NeurIPS 2026

NEURAL INFORMATION PROCESSING SYSTEMS

Masked Sobolev Training for Feasibility-Reliable Optimization Proxies

Learning a solution map shifts the burden of generalization from the solver to the learned model. **What must the model retain to stay reliable?**

Andrew W. Rosemberg[1] Joaquim Dias Garcia[2] Russell Bent[3] Pascal Van Hentenryck[1]

[1]Georgia Institute of Technology (AI4OPT) [2]PSR [3]Los Alamos National Laboratory

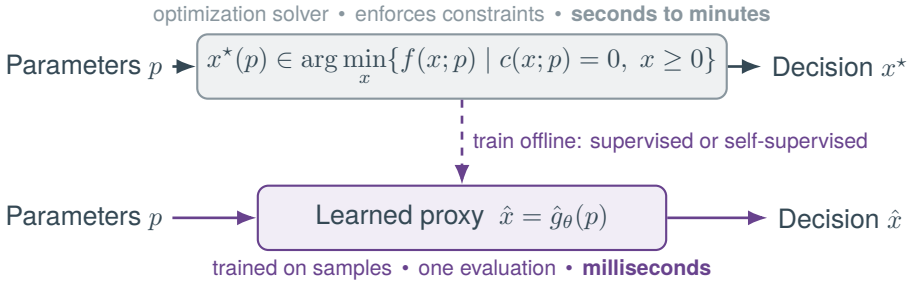
Georgia Tech

NSF AI4OPT

Los Alamos NATIONAL LABORATORY

PSR

Optimization proxies: solve offline, decide in milliseconds

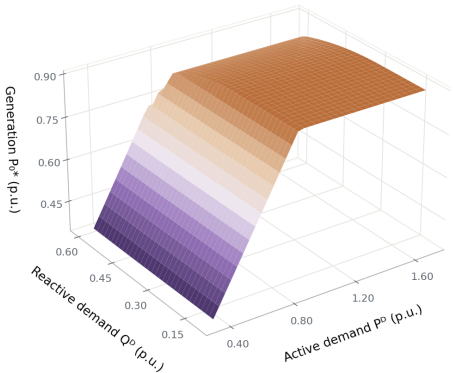


Real-time grid dispatch

Control & planning

Portfolio allocation

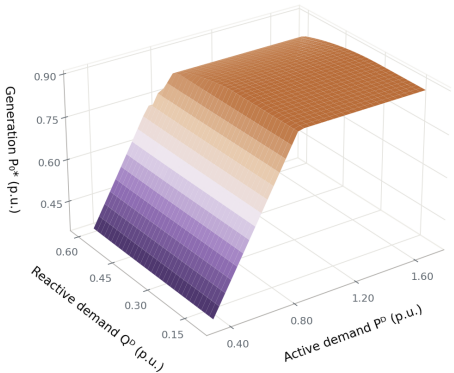
The catch: accurate on average is not reliable



Optimal generation vs. demand:
smooth within regimes, not across them.

- Solution maps are smooth only **within active-set regimes**.
- The parameter space splits into **combinatorially many** critical regions.

The catch: accurate on average is not reliable

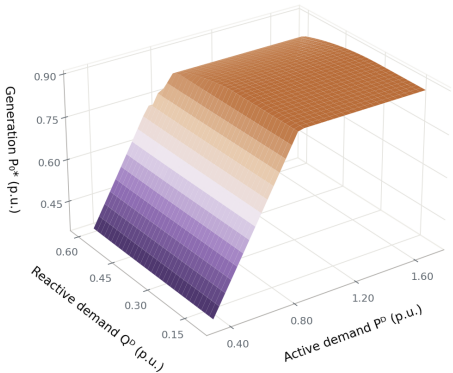


Optimal generation vs. demand:

smooth within regimes, not across them.

- Solution maps are smooth only **within active-set regimes**.
- The parameter space splits into **combinatorially many** critical regions.
- Low average error can hide **constraint violations** and large worst-case errors.

The catch: accurate on average is not reliable



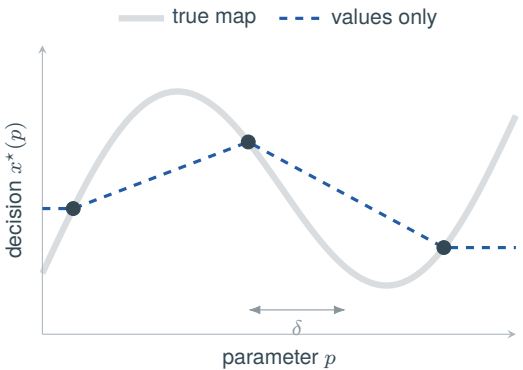
Optimal generation vs. demand:
smooth within regimes, not across them.

