Explainability, interpretability, & safety in ML for electric power systems

Priya L. Donti
Assistant Professor
Massachusetts Institute of Technology

Power systems are safety-critical systems

Am I confident that this model can be deployed?

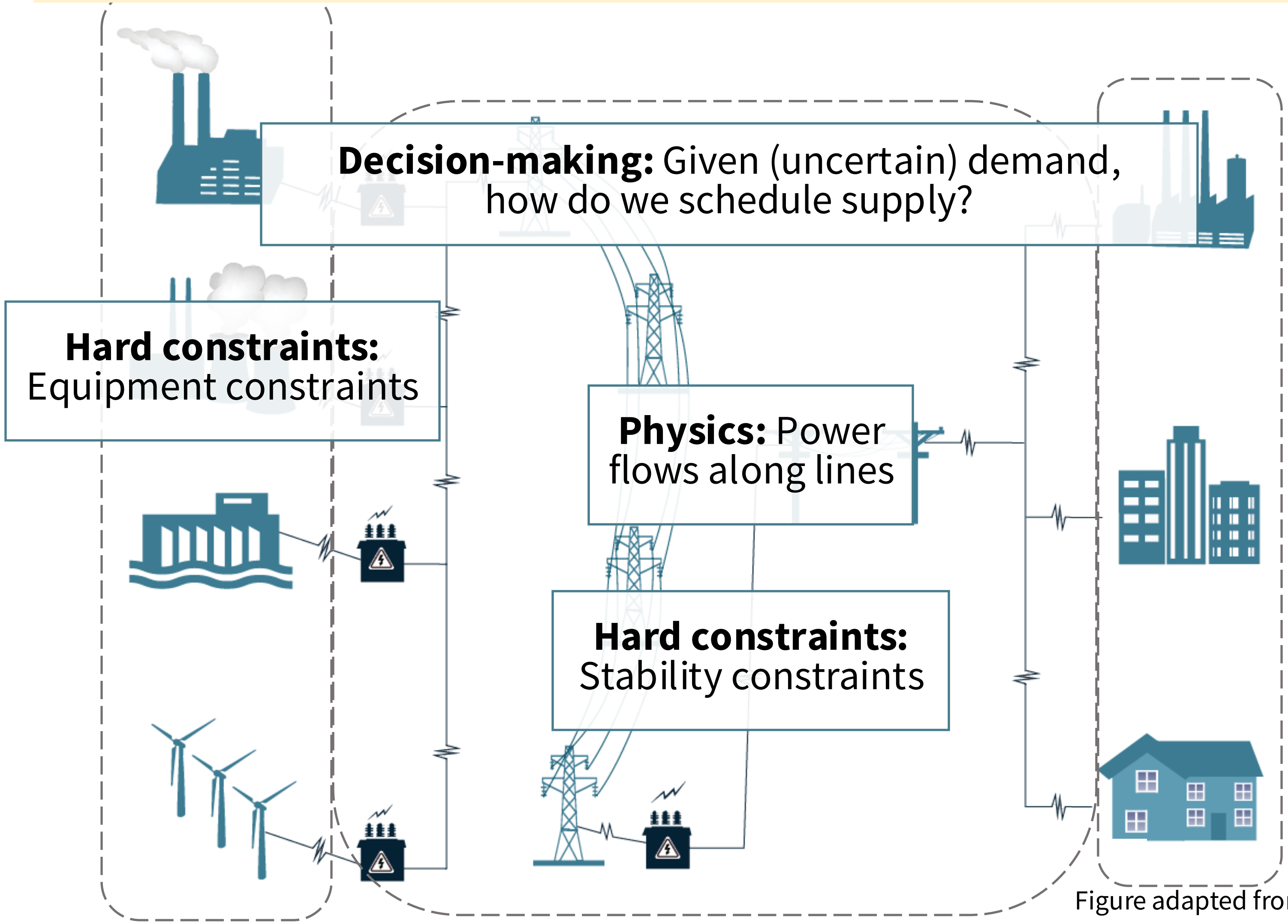


Figure adapted from: US Congressional Budget Office

Power systems are human systems

What is being prioritized and/or missed?

Image source: CNET

Image source: Utility Dive

Power systems are heavily regulated systems

How can we diagnose what went wrong?

MIKE ELIASON/SANTA BARBARA COUNTY FIRE DEPARTMENT/AP

Mechanisms for explainability, interpretability, & safety

- **Gray-box models:** Creating model components we can recognize and understand
- **Verifying model behaviors:** Testing that a model will behave as intended
- **Enforcing model behaviors:** Engineering desired behaviors into models
- **Understanding and improving data:** Ensuring data representativeness & reliability

Mechanisms for explainability, interpretability, & safety

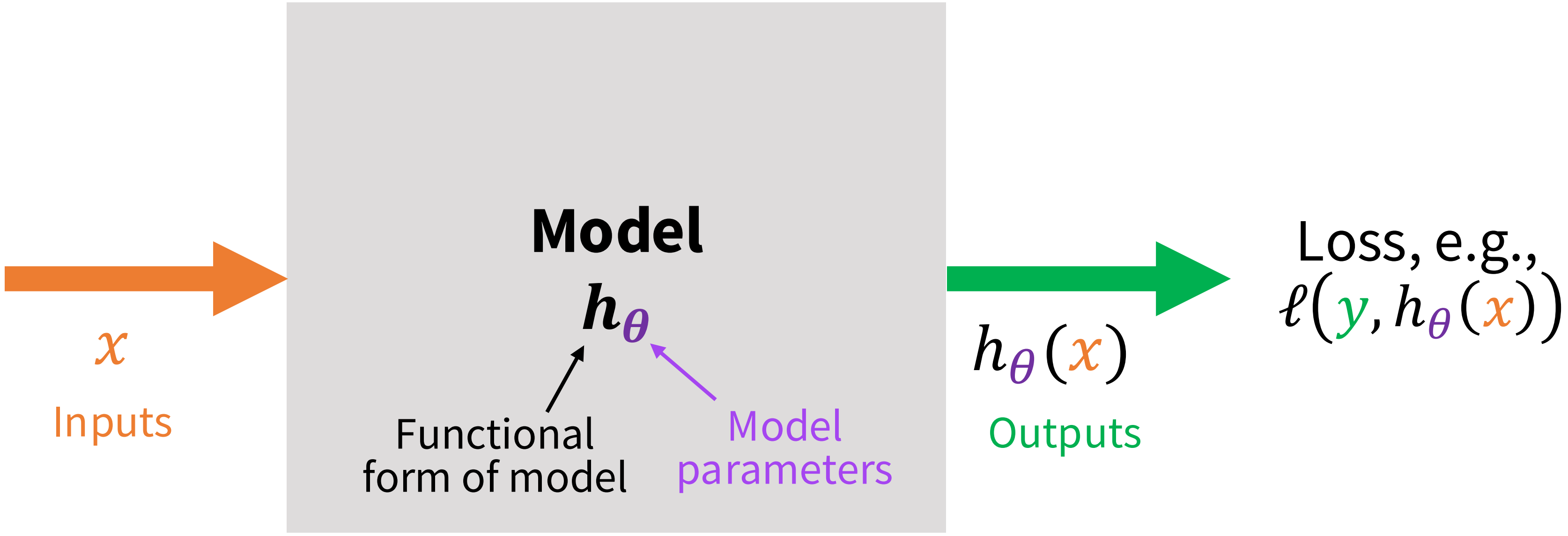
Gray-box models: Creating model components we can recognize and understand

