Differentiable Cutting-Plane Layers

For Mixed-Integer Linear Optimization

Gabriele Dragotto

CISS 2024

Princeton University

March 12-15 2024

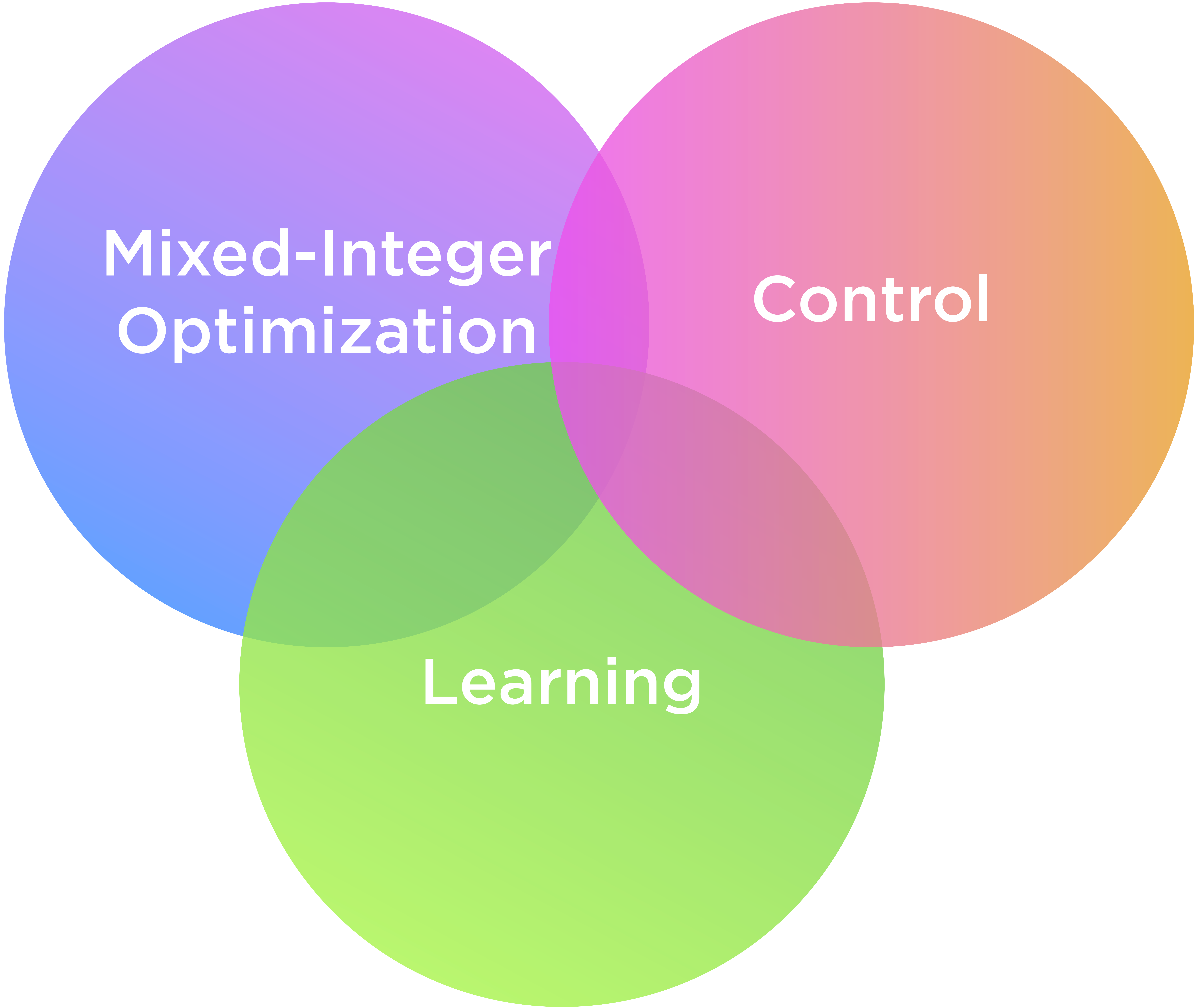
Accelerating Scientific Discovery at Princeton

