Neural Networks with Physics-Informed Architectures and Constraints for Dynamical Systems Modeling

Franck Djeumou[1], Cyrus Neary [1], Eric Goubault[2], Sylvie Putot[2], Ufuk Topcu [1]
[1] The University of Texas at Austin, United States
[2] LIX, CNRS, École Polytechnique, Institut Polytechnique de Paris, France

autonomous SYSTEMS GROUP

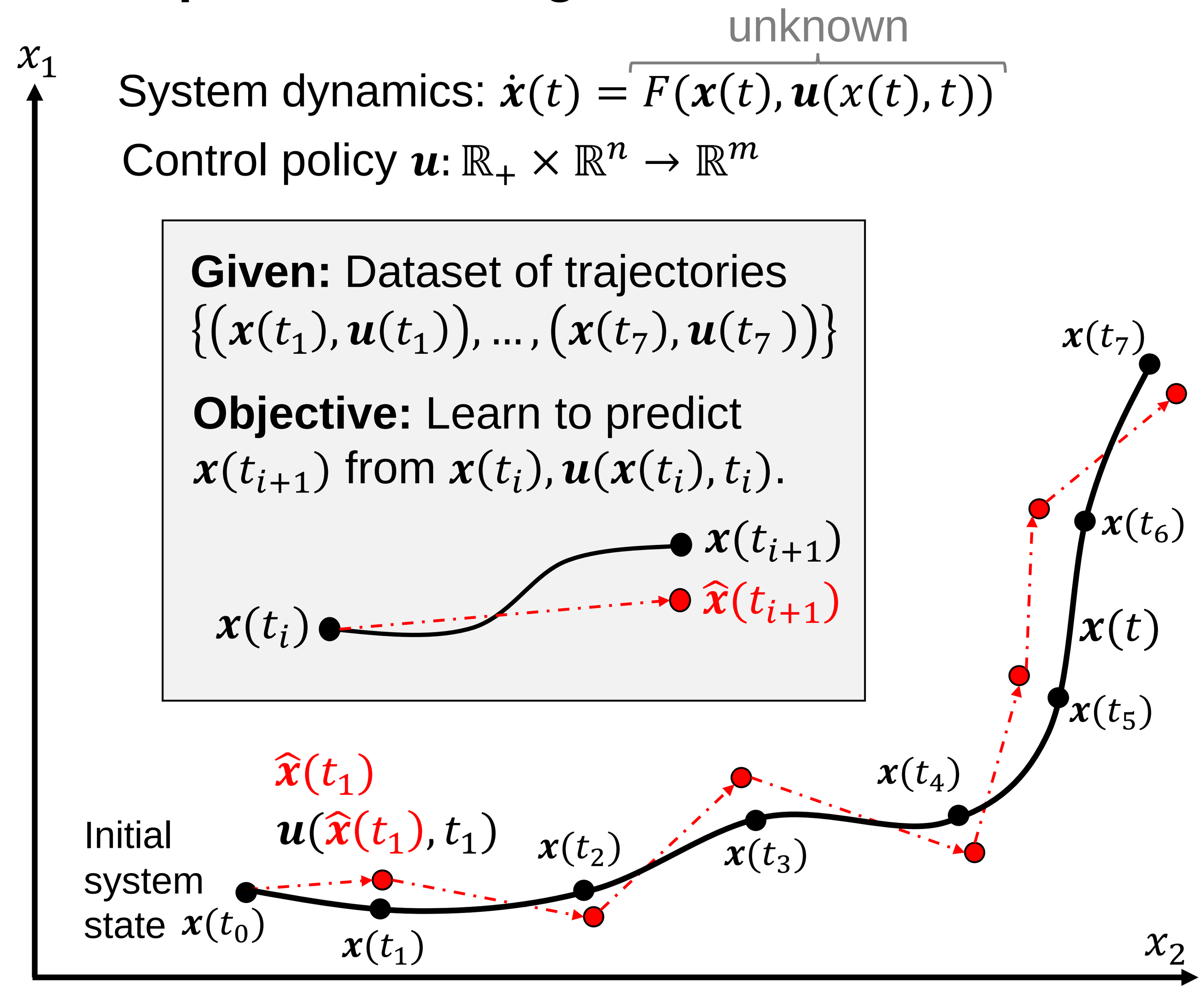
The central question.
How can we incorporate **physics-based knowledge** into neural network models of dynamical systems?

