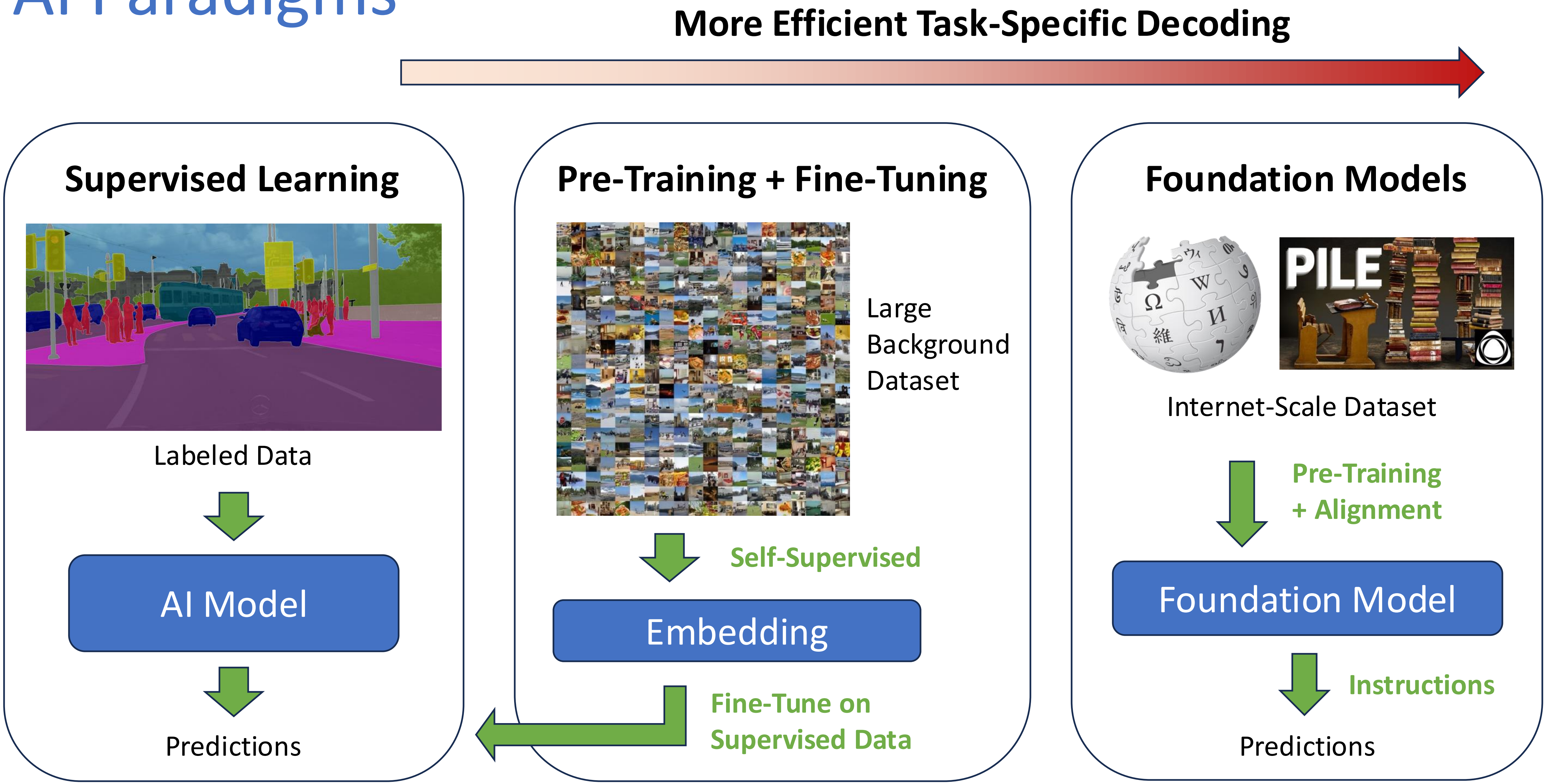
Caltech

Neurosymbolic AI for Safety-Critical Agile Control

Yisong Yue

AI Paradigms



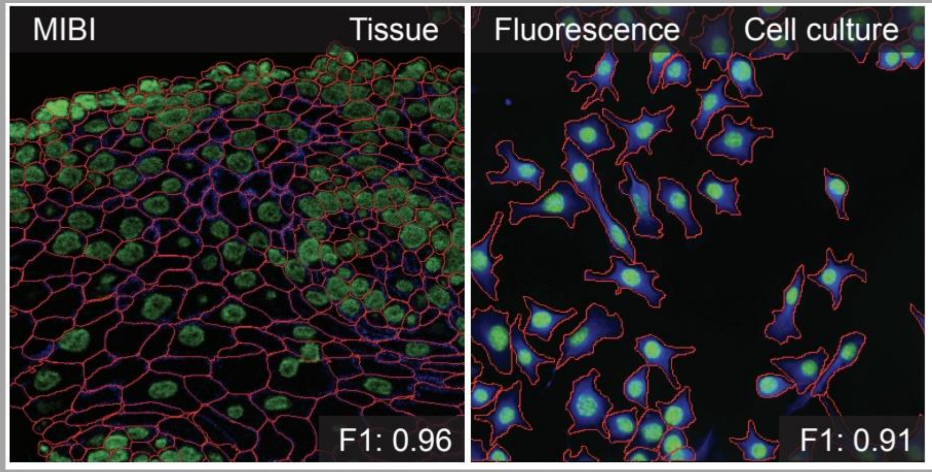
AI Paradigms

More Efficient Task-Specific Decoding

Can now build AI models for many tasks!

Autonomy

https://www.zuken.com/us/blog/how-are-satellites-bringing-low-latency-internet-to-autonomous-vehicles/



Science

https://cellsam.deepcell.org/

Knowledge Work

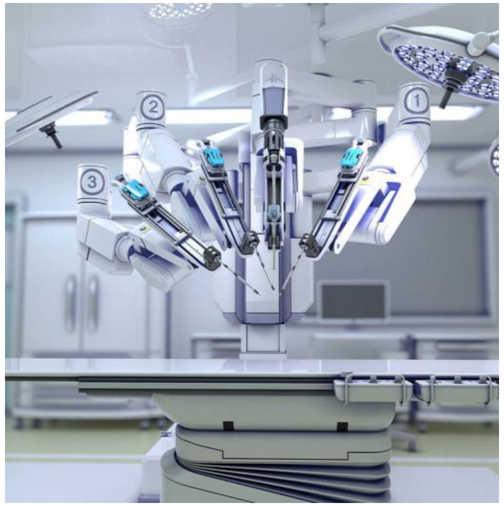
(Github Copilot)

Real Systems have Complex Requirements

"I want to use deep learning to optimize the design, manufacturing and operation of our aircrafts. But I need some guarantees."

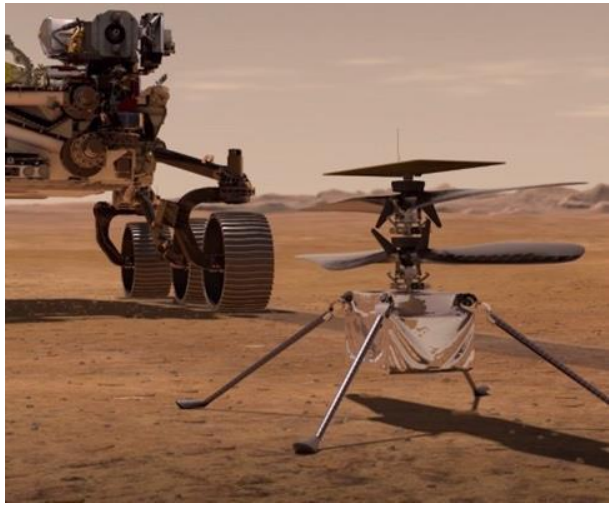
– an Aerospace Director while visiting Caltech

Social & Behavioral Dynamics

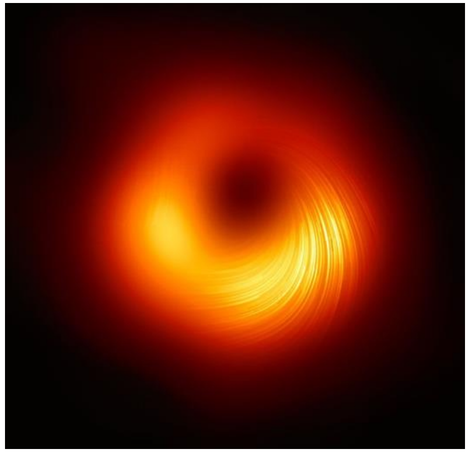


Precise Control

https://urologyaustin.com/urology-specialties/da-vinci-robotic-surgery/



Safe Exploration



Valid Inferences

Unresolved Complexity → Engineering Overhead

Example:

$160B invested in Self-Driving *

WAYMO · mobileye (An Intel Company) · TESLA · aurora · Motional · LATITUDE · cruise · waabi

(* 2022 Estimate) https://www.autonews.com/mobility-report/autonomous-vehicle-reality-check-after-160-billion-spent

Core Cause:

Leaky Abstractions

↓

Many Corner Cases

↓

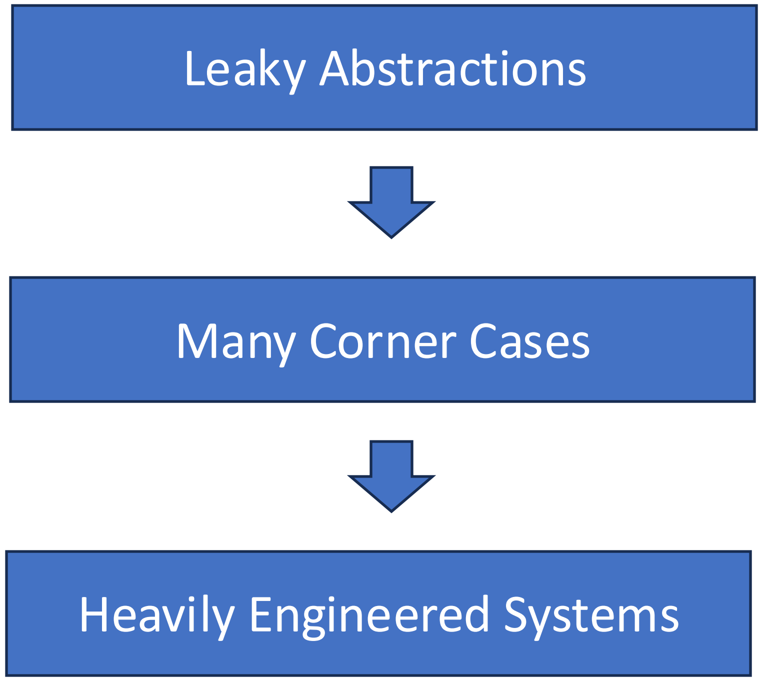
Heavily Engineered Systems

Unresolved Complexity → Engineering Overhead

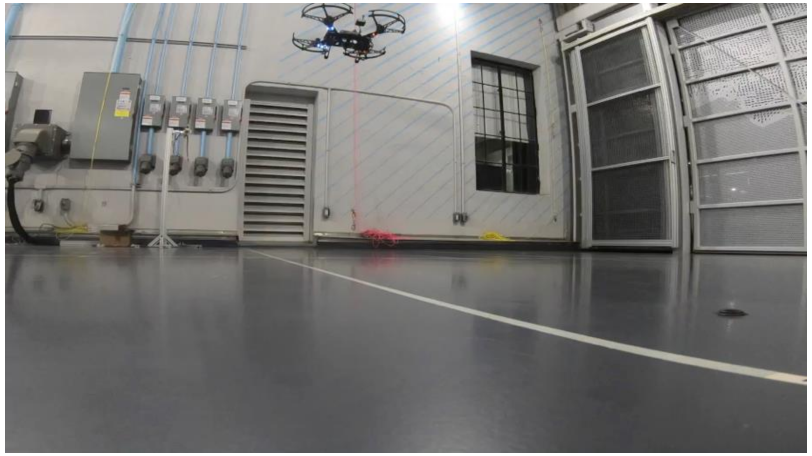
Strong Abstractions Enable Building Complex Systems

- Interfaces between components
- Contracts that should be satisfied
- Empowers debugging & verifying

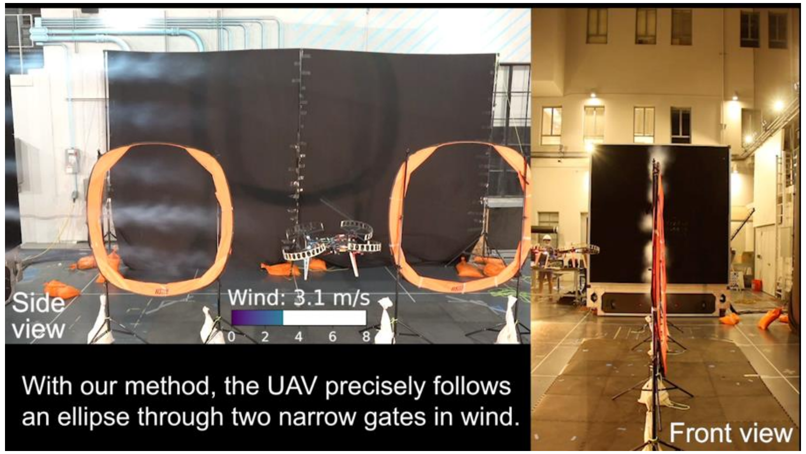
Core Cause:



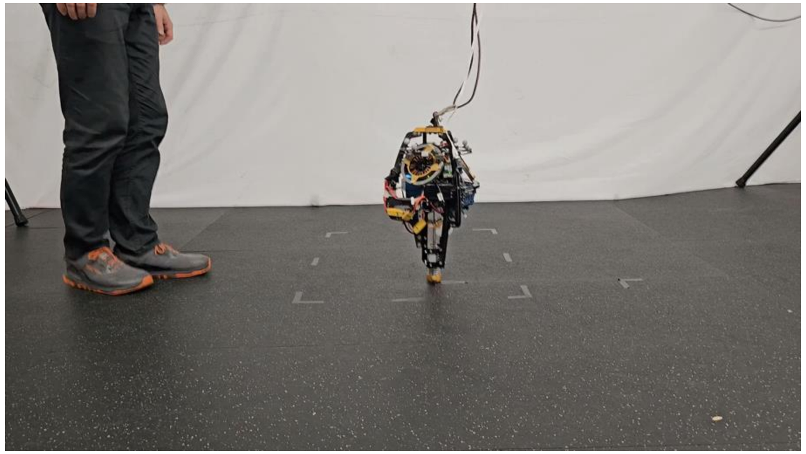
Case Study: Agile Robotic Control



Boundary Conditions
https://arxiv.org/abs/1811.08027

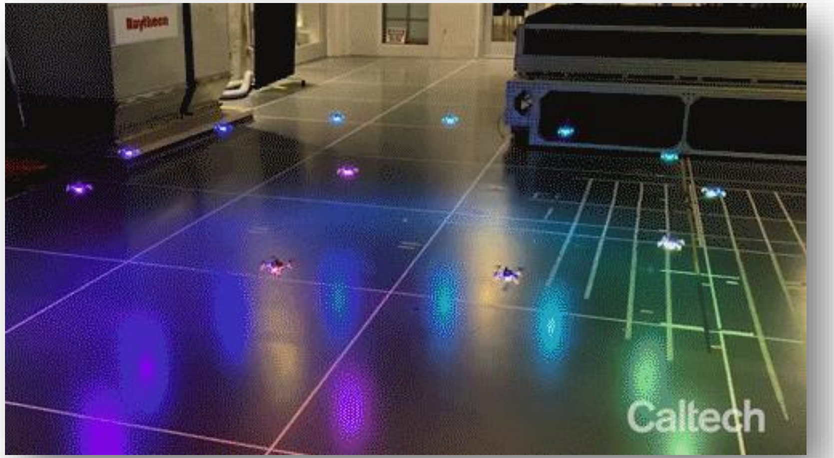


Dynamic Environments
https://arxiv.org/abs/2205.06908



Sharp Disturbances
https://arxiv.org/abs/2409.06125

Human-Robot Interactions
http://roams.caltech.edu/



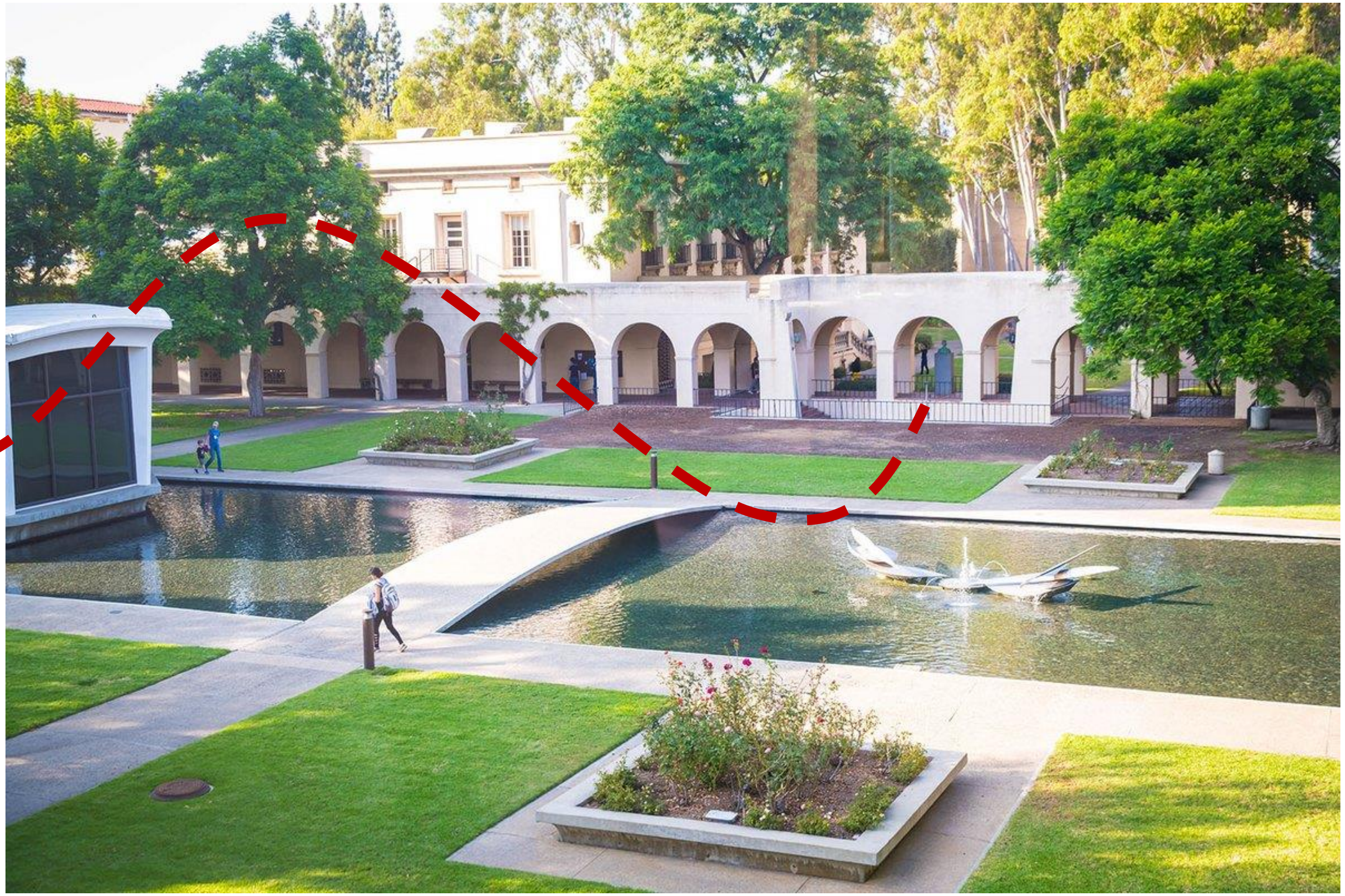
Multi-Agent Interactions
https://www.caltech.edu/about/news/machine-learning-helps-robot-swarms-coordinate

Standard Autonomy Stack (simplified)

1. Perception & Sensing

3. Motor Control

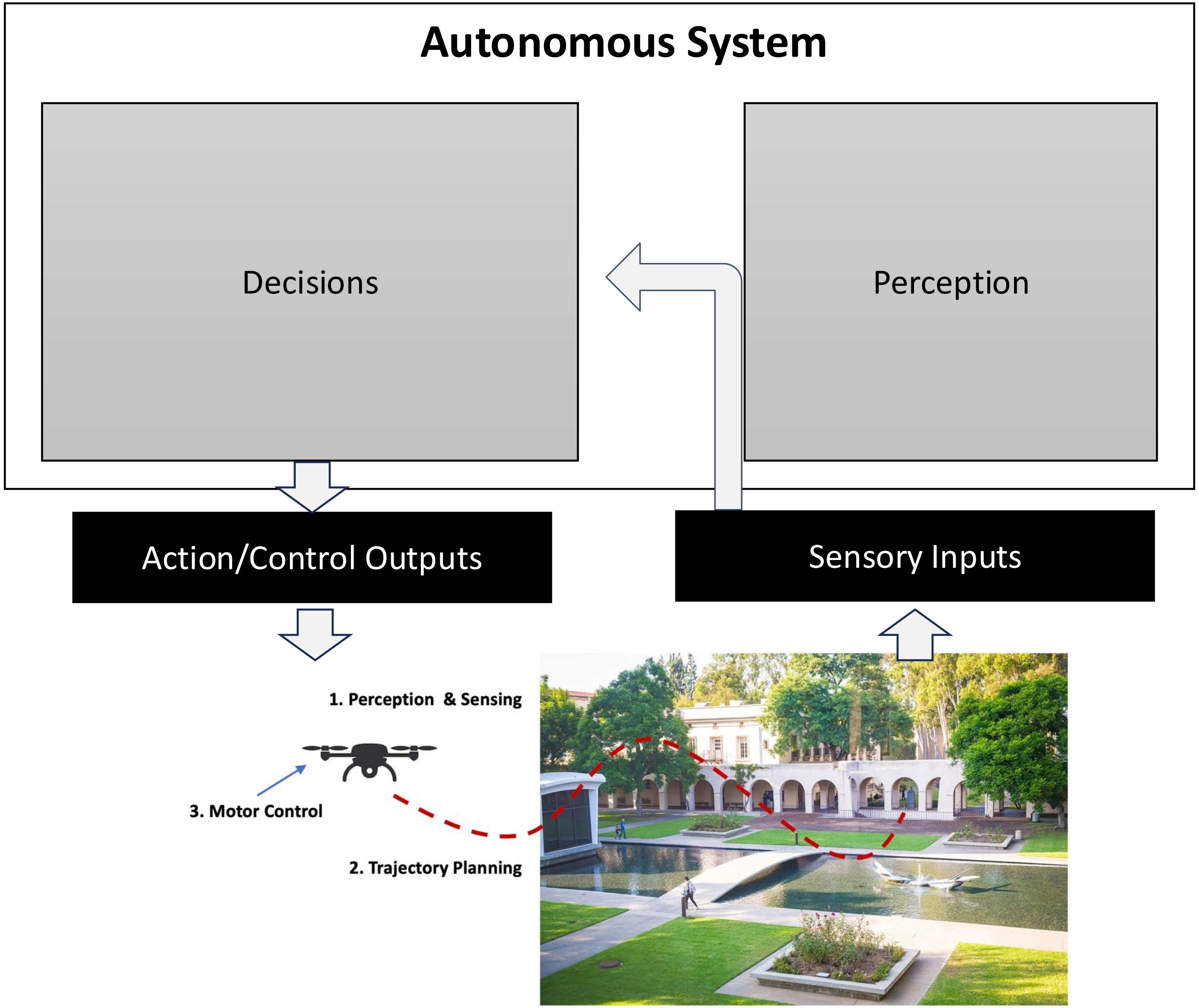
2. Trajectory Planning



... Deal With Other Agents
... Deal With Wind & Disturbances
... Precise Control Around Barriers
... Carrying Payload (eg. wobbly package)
... Etc.

