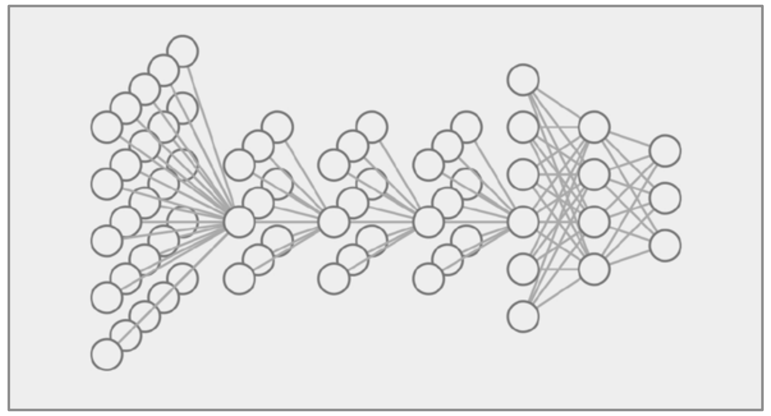
NVIDIA | Caltech

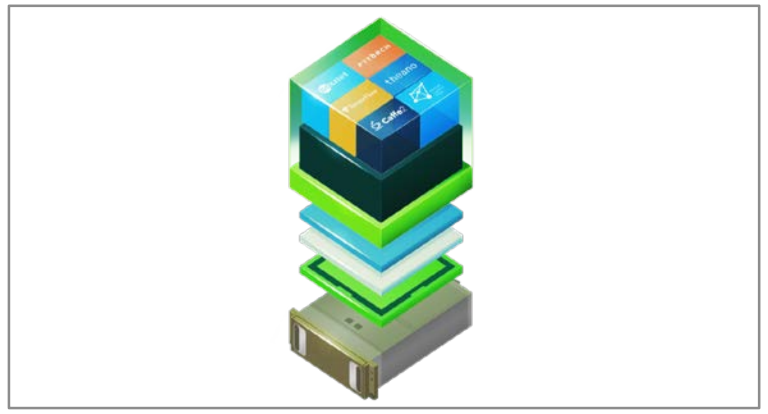
AI ALGORITHMS FOR MECHANICS

ANIMA ANANDKUMAR

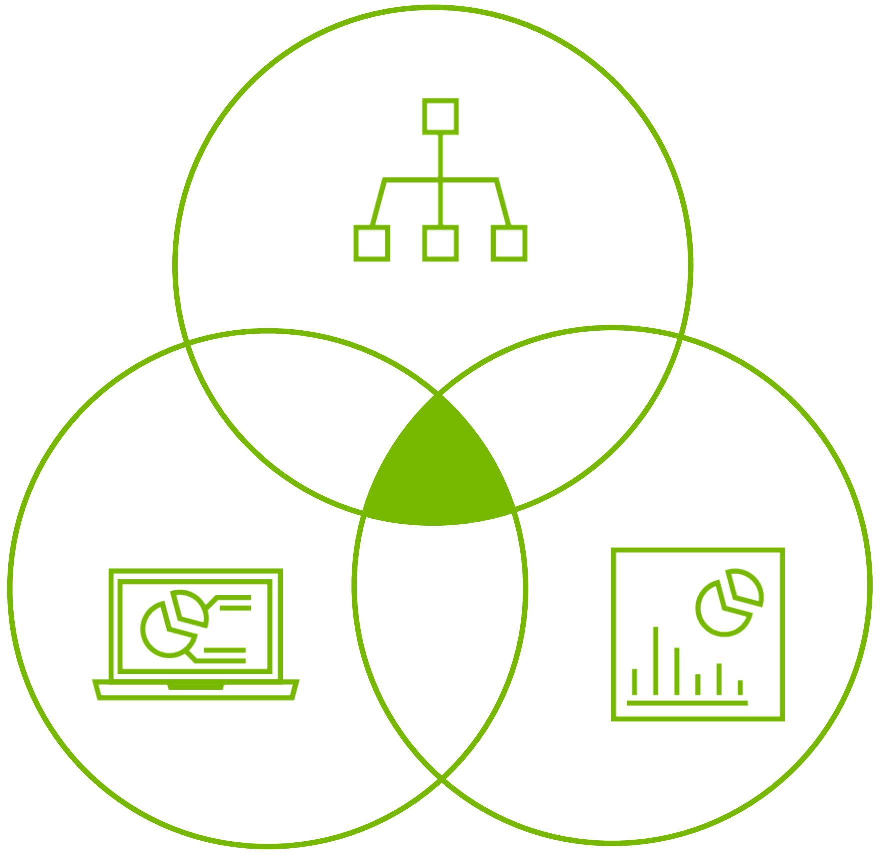
TRINITY OF AI



ALGORITHMS



COMPUTE



DATA

HARD CHALLENGES FOR AI

AI is Not Living Up to its Hype

Safety-critical Applications

STATE OF PROGRESS

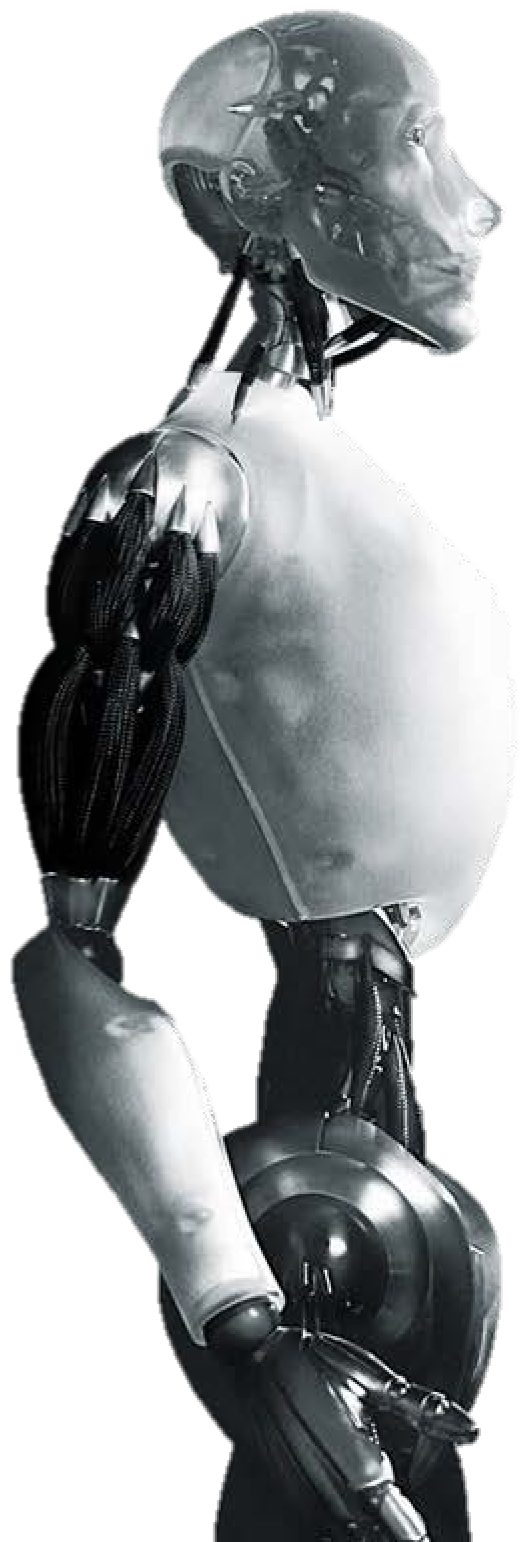
Most Robots Have No Intelligence

Boston Dynamics Atlas Robot

Fumbling Dog

EMBODIED AI

Mind + Body



Instinctive:
Fine-grained reactive Control

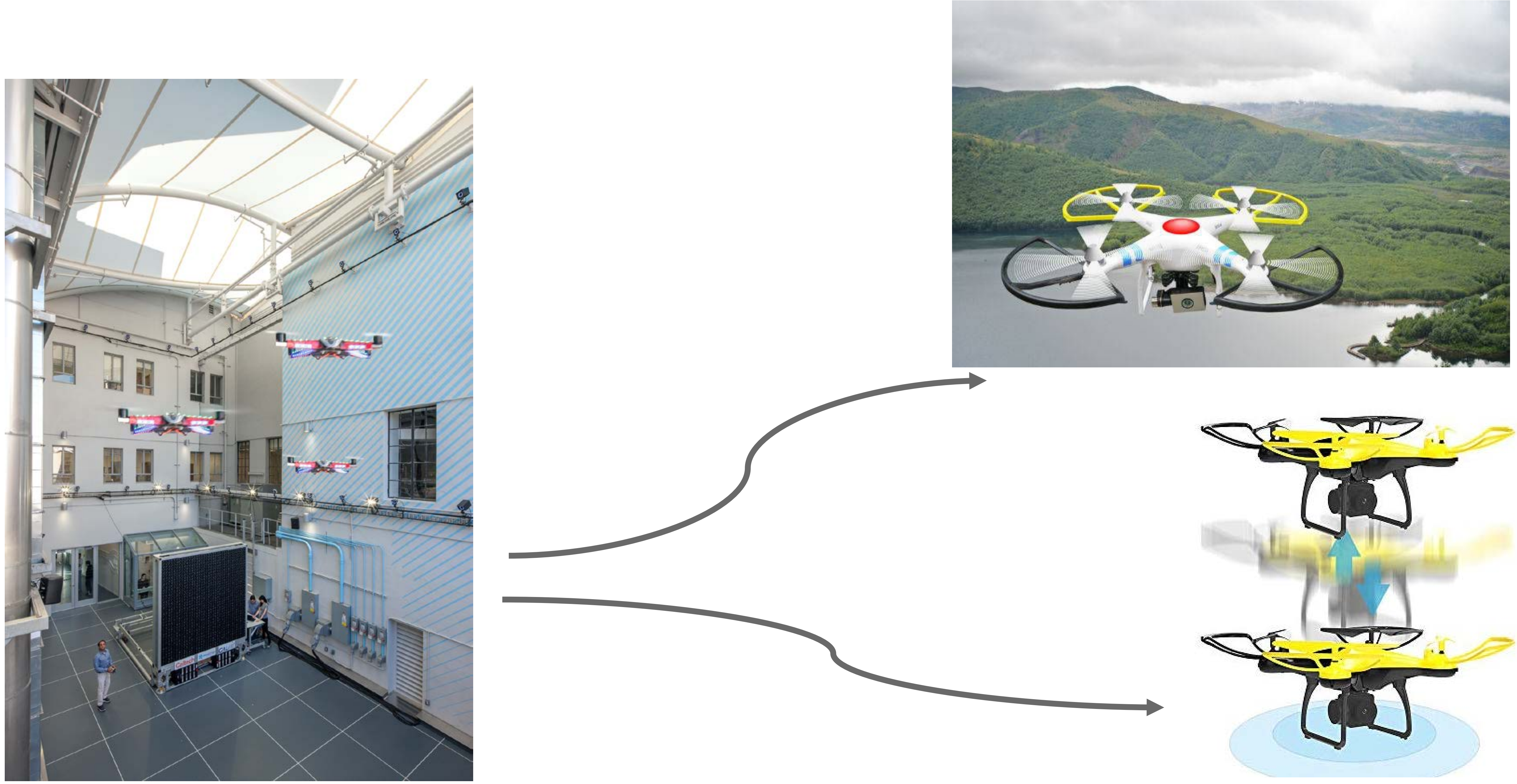
Deliberative:
Making and adapting plans

Safe:
Understand dangers

Multi-Agent:
Acting for the greater good

LEARNING IN CONTROL SYSTEM

CHALLENGE: DOMAIN SHIFT IN THE REAL WORLD

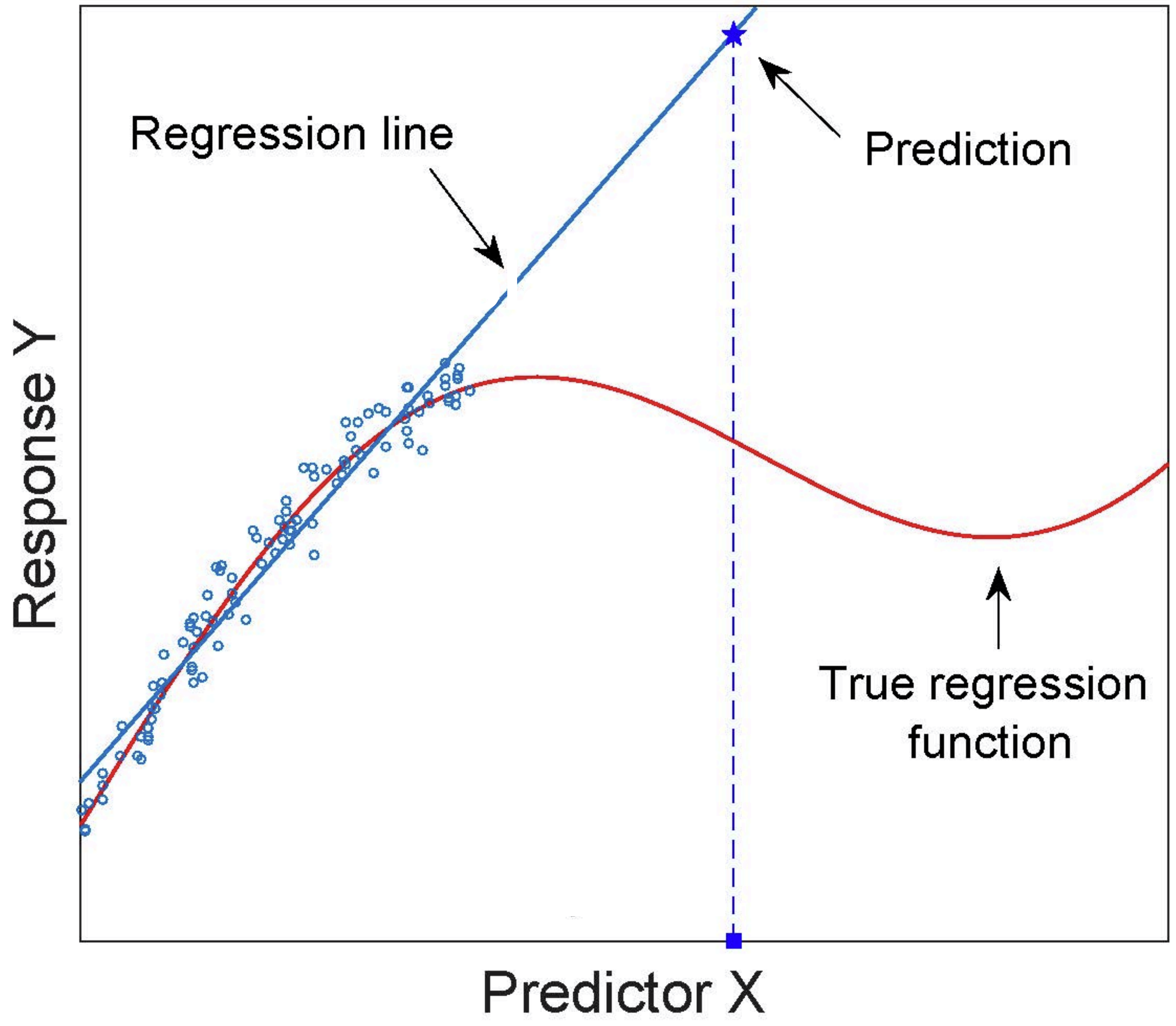


Training

Deployment

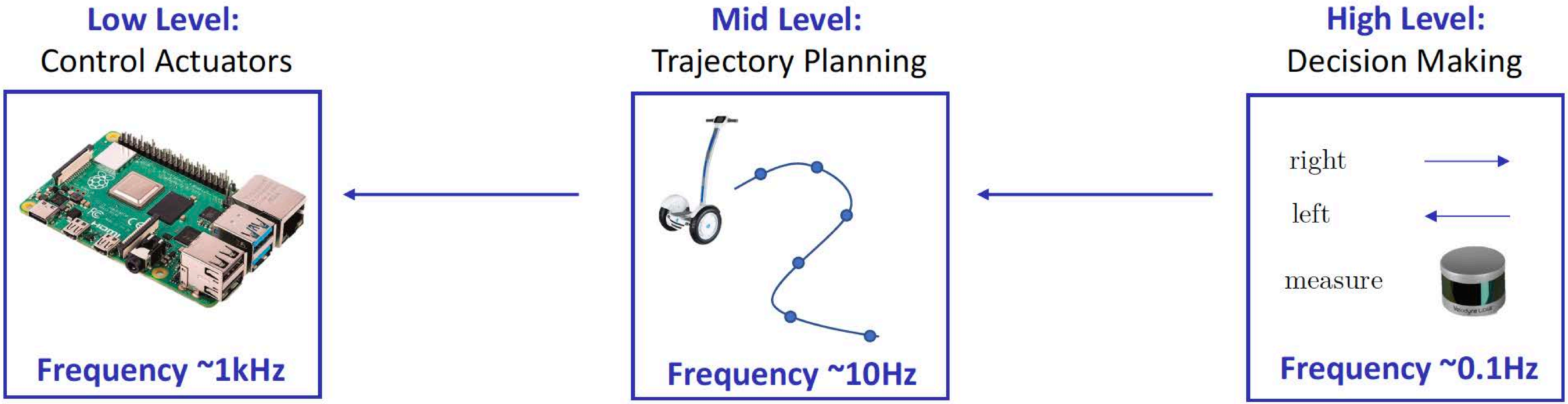
IGNORING DOMAIN SHIFT CAN BE RISKY

How to get good uncertainty estimates?



MODULARITY IN CONTROL SYSTEMS

Unlike deep learning, which tends to be monolithic



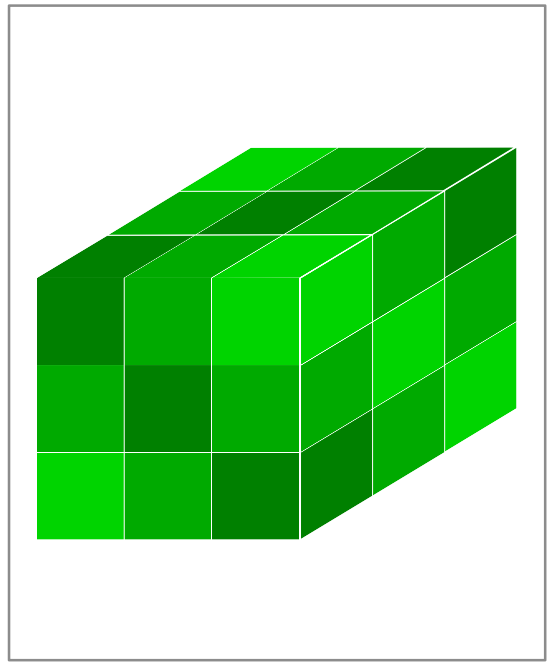
How to incorporate learning in each module while preserving safety and stability?

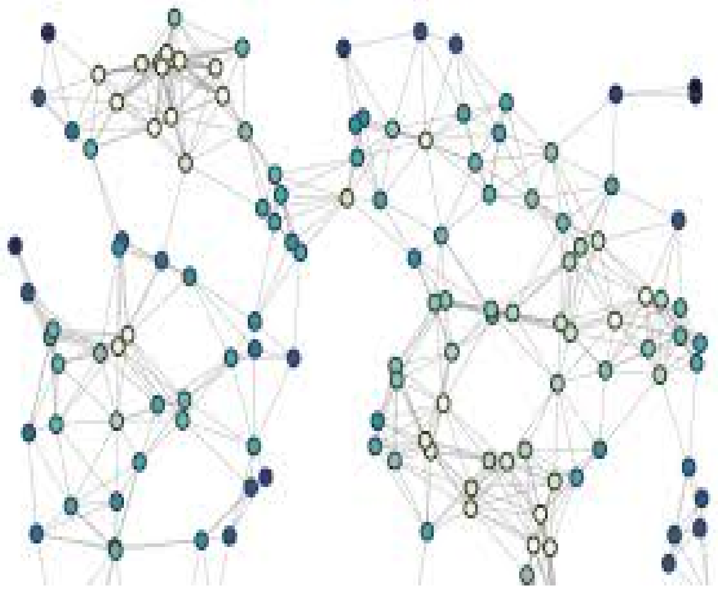
HOW TO USE STRUCTURE AND DOMAIN KNOWLEDGE TO DESIGN ROBUST PRIORS?

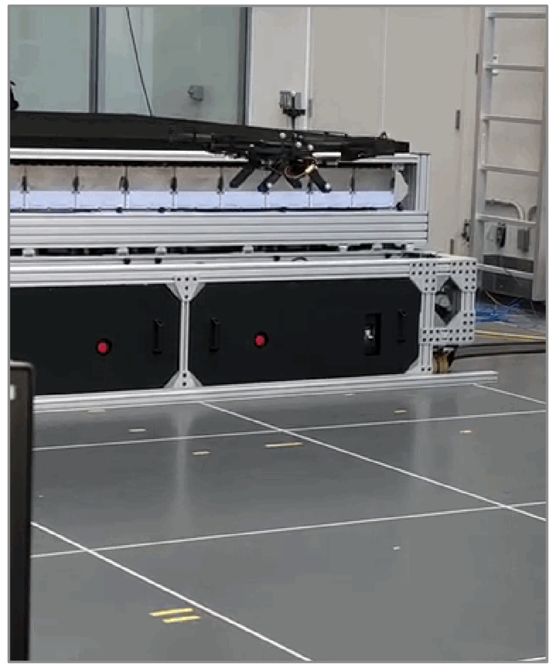
Learning = Data + Priors

Examples of Priors

- Tensors and graphs
- Laws of nature
- Simulations







Neural Lander: Stable Drone Landing Control using Learned Dynamics

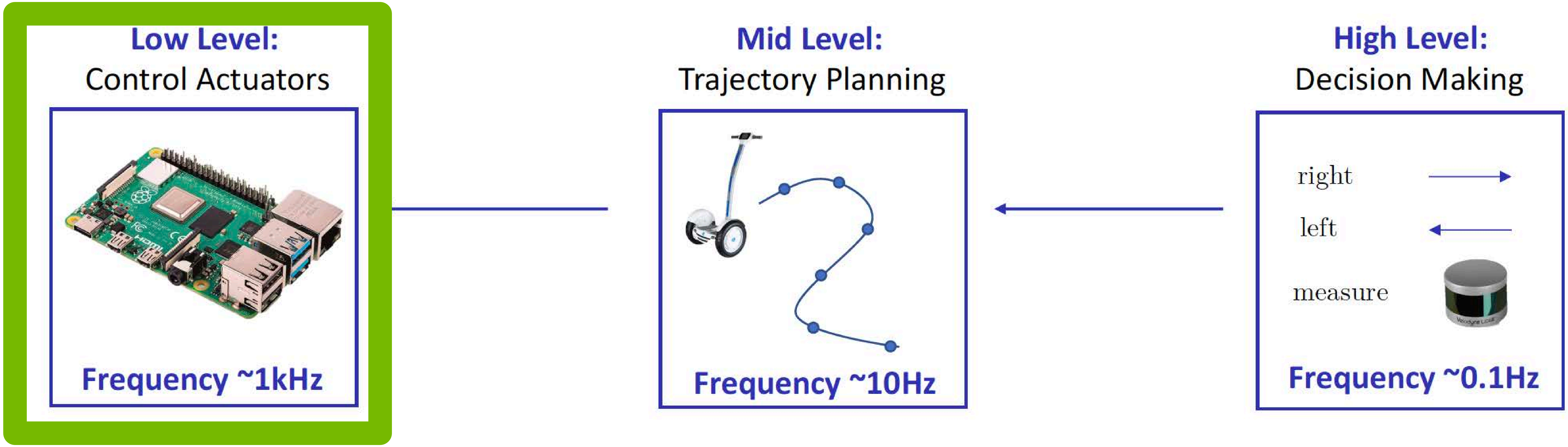







Guanya Shi
Xichen Shi
Michael O'Connell
Rose Yu
Kamyar Azizzadenesheli
Soon-Jo Chung
Yisong Yue

LEARNING IN LOW-LEVEL CONTROL



How to incorporate learning in each module while preserving safety and stability?