Our team

Stefan Clarke
ORFE, Princeton

Jaime Fernandez Fisac
ECE, Princeton

Bartolomeo Stellato
ORFE, Princeton


Mixed-Integer Optimization
Control
Learning

Discrete decisions are ubiquitous

Supply Chain and Transportation

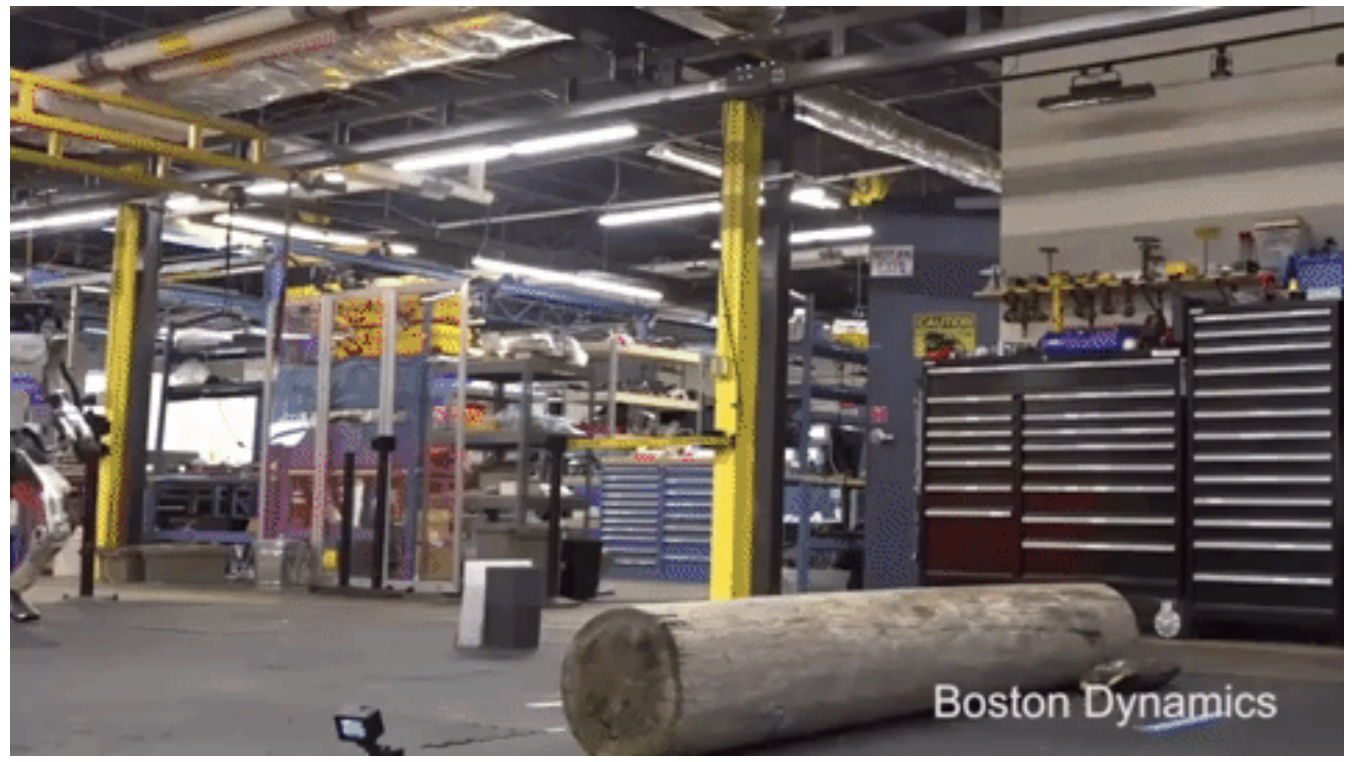
Vehicle routing

Assortment decisions

Energy

On-and-off switches

Unit commitment



Robotics

Hybrid dynamics

As **new data** arrives, we need to solve these problems in **real-time**

Mixed-integer solvers are not made for this!

Despite tremendous progress, solvers are extremely efficient at **solving one instance**

⚙️ Hard to warm-start

📈 High memory requirements

Parametric mixed-integer problems

As **new data** arrives, we need to solve these problems in **real-time**

Parameters

Decisions

$$\begin{array}{rl} \text{minimize} & c(\theta)^\top x \\ \text{subject to} & A(\theta)x \le b(\theta), \\ & x_i \in \mathbf{Z}, \quad \forall i \in I \end{array}$$

Feasible region

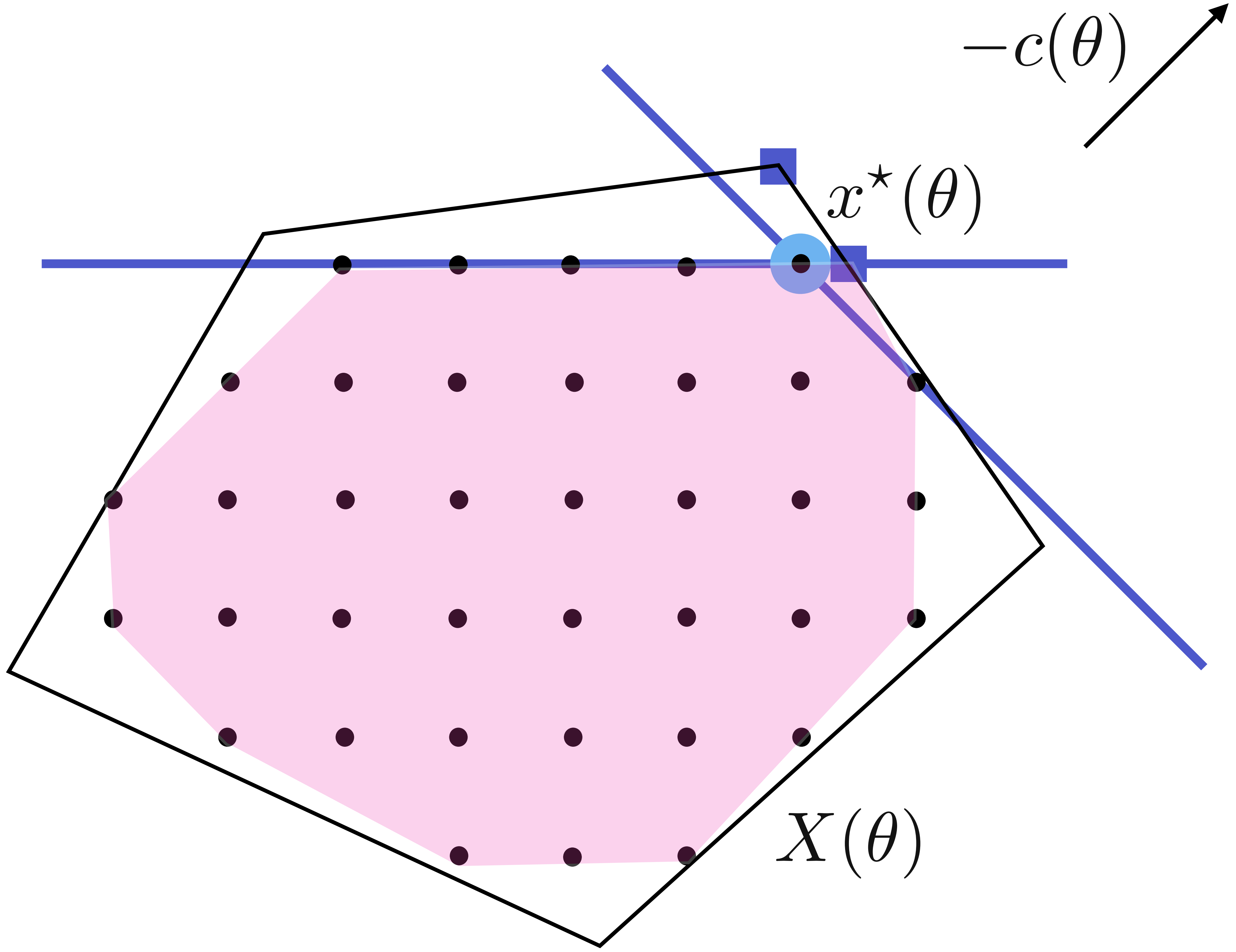
$X(\theta)$

Our goal

Build a computational architecture that exploits the *repetitive nature* of this family