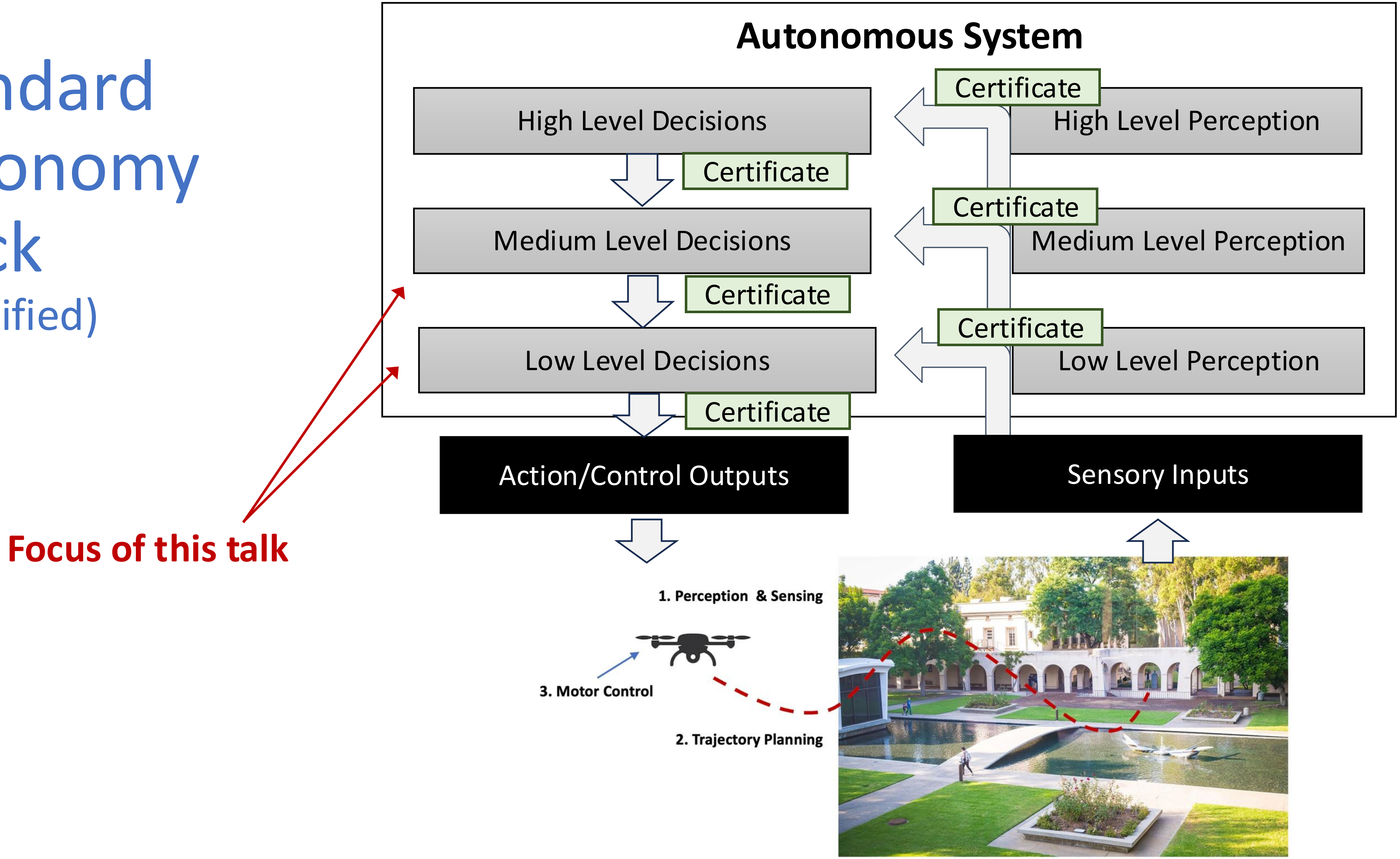
Standard Autonomy Stack

(simplified)



Standard Autonomy Stack

(simplified)

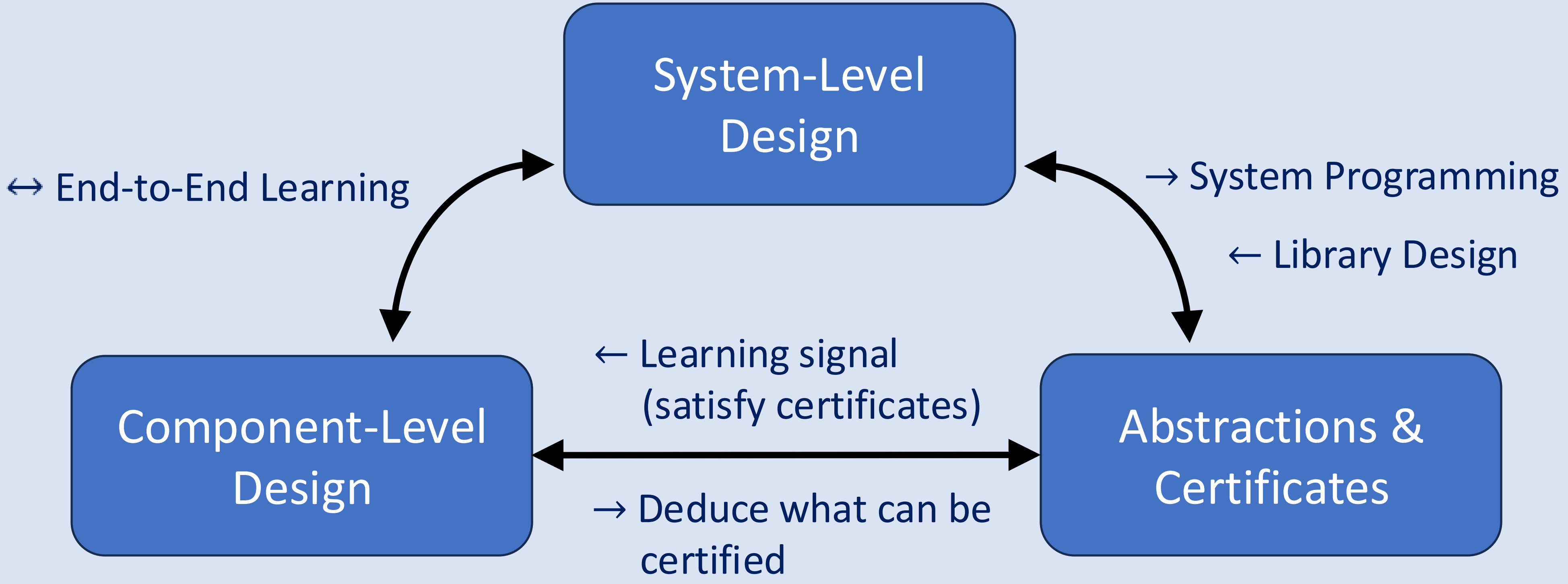


Standard

Autonomous System

Certificate

Can use AI to help optimize any aspect!



Research Questions

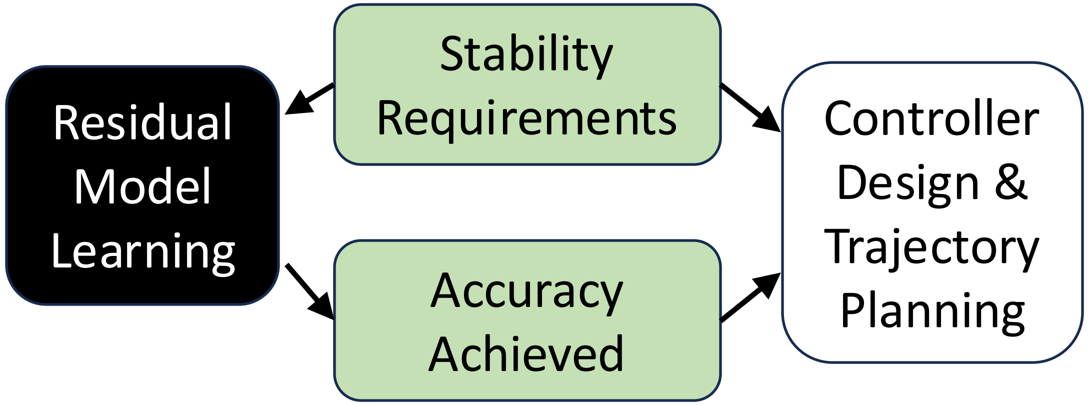
- How to define abstractions to capture system-level requirements

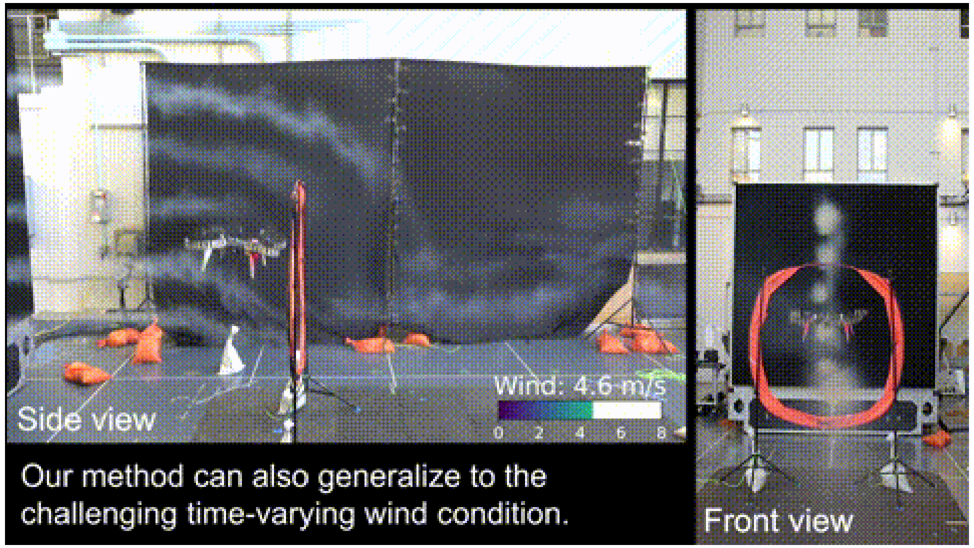
- How to constrain learning to (provably) satisfy requirements?
  - (certificates on behavior)

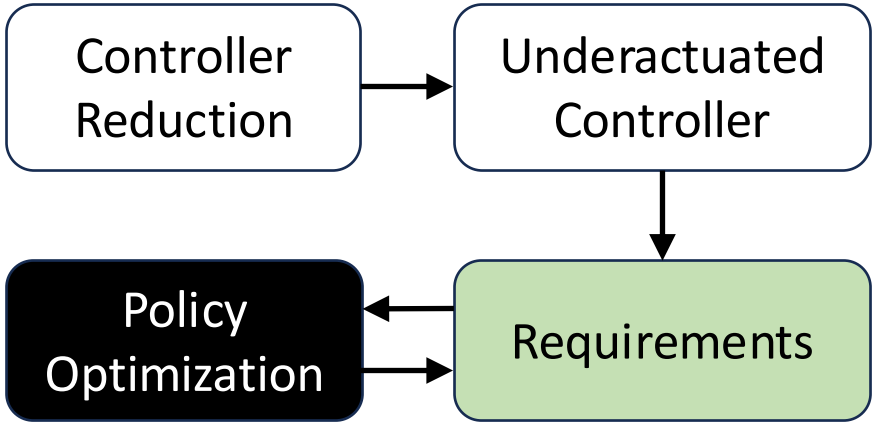
- How to exploit structure for faster learning?
  - (both computational & statistical)

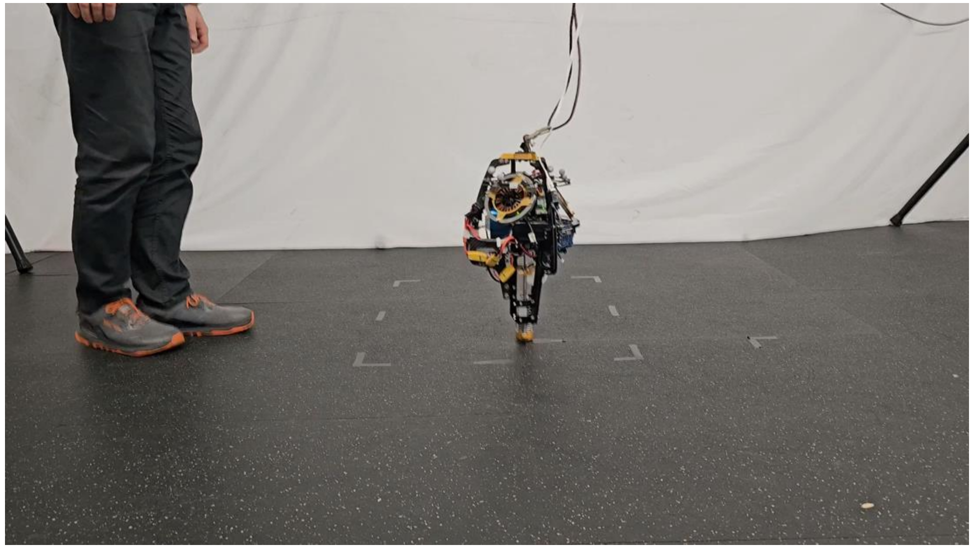
- How to interpret as a unified neurosymbolic AI system?

Rest of Talk

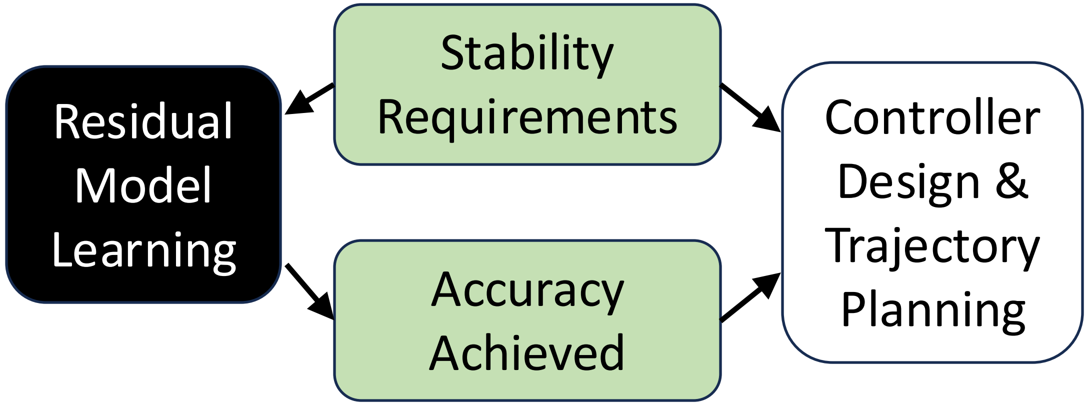

Residual Model Learning
Stability Requirements
Accuracy Achieved
Controller Design & Trajectory Planning

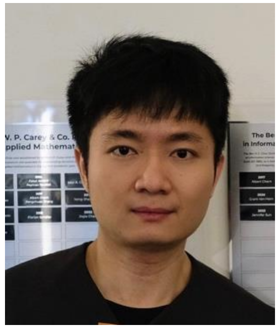

Wind: 4.6 m/s
0 2 4 6 8
Side view
Our method can also generalize to the challenging time-varying wind condition.
Front view


Controller Reduction
Underactuated Controller
Policy Optimization
Requirements



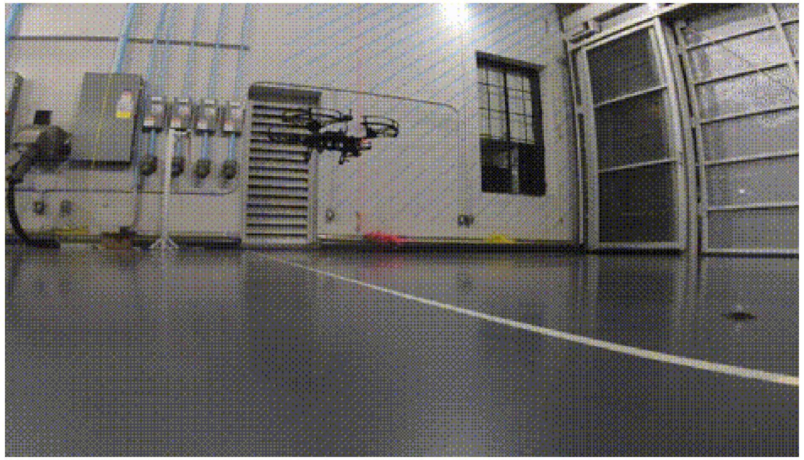
Neural Control Family





Guanya Shi

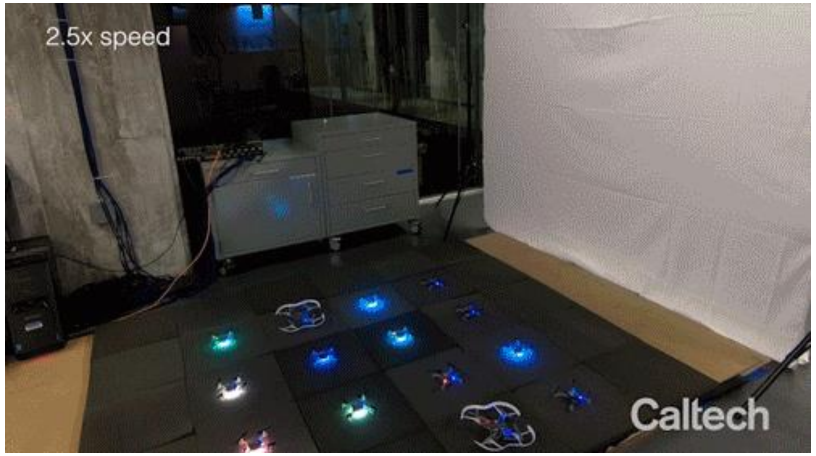
Neural Lander



<5 mm close to the ground

[ICRA'19]

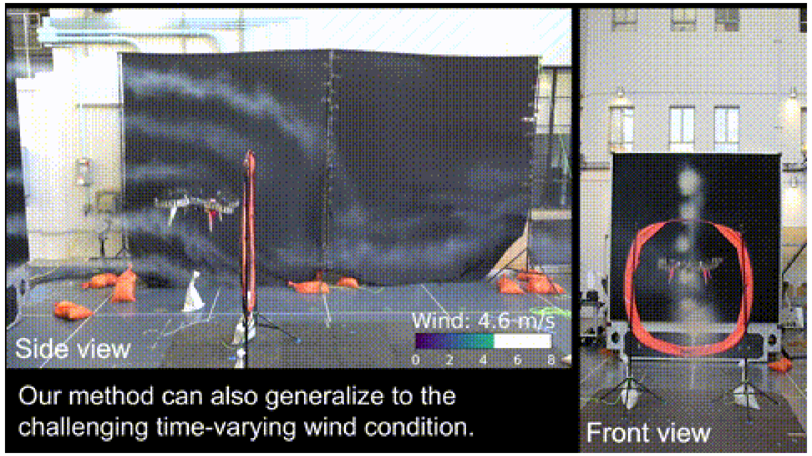
Neural Swarm



close-proximity heterogenous swarm

[ICRA'20][T-RO'21]

Neural Fly



precise flight in time-variant winds

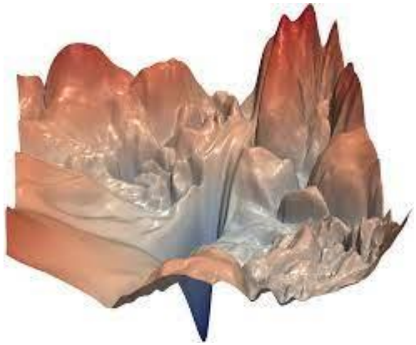
[NeurIPS'21][Science Robotics'22]

Where are the Challenges from?