BASELINE: MODEL-BASED CONTROL

(NO LEARNING)

New State

Current Action (aka control input)

$$x_{t+1} = f(x_t, u_t) + \epsilon$$

Current State

Unmodeled Disturbance / Error

Robust Control (fancy contraction mappings)

- Stability guarantees (e.g., Lyapunov)
- Precision/optimality depends on error

LEARNING RESIDUAL DYNAMICS

New State

Current Action (aka control input)

$$x_{t+1} = f(x_t, u_t) + \tilde{f}(x_t, u_t) + \epsilon$$

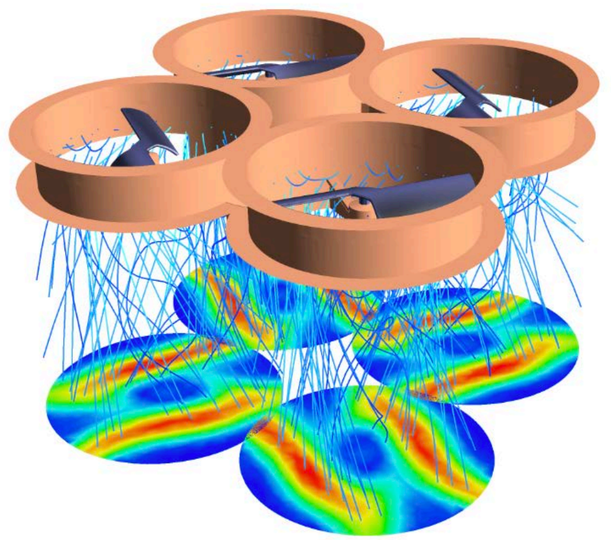
Current State

Unmodeled Disturbance

Use existing control methods to generate actions

- Provably robust (even using deep learning)
- Requires $\tilde{f}$ Lipschitz & bounded error

CONTROL SYSTEM FORMULATION



Learn the Residual
(function of state and control input)

- Dynamics:

$$\dot{\mathbf{p}} = \mathbf{v}, \qquad m\dot{\mathbf{v}} = m\mathbf{g} + R\mathbf{f}_u + \mathbf{f}_a$$
$$\dot{R} = RS(\boldsymbol{\omega}), \qquad J\dot{\boldsymbol{\omega}} = J\boldsymbol{\omega} \times \boldsymbol{\omega} + \boldsymbol{\tau}_u + \boldsymbol{\tau}_a$$

- Control:

$$\mathbf{f}_u = [0, 0, T]^\top$$
$$\boldsymbol{\tau}_u = [\tau_x, \tau_y, \tau_z]^\top$$

$$\begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & c_T l_{\text{arm}} & 0 & -c_T l_{\text{arm}} \\ -c_T l_{\text{arm}} & 0 & c_T l_{\text{arm}} & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} n_1^2 \\ n_2^2 \\ n_3^2 \\ n_4^2 \end{bmatrix}$$

- Unknown forces

$$\mathbf{f}_a = [f_{a,x}, f_{a,y}, f_{a,z}]^\top$$
$$\boldsymbol{\tau}_a = [\tau_{a,x}, \tau_{a,y}, \tau_{a,z}]^\top$$

Learn the Residual: Ground effect

CONTROLLER DESIGN

- Nonlinear Feedback Linearization:

$$u_{nominal} = K_s \eta \qquad \eta = \begin{bmatrix} p - p^* \\ v - v^* \end{bmatrix}$$

Desired Trajectory
(tracking error)

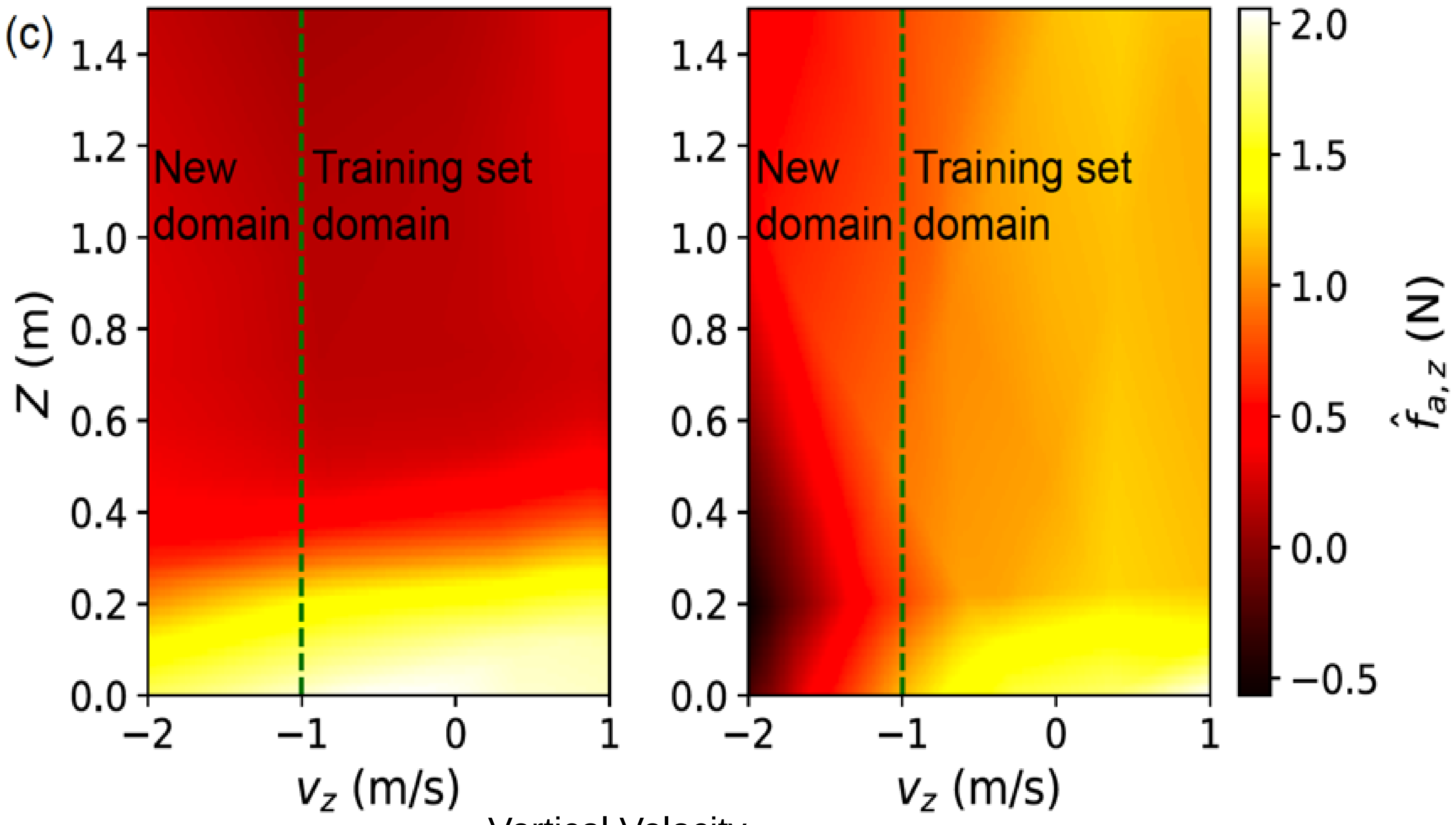
Feedback Linearization (PD control)

- Cancel out ground effect using learned model

$$u = u_{nominal} + u_{residual}$$

GENERALIZATION PERFORMANCE

Spectral Regularized NN　　　　Conventional NN



Vertical Velocity

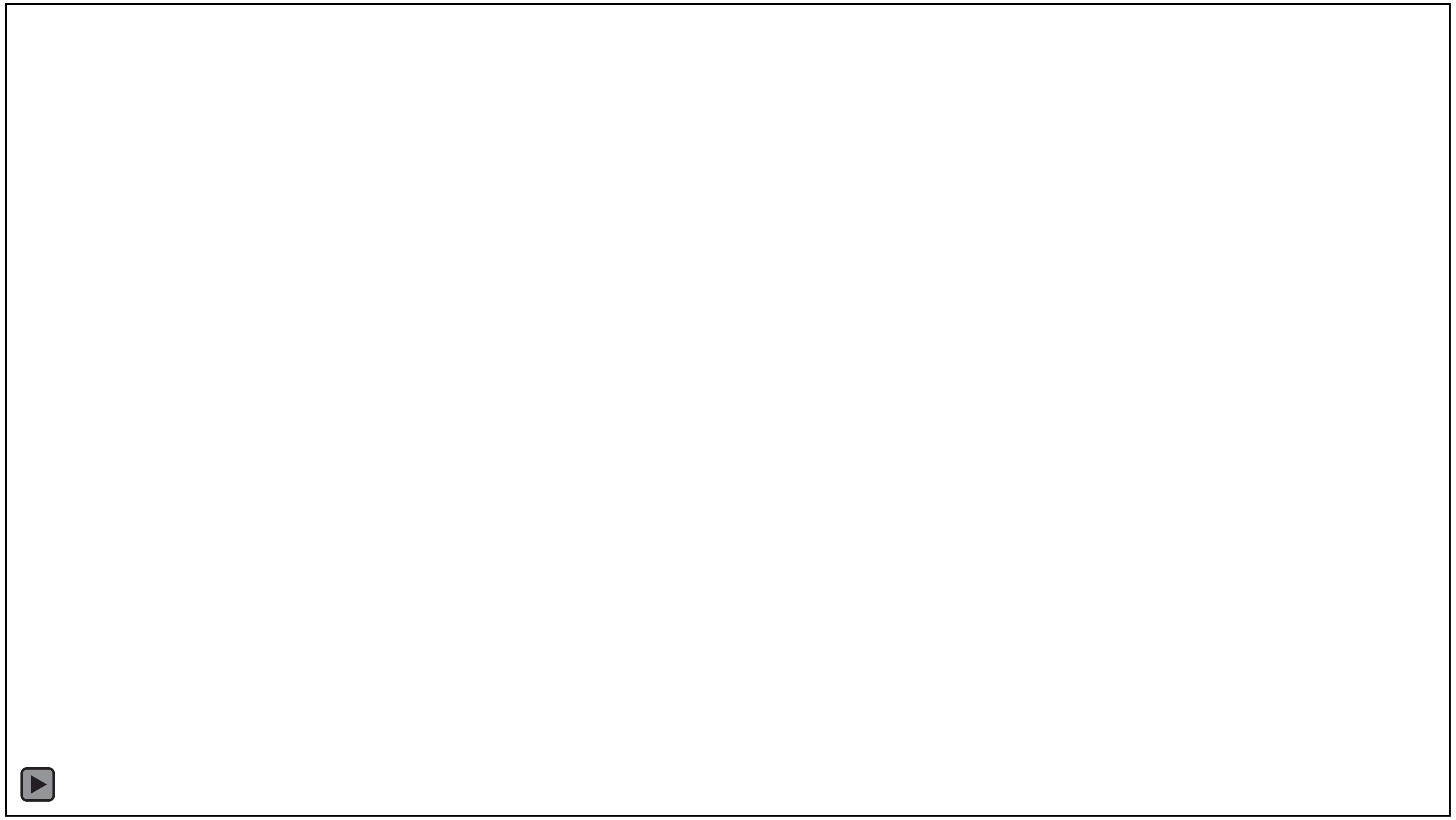
Spectral Normalization of each layer in neural network: Ensures $\tilde{f}$ is Lipshitz

<u>Guarantees stability under domain shifts</u>

LEARNING TO LAND

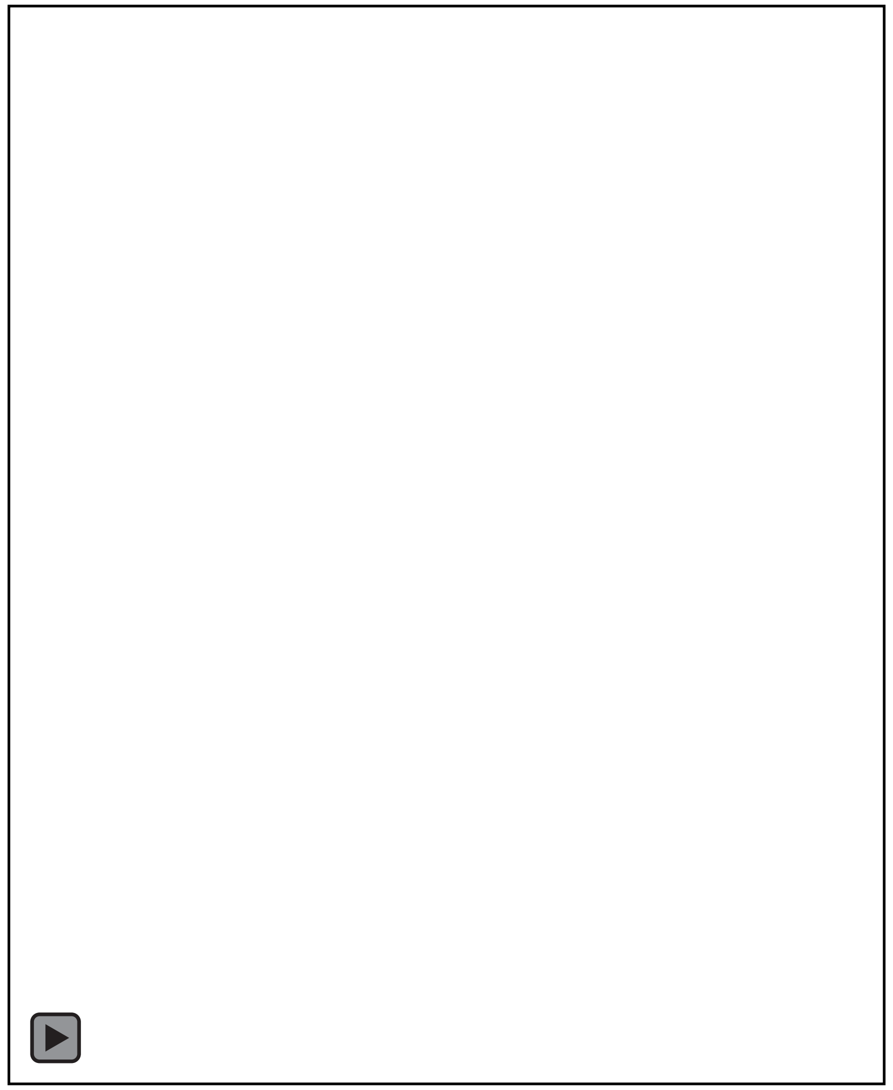
TESTING TRAJECTORY TRACKING

Move around a circle super close to the ground

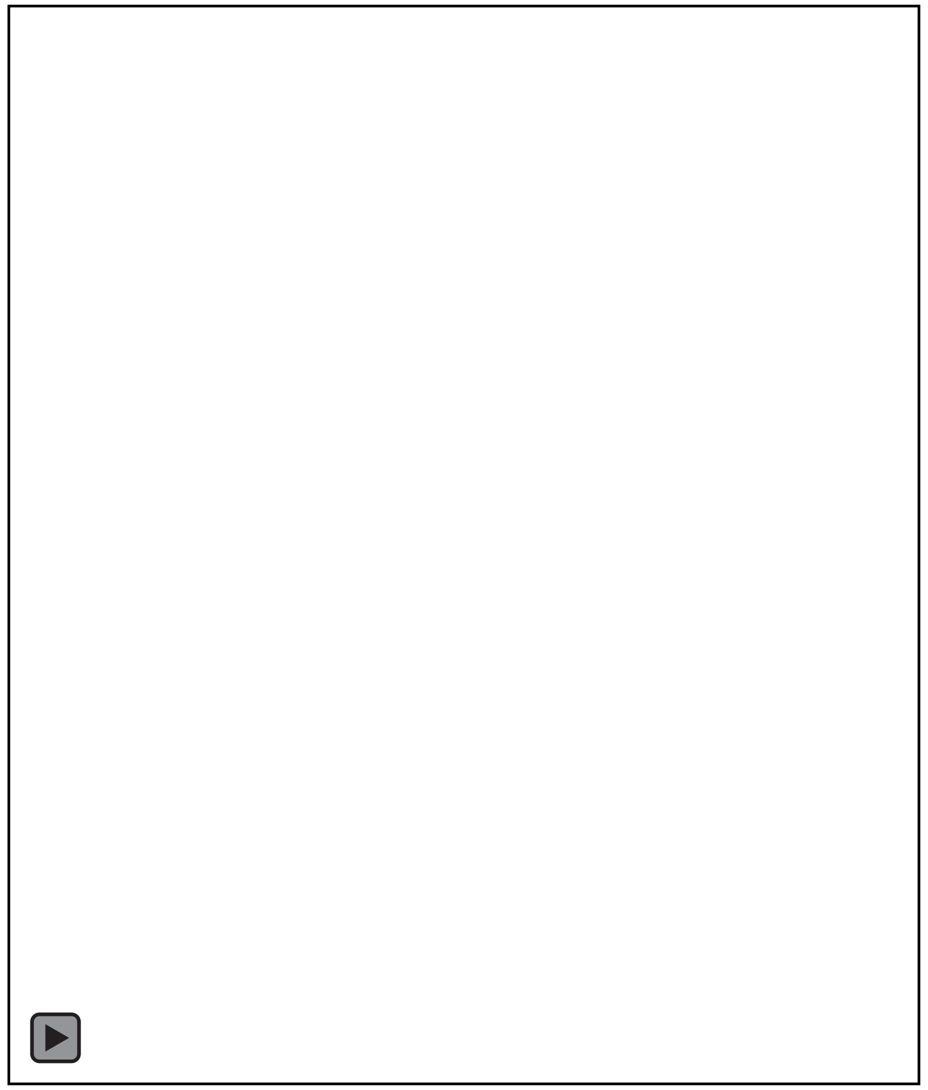


GOING BEYOND MANUAL DATA COLLECTION

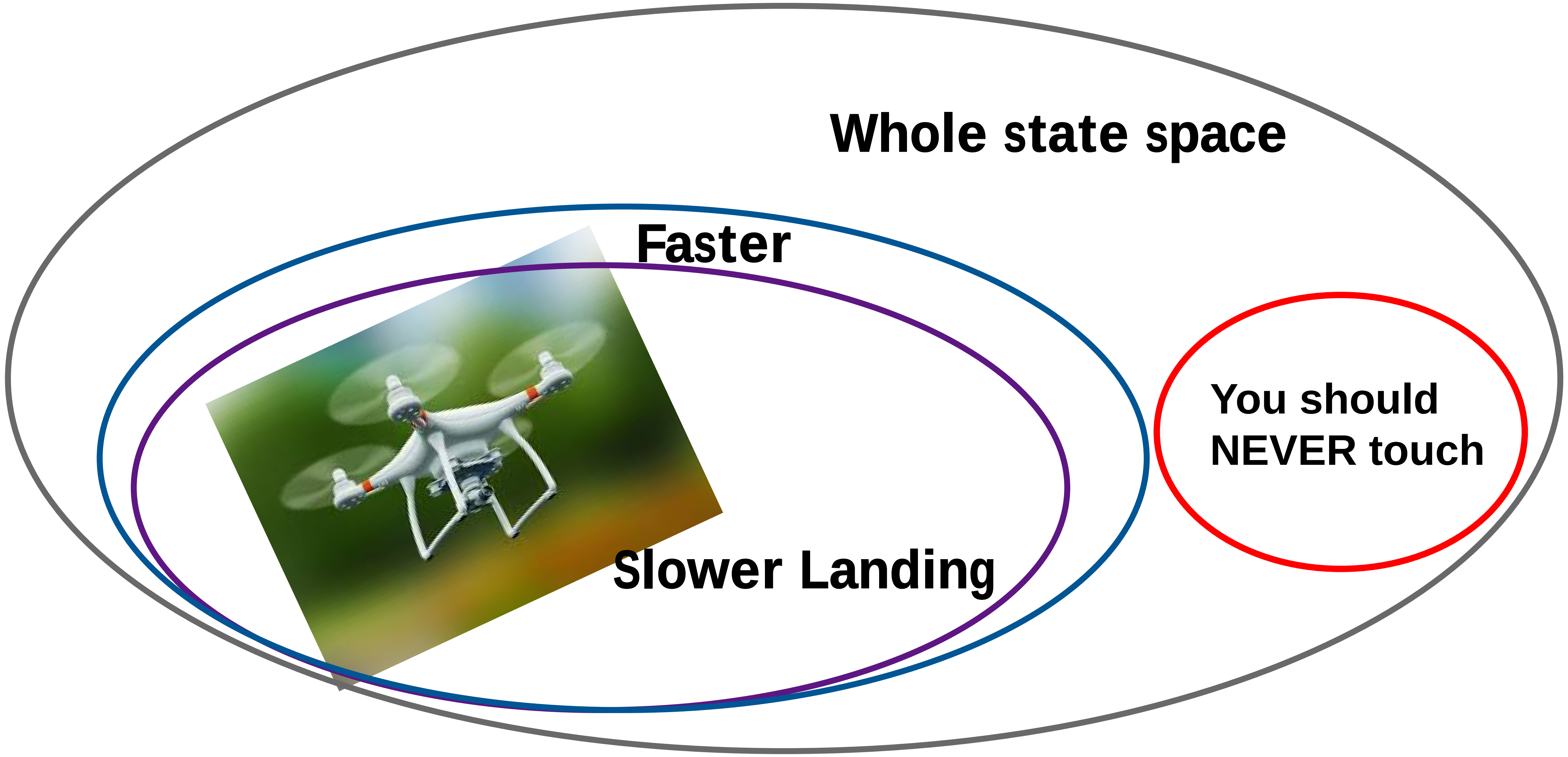
smooth landing



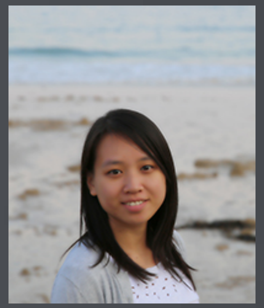
aggressive landing



CAN MACHINE EXPLORE STATE SPACE ITSELF?



