Learning to select cutting planes in MILP

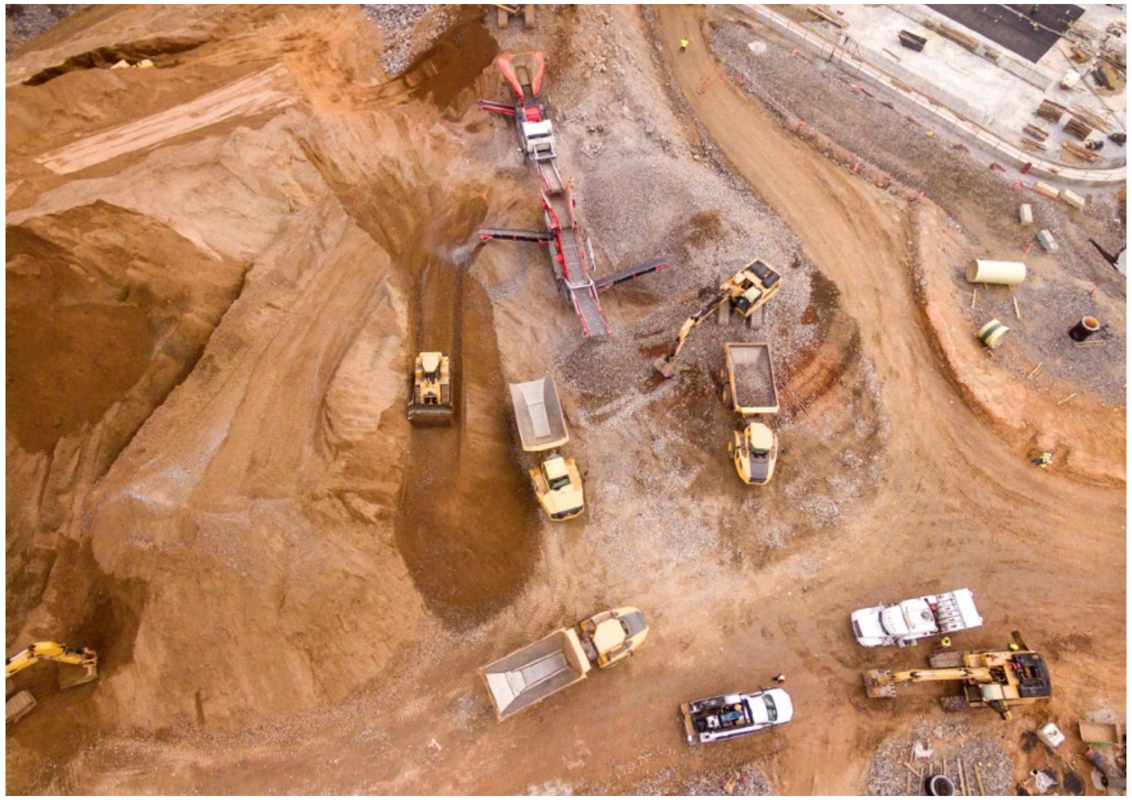
Antoine Prouvost, Aleksandr M. Kazachkov, Andrea Lodi