- Uncertainty
  - Often nonlinear & nonstationary
- Computational & Data Efficiency
- Certificates of "Good Behavior"
  - Neural networks are hard to analyze

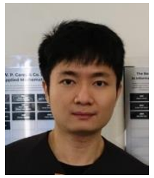
Caltech CAST wind tunnel

Crazyflie, weight 34g

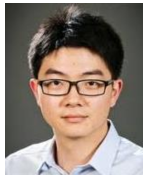


DNN landscape [Li et al., NeurIPS 2018]

Uncertain Boundary Interactions



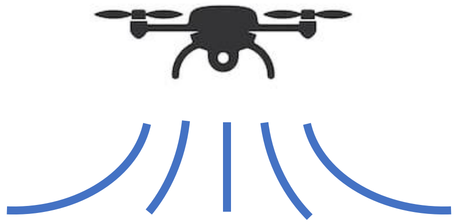
Guanya Shi



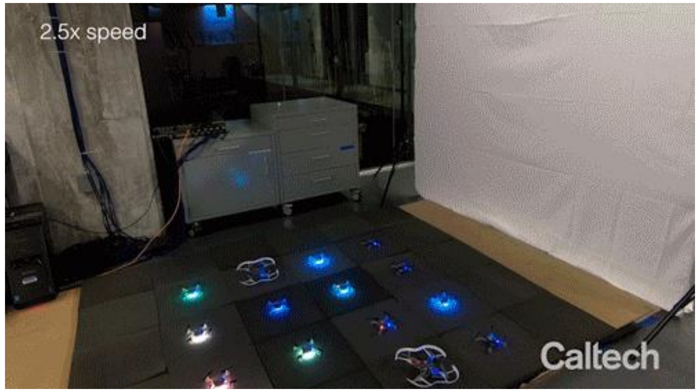
Xichen Shi

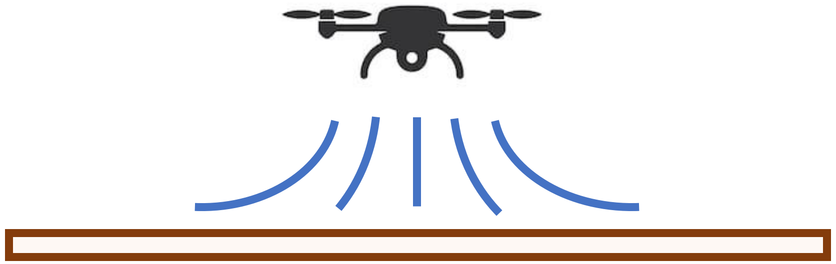
Michael O'Connell

Wolfgang Hoenig

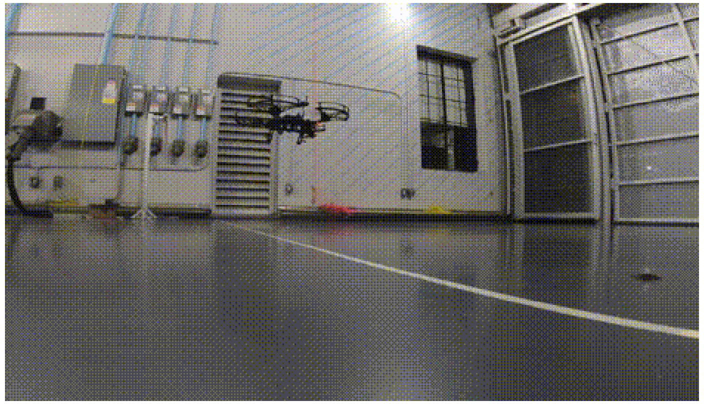


Downwash effect



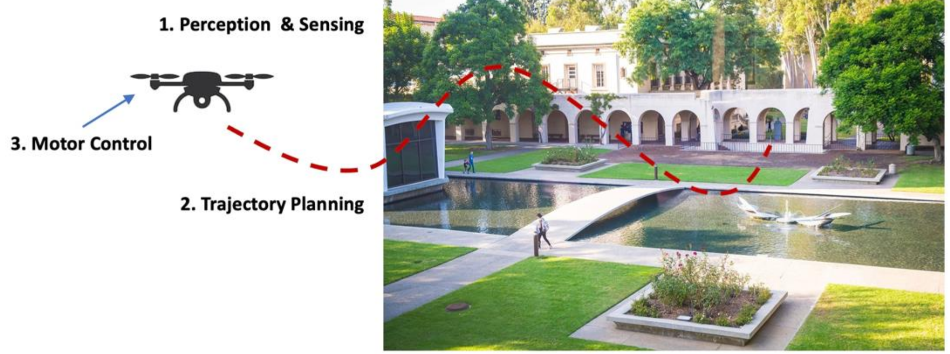


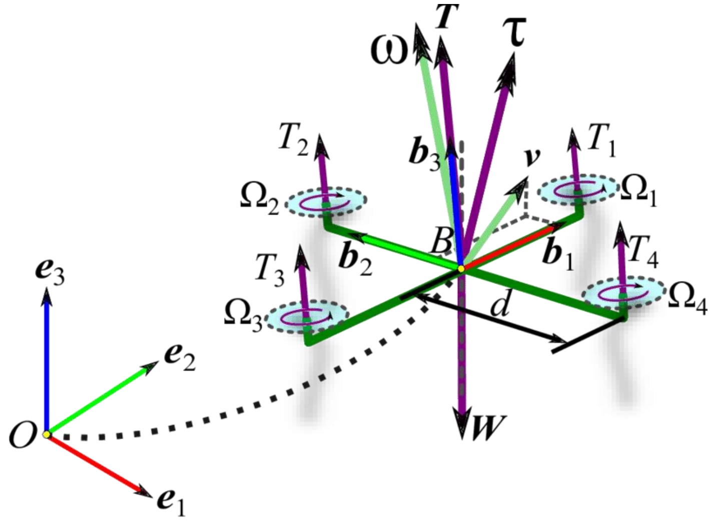
Ground effect



Neural Lander: Stable Drone Landing Control using Learned Dynamics, Guanya Shi, Xichen Shi, Michael O'Connell, et al. ICRA 2019
Neural-Swarm: Decentralized Close-Proximity Multirotor Control Using Learned Interactions, Guanya Shi et al., ICRA 2020
Neural-Swarm2: Planning and Control of Heterogeneous Multirotor Swarms using Learned Interactions, Guanya Shi et al., T-RO 1021

Model-Based Control





straightforward to model

Trajectory Planning

Stability | Robustness

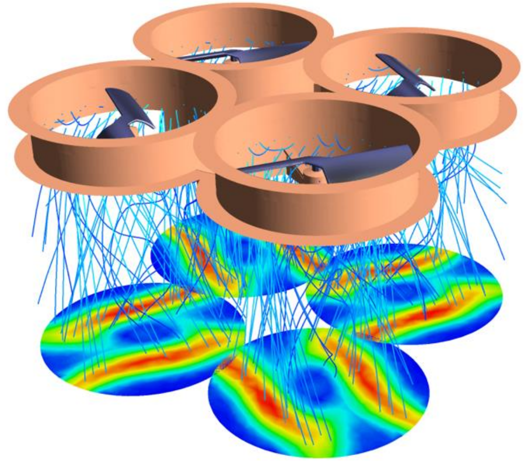
Nonlinear Controller

Stability | Accuracy

Dynamics Model

Symbolic Model

Residual Model



Very hard to model!

Model-Based Control

Change in State

Current Action (aka control input)

$$\dot{x} \equiv \frac{\partial x}{\partial t} = f(x, u) + \epsilon$$

Current State

Unmodeled Disturbance / Error

(Value Iteration is also contraction mapping)

Robust/Optimal Control (fancy contraction mappings)

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

Learning Residual Dynamics

f = nominal dynamics
g = learned dynamics

Change in State

Current Action (aka control input)

$$\dot{x} \equiv \frac{\partial x}{\partial t} = f(x,u) + g(x,u) + \epsilon(x,u)$$

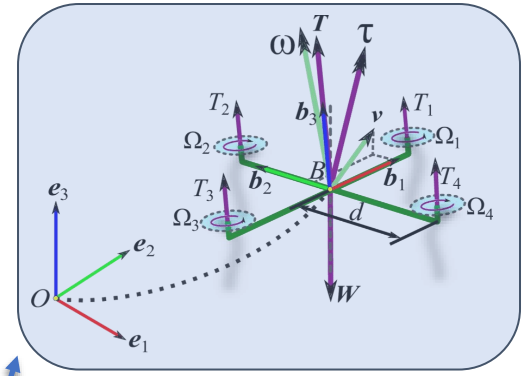
Current State

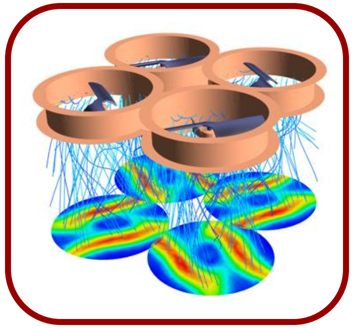
Unmodeled Disturbance / Error

Leverage robust/optimal control (fancy contraction mappings)

- Preserve stability (even using deep learning)
- Requires g Lipschitz & bounded error

Control System Formulation





- Dynamics:

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

Learn the Residual g

Symbolic Knowledge f

- Control:

$$\mathbf{f}_u = [0, 0, T]^\top$$
$$\boldsymbol{\tau}_u = [\tau_x, \tau_y, \tau_z]^\top$$
$$\begin{bmatrix} T \\ \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = \begin{bmatrix} c_T & c_T & c_T & c_T \\ 0 & c_T l_{\text{arm}} & 0 & -c_T l_{\text{arm}} \\ -c_T l_{\text{arm}} & 0 & c_T l_{\text{arm}} & 0 \\ -c_Q & c_Q & -c_Q & c_Q \end{bmatrix} \begin{bmatrix} n_1^2 \\ n_2^2 \\ n_3^2 \\ n_4^2 \end{bmatrix}$$

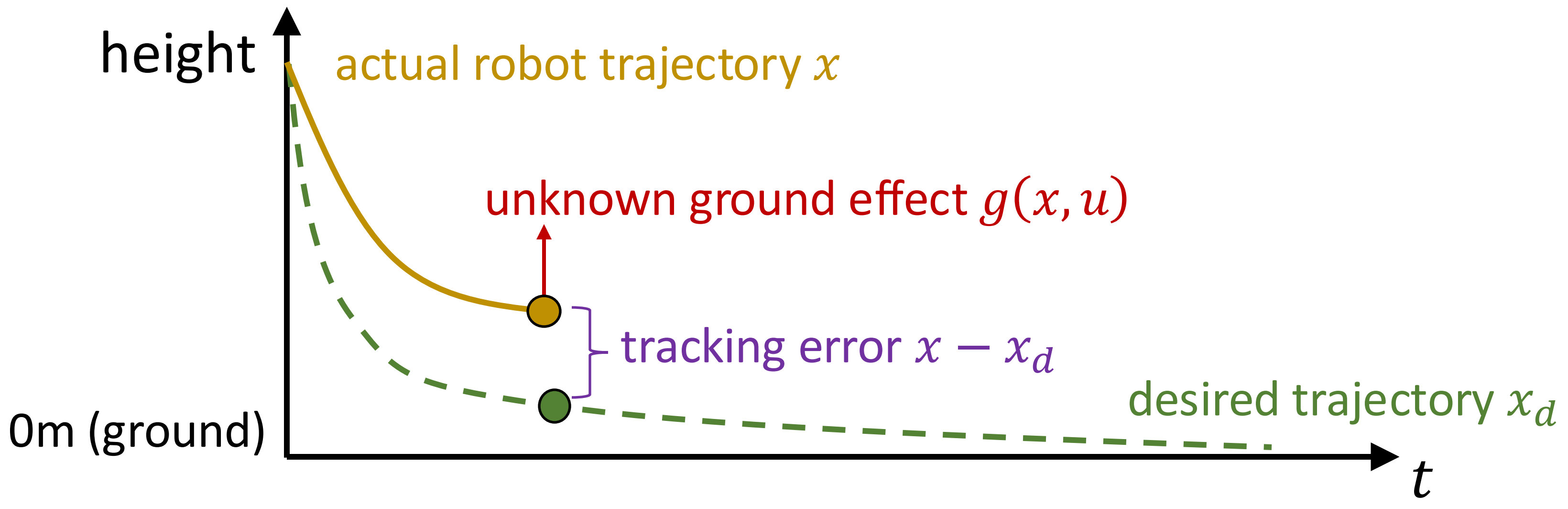
- Unknown forces & moments:

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

Learn the Residual g

Nonlinear Tracking Controller

- A simplified 1-d example



$$u = -K(x - x_d) + \text{nominal feedforward} + \text{learned feedforward } \hat{g}$$

- $-K(x - x_d)$: feedback
- nominal feedforward: depends on $f, \dot{x}_d$
- learned feedforward $\hat{g}$: compensate for boundary interaction

- **Desired Certificate:** ***guarantee stability and robustness if $\hat{g}$ is a NN***

Simple Integration Doesn't Work!

Nonlinear tracking controller (sketch):

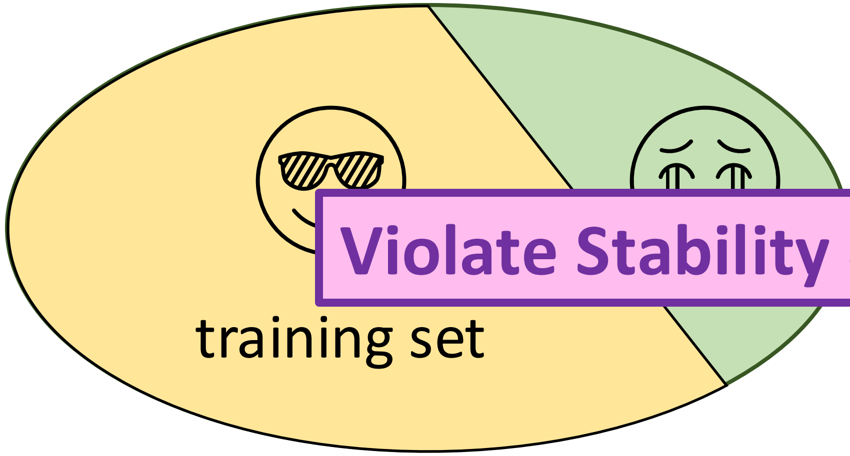
$$u = \pi(x, x_d, f + \hat{g}(x, u_{\text{old}}))$$

desired trajectory

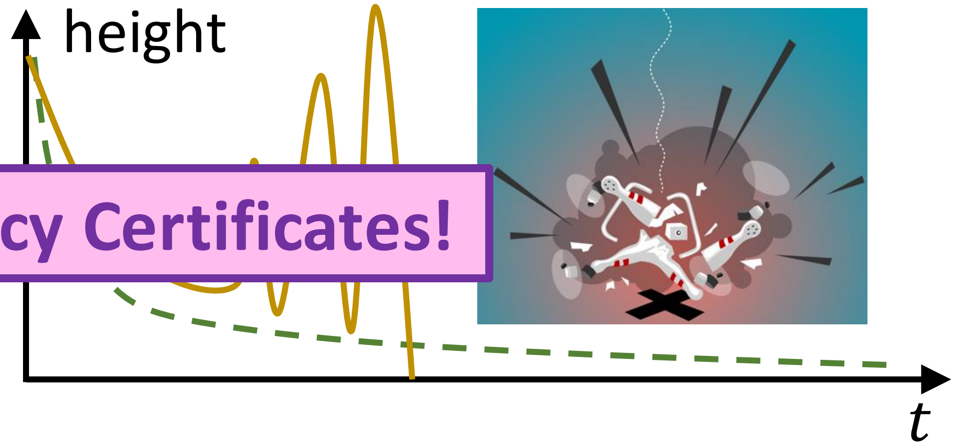
learned dynamics

- Train $\hat{g}$ using standard learning protocols
- Drone crashed!

$\hat{g}(x, u)$ can not generalize



$\hat{g}(x, u)$ has local "steepness"



Violate Stability & Accuracy Certificates!

Stability Certificate using Lipschitz NNs

Nonlinear tracking controller (sketch):

$$u = \pi(x, x_d, f + \hat{g}(x, u_{\mathrm{old}}))$$

desired trajectory | learned dynamics

"Exponential Stability" Theorem (informal) [ICRA'19]

Suppose $\hat{g}$ is **L-Lipschitz**. If $L < \gamma$, then:

$$\|x - x_d\| \to \frac{\epsilon}{\lambda - L\rho} \quad \text{exponentially fast}$$

- ϵ: approximation error ($\|g - \hat{g}\|_\infty$)
- λ: control gain
- ρ: time delay

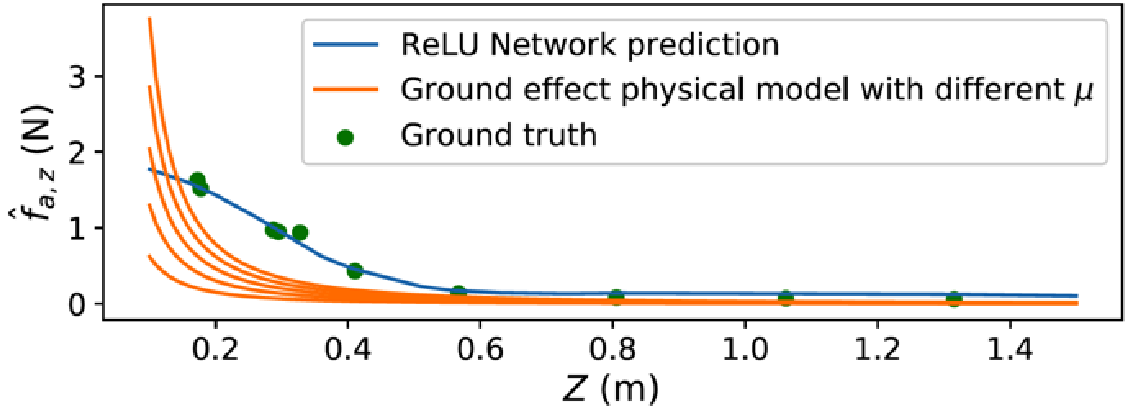
- γ: a system-dependent threshold. $L < \gamma$ is **necessary**!
- Idea: show $u_{k+1} = \pi(\cdot,\cdot, f + \hat{g}(\cdot, u_k))$ is a contraction

γ_1

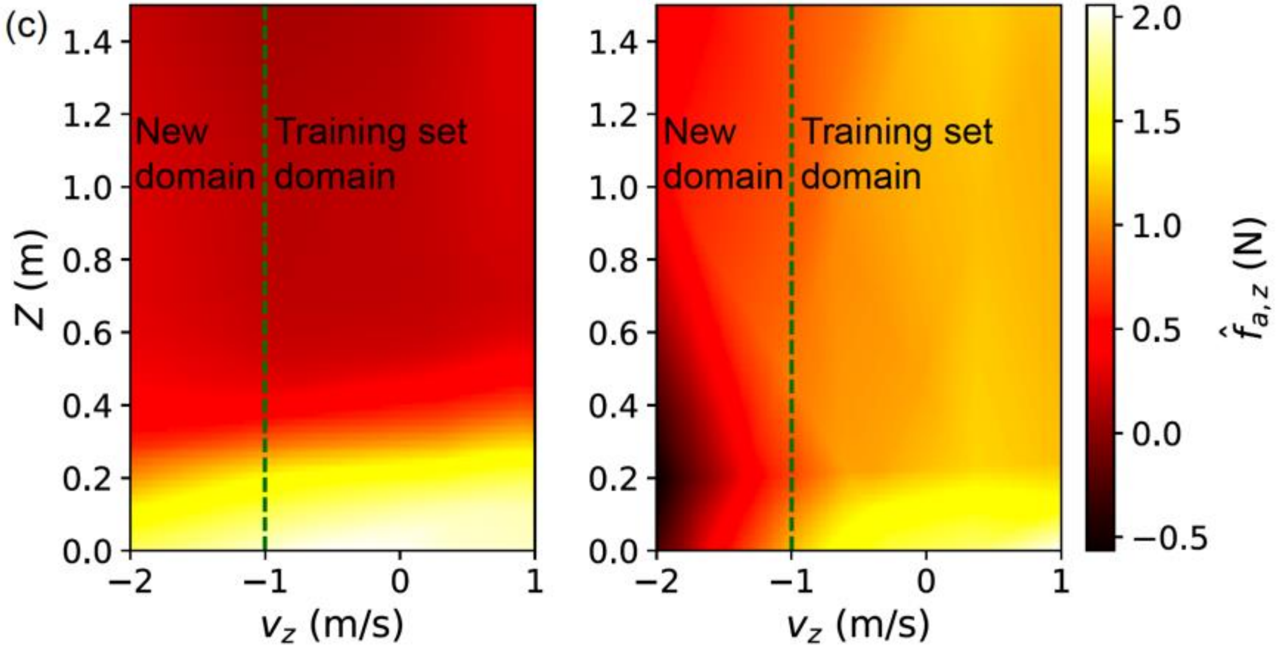
γ_2

Lipschitz Constrained Dynamics Learning

- Spectral Normalization → $L < \gamma$
- Applicable for arbitrarily large DNNs
- Graceful generalization