Safe Exploration in Control Systems

Anqi Liu

Guanya Shi

Soon-Jo Chung

Yisong Yue

L4DC 2020

RECIPE FOR SAFE EXPLORATION

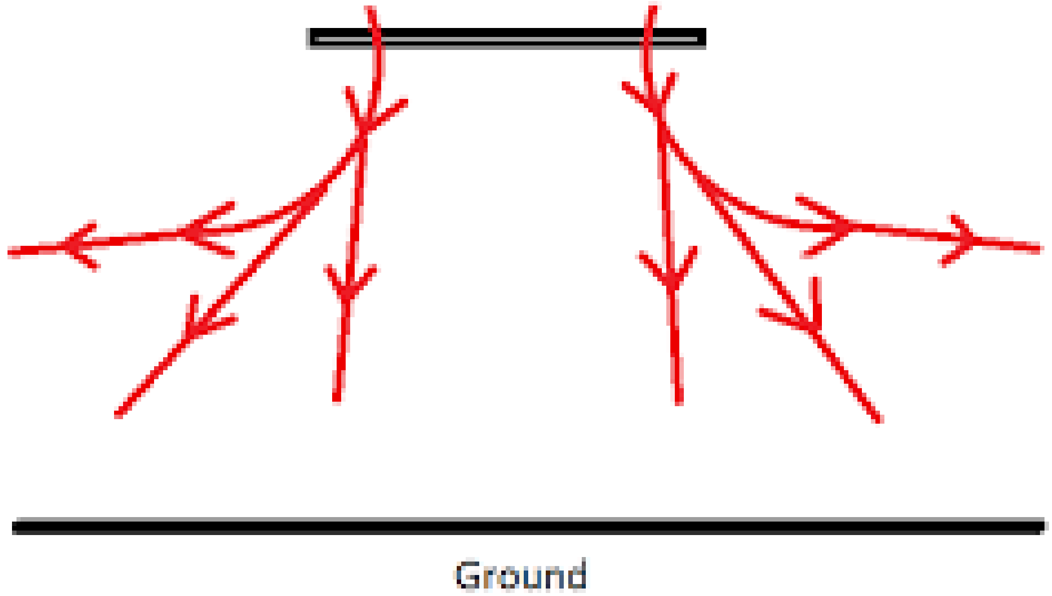
Given a pool of trajectories and initial model

Repeat:

- Evaluate model on the pool
- Check safety using uncertainty bounds
- Query safe samples that minimize landing time
- Retrain model

COVARIATE SHIFT IN DRONE DYNAMICS

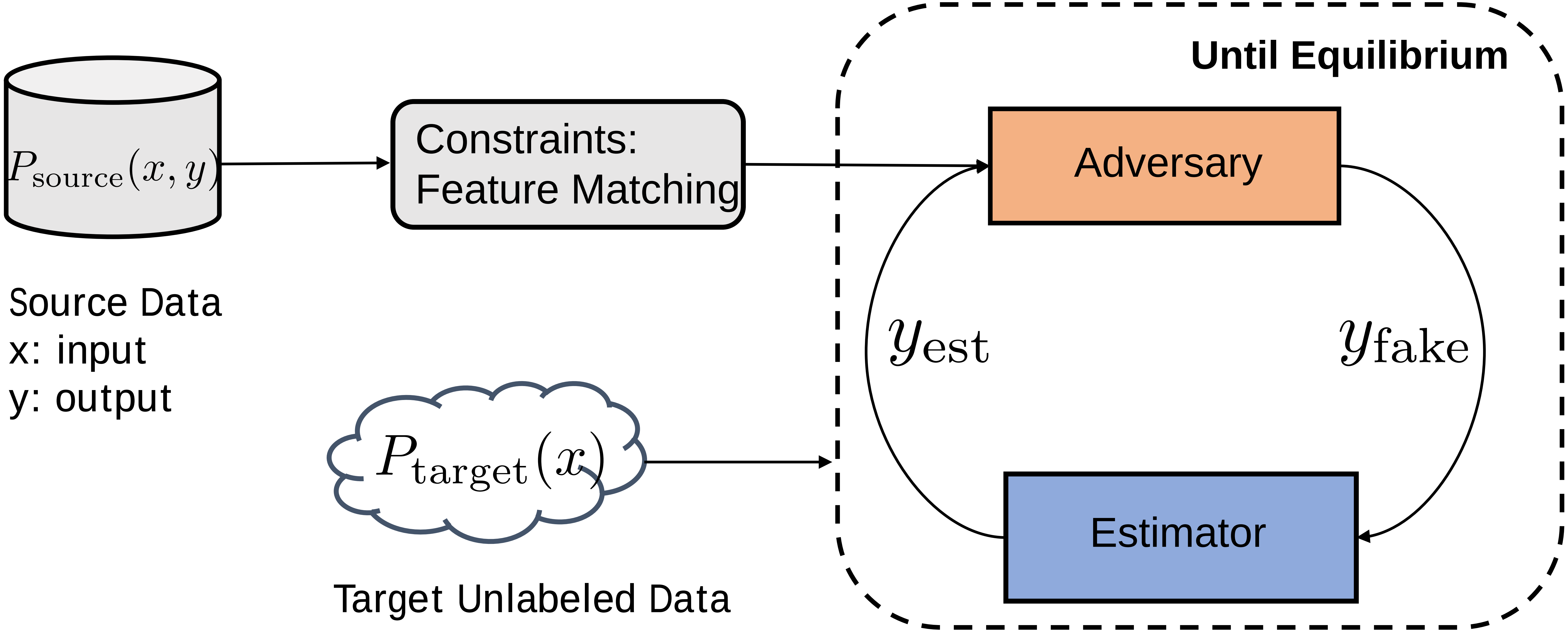
(position, velocity, orientation ....)



(ground effect)

- Ground truth physics model is unchanged
- Input data distribution changes

ADVERSARIAL LEARNING UNDER COVARIATE SHIFT

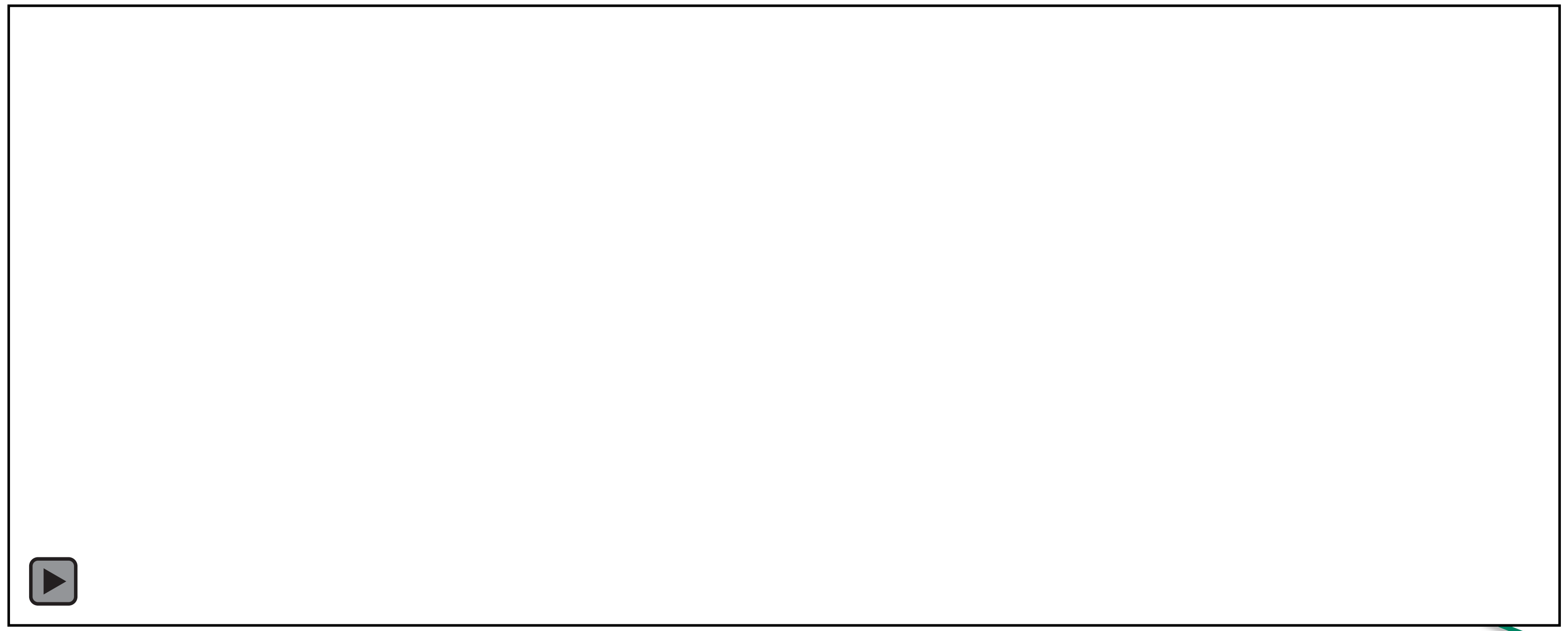


Guaranteed learning and tracking uncertainty bounds

LANDING TASK

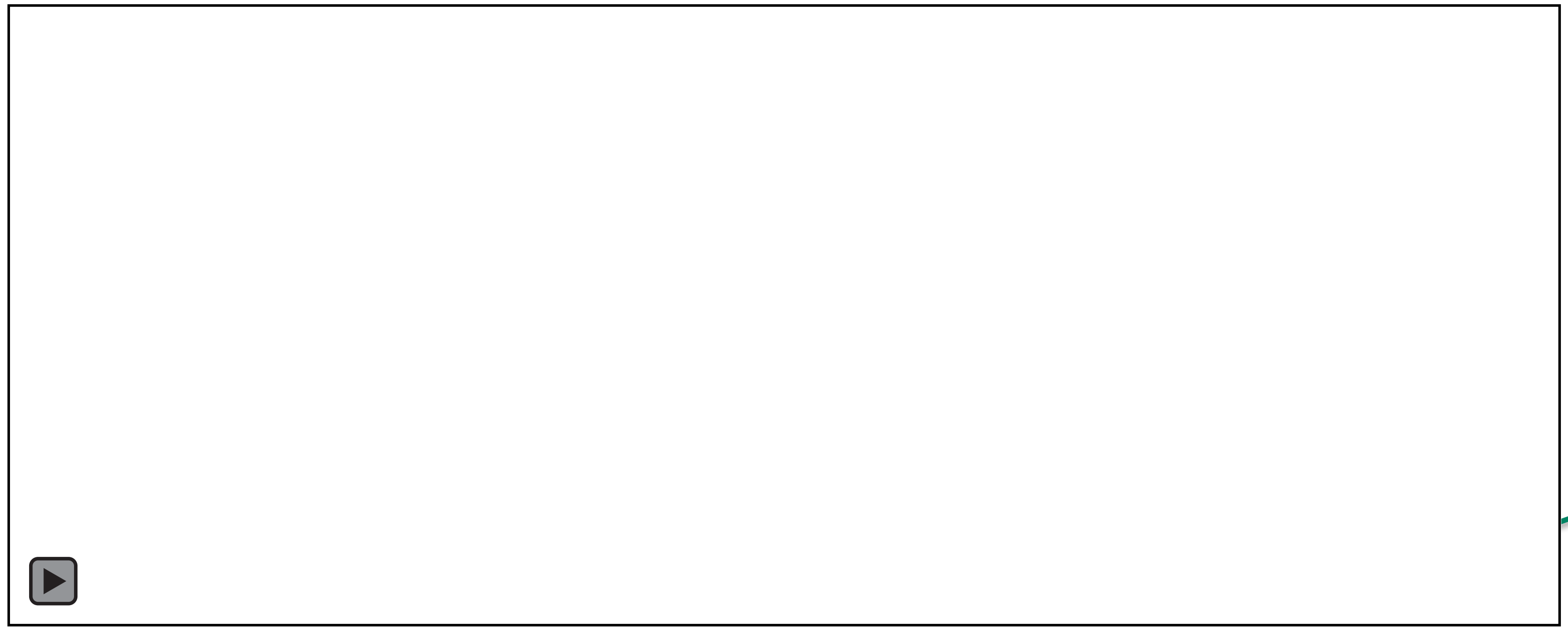
Ground effect measurement from real flying data

Epoch 1
(slow landing)
Cannot land



Unsafe region

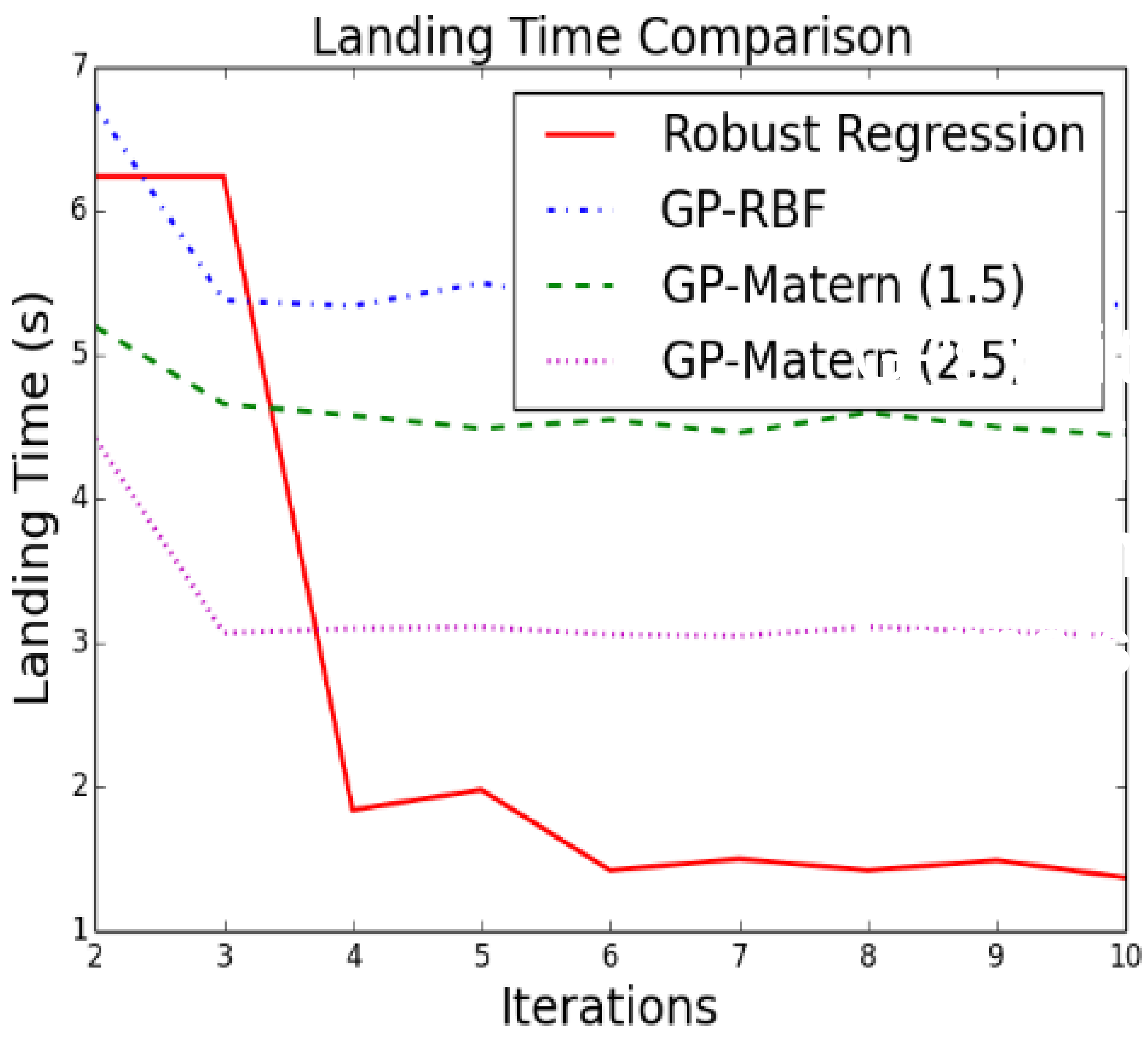
Epoch 10
(fastest landing)
Time: 1.6s



We converge to fastest landing (in our trajectory pool) in 10-15 epochs.

COMPARISON WITH GAUSSIAN PROCESS

Challenging for GPs: multiple dimension outputs



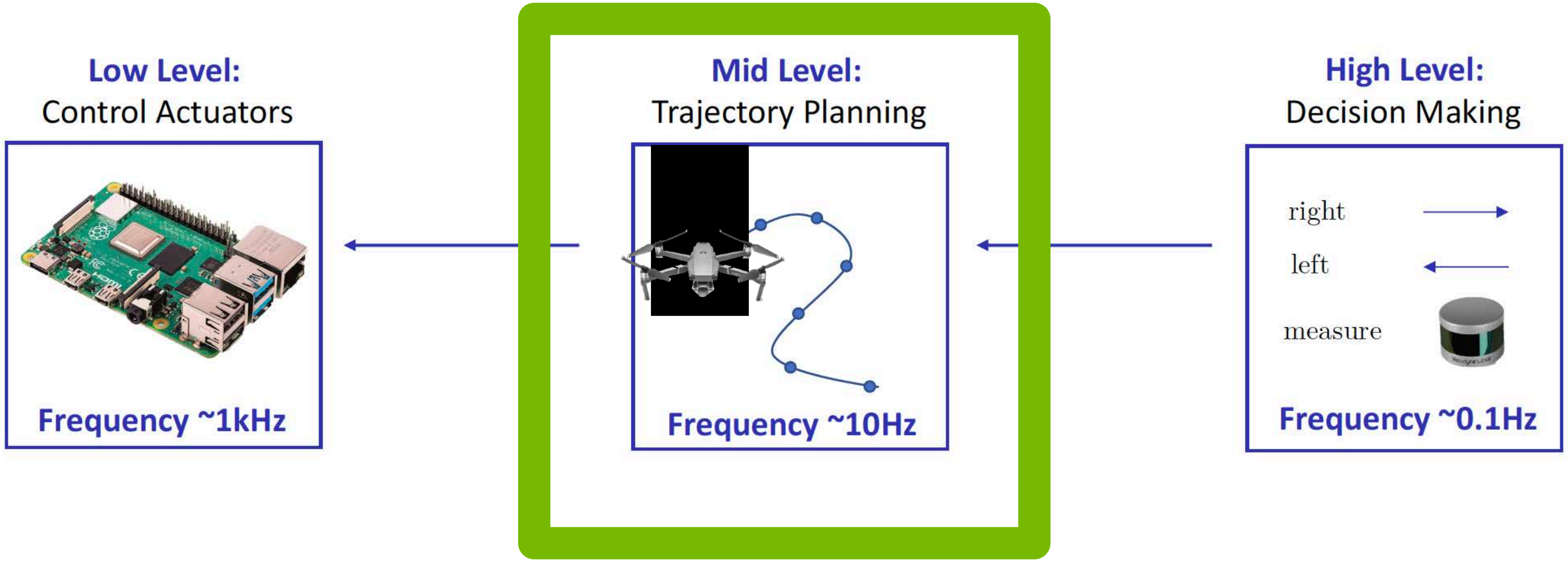
Information-Cost Stochastic Nonlinear Optimal Control For Safe Exploration

Yashwanth Nakka | Anqi Liu | Guanya Shi | Soon-Jo Chung | Yisong Yue

EFFECT OF UNCERTAINTY ON MID-LEVEL CONTROL

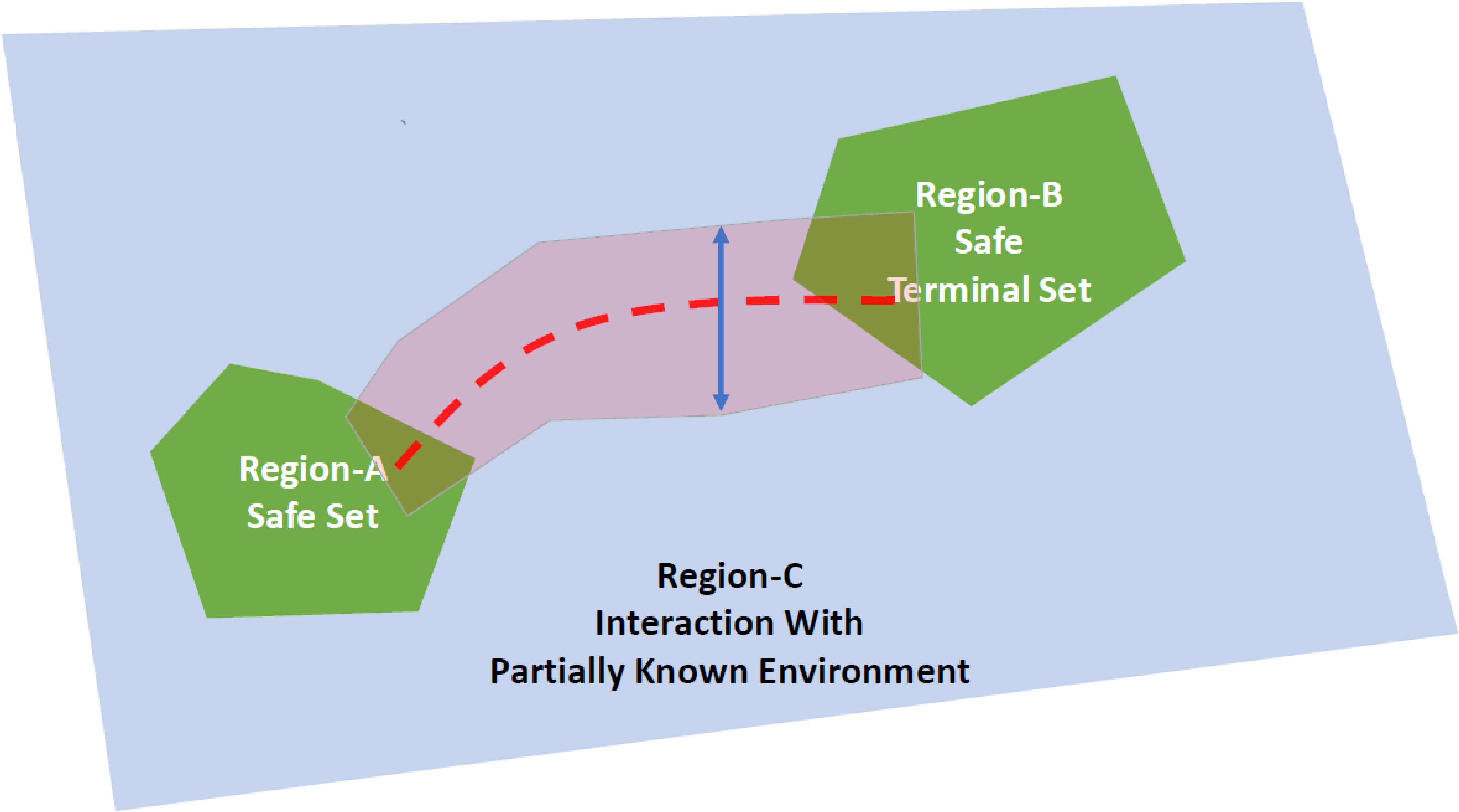
Trajectory planning under uncertainty

- Uncertainty can arise in each layer and learning is needed.
- How to keep guarantees end to end?