- Solution maps are smooth only **within active-set regimes**.
- The parameter space splits into **combinatorially many** critical regions.
- Low average error can hide **constraint violations** and large worst-case errors.

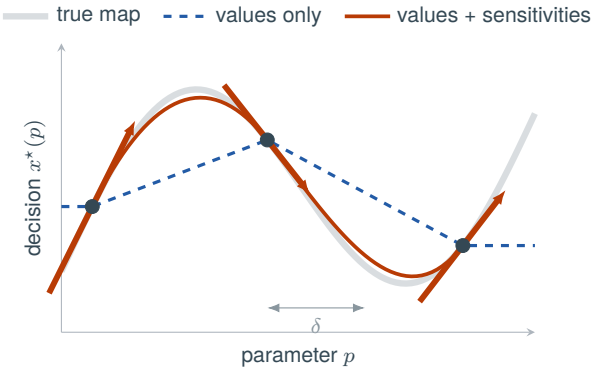
Reliable generalization from finite samples is the bottleneck.

What should the proxy retain? *How* the decision moves

Each sample carries a **decision** $x^\star(p_i)$



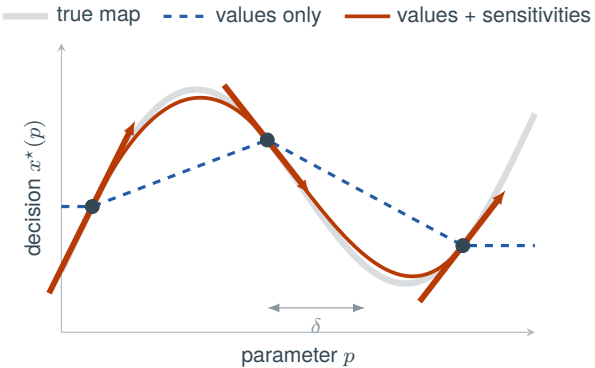
What should the proxy retain? *How* the decision moves



Each sample carries a **decision** $x^\star(p_i)$ and a **local linearization** $\partial x^\star / \partial p\,(p_i)$

from the KKT system, via implicit differentiation (DiffOpt.jl)

What should the proxy retain? *How* the decision moves



Each sample carries a **decision** $x^\star(p_i)$ and a **local linearization** $\partial x^\star / \partial p\,(p_i)$

from the KKT system, via implicit differentiation (DiffOpt.jl)

Our bounds, within a regular region (covering radius δ):

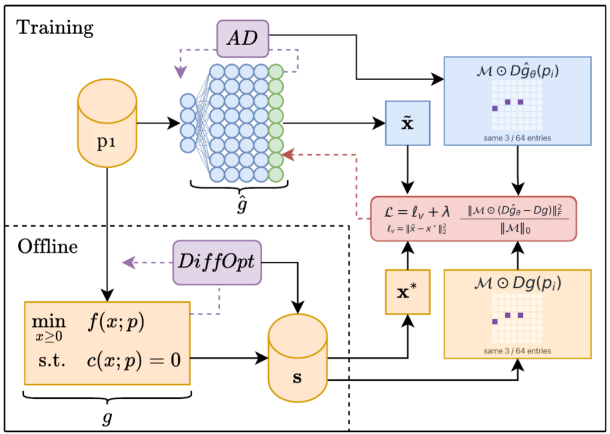
values only $\quad \|\hat{g}_\theta - g\|_\infty \leq (L_g + L_{\hat{g}})\,\delta$

+ sensitivities $\quad \|\hat{g}_\theta - g\|_\infty \leq \frac{1}{2}(M_g + M_{\hat{g}})\,\boldsymbol{\delta^2}$

Worst-case error bound improves from linear to **quadratic** in δ.

Theorems 1 & 3 in the paper.

