Sandia UQ Seminar

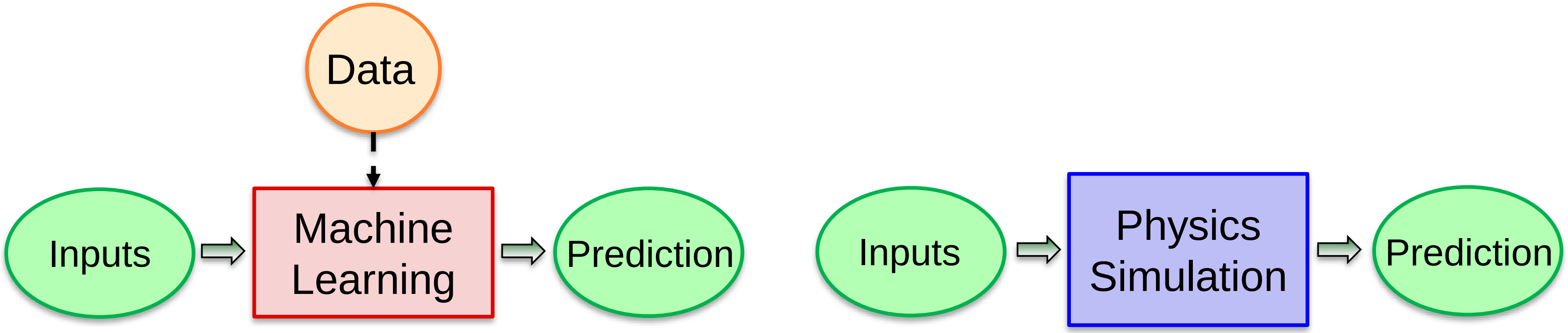
Fusing Physics and Machine Learning in Uncertainty Quantification

Geoffrey Bomarito
NASA Langley Research Center

P Leser, J Warner, W Leser, L Morrill
S Nieomoeller, J Cuevas, T Lewitt, S Lai

Motivation

Big decisions *need more than big data...*
they ***need big models*** *too**



- Efficient predictions
- Fast deployment
- Widely accessible from open-source libraries

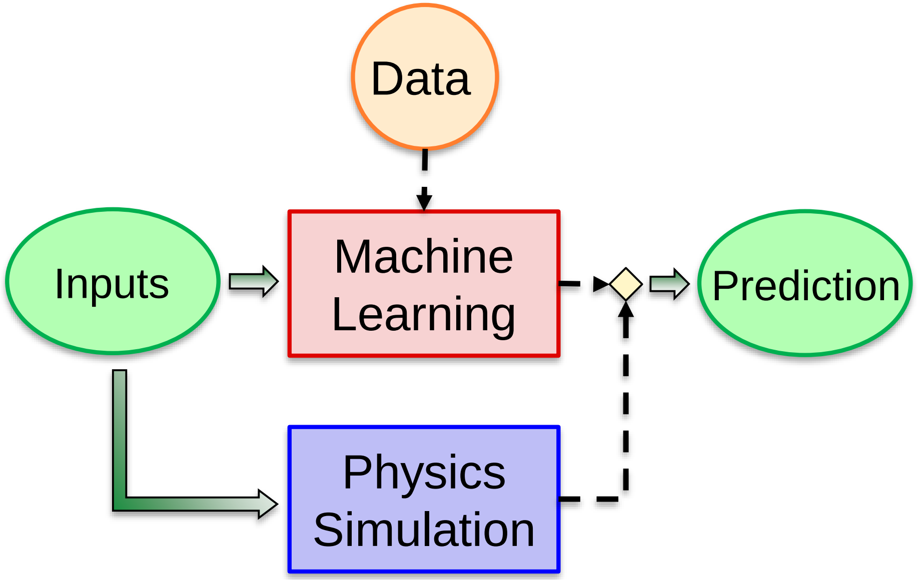
- Interpretable predictions
- Theoretically rigorous
- Historically-proven legacy codes

Scientific Machine Learning (ML)

*Karen Willcox, Supercomputing 2019

Scientific ML: Two Approaches

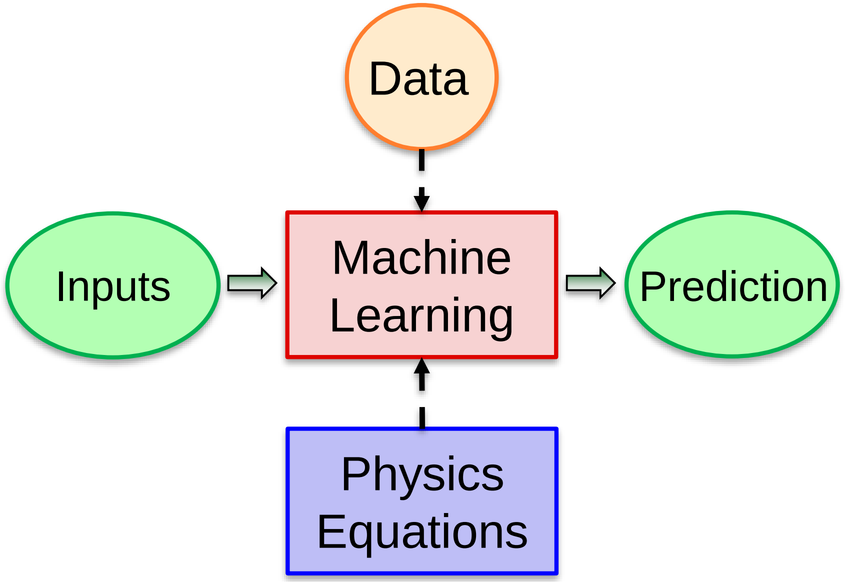
1. ***Fuse*** predictions from ML and physics-based models



- **Method:** Multi-model Monte Carlo simulation [1,2,3]

1. Multilevel Monte Carlo. Giles et al. 2015
2. Multifidelity Monte Carlo. Peherstorfer et al. 2016
3. Approximate Control Variates. Gorodetsky et al. 2019

2. ***Integrate*** physical laws into the training of ML models

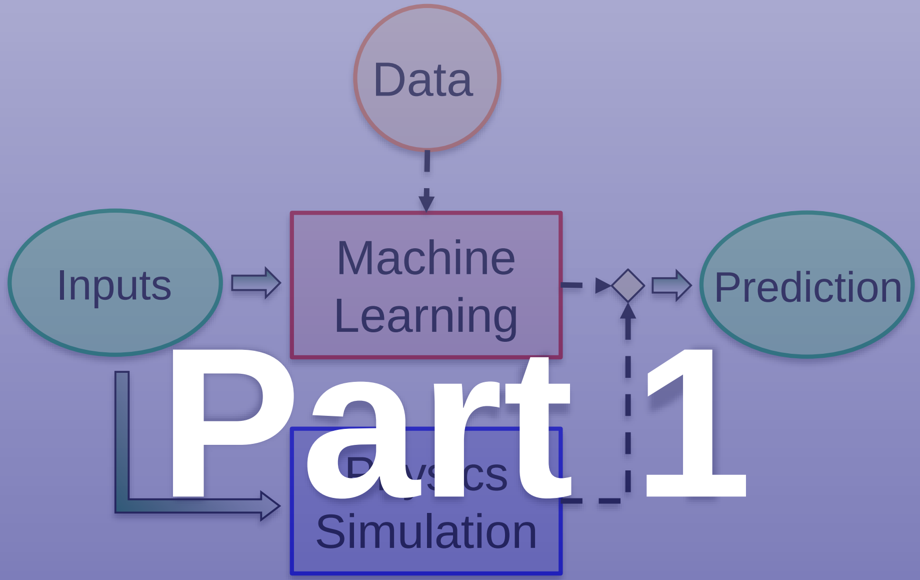


- **Method:** Physics-informed deep learning [4,5]

4. Physics-informed neural networks. Raissi et al. 2019.
5. Physics-informed generative adversarial networks Yang et al. 2018

Scientific ML: Two Approaches

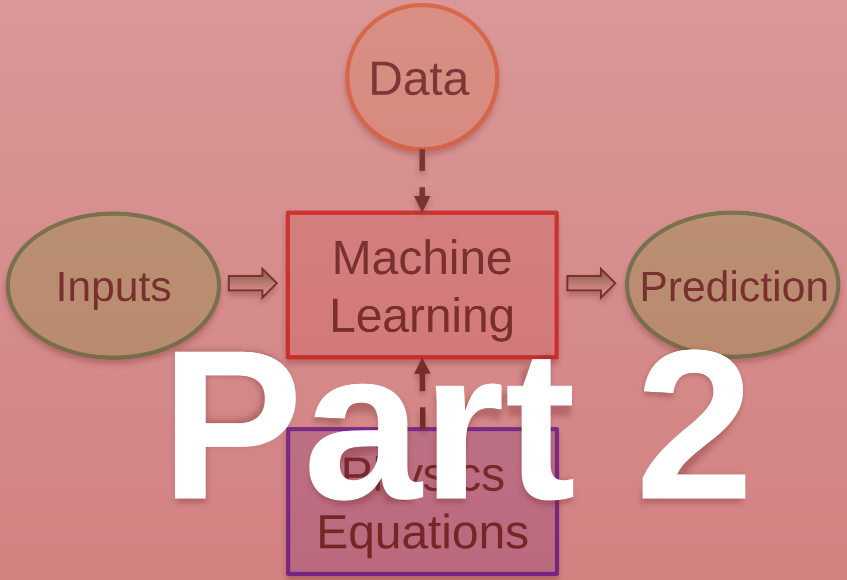
1. ***Fuse*** predictions from ML and physics-based models



- **Method:** Multi-model Monte Carlo simulation [1,2,3]

1. Multilevel Monte Carlo. Giles et al. 2015
2. Multifidelity Monte Carlo. Peherstorfer et al. 2016
3. Approximate Control Variates. Gorodetsky et al. 2019

2. ***Integrate*** physical laws into the training of ML models



- **Method:** Physics-informed deep learning [4,5]

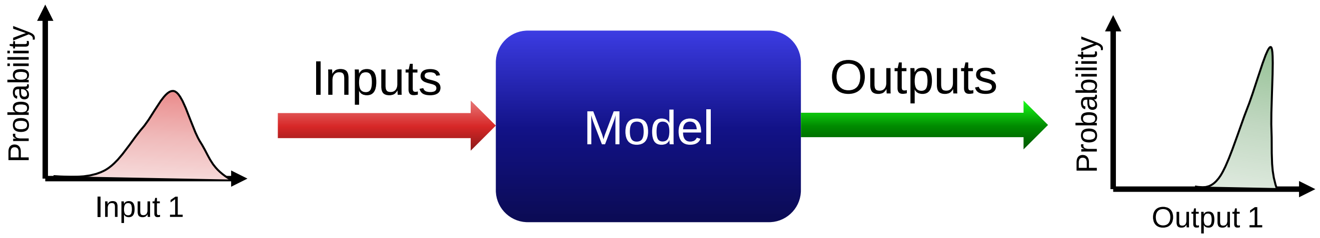
4. Physics-informed neural networks. Raissi et al. 2019.
5. Physics-informed generative adversarial networks Yang et. al. 2018

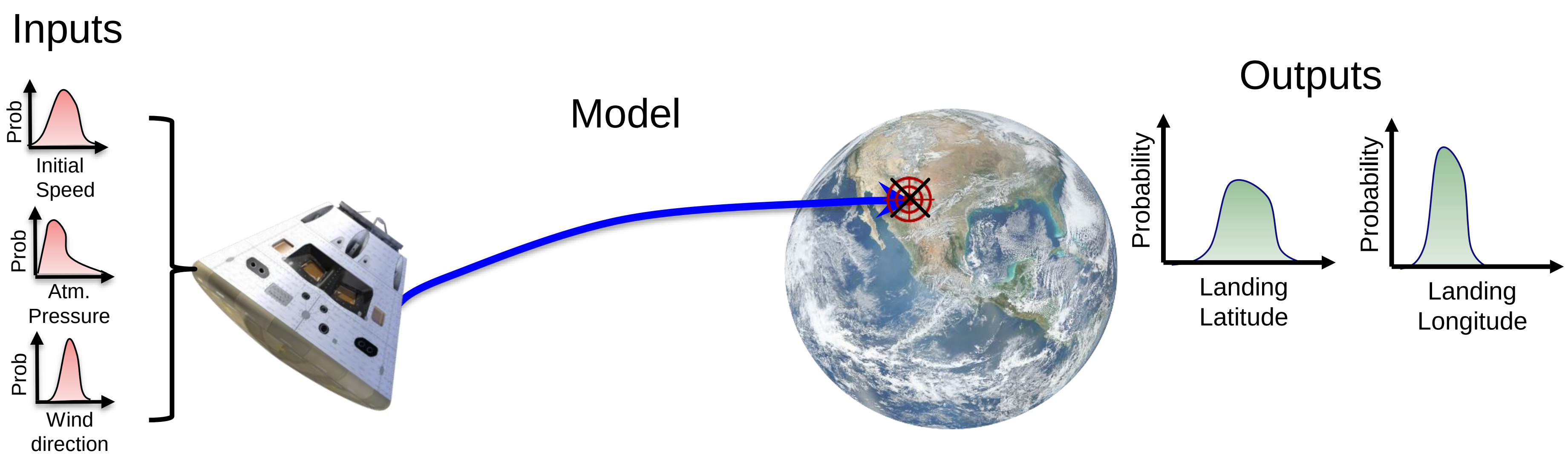
Part 1: Model Fusion

Parametrically-defined Approximate Control Variates

Uncertainty Propagation

How does uncertainty in model inputs affect predictions of output?





Monte Carlo Sampling

1) Draw N random inputs

2) Evaluate model

3) Form estimator

Unbiased: $\boldsymbol{E}[\hat{Q}_0] = \boldsymbol{E}[Q_0]$

$i = 1, \ldots, N$

$z^{(i)}$

Model
$Q_0 = \boldsymbol{M}(z^{(i)})$

$$\hat{Q}_0 = \frac{1}{N}\sum_i Q_0(z^{(i)})$$

Solution Time: $N \times C$

Model cost

How can we improve calculation time?

$z^{(i)}$

ML Surrogate
$Q_1 = \widetilde{\boldsymbol{M}}\,(z^{(i)})$

$$\hat{Q}_1 = \frac{1}{N}\sum_i Q_1(z^{(i)})$$

Solution Time: $N \times \tilde{C}$

Surrogate cost

Biased: $\boldsymbol{E}[\hat{Q}_1] \neq \boldsymbol{E}[Q_0]$

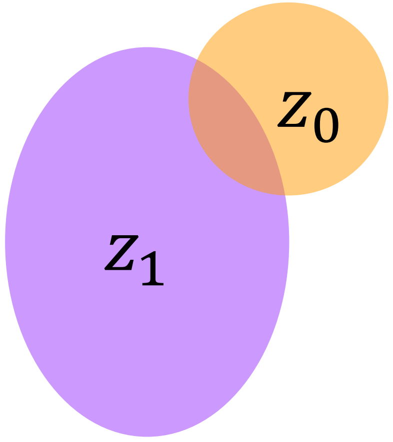
Surrogate cost << Model cost

Mean Squared Error = bias + variance

bias → irreducible error!