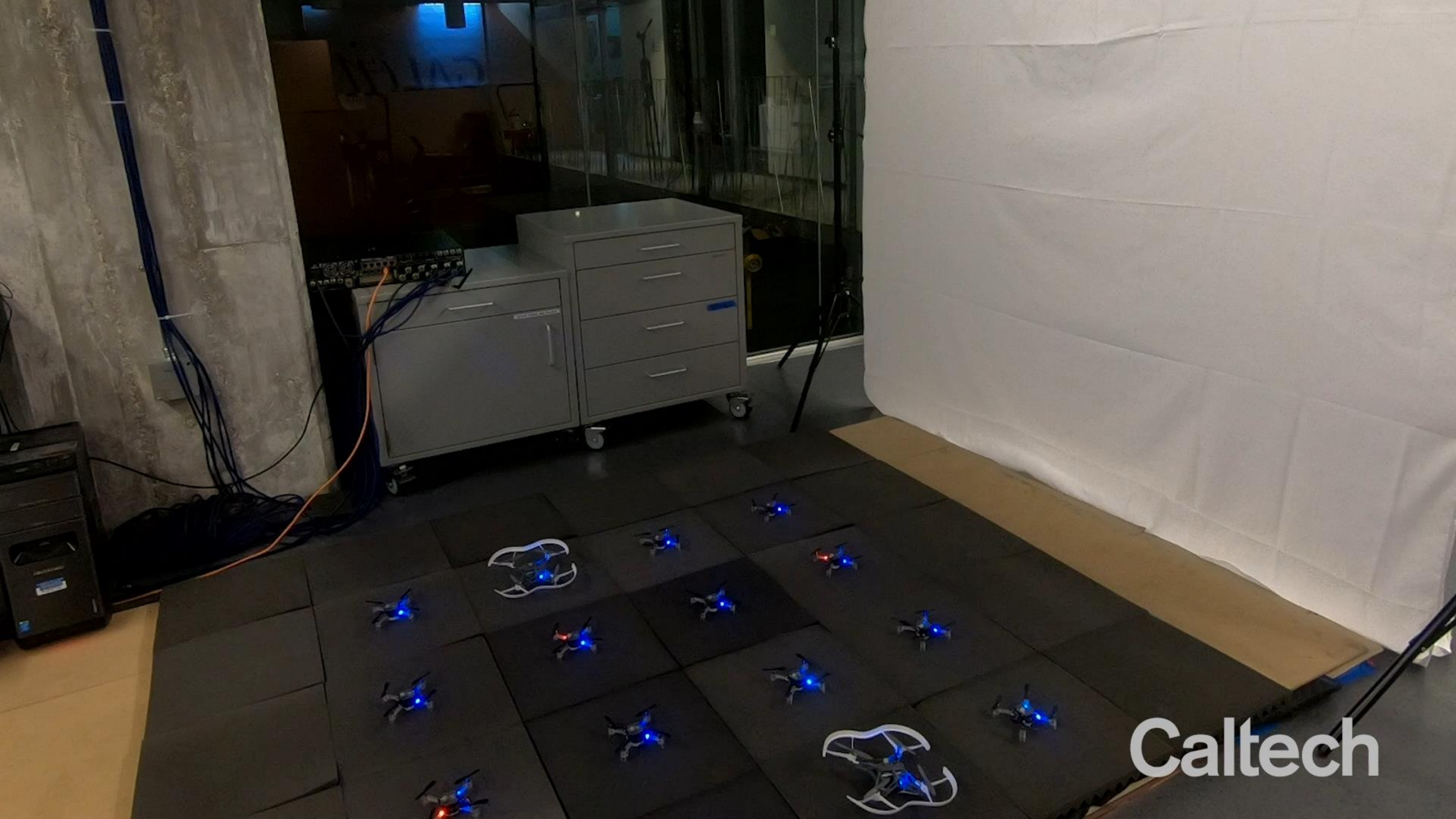


compare with fluid dynamics model

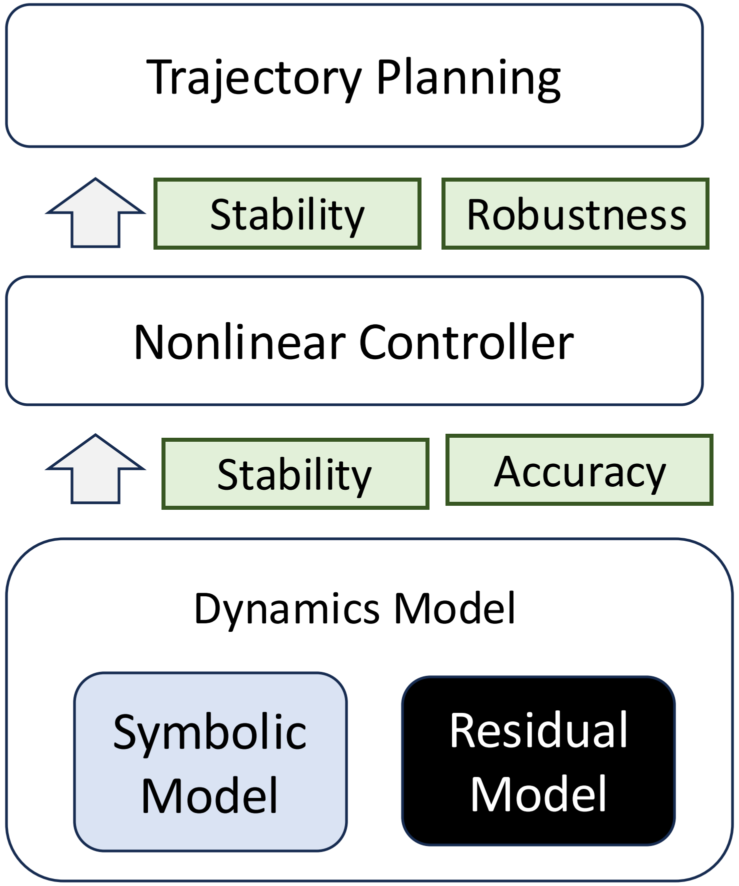


2D heatmap of the learned $\hat{g}$

Neural Lander: Stable Drone Landing Control using Learned Dynamics, Guanya Shi, Xichen Shi, Michael O'Connell, et al. ICRA 2019



Story so Far



Challenges Tackled:

Uncertainty:

Learn residual models $g(x, u)$

Efficiency:

$g(x, u)$ is tiny (4-layer NN)
5 minutes of training data

Certificates:

Lipschitz on $g(x, u)$ → Lyapunov Stability Certificate

Lipschitz constraint is necessary: $\gamma \approx 16$	
w/ constraint $L < 16$	w/o $L \approx 8000$ **Drone crashed!**

Neural-Fly

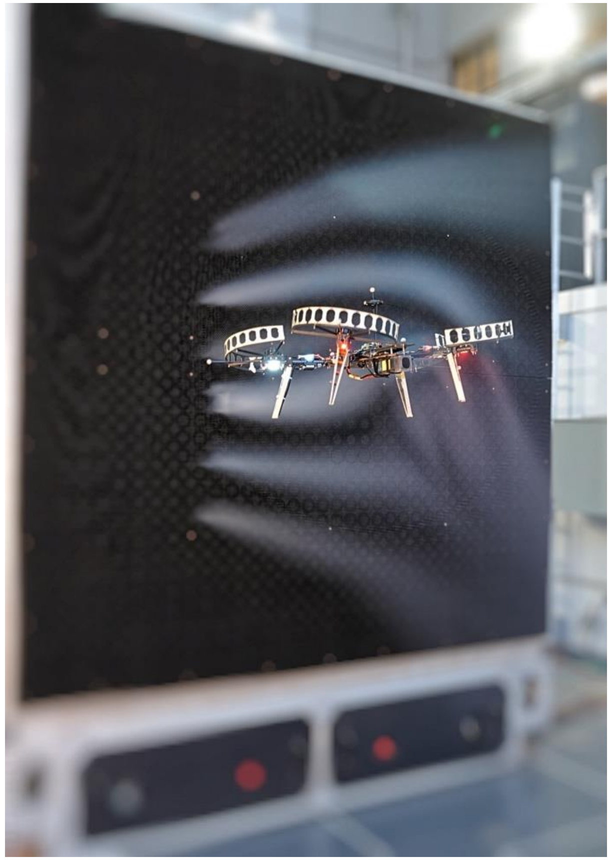
- g is governed by the environment condition $c(t)$:

$$\dot{x} = f(x, u) + g(x, c(t))$$

unknown

- Baseline's performance:

Kármán vortex street



Key Idea: Meta-Adaptive Control

$$g(x, c) \approx \phi(x) a(c)$$

shared representations
(NOT depend on c)

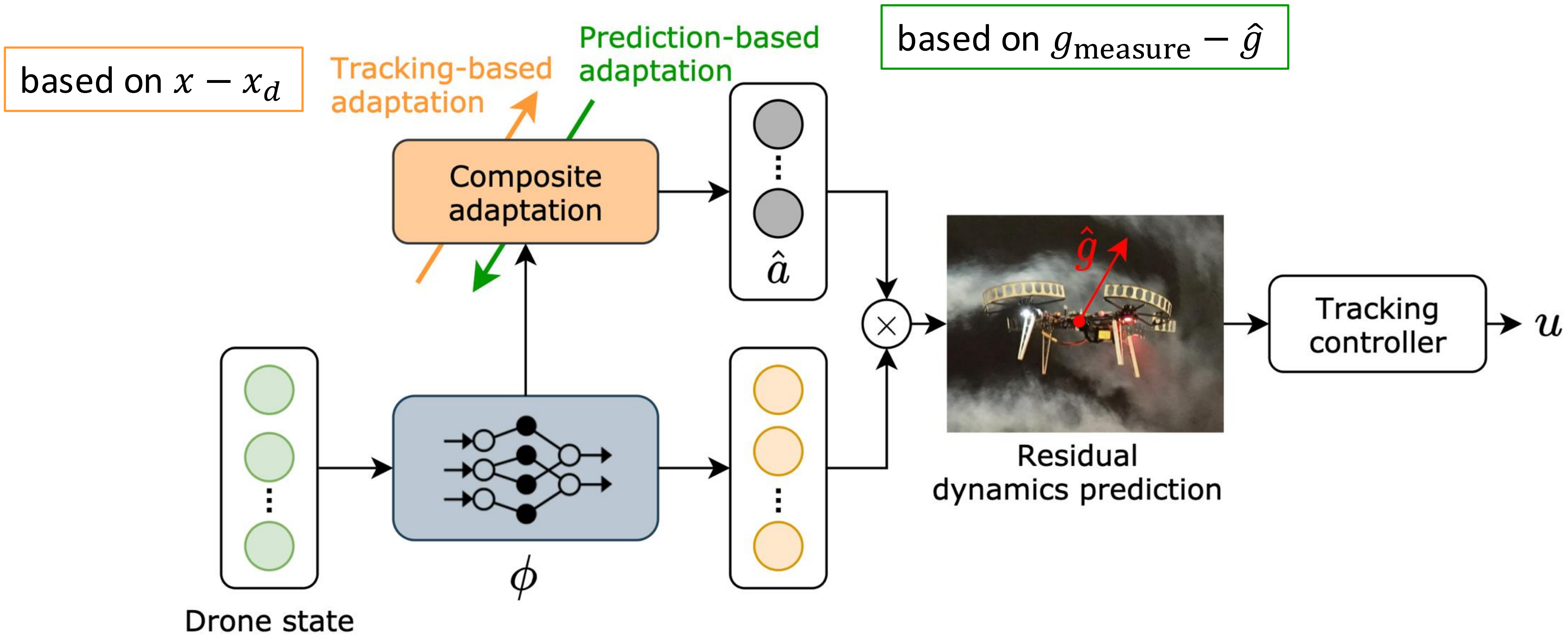
Domain Invariant Meta-Learning

environment-specific linear coefficients

Adaptive Control

Neural-Fly Enables Rapid Learning for Agile Flight in Strong Winds, O'Connell, Shi, et al., Science Robotics 2022
Meta-Adaptive Nonlinear Control: Theory and Algorithms, Shi et al., NeurIPS 2021
Hierarchical Meta-learning-based Adaptive Controller, Xie et al., ICRA 2024

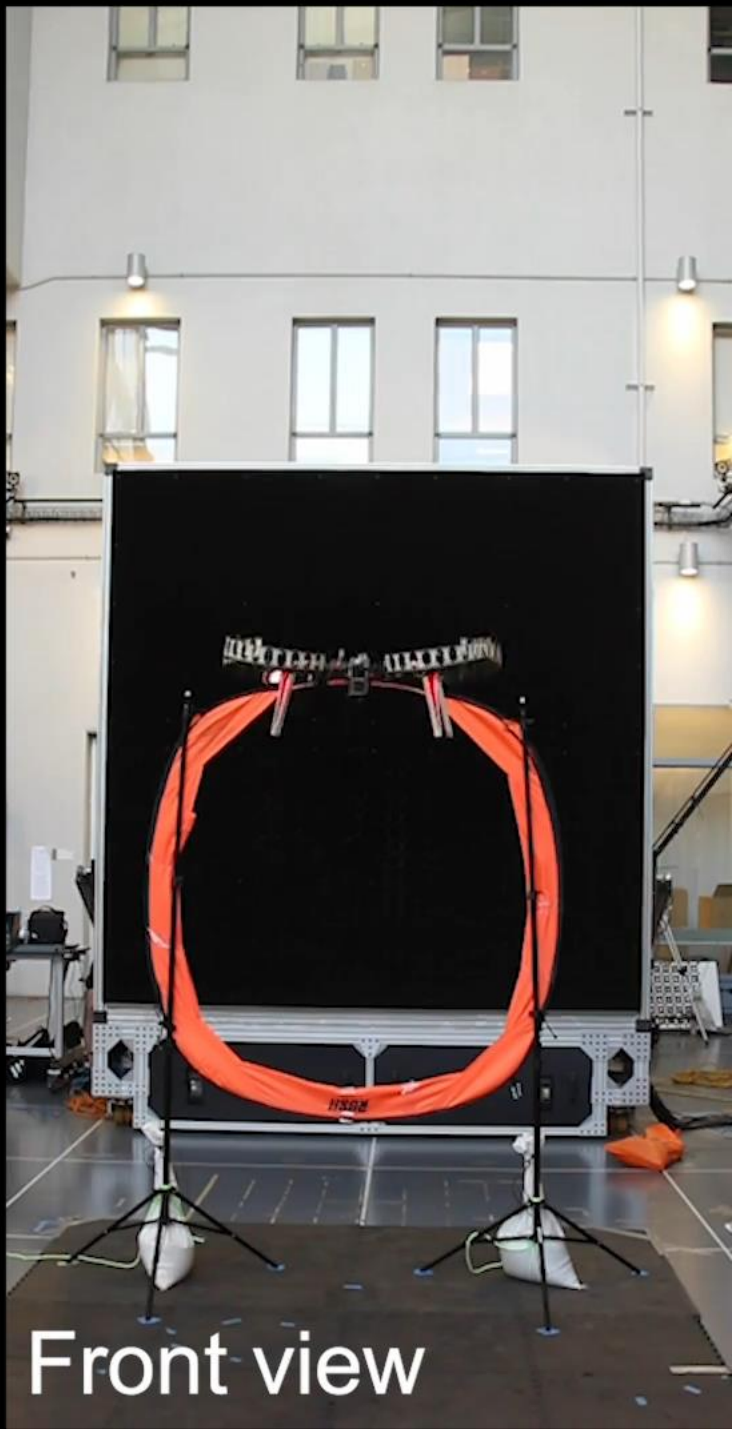
Stable and Robust Adaptive Control



Theorem (informal) [O'Connell* & Shi* et al., Science Robotics'22]

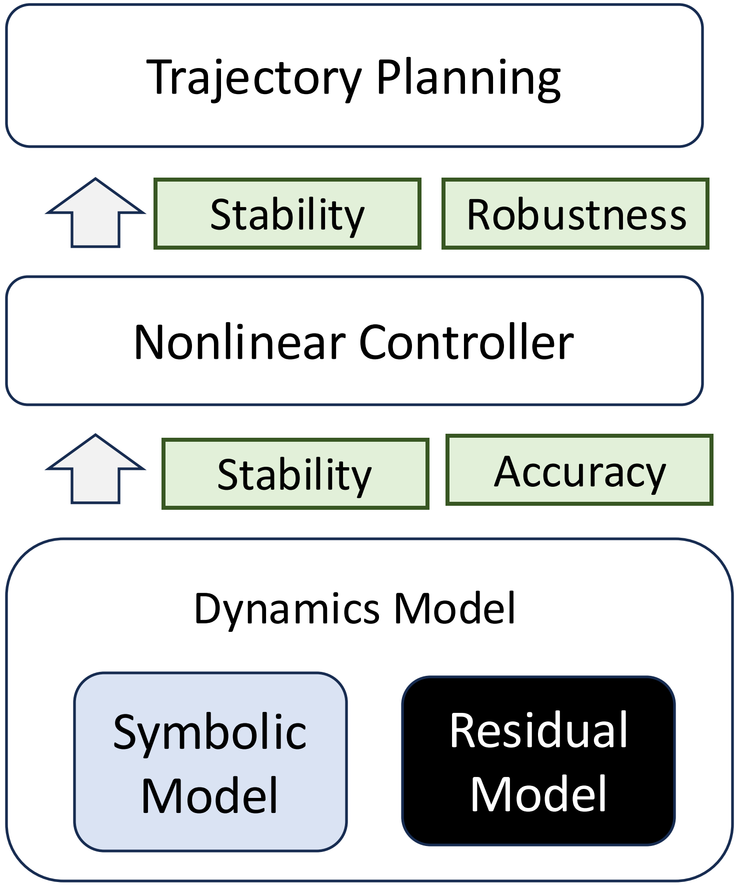
$$\|x - x_d\| \to \sup_t O(\|\epsilon(t)\| + \|\dot{a}^*(t) + \lambda a^*(t)\|) \text{ exponentially}$$

imperfect learning



Our method can also generalize to the challenging time-varying wind condition.

Neural Control Family



Uncertainty:

Learn residual models $g(x,u)$

Meta-learning for general representation

Efficiency:

$g(x,u)$ is tiny (4-layer NN)
5 minutes of training data

Adaptive control for varying environments

Neural-Fly

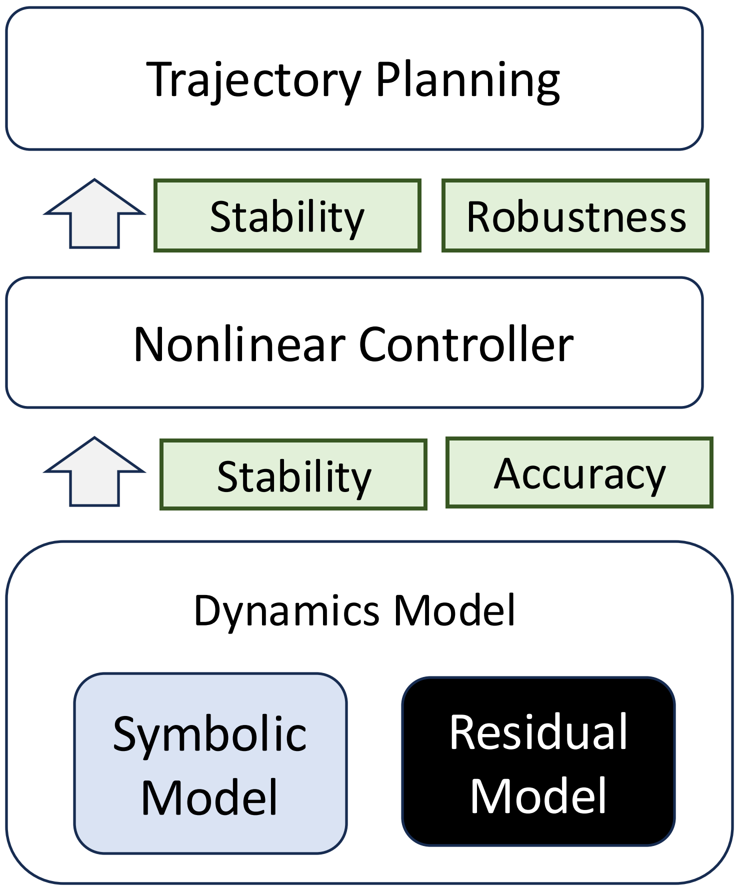
Certificates:

Lipschitz on $g(x,u)$ → Lyapunov Stability Certificate

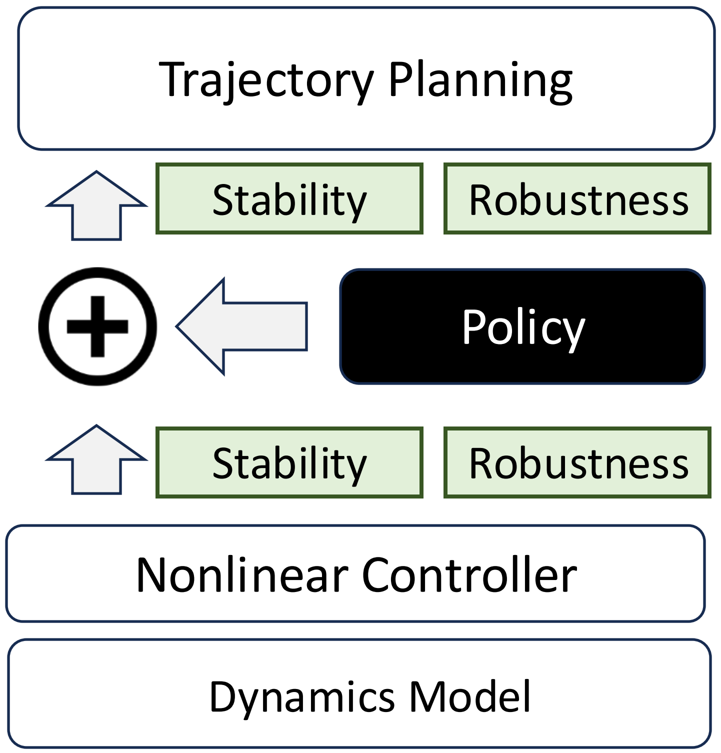
Lipschitz constraint is necessary: $\gamma \approx 16$	
w/ constraint $L < 16$	w/o $L \approx 8000$ **Drone crashed!**