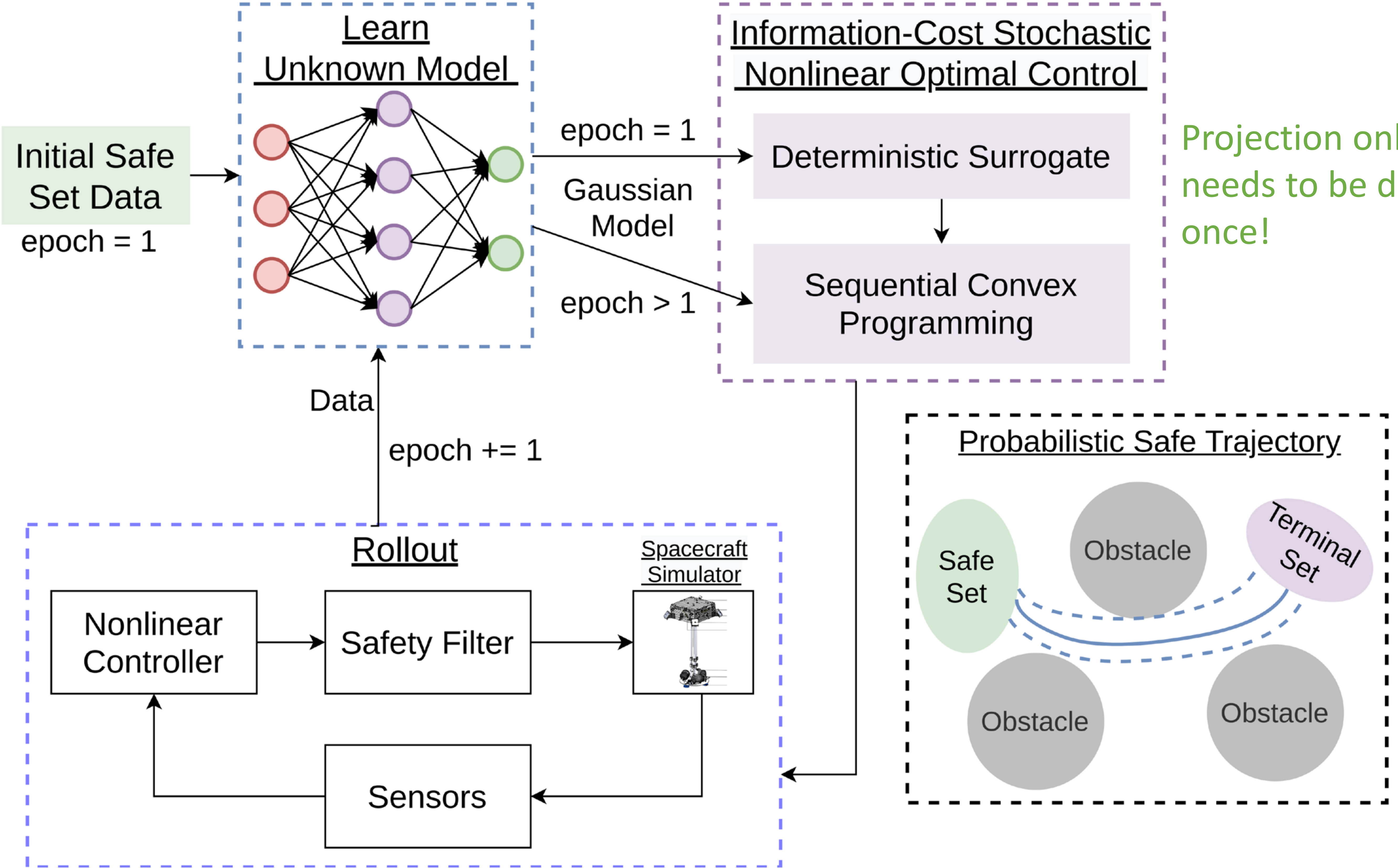


OUR GOAL

Design an **Informative and Safe Trajectory**
With a Given risk of Constraint Violation For Safe Exploration



INFORMATION-COST STOCHASTIC NONLINEAR OPTIMAL CONTROL

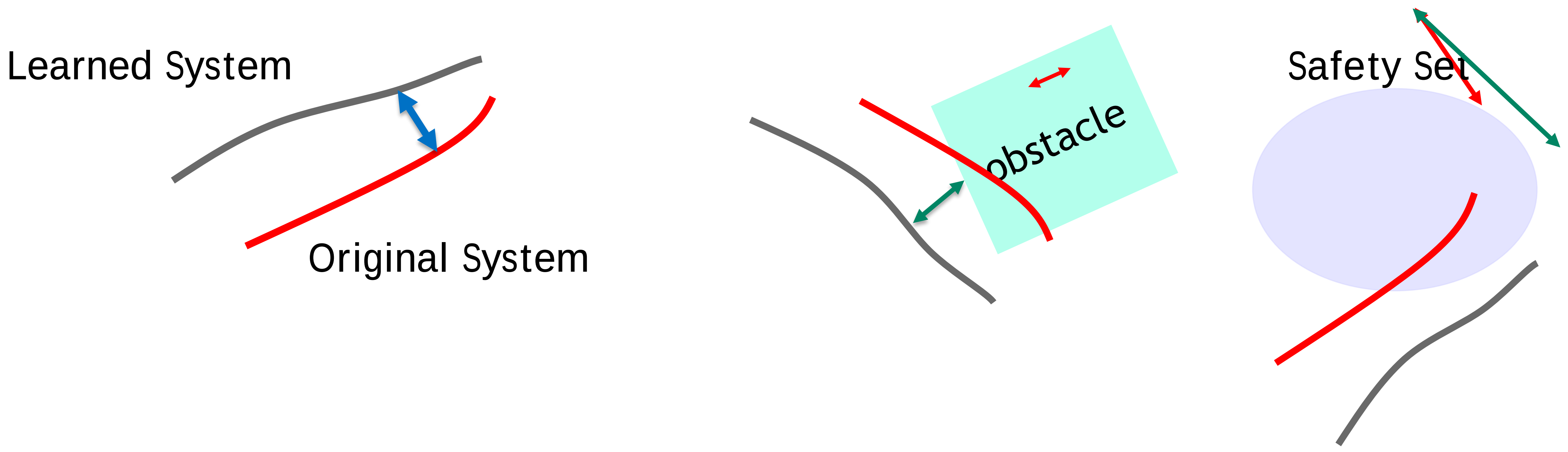


Projection only needs to be done once!

ROBUST LEARNING AND PLANNING

Uncertainty propagation

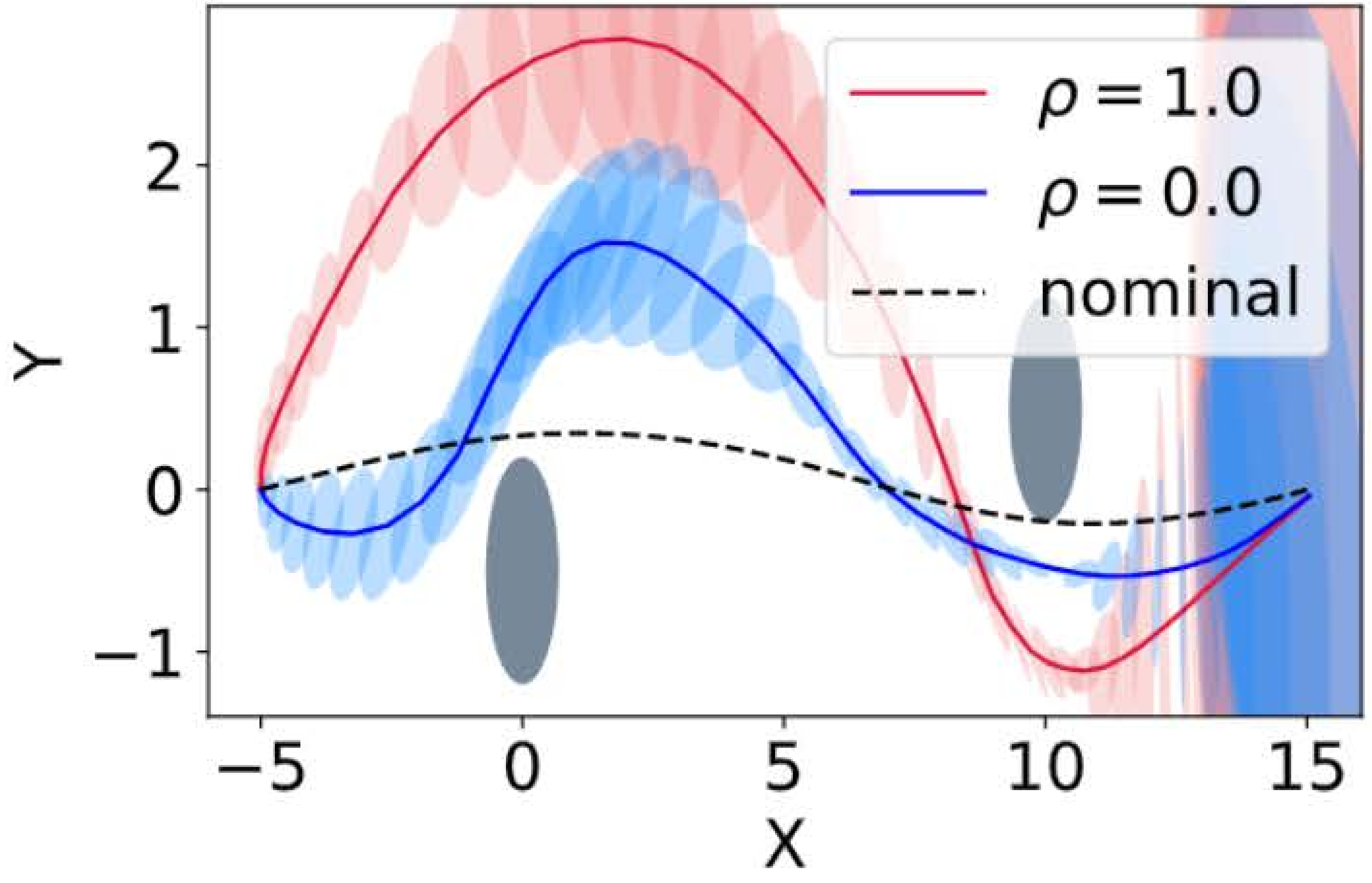
Bounded and Decreasing Variance in Planning → Convergence in Learning → Safety in Rollout with Feedback Control



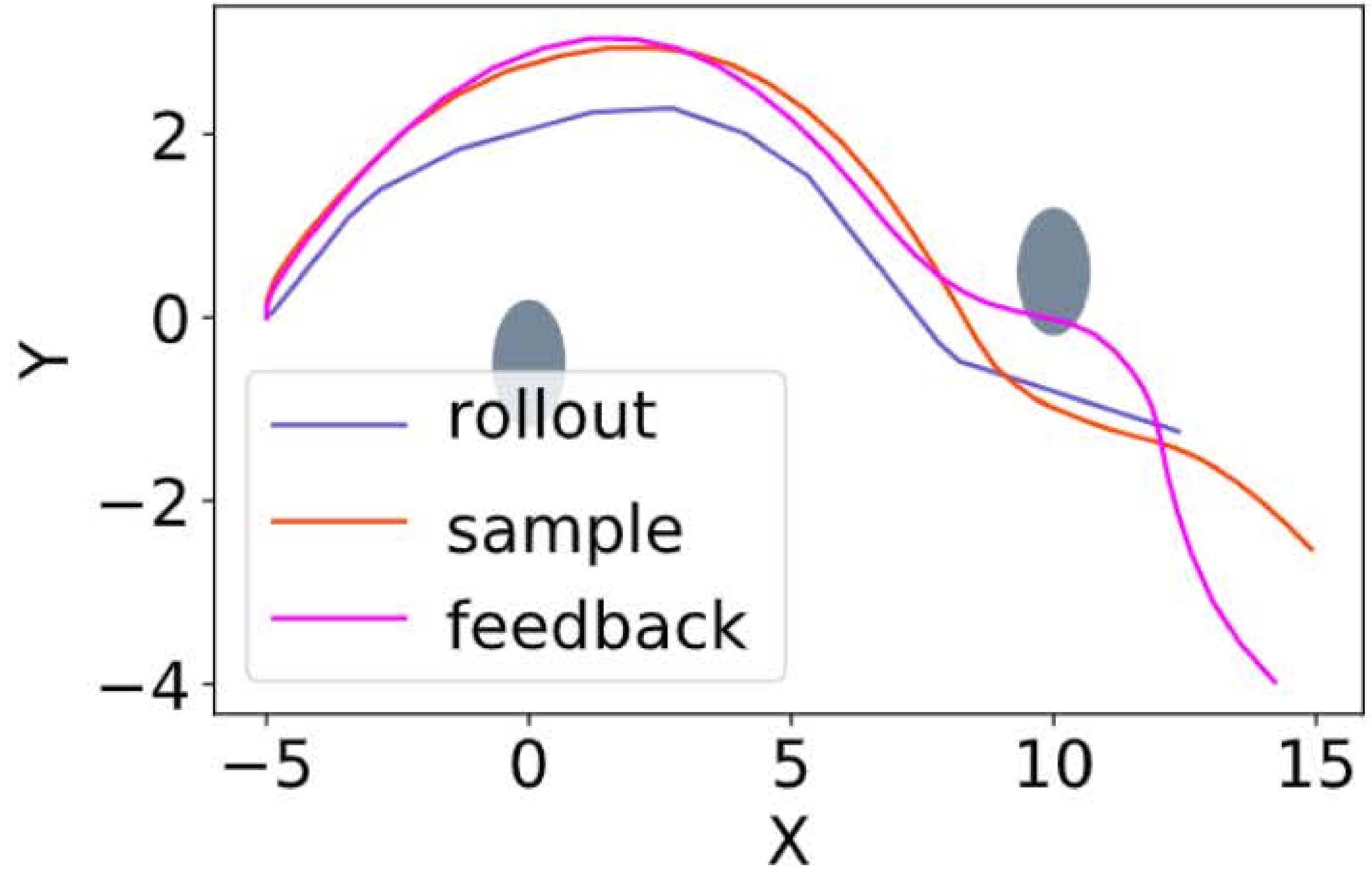
SPACECRAFT ASSEMBLY AND SIMULATION FACILITY

EXAMPLE: SPACECRAFT SIMULATORS

Info-SNOC
Probabilistic Safe and Informative Trajectory



Rollout
With and without Safety Filter at Epoch = 1



Our approach has 30% higher success rate than standard planning algorithm

ρ adjust the "importance" of information cost and performance cost
ρ = 0, only fuel cost; ρ = 1, only information cost

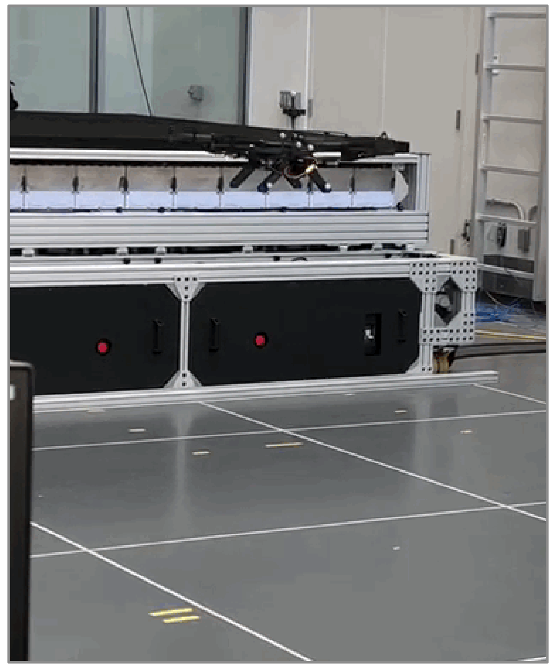
NEXT STEPS: WIND CONDITIONS

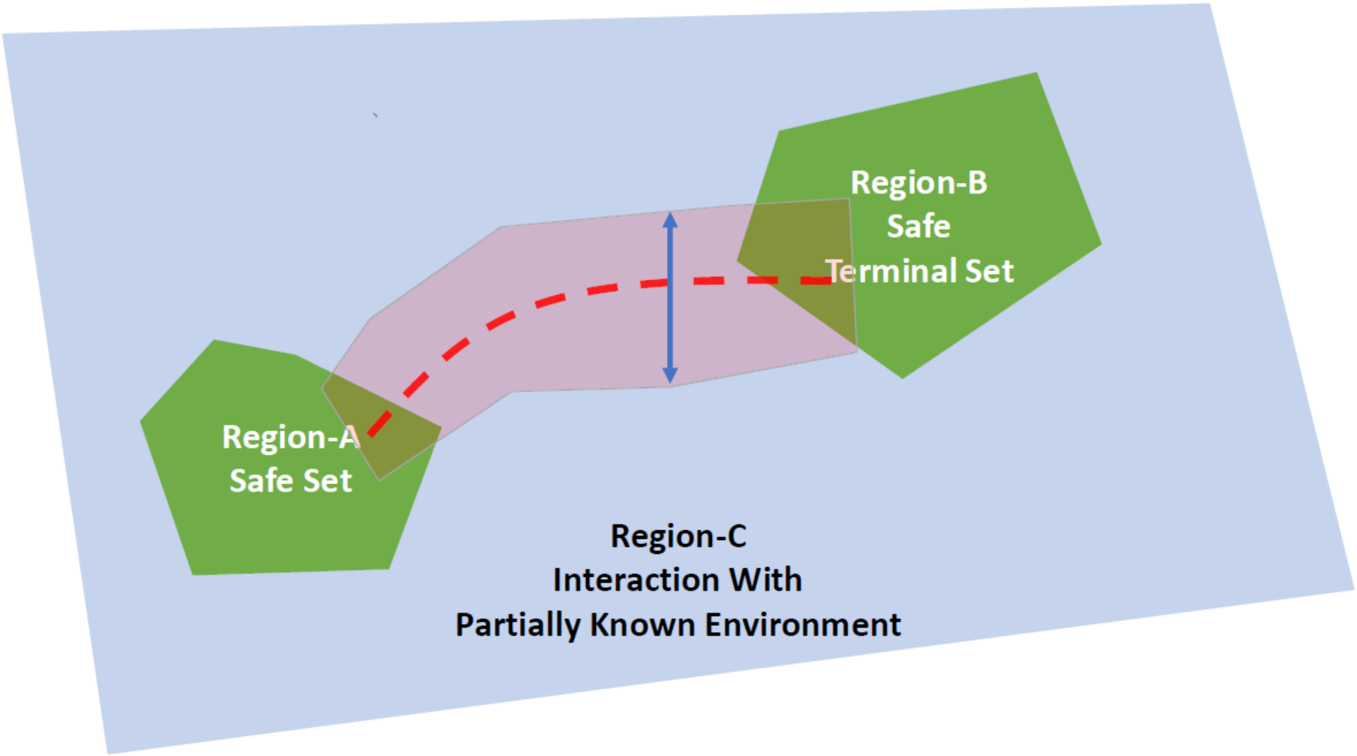
ROBUST META LEARNING

SUMMARY

Robust learning for control systems

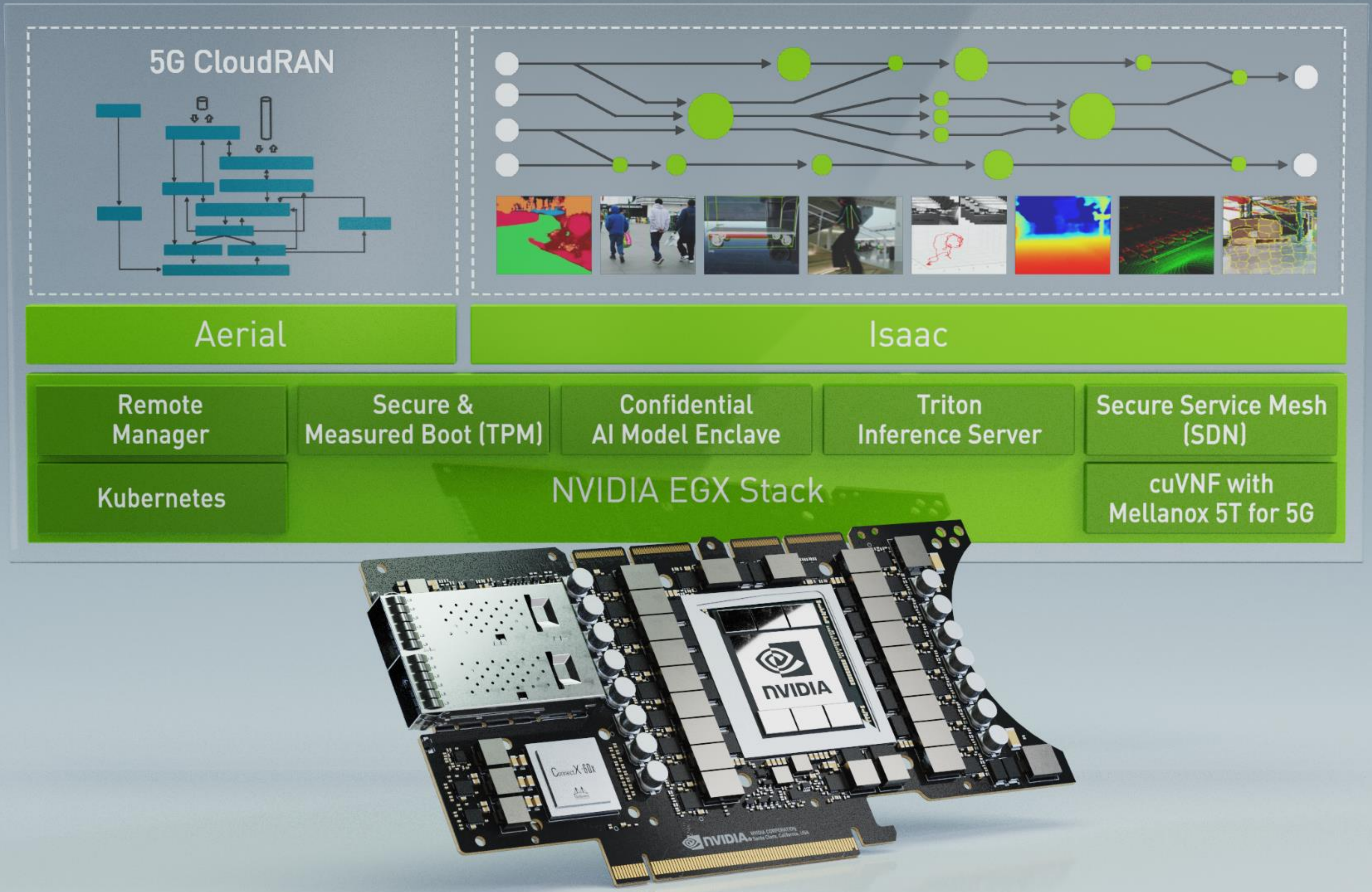
- Covariate Shift is inherent in learning in control systems.
- Adversarial estimation accounts for worst-case model.
- Uncertainty propagates from learning to planning + control.
- To encourage exploration, need to plan for more informative data collection.
- To guarantee safety, we need obtain end-to-end bounds.
- We validate our method in real-world systems.