Verifying model behaviors: Testing that a model will behave as intended

Enforcing model behaviors: Engineering desired behaviors into models

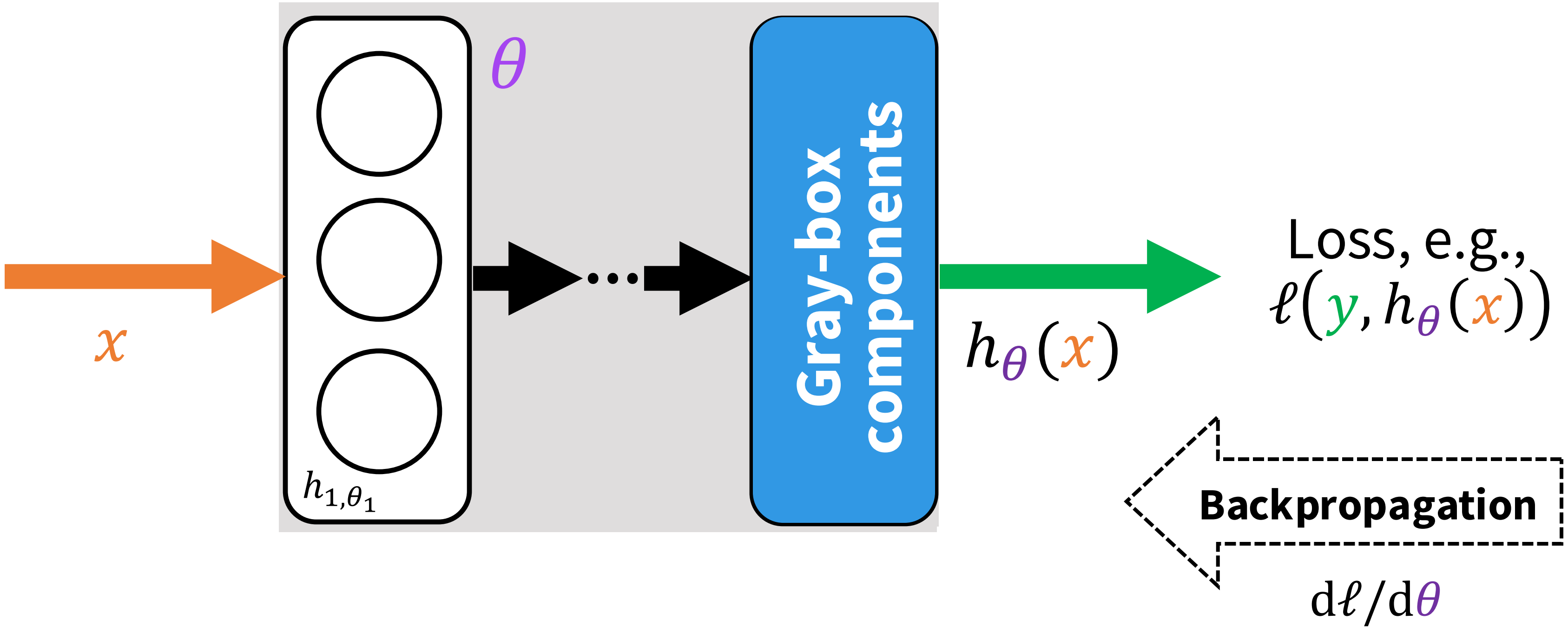
Understanding and improving data: Ensuring data representativeness & reliability

Deep learning is differentiable function composition

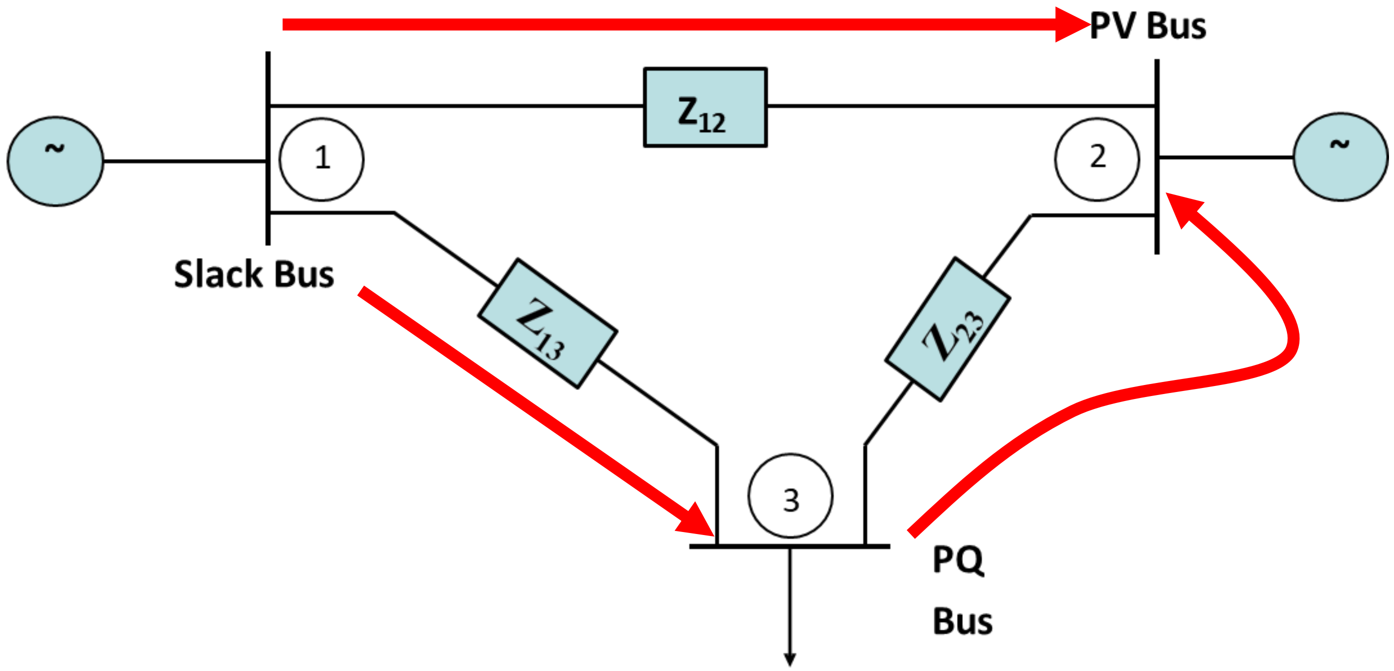


Deep learning is differentiable function composition

- Neural network h_θ = composition of nonlinear, parameterized functions (*layers*)
- Update parameters θ to minimize loss ℓ using gradients from *backpropagation*
- All components (layers and loss) must be differentiable



Many power systems processes are implicit functions

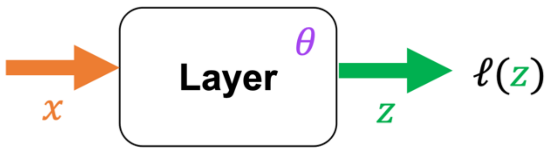


Solve for v such that:

$$\mathrm{diag}(v)\,\bar{Y}\bar{v} = \left(p_g - p_d\right) + \left(q_g - q_d\right)j$$

