Graph Neural Networks for Power Systems

Balthazar Donon - 23/03/2023

Balthazar DONON

Deep Statistical Solvers & Power Systems Applications

- 2018 - 2022 : PhD student at RTE Research & Development and Université Paris-Saclay

Graph Neural Networks for Voltage Management

- 2022-2024 : Post Doctoral Researcher at Institut Montefiore in a project for RTE Research & Development

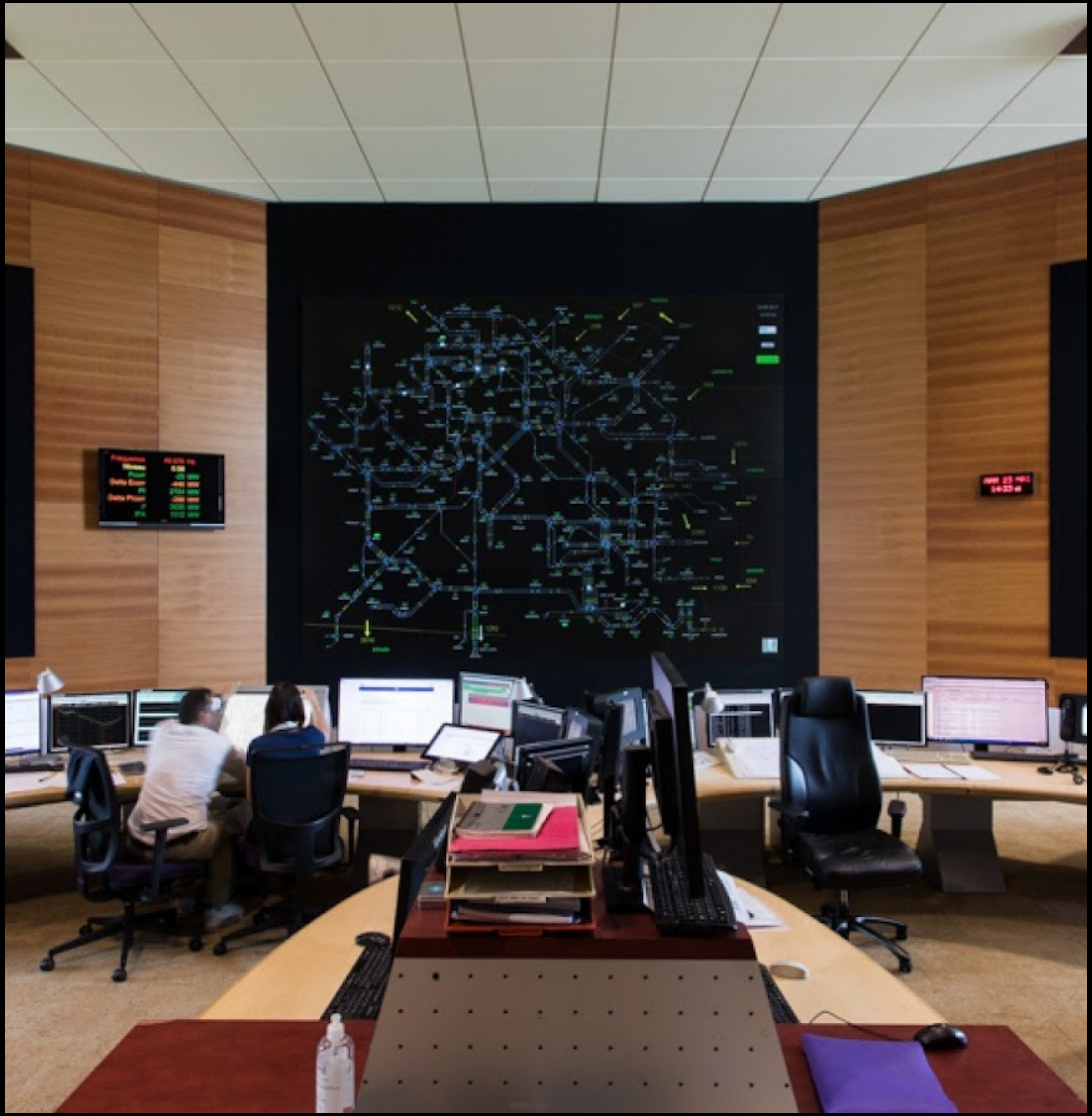
Deep Statistical Solvers

& Power Systems Applications

Introduction

Introduction
Dispatchers

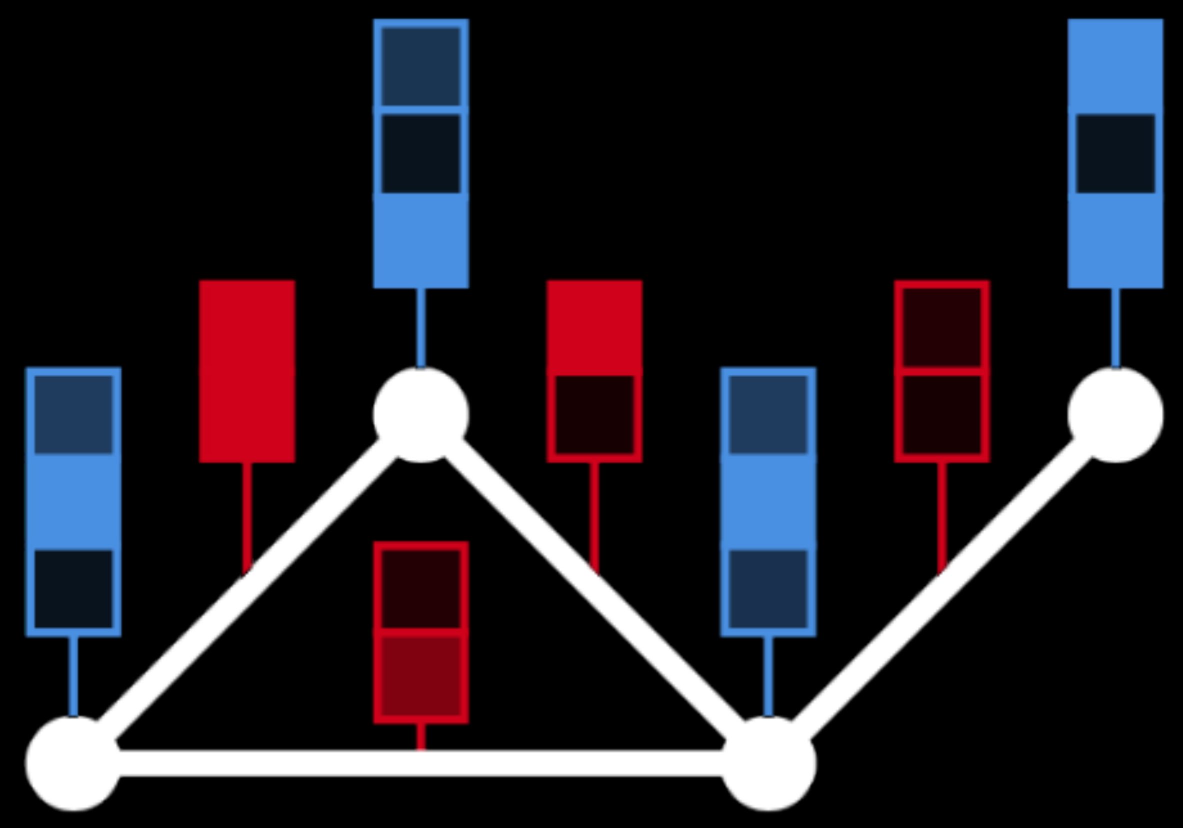
- Monitor and control transmission grids in real-time.
- Run security analyses.
- Anticipate the near future.



Introduction

AC Power Flow Problem

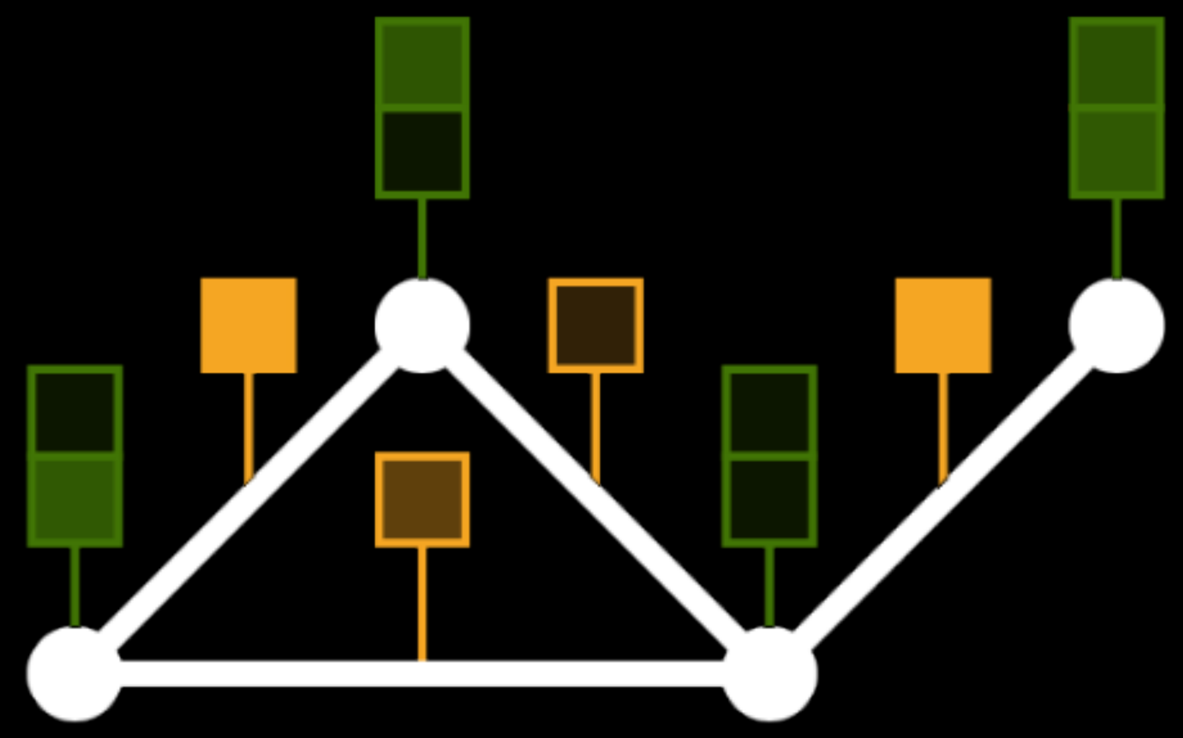
Input (x)