INFORMS Seattle
22nd of October 2019

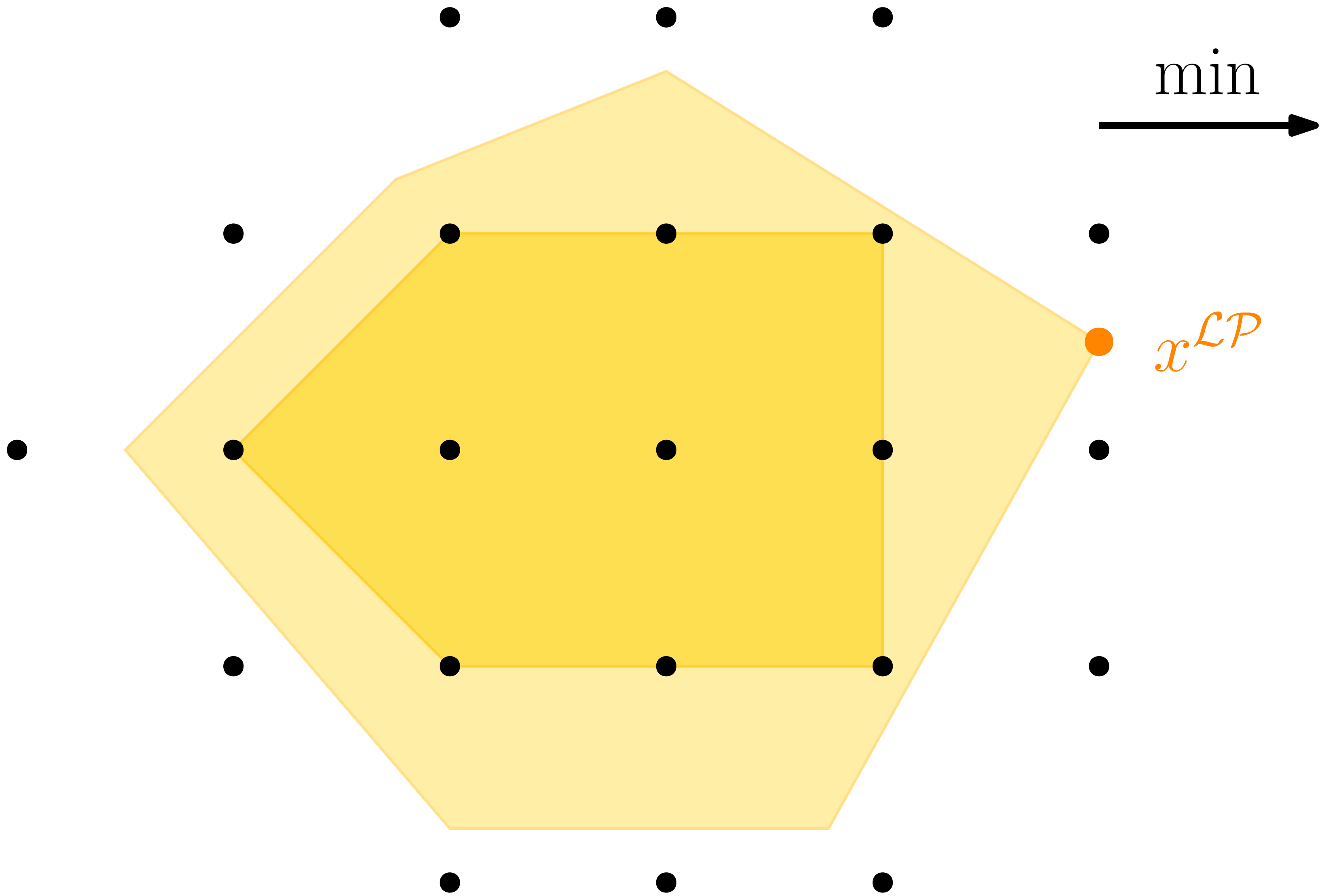
Integer Programming

$$\min c^\intercal x$$

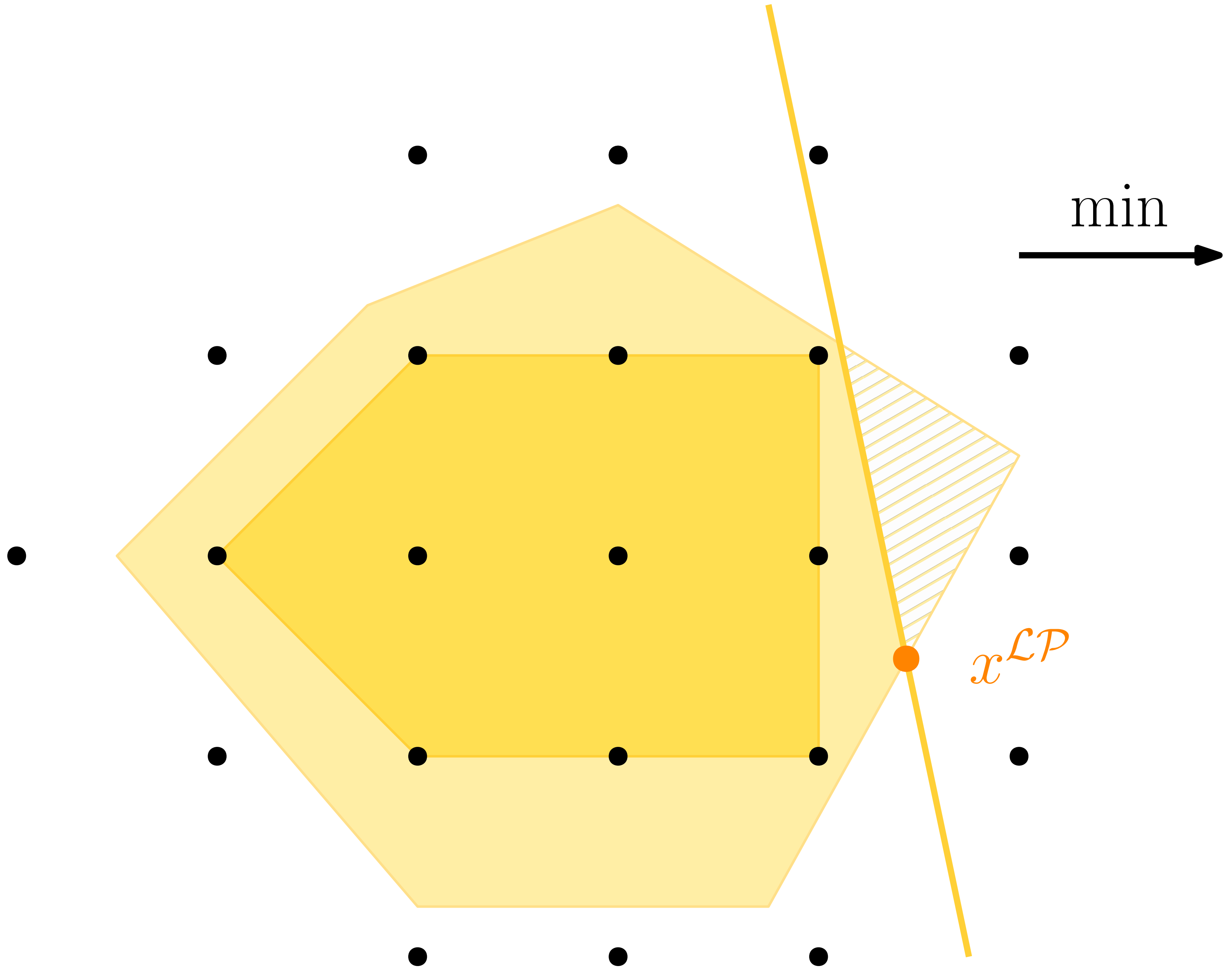
$$\begin{aligned} Ax &\geq b \\ x &\geq 0 \\ x &\in \mathbb{Z} \end{aligned}$$



Cutting Planes



Cutting Planes



Motivation

- Cutting plane can be highly effective in solving MILP
  [*Bixby and Rothberg 2007*]

- Yet, in large quantities they burden the LP relaxation

- They interfere with each other and with BnB

- No satisfying heuristic to evaluate their individual contributions
  How do we select which ones to keep?

The structure hypothesis

- We do not care about most instances that coud exist;
- Instead, we look at problem instances as data points from a specific, unknown, probability distribution;
- “Similar” instances show “similar” solving procedures.

Learning Formulation

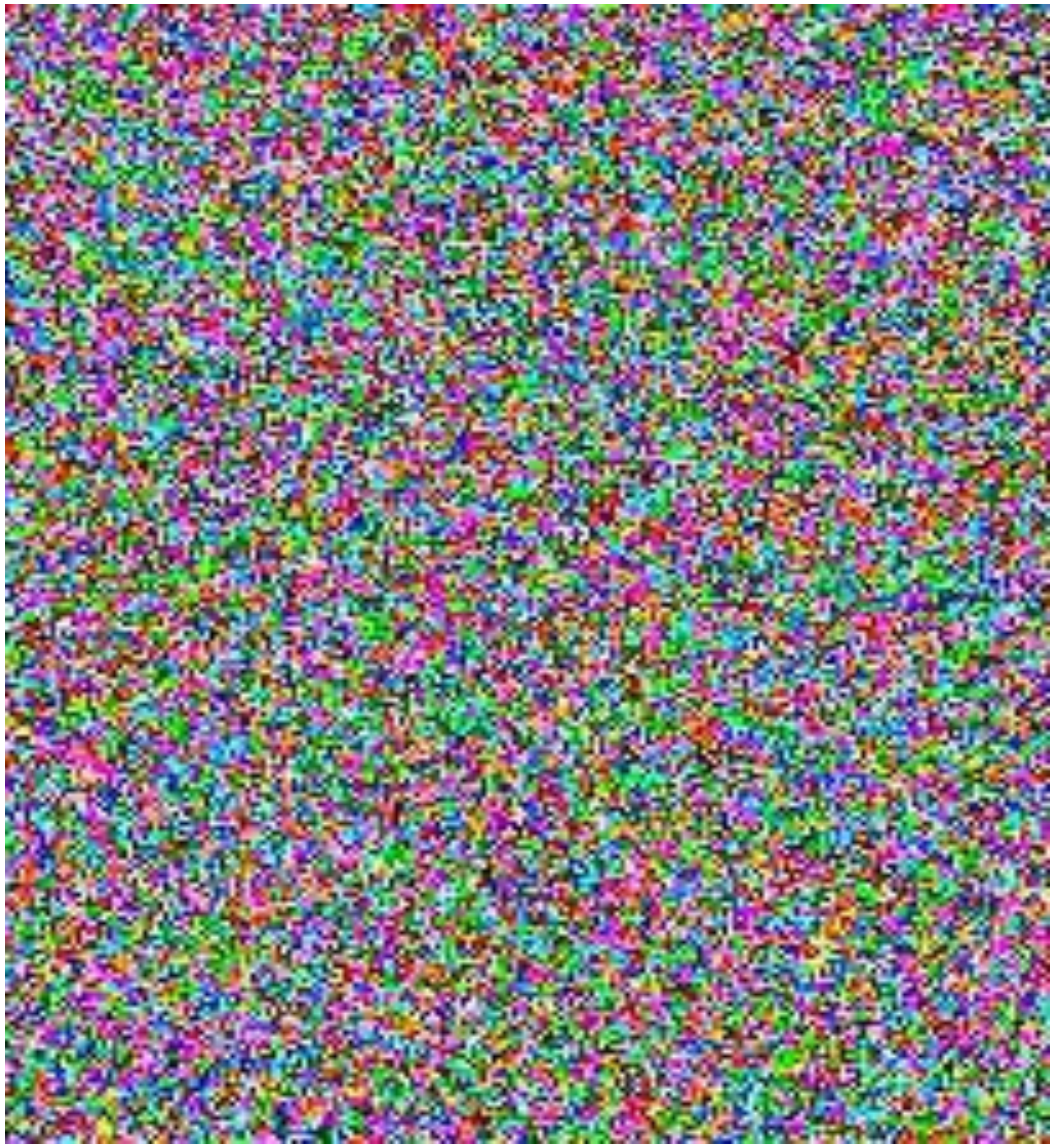
Given

- A distribution of instances $i \sim P$
- A performance measure m
- A set of algorithms parametrized by a policy $\{a(\pi_\theta) \,|\, \theta \in \mathbb{R}^n\}$