Implicit function solved using iterative processes such as Newton-Raphson

Toolkit: Implicit layers

	Explicit layer	Implicit layer
Forward pass: 	$z = f(x, \theta)$ [e.g., $z = \sigma(\theta^T x + \theta_0)$]	Find z such that $g(z, x, \theta) = 0$ [e.g., power flow]
Backward pass: $\frac{\mathrm{d}\ell}{\mathrm{d}\theta} = \frac{\mathrm{d}\ell}{\mathrm{d}z^\star}\frac{\mathrm{d}z^\star}{\mathrm{d}\theta}$ $\quad \frac{\mathrm{d}\ell}{\mathrm{d}x} = \frac{\mathrm{d}\ell}{\mathrm{d}z^\star}\frac{\mathrm{d}z^\star}{\mathrm{d}x}$	$\frac{\mathrm{d}z^\star(x)}{\mathrm{d}x} = \frac{\mathrm{d}f(x, \theta)}{\mathrm{d}x}$	Find $\mathrm{d}z^\star(x)/\mathrm{d}x$ such that $\frac{\mathrm{d}g(z^\star(x), x, \theta)}{\mathrm{d}x} = 0$ by using implicit function theorem at a solution point

See also: Zico Kolter, David Duvenaud, and Matt Johnson. “Deep Implicit Layers - Neural ODEs, Deep Equilibirum Models, and Beyond.” Tutorial at *NeurIPS 2020*. https://implicit-layers-tutorial.org/

Differentiating through optimization problems

Insight: Apply the implicit function theorem to the KKT optimality conditions

Example optimization problem

$$\begin{aligned} \underset{z}{\text{minimize}} \quad & ½\, z^T Q z + q^T z \\ \text{subject to} \quad & Az = b \\ & Gz \le h \end{aligned}$$

Selected KKT optimality conditions

$$\begin{gathered} Qz^\star + q + A^T \nu^\star + G^T \lambda^\star = 0 \\ Az^\star - b = 0 \\ \operatorname{diag}(\lambda^\star)(Gz^\star - h) = 0 \end{gathered}$$

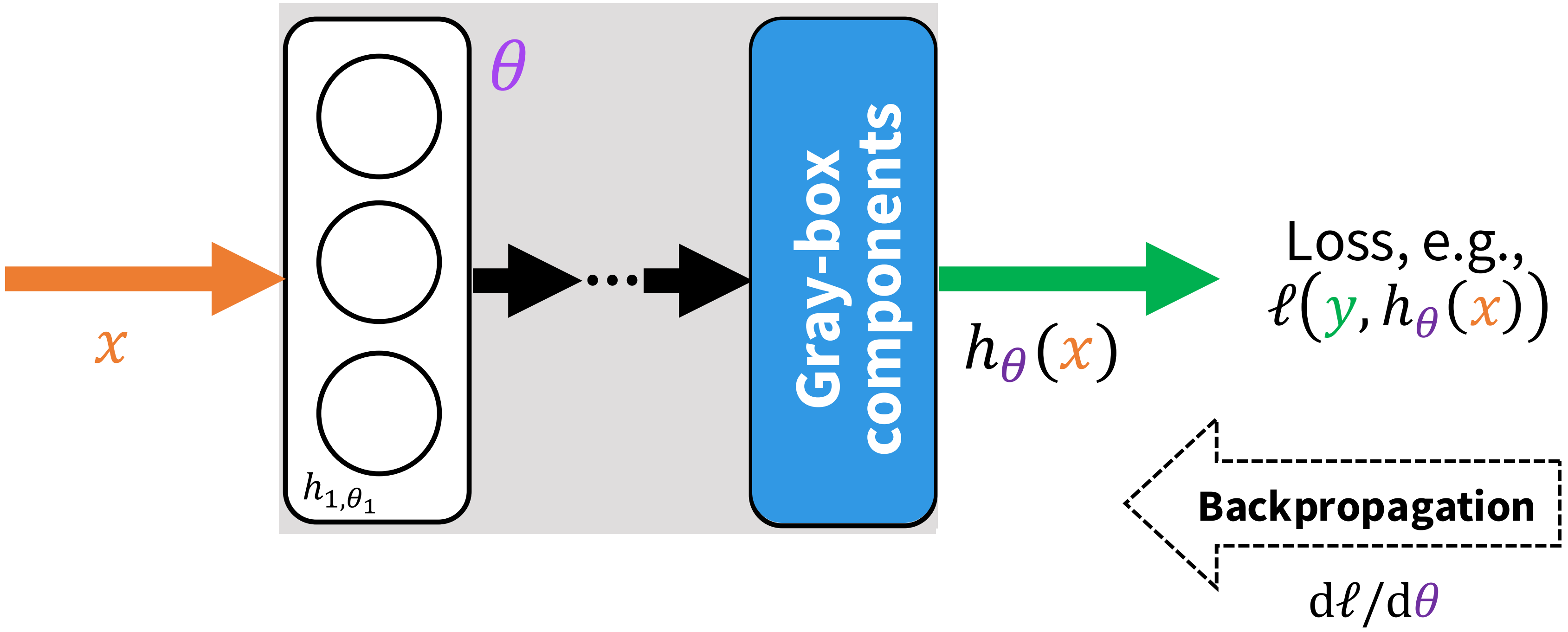
Step 1: Apply implicit function theorem to the KKT conditions

$$\underbrace{\begin{bmatrix} Q & G^T & A^T \\ \operatorname{diag}(\lambda^\star) G & \operatorname{diag}(Gz^\star - h) & 0 \\ A & 0 & 0 \end{bmatrix}}_{\text{Generalized Jacobian of KKT conditions}} \underbrace{\begin{bmatrix} \mathrm{d}z \\ \mathrm{d}\lambda \\ \mathrm{d}\nu \end{bmatrix}}_{\text{Desired gradients}} = - \underbrace{\begin{bmatrix} \mathrm{d}Qz^\star + \mathrm{d}q + \mathrm{d}G^T \lambda^\star + \mathrm{d}A^T \nu^\star \\ \operatorname{diag}(\lambda^\star)\mathrm{d}Gz^\star - \operatorname{diag}(\lambda^\star)\mathrm{d}h \\ \mathrm{d}Az^\star - \mathrm{d}b \end{bmatrix}}_{\text{Gradients of problem parameters}}$$

Step 2: Use "Jacobian-vector trick" for efficient backpropagation

Brandon Amos and J. Zico Kolter. "OptNet: Differentiable optimization as a layer in neural networks." *ICML 2017.*
Priya L. Donti, Brandon Amos, and J. Zico Kolter. "Task-based end-to-end model learning in stochastic optimization." *NeurIPS 2017.*

Deep learning is differentiable function composition



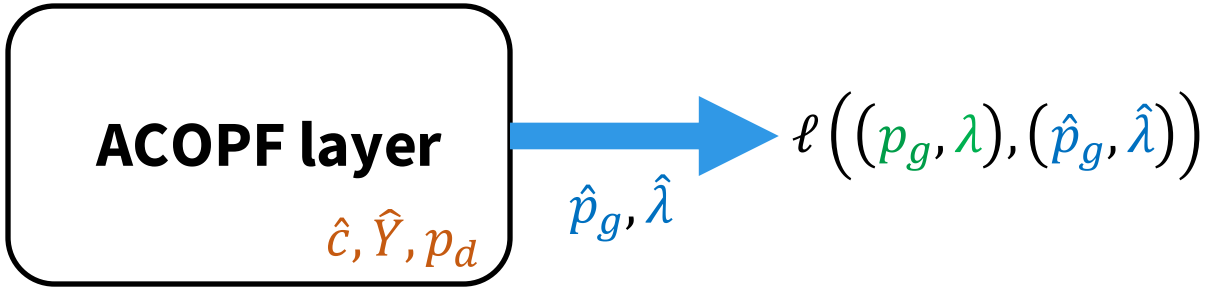
Using differentiable ACOPF for system identification

