Optimization-in-the-loop ML for energy and climate

Priya L. Donti
Executive Director, Climate Change AI
Incoming Assistant Professor, MIT
Climate change warrants rapid action

Impacts felt globally

Disproportionate impacts on most disadvantaged populations

Need net-zero greenhouse gas emissions by 2050 (IPCC 2018)

- Across energy, transport, buildings, industry, agriculture, forestry, etc.

Can machine learning play a role?
Tackling Climate Change with Machine Learning

David Rolnick1*, Priya L. Doni2, Lynn H. Kaack3, Kelly Kochanski4, Alexandre Lacoste5, Kris Sankaran6,7, Andrew Slavin Ross8, Nikola Milojevic-Dupon9,10, Natasha Jaques11, Anna Waldman-Brown11, Alexandra Luccioni6,7, Tegan Maharaj6,7, Evan D. Sherwin2, S. Karthik Mukkavilli6,7, Konrad P. Kording1, Carla Gomes12, Andrew Y. Ng13, Demis Hassabis14, John C. Platt15, Felix Creutzig9,10, Jennifer Chayes16, Yoshua Bengio6,7

1University of Pennsylvania, 2Carnegie Mellon University, 3ETH Zürich, 4University of Colorado Boulder, 5Element AI, 6Mila, 7Université de Montréal, 8Harvard University, 9Mercator Research Institute on Global Commons and Climate Change, 10Technische Universität Berlin, 11Massachusetts Institute of Technology, 12Cornell University, 13Stanford University, 14DeepMind, 15Google AI, 16Microsoft Research

Abstract
Climate change is one of the greatest challenges facing humanity, and we, as machine learning experts, may wonder how we can help. Here we describe how machine learning can be a powerful tool in reducing greenhouse gas emissions and helping society adapt to an changing climate. From smart grids to disaster management, we identify high impact problems where existing gaps can be filled by machine learning, in collaboration with other fields. Our recommendations encompass exciting research questions as well as promising business opportunities. We call on the machine learning community to join the global effort against climate change.

Introduction
The effects of climate change are increasingly visible. Storms, droughts, fires, and flooding have become stronger and more frequent [3]. Global ecosystems are changing, including the natural resources and agriculture on which humanity depends. The 2018 intergovernmental report on climate change estimated that the world will face catastrophic consequences unless greenhouse gas emissions are eliminated within thirty years [4]. Yet year after year, these emissions rise. Addressing climate change involves mitigation (reducing emissions) and adaptation (preparing for unavoidable consequences). Both are multifaceted issues. Mitigation of greenhouse gas (GHG) emissions requires changes to electricity systems, transportation, buildings, industry, and land use. Adaptation requires climate modeling, risk prediction, and planning for resilience and disaster management. Such a diversity of problems can be seen as an opportunity: there are many ways to have an impact.

In recent years, machine learning (ML) has been recognized as a broad powerful tool of technological progress. Despite the growth of movements applying ML and AI to problems of societal and global good, 2

Power & energy problems involve physics, hard constraints, and decision-making

**Physics:** Power flows along lines

**Hard constraints:** Equipment constraints

**Hard constraints:** Stability constraints

**Decision-making:** Given (uncertain) demand, how do we schedule supply?

Figure adapted from: US Congressional Budget Office
Machine learning methods struggle with physics, hard constraints, and decision-making

**Hard constraints:** Equipment constraints

**Physics:** Power flows along lines

**Hard constraints:** Stability constraints

**Decision-making:** Given (uncertain) demand, how do we schedule supply?

**Need:** Adaptive control of power generators, inverters, and batteries
- **ML:** Dynamic, data-driven control
- **Limitation:** Difficulty enforcing constraints (physics, equipment, stability)

**Need:** Electricity demand prediction
- **ML:** Time series forecasting
- **Limitation:** Difficulty making decision-cognizant error tradeoffs

How do we reap the benefits of ML methods while mitigating limitations?

Figure adapted from: US Congressional Budget Office
Optimization-in-the-loop ML

Framework for developing ML methods that incorporate knowledge of physics, hard constraints, or downstream decision-making procedures, via optimization problems.
Optimization-in-the-loop ML

Framework for developing ML methods that incorporate knowledge of physics, hard constraints, or downstream decision-making procedures, via optimization problems.