Cutting-plane algorithms

Cutting plane algorithms



continuous relaxation

$$
\begin{array}{ll}
\text{minimize} & c(\theta)^\top x \\
\text{subject to} & A(\theta)x \le b(\theta) \\
& \boxed{Gx \le h}
\end{array}
$$

They are valid

$$X(\theta) \subseteq \{x \mid g_i^T x \le h_i\}$$

They cut off some $\bar{x}$

$$g_i^T \bar{x} > h_i$$

Gilmore and Gomory (1961) and (1963), Kelley (1960)

Cutting is a complex business

There are several **structural decisions** involved in cutting

How to select cuts?

How many cuts?

How many to store?

When to restart?

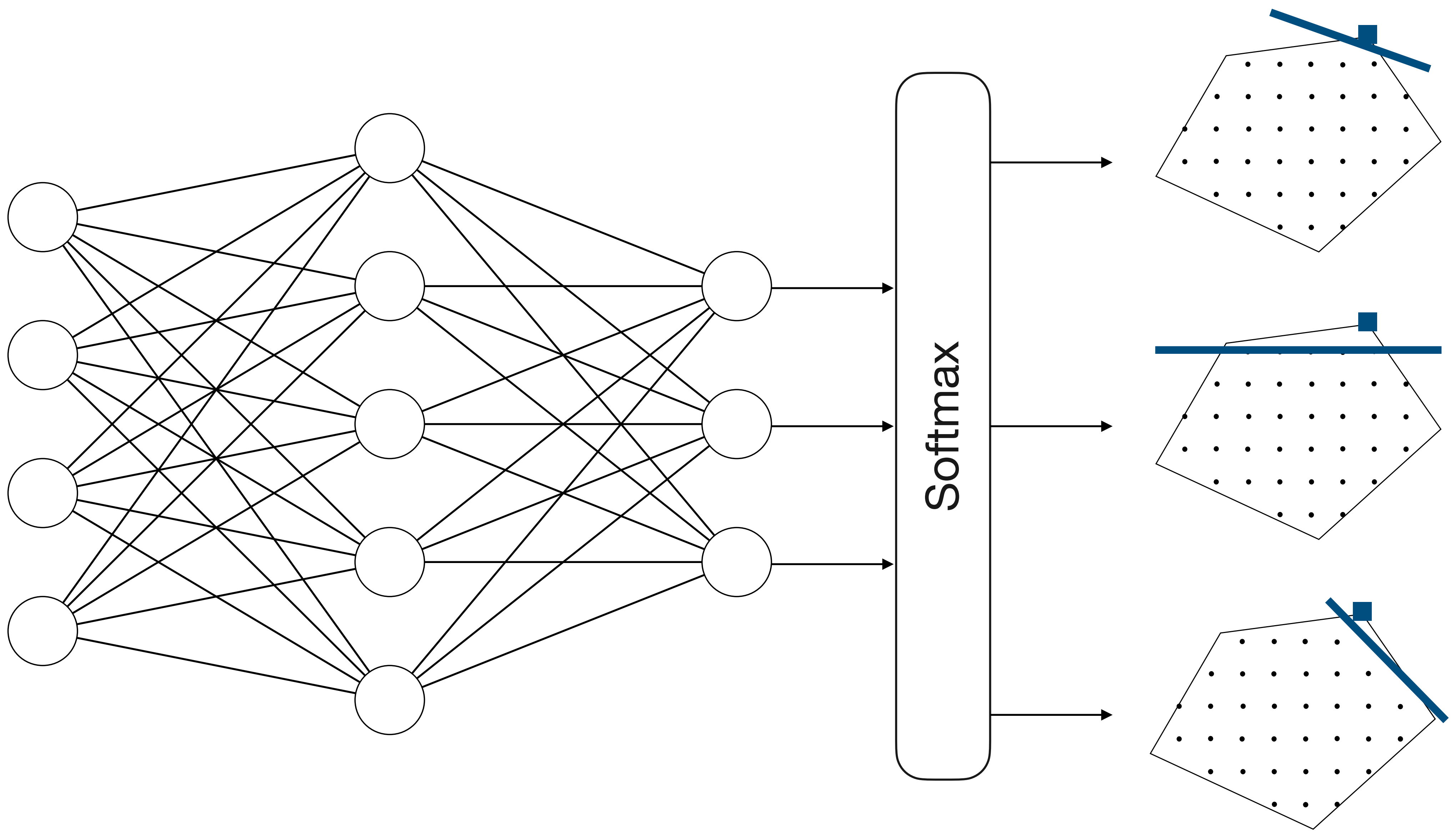
Which variable?