Problem (Inverse-OPF): Given grid data, identify generator costs and grid parameters

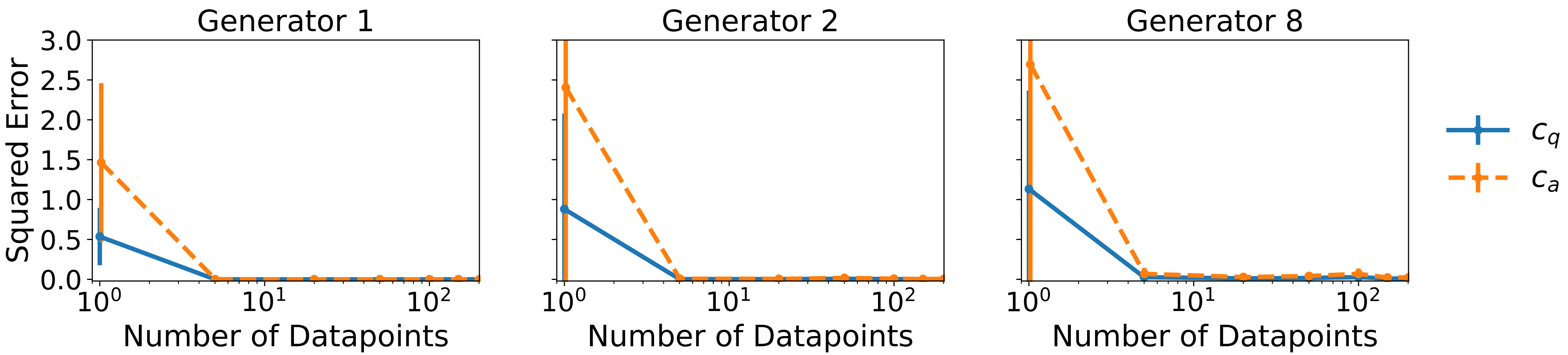
$$\underset{\hat{c},\hat{Y}}{\text{minimize}} \;\; \ell\Big((p_g,\lambda),(\hat{p}_g,\hat{\lambda})\Big)$$

$$\text{subject to} \;\; \hat{p}_g,\hat{\lambda} = \text{ACOPF}(\hat{c},\hat{Y},p_d)$$

Approach: "One-layer neural network" with implicit ACOPF layer

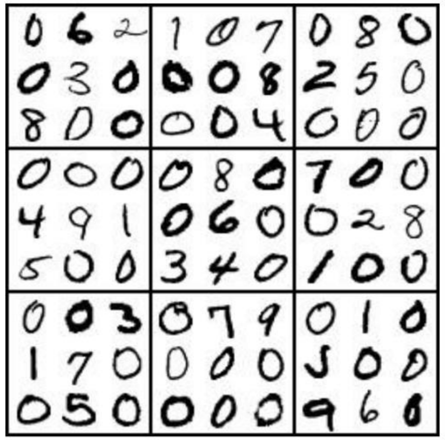


Finding: Some system parameters are recoverable, in an interpretable way



Priya L. Donti, Inês Lima Azevedo, and J. Zico Kolter. "Inverse Optimal Power Flow: Assessing the Vulnerability of Power Grid Data." *AI for Social Good Workshop at NeurIPS 2018.*

Logical reasoning layers



Problem (Visual Sudoku): Given an image of a partially–filled Sudoku board, output the Sudoku solution

Approach: End-to-end training of a CNN with a maximum satisfiability (logical reasoning) layer

Finding: MAXSAT layer implicitly learns the "logical structure" of the Sudoku rules (i.e., structure of maximum satisfiability clauses) to achieve good performance

Po-Wei Wang, Priya L. Donti, Bryan Wilder, Zico Kolter. "SATNet: Bridging deep learning and logical reasoning using a differentiable satisfiability solver." *ICML 2019.*
Zhaoyu Li, Jinpei Guo, Yuhe Jiang, Xujie Si. "Learning Reliable Logical Rules with SATNet." *NeurIPS 2023.*

Mechanisms for explainability, interpretability, & safety

Gray-box models: Creating model components we can recognize and understand

Verifying model behaviors: Testing that a model will behave as intended

Enforcing model behaviors: Engineering desired behaviors into models

Understanding and improving data: Ensuring data representativeness & reliability

Mechanisms for explainability, interpretability, & safety

Gray-box models: Creating model components we can recognize and understand

Verifying model behaviors: Testing that a model will behave as intended

Enforcing model behaviors: Engineering desired behaviors into models

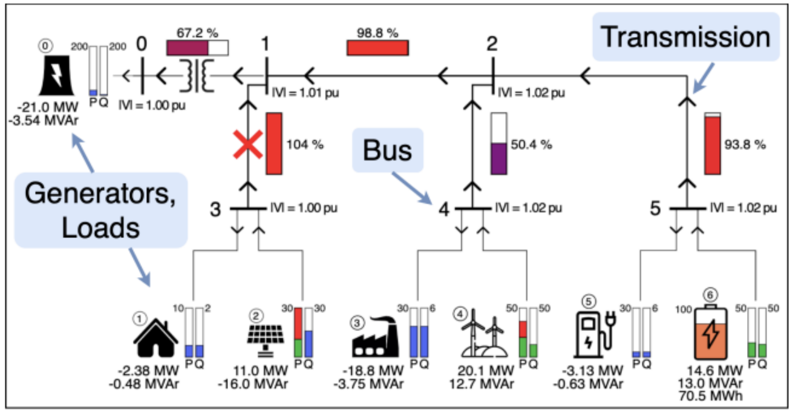
Understanding and improving data: Ensuring data representativeness & reliability

ε-retrain: Encouraging safety in RL via smart exploration

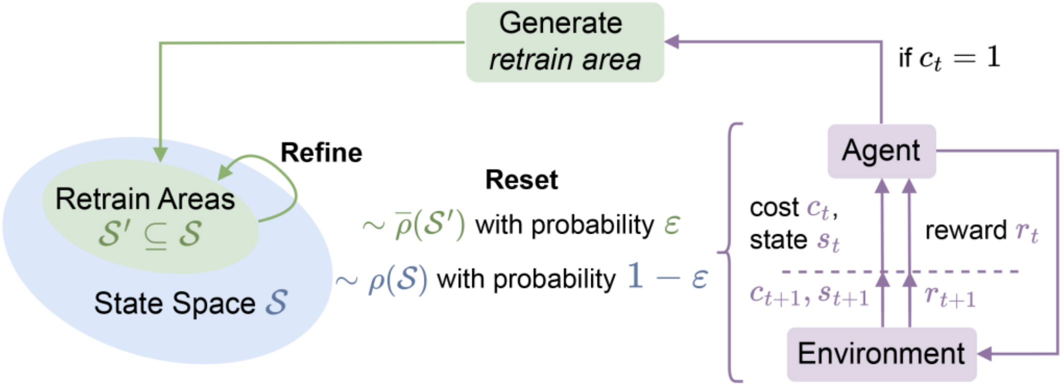
Problem: When training RL for domains such as power grids and robotics, can we prioritize behavioral preferences (e.g., safety desirada) while still getting good performance?

Approach: Modify the restart state distribution to combine standard uniform restart with restart over "retrain areas" where the agent violated a behavioral preference

Question: Beyond empirical experiments, how can we verify performance of the trained policy?