NVIDIA ISAAC — PLATFORM FOR ROBOT LEARNING

Actual Factory



Virtual Factory
Digital Twin

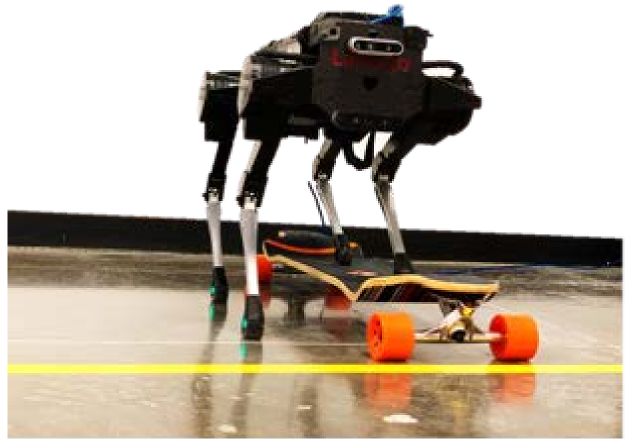
HIERARCHICAL REINFORCEMENT LEARNING ON ISAAC

Simulations for Robot Learning

ROBOT LEARNING ON NVIDIA ISAAC

Sim-to-Real in NVIDIA Isaac

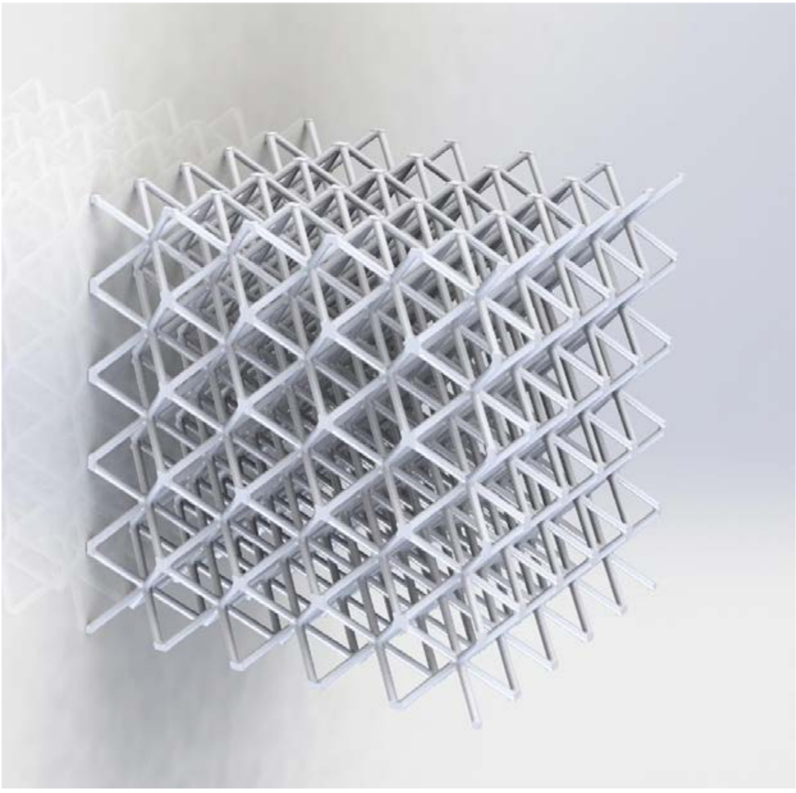
- NVIDIA Isaac provides platform for robot learning: physically valid simulations and reinforcement learning
- GPU acceleration for simulations and reinforcement learning
- Modular/hierarchical learning is essential for adaptivity to different tasks and environments
- Domain knowledge in form of existing controllers is essential
- Sim-to-real algorithms are needed for efficient domain adaptation

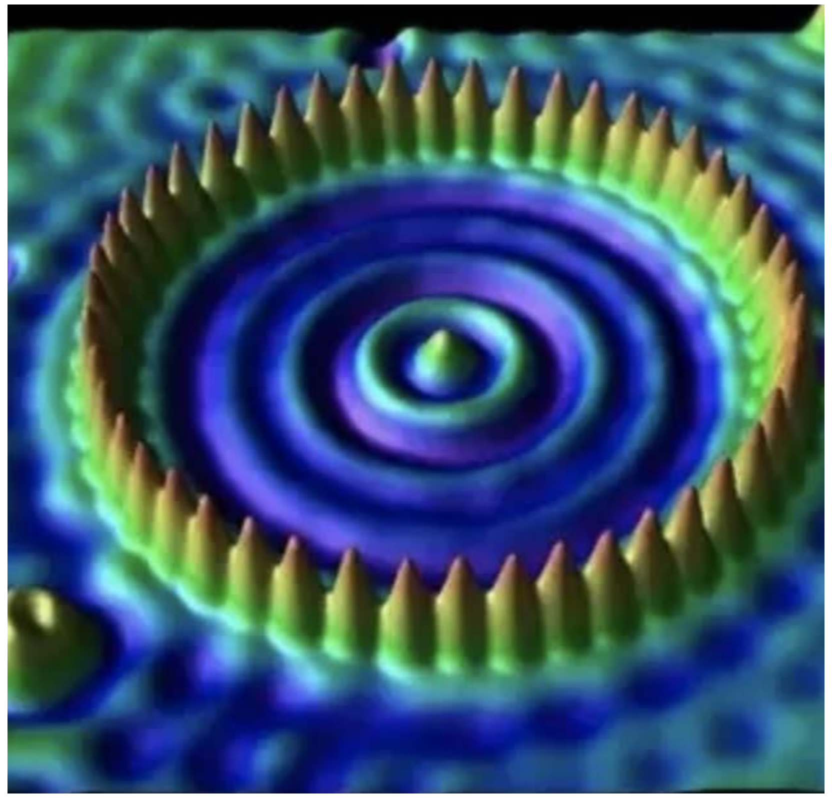


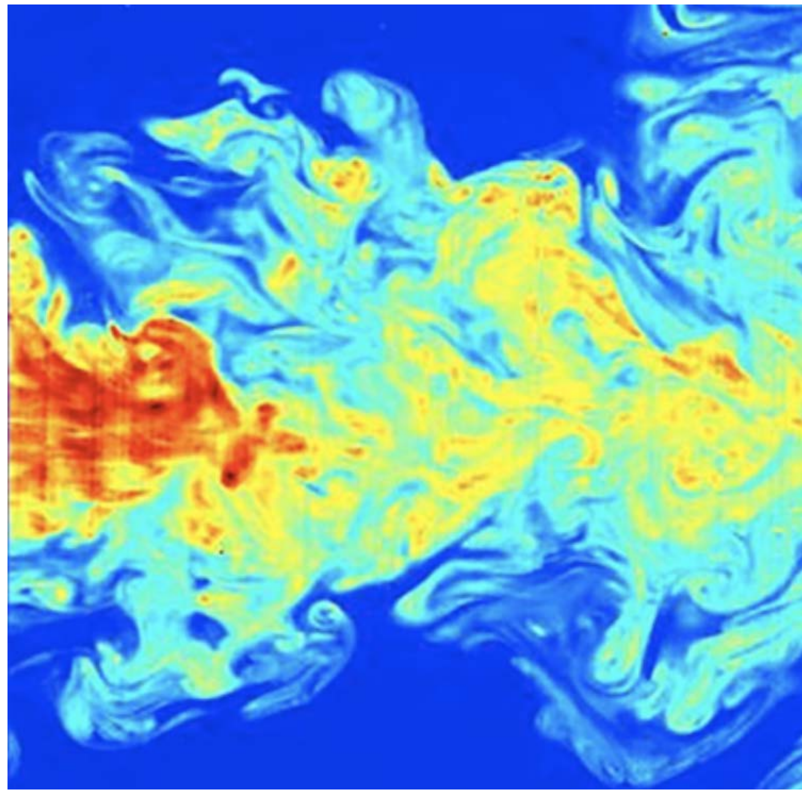
LEARNING-BASED PDE SOLVERS

PDE AS FOUNDATION FOR SCIENTIFIC MODELING

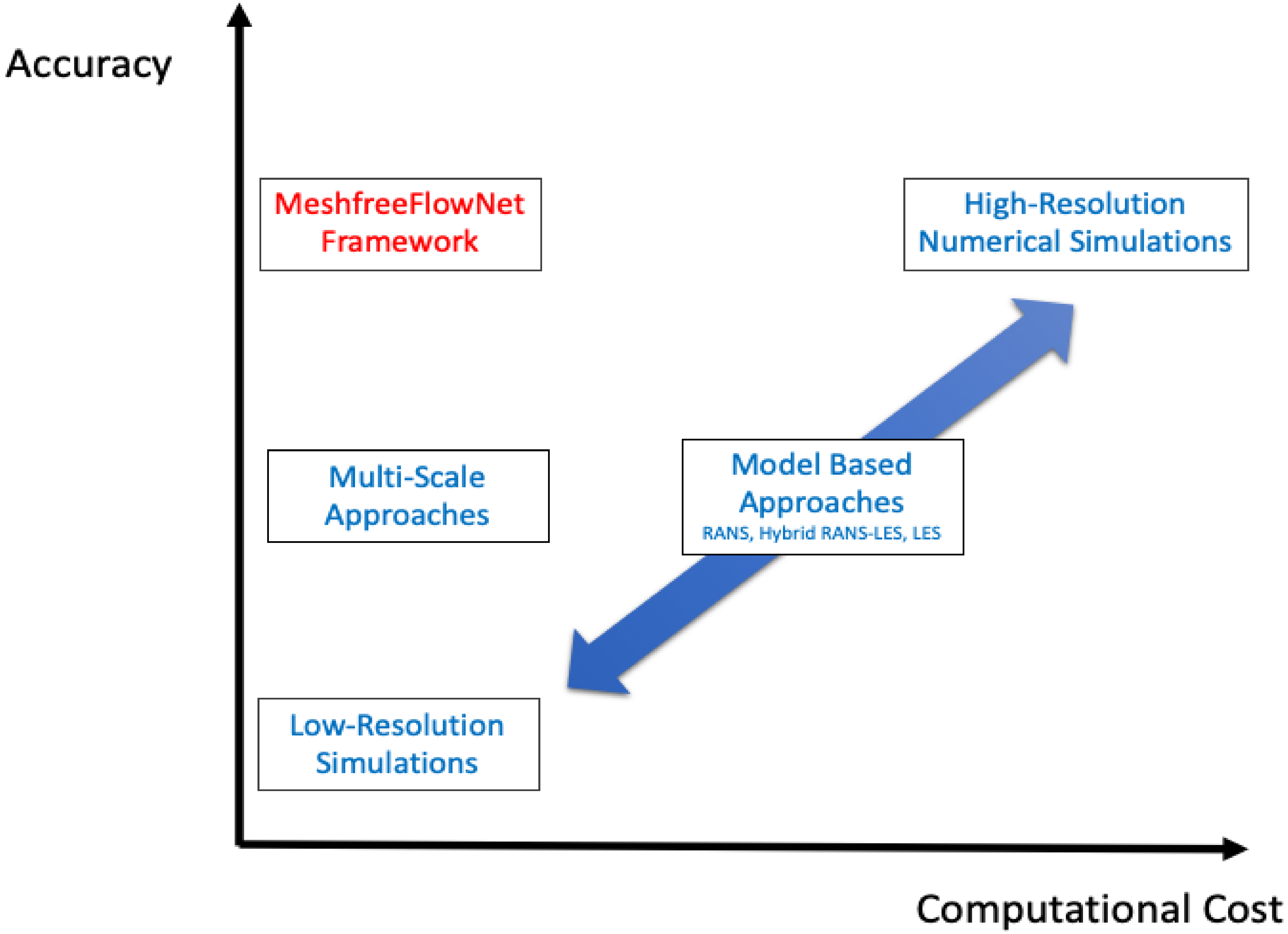
- The physical world is governed by equations.
- Problems in science and engineering reduce to PDEs.







LANDSCAPE OF PDE SOLVERS

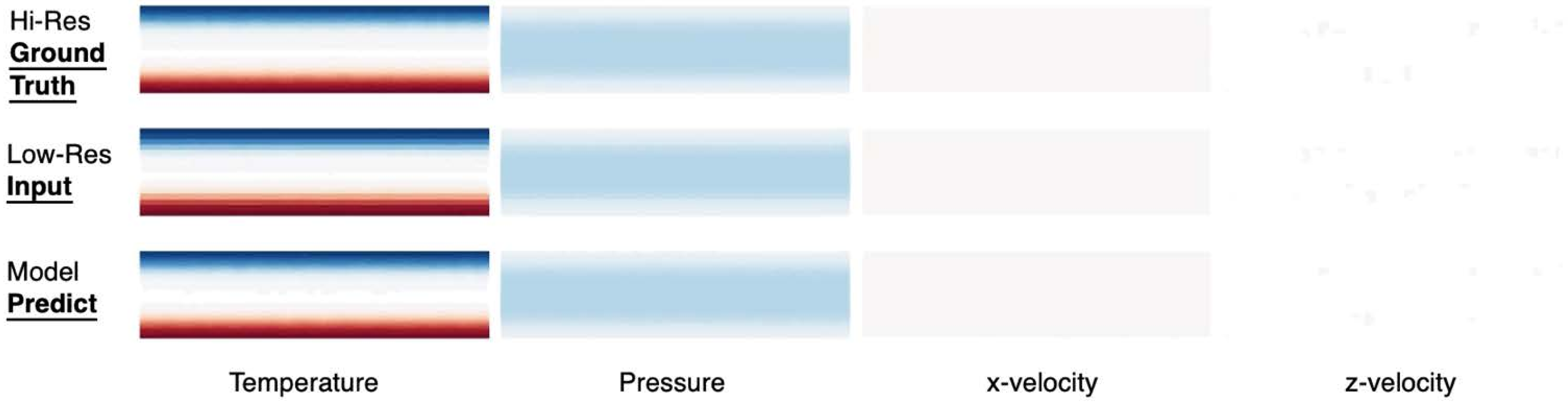


MeshfreeFlowNet: A Physics-Constrained Deep Continuous Space-Time Super-Resolution Framework

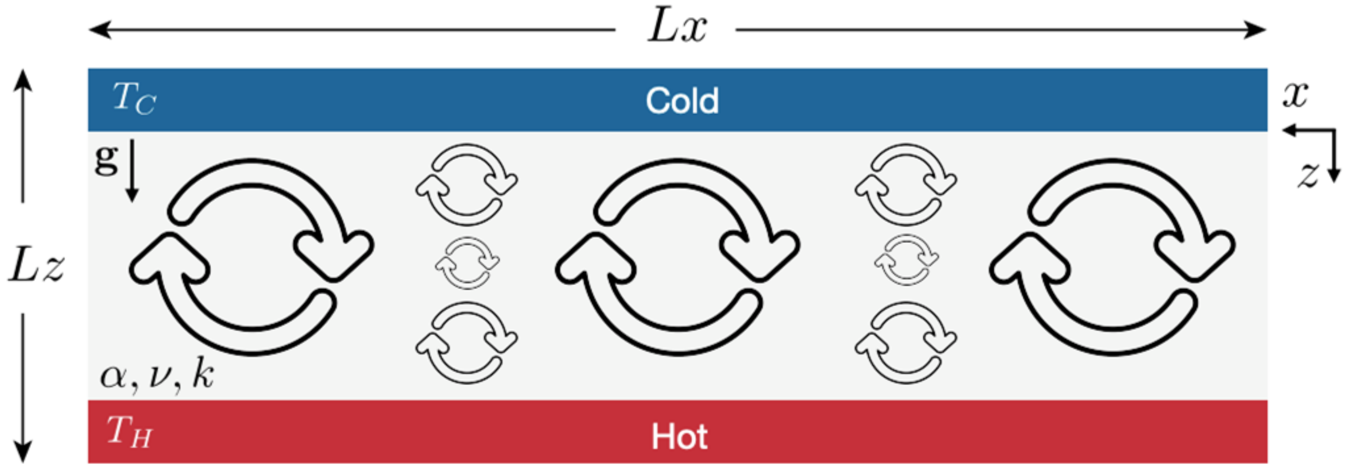
Chiyu "Max" Jiang[1], Soheil Esmaeilzadeh[2], Kamyar Azizzadenesheli[3],
Karthik Kashinath[4], Mustafa Mustafa[4],
Hamdi A.Tchelepi[2], Philip Marcus[1], Prabhat[4] ,
Anima Anandkumar[3,5]

[1] University of California Berkeley
[2] Stanford University
[3] California Institute of Technology
[4] Lawrence Berkeley National Laboratory
[5] NVIDIA

MESHFREE-FLOWNET DEMONSTRATION



RAYLEIGH–BÉNARD CONVECTION PROBLEM



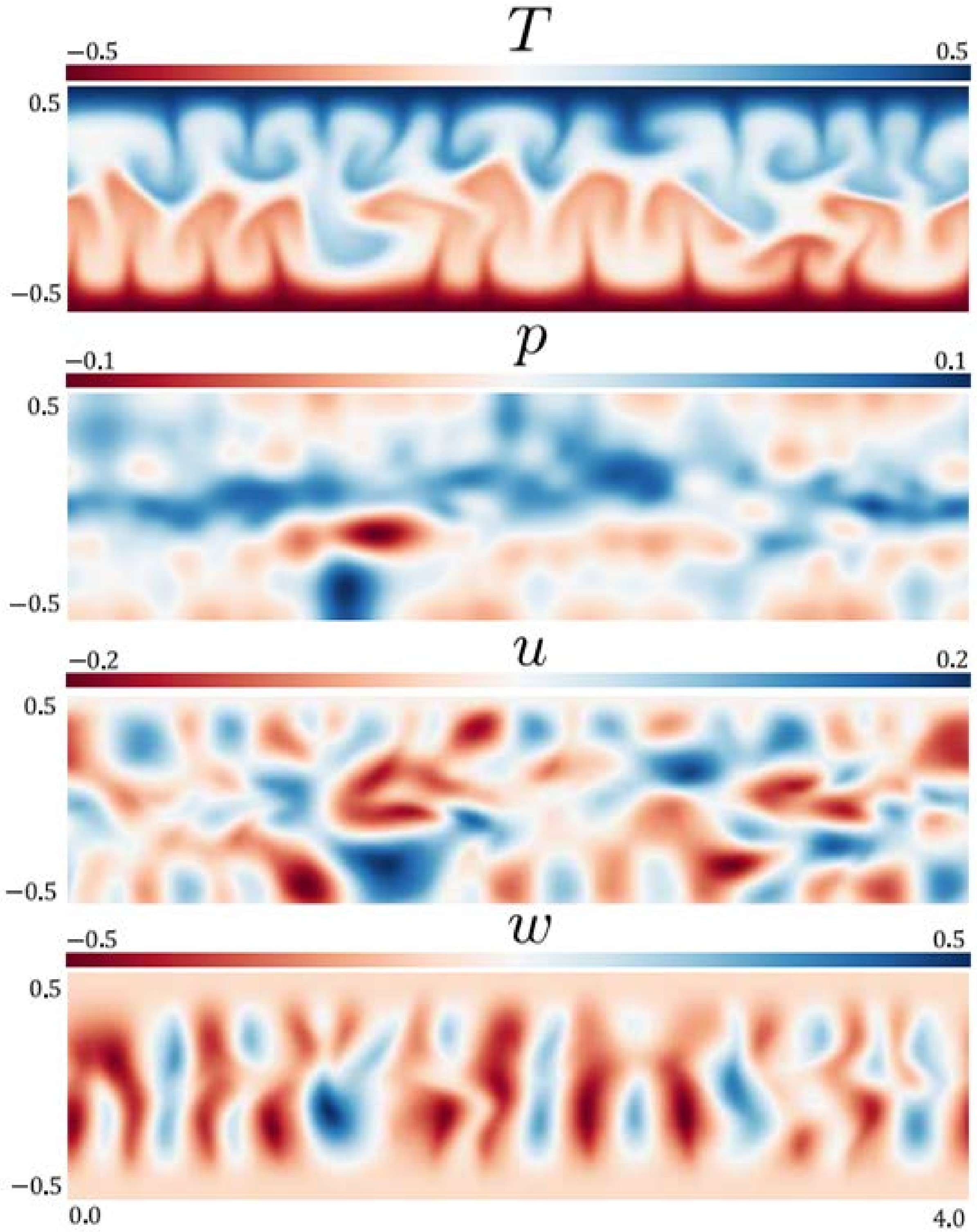
$$\nabla \cdot \mathbf{u} = 0\,,$$

$$\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T - P^* \nabla^2 T = 0\,,$$

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} + \nabla p - T\hat{z} - R^* \nabla^2 \mathbf{u} = 0\,,$$

$$P^* = (Ra\,Pr)^{-1/2} \qquad Ra = g\alpha \Delta T L^3 \nu^{-1} \kappa^{-1}$$

$$R^* = (Ra/Pr)^{-1/2} \qquad Pr = \nu \kappa^{-1}$$



Typical solution of
Rayleigh–Bénard problem

MESHFREE-FLOWNET - OVERVIEW

Loss Functions:

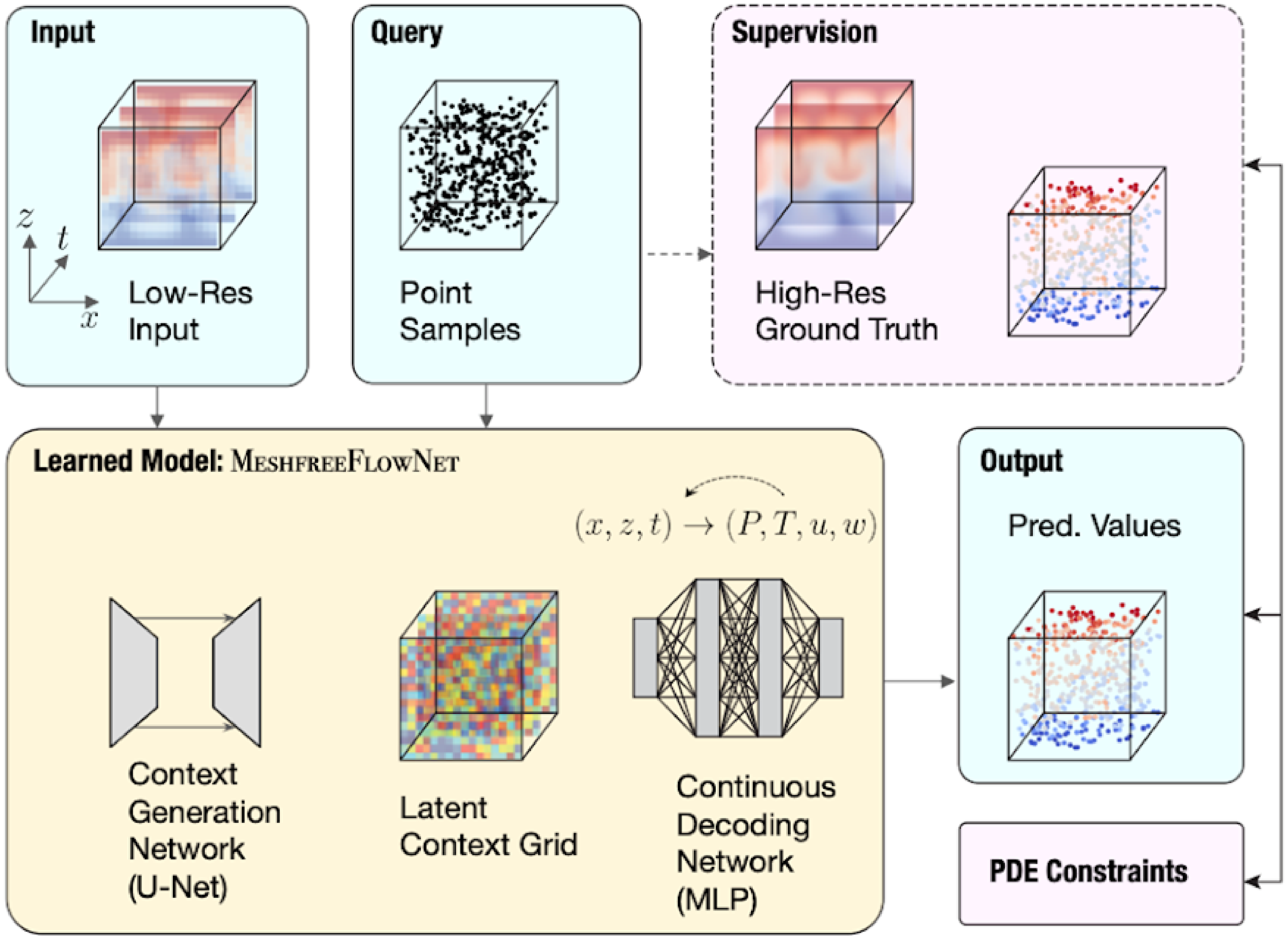
$$\mathcal{L} = \mathcal{L}_p + \gamma \mathcal{L}_e$$

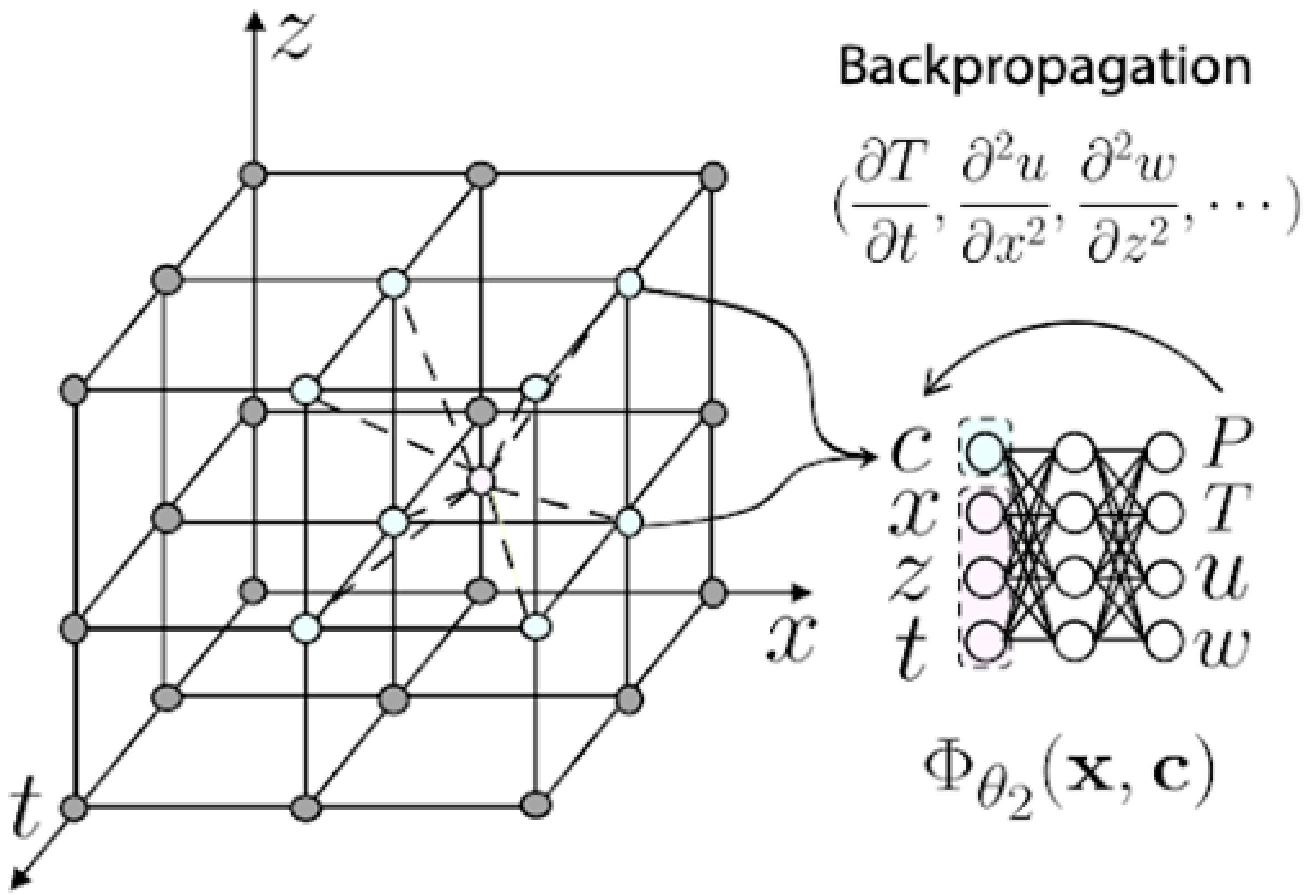
Equation Loss

$$\mathcal{L}_e = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \frac{1}{||B^i||} \sum_{j \in \mathcal{B}^i} ||\Gamma_\Phi \hat{\boldsymbol{y}}_j^i - s||_l$$

Prediction Loss

$$\mathcal{L}_p = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \frac{1}{||B^i||} \sum_{j \in \mathcal{B}^i} ||\boldsymbol{y}_j^i - \hat{\boldsymbol{y}}_j^i||_l$$





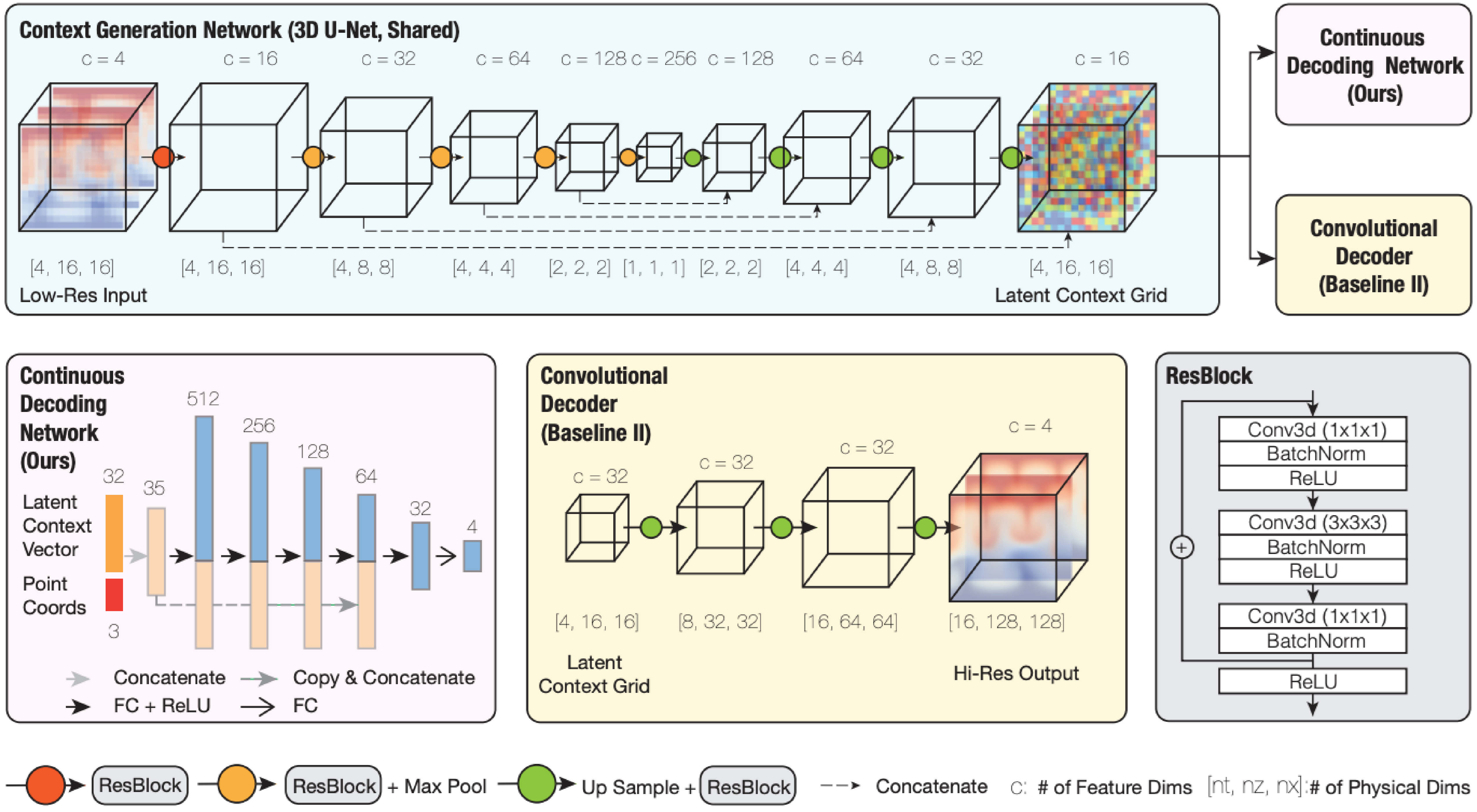
MESHFREE-FLOWNET - ARCHITECTURE

Continuous Decoding Network

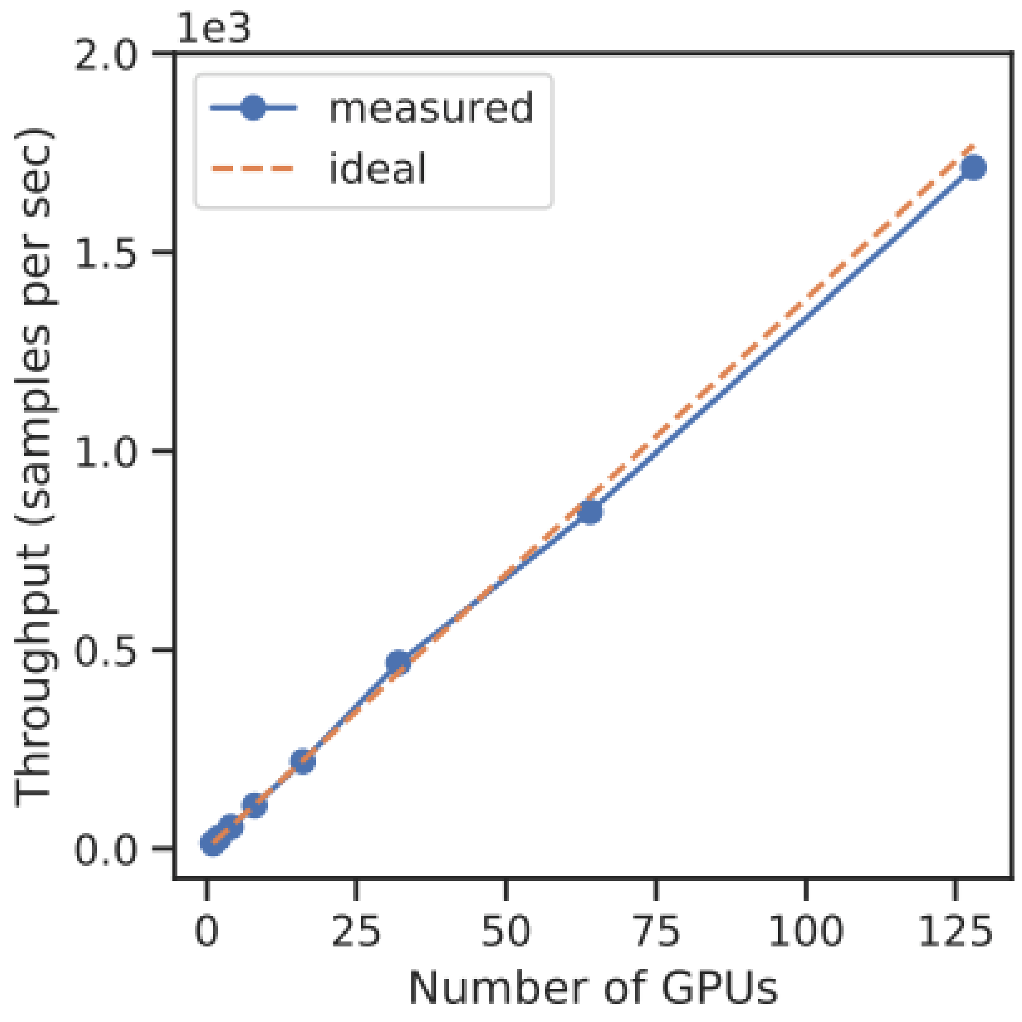
$$\Phi_{\theta_2}(\boldsymbol{x}, \boldsymbol{c}) \ , \quad \mathcal{C}(\boldsymbol{x}_i, \mathcal{G}, \Phi_{\theta_2}) = \sum_{j \in \mathcal{N}_i} w_j \Phi_{\theta_2}\left(\frac{\boldsymbol{x}_i - \boldsymbol{x}_j}{\Delta \boldsymbol{x}}, \boldsymbol{c}_j\right)$$

Context Generation Network

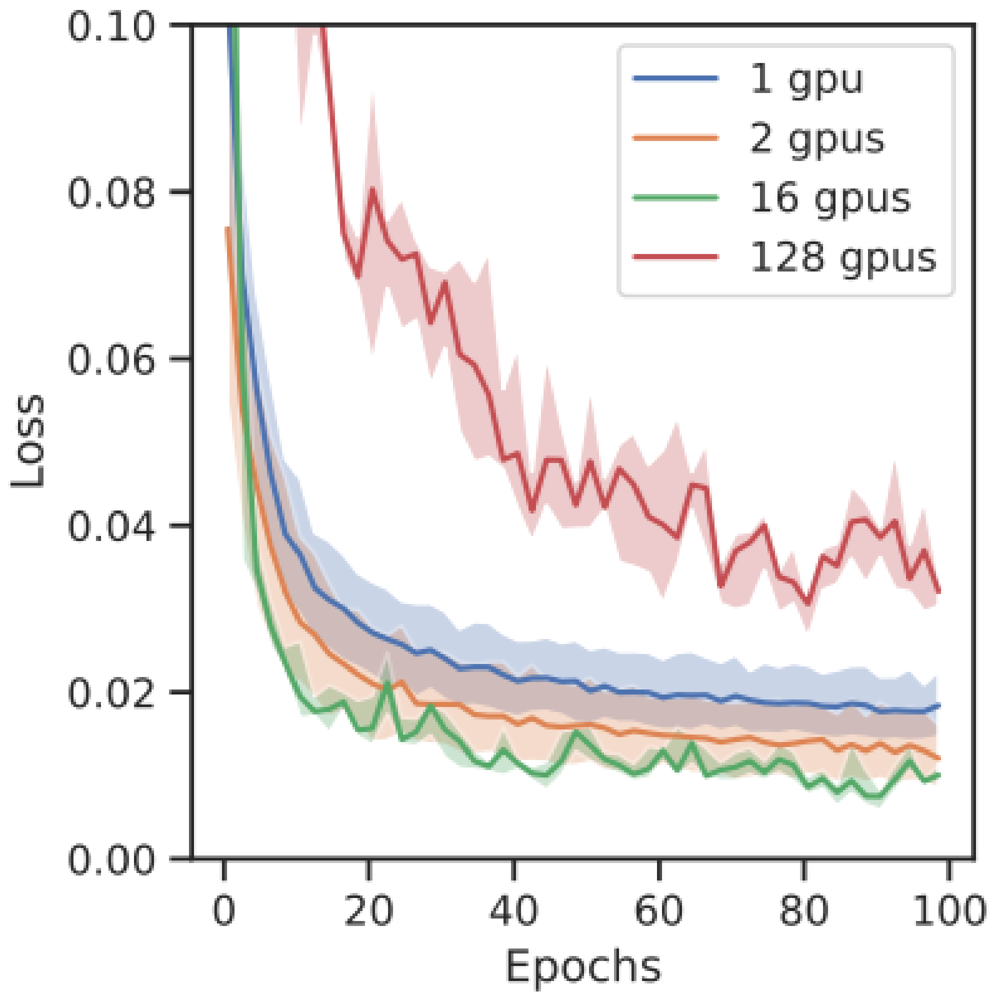
$$\mathcal{G} = \Psi_{\theta_1}(\mathcal{D}_L)$$



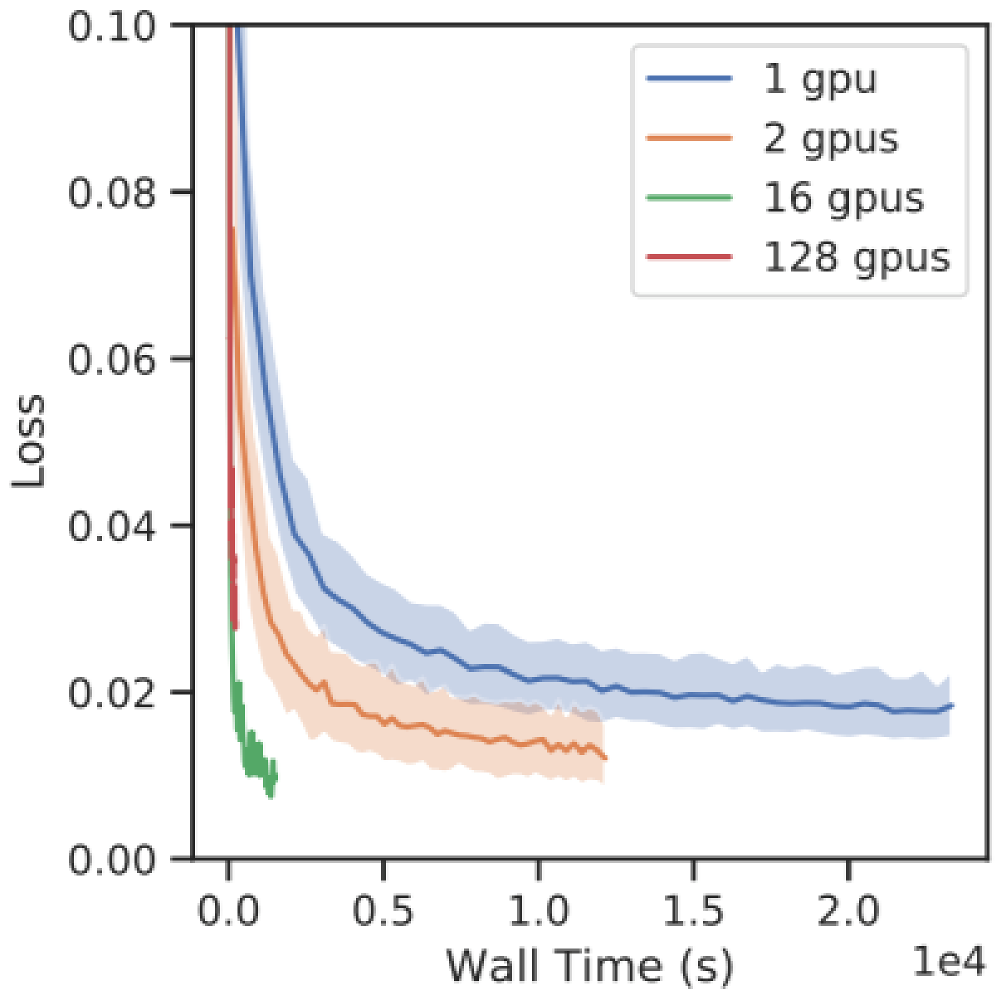
SCALABILITY



(a) Throughput vs. Num. of GPUs

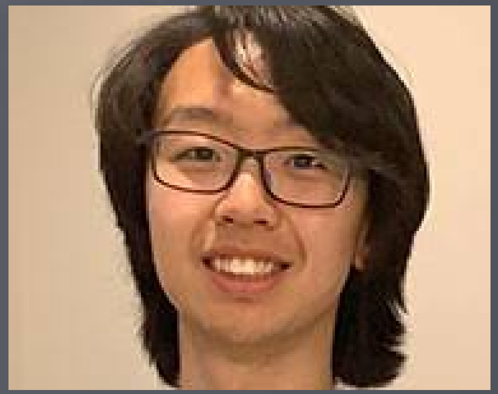


(b) Loss vs. Num. of Epochs

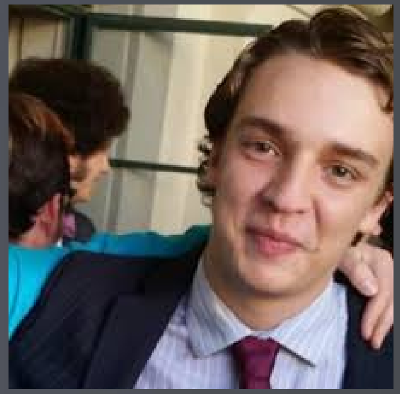


(c) Loss vs. Wall Time

OPERATOR LEARNING FOR PARAMETRIC PDE



Zongyi Li



Nikola Kovachki

Burigede Liu

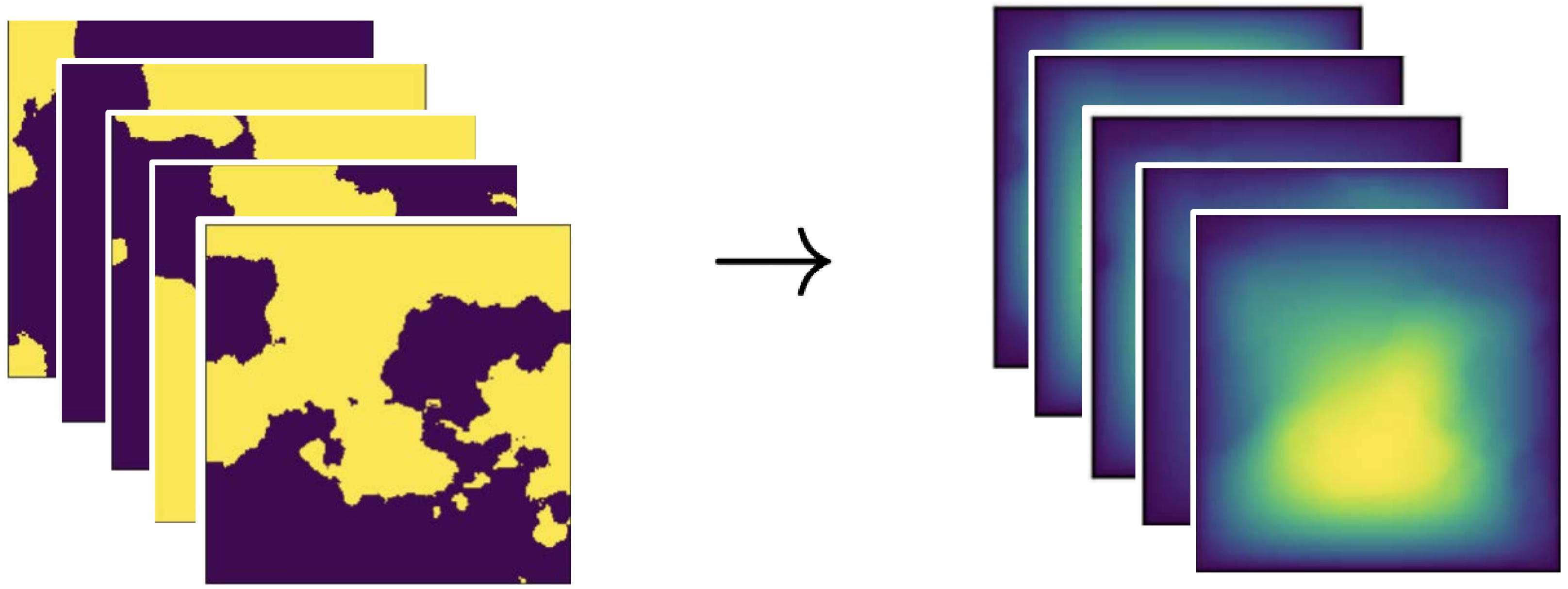
Kamyar Azzizadenesheli

Kaushik Bhattacharya

Andrew Stuart

OPERATOR LEARNING FOR SOLVING PDE

- So far, solved super-resolution problem but requires low-resolution input.
- Now: **<u>Learn operator mapping coefficients of parametric PDE to solution</u>**