Newton-Raphson

→

Output (y)



- production & consumption
- line characteristics

- voltage
- power flow

Introduction

An Increasing Complexity

- Power systems are facing an increasing uncertainty & complexity :
  - Intermittent renewables
  - New market mechanisms
  - New electricity uses

Introduction
Issue

Newton-Raphson too slow for probabilistic approaches (Monte-Carlo)...

➔ Replace it with a fast approximation based on AI?

Introduction
The AI Opportunity

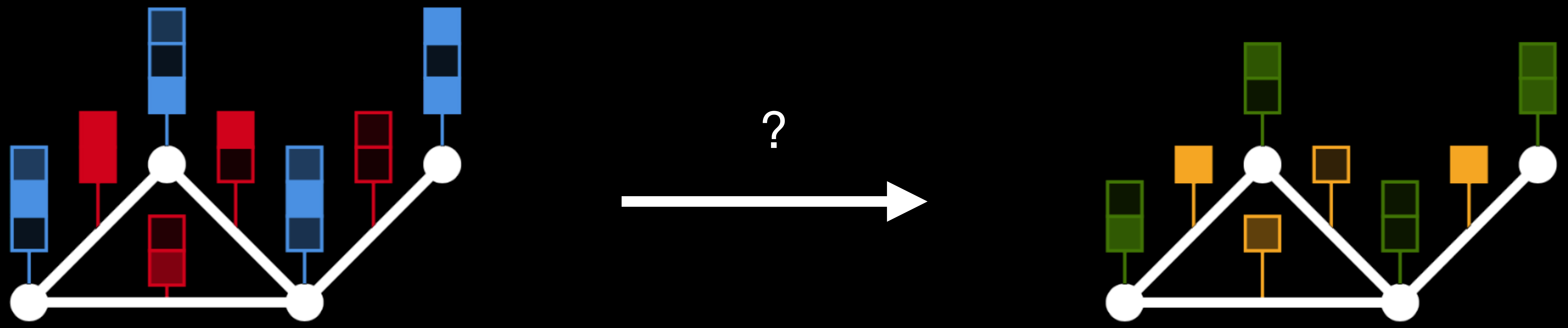
- AI / DL achieved tremendous successes in various domains :
  - Computer Vision
  - Natural language processing
  - Games (chess, go, etc.)
- Ability to solve complex problems that require a very high level of abstraction.

Introduction

The AC Power Flow Problem

Input (x)

Output (y)