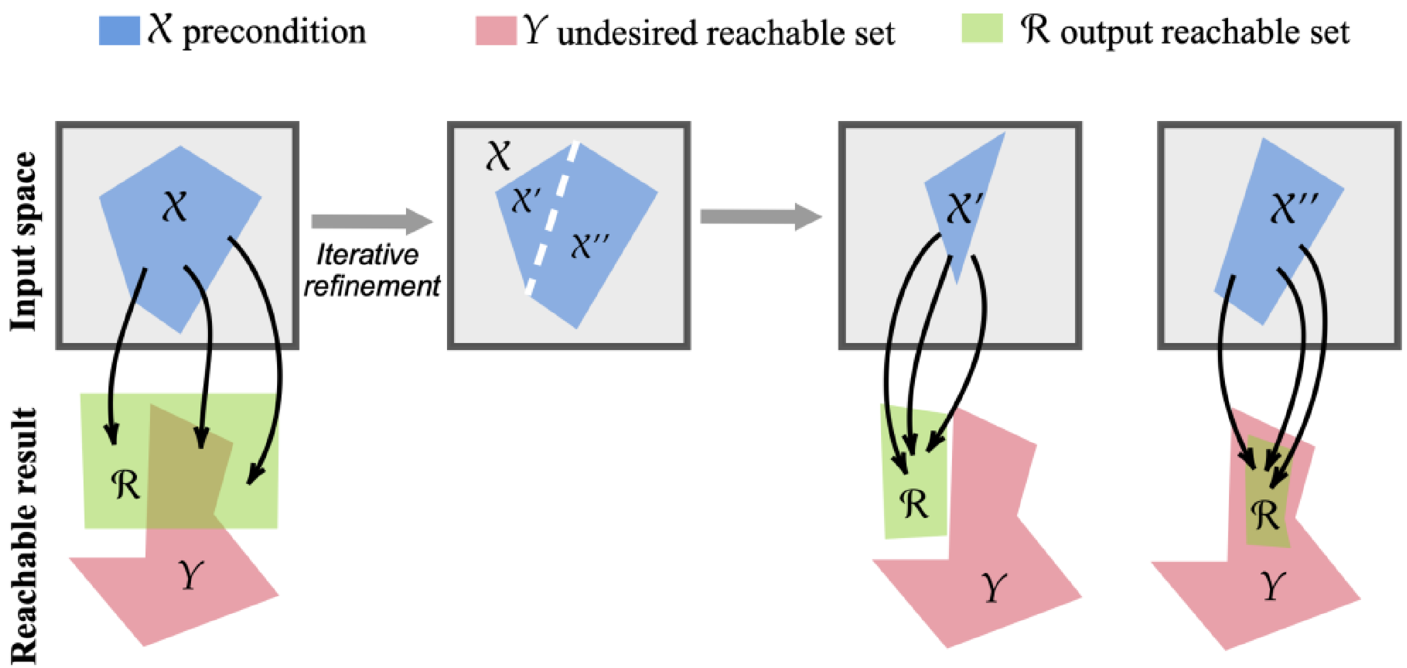
Active Network Management task



Luca Marzari, Changliu Liu, Priya L. Donti, Enrico Marchesini. "Improving Policy Optimization via ε-Retrain." Forthcoming at *AAMAS 2025*.

Toolkit: Formal verification

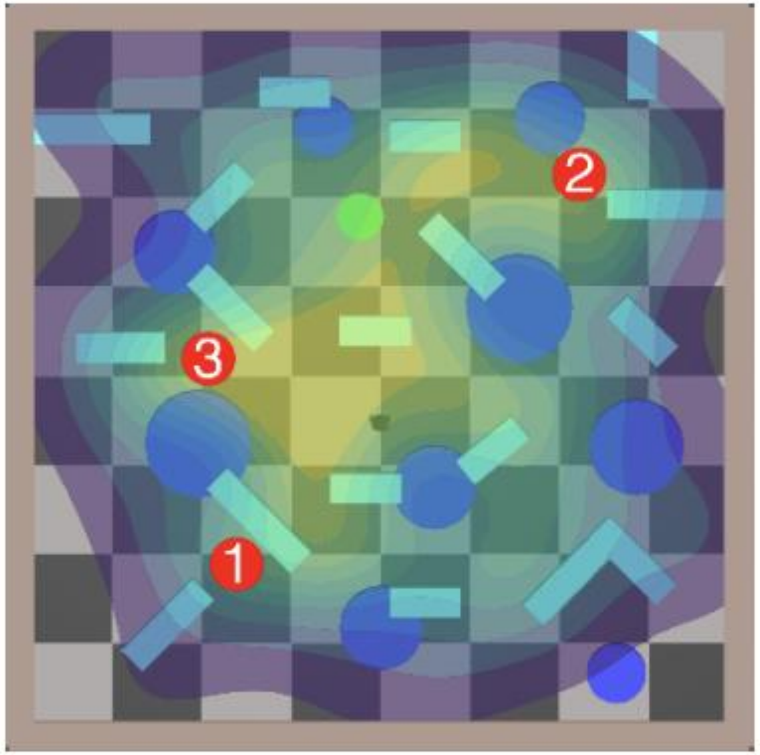
Reachability-based formal verification: Given trained policy $\mathcal{F}$ and input-output relationships $\langle\mathcal{X},\mathcal{Y}\rangle$, compute output-reachable set $\mathcal{R}(\mathcal{X},\mathcal{F})$ and check $\mathcal{R}(\mathcal{X},\mathcal{F}) \subseteq \mathcal{Y}$.



Luca Marzari, Changliu Liu, Priya L. Donti, Enrico Marchesini. "Improving Policy Optimization via ε-Retrain." Forthcoming at *AAMAS 2025.*
L. Marzari, D. Corsi, F. Cicalese, and A. Farinelli. The #DNN-Verification Problem: Counting Unsafe Inputs for Deep Neural Networks. *IJCAI 2023.*

Formal verification of ε-retrain policies

In robotic navigation setting, formal verification shows that ε-retrain-based algorithms better adhere to "collision avoidance" preference



	Retrain areas (1, 2, 3)		
ε-PPO	**0.007%**	**0.011%**	**0.22%**
PPO	0.012%	0.017%	0.59%
ε-TRPO	**0.014%**	**0.67%**	1%
TRPO	0.015%	0.69%	**0.8%**
ε-PPOLagr	**0.006%**	**0%**	**0.012%**
PPOLagr	0.013%	0.05%	0.1%
ε-TRPOLagr	0.0004%	**0.007%**	**0.46%**
TRPOLagr	**0.00005%**	0.012%	0.48%

Luca Marzari, Changliu Liu, Priya L. Donti, Enrico Marchesini. "Improving Policy Optimization via ε-Retrain." Forthcoming at *AAMAS 2025*.