Input: coefficients of PDE family

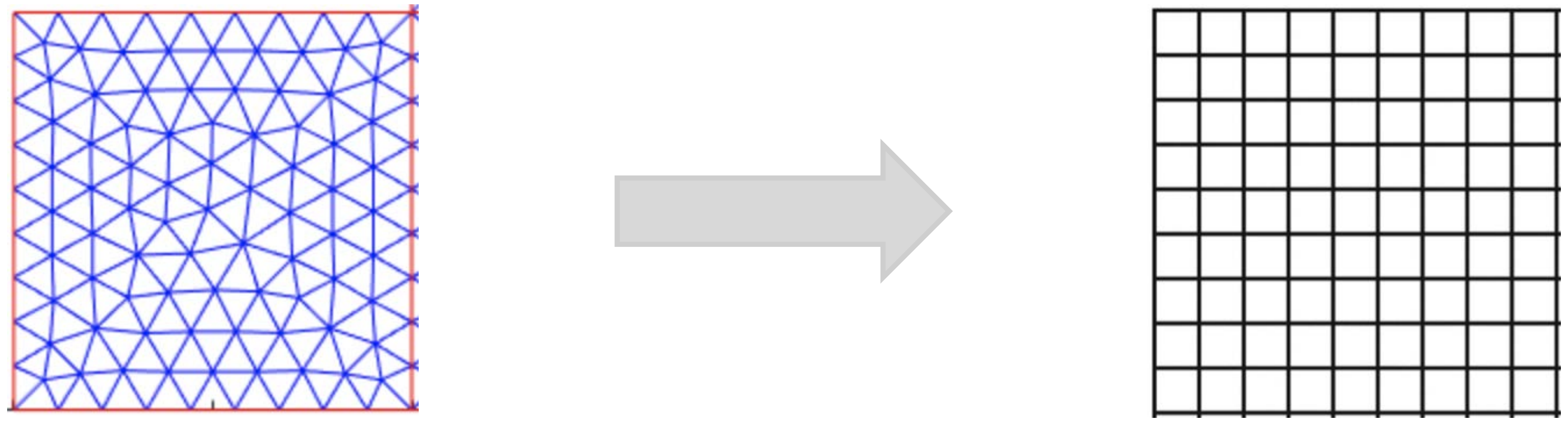
Output: solutions

Slower to train. Fast to evaluate.

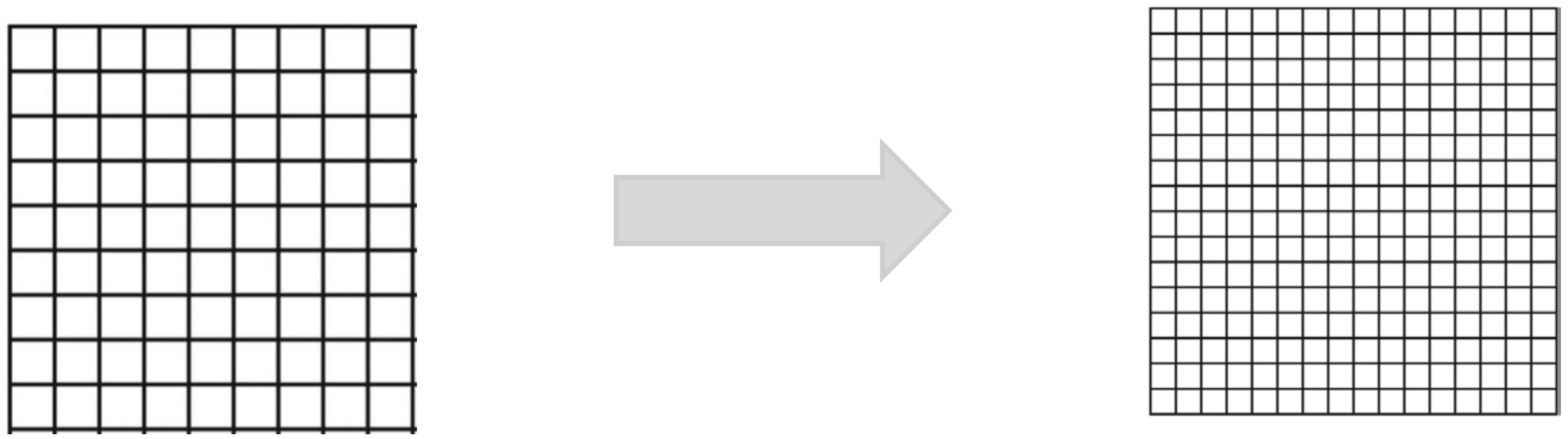
OPERATOR LEARNING

Traditional DL-based methods learn a **vector-to-vector** mapping.
Operator learning aims to learn a **function-to-function** mapping.

- Any discretization
  Any geometry



- Super-resolution



KERNEL METHOD FOR OPERATOR LEARNING

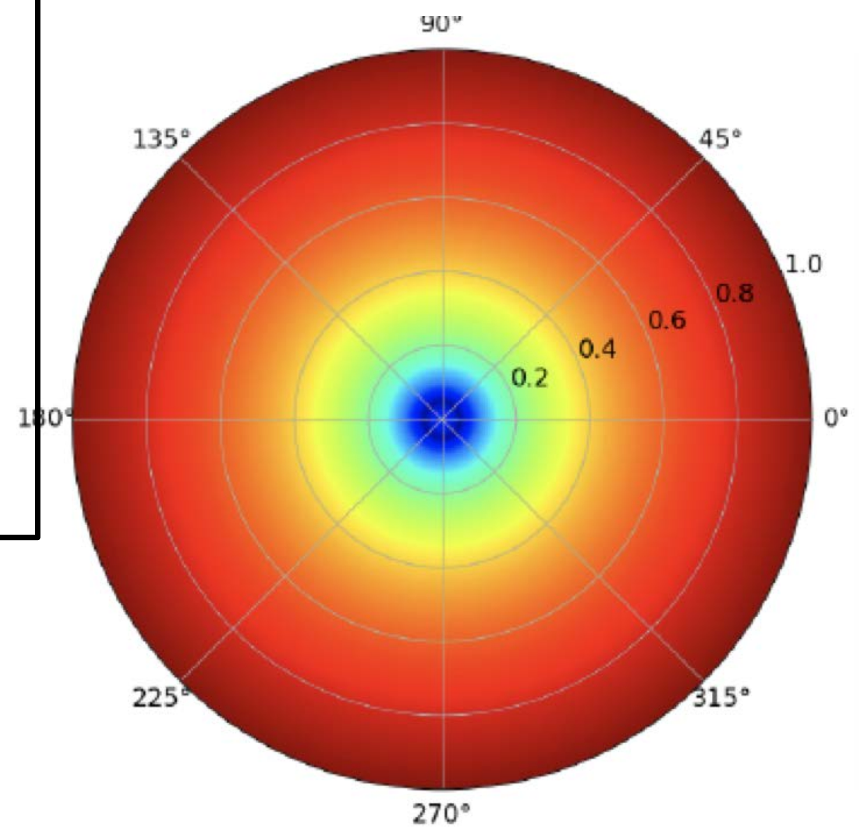
Second order elliptic PDE: $$-\nabla \cdot (a(x)\nabla u(x)) = f(x), \quad x \in D$$
$$u(x) = 0, \quad x \in \partial D$$

Solution of PDE can be written as convolution over Green's function

$$u(x) = \int_D G_a(x,y) f(y)\, dy.$$

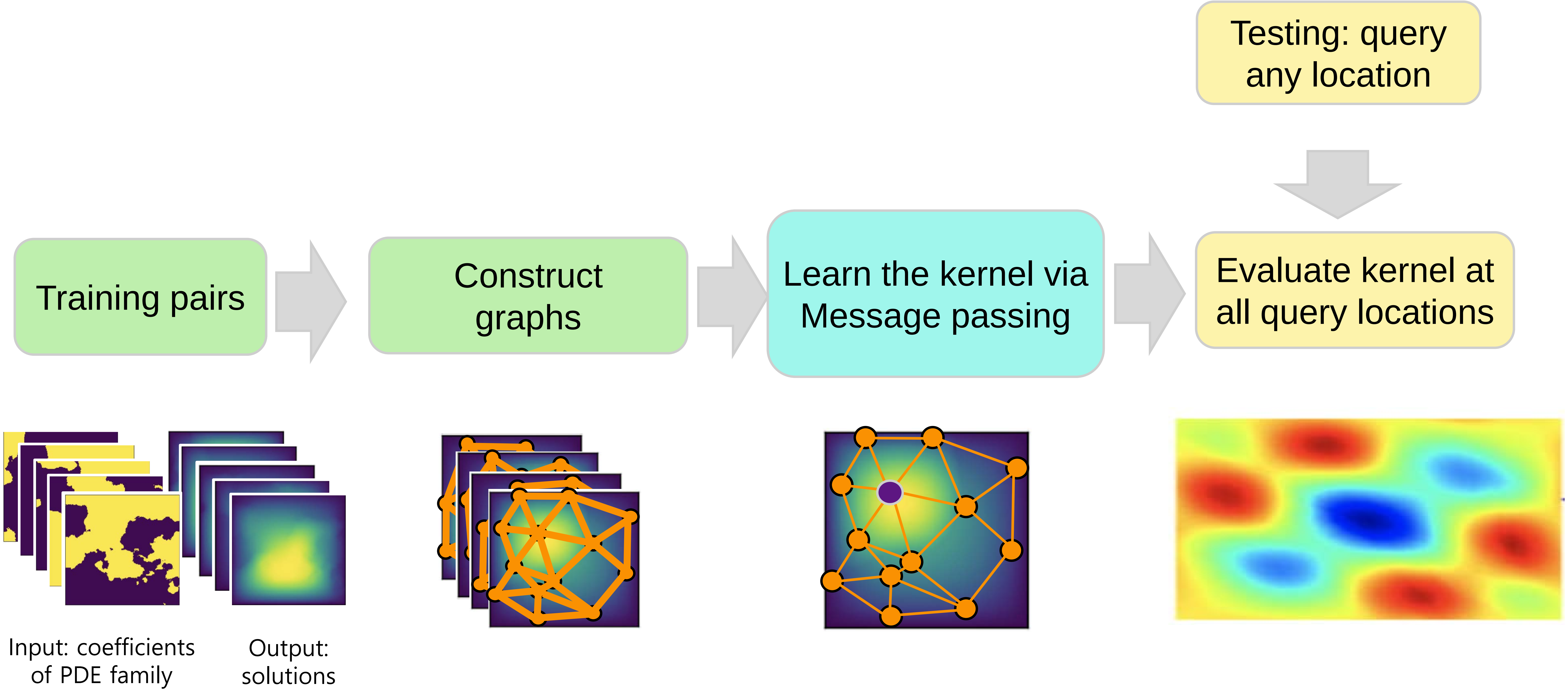
Where G is the green function

- Approximate the kernel by a neural network
- Approximate convolution as message-passing on neighborhood graph



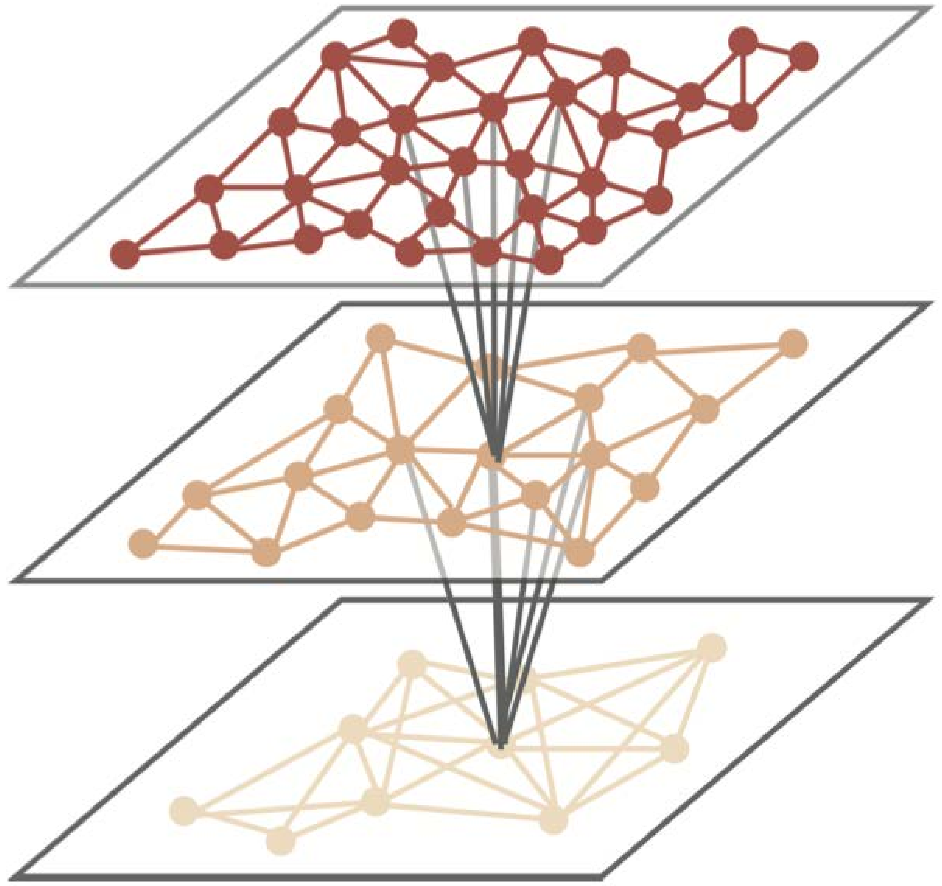
$G_a(x,y).$

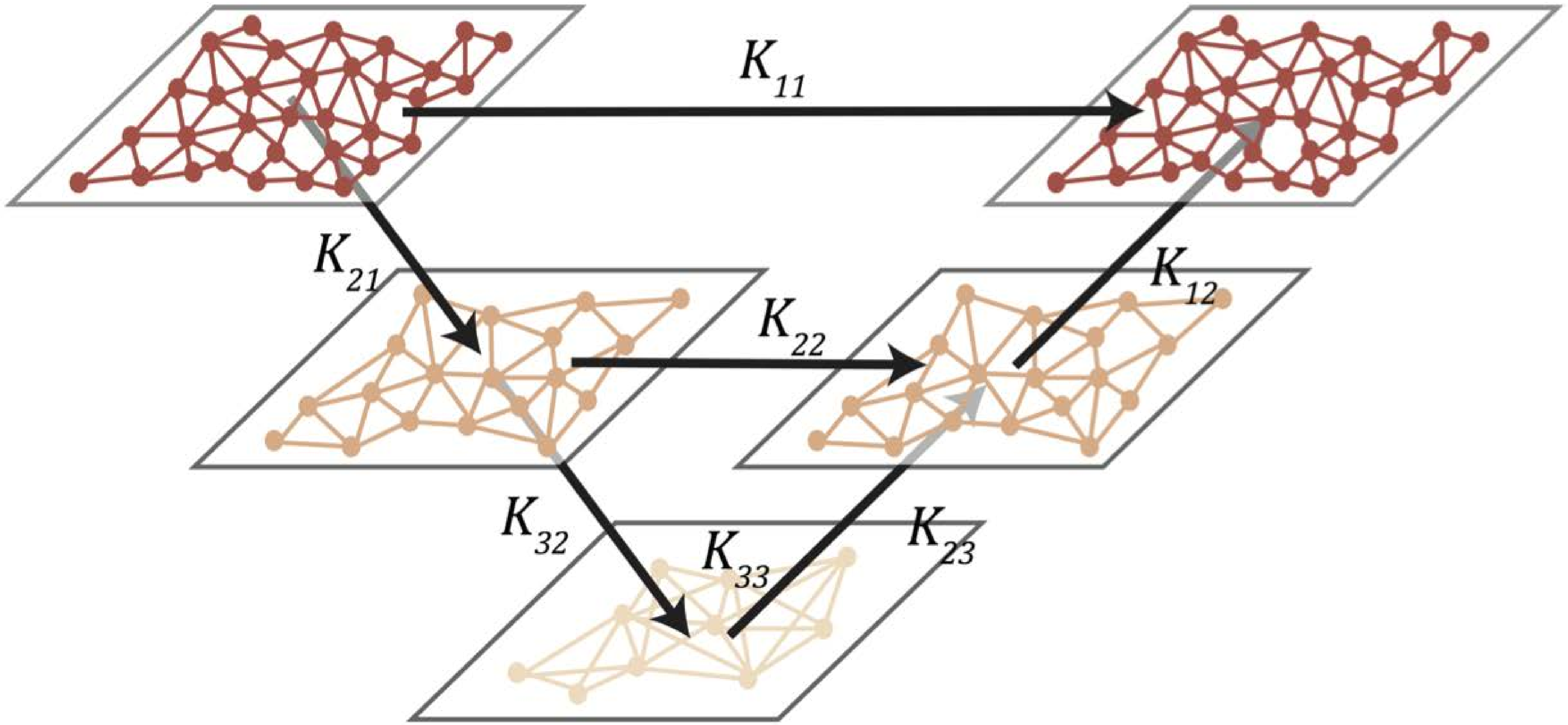
GRAPH NEURAL NETWORKS FOR PDE



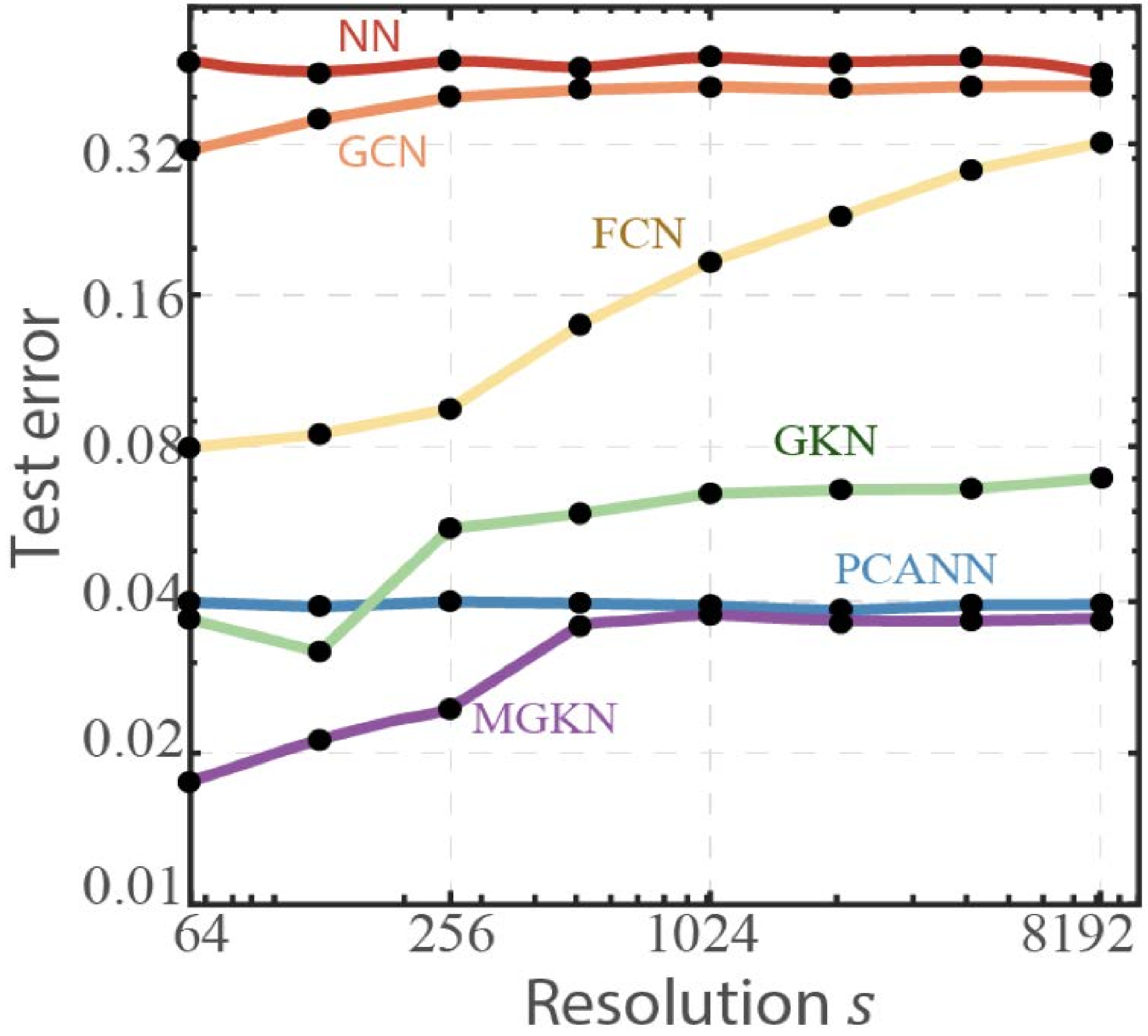
MULTIPOLE GRAPHS

- Multi-scale graphs to capture different ranges of interaction
- Linear complexity