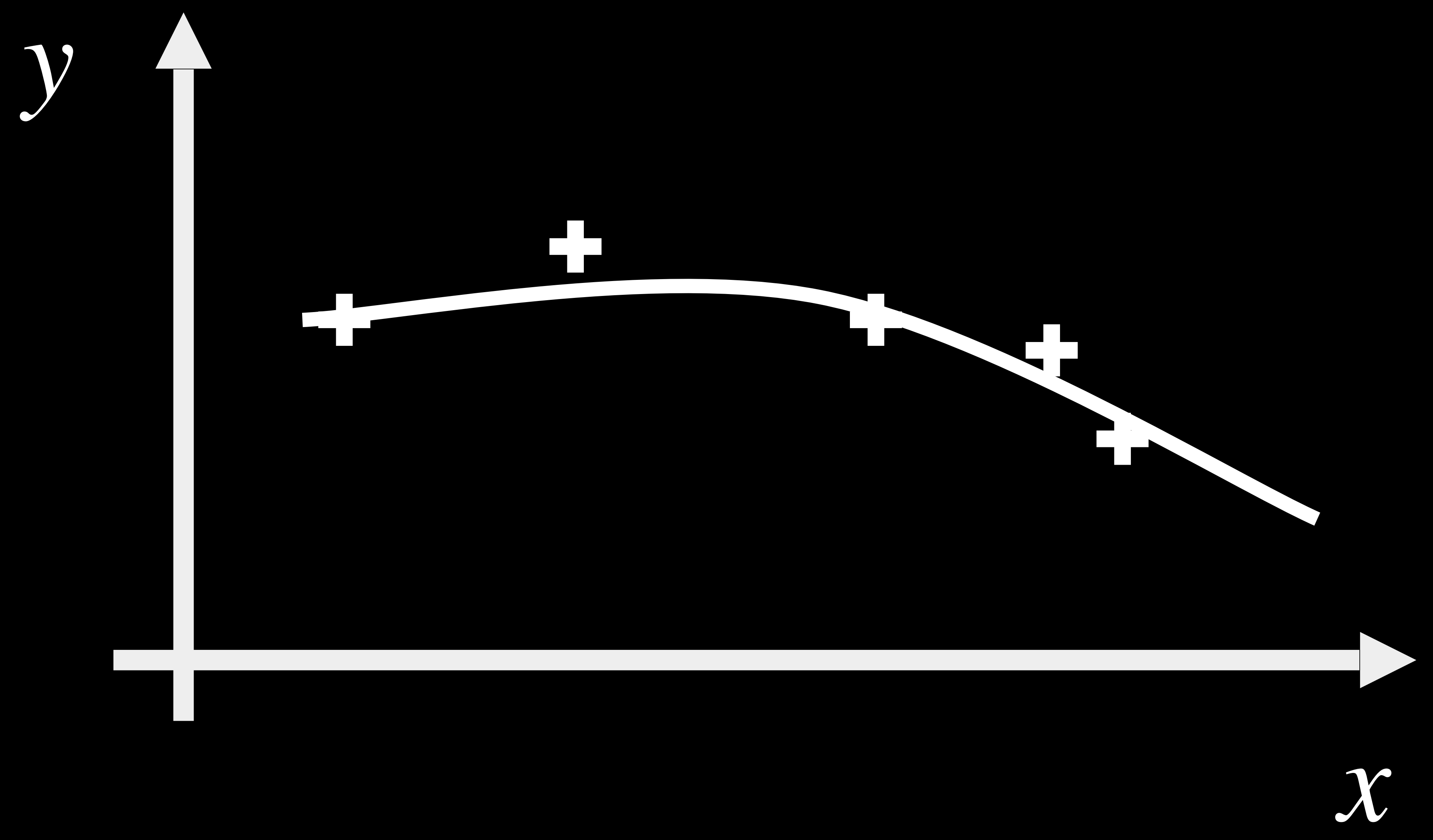


- production & consumption
- line characteristics

- voltage
- power flow

Introduction

Machine Learning



✚ Data (x, y)

x : Productions, Consumptions, Line Characteristics

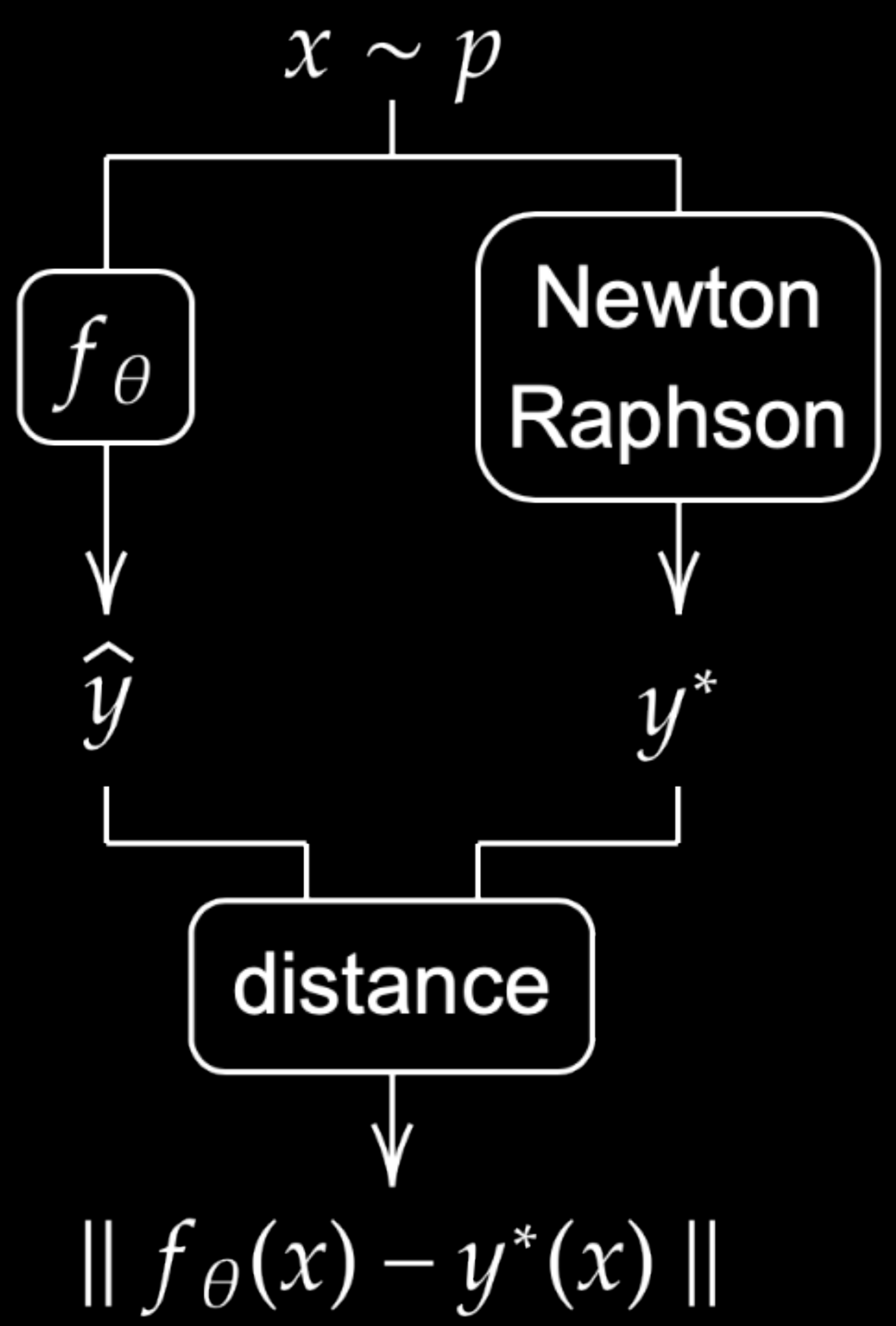
y : Voltage, Power Flows

Approximation f_θ

Learning = minimizing the error $||f_\theta(x) - y||$ by modifying the parameter θ.