Why bring physics knowledge into deep learning algorithms?
To improve **data efficiency** and **model generalization to previously unseen regions of the state space**.
Such a priori knowledge might arise from physical principles (e.g., conservation laws) or from the system's design (e.g., the Jacobian matrix of a robot), even if large portions of the system dynamics remain unknown.

A summary of the approach.
Use a neural ODE to capture the system dynamics.

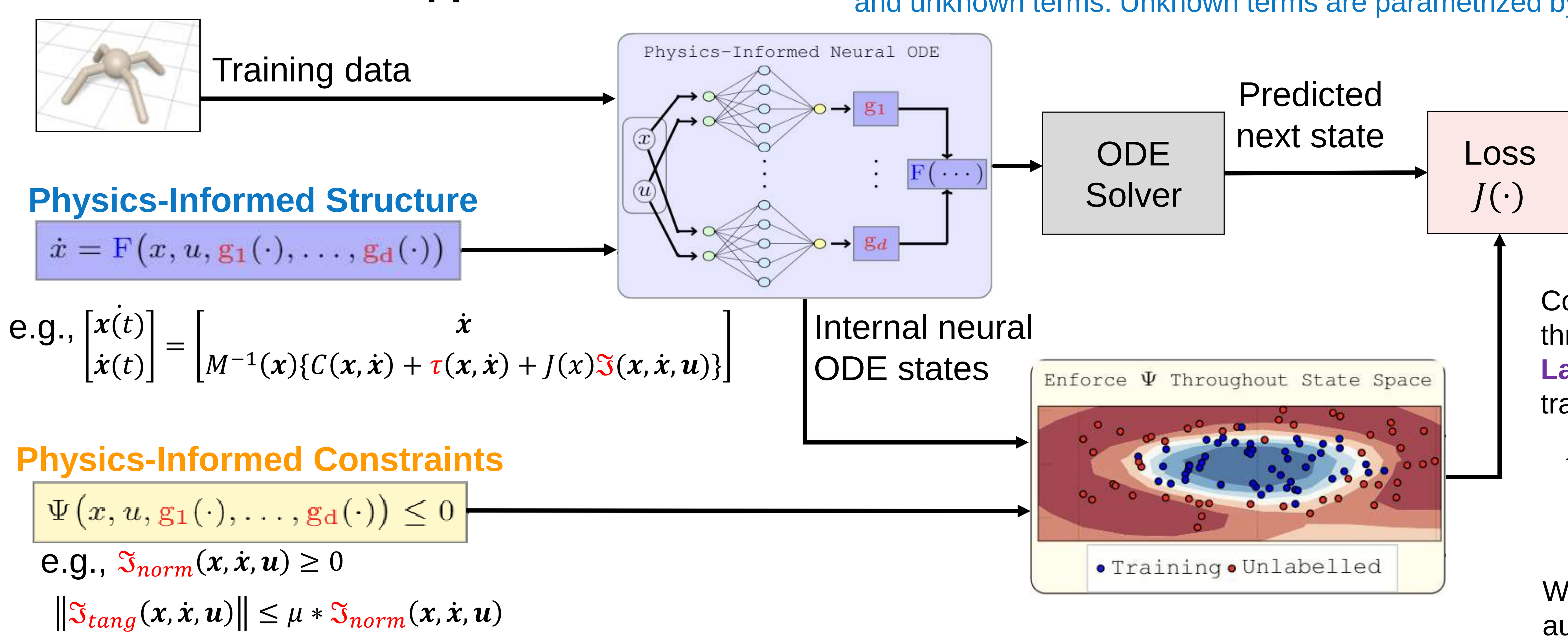
1. Develop a general framework to use physics knowledge to **inform the structure of the network**.
2. Develop an algorithm to train the model to **respect general physics-based constraints**.

The problem setting



An Illustration of the Approach

Use physics knowledge to represent vector field as a composition of known and unknown terms. Unknown terms are parametrized by neural networks.



Loss value

Constraints are enforced through an **augmented Lagrangian** inspired training algorithm:

$$Loss = loss_{data} + \lambda * loss_{constraint} + \mu * loss^2_{constraint}$$

With λ and μ updated automatically throughout training.

Use physics knowledge to impose arbitrary constraints on the outputs and internal states of the model.