Mechanisms for explainability, interpretability, & safety

- **Gray-box models:** Creating model components we can recognize and understand
- **Verifying model behaviors:** Testing that a model will behave as intended
- **Enforcing model behaviors:** Engineering desired behaviors into models
- **Understanding and improving data:** Ensuring data representativeness & reliability

Mechanisms for explainability, interpretability, & safety

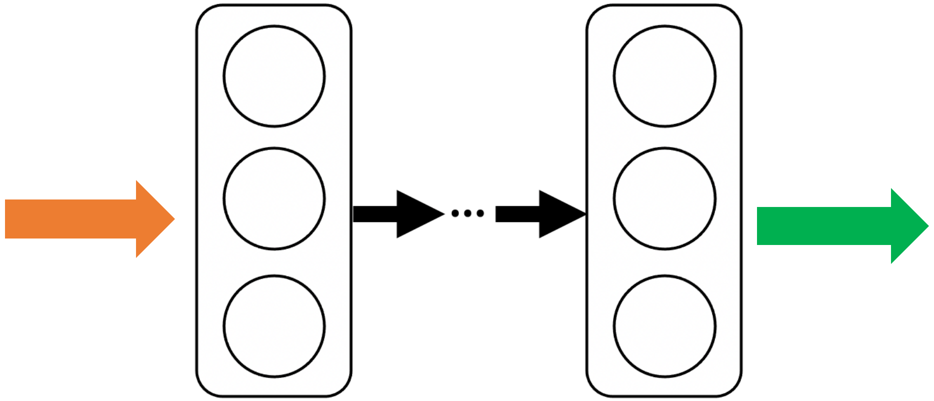
Gray-box models: Creating model components we can recognize and understand

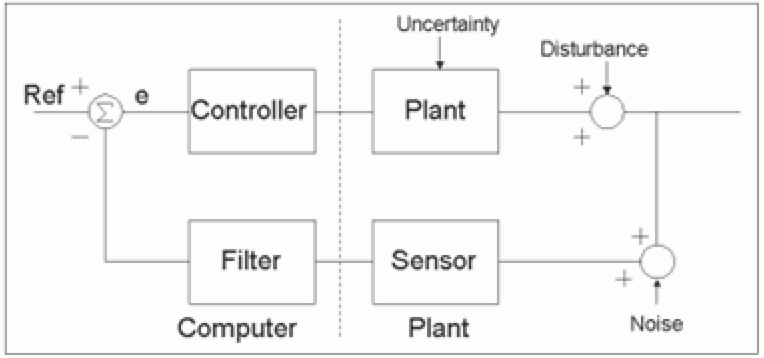
Verifying model behaviors: Testing that a model will behave as intended

Enforcing model behaviors: Engineering desired behaviors into models

Understanding and improving data: Ensuring data representativeness & reliability

Deep reinforcement learning vs. robust control





Deep RL

Pro: Expressive, well-performing policies
Con: Potential (catastrophic) failures

Robust control

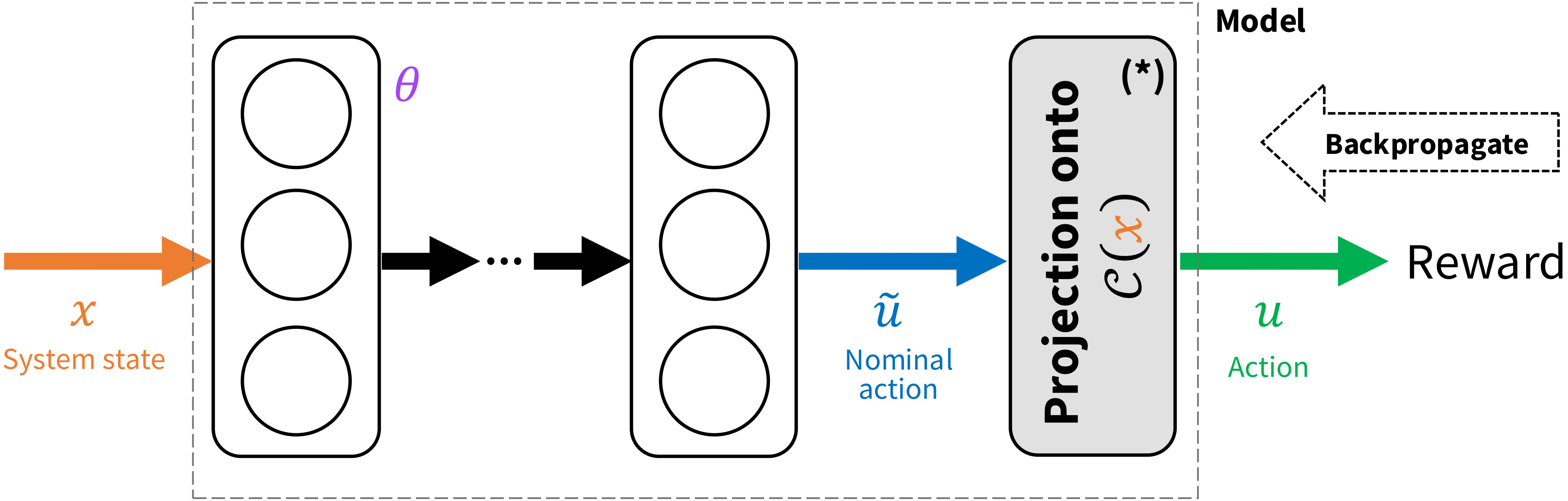
Pro: Provable stability guarantees
Con: Simple policies (e.g., linear)

Can we improve performance while still guaranteeing stability?

Priya L. Donti, Melrose Roderick, Mahyar Fazlyab, and J. Zico Kolter. "Enforcing robust control guarantees within neural network policies." *International Conference on Learning Representations (ICLR) 2021.*

Differentiable projection onto stabilizing actions

Deep learning-based policy with **provable robustness guarantees** (even for a randomly initialized neural network), trainable using reinforcement learning



Priya L. Donti, Melrose Roderick, Mahyar Fazlyab, and J. Zico Kolter. "Enforcing robust control guarantees within neural network policies." *International Conference on Learning Representations (ICLR) 2021.*

Finding a set of stabilizing actions (example)

Insight: Find a set of actions that are guaranteed to satisfy relevant Lyapunov stability criteria at a given state, even under worst-case conditions

Given the following (from robust control)**:**

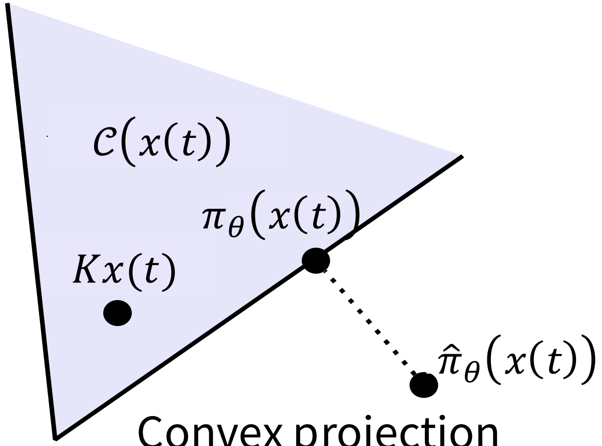
- Uncertainty model: e.g., $\dot{x}(t) \in Ax(t) + Bu(t) + Gw(t)$ s.t. $||w(t)||_2 \leq \ ||Cx(t) + Du(t)||_2$
- Lyapunov function V obtained via robust control synthesis
- Exponential stability criterion: $\dot{V}\big(x(t)\big) \leq -\alpha V(x(t)), \forall x \neq 0$