Introduction

Proxy Approach



Introduction

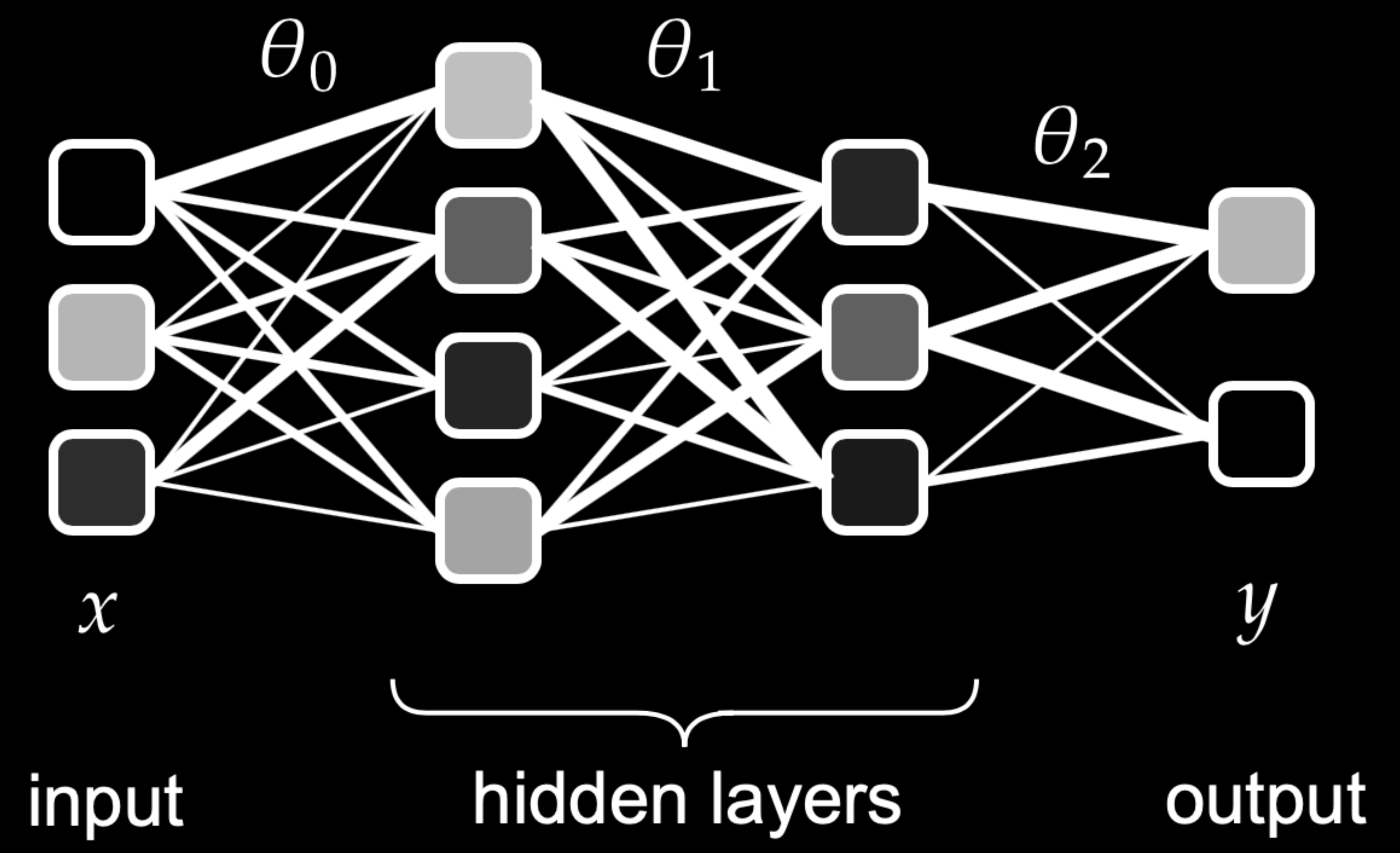
Deep Learning

Deep Neural Network

$$\hat{y} = f_\theta(x)$$

Trainable weights

$$\theta = (\theta_0, \theta_1, \theta_2)$$



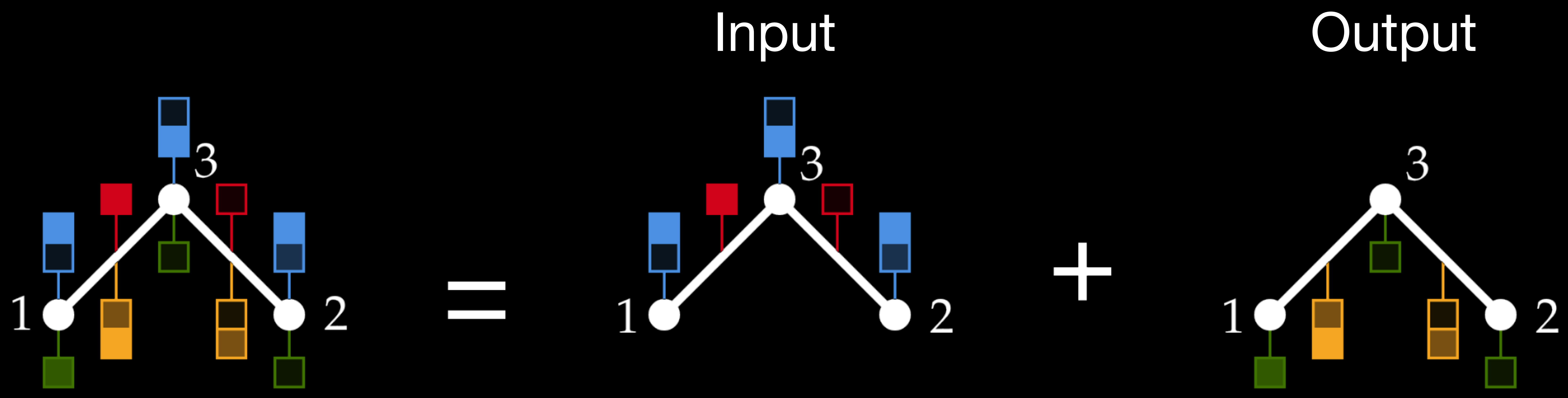
Learning = minimizing the error $||f_\theta(x) - y||$ by modifying the parameter θ.

Introduction
Limitations