Multi-model Monte Carlo

1) Draw random inputs for multiple models



$z_0^{(i)}$

$z_1^{(i)}$

2) Evaluate model

Hifi Model

$Q_0 = \boldsymbol{M}(z^{(i)})$

Lofi Model

$Q_1 = \widetilde{\boldsymbol{M}}\,(z^{(i)})$

3) Form estimator

$$\boldsymbol{E}[Q_0] \approx f(Q_0, Q_1, z_0, z_1)$$

Constructed to be **Unbiased** and **Fast!**

Problem Statement

Given:

- A high fidelity model Q_0 with computational cost w_0
- M lower-fidelity models Q_i with costs w_i
- The covariance among models
- A time budget $\hat{w}$

Find:

- The best estimate of the mean of the high fidelity model $\boldsymbol{E}[Q_0]$

Approximate Control Variates (ACV)[1]

Monte Carlo estimators

$$\tilde{Q} = \hat{Q}_0(z_0) + \sum_{i=1}^{M} \alpha_i \left(\hat{Q}_i(z_i^*) - \hat{Q}_i(z_i)\right)$$

subsets of input samples

control variate weights

- Multilevel Monte Carlo (**MLMC**)[2] and Multifidelity Monte Carlo (**MFMC**)[3] are instances of this estimator
- New ACV estimators[1] based on independent sampling (**ACVIS**), multifidelity sampling (**ACVMF**), and 2-valued parameterization (**ACVKL**)
- Estimator is unbiased (wrt $E[Q_0]$)
- Variance (error) controlled by size and intersections of z_0, z_i^*, z_i
- Total estimator cost also controlled by size and intersections of z_0, z_i^*, z_i

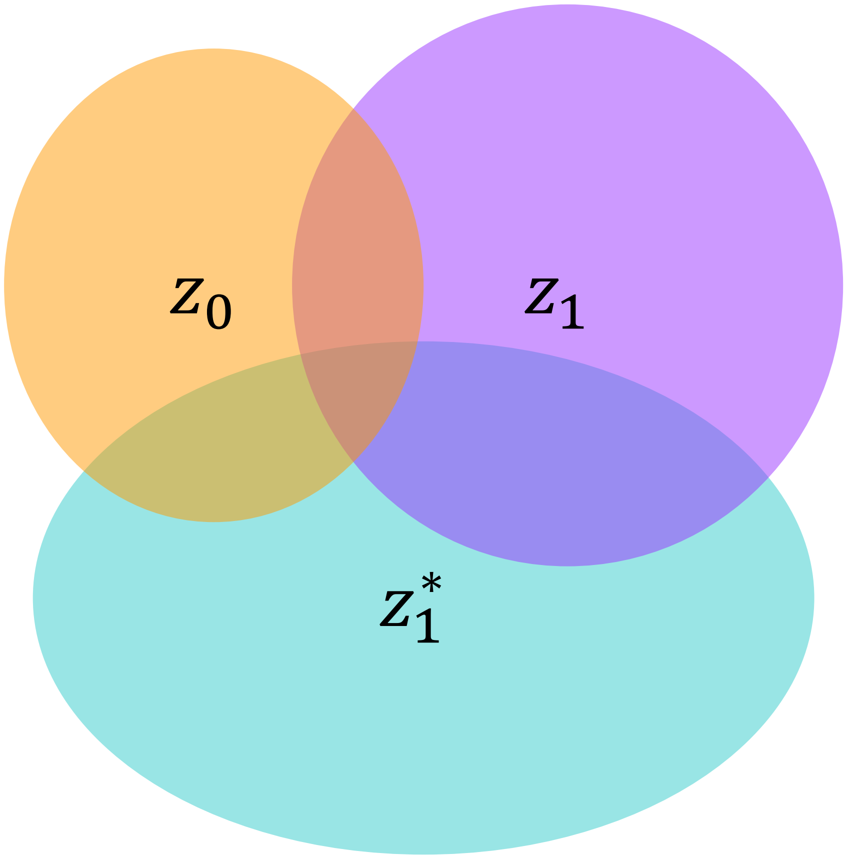
Goal: Sample allocation optimization

Find z_0, z_i^*, z_i such that variance is minimal and cost is under budget

[1] Gorodetsky, A A., et al. Journal of Computational Physics (2020)
[2] Giles, M B. Operations Research (2008)
[3] Peherstorfer, B, et al. SIAM Journal on Scientific Computing (2016)

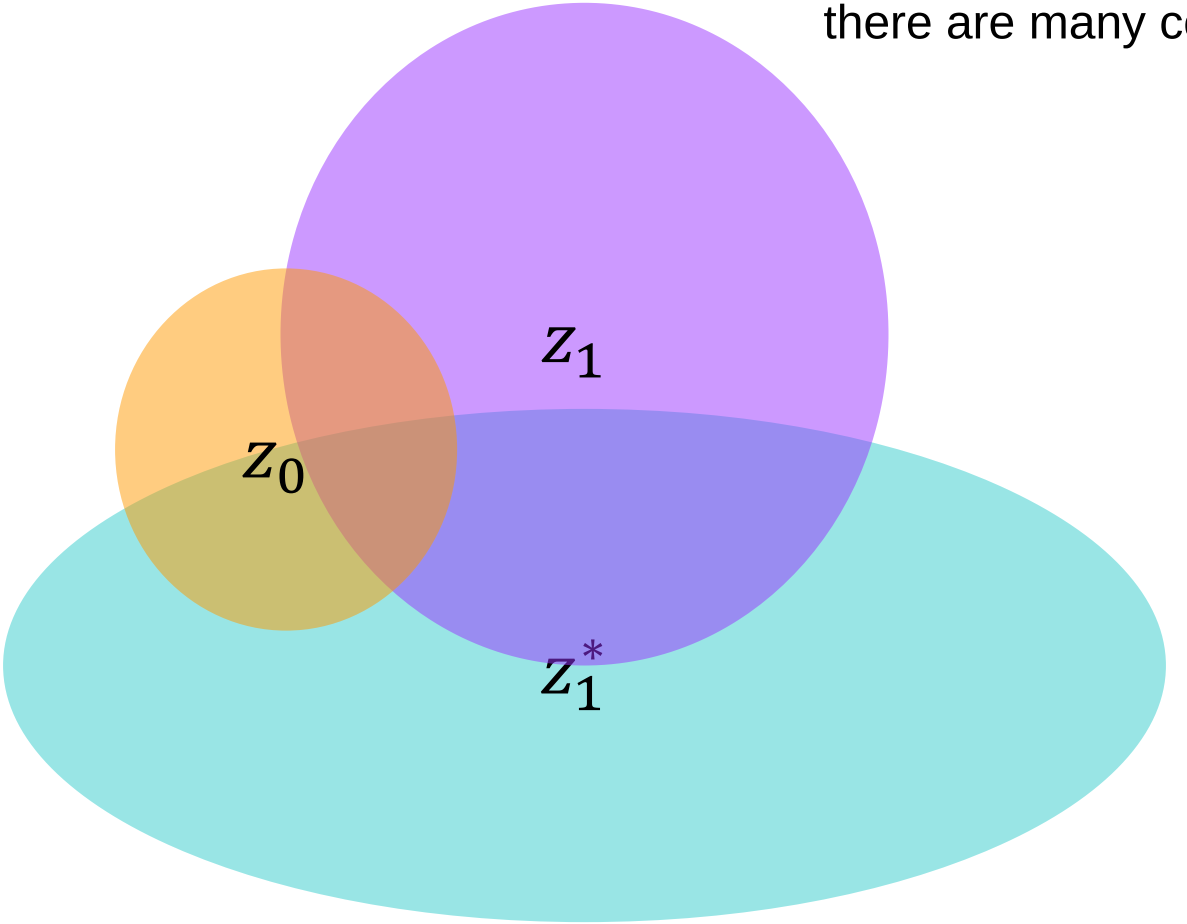
Sample Allocation Optimization

must optimize size and overlap



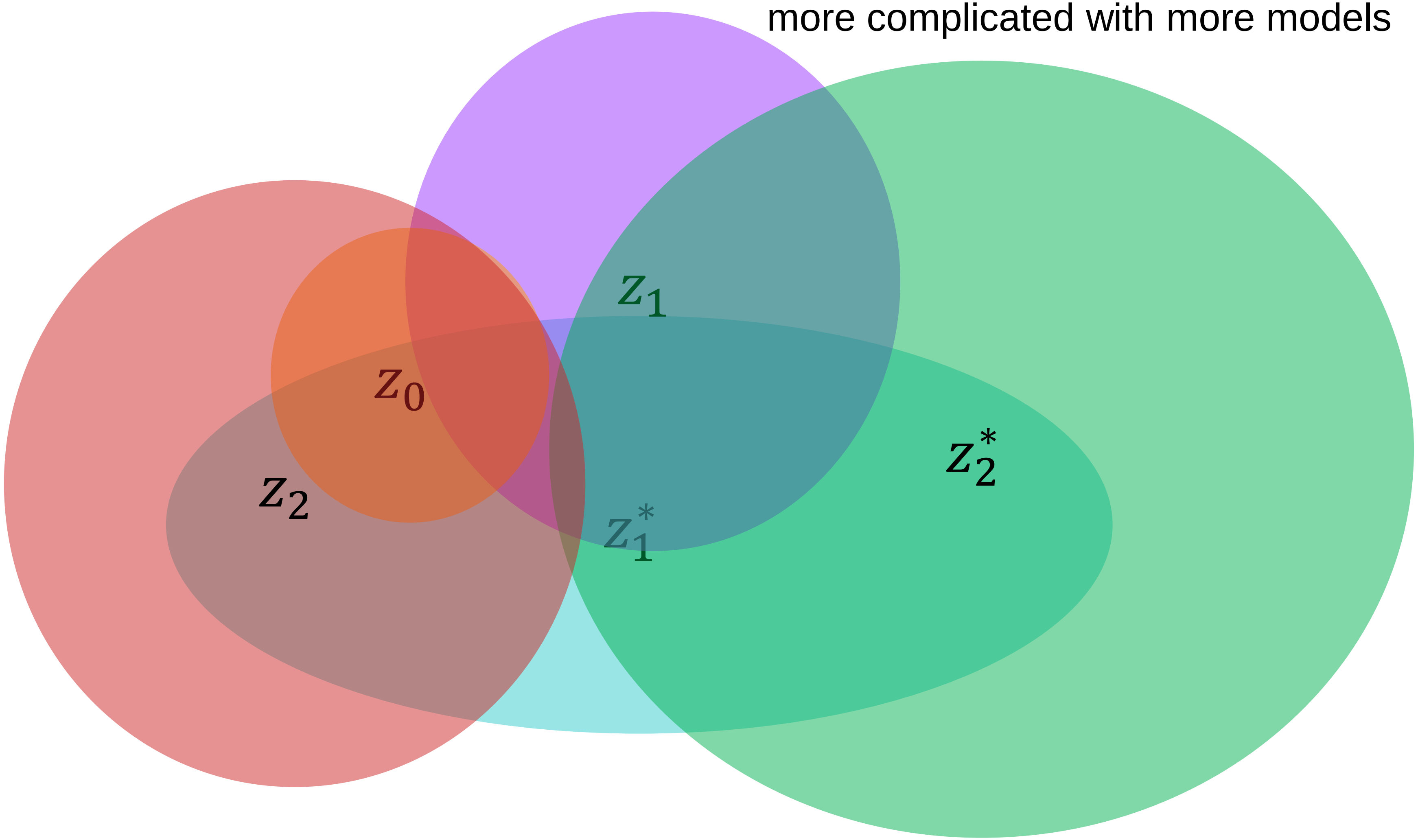
Sample Allocation Optimization

there are many configurations



Sample Allocation Optimization

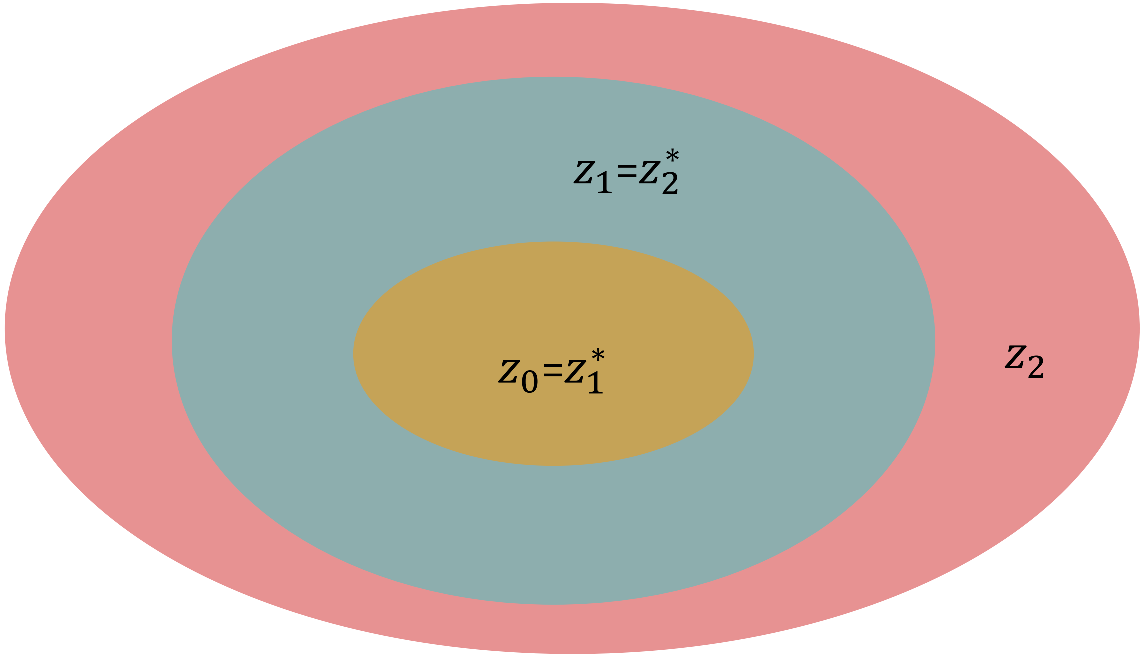
more complicated with more models



The domain of optimization is huge!