Find the policy that minimize

$$\min_{\theta \in \mathbb{R}^p} \mathbb{E}_{i \sim P}\, m(i, a(\pi_\theta))$$

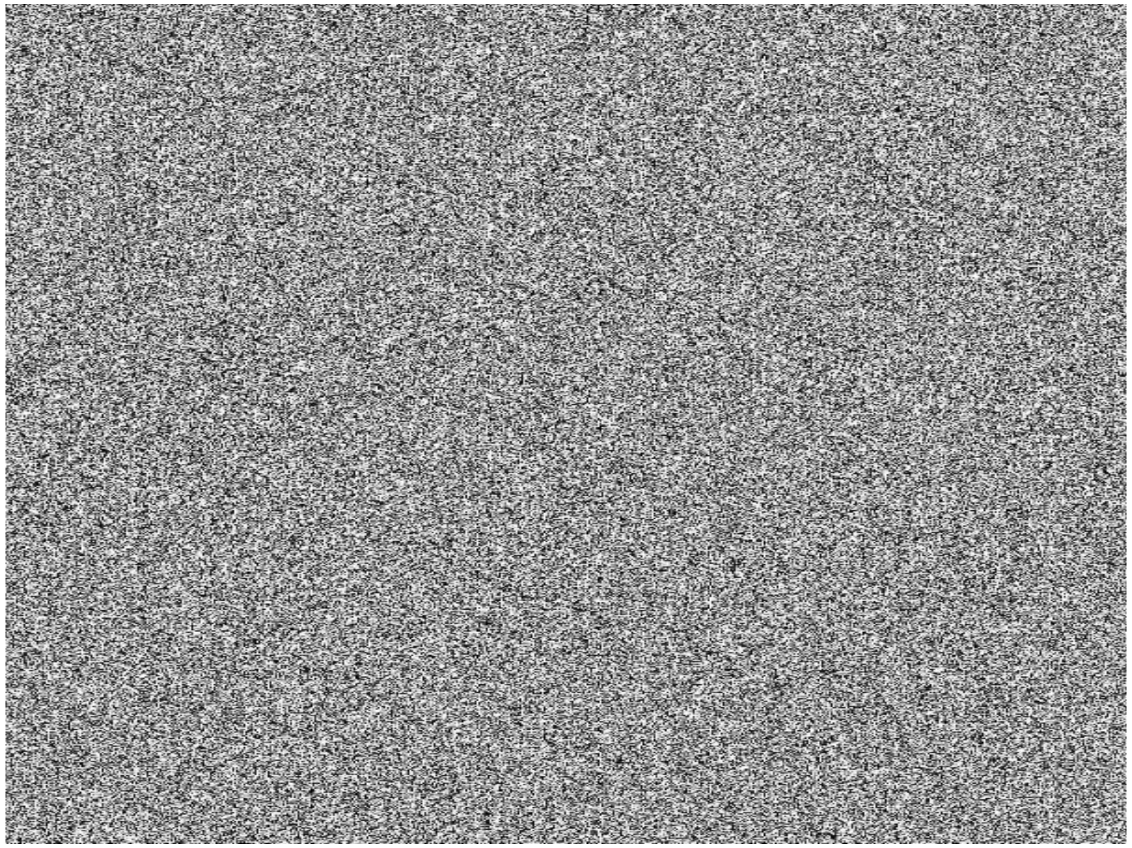
Random Images



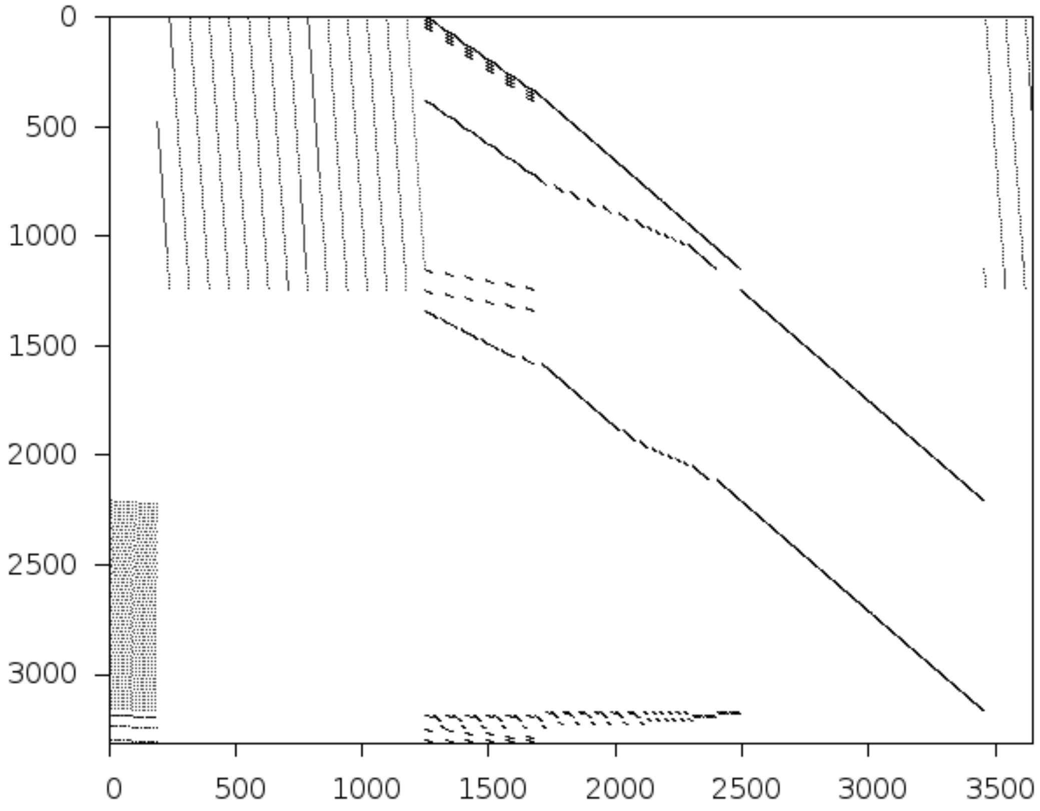
Random iid pixels

Random face (GAN)
thispersondoesnotexist.com

Random Instances

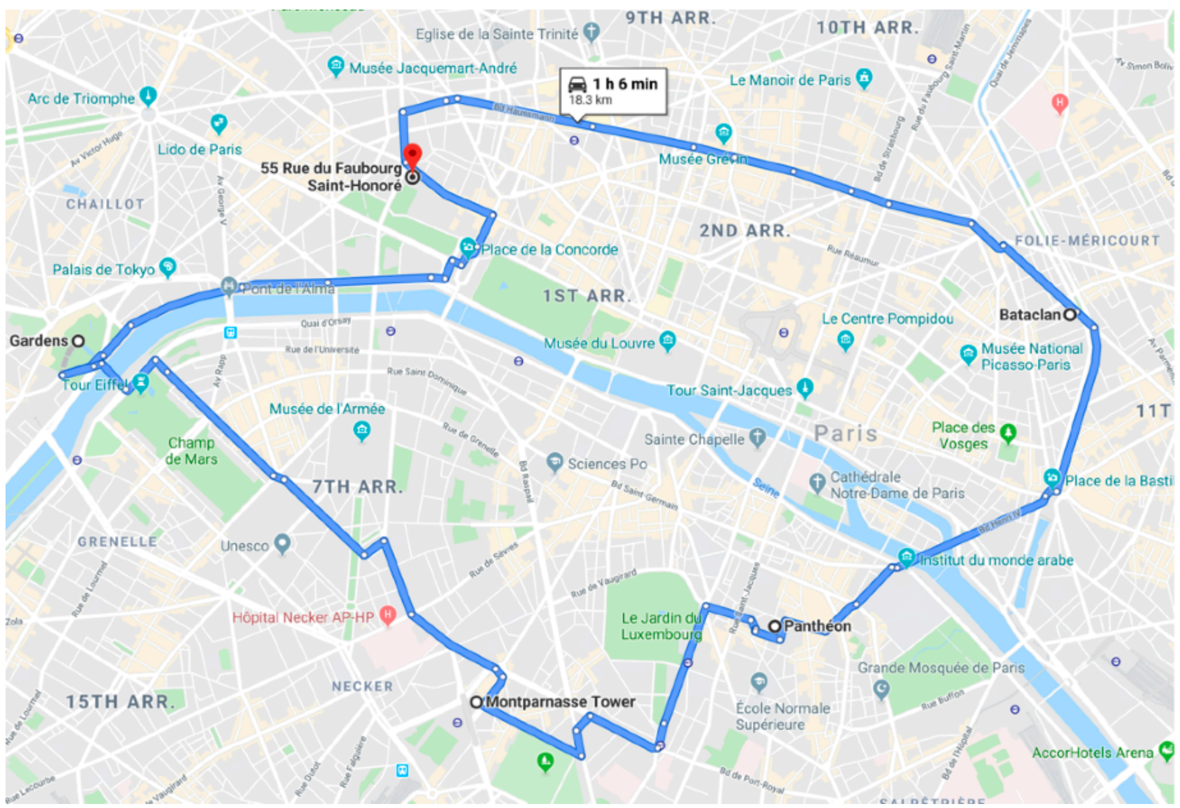


Random iid coefficients

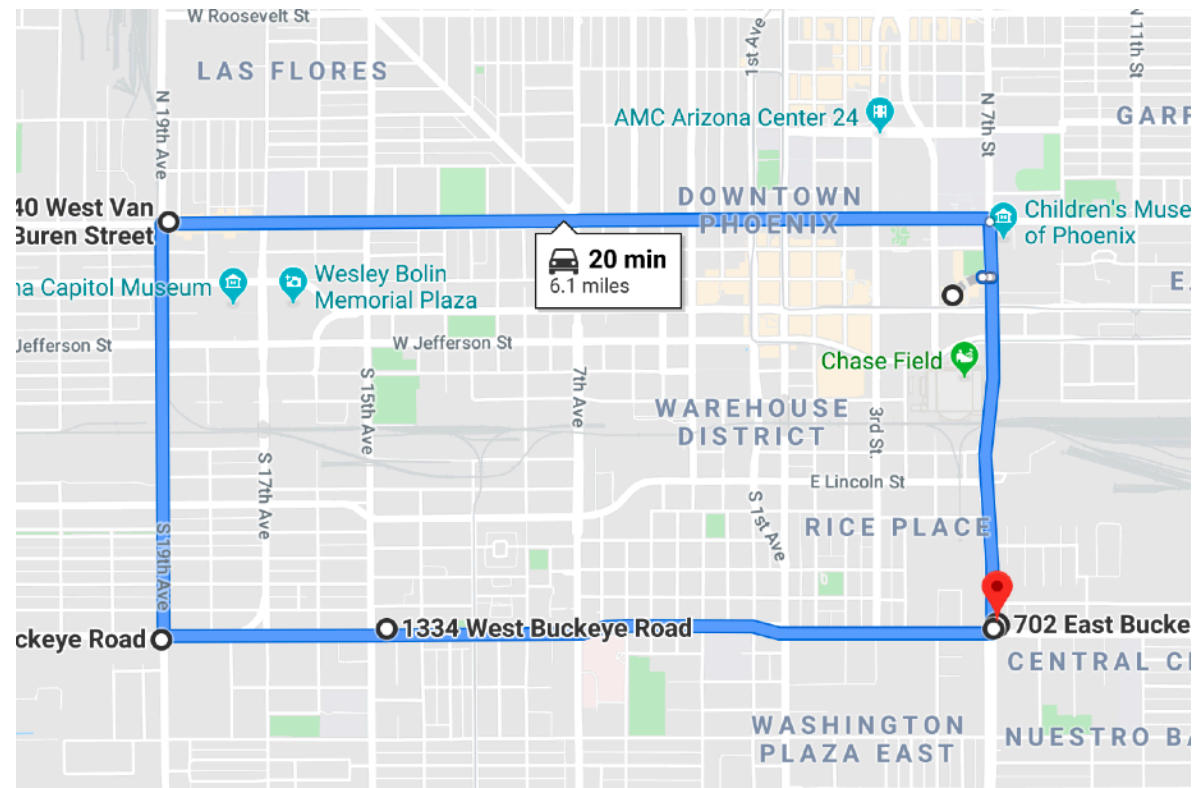