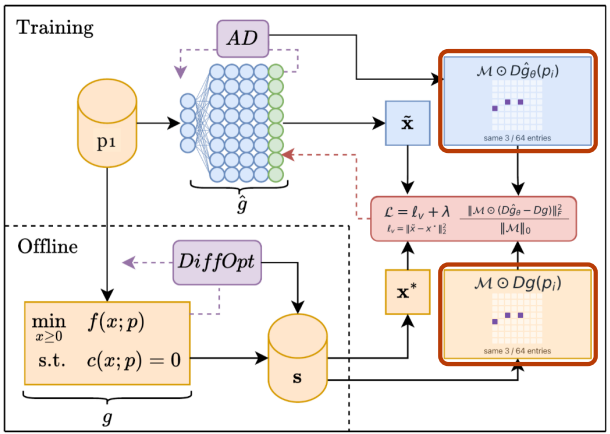
Masked Sobolev training



Matching the full Jacobian has two problems:

- $n \times d$ entries: **costly** for large problems
- dense matching **over-constrains**: conditions from different active sets compete

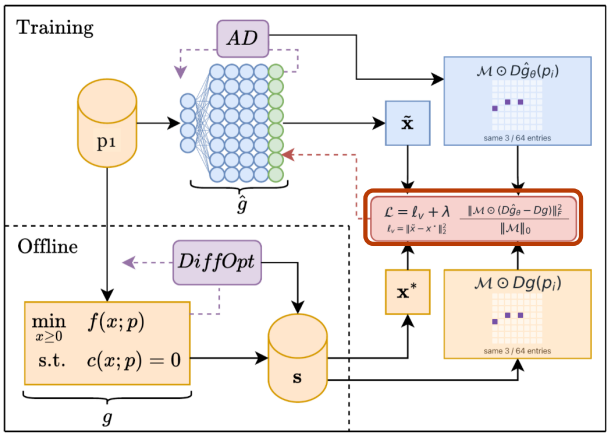
Masked Sobolev training



A fresh, sparse, uniformly random mask $\mathcal{M}$
at each update selects *which* Jacobian entries are matched.

Beyond cutting cost, the mask is an **algorithmic control** on how many derivative conditions are imposed at once.

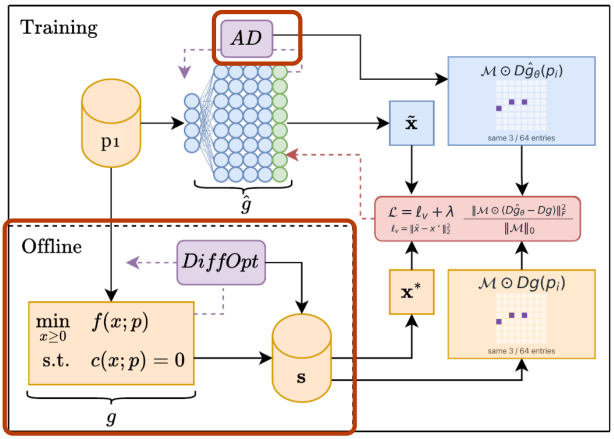
Masked Sobolev training



$$\mathcal{L} = \ell_v + \lambda \frac{\|\mathcal{M} \odot (D\hat{g}_\theta - Dg)\|_F^2}{\|\mathcal{M}\|_0}$$

- λ: **how much** derivative weight
- density of $\mathcal{M}$: **how many** conditions at once
- normalization $\Rightarrow$ **unbiased** estimate of the full Jacobian loss

Masked Sobolev training

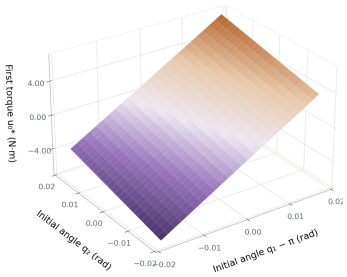


Solver solution targets and sensitivities: computed **offline**. Proxy sensitivities: only **during training**.

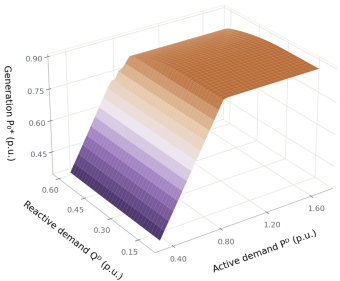
At deployment the proxy is still **one evaluation**: no extra inference cost.

Works with **any differentiable proxy**, and composes with self-supervised losses and repair / projection layers.