Aside: Residual Policy Learning

Residual Model Learning



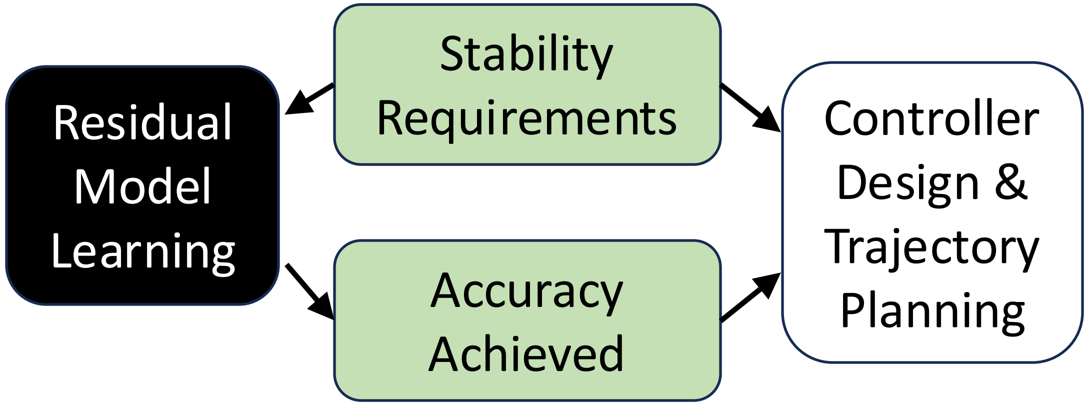
Residual Policy Learning

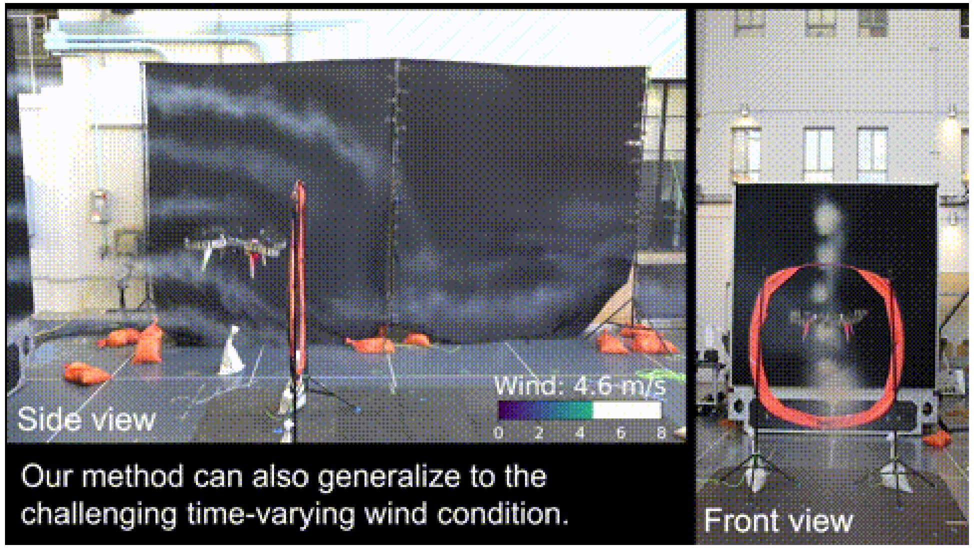


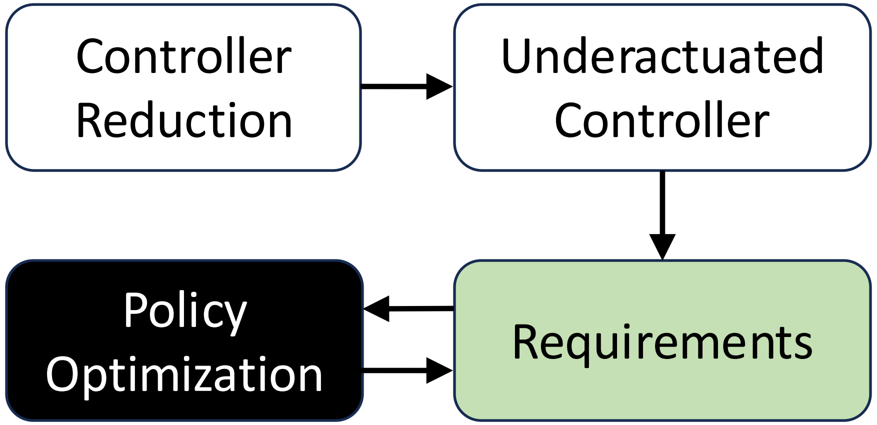
https://arxiv.org/abs/1606.00968 https://arxiv.org/abs/1903.08738

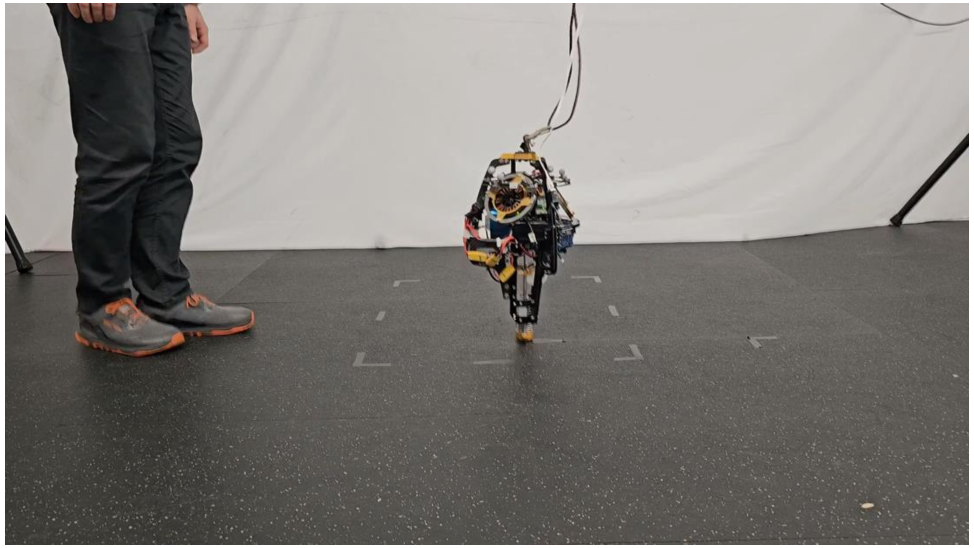
https://arxiv.org/abs/1905.05380 https://arxiv.org/abs/1907.05431

Rest of Talk

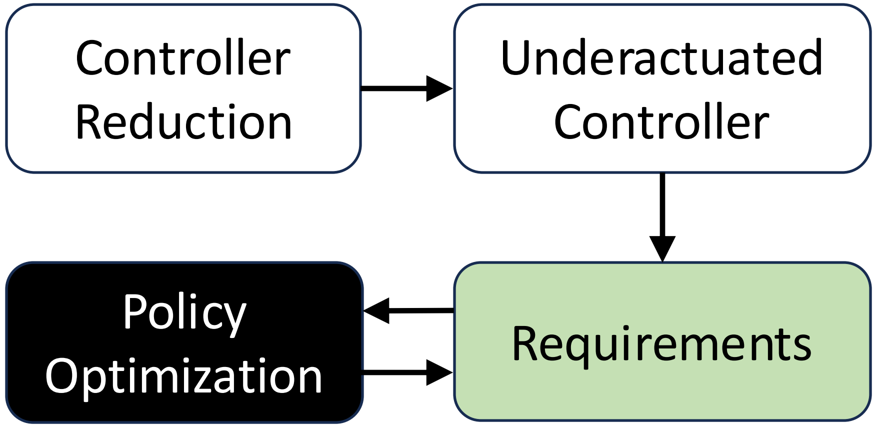

Residual Model Learning
Stability Requirements
Accuracy Achieved
Controller Design & Trajectory Planning

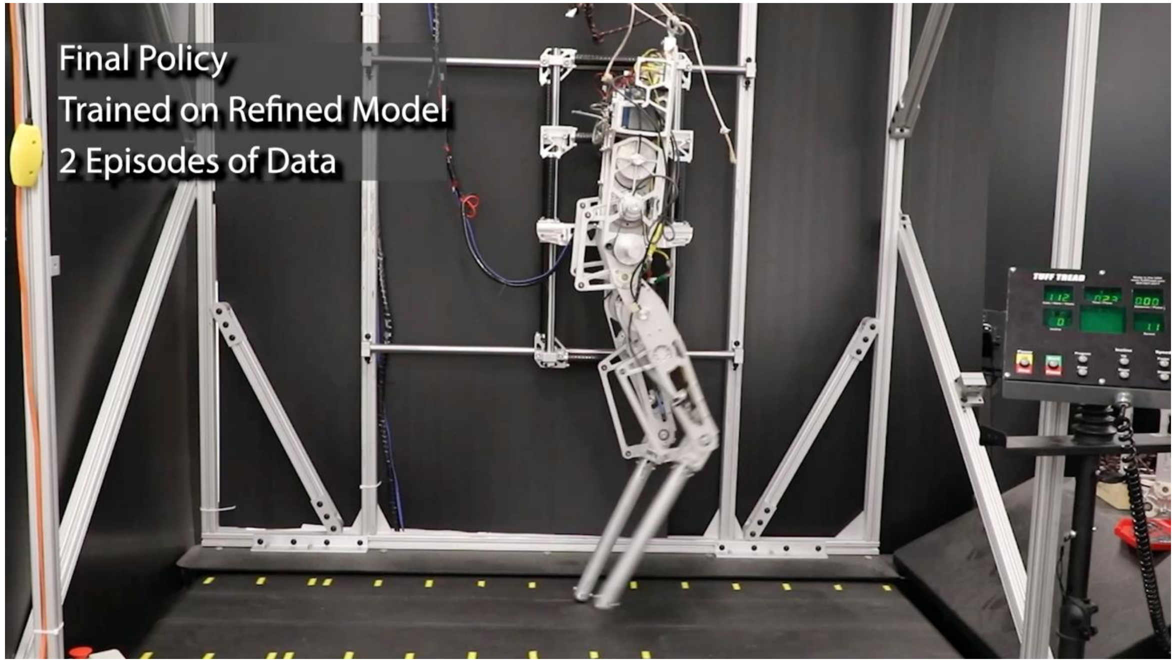

Side view
Wind: 4.6 m/s
0 2 4 6 8
Our method can also generalize to the challenging time-varying wind condition.
Front view

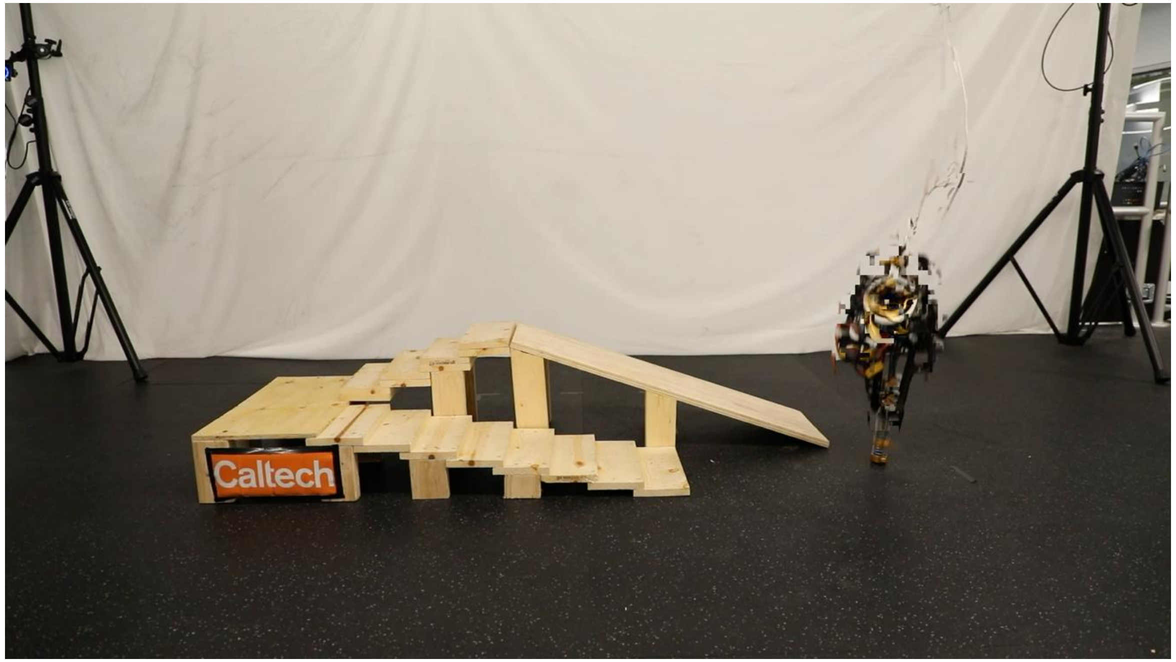

Controller Reduction
Underactuated Controller
Policy Optimization
Requirements



Motivation:
Underactuated Agility





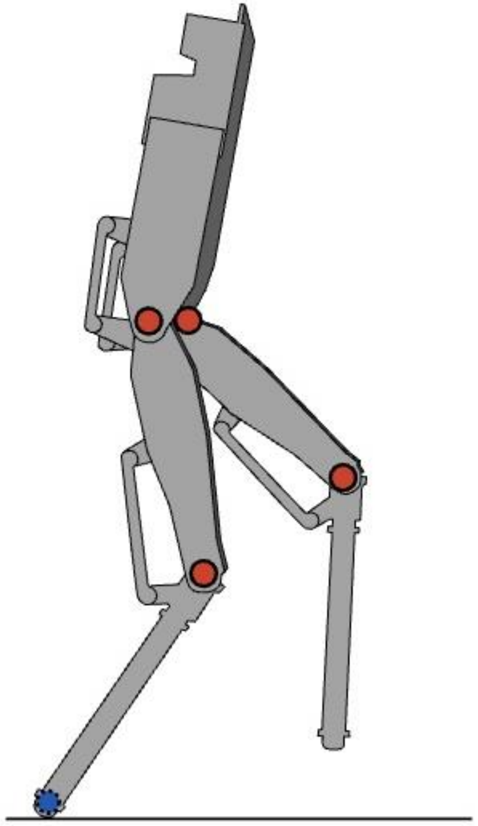


Actuated vs Unactuated Dynamics

Actuated η

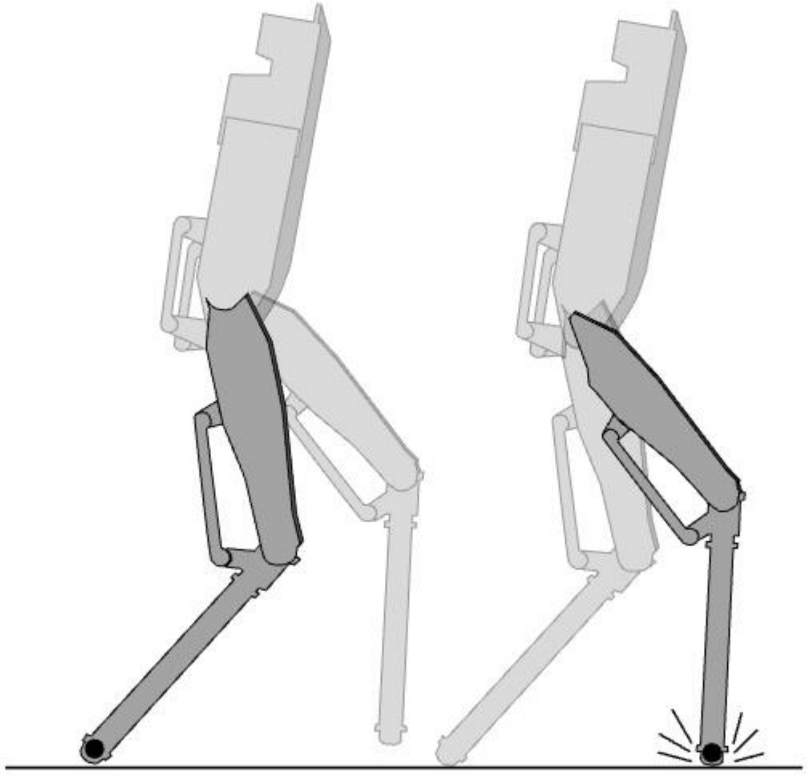
Unactuated z

(Null space of control matrix)



Continuous

(Not Emphasized in this Talk)



Discrete

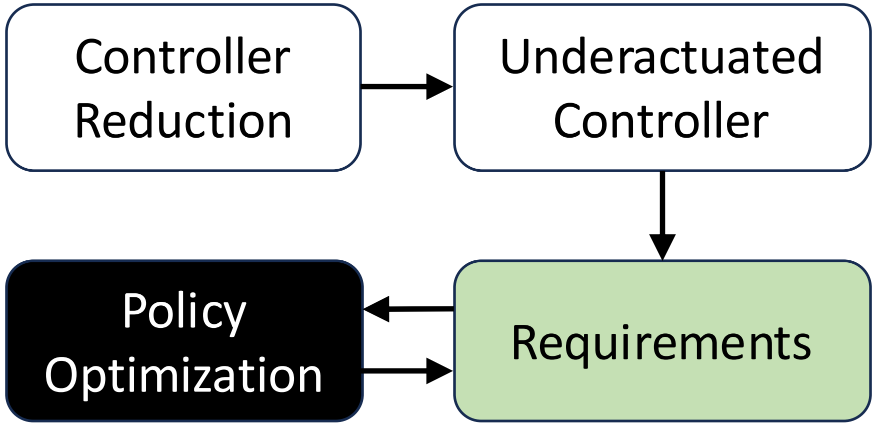
Key Challenge

If always on the controllable manifold → standard policy optimization

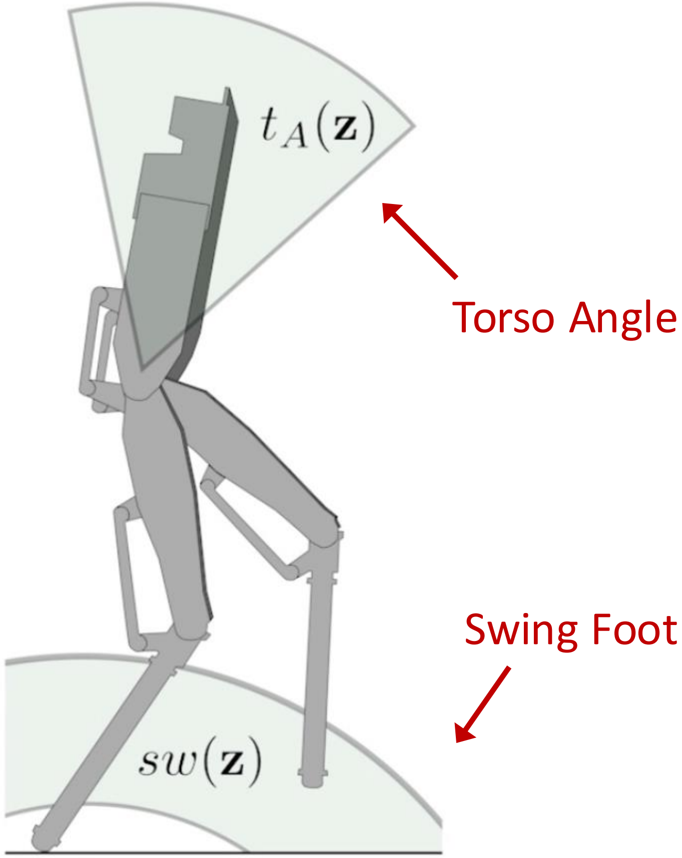
I.e., ignore the uncontrollable parts
(null space of controllable manifold)

Key idea: “cancel out” the null space

Requirement:
Cancel out effect of unactuated component



Example:

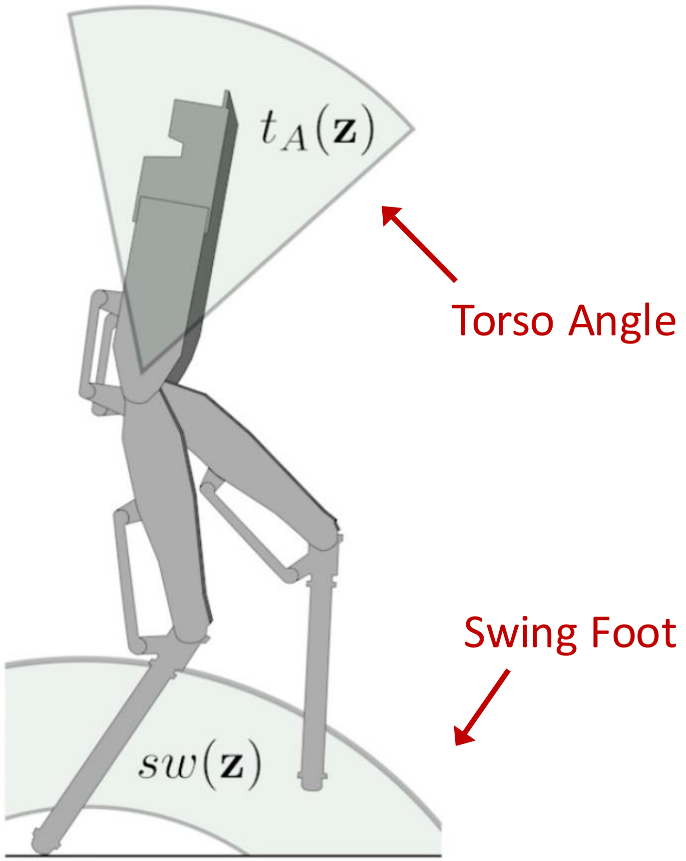


Conservative Specification
(sufficient conditions)

