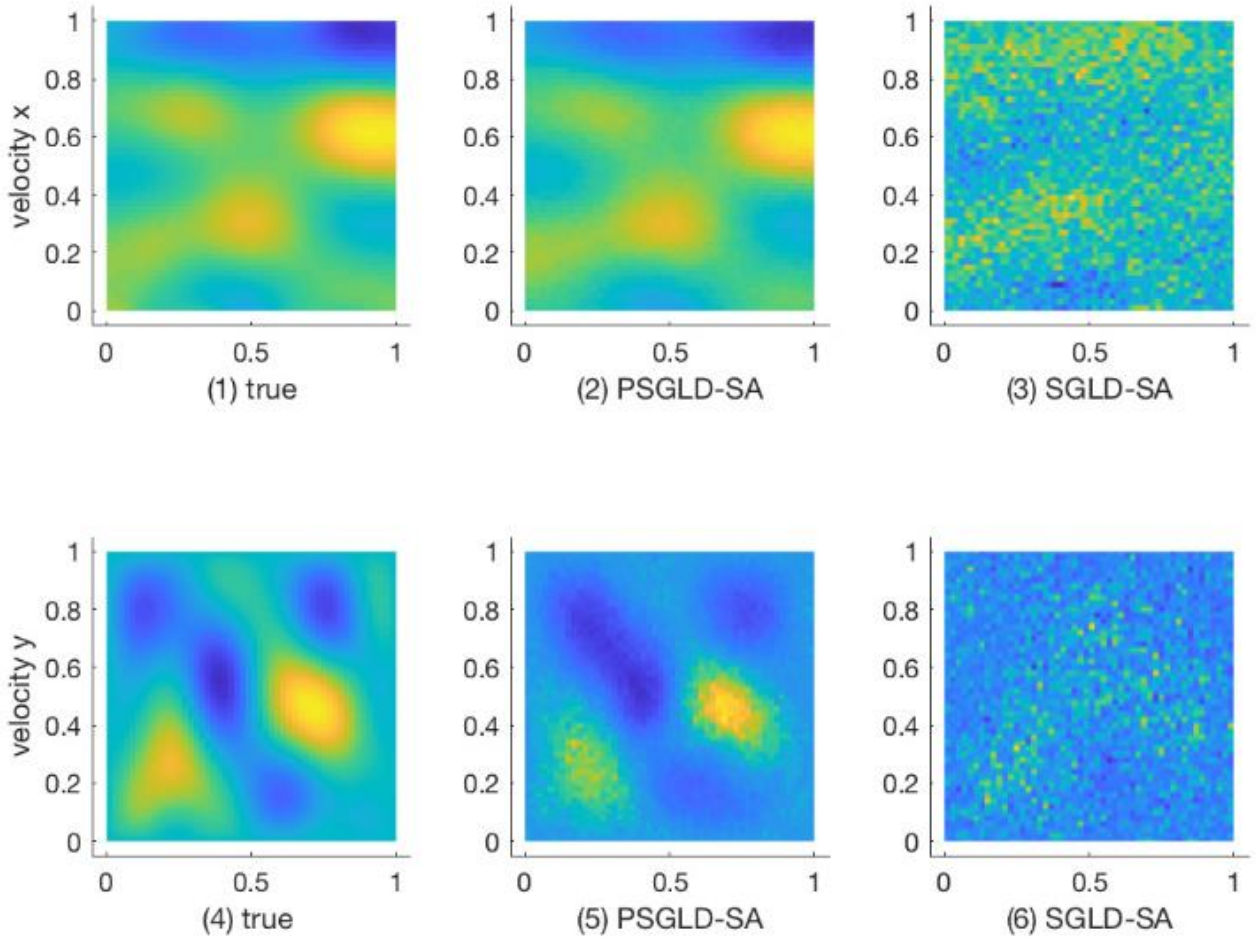


Figure 3: KLE 32. True and prediction solutions.

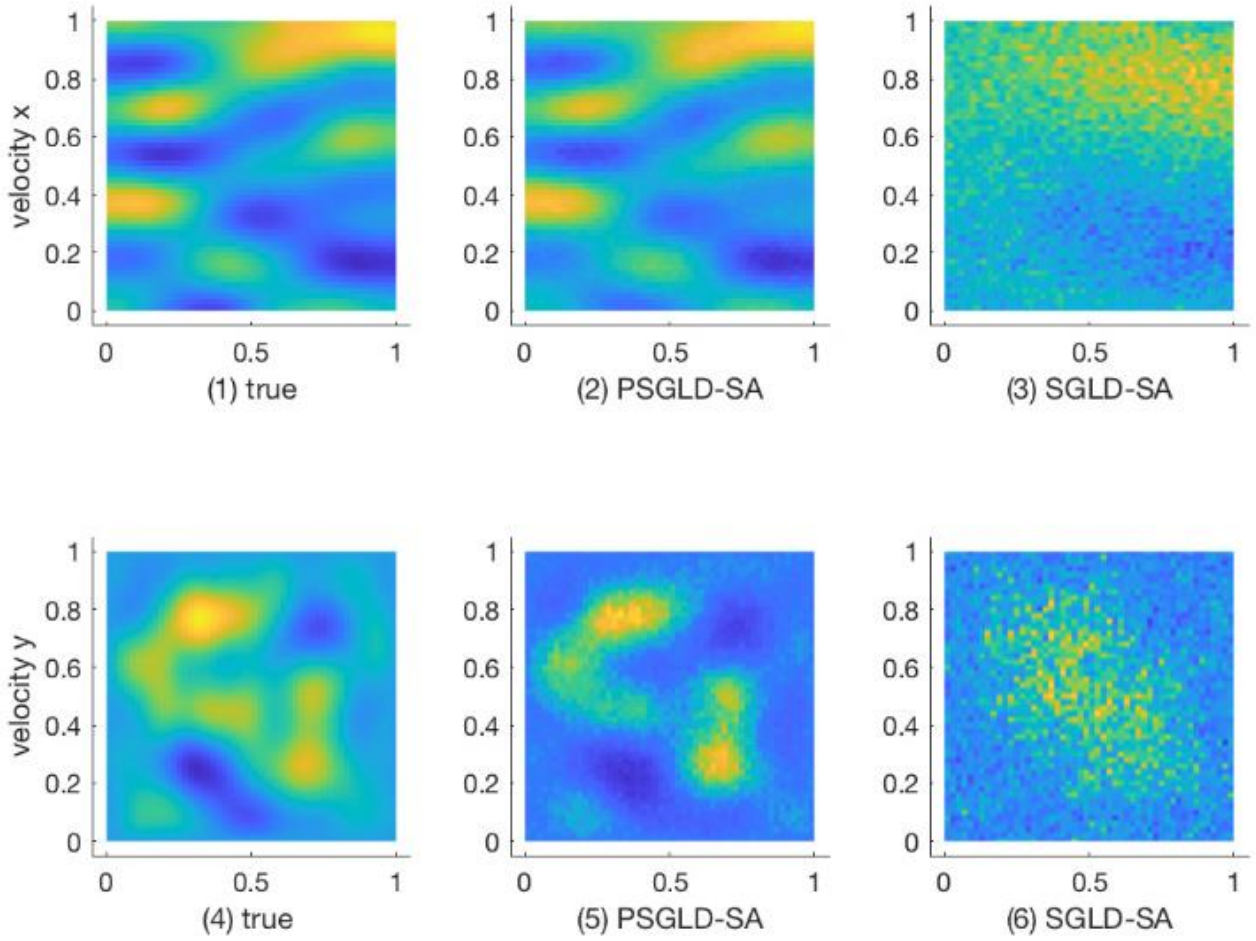


Figure 4: KLE 64. True and prediction solutions.