Find: For given x, set of actions satisfying exponential stability criterion even in worst case

$$\mathcal{C}(x) \equiv \left\{ u : \left(\sup_{w \,:\, \|w\|_2 \leq \|Cx + Du\|_2} \dot{V}(x) \right) \leq -\alpha V(x) \right\}$$

$$\Rightarrow \{u : \|k_1(x) + Du\|_2 \leq k_2(x) + k_3(x)^T u\}$$

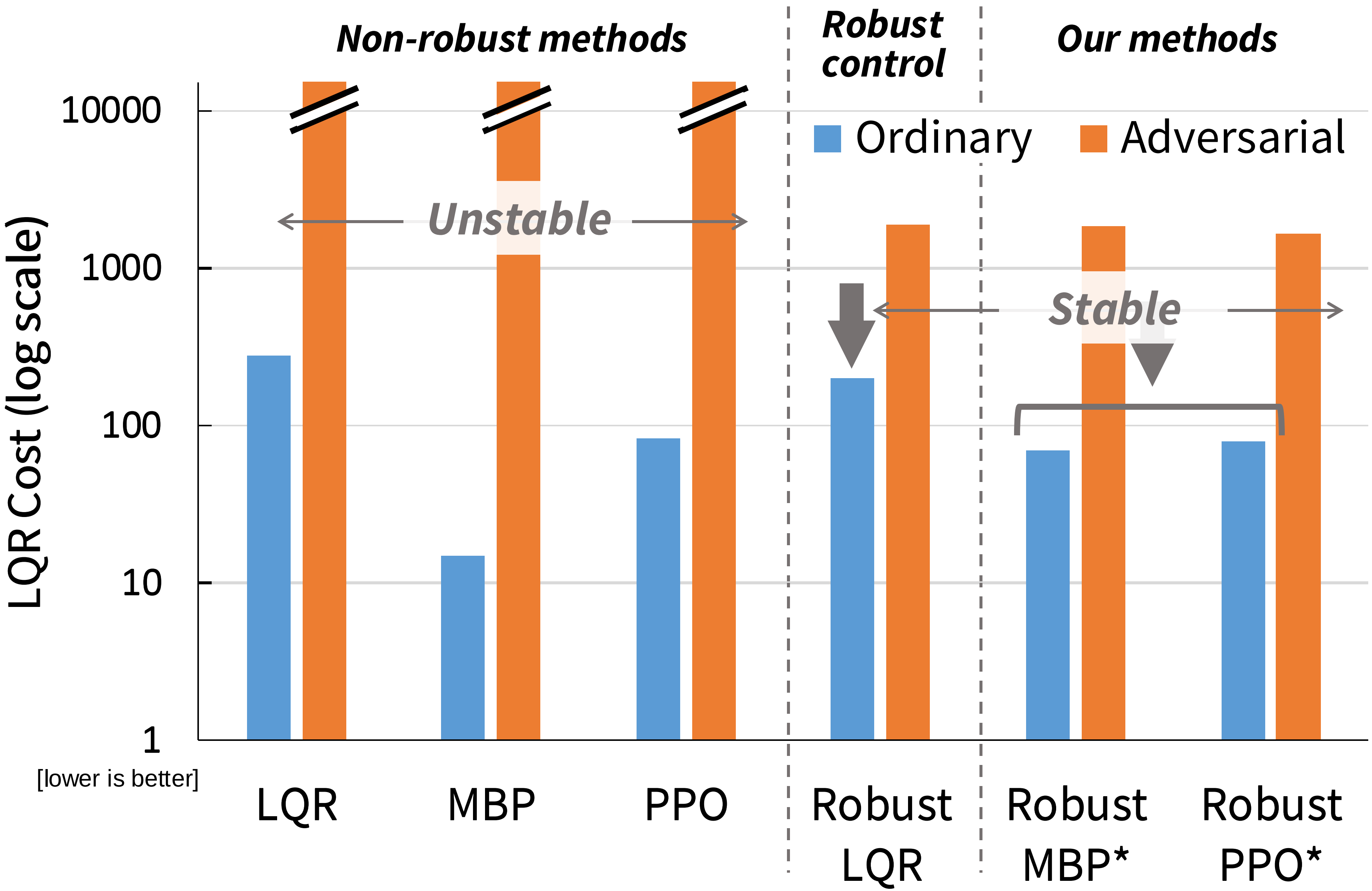
Convex (non-empty) set in $u(t)$



Convex projection

Note: t-dependence has been dropped for brevity

Illustrative results: Synthetic NLDI system



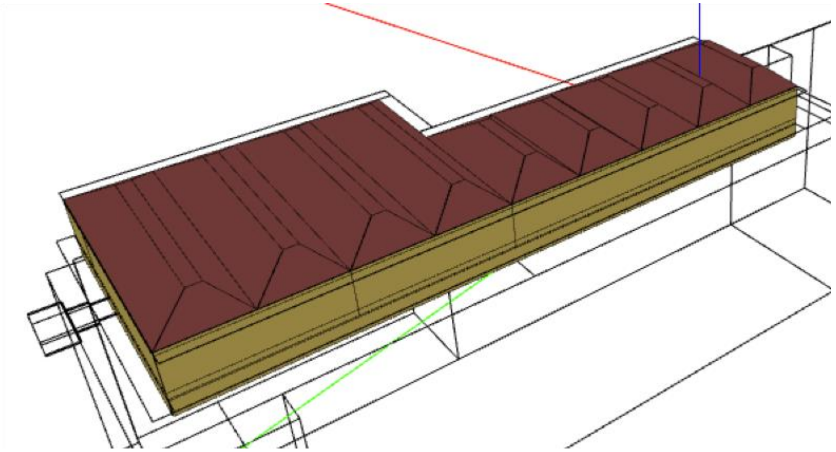
Improved "average-case" performance over robust baselines

Provably stable under "worst-case" dynamics (unlike non-robust baselines)

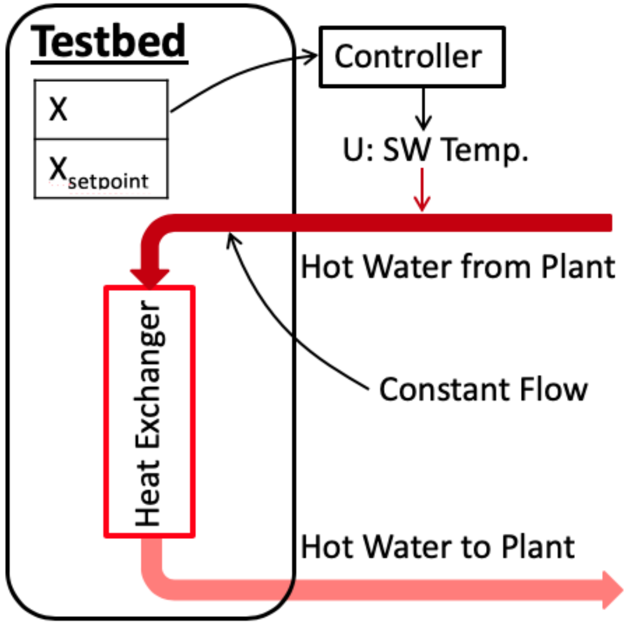
Downside: Speed / computational cost

Energy-efficient heating and cooling

Goal: Control the HVAC supply water temperature to minimize energy use, while respecting equipment constraints and maintaining thermal comfort