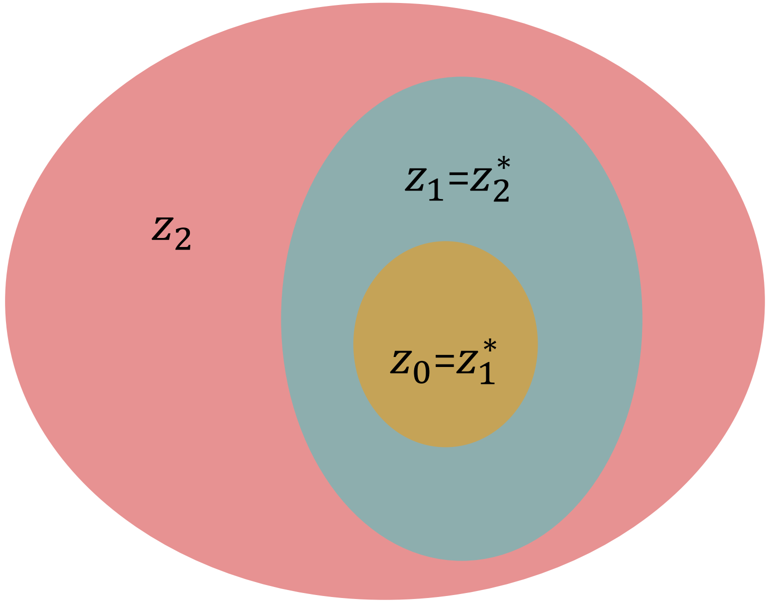
Sub-domain Technique

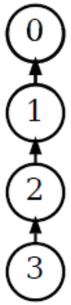
MFMC



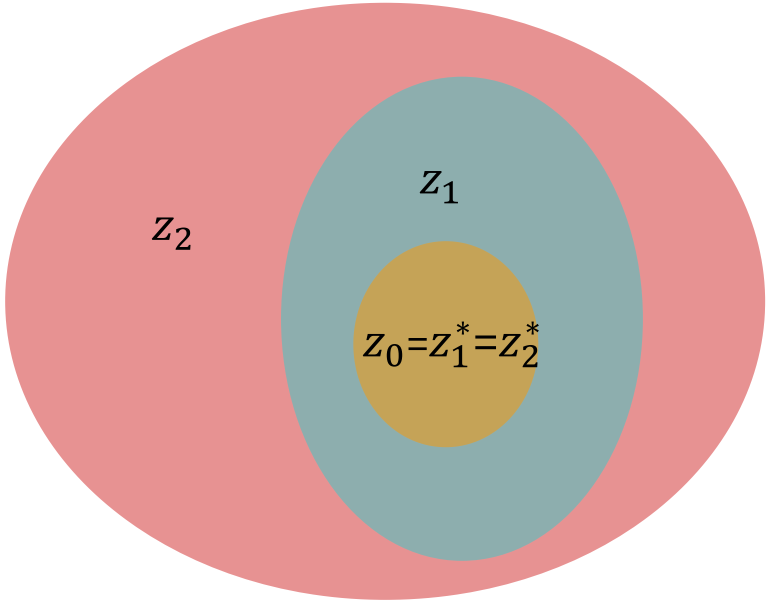
Sub-domain Technique

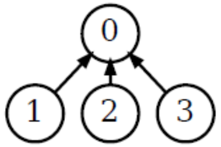
MFMC



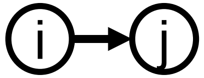


ACVMF





$$z_i^* = z_j$$

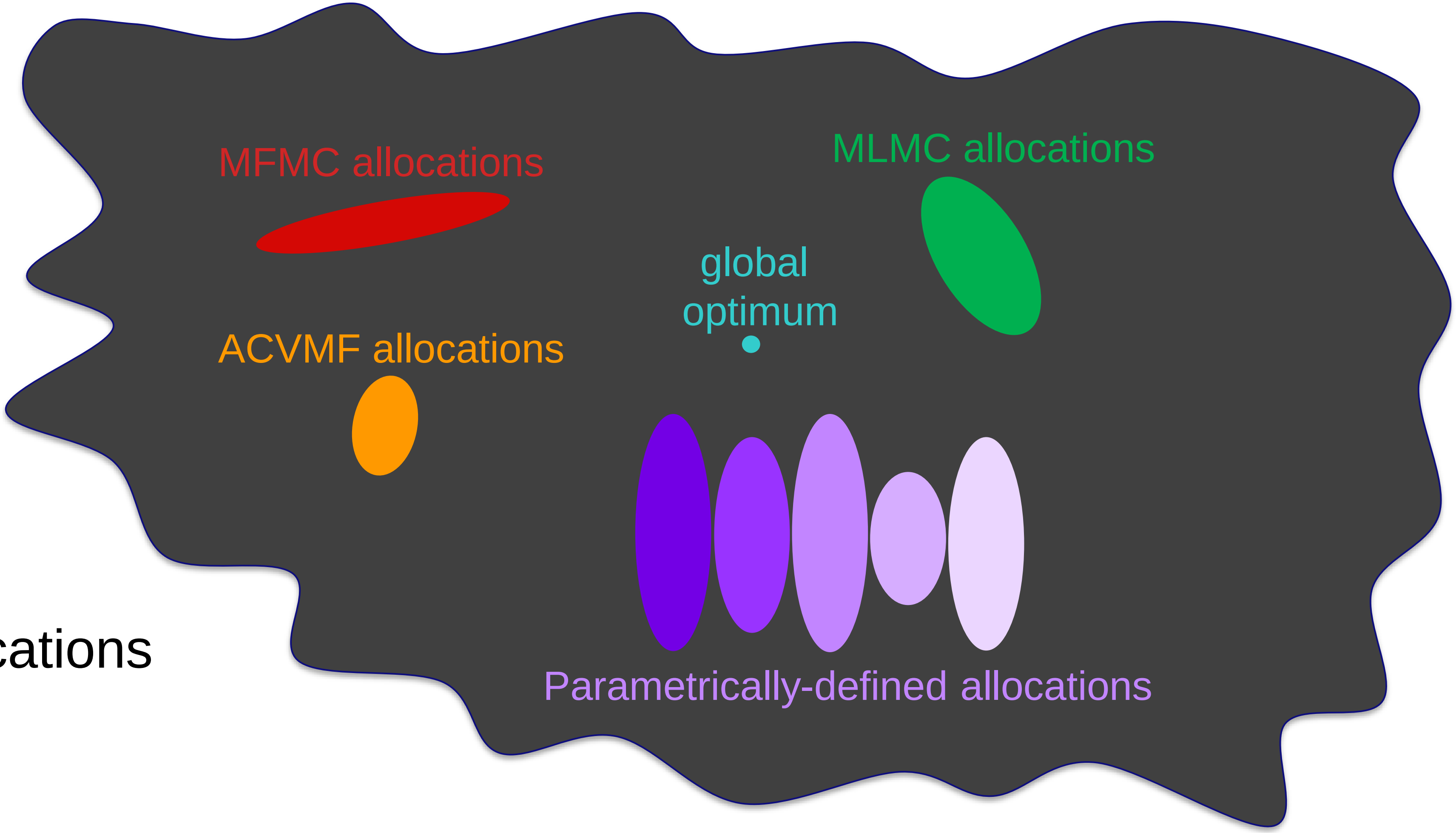


defines a tree structure

Key: sub-domain optimization is much easier

Parametrically-defined ACV Estimators

Domain of all possible
ACV sample allocations

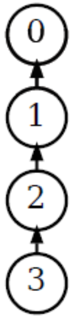


Parametric sample allocations
- Tunable recursion
- Model selection

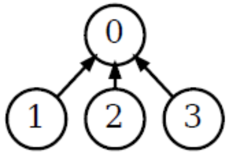
Tunable Recursion

- $z_i^* = z_j$ defined trees
- Search all possible trees!
- Can limit by depth/order

MFMC / MLMC

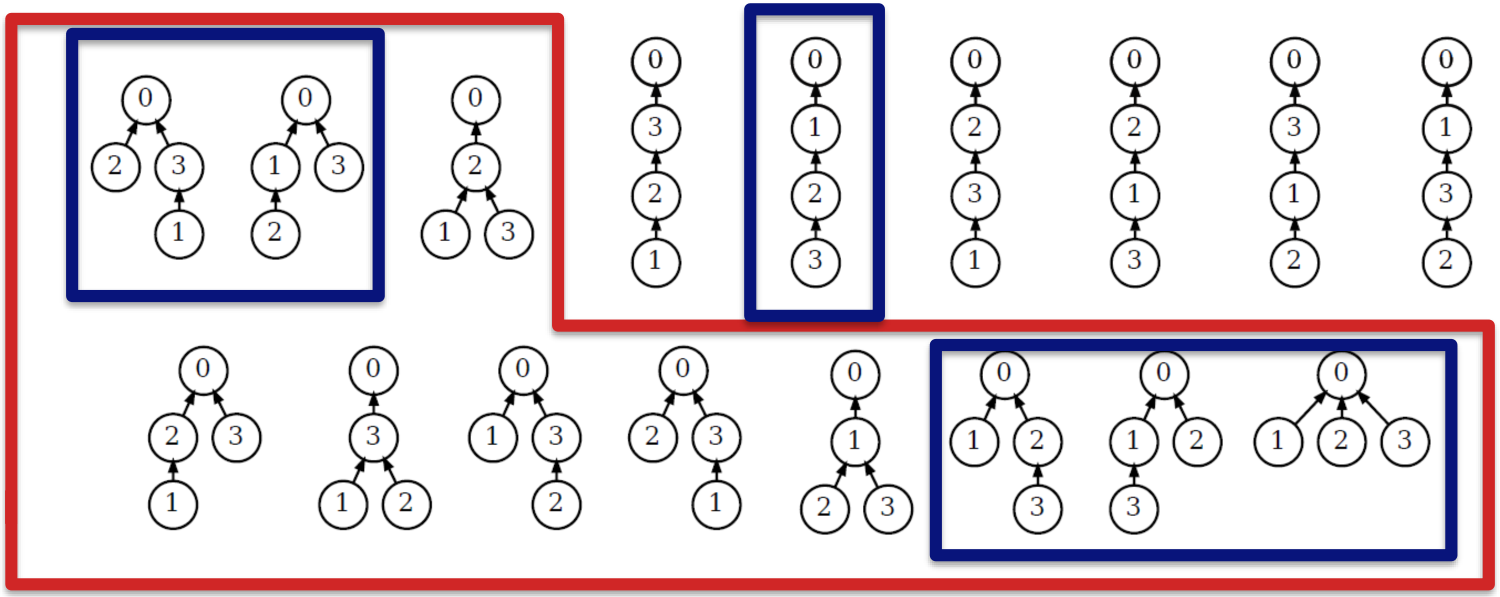


ACVMF / ACVIS



Ordered

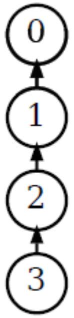
Max depth 2



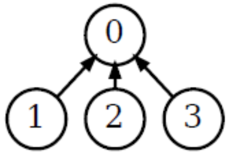
Tunable Recursion

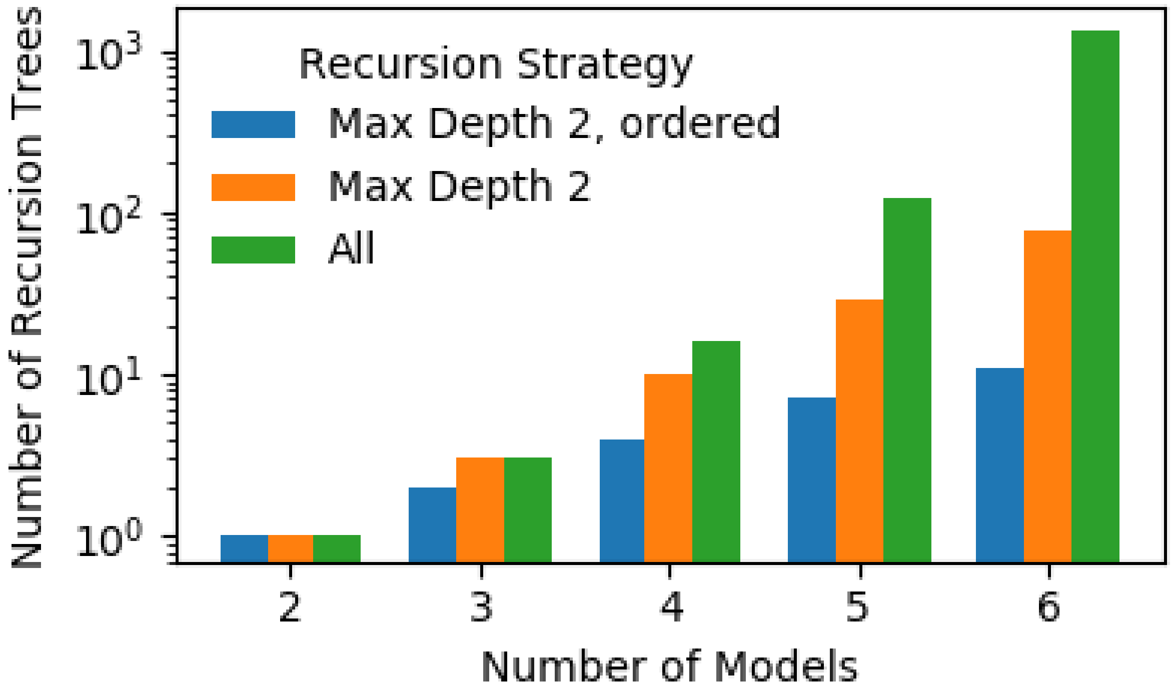
- $z_i^* = z_j$ defined trees
- Search all possible trees!
- Can limit by depth/order

MFMC / MLMC



ACVMF / ACVIS

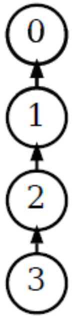




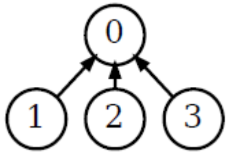
Tunable Recursion

- $z_i^* = z_j$ defined trees
- Search all possible trees!
- Can limit by depth/order

MFMC / MLMC



ACVMF / ACVIS

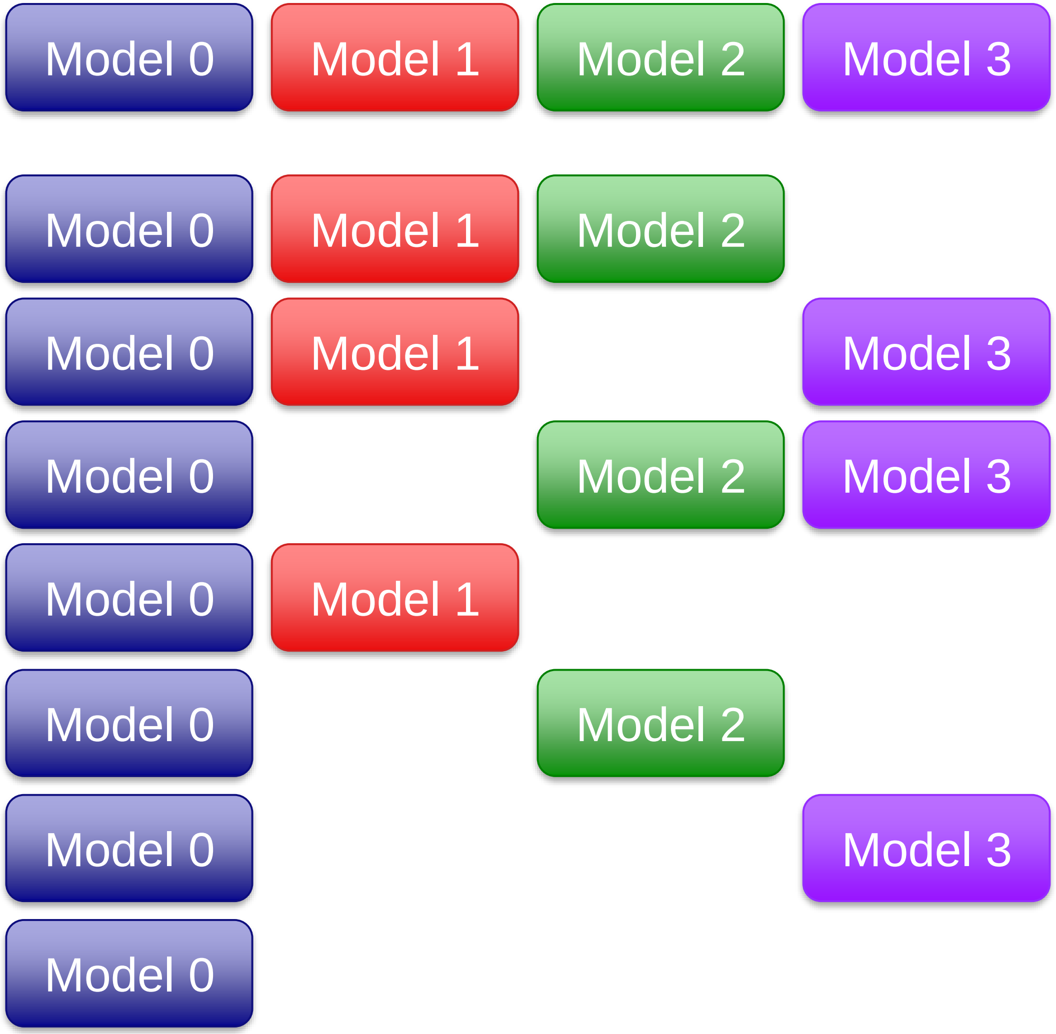


- Must decide how to structure z_i
- All tree structures + sampling strategy
  - Generalized Multifidelity (GMF)
  - Generalized Recursive Difference (GRD)
  - Generalized Independent Samples (GIS)