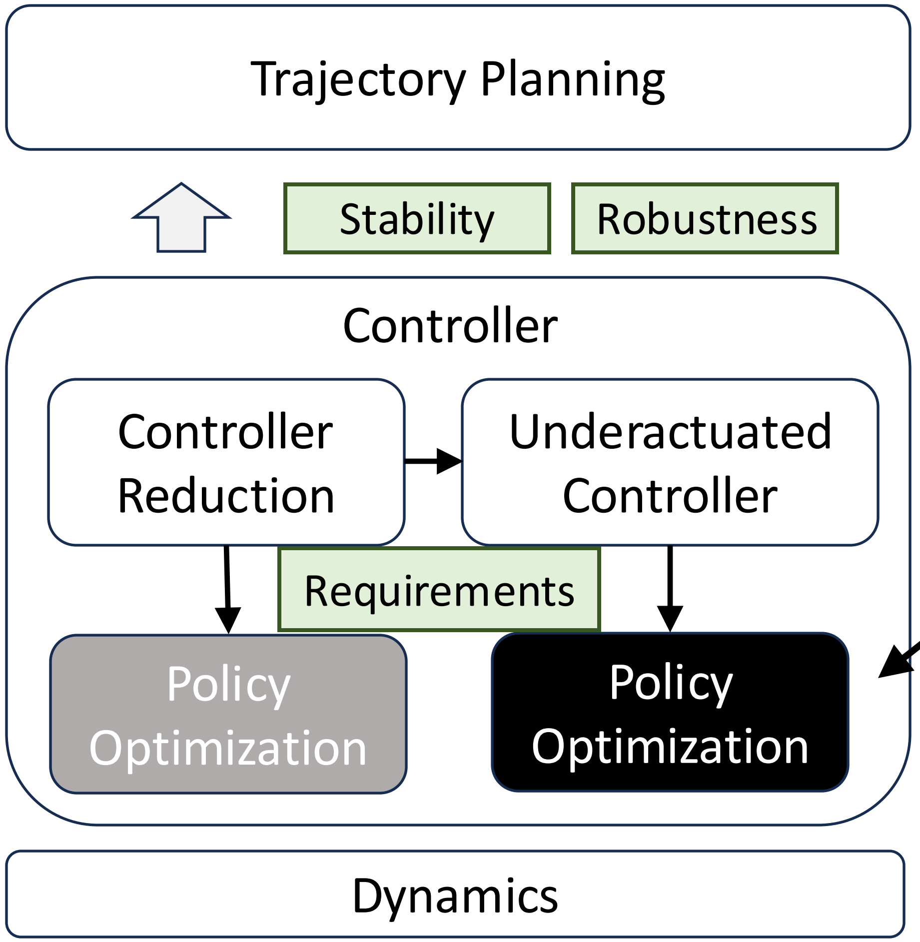
Treat Policy as Continuous-time Map

- Assume dynamics model is known

- Learn policy that satisfies specification



Policy as Neural Ordinary Differential Equation



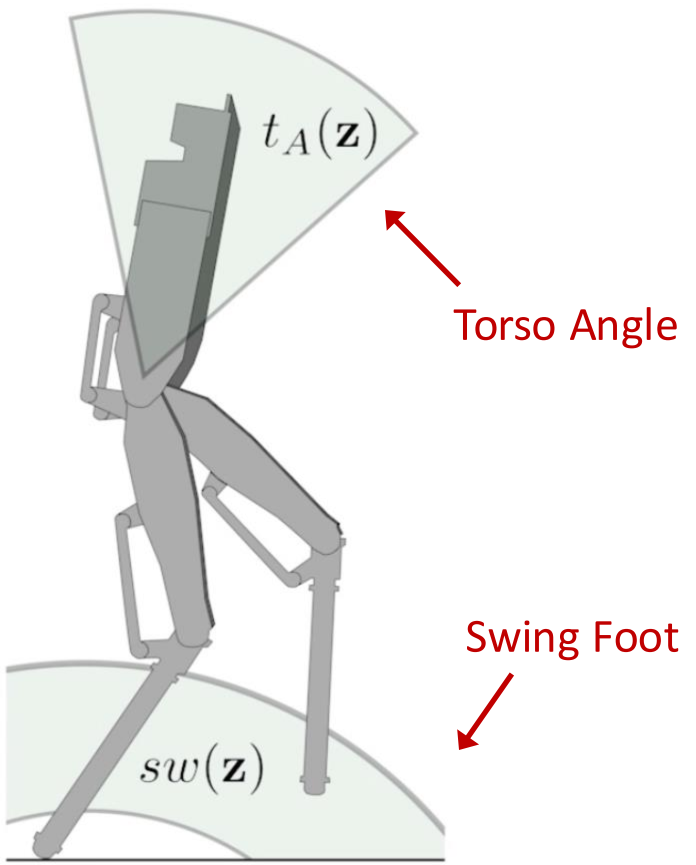
Change in state (sent to lower-level controller)

$$\frac{\partial \eta}{\partial t} = \pi(\eta, x)$$

Current state (actuated)

Extra Context (optional)

Capture Requirements via Potential Function



$V(\eta)$

Large if requirements are being violated

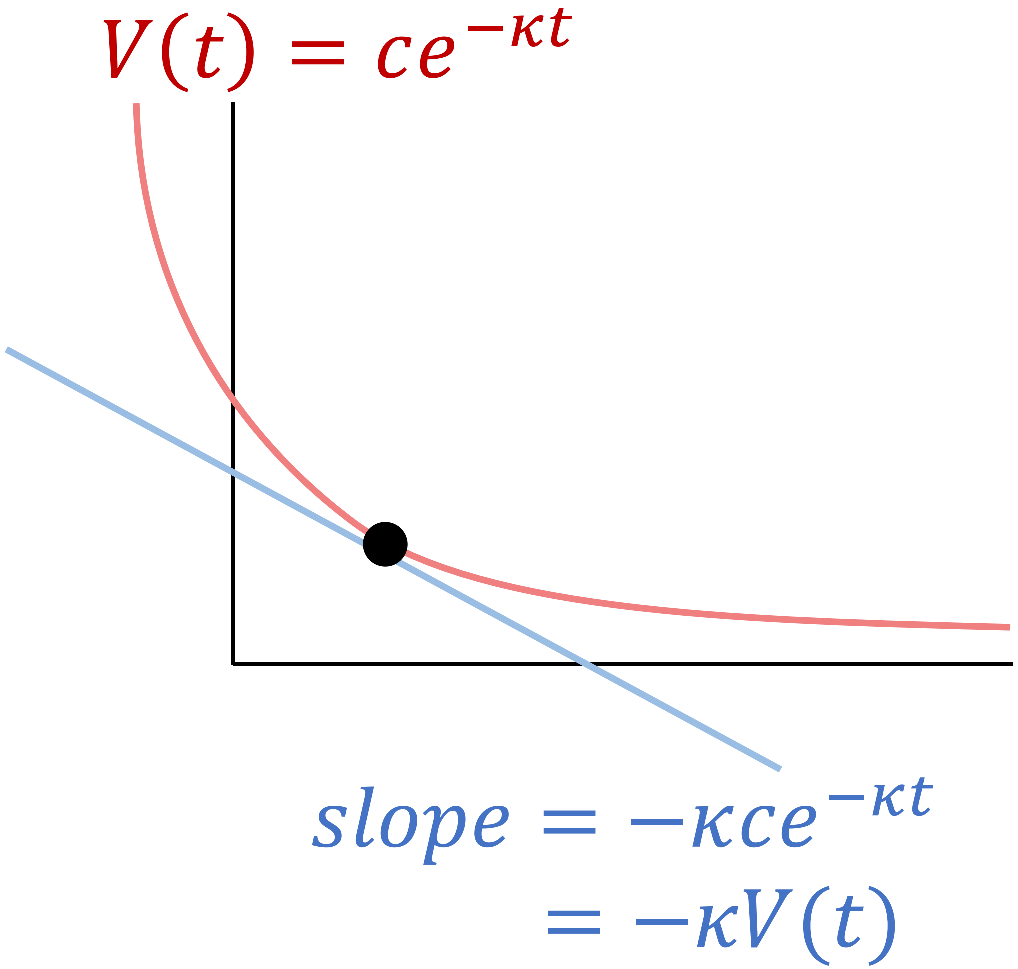
E.g., Torso angle destabilizing

Exponential Stability & Contraction

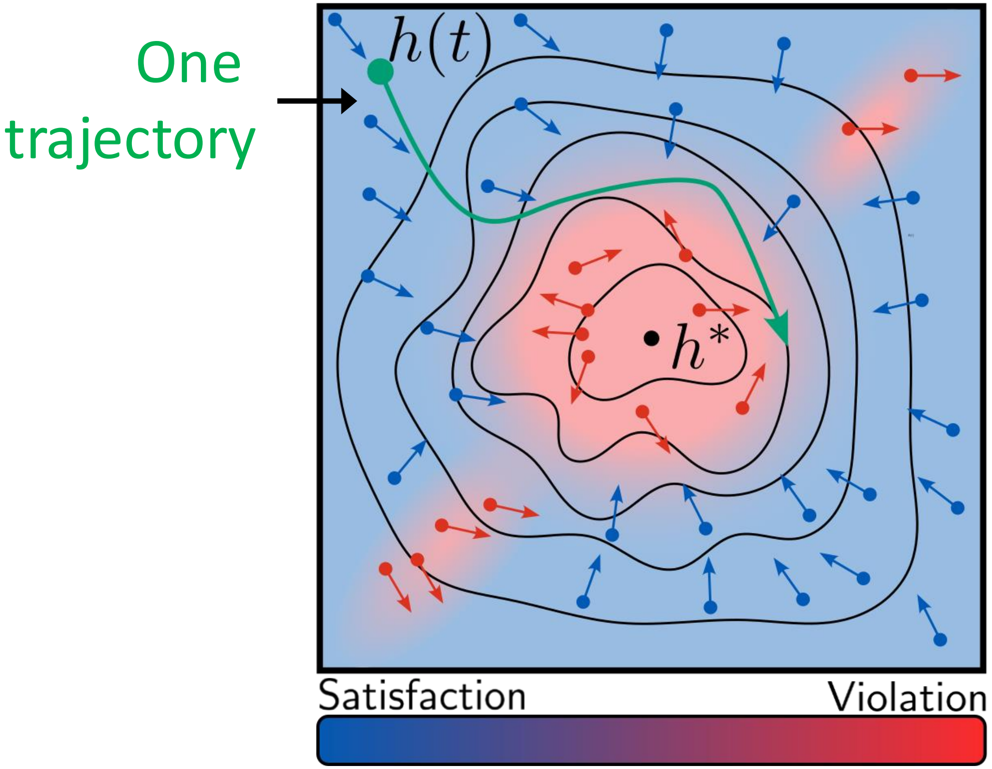
Goal: $V(h(t)) \leq V(h(0))e^{-\kappa t}$

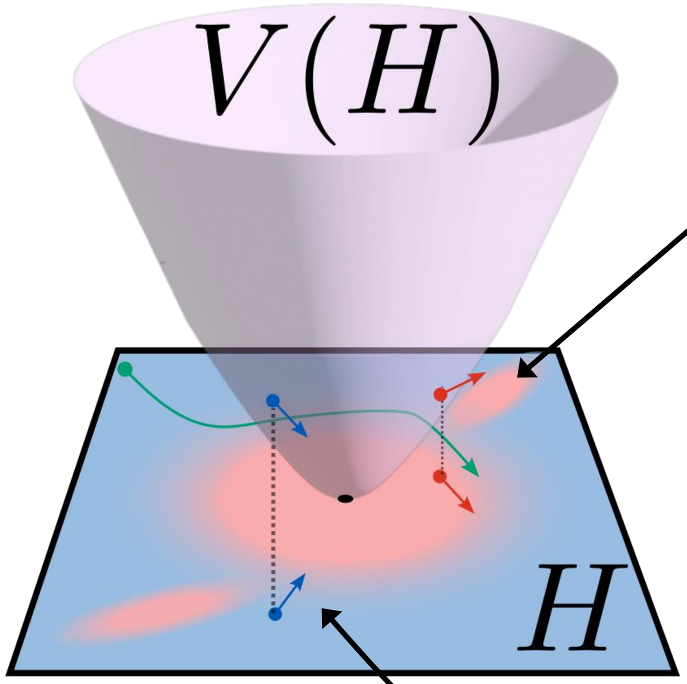
Key invariant: $\frac{\partial V}{\partial h}^{T} \pi(h) \leq -\kappa V(h)$

Benefits: Fast convergence & Robustness



Certification via Contraction Condition





Violation:

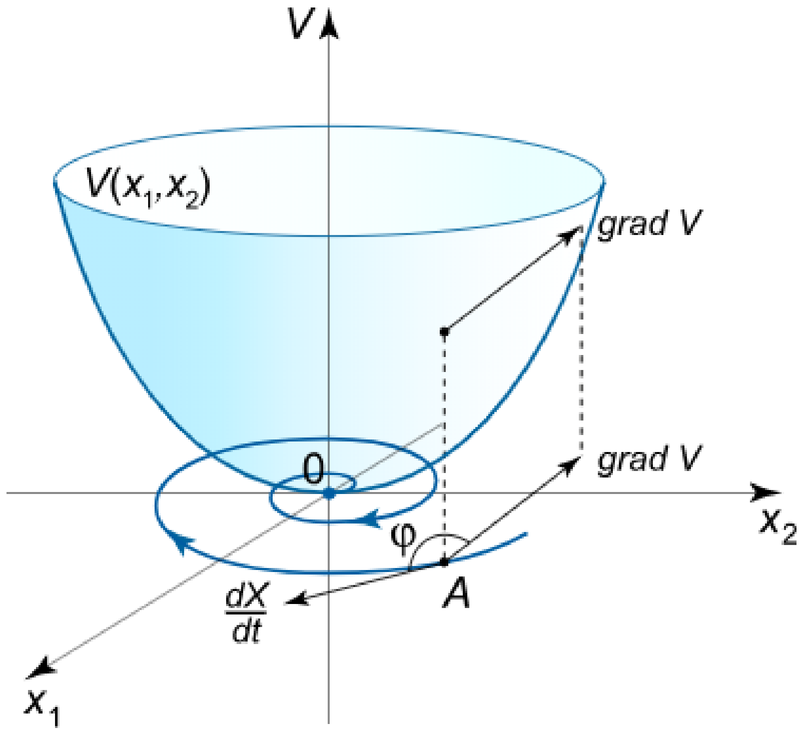
$$\frac{\partial V^T}{\partial h} \pi(h) > -\kappa V(h)$$

Satisfaction:

$$\frac{\partial V^T}{\partial h} \pi(h) \leq -\kappa V(h)$$

Contraction Satisfied Everywhere => Exponential Stability!

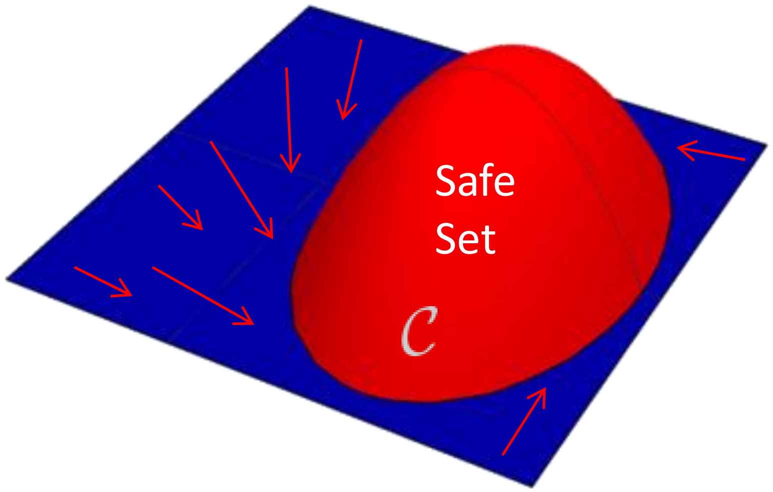
Types of Contraction



Stability

Towards a Set

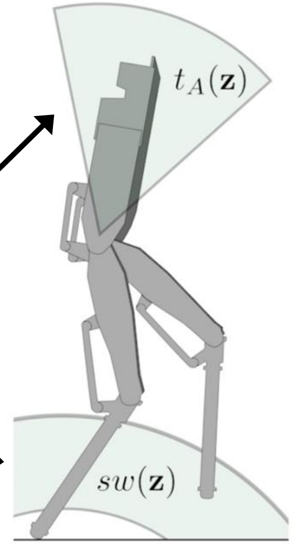
"Lyapunov Functions"



Forward Invariance

Never Leave a Set

"Barrier Functions"



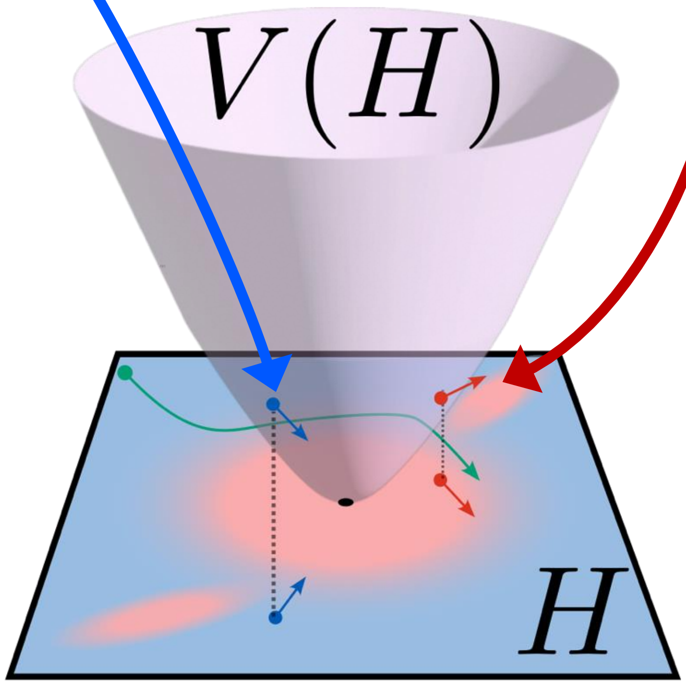
Lyapunov Loss

- Point-wise Lyapunov Loss

$$L_V(h) \equiv \max\left\{0, \underbrace{\frac{\partial V^T}{\partial h} f(h) + \kappa V(h)}_{\text{Contraction condition violation}}\right\}$$

- Lyapunov Loss:

$$L_V(\theta) \equiv \boldsymbol{E}_{h_0}\left[\int L_V(h(t))dt\right]$$



Achieving zero Lyapunov Loss (almost) everywhere implies exponential stability!

https://arxiv.org/abs/2202.02526

LyaNet

A Lyapunov Framework for Training Neural ODEs

Ivan
Jimenez Rodriguez

1. Interpret requirements as potential function: $V(h(t))$

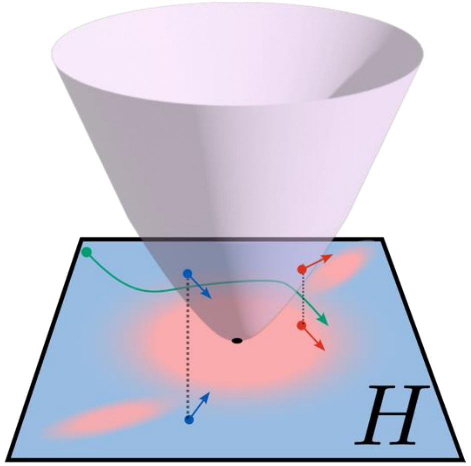
2. Instantiate (point-wise) Lyapunov Loss:

$$L_V(h) \equiv \max\left\{0, \frac{\partial V}{\partial h}^T \pi(h) - \kappa V(h)\right\}$$

3. Optimize Lyapunov Loss everywhere

[ICML 2022] https://arxiv.org/abs/2202.02526

Optimization Considerations



$$L_V(\theta) \equiv \boldsymbol{E}_{h_0}\left[\int L_V\big(h(t)\big)dt\right]$$

Lyapunov Loss

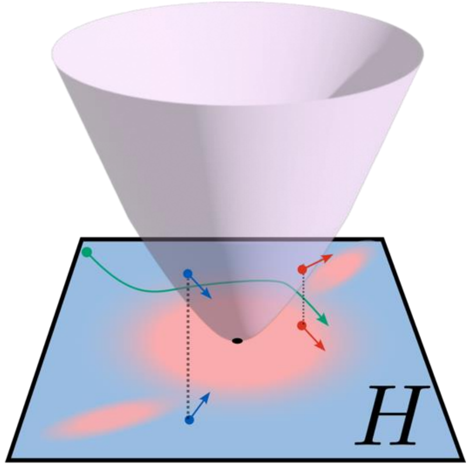
- Evaluating integral exactly is hard
- Approximate by sampling (simplest is Monte Carlo)
  - Sample h_0 uniformly at random
  - Backprop on point-wise Lyapunov Loss

$$L_V(h) \equiv \max\left\{0, \frac{\partial V^T}{\partial h}\pi(h) - \kappa V(h)\right\}$$

Point-wise Lyapunov Loss

https://arxiv.org/abs/2202.02526

Benefits of Sampling



- Avoids expensive ODE solve

- Goal is to minimize Lyapunov Loss everywhere

$$L_V(\theta) \equiv \boldsymbol{E}_{h_0}\left[\int L_V\big(x, y, h(t)\big)dt\right]$$

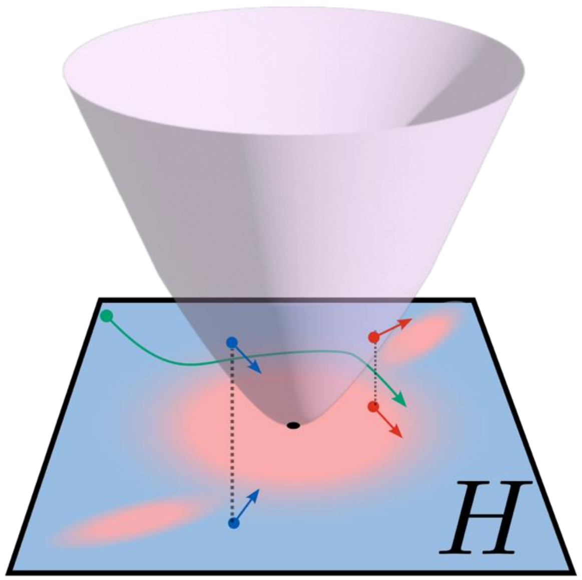
Achieving $L_V(\theta) = 0$ under uniform measure implies $L_V(\theta) = 0$ in original measure