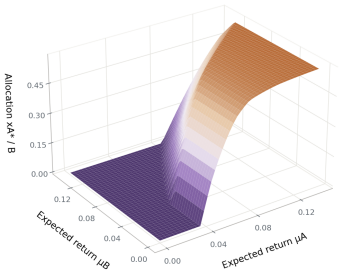
When does it help? Three solution-map geometries



Optimal control
smooth

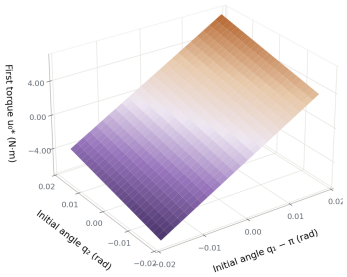


AC optimal power flow
highly constrained



Mean–variance portfolio
sparse, flat + sharp

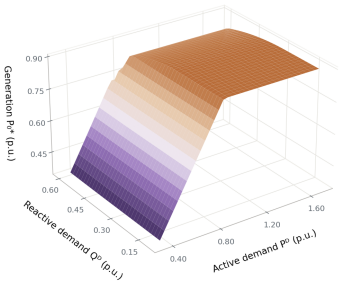
When does it help? Three solution-map geometries



Optimal control

smooth

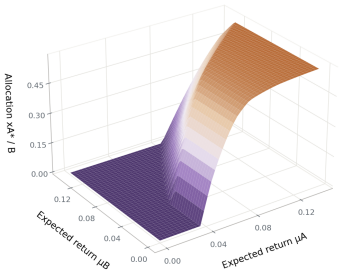
⇒ faster learning



AC optimal power flow

highly constrained

⇒ better feasibility

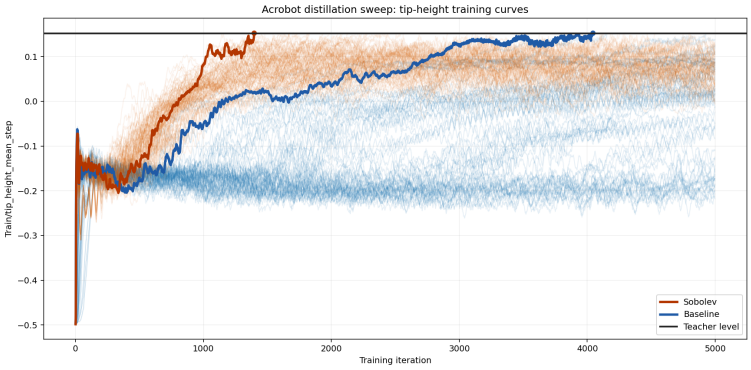


Mean–variance portfolio

sparse, flat + sharp

⇒ limited benefit

Smooth regime: Acrobot swing-up distillation



Same DAgger pipeline; **Sobolev** also matches selected state-to-action sensitivities.

2.5×
fewer iterations to 95% of reference performance

16 vs. 40 min
wall-clock, despite +2% cost per iteration

95% vs. 4%
of sweep runs reach the threshold

Constrained regime: AC optimal power flow, up to 6,468 buses

Same architecture: **value-only** vs. masked **Sobolev**. No repair layer.

	Opt. gap		Mean infeas.	
Grid	Value	Sob.	Value	Sob.
IEEE-300	0.02%	0.22%	0.0116	**0.0099**
PEGASE-1k	0.07%	0.15%	0.0098	**0.0066**
RTE-6k	0.07%	0.07%	0.0059	**0.0042**

Optimality gaps stay **below 0.22%**.

Constrained regime: AC optimal power flow, up to 6,468 buses

Same architecture: **value-only** vs. masked **Sobolev**. No repair layer.

	Opt. gap		Mean infeas.	
Grid	Value	Sob.	Value	Sob.
IEEE-300	0.02%	0.22%	0.0116	**0.0099**
PEGASE-1k	0.07%	0.15%	0.0098	**0.0066**
RTE-6k	0.07%	0.07%	0.0059	**0.0042**

Mean infeasibility **15–33% lower** on every grid, while gaps stay **below 0.22%**.

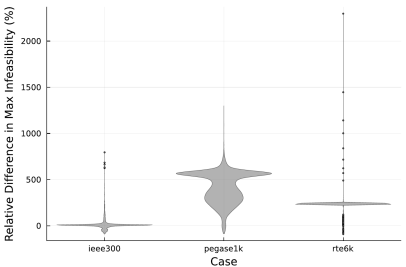
Constrained regime: AC optimal power flow, up to 6,468 buses

Same architecture: **value-only** vs. masked **Sobolev**. No repair layer.