- Generalized methods find optimum of larger domain

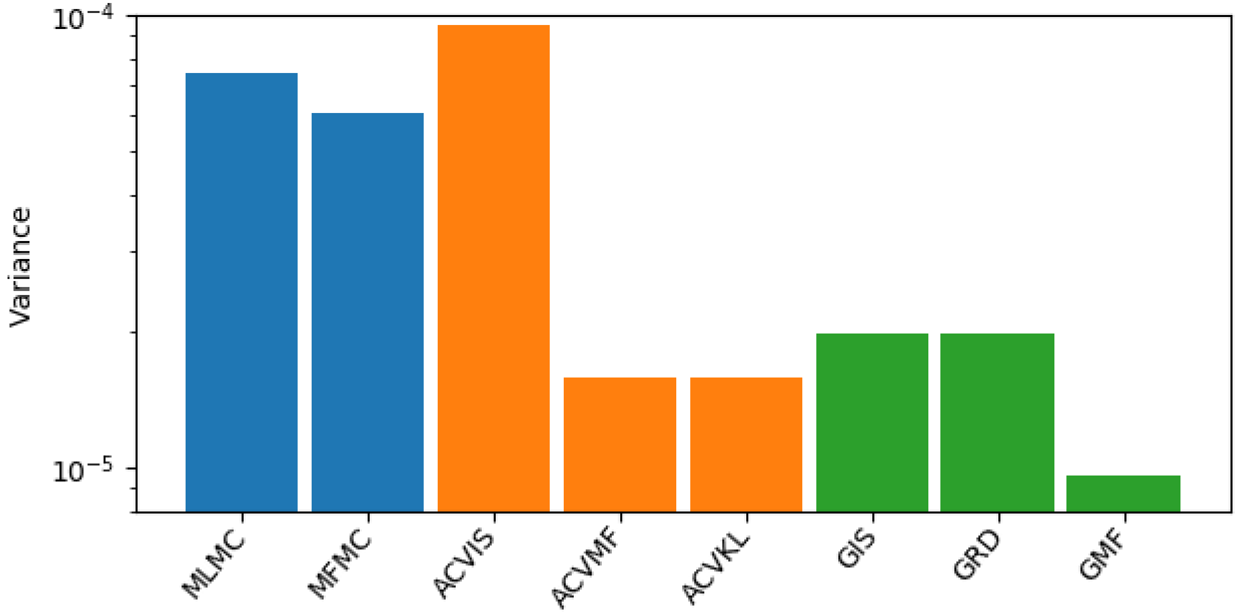
Automatic Model Selection

- Single optimization **should** cover all cases
  - e.g., $N_1 = 0$
- In practice, it doesn't
- Consider all possible subsets



Toy Problem

- 5 models: $Q_i(z) = z^{5-i}$
- Input distribution: $z \sim U(0,1)$
- Model costs: $w_0 = 1, w_i = 10^{-i-1}$
- Cost budget: $\hat{w} = 20$

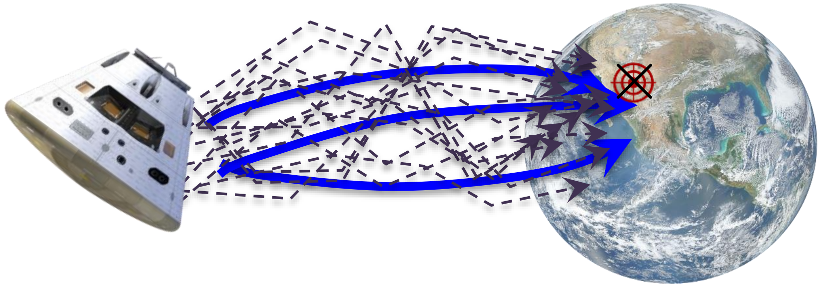


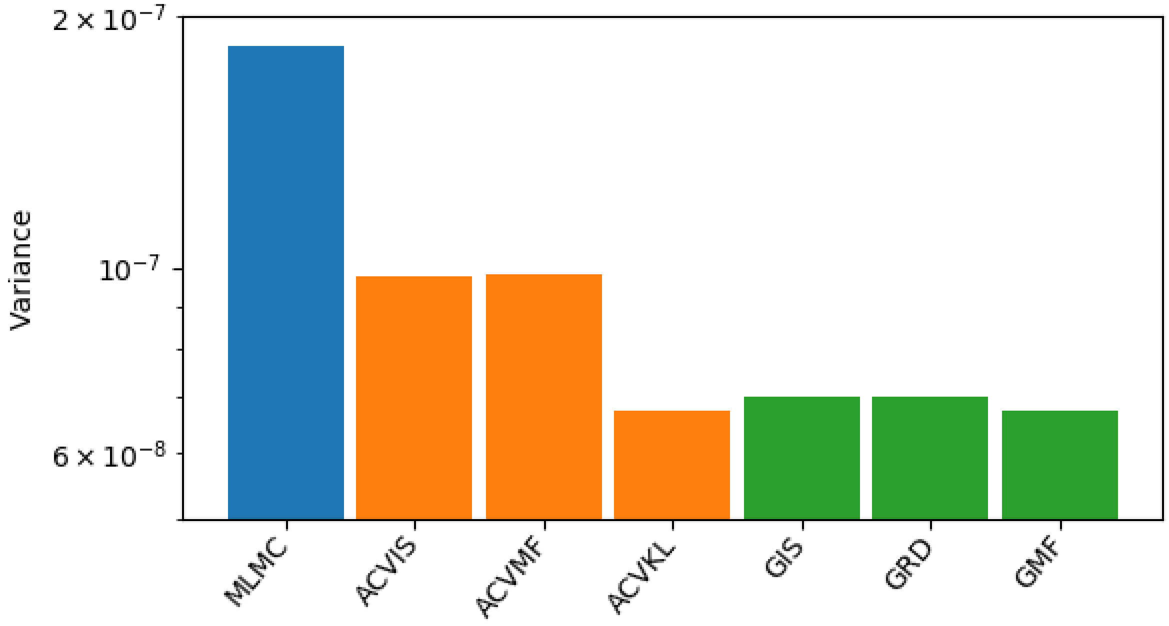
Gorodetsky, A A., et al. Journal of Computational Physics (2020)

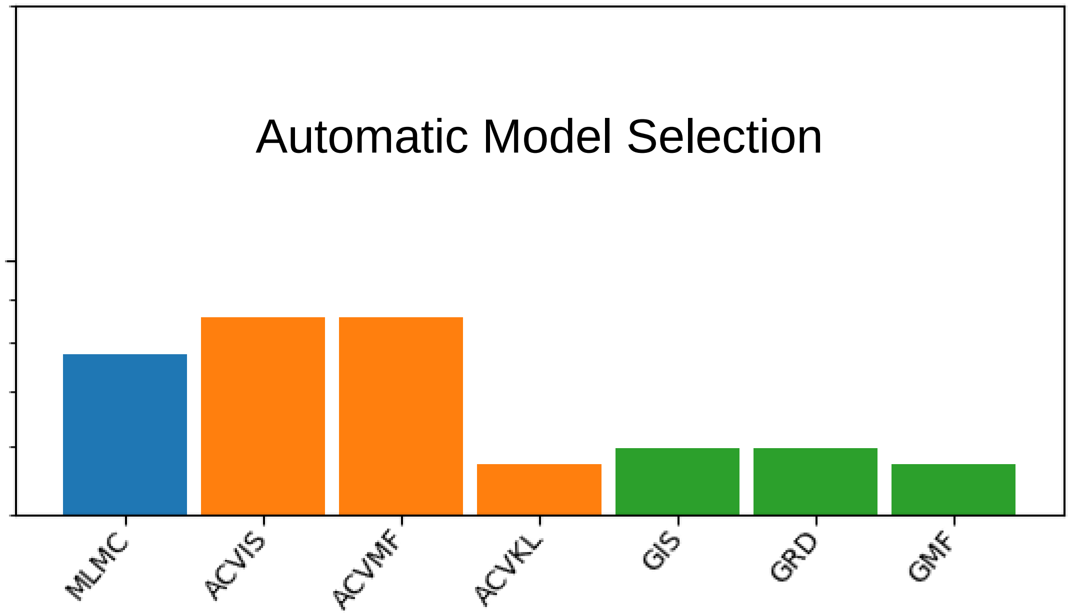
Trajectory Simulation Example

4 Models

- Trajectory Simulator
- Increased time integration
- Decreased physics
- Machine learning surrogate



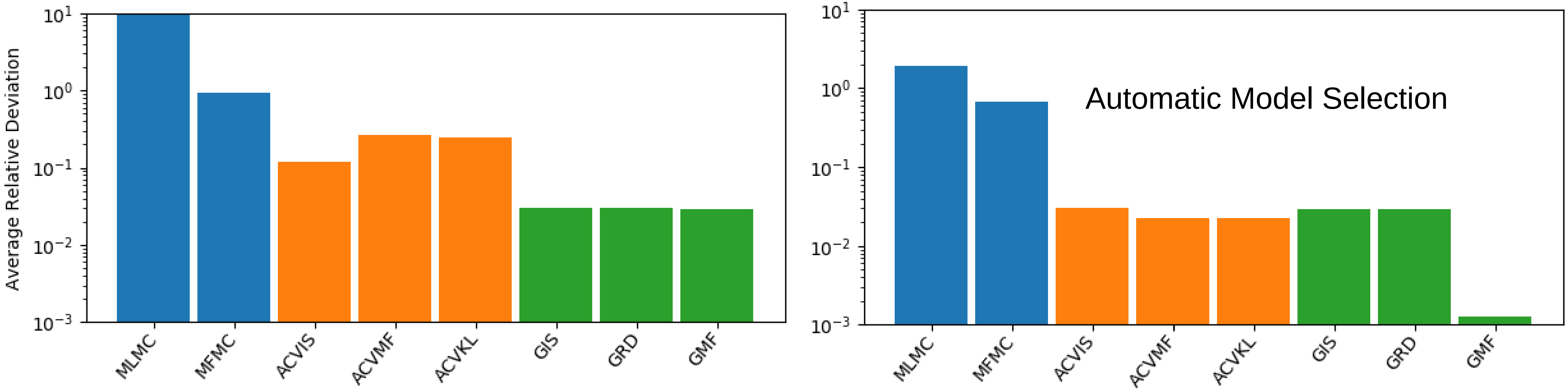




Random Model Scenarios

~5 Million

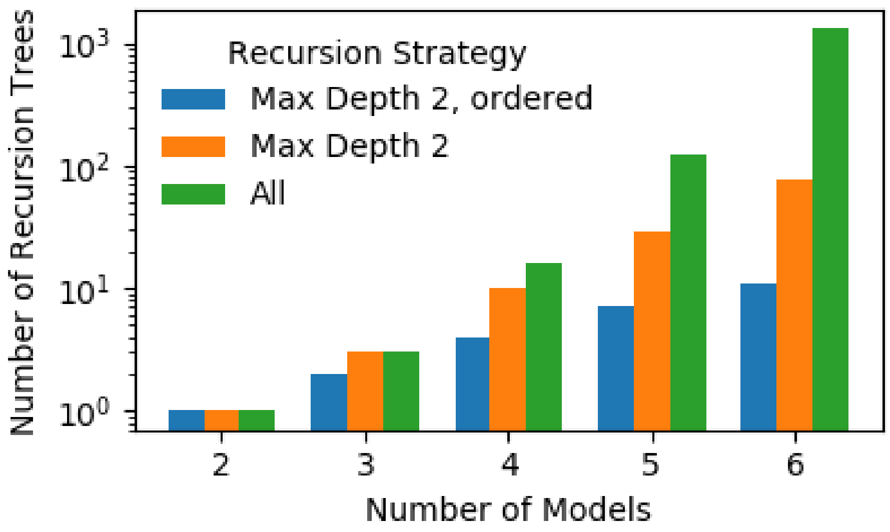
1. Choose number of models [2-6]
2. Sample random correlation matrix
3. Sample random variances
4. Sample random model costs (relative to hifi cost)
5. Perform allocation optimization for all algorithms (fixed cost budget)

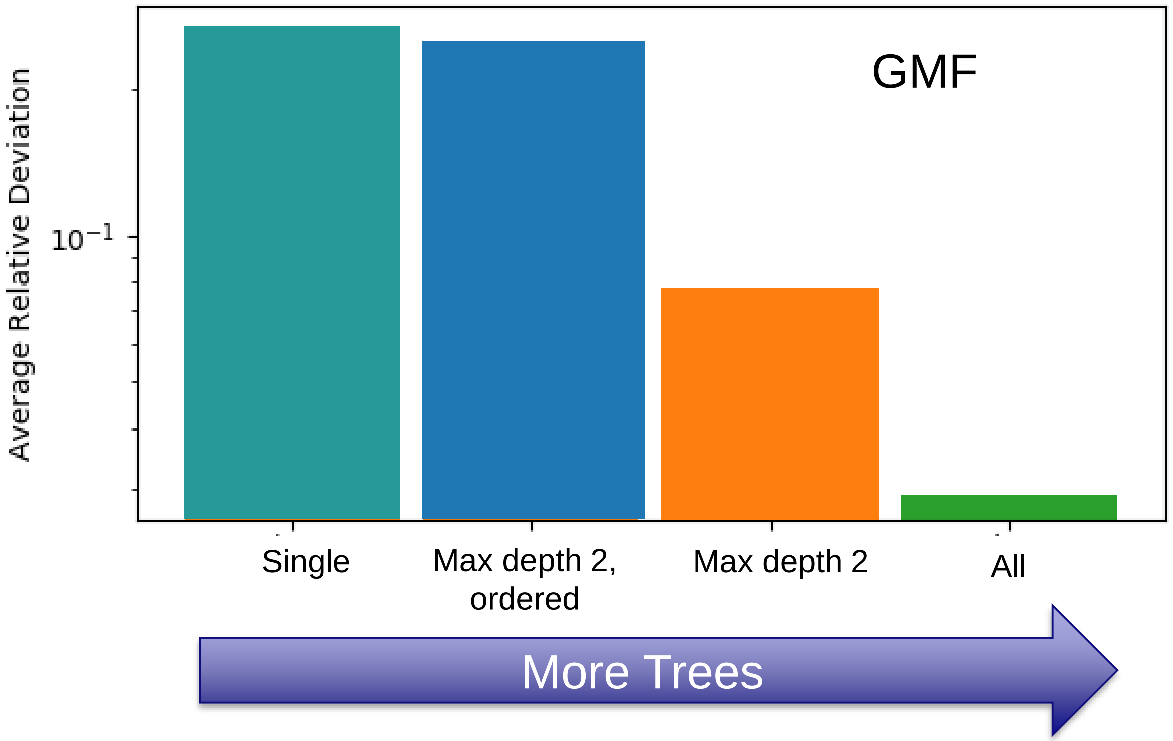


Random Model Scenarios

~5 Million

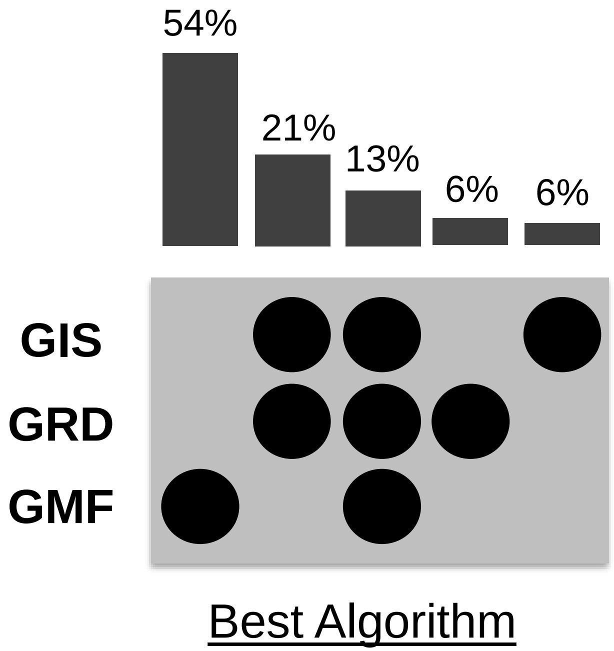
1. Choose number of models [2-6]
2. Sample random correlation matrix
3. Sample random variances
4. Sample random model costs (relative to hifi cost)
5. Perform allocation optimization for all algorithms (fixed cost budget)

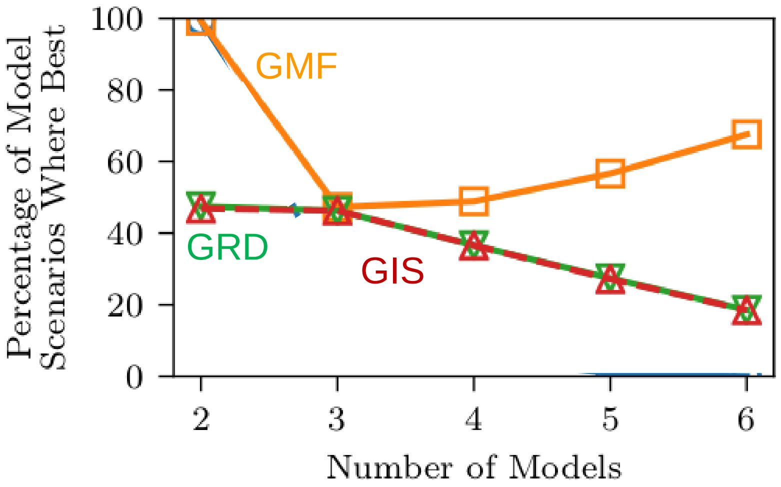




~5 Million Random Model Scenarios

Comparison of sampling strategies





GMF best 67% of model scenarios

GMF scales better to more models

Automatic model selection skews even more to GMF (81%)