EXPERIMENTAL RESULTS

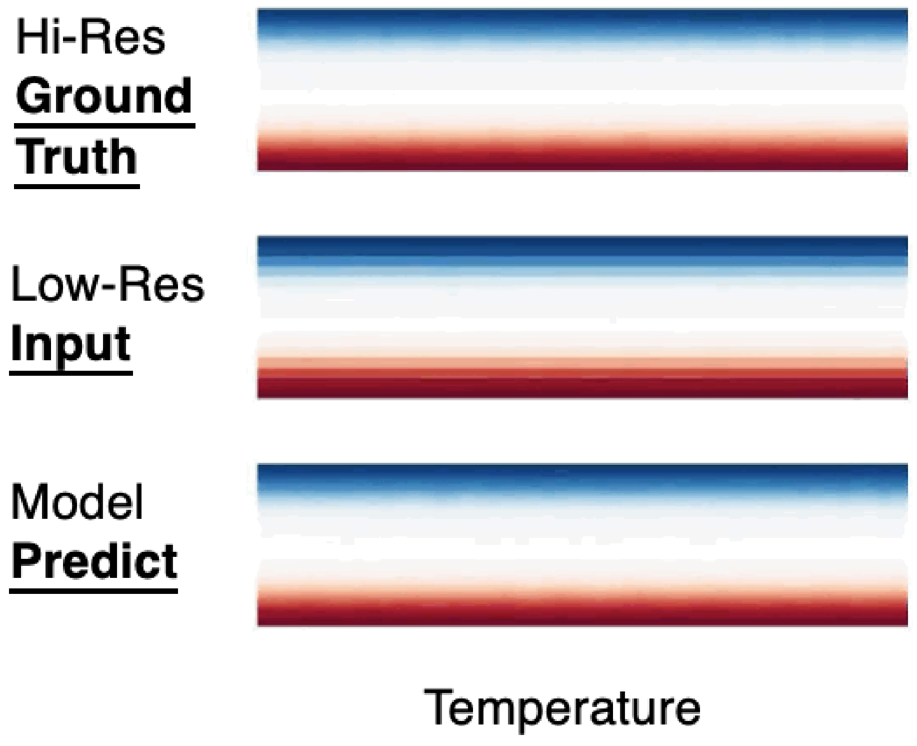


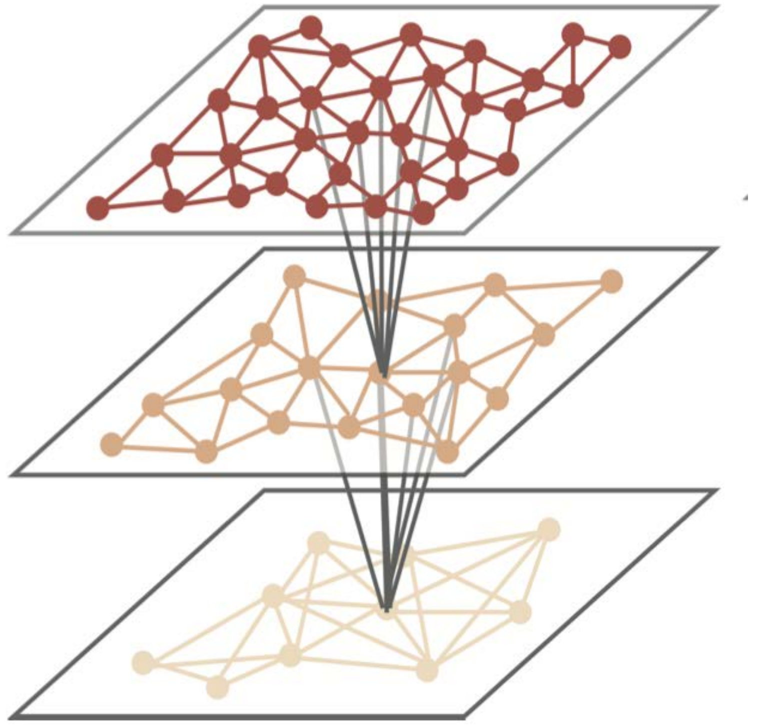
- MGKN has best error and low computational complexity.
- Able to solve PDE from scratch

SUMMARY

Principled approaches for data-driven PDE solvers

- Meshfree neural networks enable super-resolution with low computational cost
- PDE constraints preserve physical validity
- Operator learning solves PDEs from scratch in any resolution
- Multipole graphs can capture long-range correlations





Causal Discovery in Physical Systems from Videos

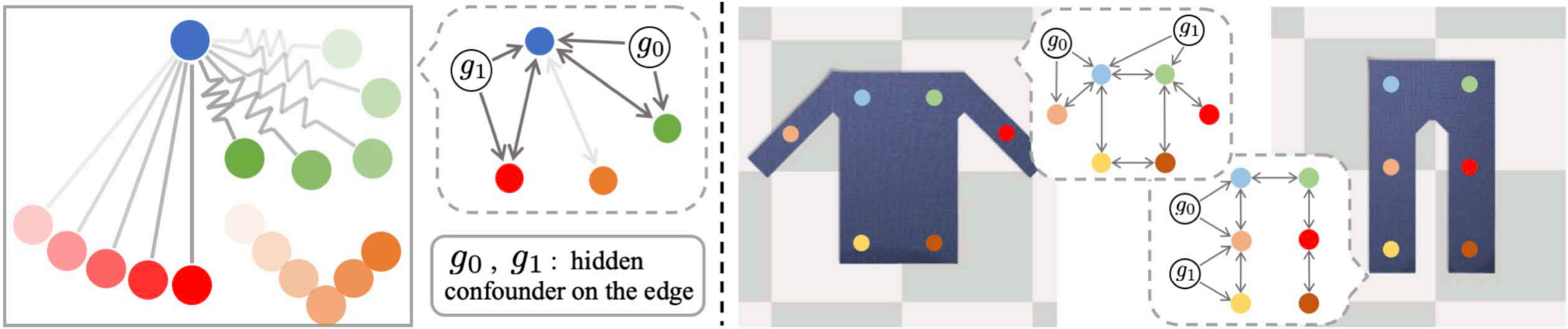
Yunzhu Li

Antonio Torralba

Dieter Fox

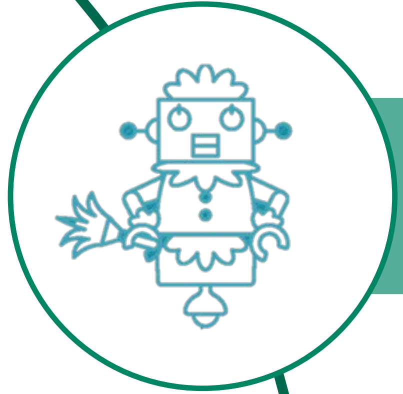
Animesh Garg

MANY EXAMPLES OF CAUSAL SYSTEMS



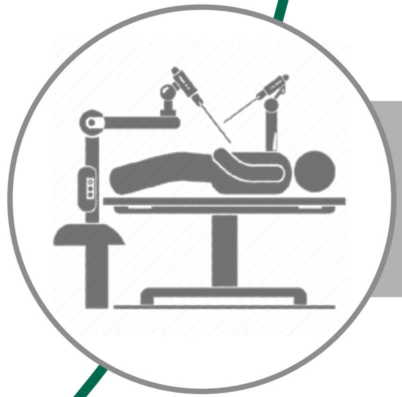
- Learning causality vs correlations
- Hidden variables
- Learning from visual data: high dimensional

COMPOSITIONAL REPRESENTATIONS

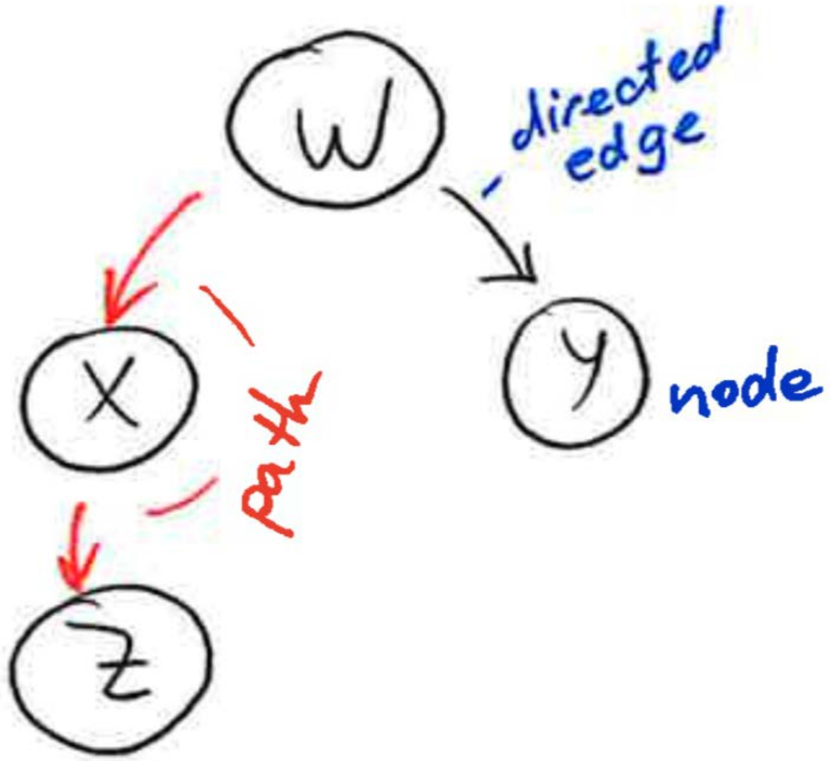


Learning Disentanglement

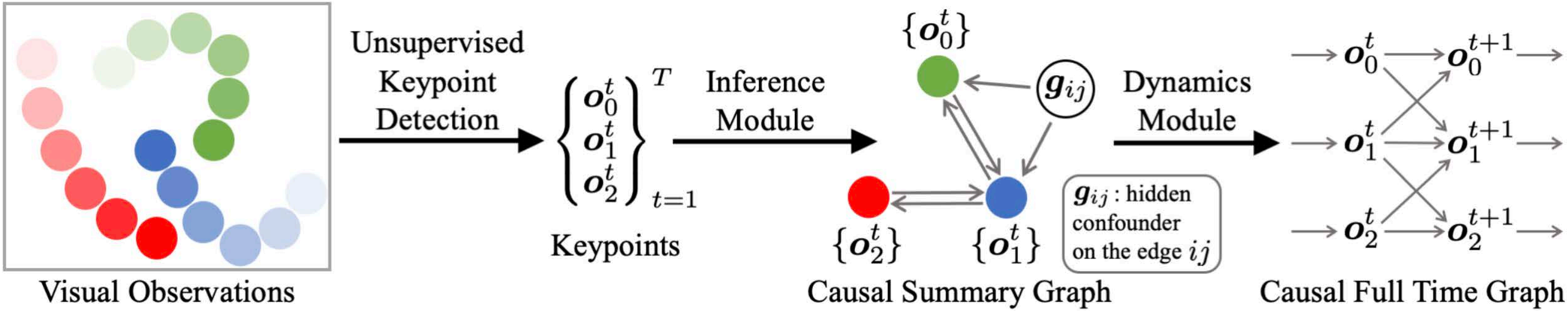
Learning Keypoints



Learning Causal Graphs



LEARNING CAUSALITY

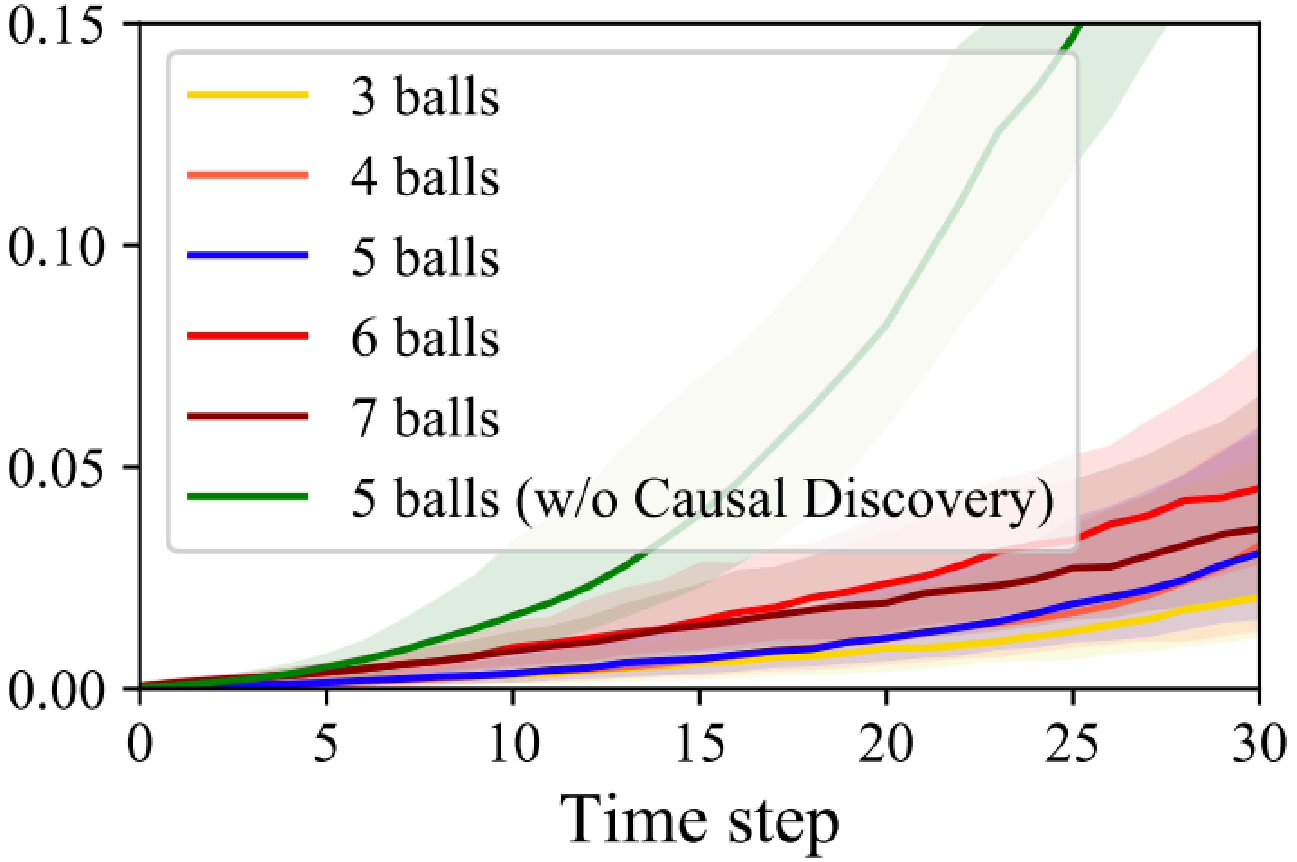


PERFORMANCE ON CLOTH SIMULATION

PERFORMANCE ON CLOTH SIMULATION

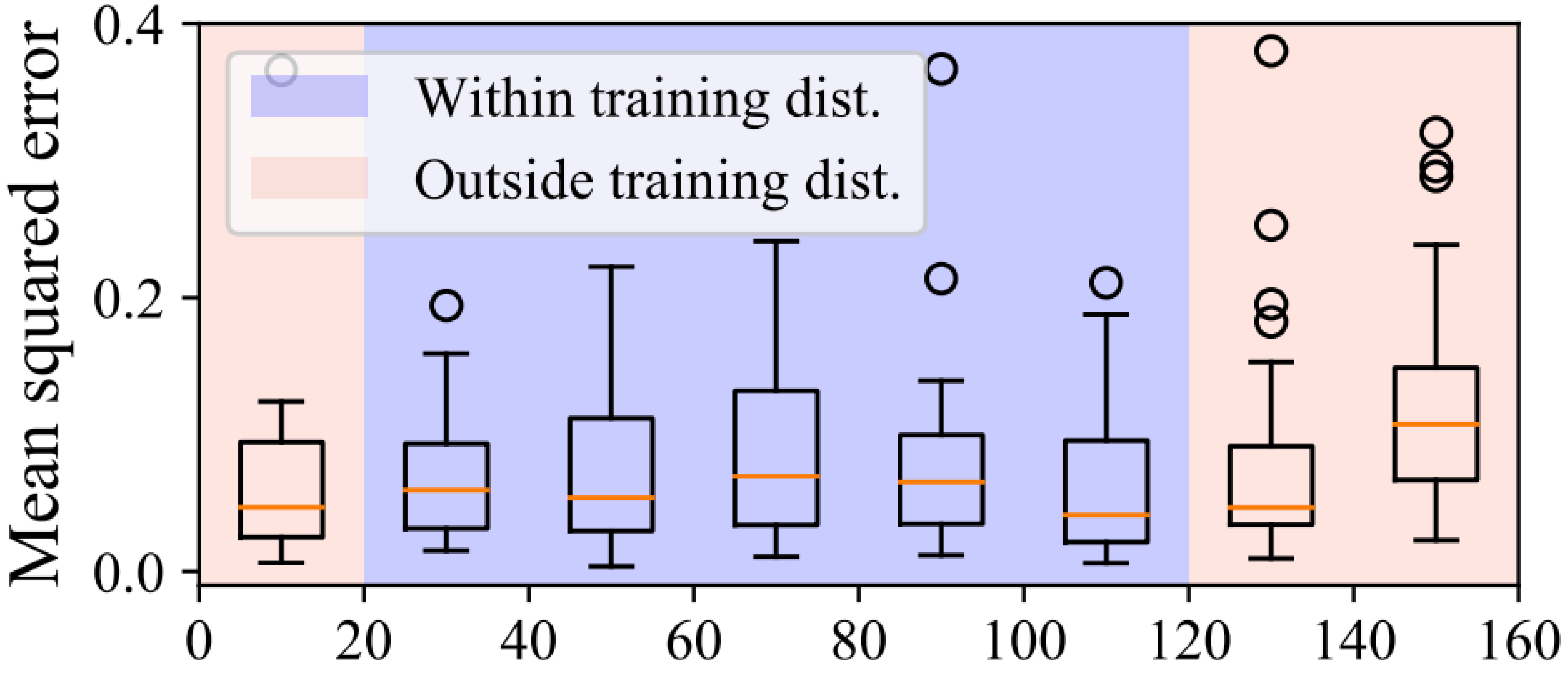
RESULTS ON BALL COLLISIONS

Extrapolation



(d) Mean squared error on future prediction

Counterfactual

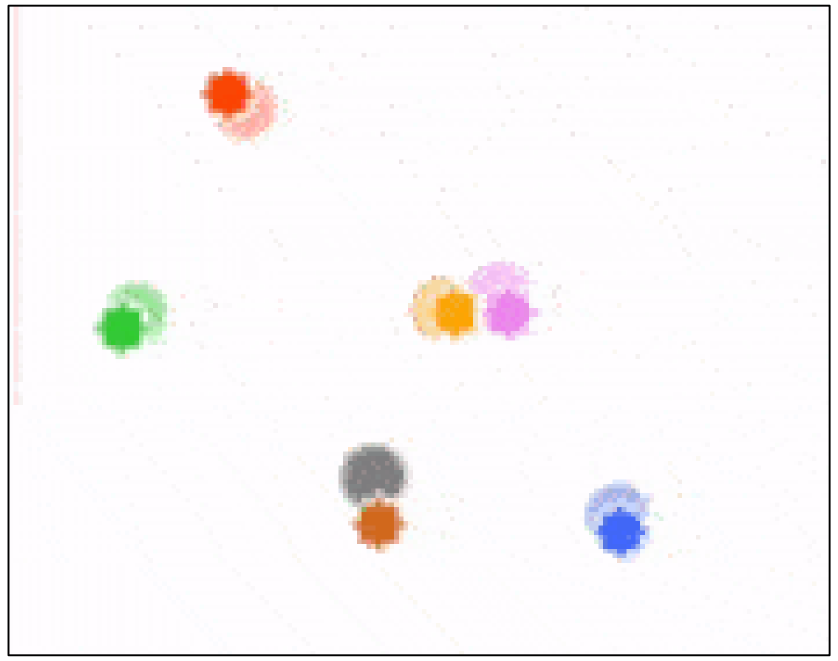


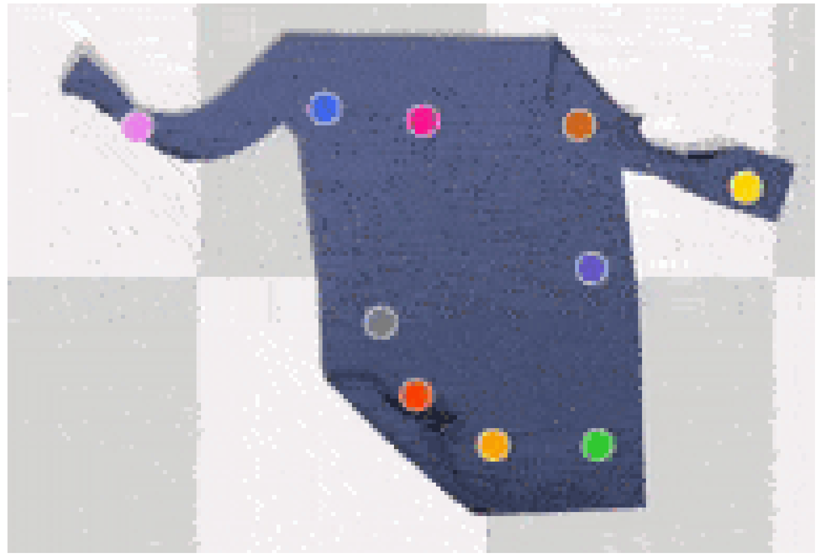
(a) Intervention on the rest length in spring

SUMMARY

Causal learning from videos

- Learning from high-dimensional videos is challenging
- Keypoint detection provides dimensionality reduction
- Graph neural networks to infer interactions
- Learning on different configurations allows for causal learning





CONCLUSION

- AI4control requires preserving stability and safety guarantees
- Robust learning guarantees safe exploration and planning
- NVIDIA Isaac enables physically valid simulations for robot learning
- Deep learning can speed up or even completely replace traditional PDE solvers
- Meshfree neural networks enable super-resolution
- Operator learning can completely replace traditional PDE solvers