- Standard neural networks only work on **vector** data.
- Problem: power grids are **graphs**, and have a varying size and structure.

Introduction

Graph Data



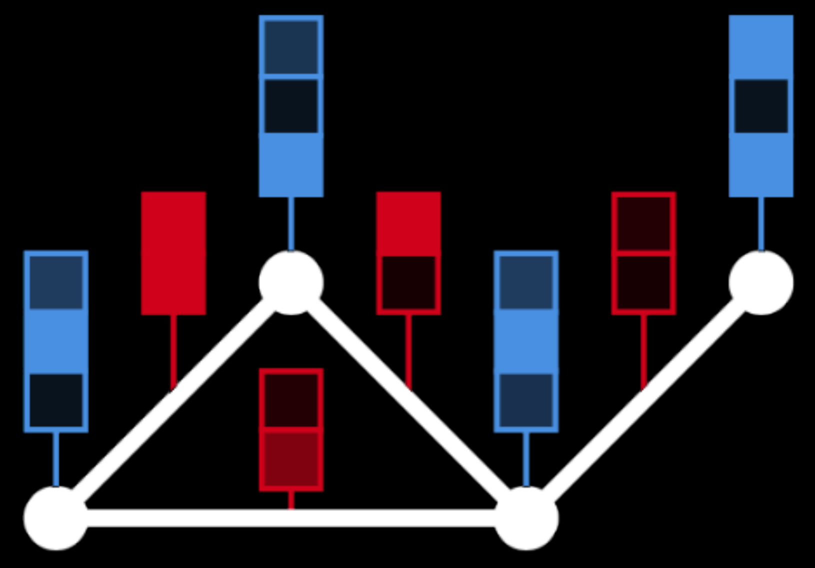
(x, y) $\quad$ $x = (x^v, x^e)$ $\quad$ $y = (y^v, y^e)$

Statistical Solver Problems

Statistical Solver Problems

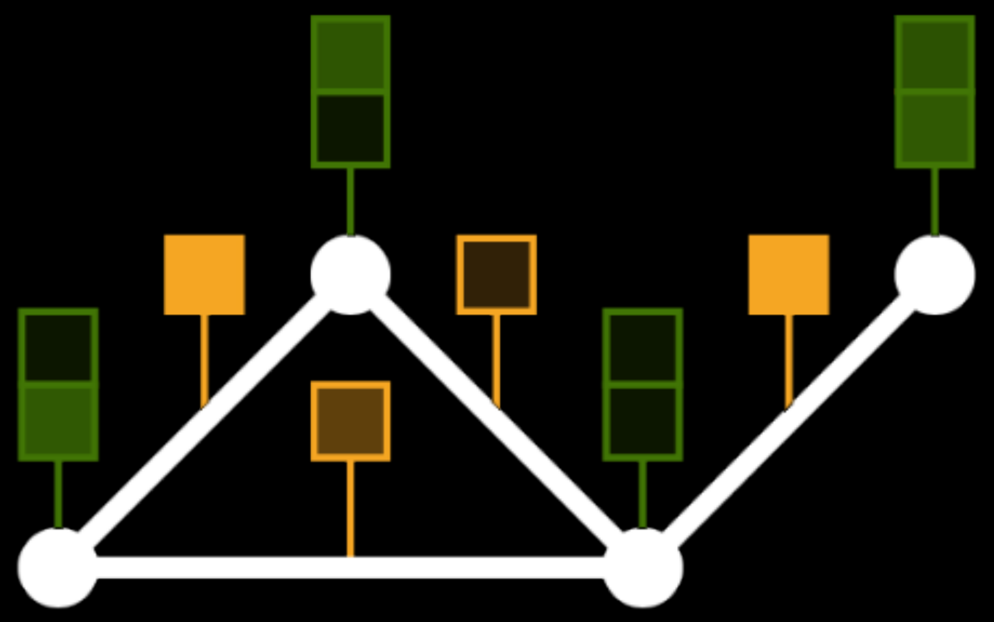
Target Optimization Problem

Input (x)