Part 1 Summary

- Parametrically-defined ACVs enable optimization over an increased domain

- Parametrically-defined ACVs (GMF, GRD, GIS) give more optimal sample allocations than existing ACVs
  - at worst, the same
  - automatic model selection enhances this further

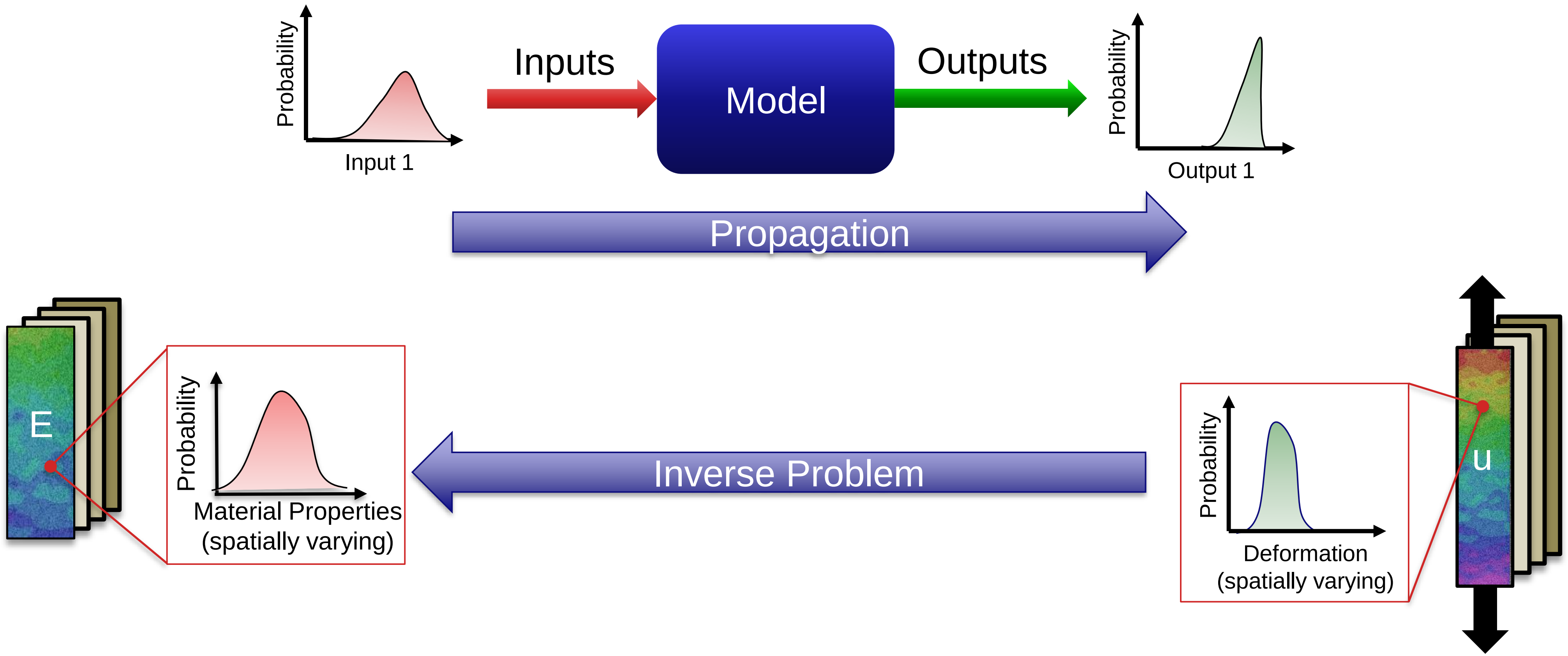
- GMF with automatic model selection best in most cases studied

Part 2: Physics-informed ML

Physics-informed Generative Adversarial Networks (PIGANs)

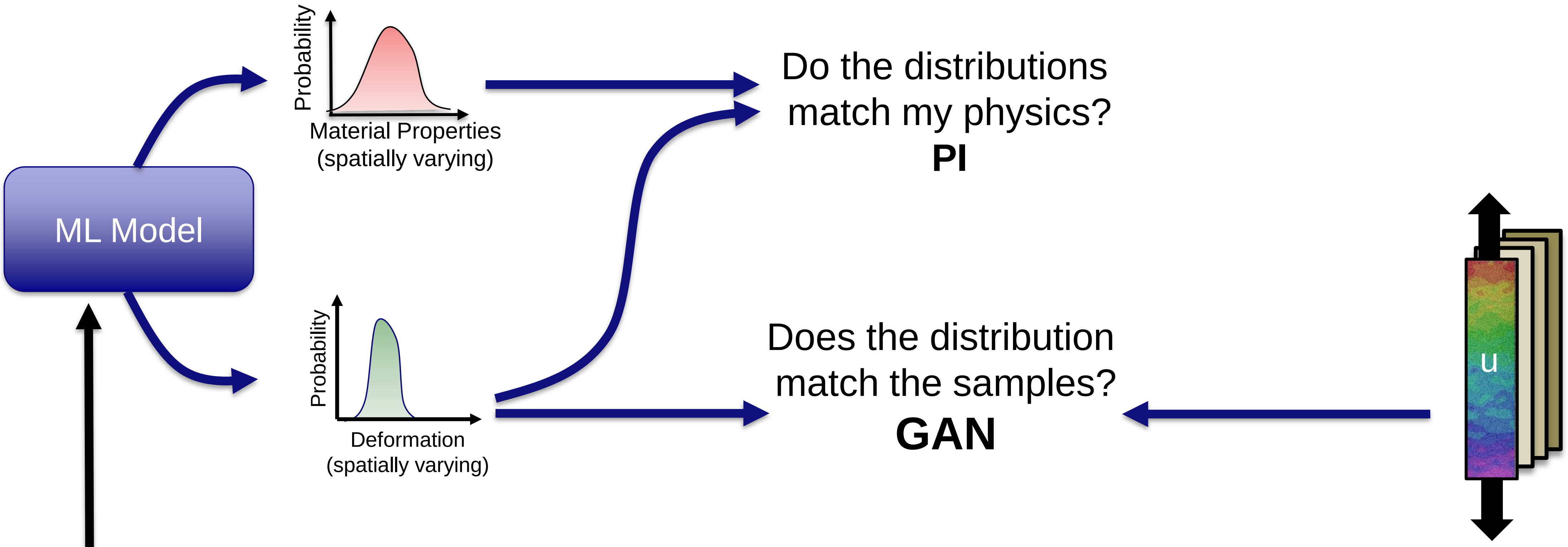
Inverse Problem

Given output distributions can we reconstruct input distributions?



Motivation: Infinite virtual testing

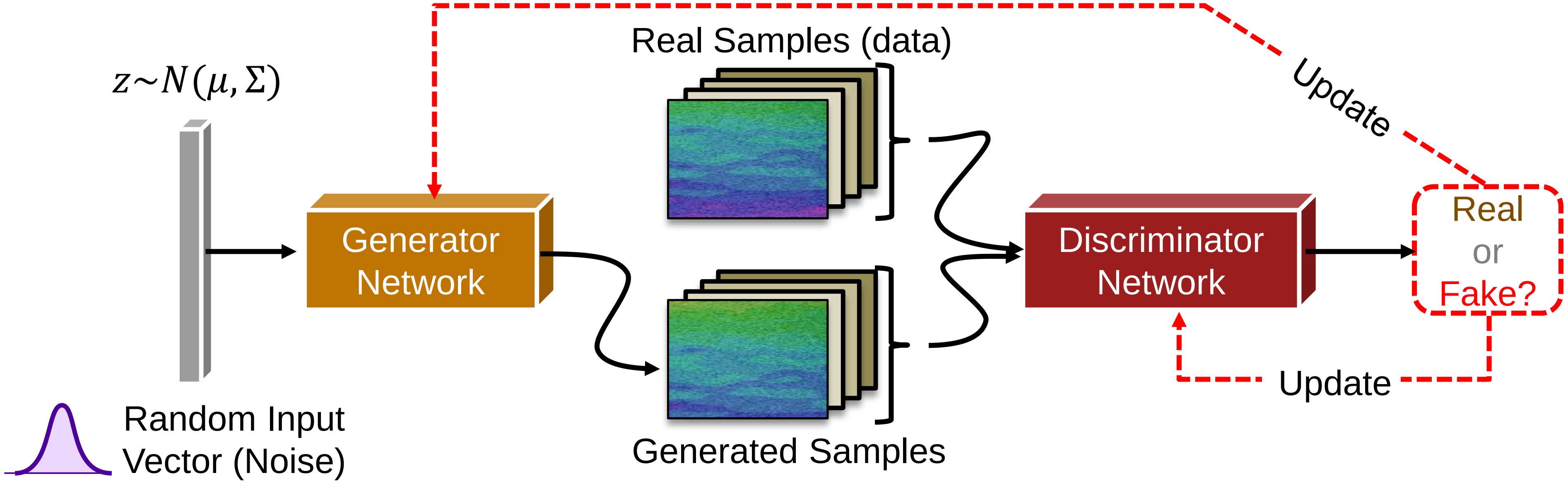
Physics-informed Generative Adversarial Networks (PIGANs)



Train until the answer is “yes” to both questions

Generative Adversarial Networks (GANs)

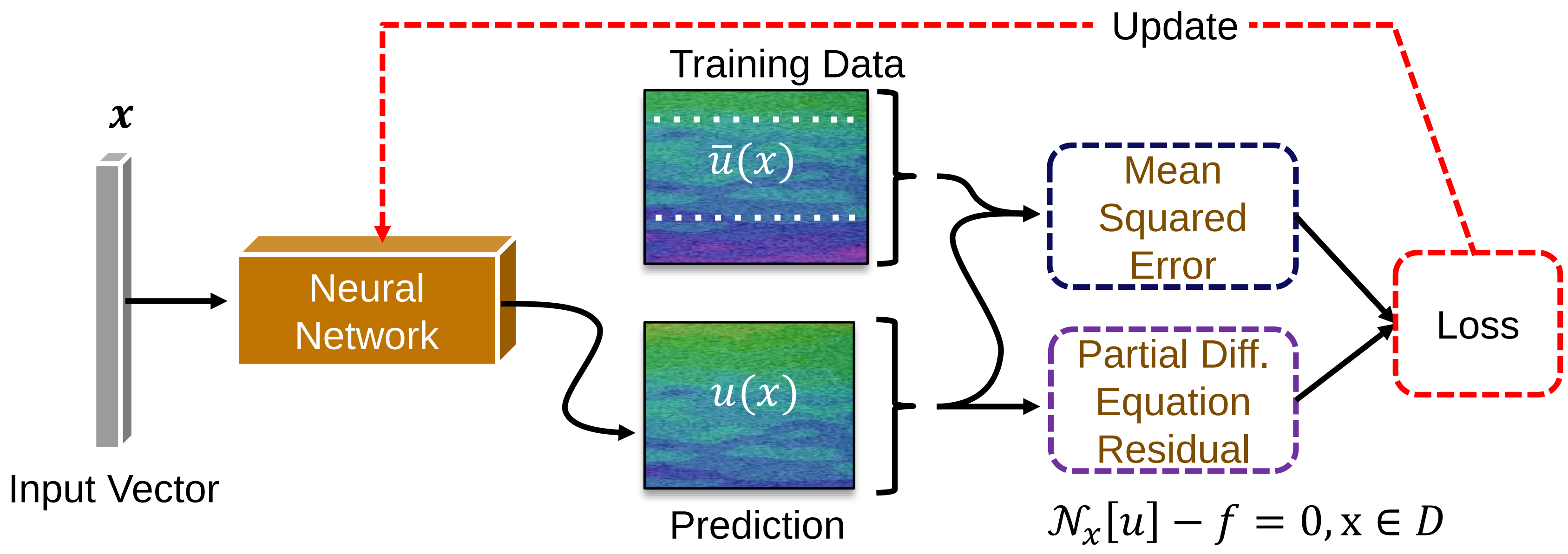
- Two neural networks compete in a zero-sum game
  - Discriminator: Try to discern real from fake samples
  - Generator: Try to fool the discriminator
- GANs are capable of learning to sample from complex probability distributions



Goodfellow, Ian J., et al. "Generative adversarial networks." arXiv preprint arXiv:1406.2661 (2014)

Physics-informed Neural Networks (PINNs)

- Neural network trained to produce solution $u(x)$ to a partial differential equation
- Two-part objective:
  1. Minimize mean squared error with respect to data
  2. Output physically admissible solutions throughout the problem domain

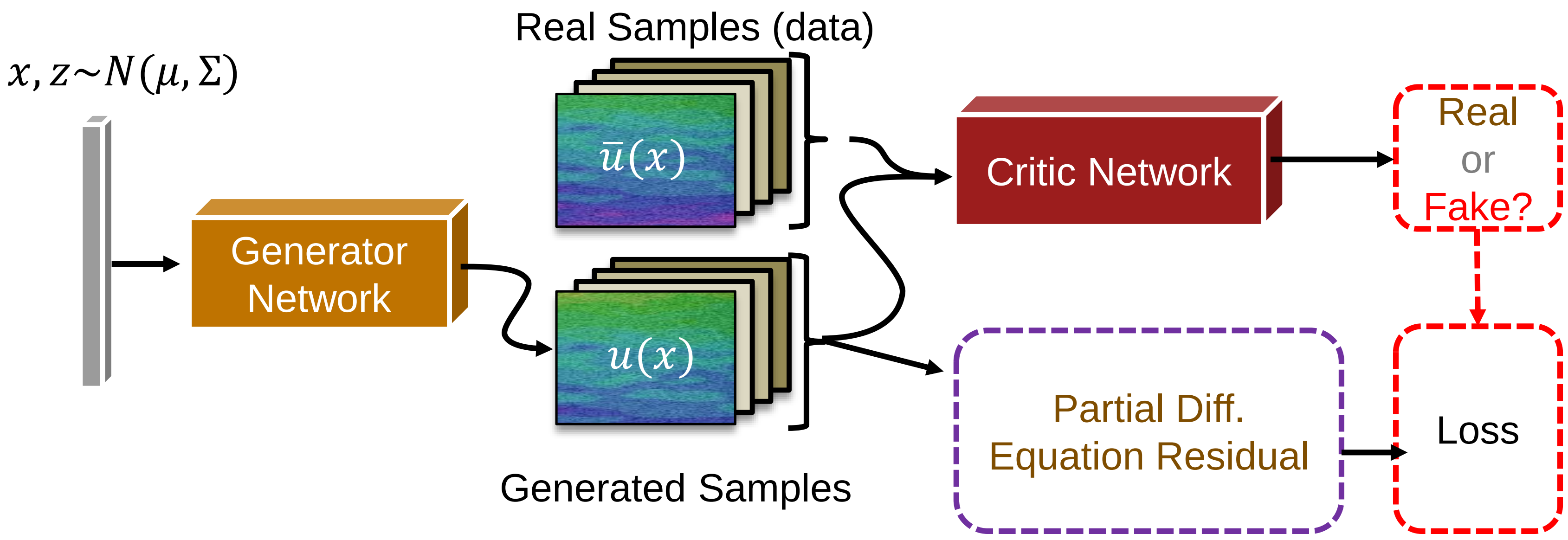


$$\mathcal{N}_x[u] - f = 0, \mathrm{x} \in D$$
$$B_x[u] - b = 0, x \in \Gamma$$

Raissi et al. "Physics-informed neural networks: A deep learning framework for solving forward and inverse..." *J. Comp. Phys.* (2019)