Model

$h_\theta$

$x$

Inputs

$h_\theta(x)$

Outputs

Functional form of model

Model parameters
Optimization-in-the-loop ML

Framework for developing ML methods that incorporate knowledge of physics, hard constraints, or downstream decision-making procedures, via optimization problems

**Example:** Robust control (e.g., power system control with stability constraints)

**Example:** Prediction (e.g., decision-cognizant demand forecasting)
Talk outline

Optimization-in-the-loop ML

- Toolkit: Differentiable optimization
- Setting: Hard control constraints
- Setting: Downstream decision-making
Talk outline

Optimization-in-the-loop ML

Toolkit: Differentiable optimization

Setting: Hard control constraints

Setting: Downstream decision-making
Overview: Differentiable optimization

**Motivation:** Need for tools to implement optimization-in-the-loop methods

**Approach:** Differentiable optimization in deep learning
- General framework [GFCASG2016, AK2017, DAK2017]
- Additional tools
Background: Deep learning

Model

$h_\theta$

Inputs $x$

Functional form of model

Model parameters

$h_\theta(x)$

Outputs

Loss, e.g., $\ell(y, h_\theta(x))$

“Score” for quality of output
Background: Deep learning

- Neural network $h_\theta = \text{composition of nonlinear, parameterized functions (layers)}$
- Update parameters $\theta$ to minimize loss $\ell$ using gradients from backpropagation
- All components (layers and loss) must be differentiable
Differentiating through optimization problems

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

**Example optimization problem**

\[
\begin{align*}
\text{minimize} \quad & \frac{1}{2} z^T Q z + q^T z \\
\text{subject to} \quad & A z = b \\
& G z \leq h
\end{align*}
\]

**Selected KKT optimality conditions**

\[
\begin{align*}
Q z^* + q + A^T \nu^* + G^T \lambda^* &= 0 \\
A z^* - b &= 0 \\
\text{diag}(\lambda^*)(G z^* - h) &= 0
\end{align*}
\]

**Step 1:** Apply implicit function theorem to the KKT conditions

\[
\begin{bmatrix}
Q & G^T \\
\text{diag}(\lambda^*)G & \text{diag}(G z^* - h) \\
A & 0
\end{bmatrix}
\begin{bmatrix}
dz \\
d\lambda \\
d\nu
\end{bmatrix}
= -
\begin{bmatrix}
dQ z^* + dq + dG^T \lambda^* + dA^T \nu^* \\
\text{diag}(\lambda^*)dG z^* - \text{diag}(\lambda^*)dh \\
dAz^* - db
\end{bmatrix}
\]

**Step 2:** Use “Jacobian-vector trick” for efficient backpropagation


Follow-on work in differentiable optimization

[DAK2017, AK2017]: KKT differentiation techniques for convex optimization problems

Many additional tools since then:
- Combinatorial optimization [DK2017, TSK2018, WDT2018]
- AC optimal power flow [DAK2018]
- Disciplined convex programs [AABBDK2019]
- Maximum satisfiability problems [WDWK2019]
- Additional optimization problems [GHC2019]

Powerful toolkit for optimization-in-the-loop ML in the context of deep learning
Talk outline

Optimization-in-the-loop ML

- Toolkit: Differentiable optimization
- Setting: Hard control constraints
- Setting: Downstream decision-making
Overview: Enforcing hard control constraints

**Motivation:** Need for well-performing control methods that also guarantee enforcement of hard constraints

**Approach:** Optimization-in-the-loop reinforcement learning (RL) techniques with guaranteed enforcement of hard constraints

**Settings:**
- Asymptotic stability in power grids [DRFK2021]
- Realistic-scale building control [CDBKB2021]
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?