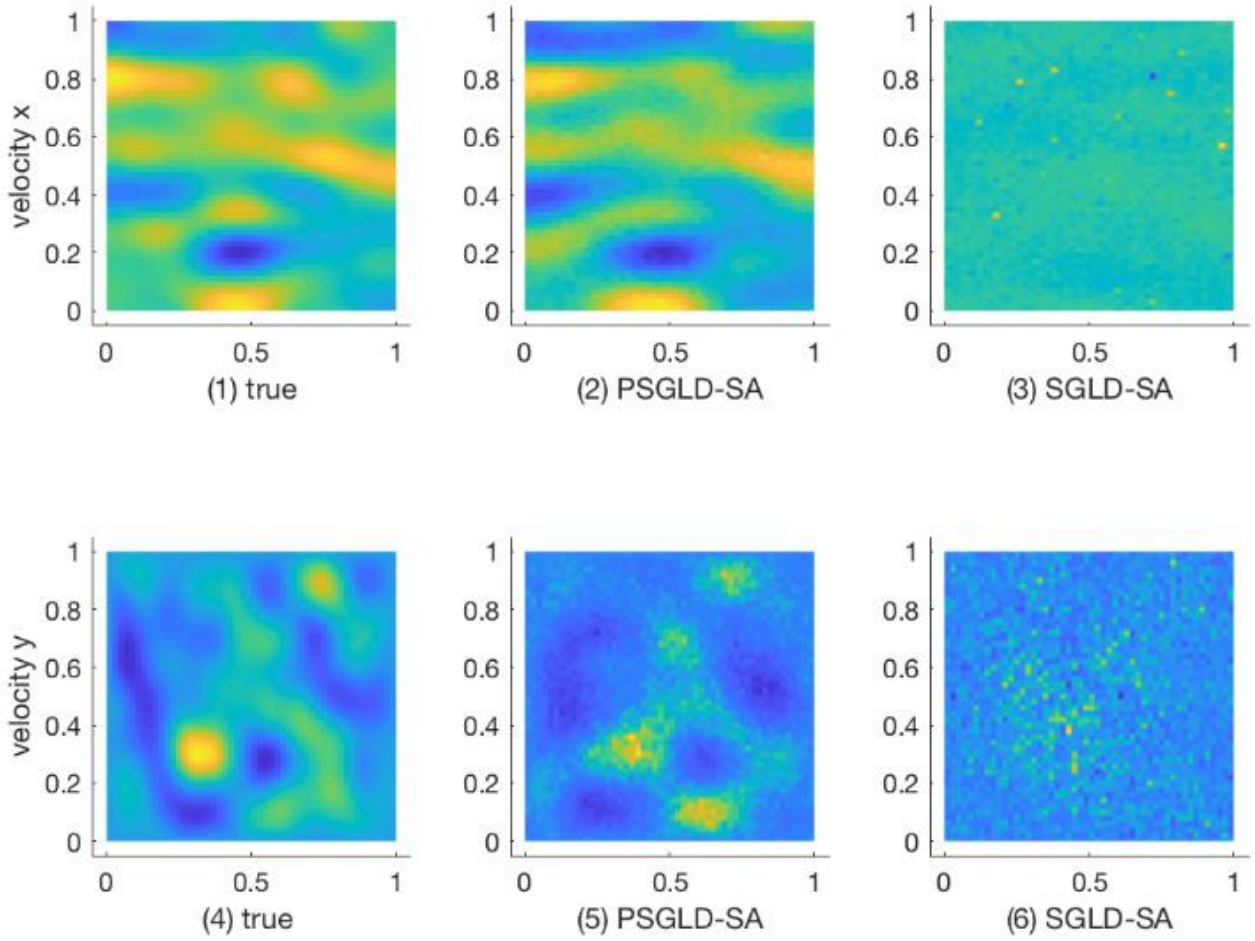
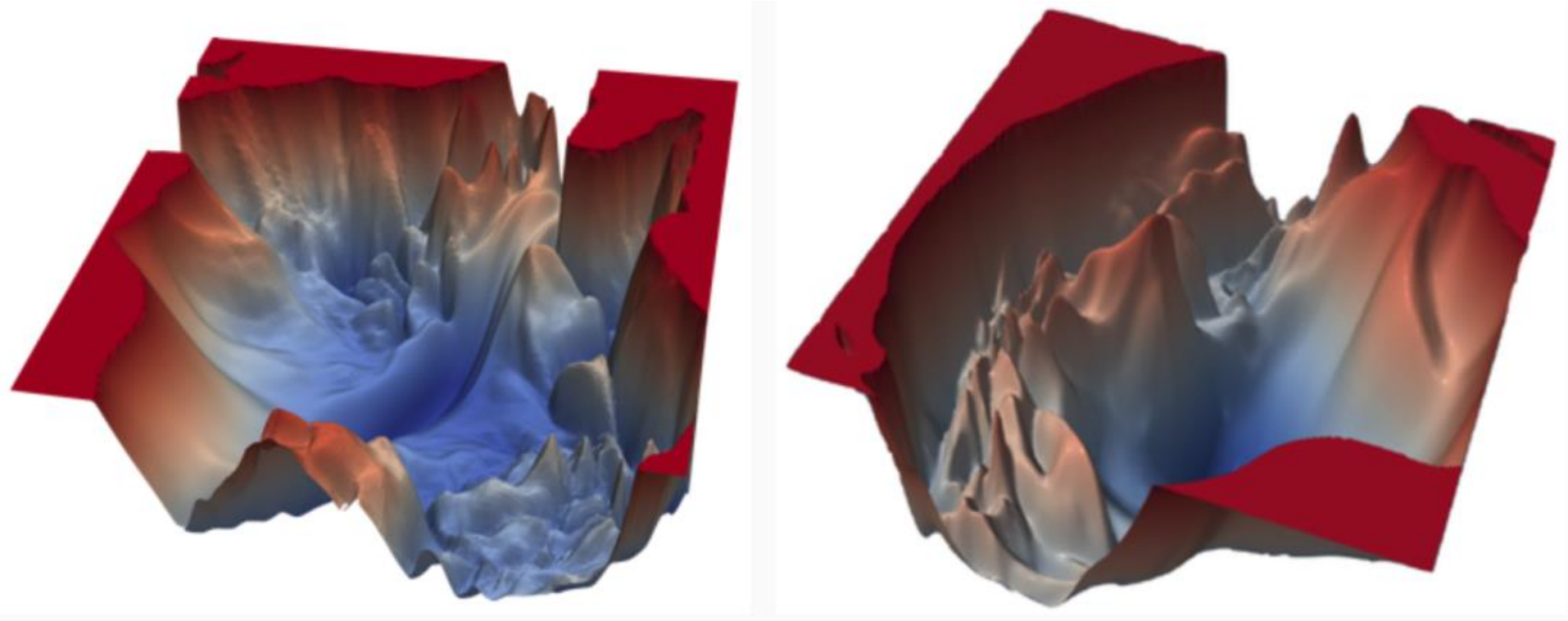


Figure 5: KLE 128. True and prediction solutions.