- Similar idea used in Score-Based Generative Models & Moser Flows

https://arxiv.org/abs/2202.02526

Back to Application: Underactuated Control

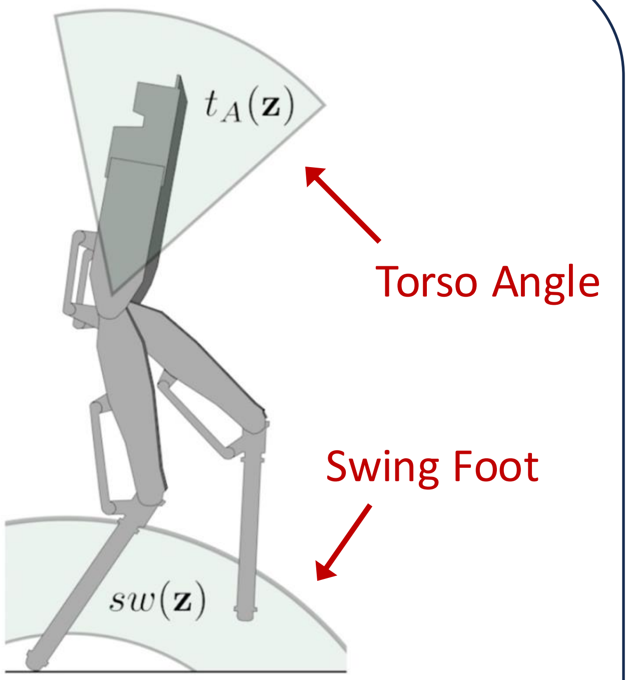
Note: Dynamics of Control System included in Neural ODE



$$\min_{\theta}\left[\frac{\partial V^T}{\partial h}\,\pi(h;\theta)+\kappa V(h)\right]\leq 0$$

Sample States
Compile Loss

Deploy Controller



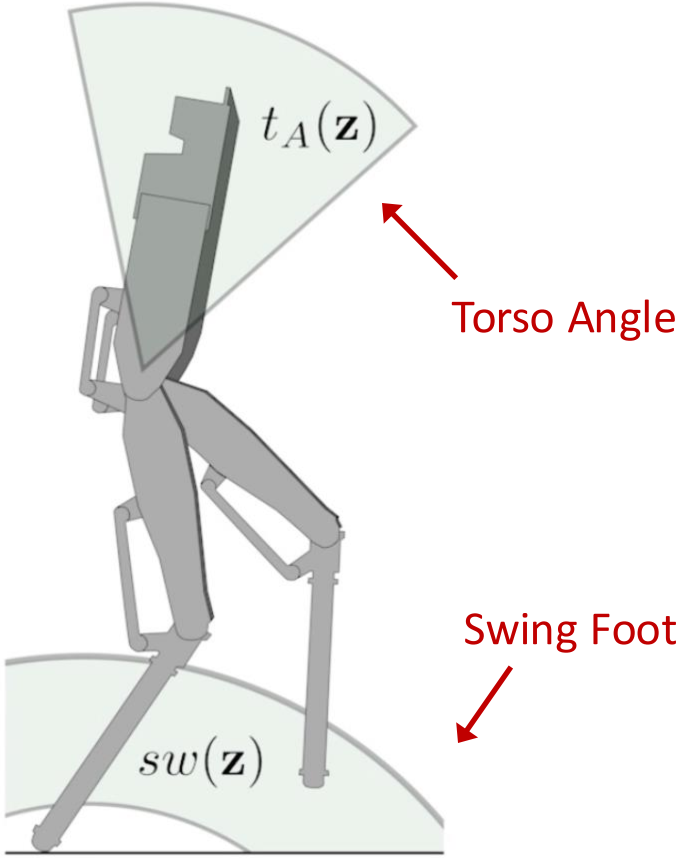
Stability/Safety Specification

Neural Gaits

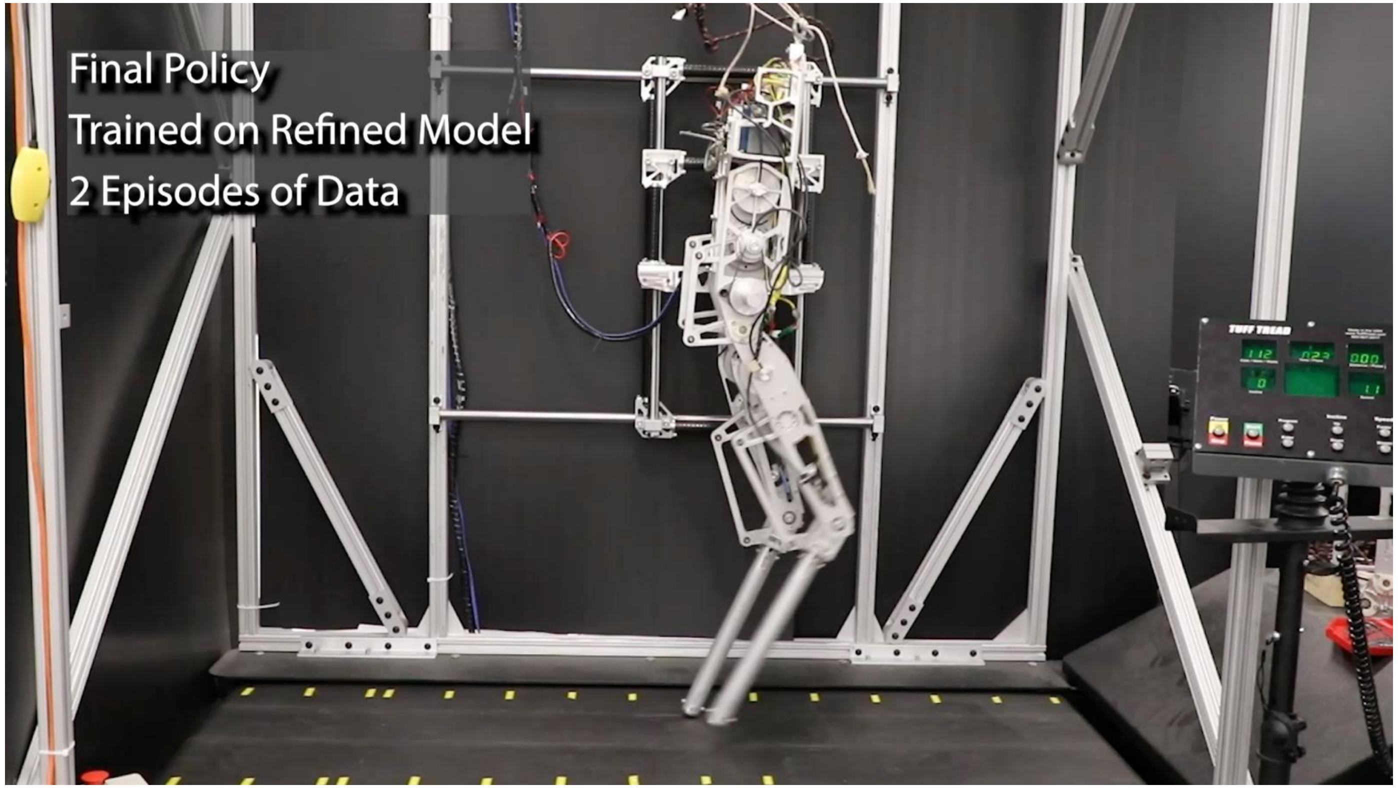
Learn policy to satisfy composition of continuous-time conditions
Implies indefinite walking (forward-invariance)

Ivan
Jimenez Rodriguez

Noel
Csomay-Shanklin

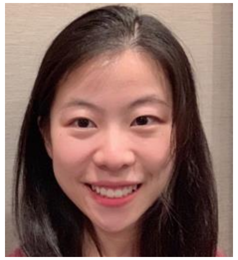


Example Barriers

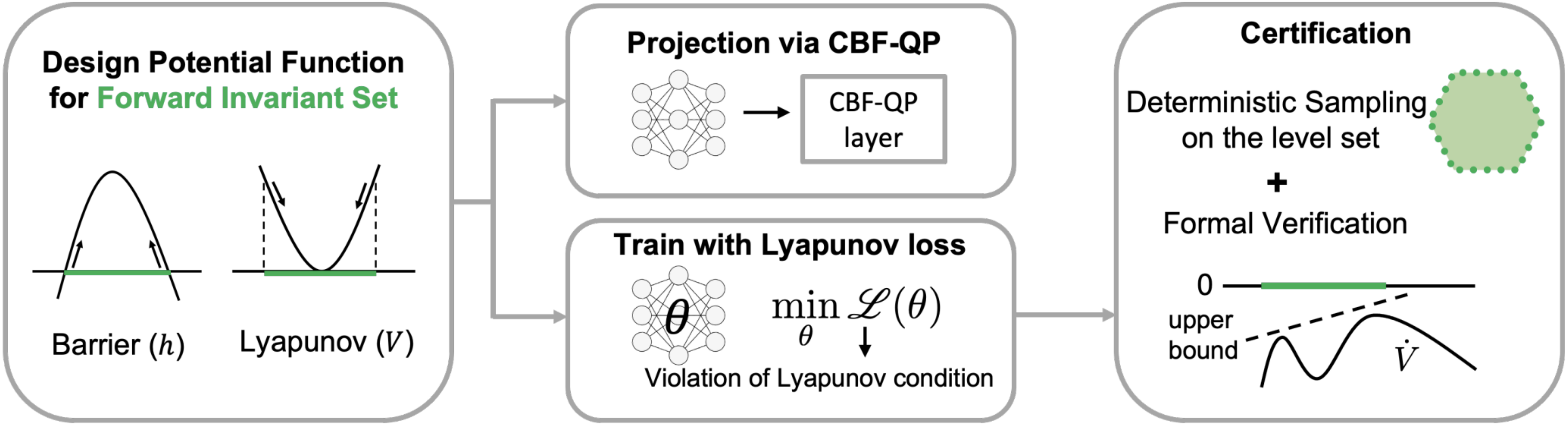


https://arxiv.org/abs/2204.08120

Certified Forward-Invariance in NODEs


Yujia
Huang

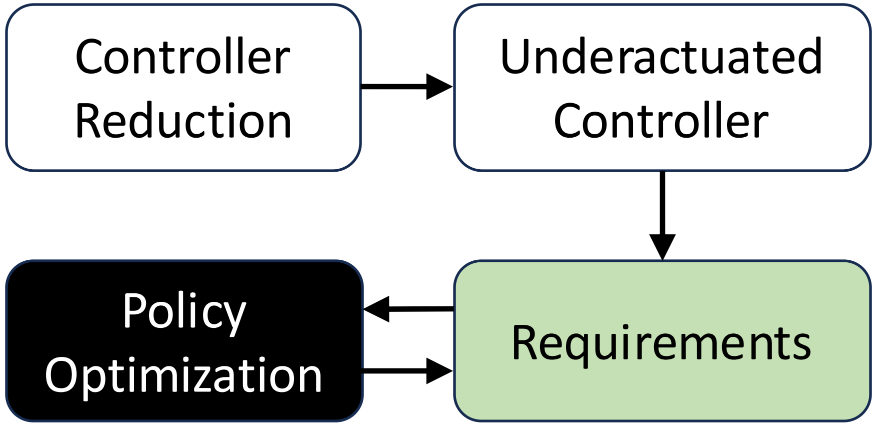

Ivan
Jimenez Rodriguez



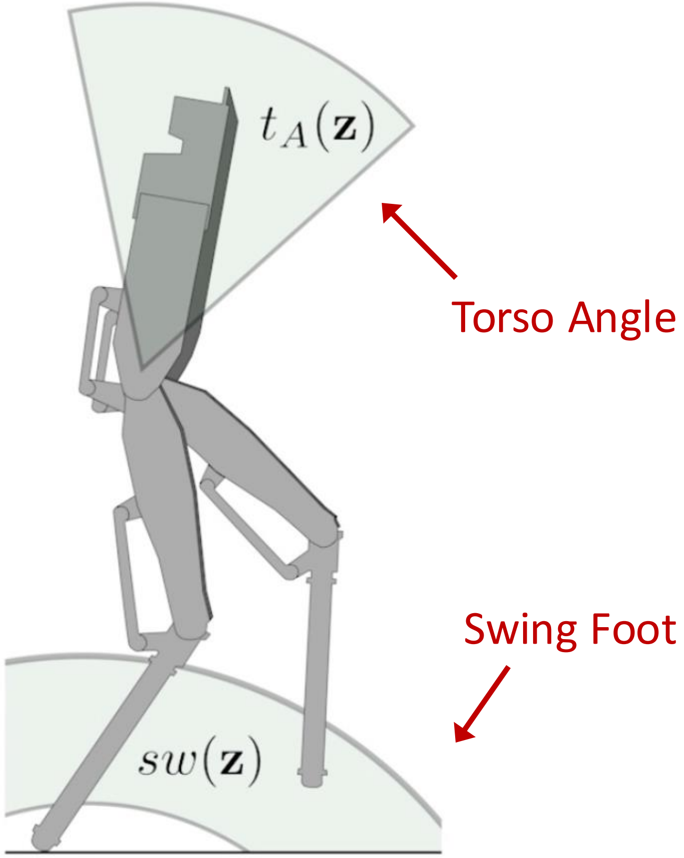
Certified Robust Forward Invariance
(First Ever Result)

https://arxiv.org/abs/2210.16940

Recall Requirement:
Cancel out effect of unactuated component

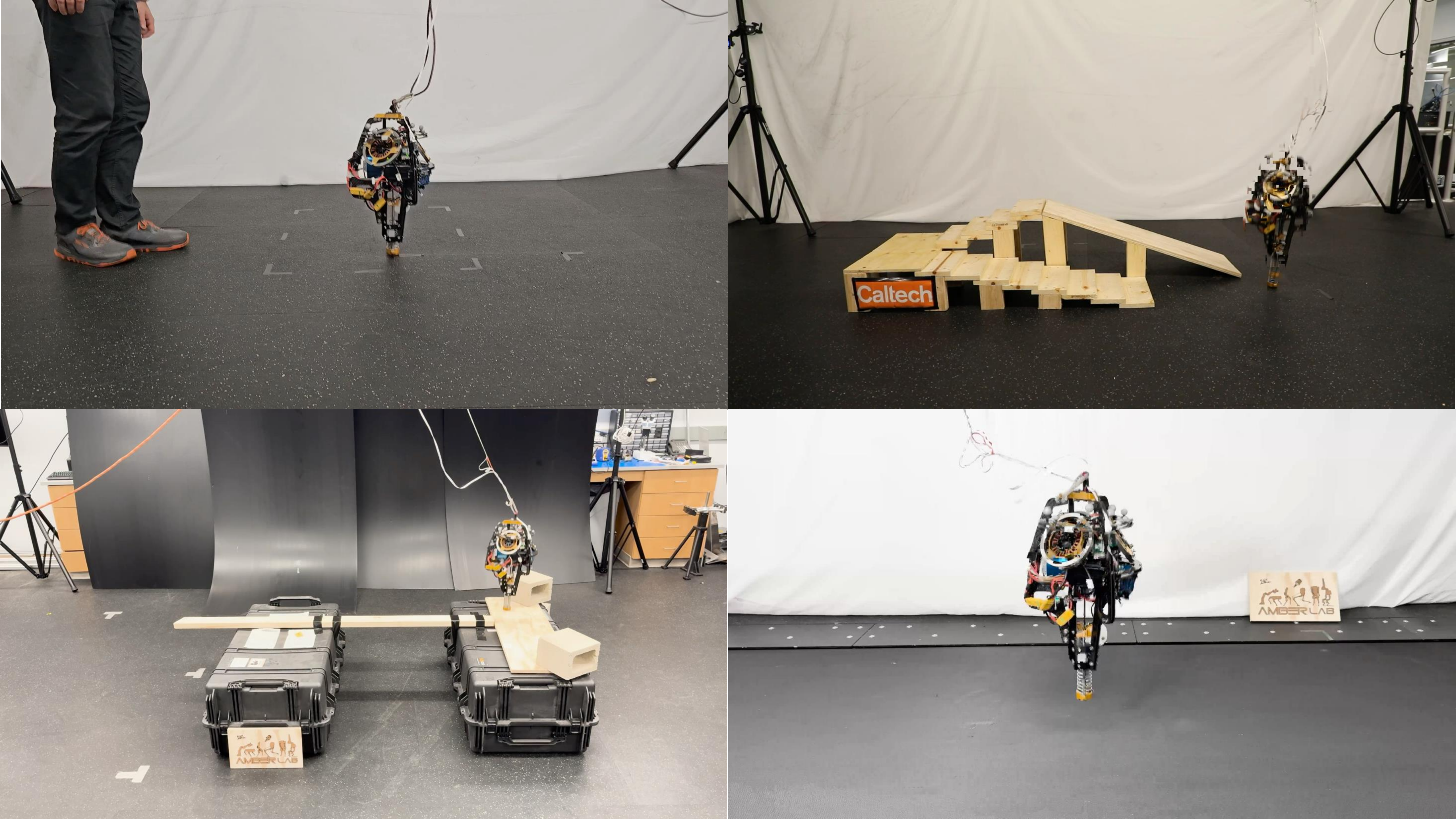


Example:

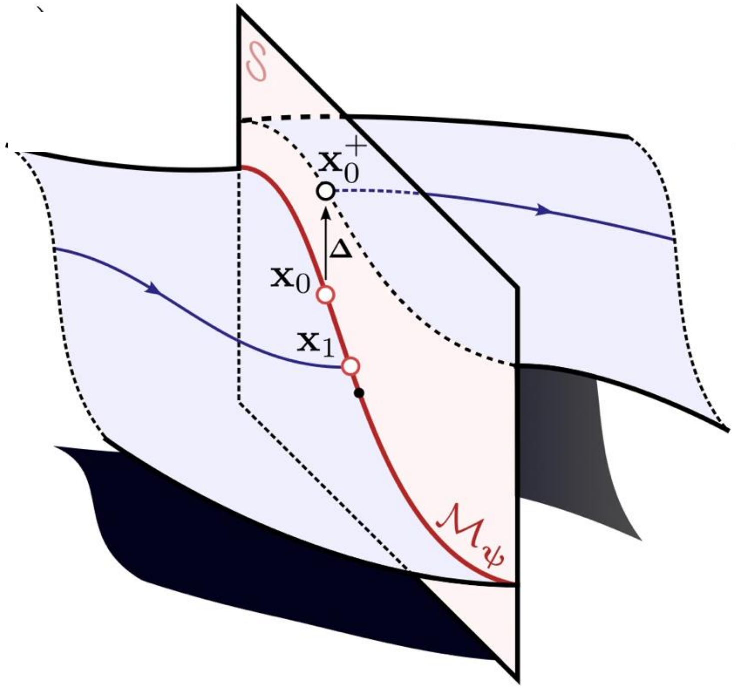


NOT OPTIMAL!