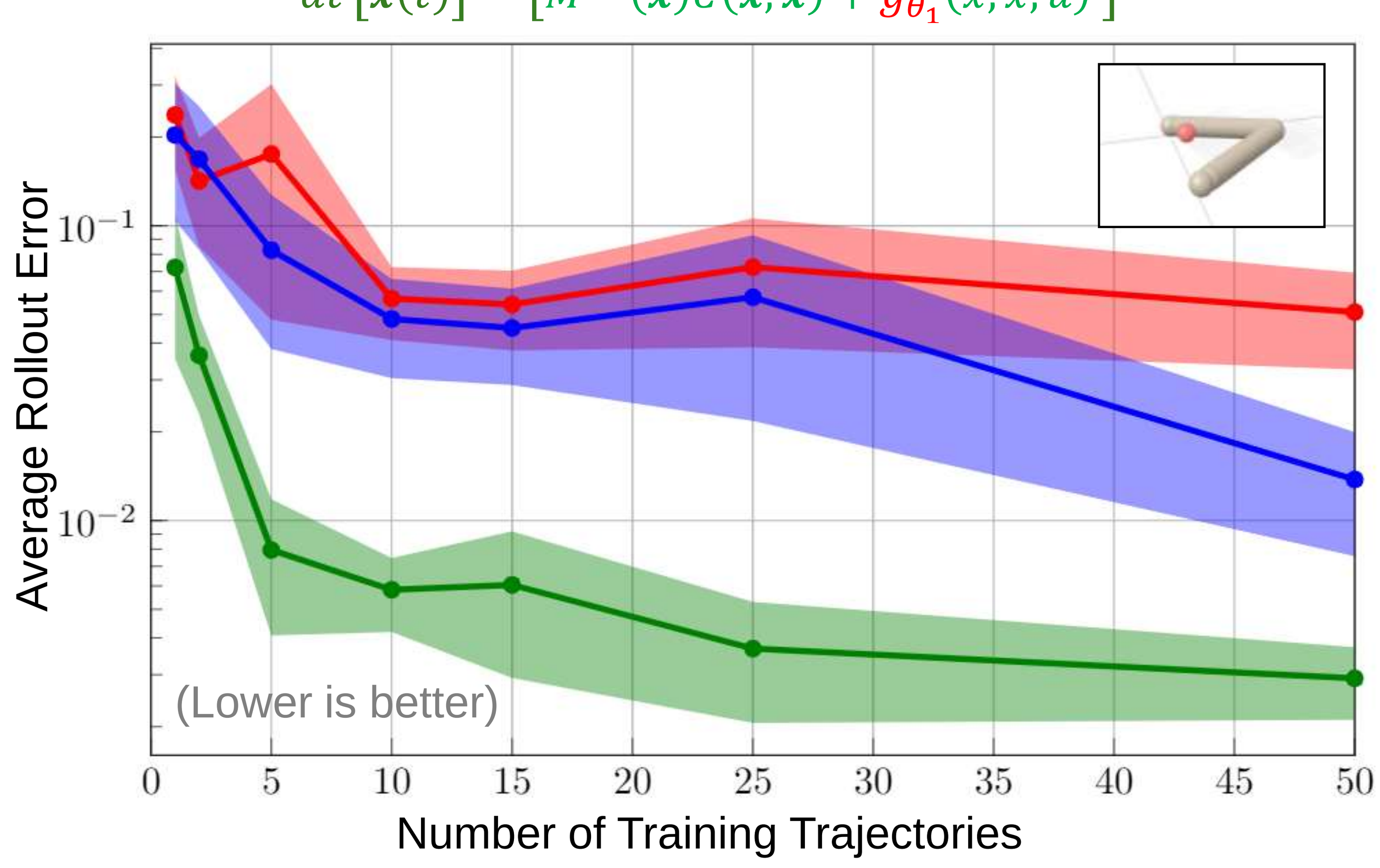
Enforce the constraints at both (training + unlabeled) points.

Physics knowledge improves data efficiency

Baseline, no side information
$$\frac{d}{dt}\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} = F_\theta(x(t), u(x(t), t))$$

Basic vector field structure
$$\frac{d}{dt}\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} \dot{x}(t) \\ F_\theta(x(t), u(x(t), t)) \end{bmatrix}$$