Summary

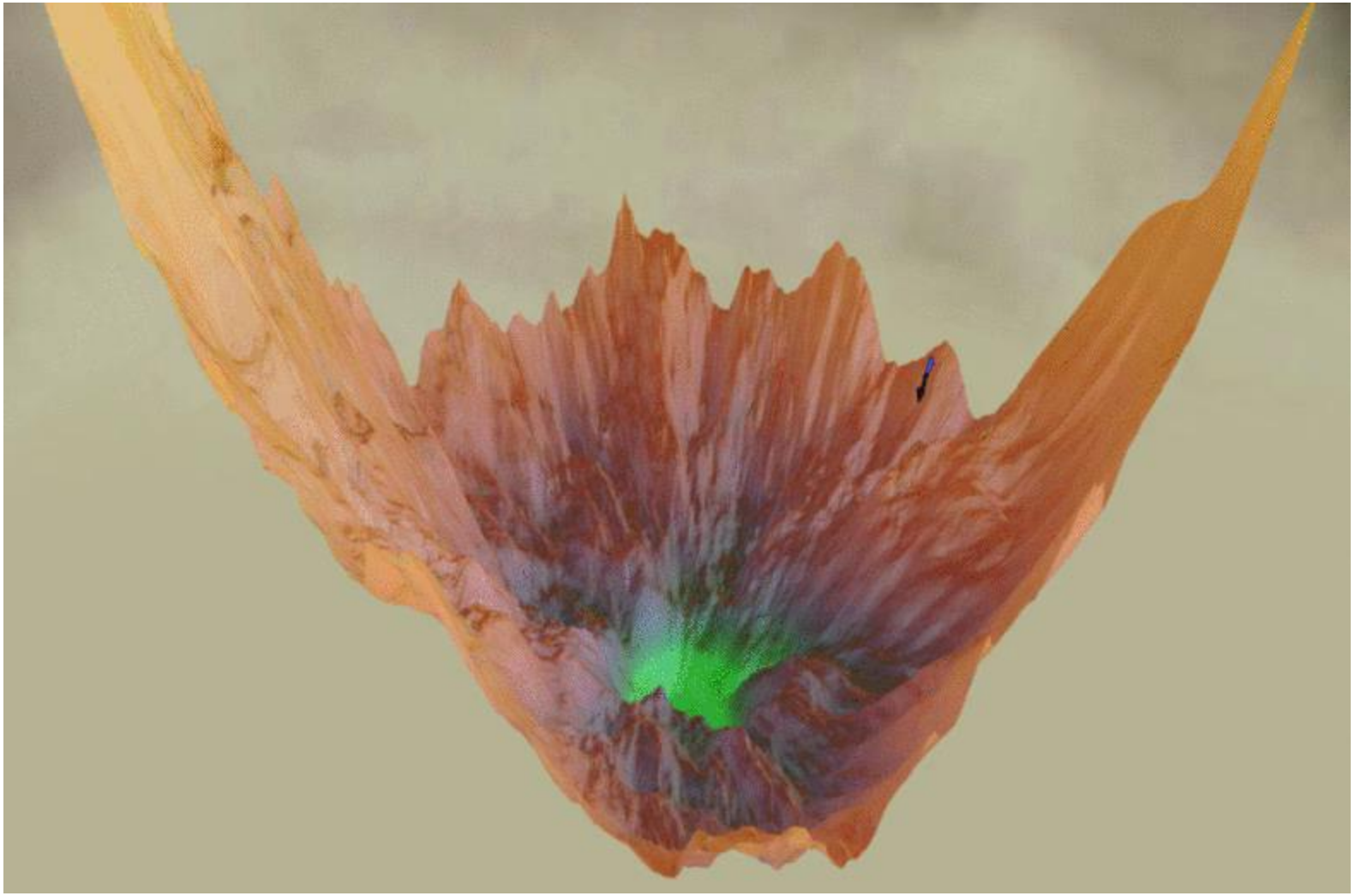
- Propose a class of adaptive stochastic gradient MCMC framework with wide applications and proved the convergence under mild assumptions.
- It can be applied to the empirical Bayesian method to learn a class of hierarchical Bayes models, yielding data-driven adaptive penalties.
- Propose an adaptive Bayesian sparse deep learning algorithm for regression problems
- Optimize SSGL priors through stochastic approximation. Sparsity is ensured, resulting in less usage in memory and computational power
- Sample with preconditioner SGLD, which is adaptive to local geometric and results in faster convergence
- Converge to the asymptotically correct distribution with a controllable bias introduced by SA
- It achieves the state-of-the-art compression performance on Resnet20, which outperforms the existing methods by a large margin.

Visualization the Loss Landscape of Deep Neural Nets



The loss landscape of modern deep neural nets [Li et al., 2018]

Gradient Descent Fails



Credit to losslanscape.com

Outline:

- Incorporate Physics Knowledge and AI to design new interpretable models
- Incorporate Physics Knowledge into AI to predict multiscale problems: NH-PINN
- Interpretable AI enables data-driven scientific discovery with uncertainty quantification capability – ALZHEIMER's Disease Prediction
- Sparse Neural Architecture Design with quantified uncertainties
- **Scalable training large-scale Deep Neural Network**

Scalable training large-scale Deep Neural Network:

Question: How can we design efficient optimization/sampling algorithms to train large-scale deep neural networks?

Goal: Enable Fast training large-scale DNN.

W. Deng, X. Zhang, F. Liang, **G. Lin**, An adaptive empirical Bayesian method for sparse deep learning, **2019 Conference on Neural Information Processing Systems (NIPS)**, Dec. 8 – Dec. 14, 2019, Vancouver, Canada.