Cornuejols (2012), Contardo, Lodi, Tramontani (2022), Dey and Molinaro (2018)

Learning to cut

Cut selection

Tang, Agrawal, Faenza (2020), Paulus, Zarpellon, Krause, Charlin, Maddison (2022), Deza and Khalil (2023)

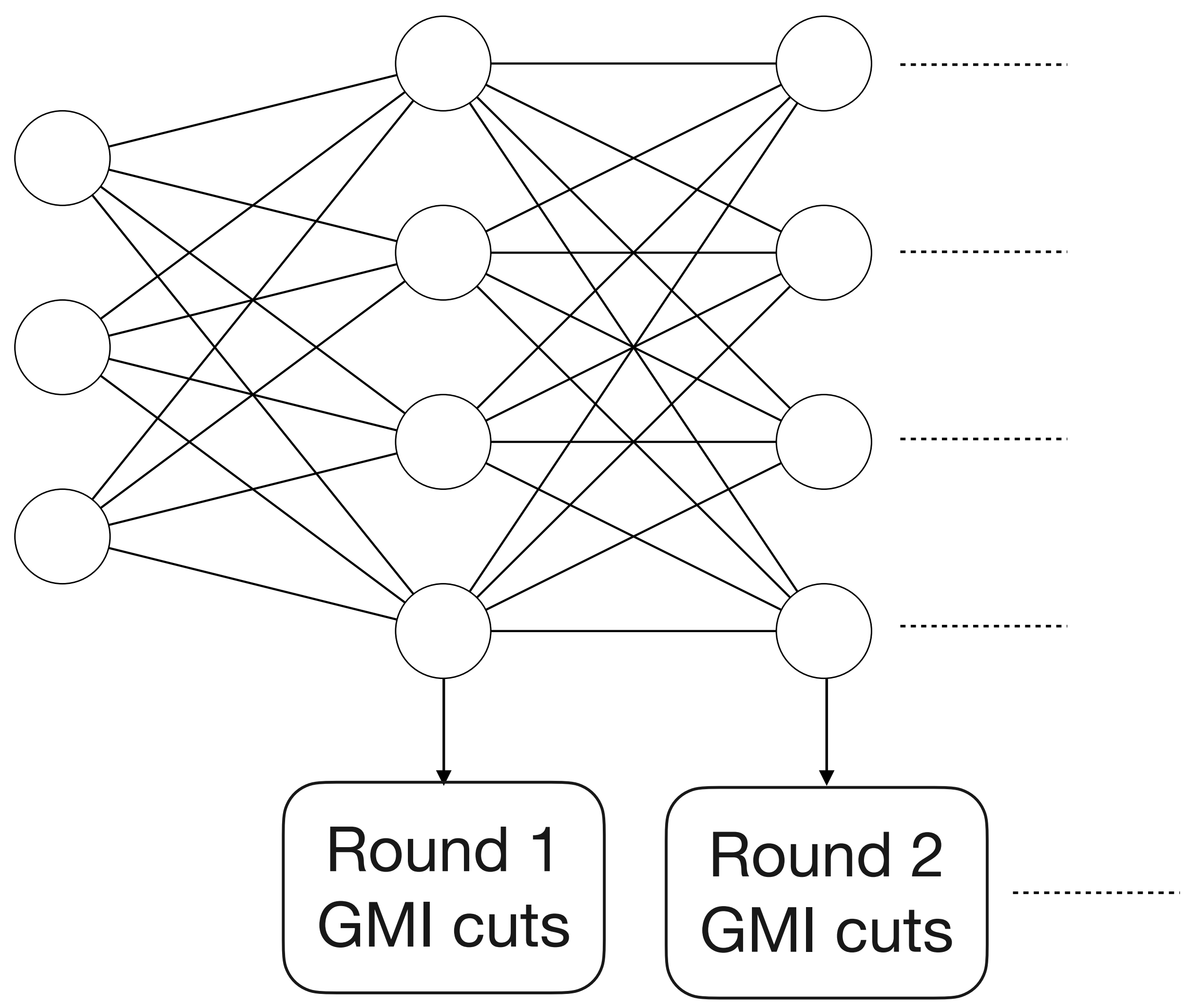


👍 well-defined learning task

👎 less flexibility

Continuous cut generation

Chetelat and Lodi (2023)