a1c1s1 from MipLib 2017

Example

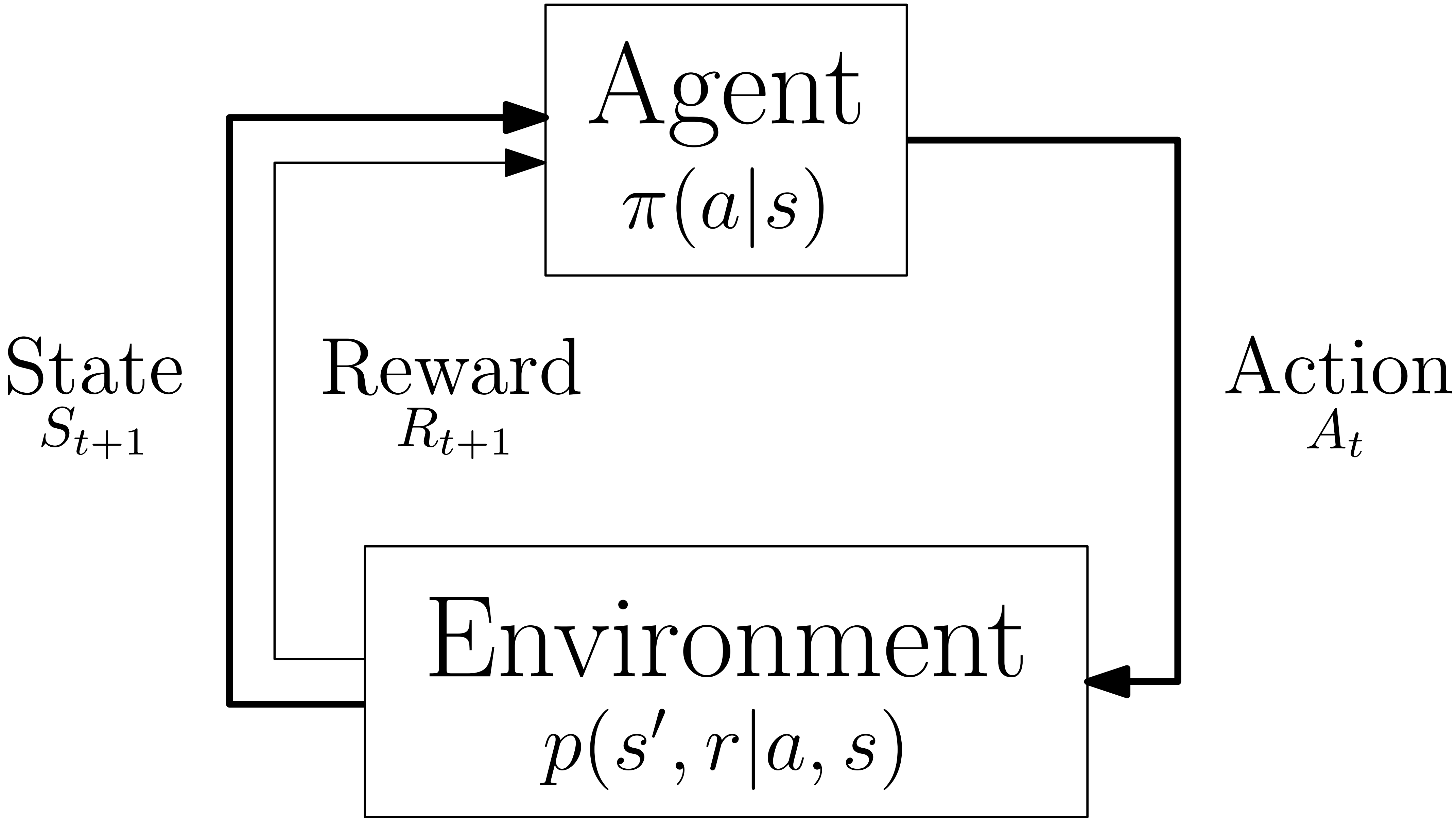


Paris

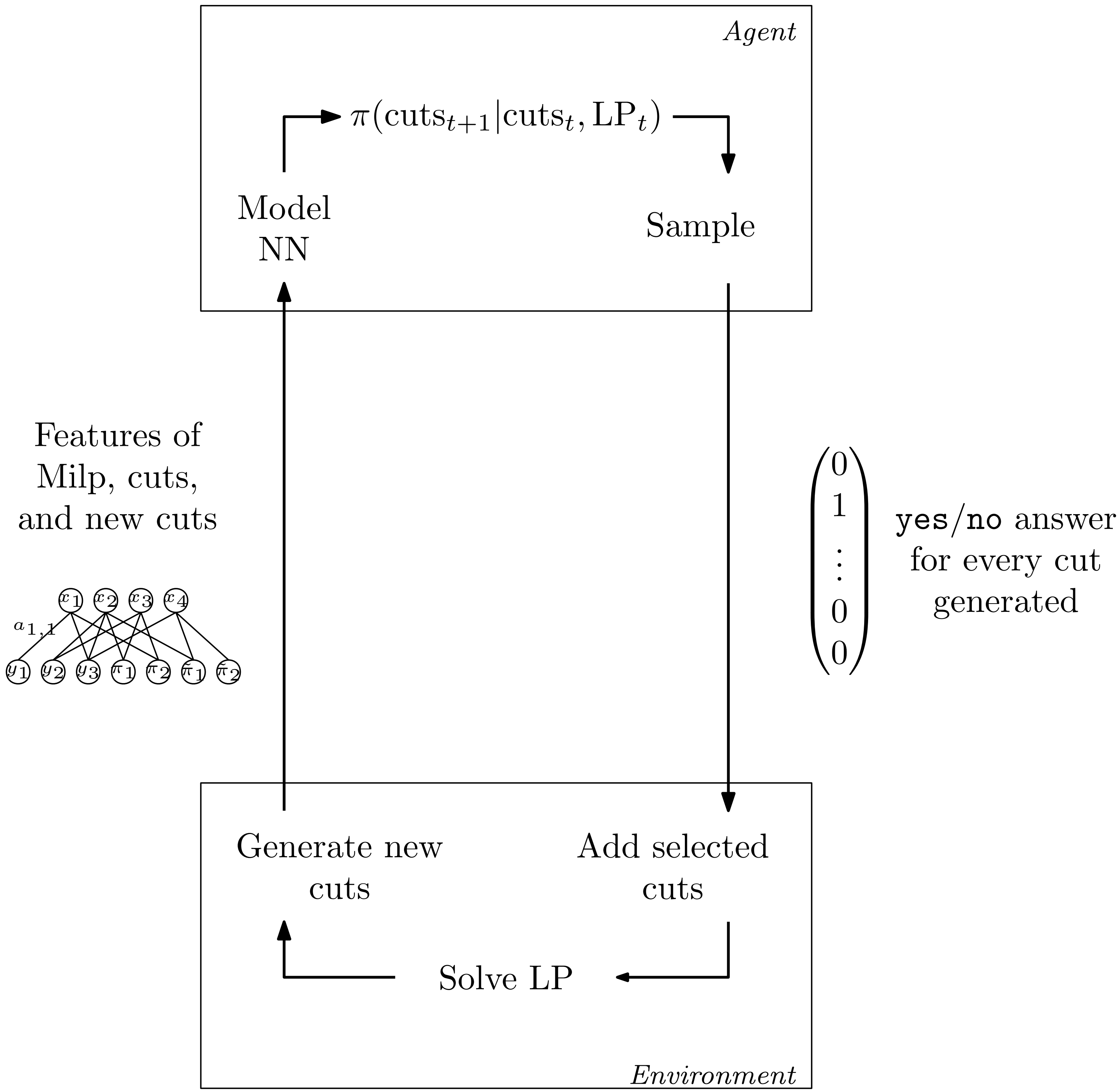


Phoenix

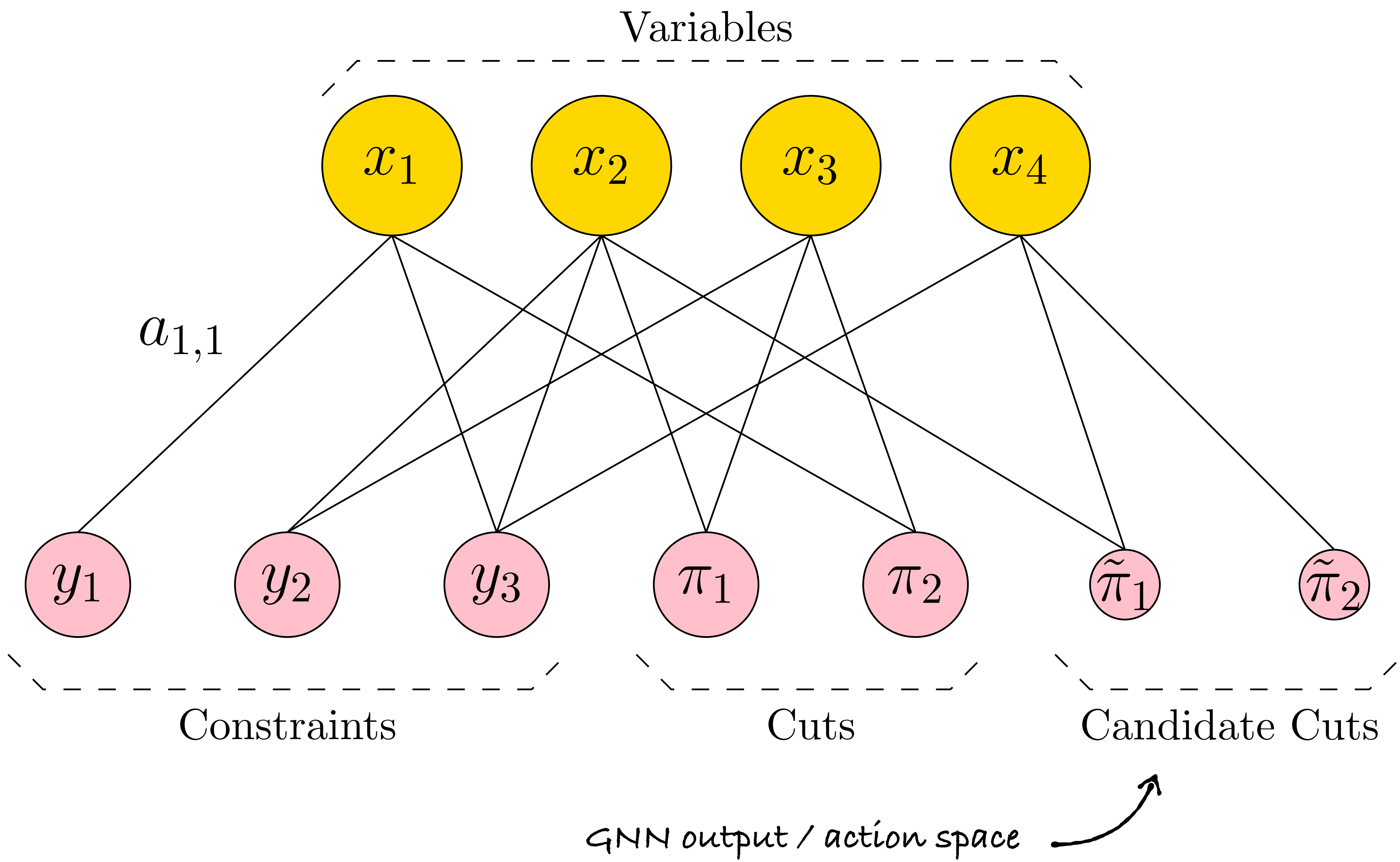
MDP Formulation



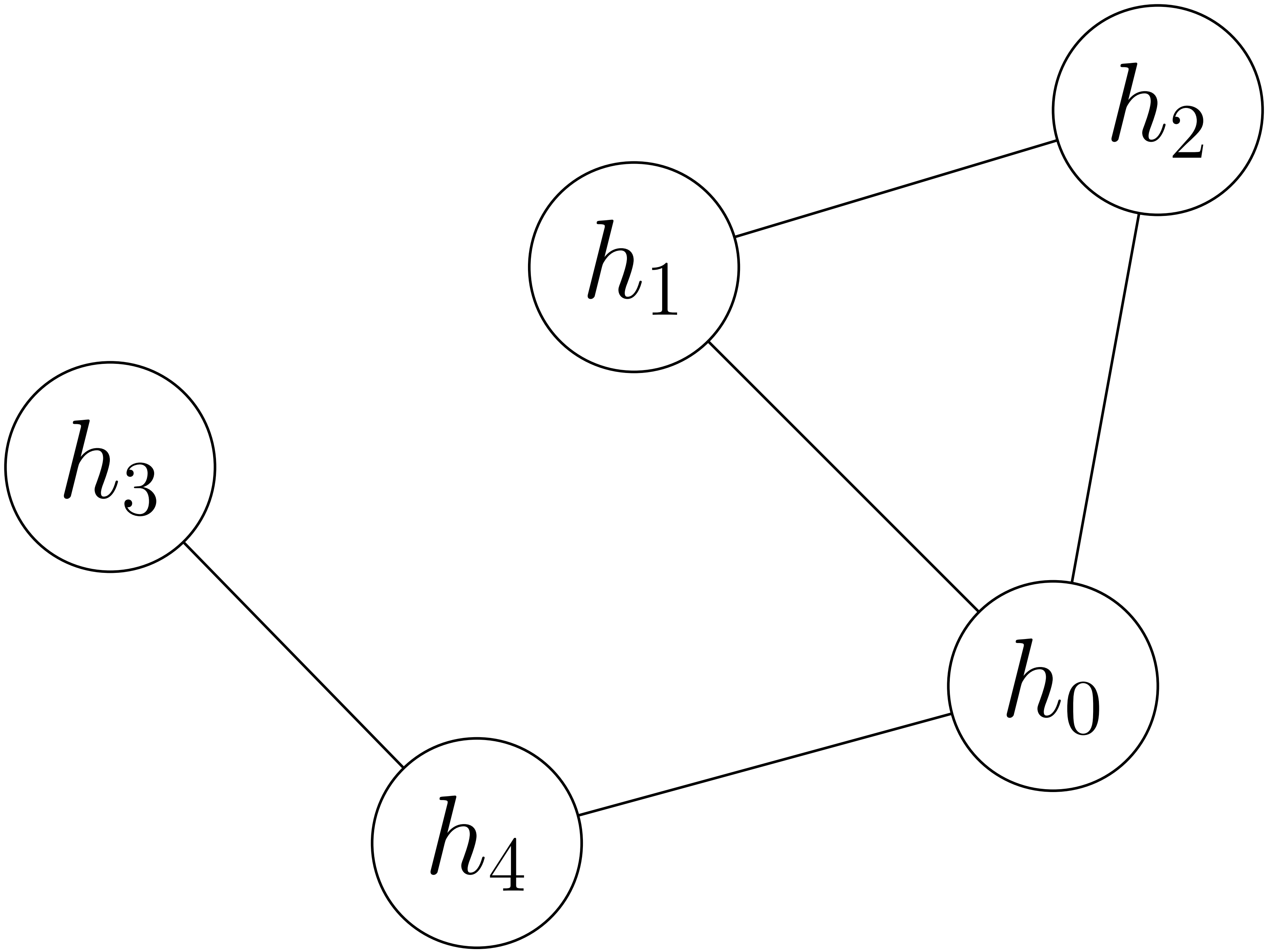
MDP Formulation



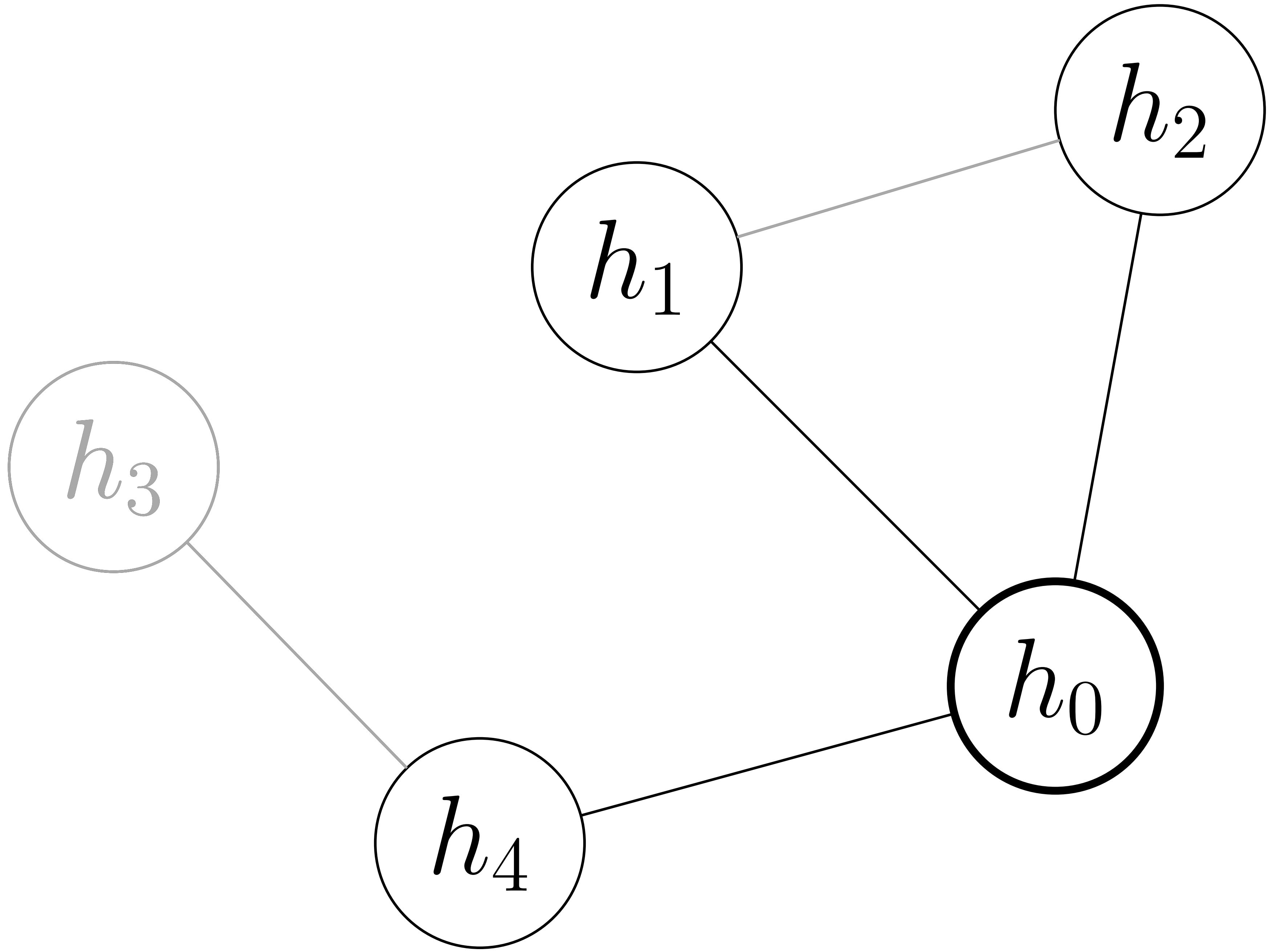
Observation



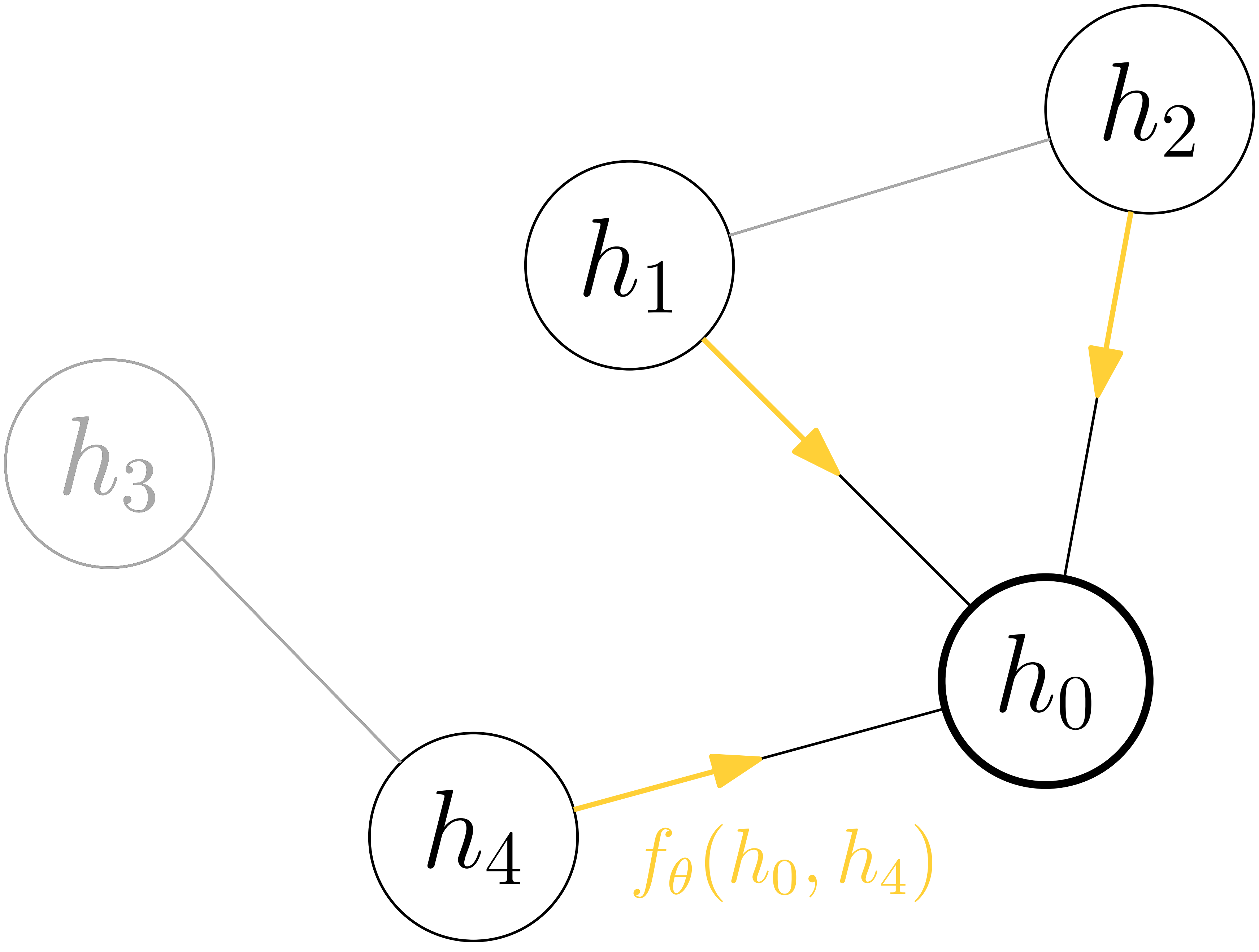
Graph Neural Network