NeurIPS'19, NeurIPS'20, ICML'20, ICLR'21, JCP'20, JCP'21a, JCP'21b

Scalable algorithms for Bayesian deep learning via Stochastic Gradient Monte Carlo and Beyond

Guang Lin [1]

Joint work with **W. Deng, Y. Wang, Q. Feng, L. Gao, G. Karagiannis, F. Liang**

August 13, 2021

[1]Departments of Mathematics & School of Mechanical Engineering, Purdue University

NeurIPS'19, NeurIPS'20, ICML'20, ICLR'21, JCP'20, JCP'21a, JCP'21b

Markov chain Monte Carlo

Uncertainty quantification is crucial for AI safety problems and reinforcement learning, which draws our attention to **Markov chain Monte Carlo (MCMC)**, which is known for

- Multi-modal sampling $\rightarrow$ Accurate predictive **confidence interval**
- Non-convex optimization $\rightarrow$ Better **point estimate**

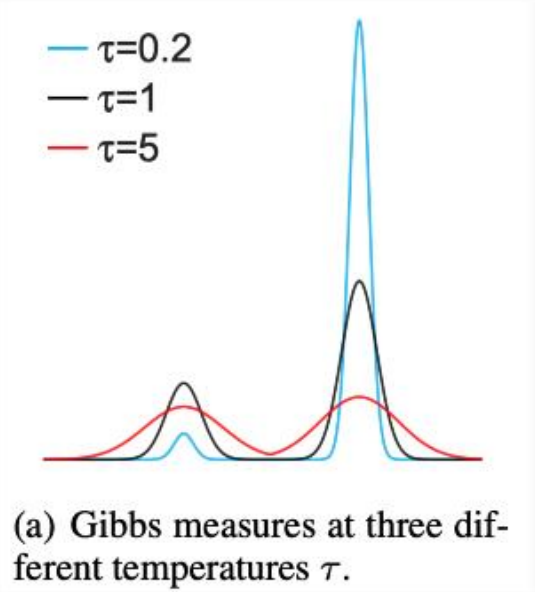
Langevin diffusion

A famous sampling algorithm is called Langevin diffusion.

$$d\boldsymbol{\beta}_t = -\nabla U(\boldsymbol{\beta}_t)dt + \sqrt{2\tau}d\boldsymbol{W}_t,$$

where $\boldsymbol{\beta}_t$ is the parameter at time t, $U(\cdot)$ is the energy function, $\boldsymbol{W}_t$ is a Brownian motion and τ is the temperature.

As $t \to \infty$, $\boldsymbol{\beta}_t$ converges to the stationary Gibbs distribution $Ce^{-\frac{U(\boldsymbol{\beta})}{\tau}}$.



(a) Gibbs measures at three different temperatures τ.

Stochastic gradient Langevin dynamics

However, evaluating gradient in big data problems is **too costly**.

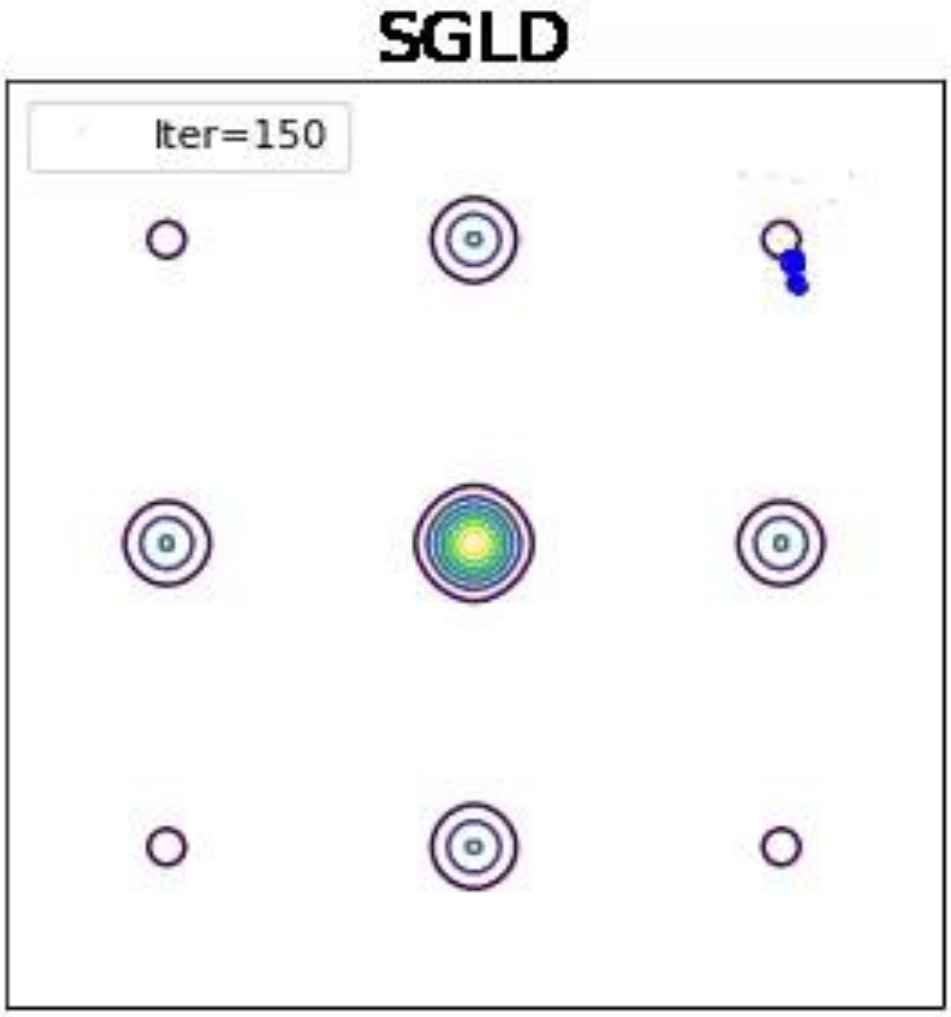
To tackle this issue, Max Welling, etc [Welling and Teh, 2011] proposed the stochastic gradient Langevin dynamics algorithm (SGLD)

$$\boldsymbol{\beta}_{k+1} = \boldsymbol{\beta}_k - \eta \nabla \widetilde{U}(\boldsymbol{\beta}_k) + \mathcal{N}(0, 2\eta\tau \boldsymbol{I}). \tag{1}$$

As $t \to \infty$ and $\eta \to 0$, $\boldsymbol{\beta}_t$ converges weakly to the stationary Gibbs distribution $Ce^{-\frac{U(\boldsymbol{\beta})}{\tau}}$.

Stochastic gradient Langevin dynamics

Sample from a multi-modal distribution



Acceleration strategies for MCMC

Most popular strategies to *accelerate* MCMC:

- Simulated annealing [Kirkpatrick et al., 1983]
- **Replica exchange MCMC [Swendsen and Wang, 1986]**

Replica Exchange SGLD
Wei Deng, et al., ICML 2020

Replica exchange Langevin diffusion

Consider two Langevin diffusion processes with $\tau_1 > \tau_2$

$$d\beta_t^{(1)} = -\nabla U(\beta_t^{(1)})dt + \sqrt{2\tau_1}d\boldsymbol{W}_t^{(1)}$$
$$d\beta_t^{(2)} = -\nabla U(\beta_t^{(2)})dt + \sqrt{2\tau_2}d\boldsymbol{W}_t^{(2)},$$

Moreover, the positions of the two particles swap with a probability

$$S(\beta_t^{(1)}, \beta_t^{(2)}) := e^{\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)\left(U(\beta_t^{(1)}) - U(\beta_t^{(2)})\right)}$$