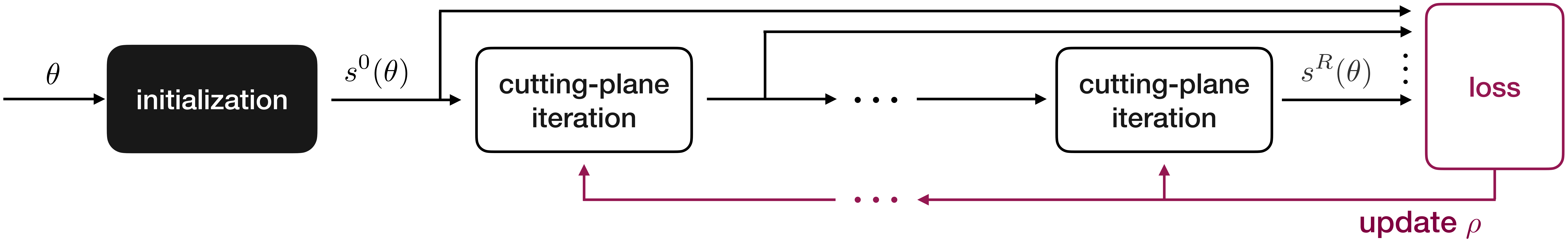


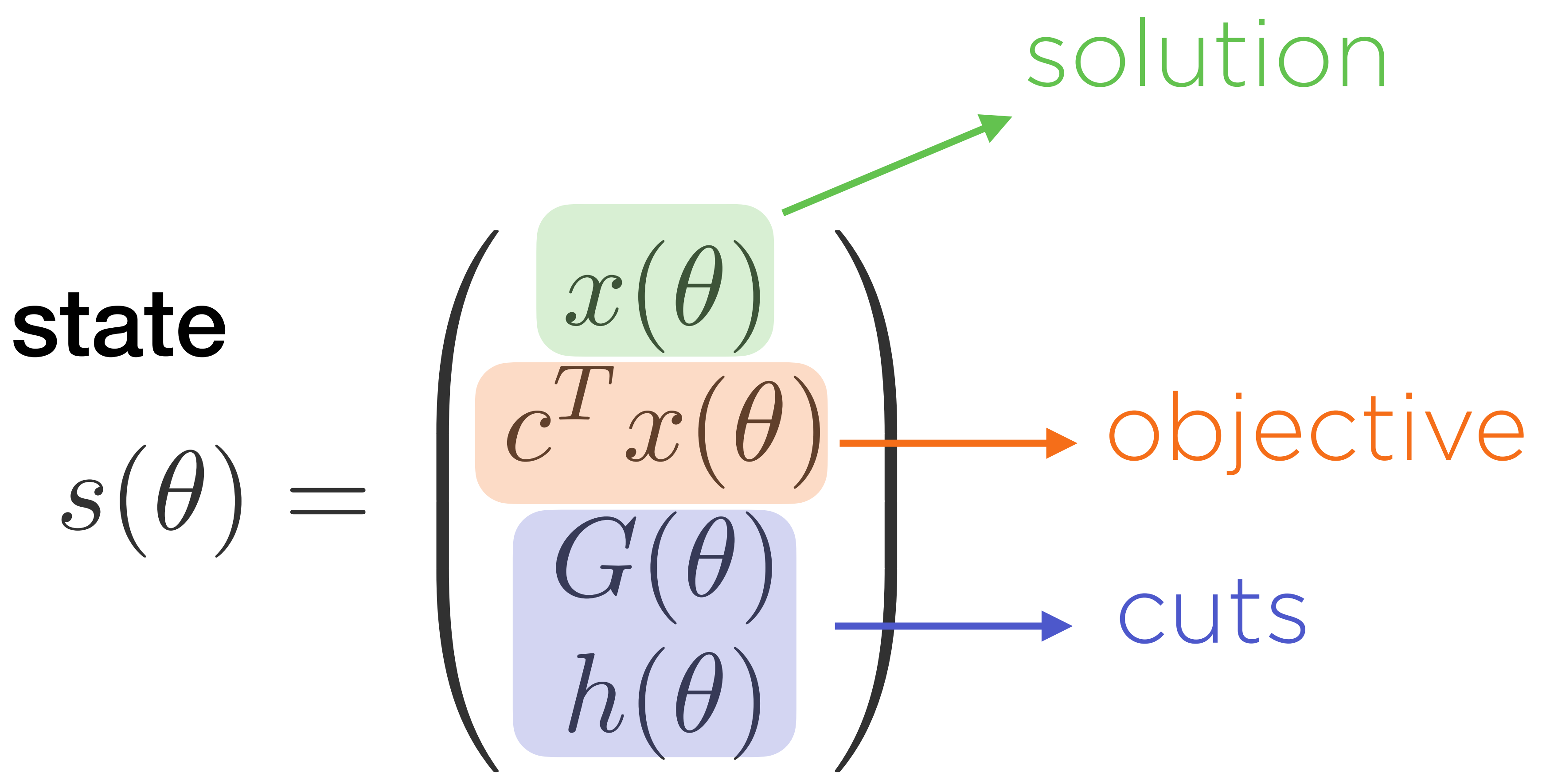
👍 more flexibility

👎 hard learning task

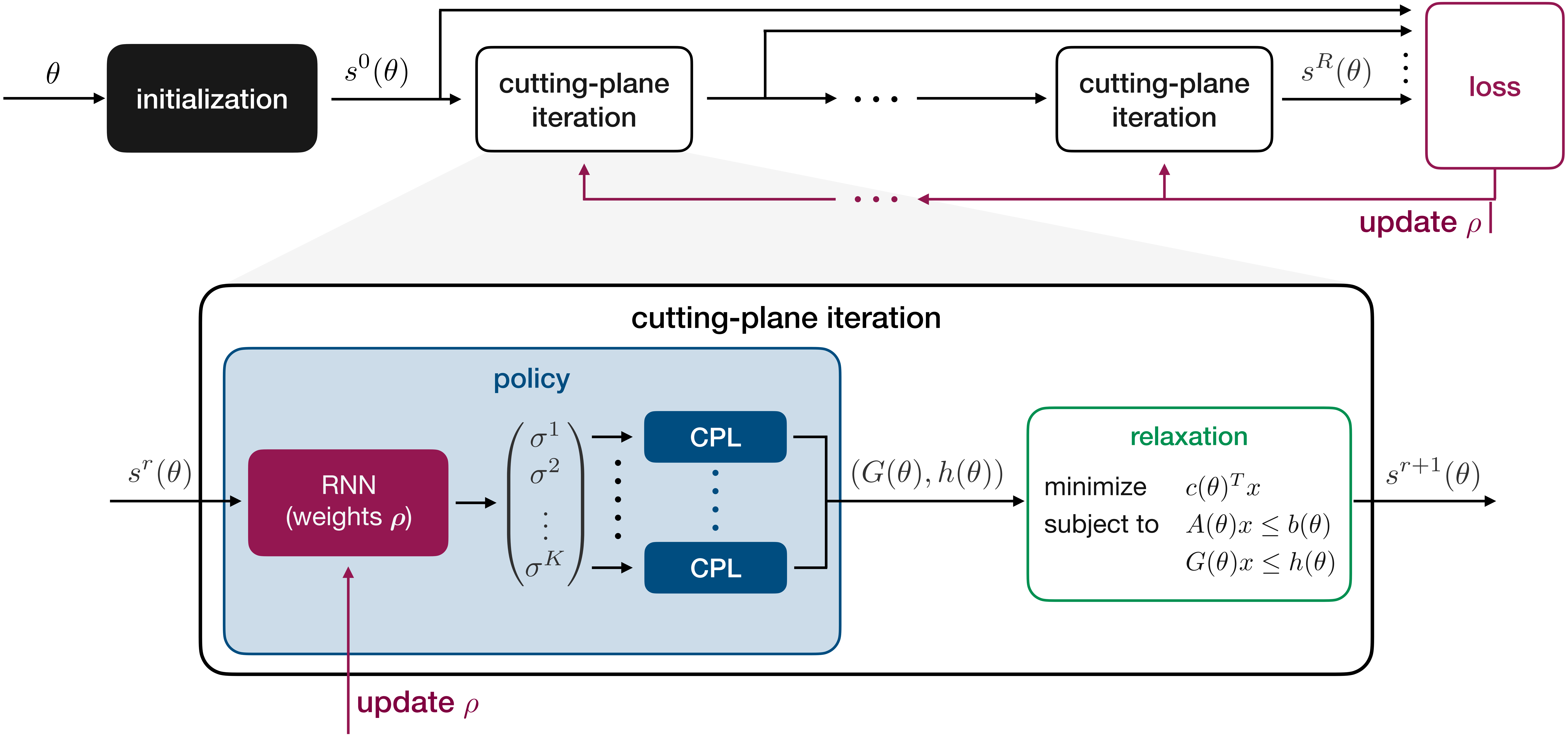
Our differentiable architecture

We deploy R iterations of a cutting plane algorithm

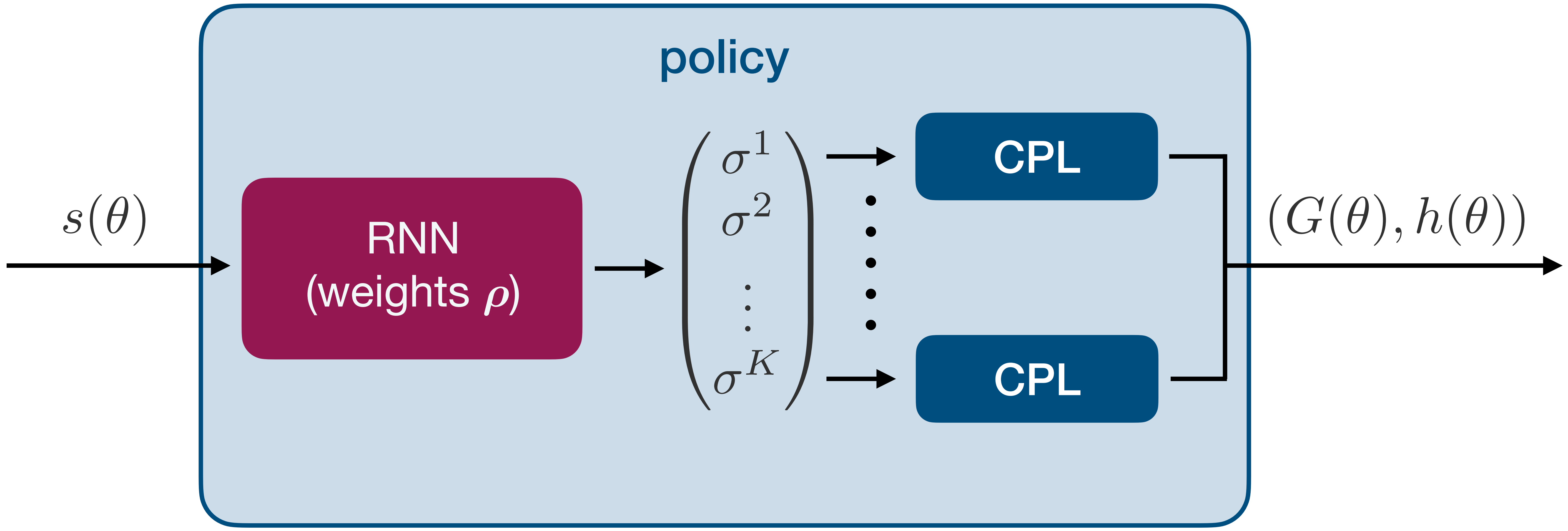




An iteration is policy + relaxation



The RNN provides the “cut parameters”



Cut parameters

$$\sigma = (\pi, \eta, D)$$

disjunction

$\pi^T x \leq \eta$ or $\pi^T x \geq \eta + 1$

normalization

Cutting plane layers (CPLs)

We train with SGD

policy weights

Loss

negative improvement

$$L(\rho) = \mathbf{E}_\theta \left(\sum_{r=0}^{R-1} c(\theta)^T (x^r(\theta) - x^{r+1}(\theta)) \right)$$

Stochastic gradient descent (SGD)

$$\rho^{t+1} = \rho^t - \gamma \nabla \hat{L}(\rho^t)$$

Backpropagate through CPL

Agrawal, Barratt, Boyd, Busseti, Moursi (2019), Agrawal, Amos, Barratt, Boyd, Diamond, and Kolter (2019)