Differentiable projection onto stabilizing actions

Deep learning-based policy with **provable robustness guarantees**, trainable using standard reinforcement learning approaches

\[ \text{System state} \xrightarrow{} \theta \xrightarrow{} \ldots \xrightarrow{} \tilde{u} \xrightarrow{\text{Projection onto } C(\chi)} u \xrightarrow{\text{Backpropagate}} \text{Reward} \]
Details: Finding a set of stabilizing actions

**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: \( \dot{V}(x(t)) \leq -\alpha V(x(t)), \forall x \neq 0 \)

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

\[
\mathcal{C}(x) \equiv \{ u : \sup_{w : \|w\|_2 \leq \|Cx+Du\|_2} \dot{V}(x) \leq -\alpha V(x) \}
\]

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

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

Note: \( t \)-dependence has been dropped for brevity
Illustrative results: Synthetic NLDI system

**Non-robust methods**

- LQR
- MBP
- PPO

**Robust control**

- Robust LQR
- Robust MBP*
- Robust PPO*

**Our methods**

- Ordinary
- Adversarial

**LQR Cost (log scale)**

- Ordinary
- Adversarial

**Improvement**

- Improved "average-case" performance over robust baselines

**Stability**

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

**Downside**

- Downside: Speed / computational cost

[lower is better]
Energy-efficient heating and cooling

**Goal:** Control the HVAC supply water temperature to minimize energy use, while respecting equipment constraints and maintaining thermal comfort
Differentiable projection onto feasible actions

\( x \)  
System state

\( \theta \)

\( \ldots \)

\( \tilde{u} \)
Nominal action

Projection onto feasible actions

(\( (*) \))

\( u \)
Action

Reward
Summary: Enforcing hard control constraints

**Motivation:** Need for well-performing control methods that also guarantee enforcement of hard constraints

**Settings:**
- Asymptotic stability in power grids [DRFK2021]
- Realistic-scale building control [CDKBK2021]

**Insight:** Project outputs of neural network onto a set of “safe” actions
- Obtain safe actions using domain knowledge
- Differentiable projection (optimization layer) = end-to-end training

**Future directions:**
- Additional paradigms for bridging RL and robust control
- Improving computational costs
Talk outline

- Toolkit: Differentiable optimization
- Setting: Hard control constraints
- Setting: Downstream decision-making
Overview: Incorporating downstream decision-making

Motivation: Predictive methods operate within some larger decision-making process but do not often take this into account, potentially leading to critical mistakes.

Approach: Construction of decision-cognizant ("task-based") models via optimization-in-the-loop learning

Settings:
- Decision-cognizant electricity demand forecasting [DAK2017]
- Approximating AC optimal power flow [DRK2021]
Decision-cognizant demand forecasting

\[ \text{Past demand, weather, time} \equiv x \]

\[ h_\theta ? \]

\[ \text{Generation schedule (e.g.)} \equiv z \]

\[ \text{Future demand (w/ uncertainty)} \equiv y \]

**Goal:** Optimize for quality of generation schedule when we observe actual demands

\[ \min_{\theta} f_c(y, z^*(x; \theta)) \]


Decision-cognizant model

\[ f_c(y, z^*) \]

\( \chi \)
Past demand, weather, time

\( \theta \)

\( \ldots \)

\( \hat{y} \)
Predicted demand

Power system optimization

Loss function

\( z^* \)
Generation schedule
Decision-cognizant approach can dramatically improve generation scheduling outcomes.

Decision-cognizant approach gives ~39% improvement in decision cost.
**Approximating AC optimal power flow**

**Goal:** Provide fast, feasible approximations to AC optimal power flow (ACOPF)

**Approach:**


**Results** (57-bus test case): High-quality solutions 10x faster than baseline optimizer
Motivation: Predictive methods operate within some larger decision-making process but do not often take this into account, potentially leading to critical mistakes.

Settings:
- Electricity demand forecasting [DAK2017]
- Approximating ACOPF [DRK2021]

Insight: Incorporate knowledge of downstream decision-making (or physics) into the loss function, via differentiable optimization.

Future directions:
- Incorporating a wider range of decision-making paradigms
- Understanding tradeoffs between task-agnostic vs. task-based models
Summary

- **Optimization-in-the-loop ML (framework)**, via differentiable optimization in deep learning

- **Enforcing hard control constraints**: RL with provable robustness / constraint enforcement
  - Asymptotic stability (power grids)
  - Operational constraints (HVAC in buildings)

- **Incorporating downstream decision-making**: Decision-cognizant predictive models
  - Electricity demand forecasting
  - Approximate power system optimization

**Priya L. Donti**: donti@mit.edu