In other words, a jump process is included in a Markov process

$$\mathbb{P}(\boldsymbol{\beta}_{t+dt} = (\beta_t^{(2)}, \beta_t^{(1)}) | \boldsymbol{\beta}_t = (\beta_t^{(1)}, \beta_t^{(2)})) = rS(\beta_t^{(1)}, \beta_t^{(2)})dt$$
$$\mathbb{P}(\boldsymbol{\beta}_{t+dt} = (\beta_t^{(1)}, \beta_t^{(2)}) | \boldsymbol{\beta}_t = (\beta_t^{(1)}, \beta_t^{(2)})) = 1 - rS(\beta_t^{(1)}, \beta_t^{(2)})dt$$

A demo

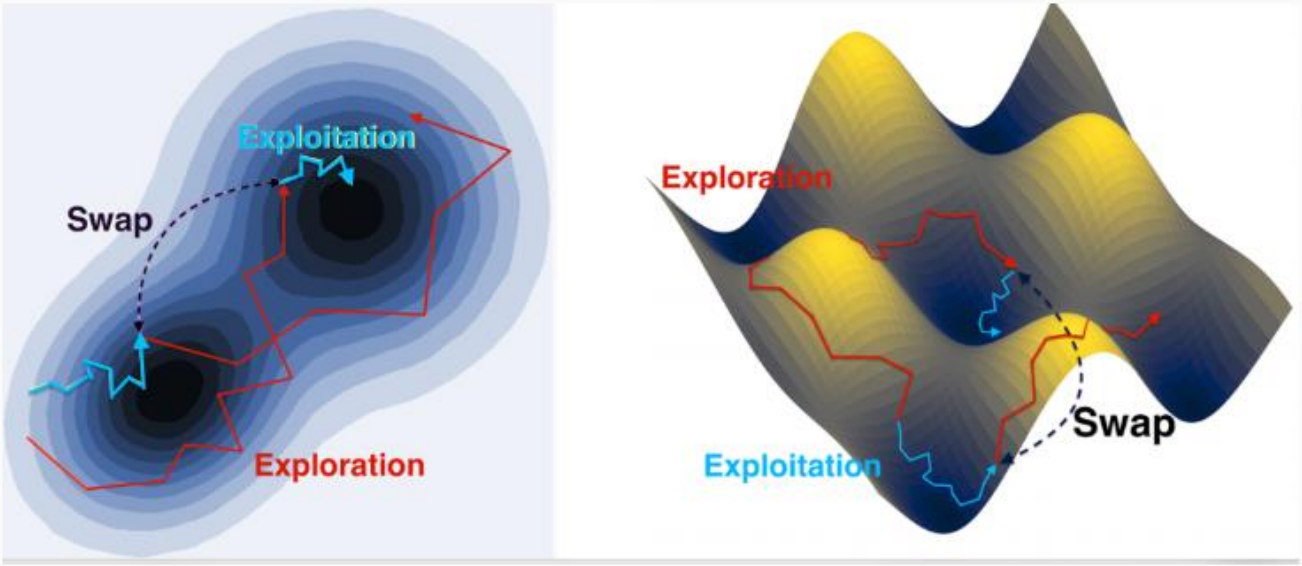


Figure 1: Trajectory plot for replica exchange Langevin diffusion.

Why the naïve numerical algorithm fails

Consider the scalable stochastic gradient Langevin dynamics algorithm [Welling and Teh, 2011]

$$\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)} = \widetilde{\boldsymbol{\beta}}_{k}^{(1)} - \eta_k \nabla \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k}^{(1)}) + \sqrt{2\eta_k \tau_1}\boldsymbol{\xi}_k^{(1)}$$
$$\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)} = \widetilde{\boldsymbol{\beta}}_{k}^{(2)} - \eta_k \nabla \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k}^{(2)}) + \sqrt{2\eta_k \tau_2}\boldsymbol{\xi}_k^{(2)}.$$

Swap the chains with a **naïve** swapping rate $r\mathbb{S}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}, \widetilde{\boldsymbol{\beta}}_{k+1}^{(2)})\eta_k$[§]:

$$\mathbb{S}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}, \widetilde{\boldsymbol{\beta}}_{k+1}^{(2)}) = e^{\left(\frac{1}{\tau_1} - \frac{1}{\tau_2}\right)\left(\widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(1)}) - \widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(2)})\right)}. \tag{2}$$

Exponentiating the unbiased estimators $\widetilde{L}(\widetilde{\boldsymbol{\beta}}_{k+1}^{(\cdot)})$ leads to a **large bias**.

[§]In the implementations, we fix $r\eta_k = 1$ by default.

A corrected algorithm

Assume $\widetilde{L}(\theta) \sim \mathcal{N}(L(\theta), \sigma^2)$ and consider the **geometric Brownian motion** of $\{\widetilde{S}_t\}_{t\in[0,1]}$ in each swap as a Martingale

$$\begin{aligned}\widetilde{S}_t &= e^{\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\left(\widetilde{L}(\widetilde{\beta}^{(1)})-\widetilde{L}(\widetilde{\beta}^{(2)})-\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\sigma^2 t\right)} \\ &= e^{\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\left(L(\widetilde{\beta}^{(1)})-L(\widetilde{\beta}^{(2)})-\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\sigma^2 t+\sqrt{2}\sigma W_t\right)}.\end{aligned} \tag{3}$$