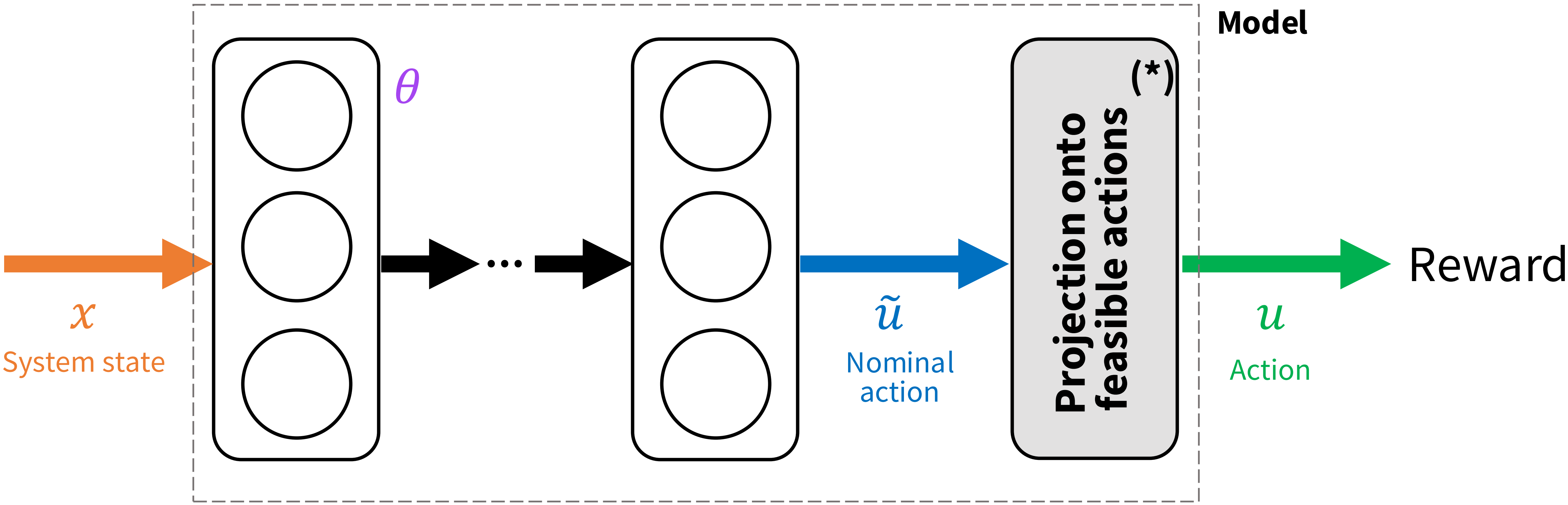
Intelligent Workplace
Margaret Morrison Hall, 4th Floor
(✤ Zhang & Lam, 2018)



HVAC Schematic

Bingqing Chen*, **Priya L. Donti***, Kyri Baker, J. Zico Kolter, and Mario Berges. "Enforcing Policy Feasibility Constraints through Differentiable Projection for Energy Optimization." *ACM International Conference on Future Energy Systems (ACM e-Energy) 2021*.

Differentiable projection onto feasible actions



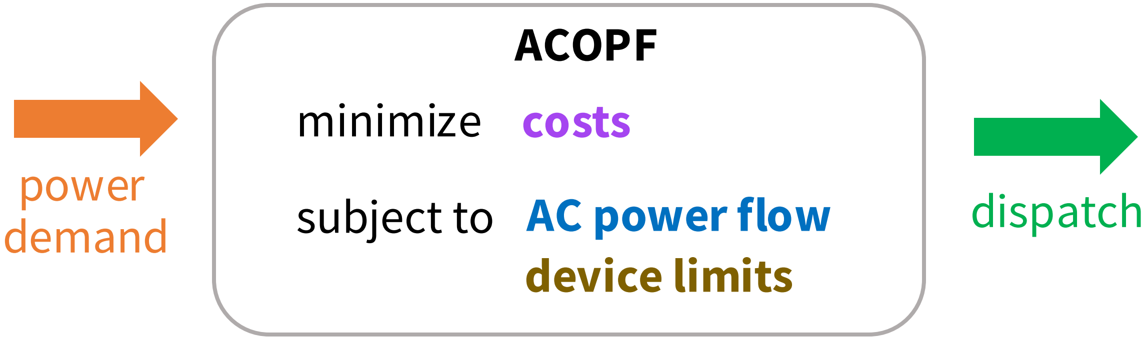
Results on realistic-scale building simulator

Improved energy efficiency (4-24%) **Comparable thermal comfort**

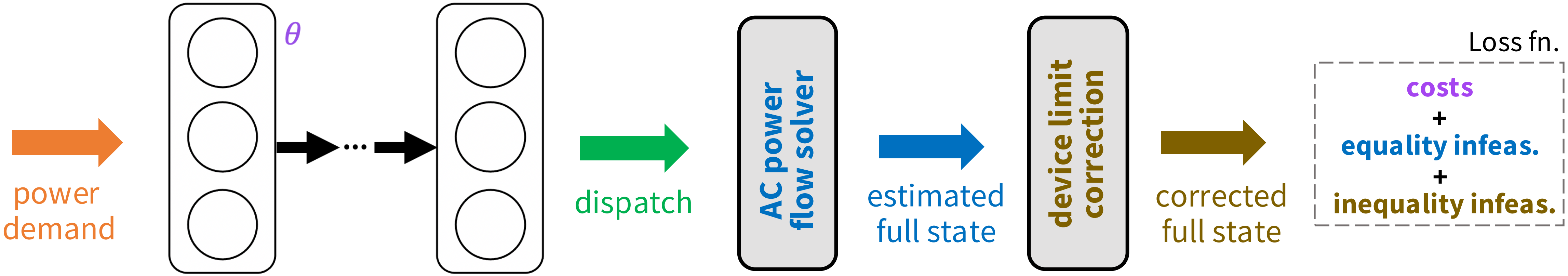
	Total Heating Demand (kWh)	Predicted Percentage Dissatisfied Mean (%)	Predicted Percentage Dissatisfied Std (%)
Existing controller	43709	**9.45**	5.59
Agent #6 (Zhang & Lam, 2018)	37131	11.71	3.76
Gnu-RL (Chen et al., 2019)	34678	9.56	6.39
PROF (ours)	**33271**	9.68	**3.66**

Feasible optimization proxies

Goal: Provide fast, feasible approx. to AC optimal power flow (ACOPF)



Approach:



Note: Learns directly from problem specification (no supervised training dataset)

Priya L. Donti*, David Rolnick*, and J. Zico Kolter. "DC3: A learning method for optimization with hard constraints." *International Conference on Learning Representations (ICLR) 2021.*

Approximating ACOPF: 57-bus test case

	Comparable objective value	Satisfies all constraints (unlike baselines)		10x faster than IPOPT
	Objective value	**Max equality violation**	**Mean equality violation**	**Time (s)**
IPOPT	3.81 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.949 ± 0.002
Baseline NN	–	$0.19 + 0.01$	0.03 ± 0.00	–
Our approach	3.82 ± 0.00	0.00 ± 0.00	0.00 ± 0.00	0.089 ± 0.000

Mechanisms for explainability, interpretability, & safety

Gray-box models: Creating model components we can recognize and understand

Verifying model behaviors: Testing that a model will behave as intended

Enforcing model behaviors: Engineering desired behaviors into models

Understanding and improving data: Ensuring data representativeness & reliability

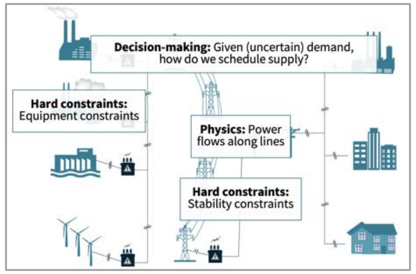
Mechanisms for explainability, interpretability, & safety

- **Gray-box models:** Creating model components we can recognize and understand
- **Verifying model behaviors:** Testing that a model will behave as intended
- **Enforcing model behaviors:** Engineering desired behaviors into models
- **Understanding and improving data:** Ensuring data representativeness & reliability

Priya L. Donti: donti@mit.edu

Road ahead: Bridging needs & mechanisms

Needs



Am I confident this model can be deployed?

What is being prioritized and/or missed?

How can we diagnose what went wrong?

Mechanisms

Gray-box models

Verifying model behaviors

Enforcing model behaviors

Understanding & improving data