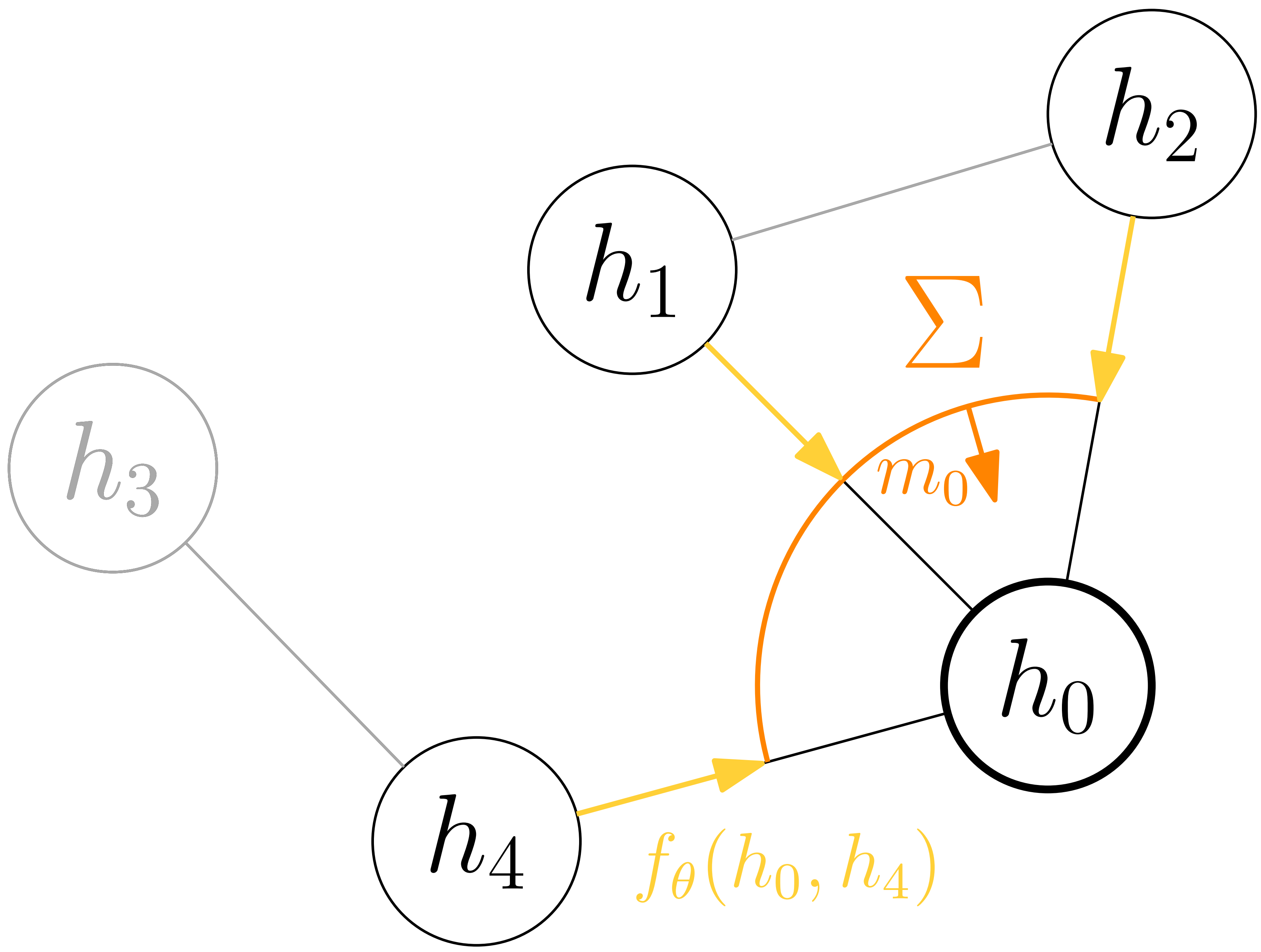
Graph Neural Network



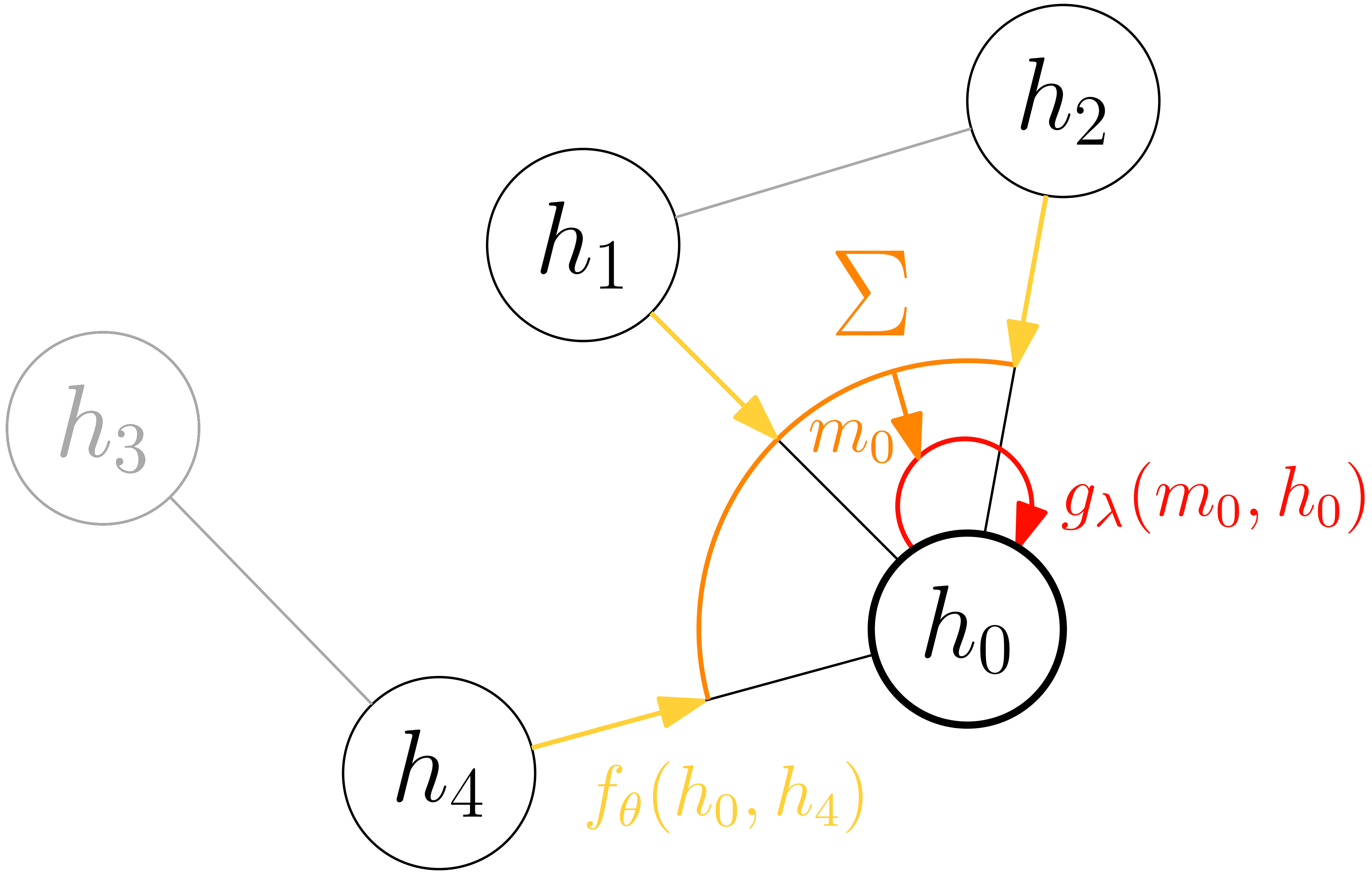
Graph Neural Network



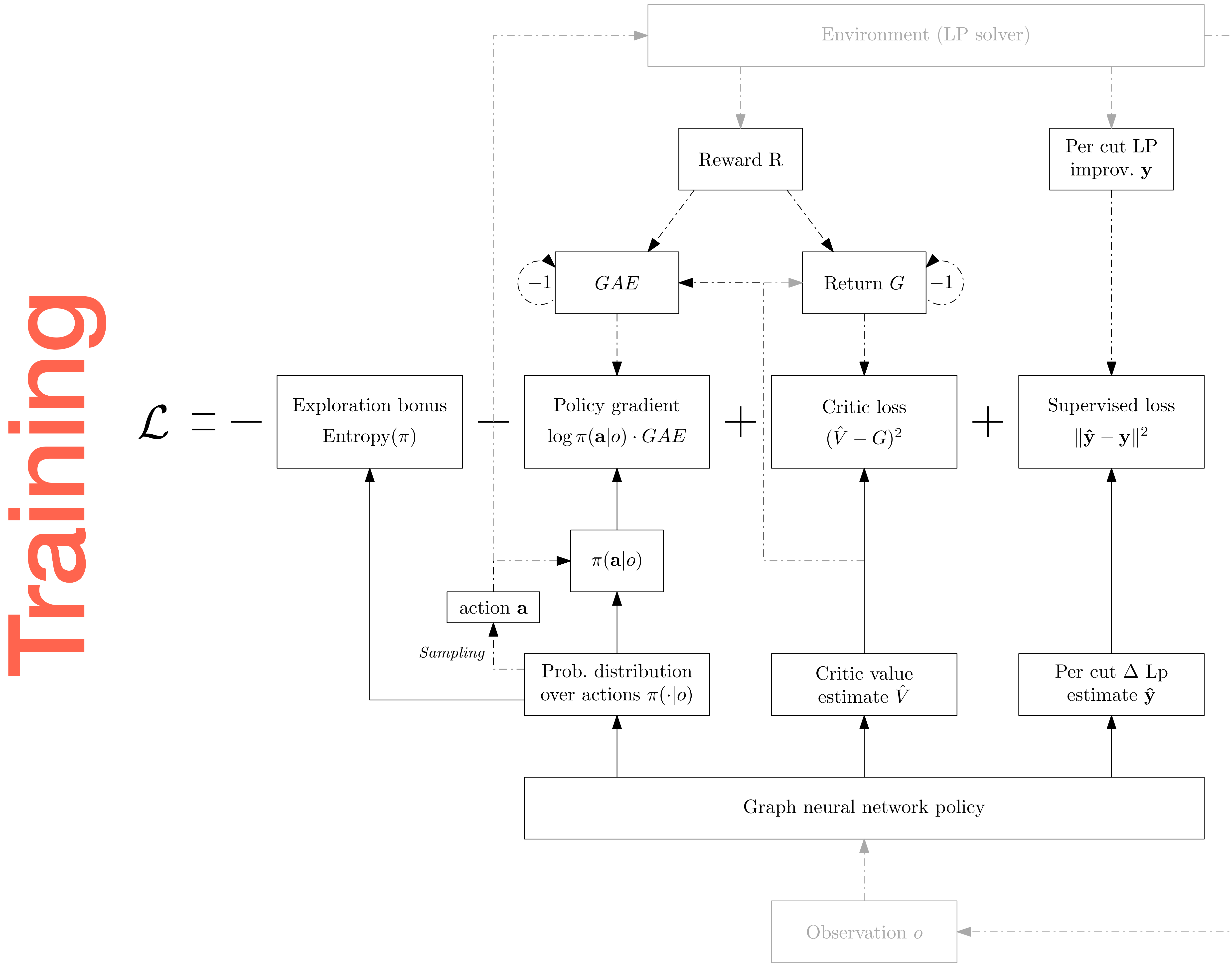
Graph Neural Network



Graph Neural Network



Formulation from [*Schoenholz et al. 2017*]



Training
Environment (LP solver)
Reward R
Per cut LP improv. $\mathbf{y}$
-1
GAE
Return G
-1
$\mathcal{L} = -$