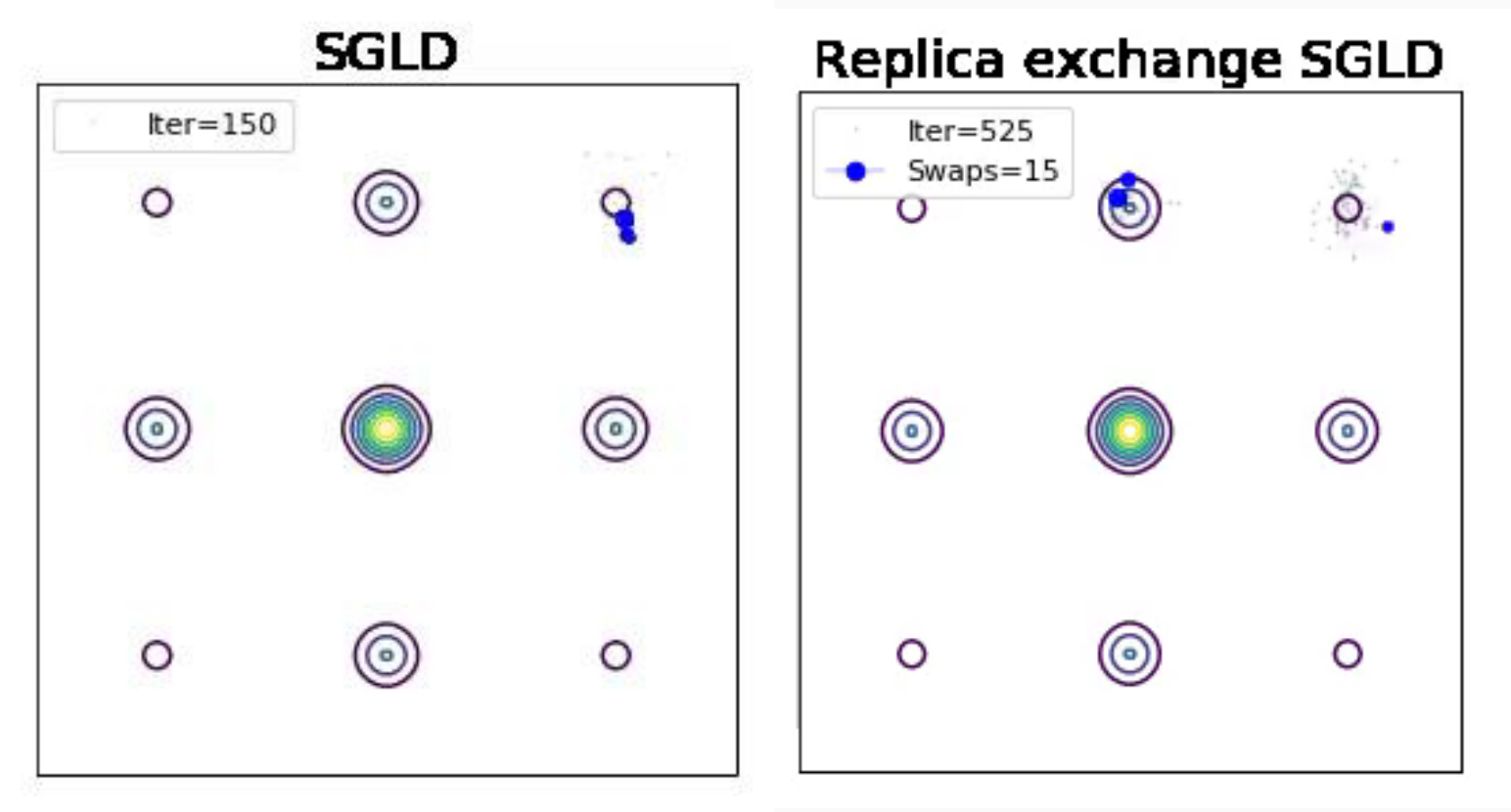
Taking the derivative of $\widetilde{S}_t$ with respect to t and W_t, Itô's lemma gives,

$$d\widetilde{S}_t = \left(\frac{d\widetilde{S}_t}{dt} + \frac{1}{2}\frac{d^2\widetilde{S}_t}{dW_t^2}\right)dt + \frac{d\widetilde{S}_t}{dW_t}dW_t = \sqrt{2}\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\sigma\widetilde{S}_t dW_t.$$

By fixing $t = 1$ in (3), we have the **suggested unbiased swapping rate**

$$\widetilde{S}_1 = e^{\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\left(\widetilde{L}(\widetilde{\beta}^{(1)})-\widetilde{L}(\widetilde{\beta}^{(2)})-\left(\frac{1}{\tau_1}-\frac{1}{\tau_2}\right)\sigma^2\right)}.$$

Replica exchange Stochastic gradient Langevin dynamics



Acceleration via replica exchange

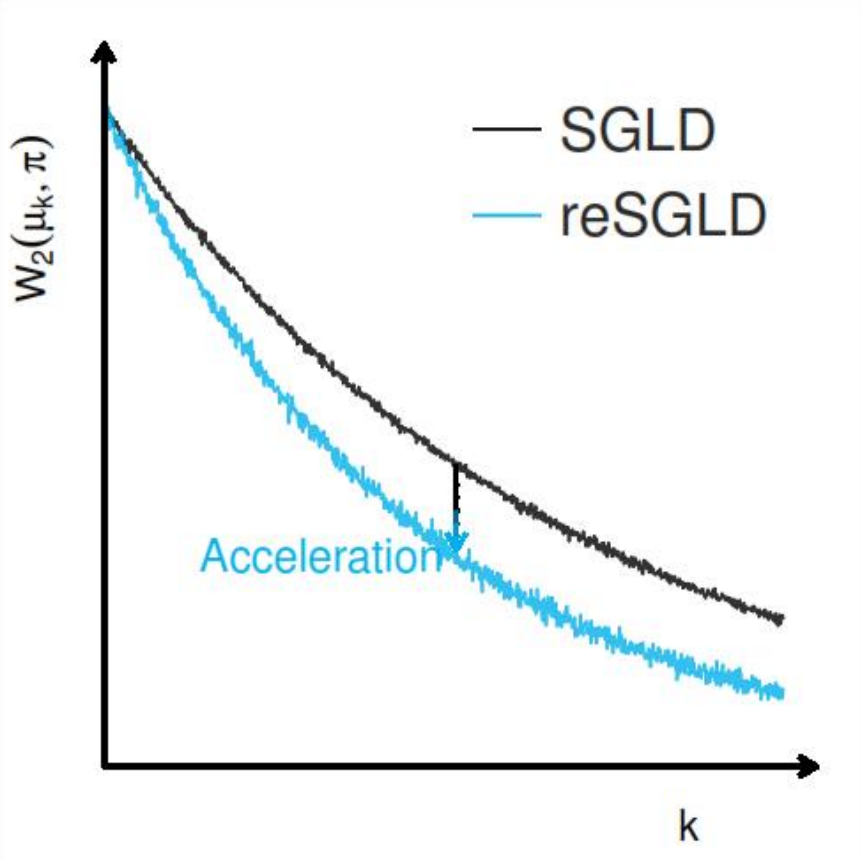


Figure 2: Acceleration via replica exchange (swaps/ interactions)

Accelerating convergence via variance reduction

Can we do better?

Exponential acceleration via variance reduction
Wei Deng et al., ICLR 2021

Accelerating convergence via variance reduction

The desire to obtain more effective swaps drives us to design more efficient energy estimators.

To reduce the variance of the noisy energy estimator $L(B|\beta^{(h)}) = \frac{N}{n}\sum_{i\in B} L(\boldsymbol{x}_i|\beta^{(h)})$ for $h \in \{1,2\}$, we consider an unbiased estimator $L(B|\widehat{\beta}^{(h)})$ for $\sum_{i=1}^{N} L(\boldsymbol{x}_i|\widehat{\beta}^{(h)})$ and a constant c, we see that a new estimator $\widetilde{L}(B|\beta^{(h)})$, which follows

$$\widetilde{L}(B|\beta^{(h)}) = L(B|\beta^{(h)}) + c\left(L(B|\widehat{\beta}^{(h)}) - \sum_{i=1}^{N} L(\boldsymbol{x}_i|\widehat{\beta}^{(h)})\right), \qquad (4)$$

is still the unbiased estimator for $\sum_{i=1}^{N} L(\boldsymbol{x}_i|\beta^{(h)})$.

Accelerating convergence via variance reduction

By decomposing the variance, we have

$$Var(\widetilde{L}(B|\beta^{(h)})) = Var\left(L(B|\beta^{(h)})\right) + c^2 Var\left(L(B|\widehat{\beta}^{(h)})\right) + 2c\text{Cov}\left(L(B|\beta^{(h)}), L(B|\widehat{\beta}^{(h)})\right).$$

In such a case, $Var(\widetilde{L}(B|\beta^{(h)}))$ achieves the minimum variance $(1-\rho^2)Var(L(B|\beta^{(h)}))$ given $c^\star := \frac{-\text{Cov}(L(B|\beta^{(h)}), L(B|\widehat{\beta}^{(h)}))}{Var(L(B|\beta^{(h)}))}$, where $\text{Cov}(\cdot,\cdot)$ denotes the covariance and ρ is the correlation coefficient.