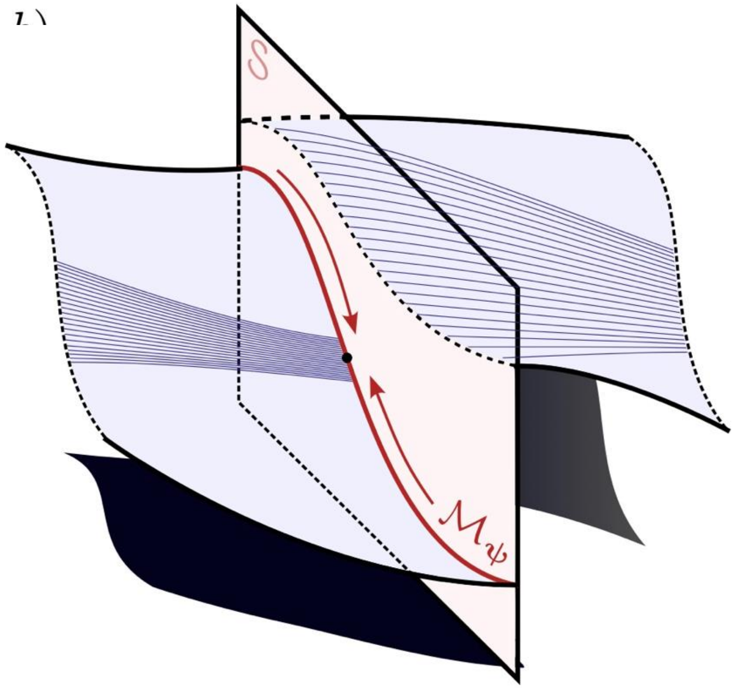
Conservative Specification (sufficient conditions)


Caltech
AMBER LAB
AMBER LAB

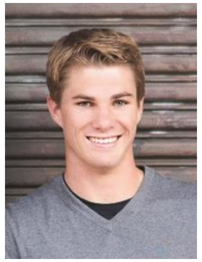
True Specification:
Cancel Out Effect on (Null) Unactuated Space



Invariance: Stays on Manifold



Stability: Converges to Optimality



Will
Compton

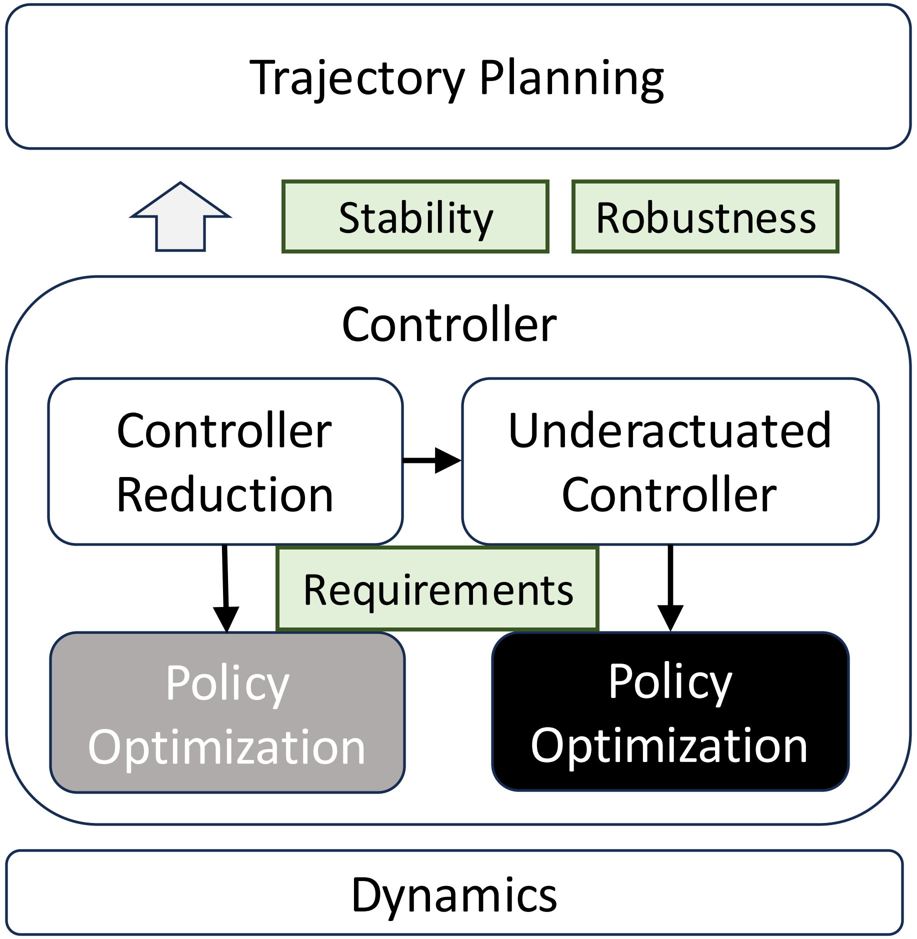
Noel
Csomay-Shanklin

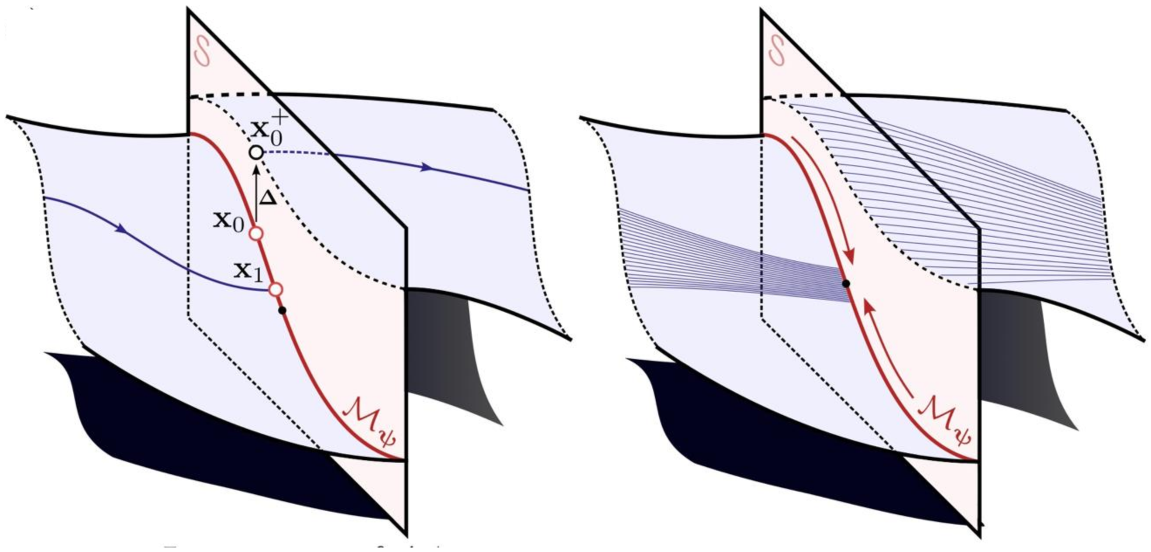
Ivan
Jimenez Rodriguez

Robust Agility via Learned Zero Dynamics Policies, IROS 2024 https://arxiv.org/abs/2409.06125

Constructive Nonlinear Control of Underactuated Systems via Zero Dynamics Policies, CDC 2024 https://arxiv.org/abs/2408.14749

Policy Learning for Specification Satisfaction



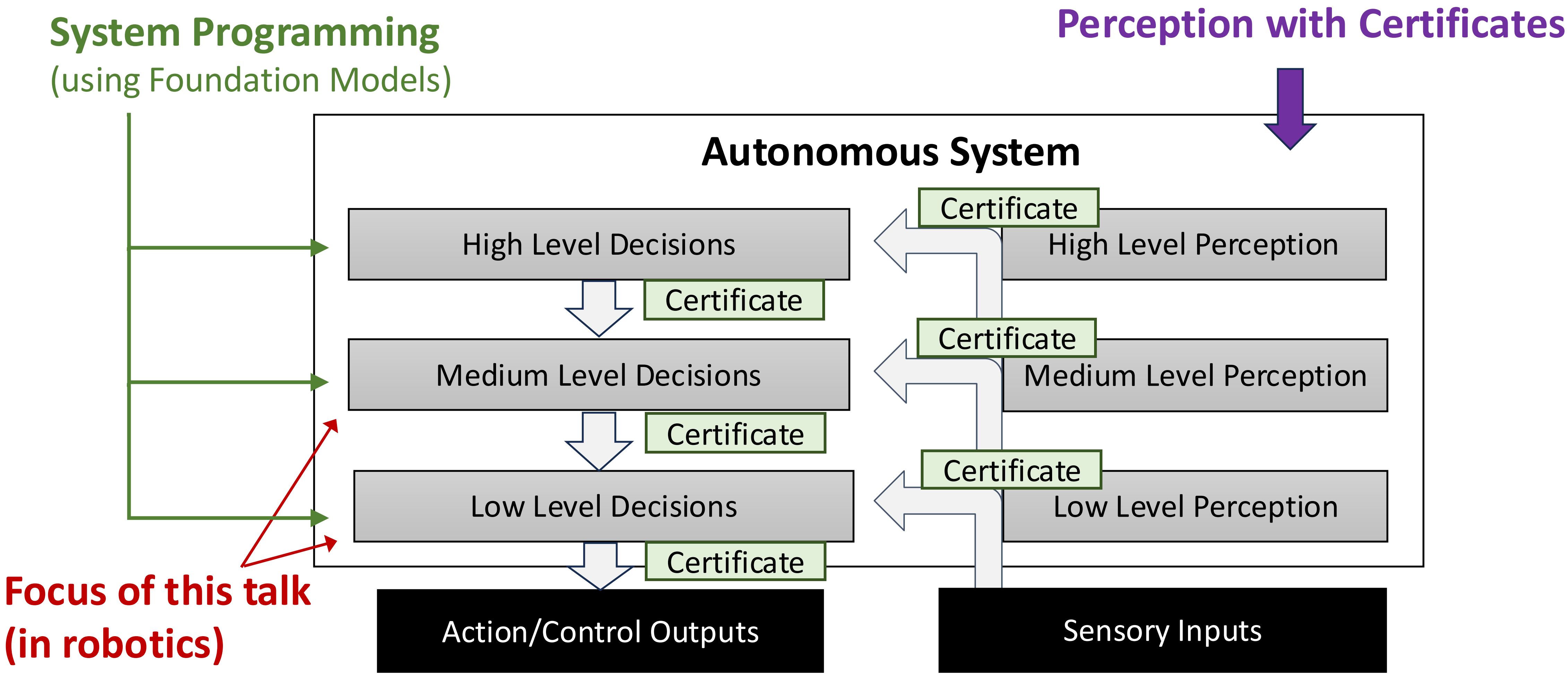


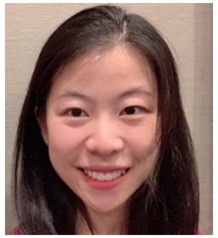
Many Specifications are Combinations of:

- Stability
- Invariance
- Optimality
- Robustness

Can Directly Learn to Satisfy!

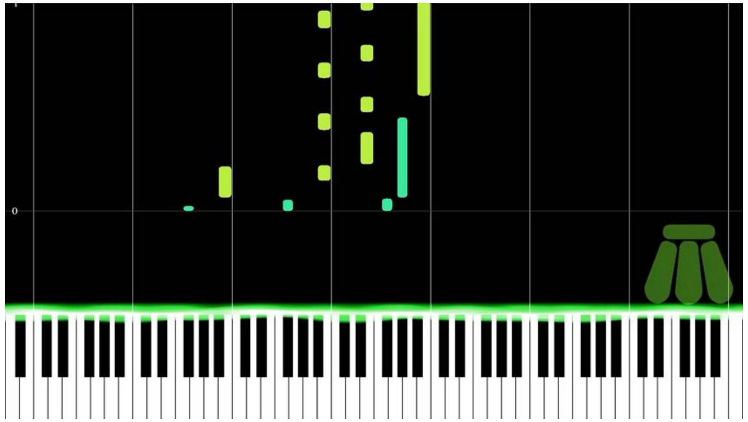
Moving Forward

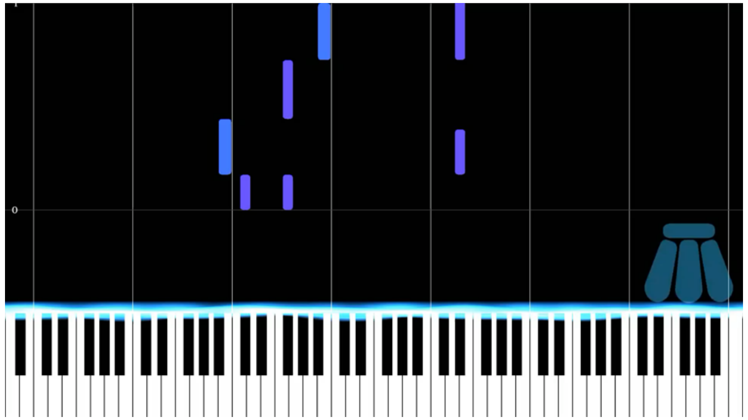


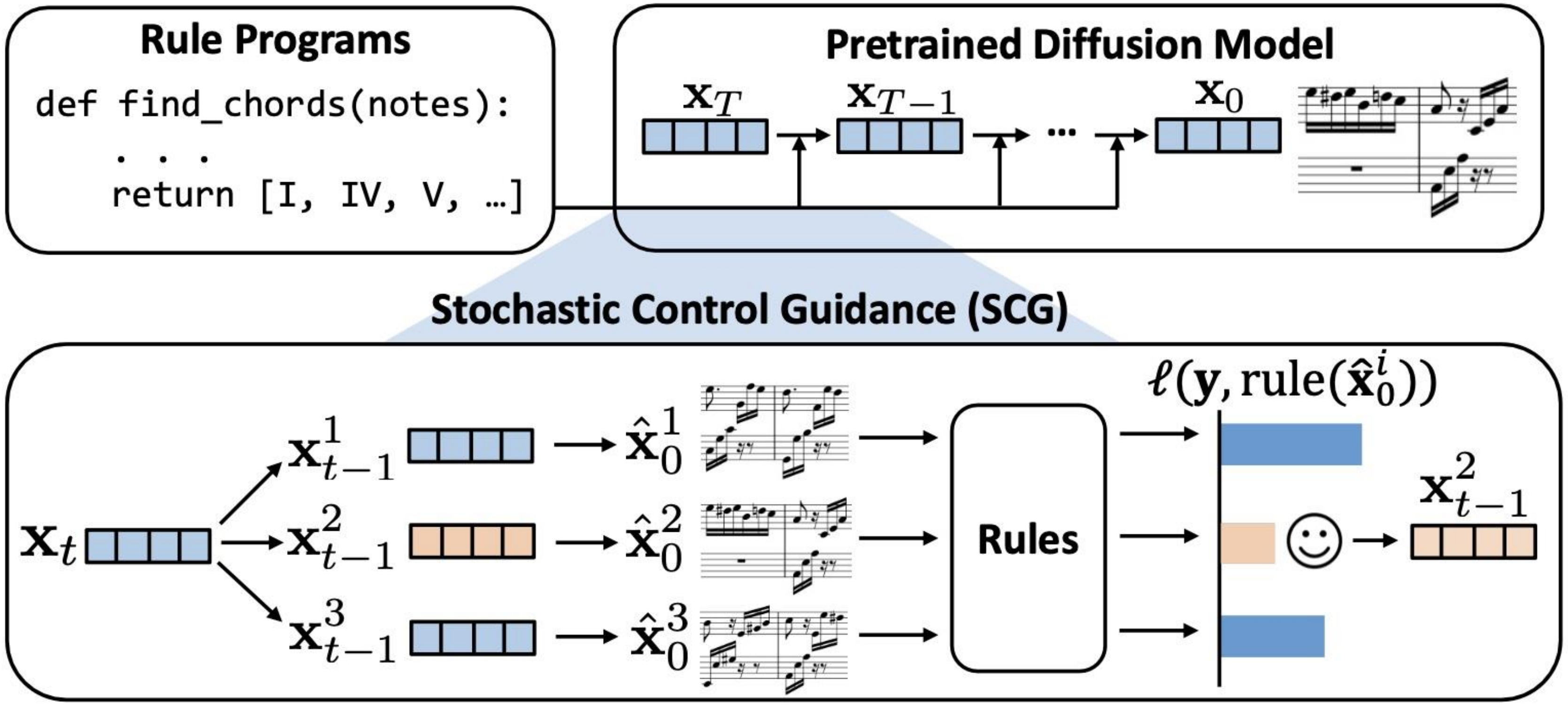


Yujia Huang

Aside: Symbolic Music Generation via Stochastic Control





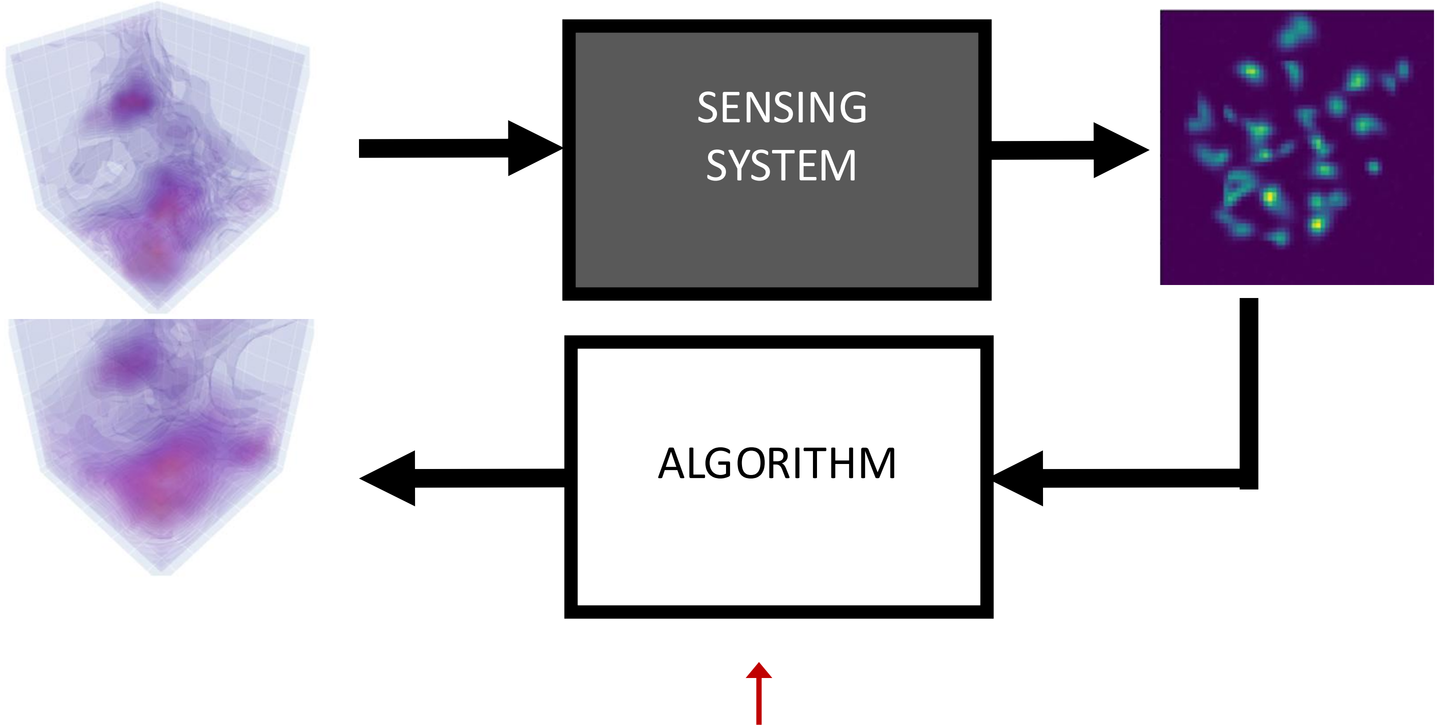


Symbolic Music Generation with Non-Differentiable Rule-Guided Diffusion

Yujia Huang, et al., ICML 2024 https://scg-rule-guided-music.github.io/

Perception: Scientific Imaging

Collaboration with
Katie Bouman's Group

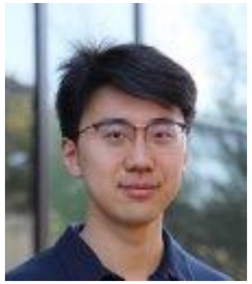


Imaging Algorithm is Neurosymbolic!

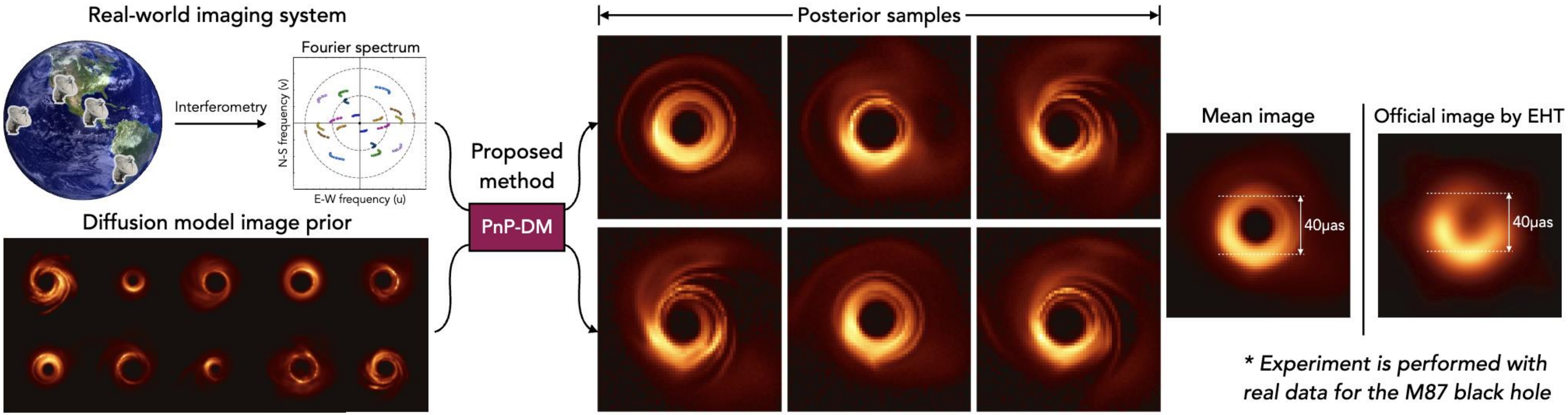
Requirements include:

- Consistent with known physics
- Proper posterior inference
  - (uncertainty calibration)

Plug-and-Play Bayesian Inversion
(Diffusion Model + Physics)



Ray Wu



Principled Probabilistic Imaging using Diffusion Models as Plug-and-Play Priors, NeurIPS 2024
https://arxiv.org/abs/2405.18782

Neural Control

Neural Lander: Stable Drone Landing Control using Learned Dynamics, Shi, et al., ICRA 2019

Neural-Swarm: Decentralized Close-Proximity Multirotor Control Using Learned Interactions, Shi et al., ICRA 2020

Neural-Swarm2: Planning and Control of Heterogeneous Multirotor Swarms using Learned Interactions, Shi et al., T-RO 2021.

Neural-Fly Enables Rapid Learning for Agile Flight in Strong Winds, O'Connell, Shi, et al., Science Robotics 2022

Meta-Adaptive Nonlinear Control: Theory and Algorithms, Shi et al., NeurIPS 2021

Hierarchical Meta-learning-based Adaptive Controller, Xie et al., ICRA 2024

Residual Policy Learning

Smooth Imitation Learning for Online Sequence Prediction, Hoang Le, et al., ICML 2016

Control Regularization for Reduced Variance Reinforcement Learning, Richard Cheng et al. ICML 2019

Batch Policy Learning under Constraints, Hoang Le, et al. ICML 2019

Imitation-Projected Programmatic Reinforcement Learning, Abhinav Verma, Hoang Le, et al., NeurIPS 2019

Policy Learning for Specification Satisfaction

Neural Gaits: Learning Bipedal Locomotion via Control Barrier Functions and Zero Dynamics Policies, L4DC 2022

Robust Agility via Learned Zero Dynamics Policies, Csomay-Shanklin*, Compton*, Jimenez Rodriguez*, et al., IROS 2024

Constructive Nonlinear Control of Underactuated Systems via Zero Dynamics Policies, CDC 2024

LyaNet: A Lyapunov Framework for Training Neural ODE, , Jimenez Rodriguez, et al., ICML 2022

FI-ODE: Certifiably Robust Forward Invariance in Neural ODEs, Huang, et al., *arxiv*

Misc

Symbolic Music Generation with Non-Differentiable Rule-Guided Diffusion, Huang et al., ICML 2024

Principled Probabilistic Imaging using Diffusion Models as Plug-and-Play Priors, Wu et al., NeurIPS 2024

Thanks!

@yisongyue

http://www.yisongyue.com