	Opt. gap		Mean infeas.	
Grid	Value	Sob.	Value	Sob.
IEEE-300	0.02%	0.22%	0.0116	**0.0099**
PEGASE-1k	0.07%	0.15%	0.0098	**0.0066**
RTE-6k	0.07%	0.07%	0.0059	**0.0042**

Mean infeasibility **15–33% lower** on every grid, while gaps stay **below 0.22%**.

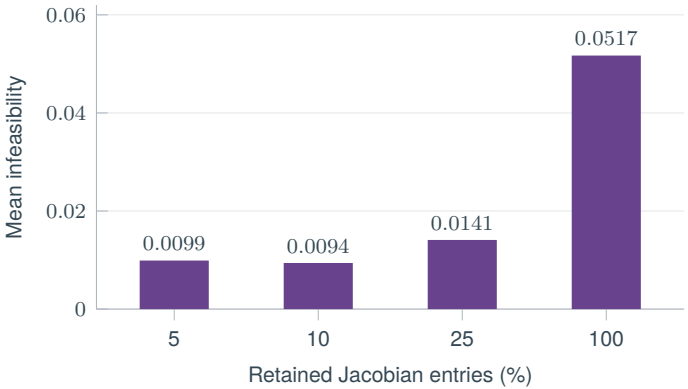
Worst-case violation per instance



Median worst-case violation reduced by **up to 4×**; **Sobolev** is worse in **<15%** of instances.

Positive = Sobolev better (normalized RMI, see paper).

The mask matters: sparse beats dense



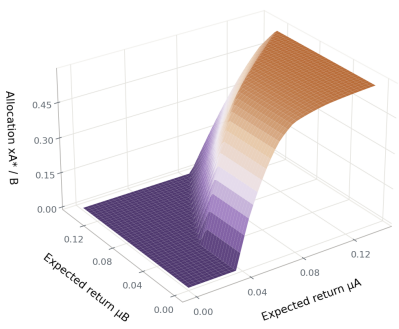
IEEE-300 mask-density ablation, same Sobolev setup.

5.2×

lower infeasibility with a **5% mask** than dense matching

Dense Jacobian matching **over-constrains** training: derivative conditions from different active sets compete.

Limitation: when sensitivities are too local

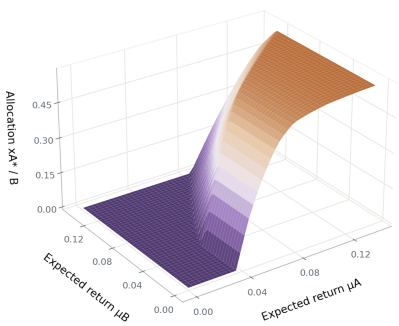


Sparse portfolios: many derivatives are zero, then change sharply.

Mean–variance portfolio (convex, SOC risk cap $\sigma_{\max}$).
Test optimality gap (%), mean $\pm$ std:

Risk cap / budget	Sobolev	Objective-only
tight ($\leq 10\%$)	29.5 ± 26.9	$\mathbf{14.7 \pm 21.2}$
loose ($> 10\%$)	$\mathbf{8.7 \pm 9.5}$	18.9 ± 20.2

Limitation: when sensitivities are too local



Sparse portfolios: many derivatives are zero, then change sharply.

Mean–variance portfolio (convex, SOC risk cap $\sigma_{\max}$).
Test optimality gap (%), mean $\pm$ std:

Risk cap / budget	Sobolev	Objective-only
tight ($\leq 10\%$)	29.5 ± 26.9	$\mathbf{14.7 \pm 21.2}$
loose ($> 10\%$)	$\mathbf{8.7 \pm 9.5}$	18.9 ± 20.2

A valid sensitivity at one sample can give **little guidance** at the next.

Use derivative supervision **selectively by regime** (e.g. a mixture of experts).

Summary

Learn how decisions respond to the data, not just the decisions.

- **Sensitivities** make optimization proxies more reliable when they stay informative about nearby decisions.
- **Masking** makes them practical, limiting how many derivative conditions compete at once.
- **The decision problem**, through its objectives and constraints, **determines how much they help:** faster learning when smooth, better feasibility when highly constrained, little help when sensitivities are too local.

Paper, code & website

Thank you!

arosemberg3@gatech.edu