Accelerating convergence via variance reduction

To make variance reduction work, it requires two crucial components.

- To propose a correlated control variate $\widehat{\beta}$
  $\rightarrow$ Update $\widehat{\beta}^{(h)} = \beta^{(h)}_{m\lfloor \frac{k}{m} \rfloor}$ every m iterations
- The optimal c is unknown.
  $\rightarrow$ Set $c = -1$ for highly correlated energy estimators.
  $\rightarrow$ Set adaptive c for the less correlated.

Reduction of Variance

VR-reSGLD may lead to a more efficient energy estimator with a much smaller variance.

Lemma (Variance-reduced energy estimator)

Under the smoothness and dissipativity assumptions, the variance of the variance-reduced energy estimator $\widetilde{L}(B|\beta^{(h)})$, *where* $h \in \{1, 2\}$, *is upper bounded by*

$$Var\left(\widetilde{L}(B|\beta^{(h)})\right) \leq \min\left\{\mathcal{O}\left(\frac{m^2\eta}{n}\right), Var\left(\frac{N}{n}\sum_{i\in B} L(\boldsymbol{x}_i|\beta^{(1)})\right)\right\},$$

where the detailed $\mathcal{O}(\cdot)$ *constants is shown in the appendix [Deng et al., 2021].*

A smaller variance implies more effective swaps

The variance-reduced energy estimator $\widetilde{L}(B|\boldsymbol{\beta}^{(h)})$ doesn't directly affect $\mathbb{E}[\widetilde{S}_{\eta,m,n}]$ within the support $[0,\infty]$. However, the unbounded support is not appropriate for numerical algorithms, and only the truncated swapping rate $S_{\eta,m,n} = \min\{1, \widetilde{S}_{\eta,m,n}\}$ is considered. As such, the truncated swapping rate becomes significantly smaller.

Lemma (Variance reduction for larger swapping rates)

Given a large enough batch size n, the variance-reduced energy estimator $\widetilde{L}(B_k|\boldsymbol{\beta}_k^{(h)})$ *yields a truncated swapping rate that satisfies*

$$\mathbb{E}[S_{\eta,m,n}] \approx \min\left\{1, S(\boldsymbol{\beta}^{(1)}, \boldsymbol{\beta}^{(2)})\left(\mathcal{O}\left(\frac{1}{n^2}\right) + e^{-\mathcal{O}\left(\frac{m^2\eta}{n}+\frac{1}{n^2}\right)}\right)\right\}. \quad (5)$$

Acceleration via variance-reduced replica exchange

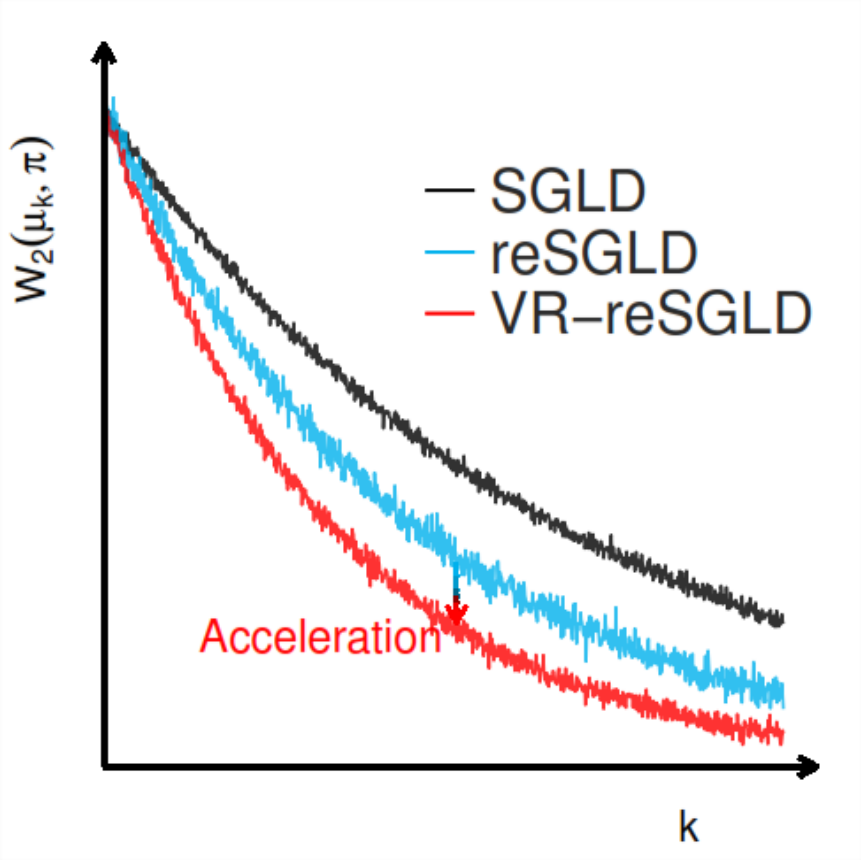
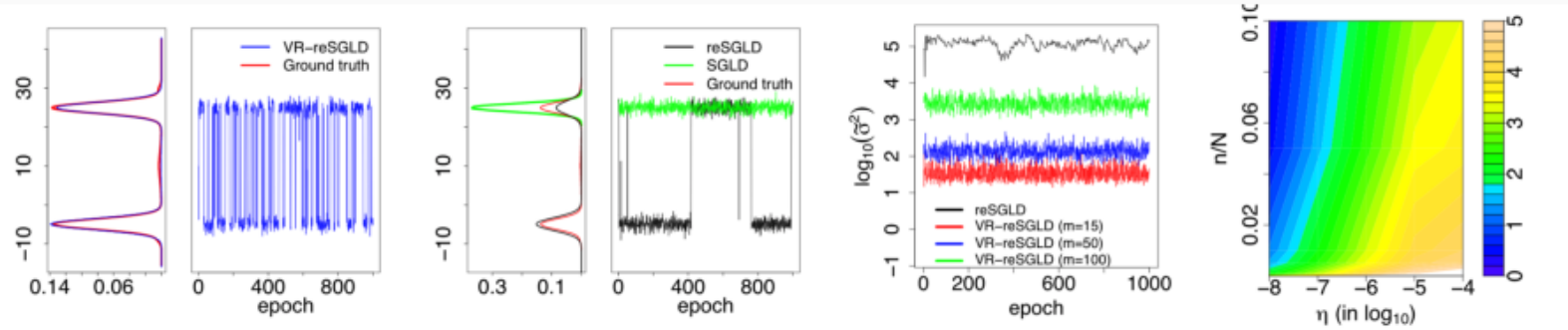


Figure 3: Acceleration via variance-reduced replica exchange.