Cuts representability

Disjunctive cuts

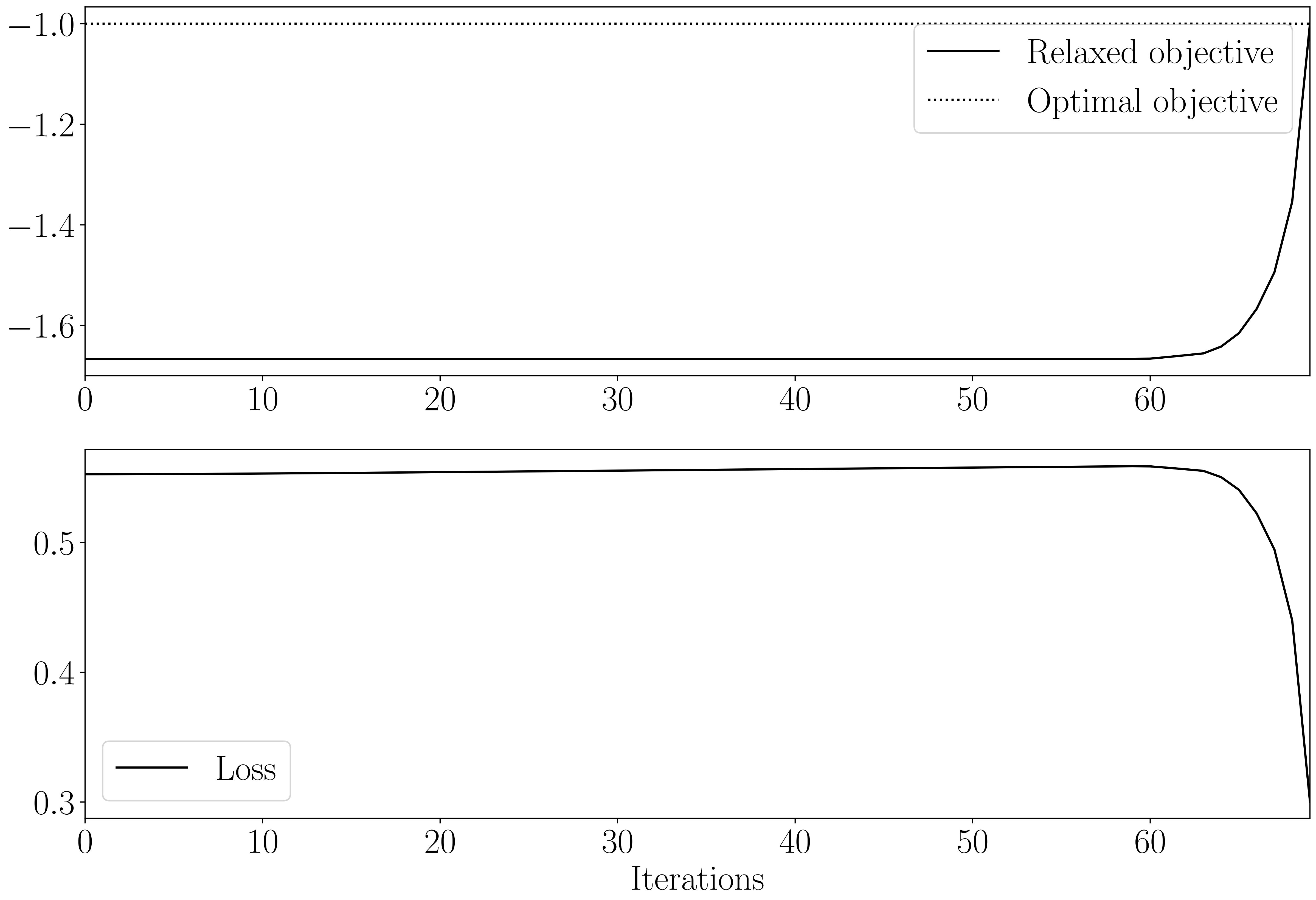
We can represent any disjunctive cuts, even with **multiple-terms disjunctions**

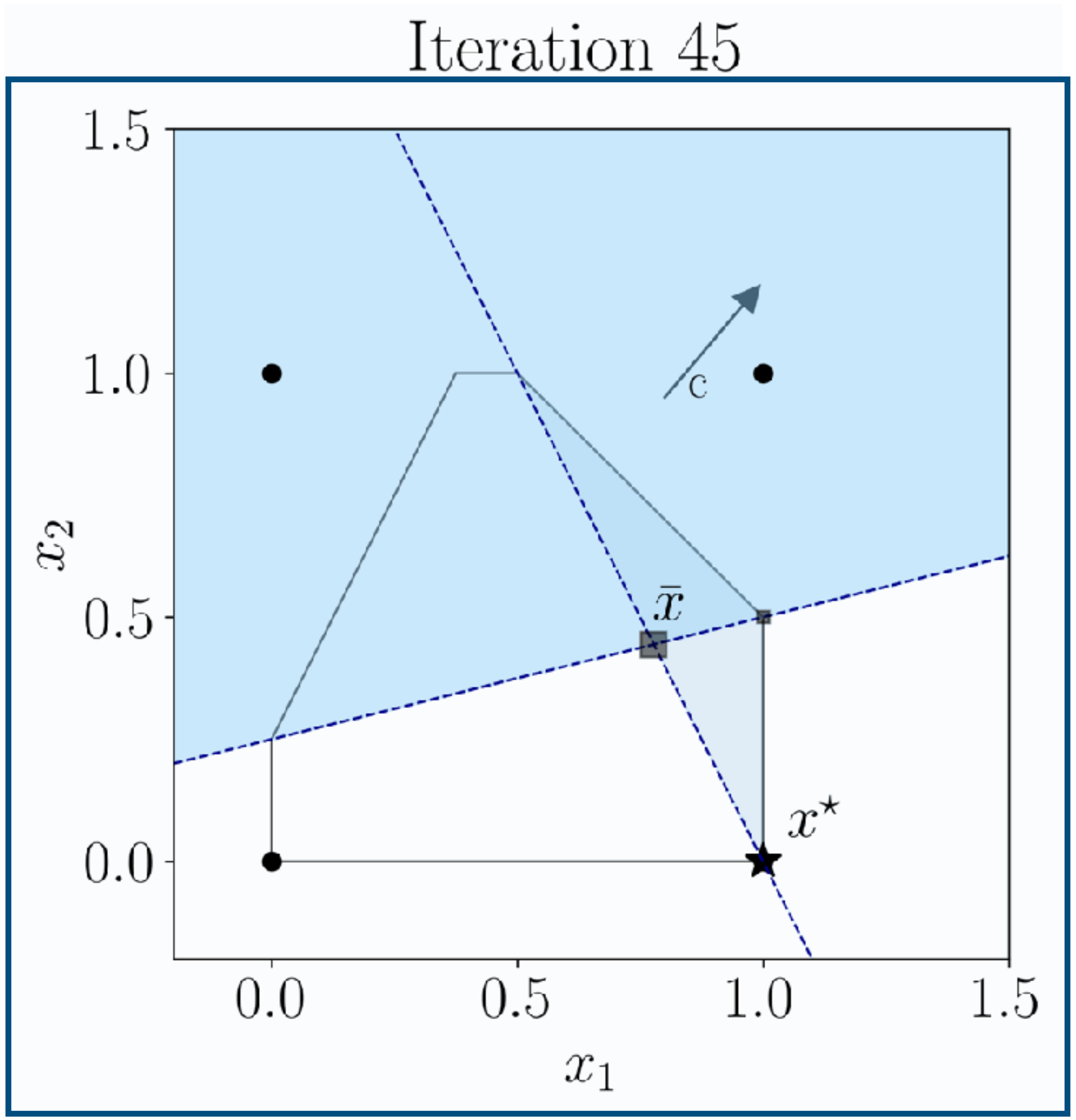
- Gomory Mixed-integer
- Mixed-integer rounding
- Lift-and-project
- Undominated Generalized Gomory