- production & consumption
- line characteristics

Output (y)

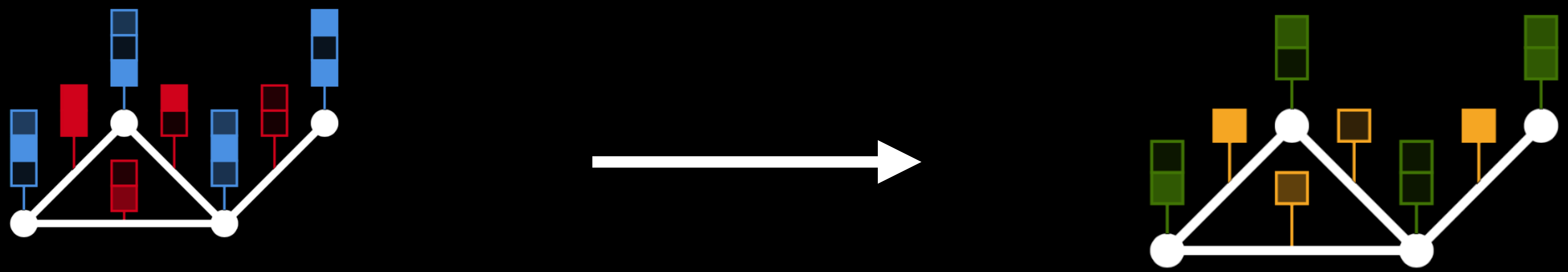


- voltage
- power flow

Statistical Solver Problems

Target Optimization Problem

Input (x) Output (y)



- production & consumption
- line characteristics

- voltage
- power flow

$$y^*(x) = \arg\min_y \ \ell(x, y)$$

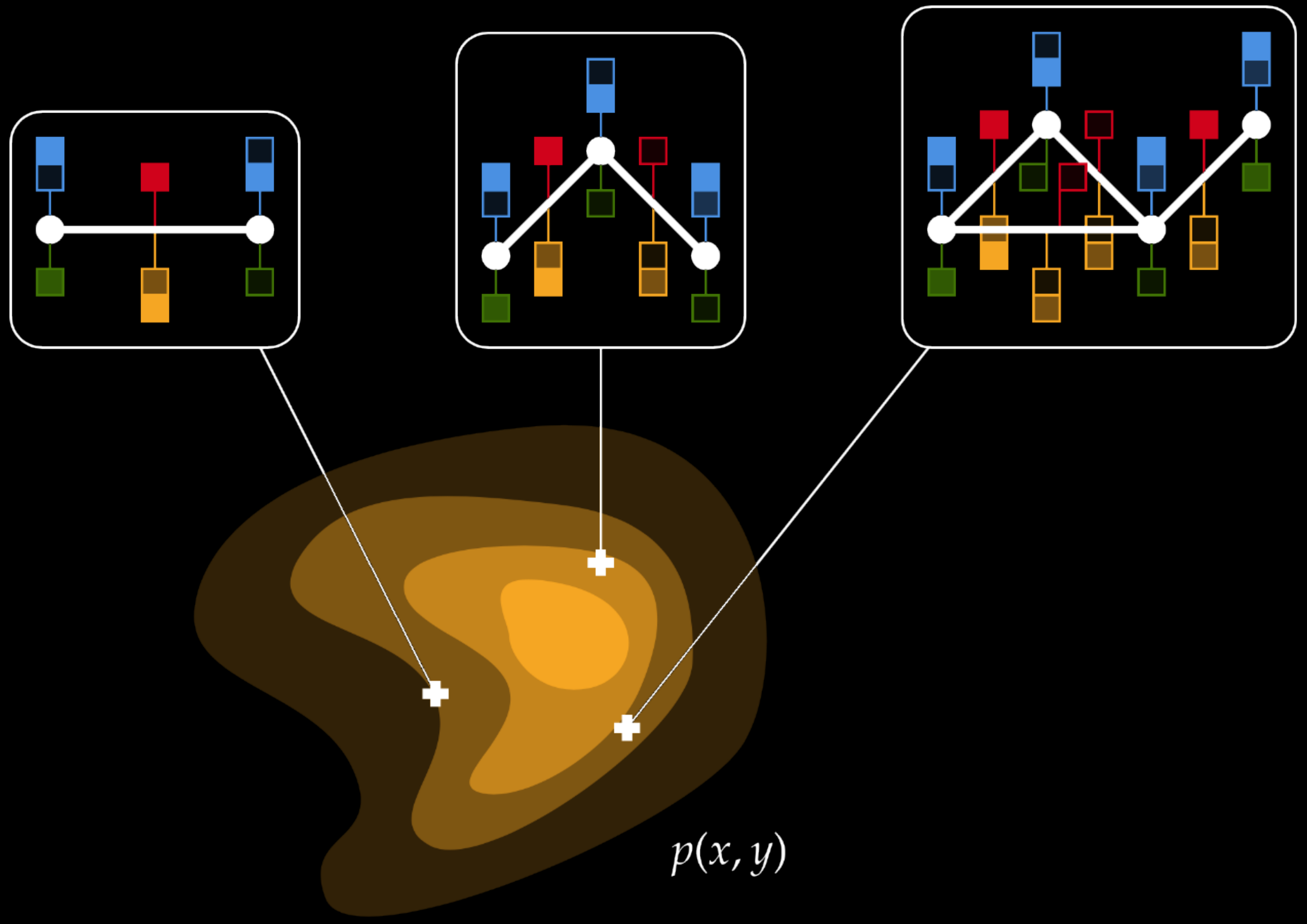
ℓ : Violation of Physical Laws (Kirchhoff)