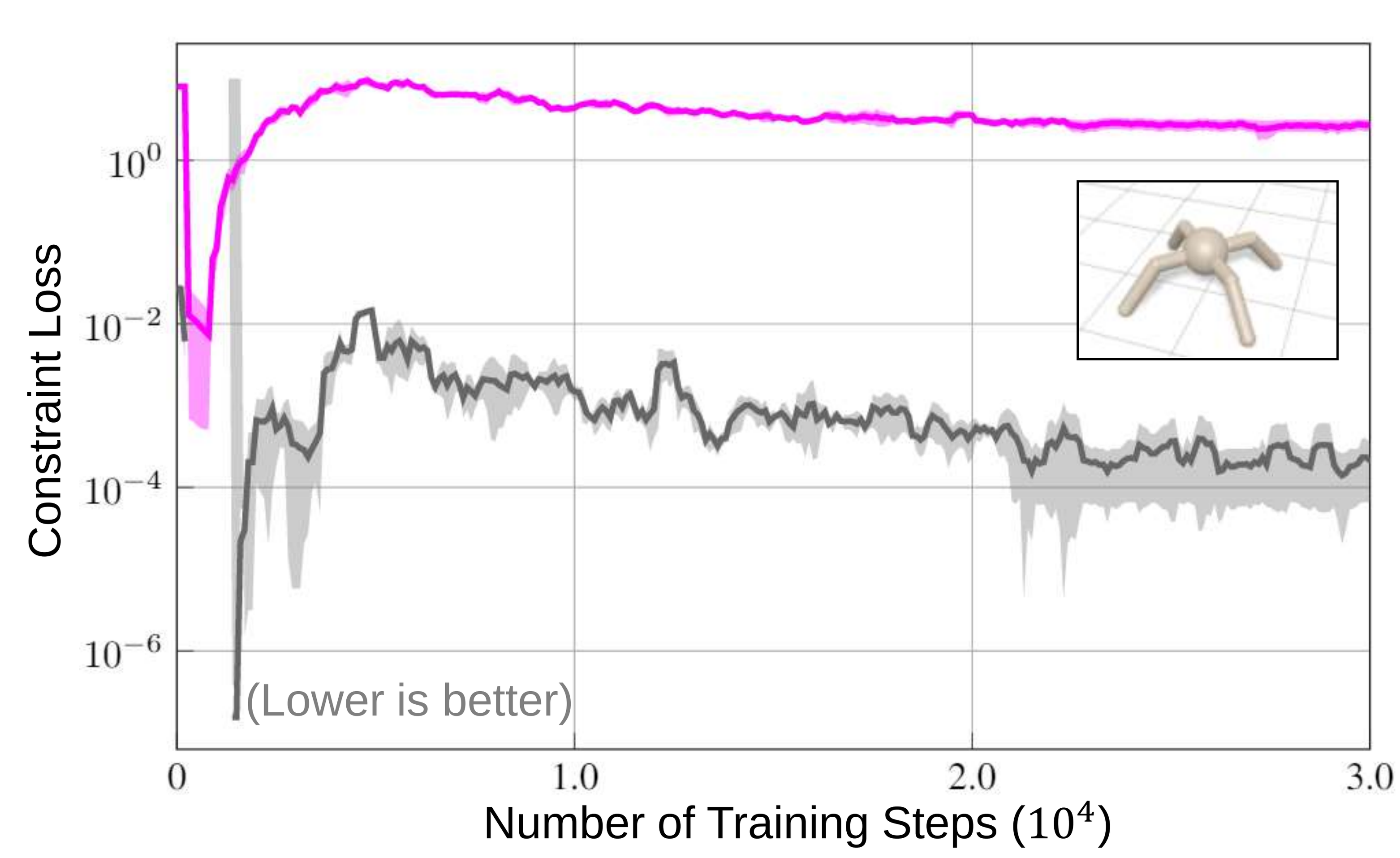
Knowledge of mass matrix
$$\frac{d}{dt}\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} \dot{x}(t) \\ M^{-1}(x)C(x,\dot{x}) + g_{\theta_1}(x,\dot{x},u) \end{bmatrix}$$



Constraints hold outside the training dataset

Knowledge of Jacobian matrix
$$\frac{d}{dt}\begin{bmatrix} x(t) \\ \dot{x}(t) \end{bmatrix} = \begin{bmatrix} \dot{x}(t) \\ M^{-1}(x)C(x,\dot{x}) + g_{\theta_1}(x,\dot{x},u) + J(x)g_{\theta_2}(x,\dot{x},u) \end{bmatrix}$$

Knowledge of Jacobian matrix + enforcing contact constraints
$$g^{norm}_{\theta_2}(x,\dot{x},u) \geq 0, \left\| g^{tang}_{\theta_2}(x,\dot{x},u) \right\| \leq \mu * g^{norm}_{\theta_2}(x,\dot{x},u)$$



Contact: fdjeumou@utexas.edu, cneary@utexas.edu