Exploration bonus
Entropy(π)
$-$
Policy gradient
$\log \pi(\mathbf{a}|o) \cdot GAE$
$+$
Critic loss
$(\hat{V} - G)^2$
$+$
Supervised loss
$\|\hat{\mathbf{y}} - \mathbf{y}\|^2$
$\pi(\mathbf{a}|o)$
action $\mathbf{a}$
Sampling
Prob. distribution over actions $\pi(\cdot|o)$
Critic value estimate $\hat{V}$
Per cut Δ Lp estimate $\hat{\mathbf{y}}$
Graph neural network policy
Observation o

Case Study

- 2-matchings → blossom inequalities
  [Edmonds 1965][Chandrasekaran et al. 2016]
- Using a superset: 0-½ cuts and GMI cuts
- Terminate only when integer solution
- Agent select one cut per round
- Minimize #cuts (reward -1 per cut)

Case Study

Two training runs over gr48, one cuts at the time.

Case Study

- Agent is able to improve over random
- Unstable across different hyper-parameters
- No generalization study

Highly Challenging

- Cut selection is **hard**
- Reinforcement Learning is **erratic**
- Graph Neural Network are **unstable**
- **No existing** benchmarks
- Software engineering efforts are **considerable**

[*Tang et al. Reinforcement Learning for Integer Programming:Learning to Cut*]

Questions ?

antoine.prouvost@polymtl.ca