PIGANs

- PINNs + GANs
- Learns physically admissible, *probabilistic* solution, $u(x)$

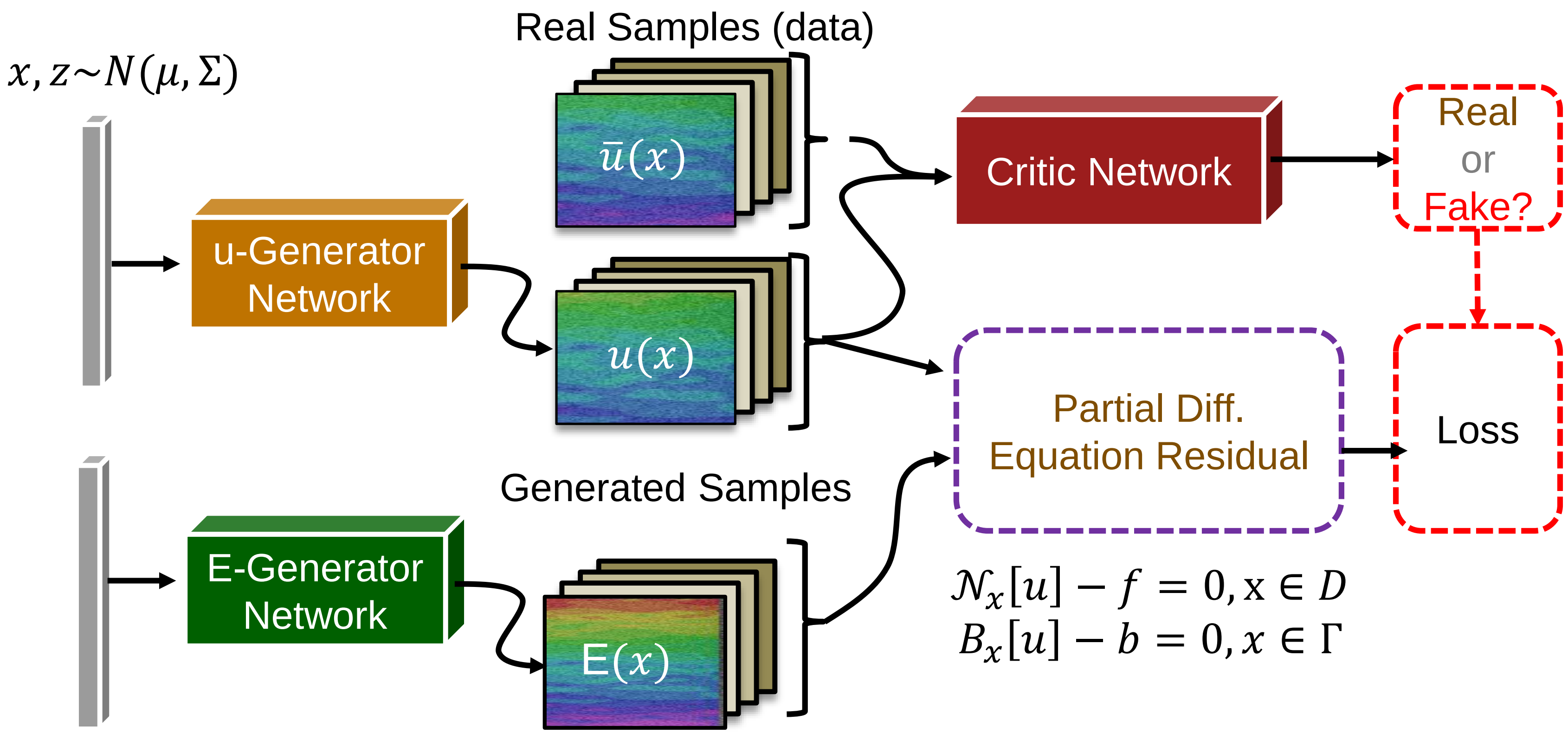


$$\mathcal{N}_x[u] - f = 0, \mathrm{x} \in D$$
$$B_x[u] - b = 0, x \in \Gamma$$

Yang et al. "Physics-informed generative adversarial networks for stochastic differential equations." arXiv preprint arXiv:1811.02033 (2018)

PIGANs

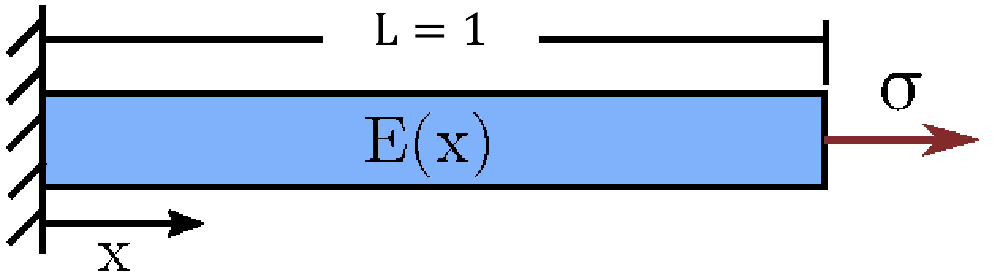
- Additional generators
  - linked through the partial differential equations
  - solve inverse problems (even without direct observations)



$$\mathcal{N}_x[u] - f = 0, x \in D$$
$$B_x[u] - b = 0, x \in \Gamma$$

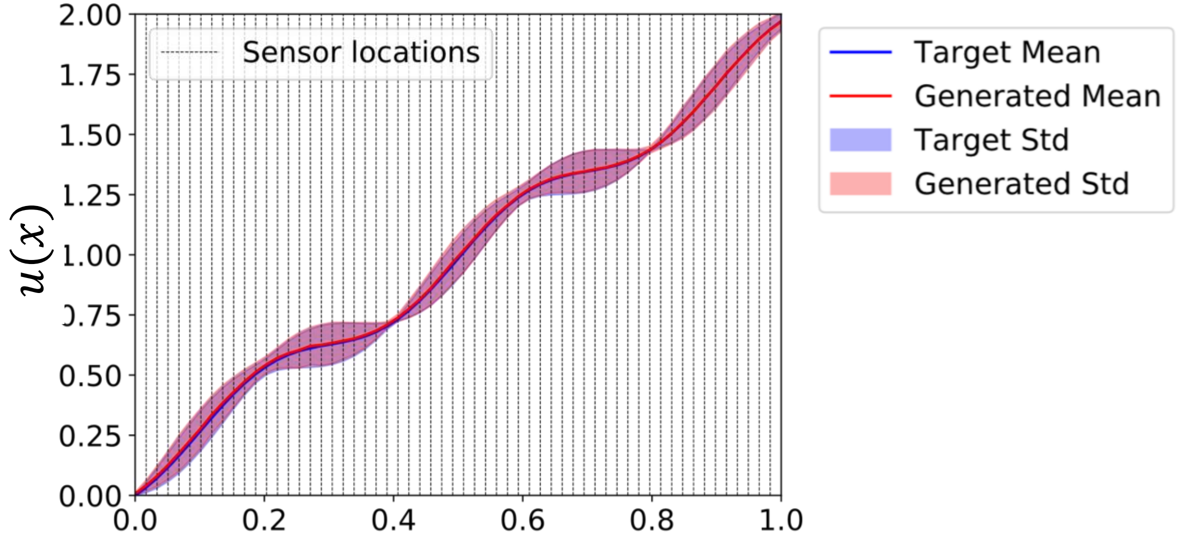
Example Problem: One-dimensional Bar in Tension

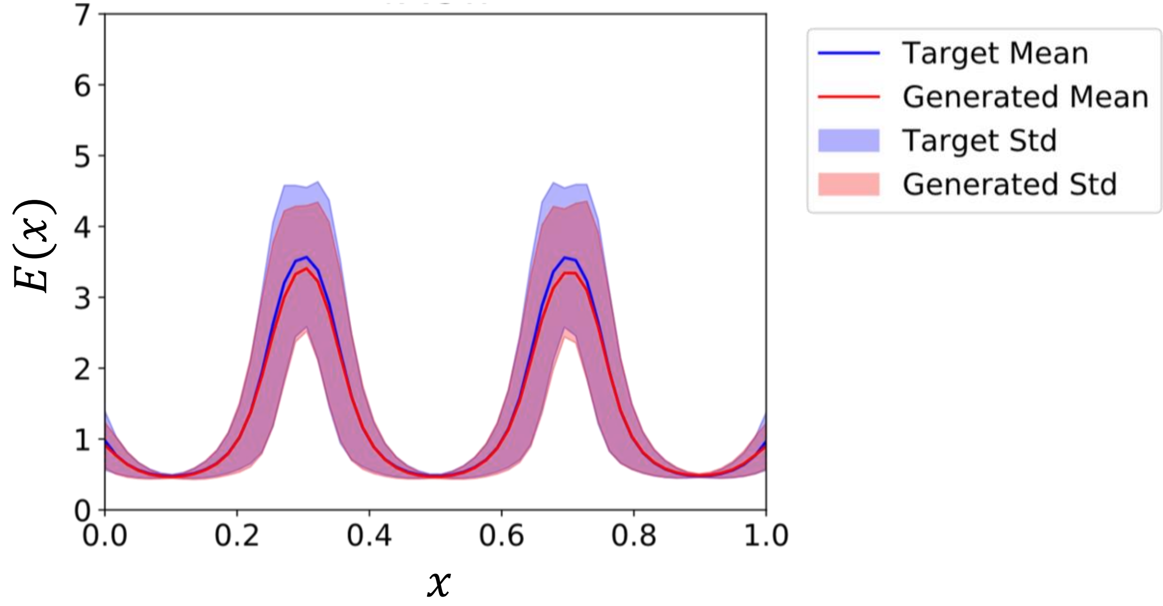
- 1D bar in tension with random, spatially-varying elastic modulus, $E(x)$
- Given measurements of displacement $u(x)$
- PIGAN learns material behavior and simultaneously quantifies uncertainty
  1. Learns the distribution $u(x)$ given 1,000 measurements at 60 sensors along bar
  2. Uses governing physics to inversely determine $E(x)$; i.e., no measurements of $E(x)$



$$E(x) = 1.2 + (sin(\frac{5\pi}{L}x + \frac{\pi}{2}(\theta - \frac{1}{2})))^{-1}$$

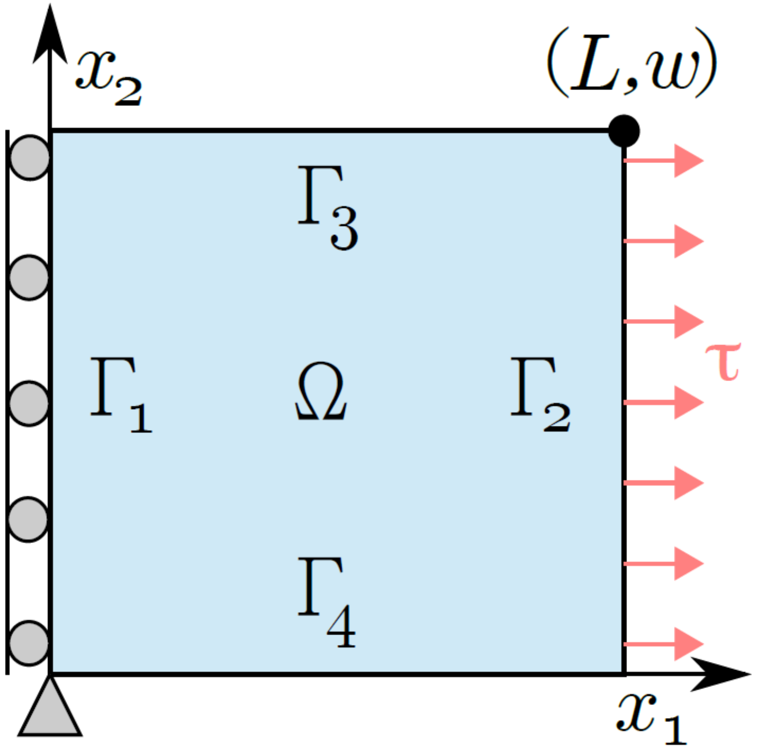
Random variable



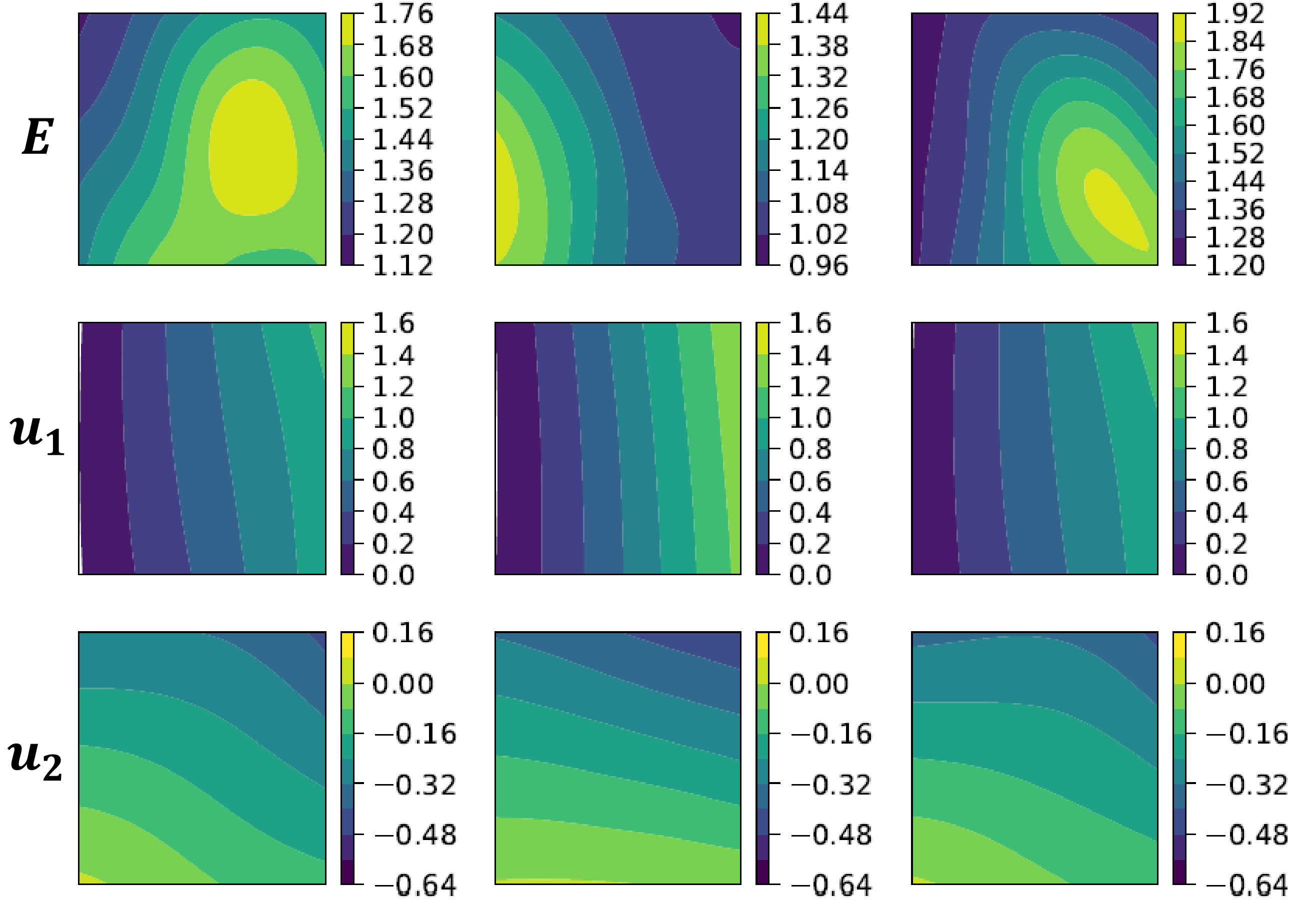


Example Problem: Two-dimensional Plate in Tension

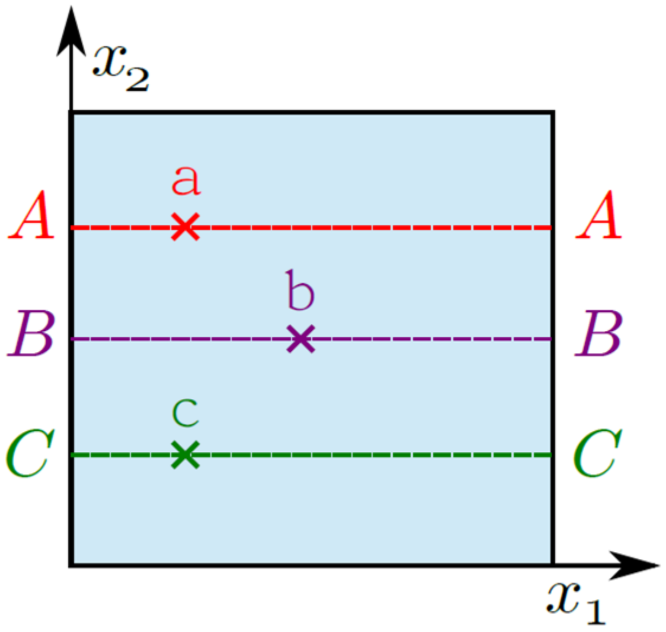
- 2D plate in tension with random, spatially-varying elastic modulus, $E(\boldsymbol{x})$
- Measurements of $\vec{u}(\boldsymbol{x})$ generated using Python finite element package FEniCS
- PIGAN learned both the u and E random fields