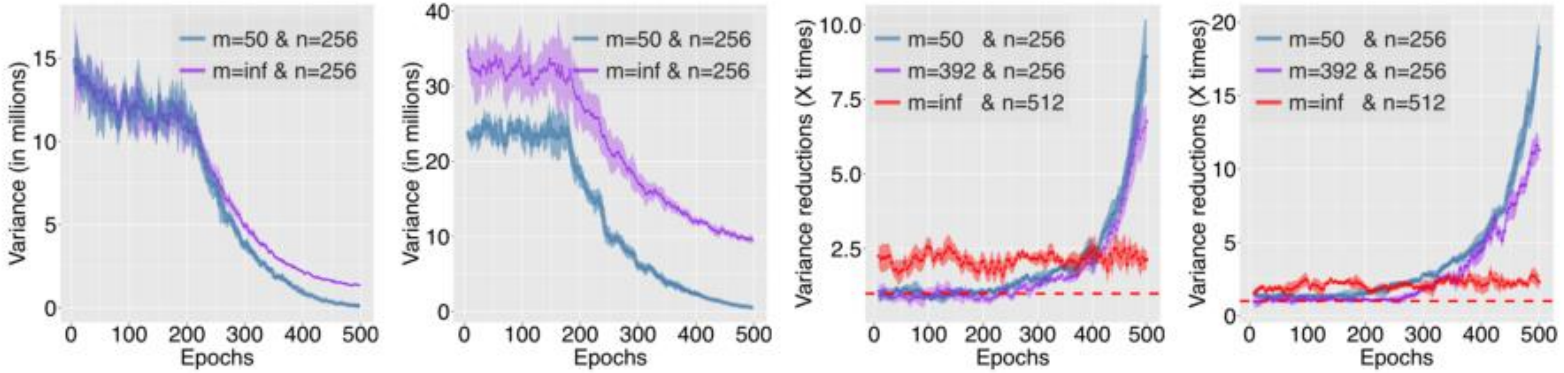
1D simulation of Gaussian mixture



(a) Trace plot for $\beta^{(1)}$ (b) Trace plot for $\beta^{(1)}$ (c) Paths of $\log_{10} \widetilde{\sigma}^2$ (d) Contour of $\log_{10} \widetilde{\sigma}^2$

Figure 2: Trace plots, KDEs of $\beta^{(1)}$, and sensitivity study of $\widetilde{\sigma}^2$ with respect to m, η and n.

Non-convex optimization on CIFAR10 and CIFAR100



(a) CIFAR10: Original v.s. proposed (m=50)

(b) CIFAR100: Original v.s. proposed (m=50)

(c) Variance reduction setups on CIFAR10

(d) Variance reduction setups on CIFAR100

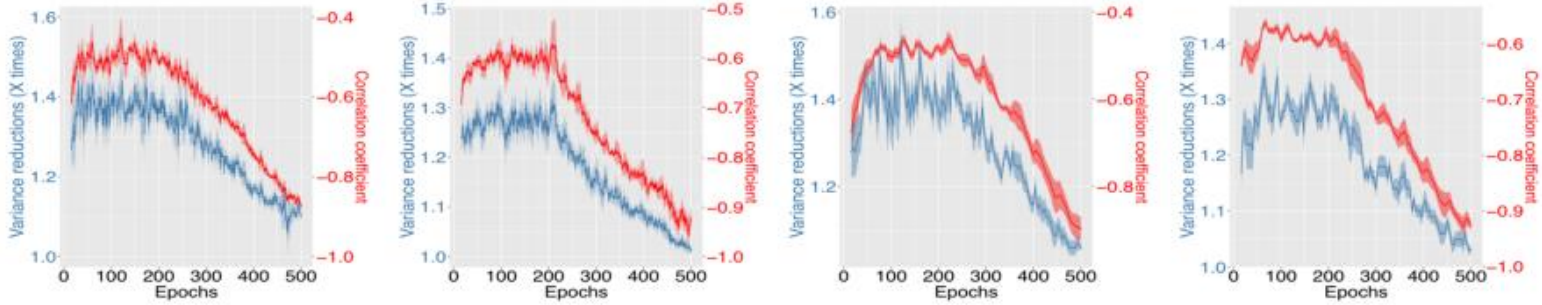
Figure 3: Variance reduction on the noisy energy estimators on CIFAR10 & CIFAR100 datasets.

Non-convex optimization on CIFAR10 and CIFAR100

TABLE 1: PREDICTION ACCURACIES (%) BASED ON BAYESIAN MODEL AVERAGING.

METHOD	CIFAR10			CIFAR100		
	RESNET20	RESNET32	RESNET56	RESNET20	RESNET32	RESNET56
M-SGD	94.07±0.11	95.11±0.07	96.05±0.21	71.93±0.13	74.65±0.20	78.76±0.24
SGHMC	94.16±0.13	95.17±0.08	96.04±0.18	72.09±0.14	74.80±0.19	78.95±0.22
reSGHMC	94.56±0.23	95.44±0.16	96.15±0.17	73.94±0.34	76.38±0.23	79.86±0.26
VR-reSGHMC	**94.84±0.11**	**95.62±0.09**	**96.32±0.15**	**74.83±0.18**	**77.40±0.27**	**80.62±0.22**
cycSGHMC	94.61±0.15	95.56±0.12	96.19±0.17	74.21±0.22	76.60±0.25	80.39±0.21
cVR-reSGHMC	**94.91±0.10**	**95.64±0.13**	**96.36±0.16**	**75.02±0.19**	**77.58±0.21**	**80.50±0.25**

Non-convex optimization on CIFAR10 and CIFAR100



(a) CIFAR10 & m=50 (b) CIFAR100 & m=50 (c) CIFAR10 & m=392 (d) CIFAR100 & m=392

Figure 5: A study of variance reduction techniques using adaptive coefficient and non-adaptive coefficient on CIFAR10 & CIFAR100 datasets.

Summary

- Replica exchange stochastic gradient MCMC shows a potential in exponentially accelerating the convergence in non-convex learning. [Deng et al., 2020]
- Variance reduction of energy estimators yields exponential more effective swaps, which further accelerates the exponential convergence in non-convex learning. [Deng et al., 2021]
- This is the first work to do variance reduction on energy estimators in deep learning, which paves the road for accelerating advanced stochastic gradient MCMC algorithms in non-convex learning.

References i

Deng, W., Feng, Q., Gao, L., Liang, F., and Lin, G. (2020).
Non-Convex Learning via Replica Exchange Stochastic Gradient MCMC.
In *Proc. of the International Conference on Machine Learning (ICML)*.

Deng, W., Feng, Q., Karagiannis, G., Lin, G., and Liang, F. (2021).
Accelerating Convergence of Replica Exchange Stochastic Gradient MCMC via Variance Reduction.
In *Proc. of the International Conference on Learning Representation (ICLR)*.

Kirkpatrick, S., Jr, D. G., and Vecchi, M. P. (1983).
Optimization by Simulated Annealing.
Science, 220(4598):671–680.

Towards Third Wave AI: Interpretable, Robust Trustworthy Machine Learning for Diverse Applications in Science and Engineering

"…Because I had worked in the closest possible ways with physicists and engineers, I knew that our data can never be precise…"

Norbert Wiener