Any undominated subadditive cut from Chetelat and Lodi (2023)

Examples

Fischetti, Lodi and Tramontani infamous example





Learning better cuts

better optimality gap

Violation might be misleading

Matching

(3 rounds, 2 cuts each)

	training			validation			test		
	gap	infeasibility	violation	gap	infeasibility	violation	gap	infeasibility	violation
cplayers	1.09	0.01	0.50	0.70	0.01	0.49	0.00	0.00	0.50
SNC	3.26	0.01	0.05	2.59	0.03	0.49	1.17	0.02	0.50

Hybrid control

(2 rounds, 5 cuts each)

	training			validation			test		
	gap	infeasibility	violation	gap	infeasibility	violation	gap	infeasibility	violation
cplayers	0.38	0.17	0.30	2.28	0.17	0.25	5.47	0.14	0.13
SNC	18.10	0.05	0.16	21.60	0.05	0.19	21.65	0.05	0.08

Summing up

Differentiable

Mixed-integer optimization algorithms are **differentiable**

We can learn the underlying properties of **parametric families** of problems

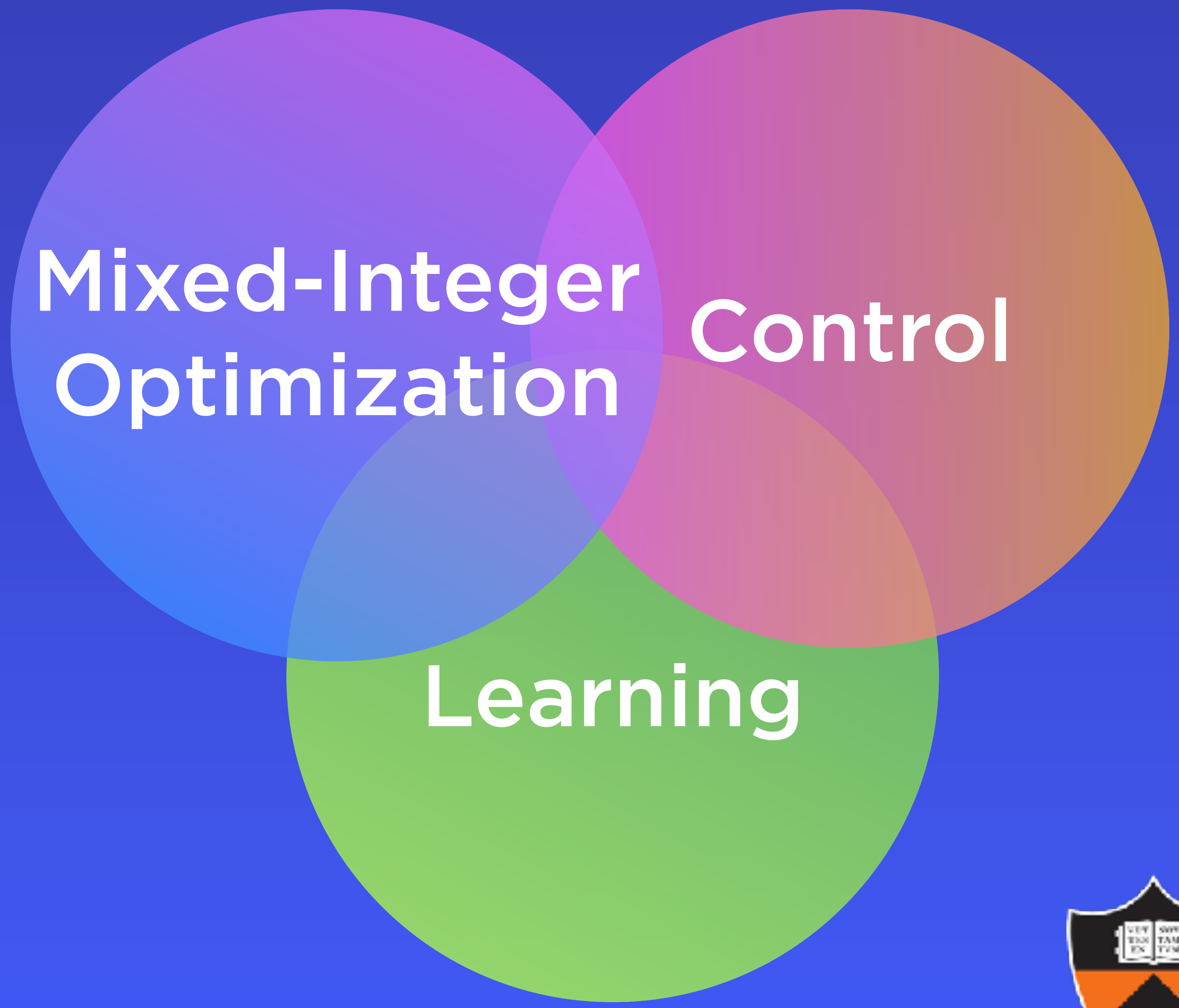
Learning

Optimization

We can build **efficient** algorithms to solve the parametric families

Differentiable Cutting Plane Layers for Mixed-Integer Optimization

arXiv 2311.03350



DataX
Accelerating Scientific Discovery at Princeton

PRINCETON UNIVERSITY