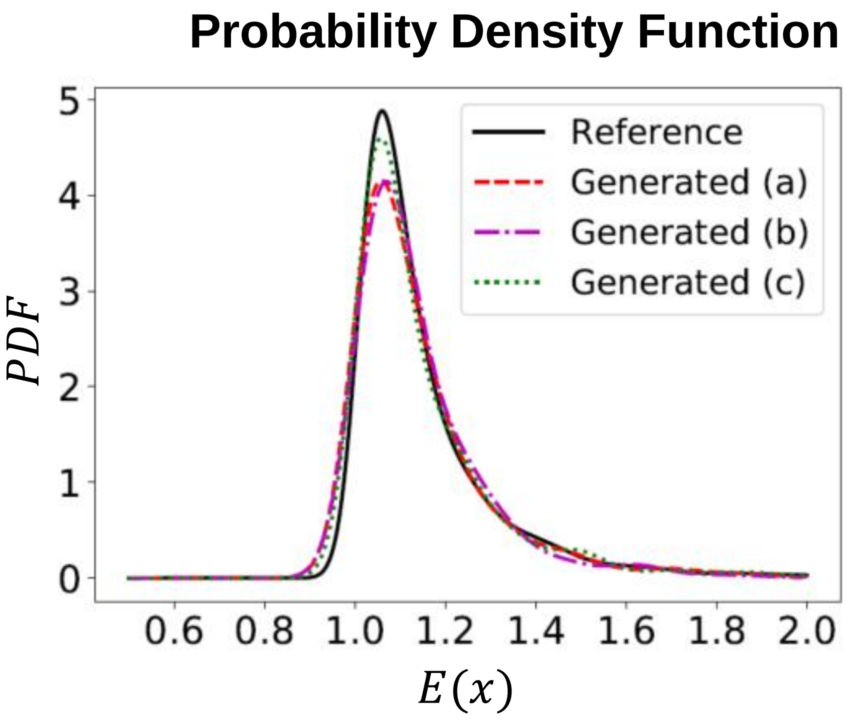


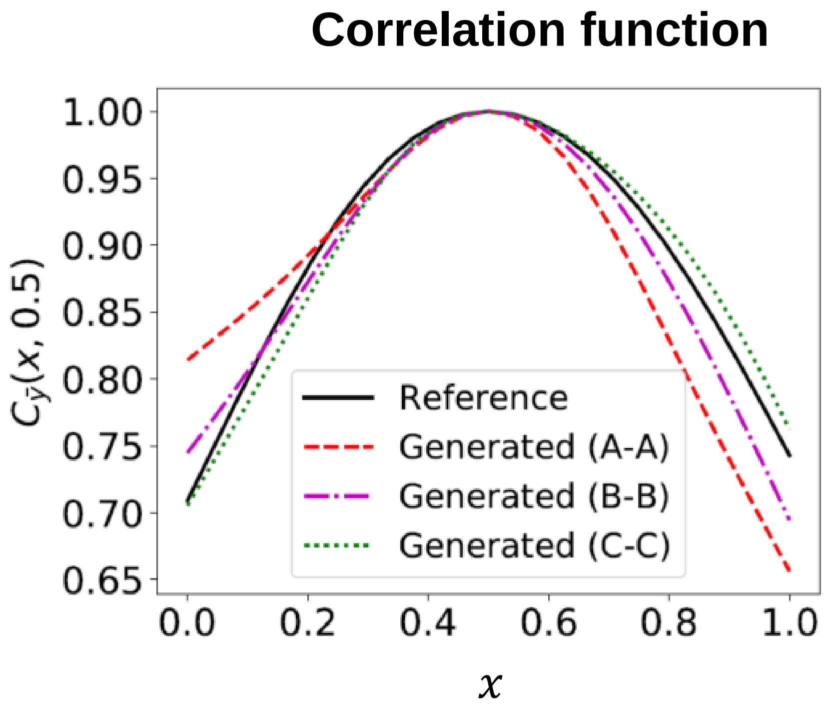
Randomly Generated Fields (Data Snapshots)



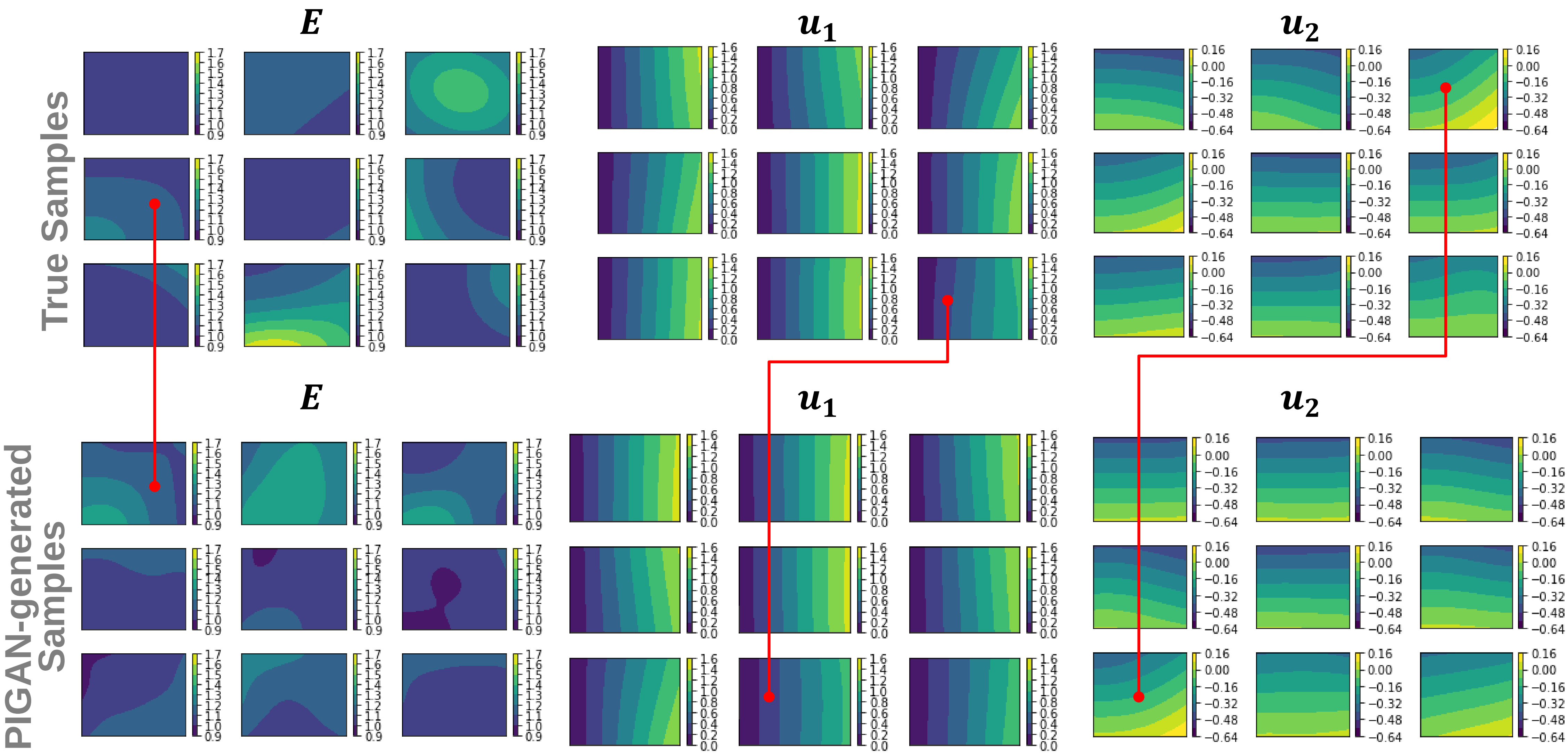
Example Problem: Two-dimensional Plate in Tension



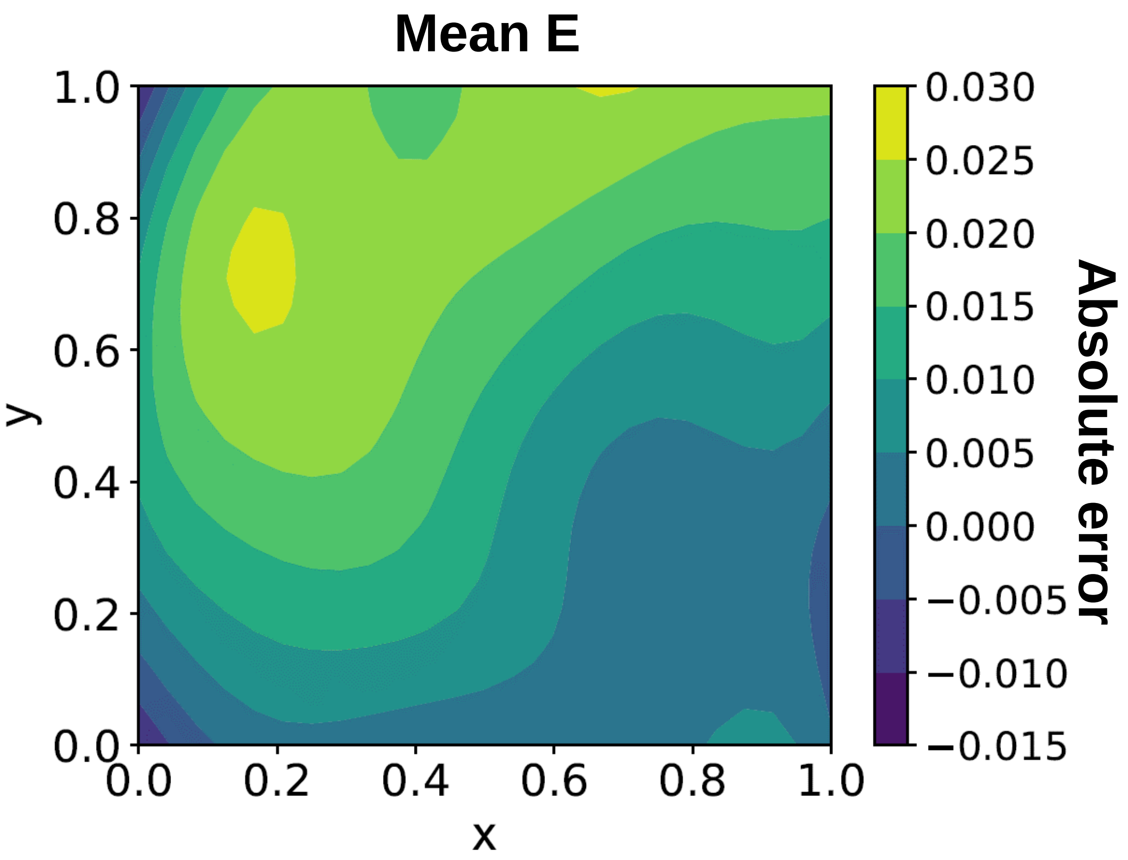


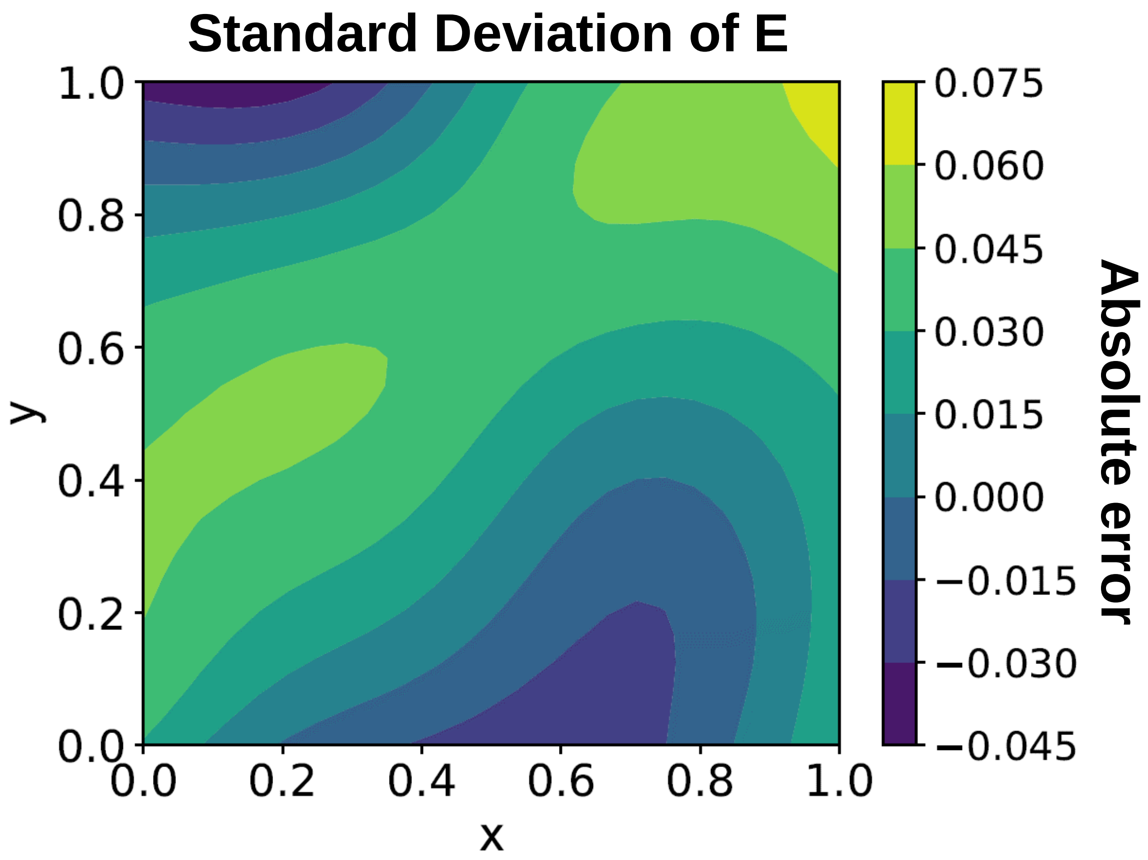


Example Problem: Two-dimensional Plate in Tension



Example Problem: Two-dimensional Plate in Tension

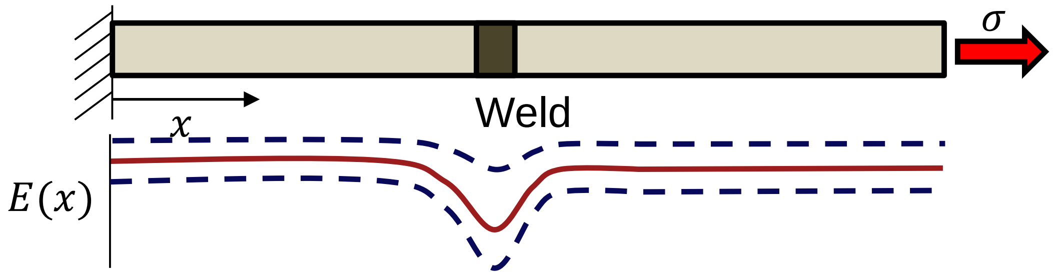




Note: magnitude of E is $\mathcal{O}(1)$

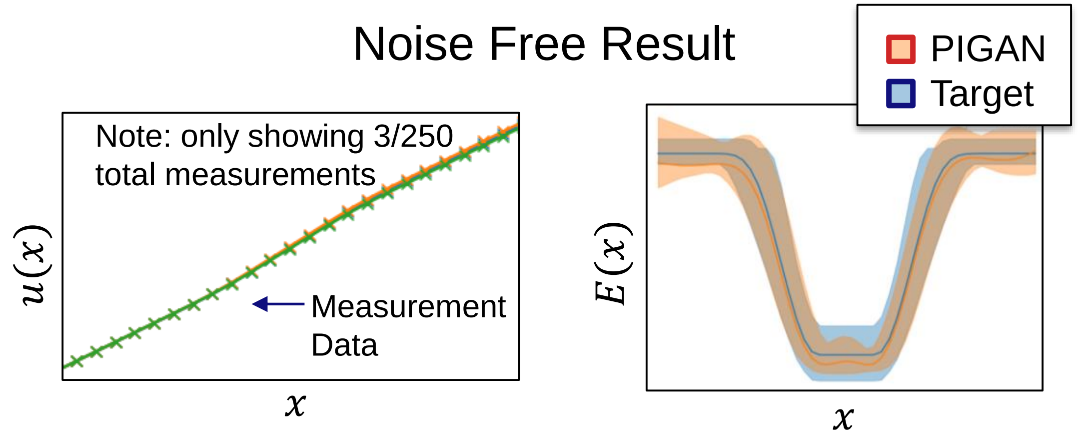
Addressing Noisy Measurements

- 1D bar in tension with random, spatially-varying elastic modulus, $E(x)$
- Given measurements of displacement $u(x)$
- Gaussian noise added to measurements

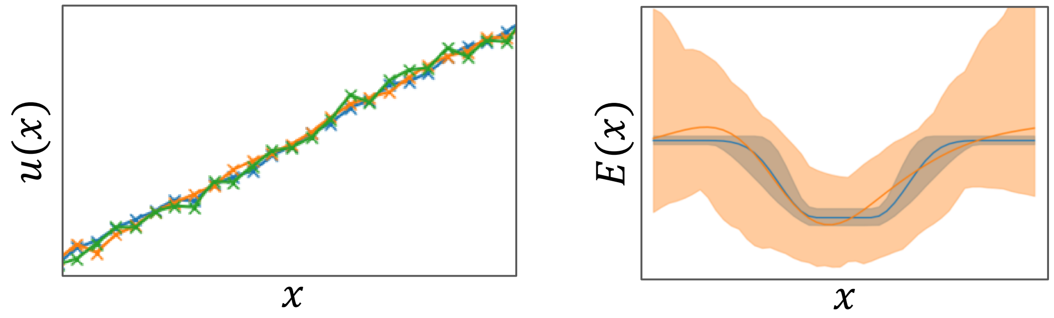


Noise in measurements leads to inaccuracies in inference!

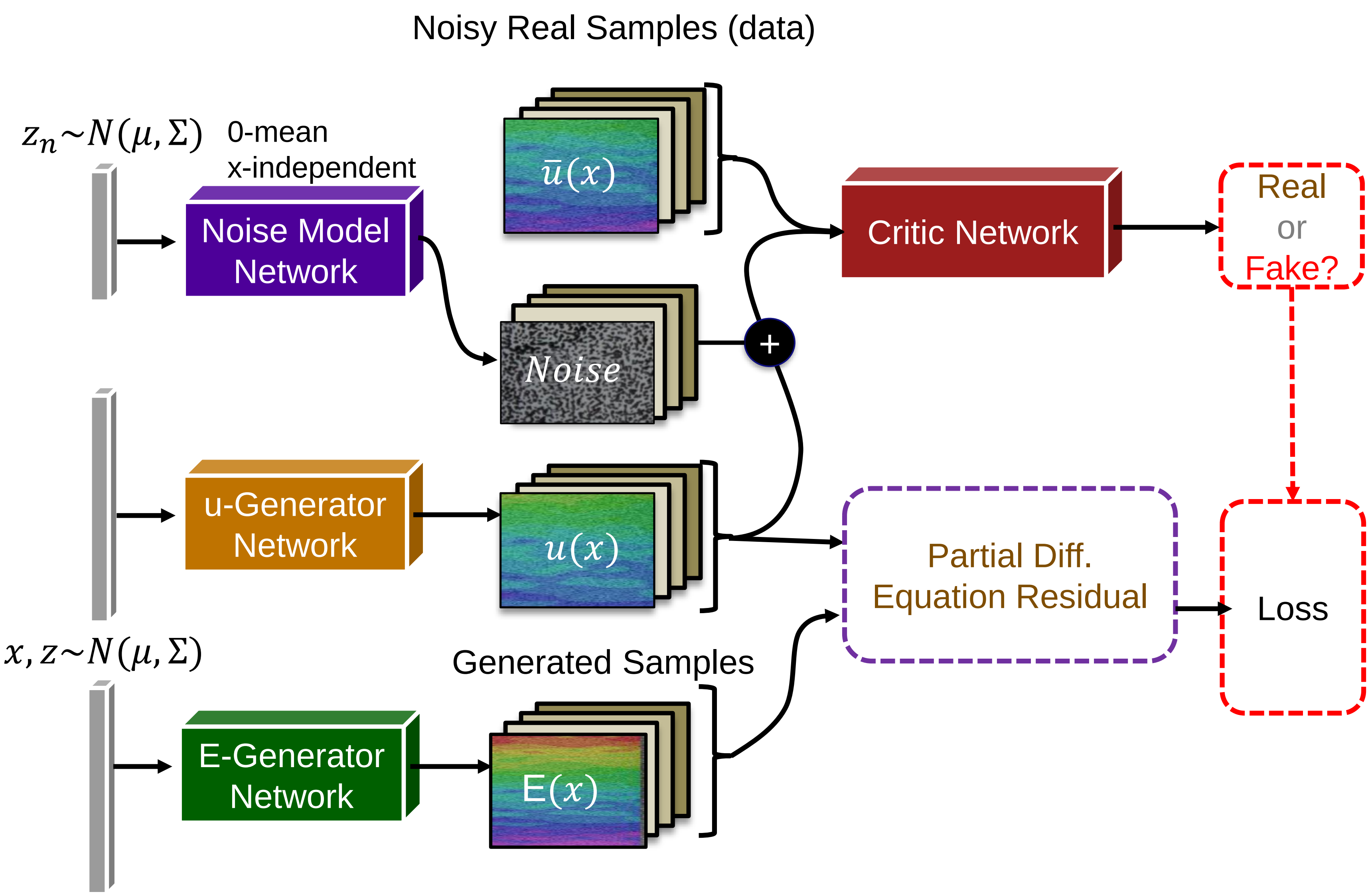
Noise Free Result



Noisy Result

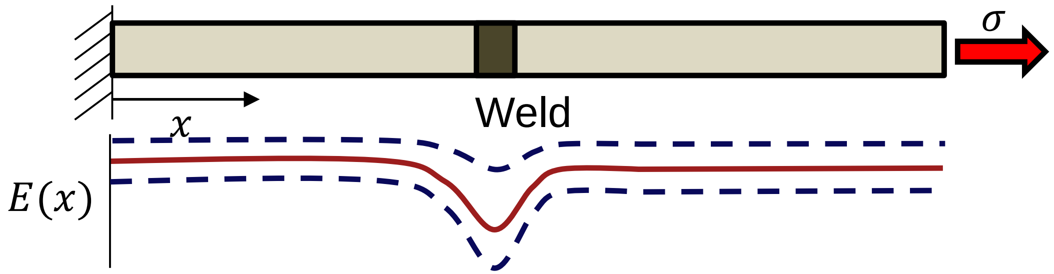


Addressing Noisy Measurements

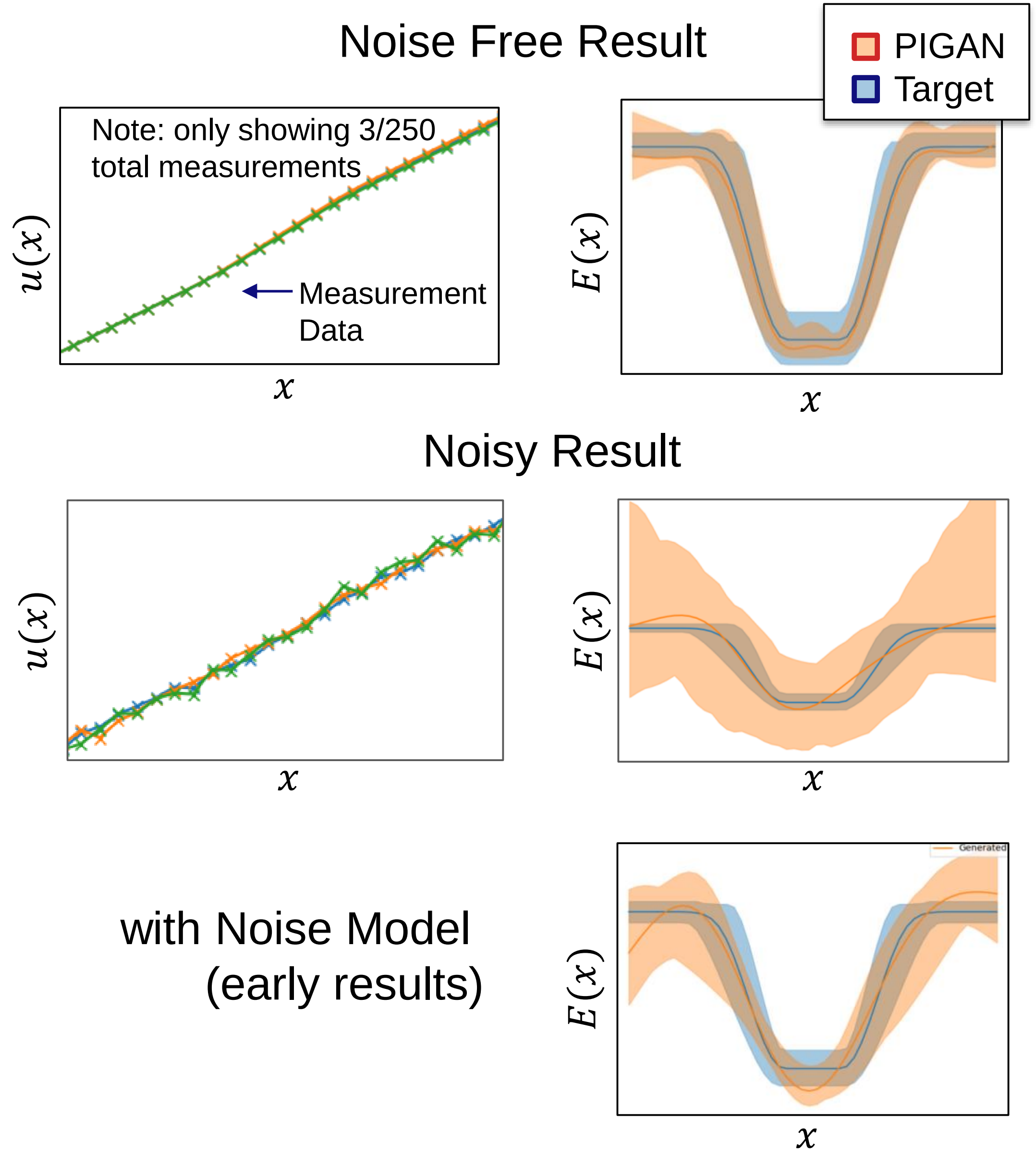


Addressing Noisy Measurements

- 1D bar in tension with random, spatially-varying elastic modulus, $E(x)$
- Given measurements of displacement $u(x)$
- Gaussian noise added to measurements



Explicitly modeling noise can help regain precision



Part 2 Summary

- PIGANS enable inference on unseen data
  - …as long as it can be related to observed data through PDEs

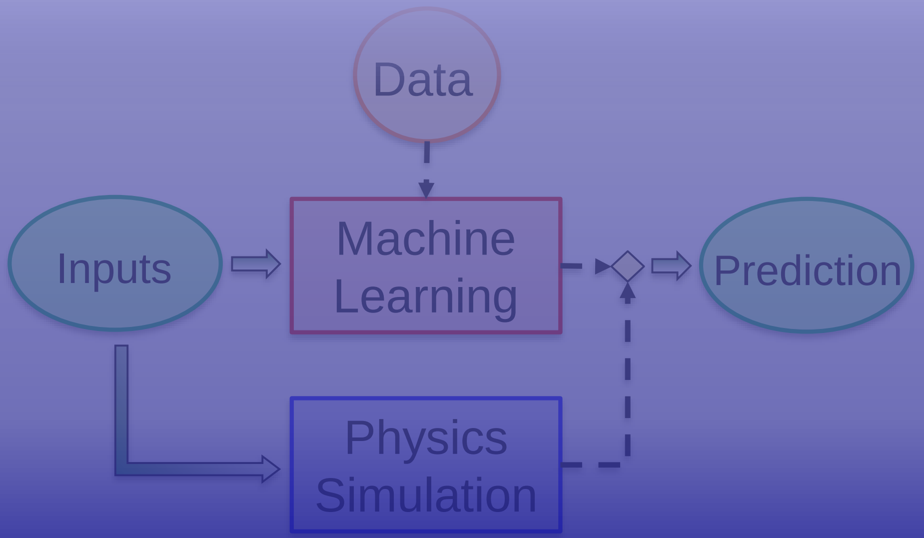
- Illustrated through both 1-D and 2-D material identification problems

- Noisy measurements adversely affect inference
  - …but can be mitigated by explicitly modeling measurement noise

Thank you!

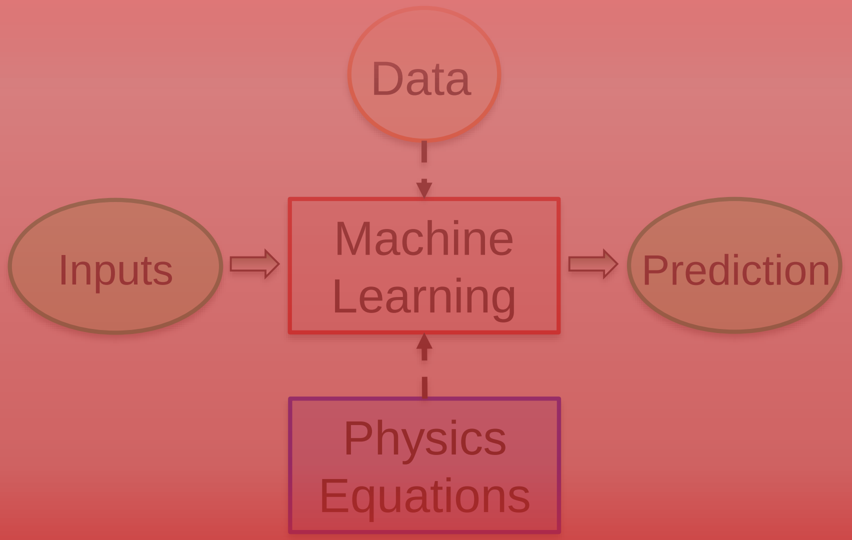
geoffrey.f.bomarito@nasa.gov

Part 1



Bomarito, Geoffrey F., et al. "On the Optimization of Approximate Control Variates with Parametrically Defined Estimators." arXiv preprint arXiv:2012.02750 (2020).

Part 2



Warner, James E., et al. "Inverse Estimation of Elastic Modulus Using Physics-Informed Generative Adversarial Networks." *arXiv:2006.05791* (2020).