Statistical Solver Problems

Statistical Conversion

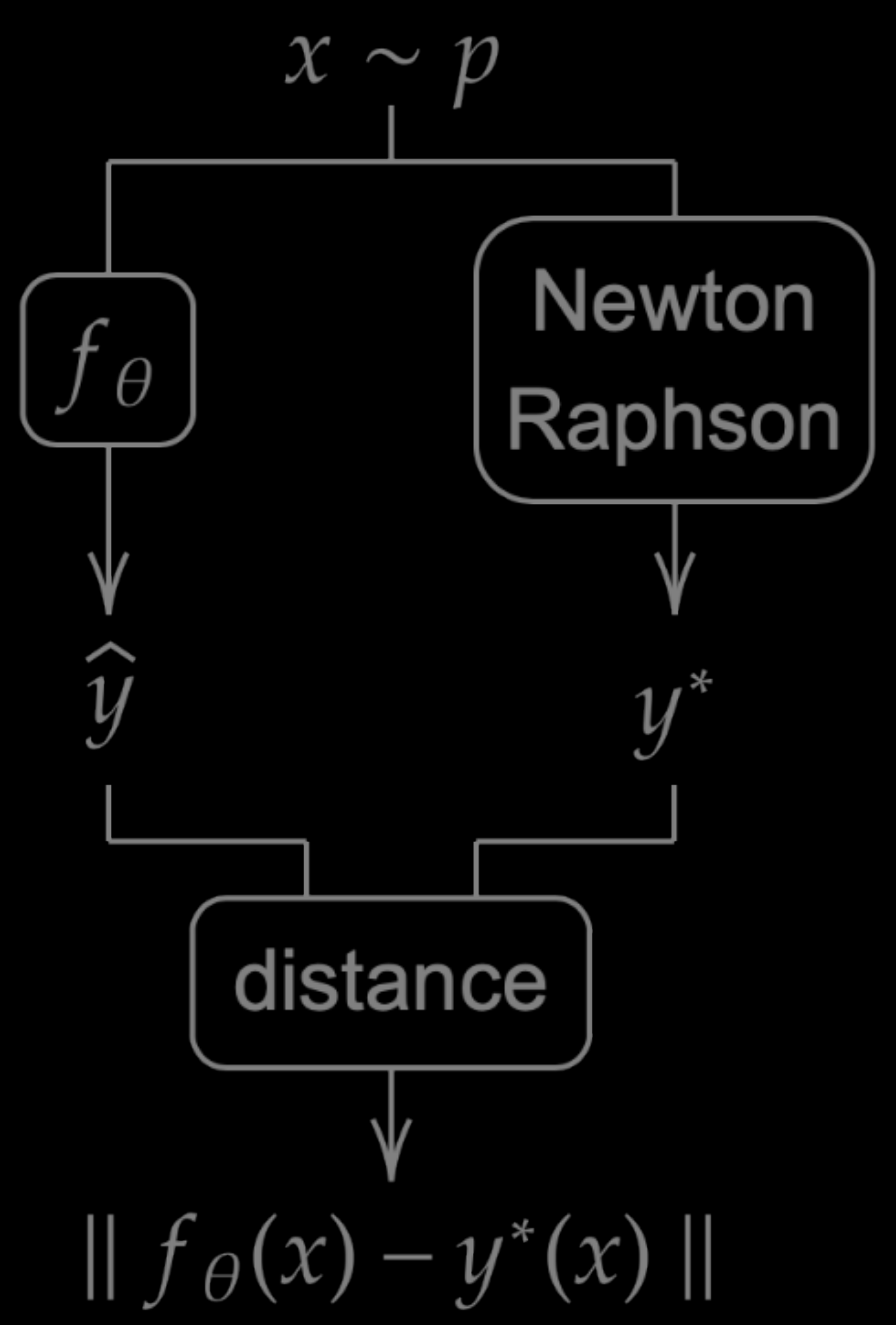
The holy grail :

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{x\sim p}\left[\ell(x, f_\theta(x))\right]$$

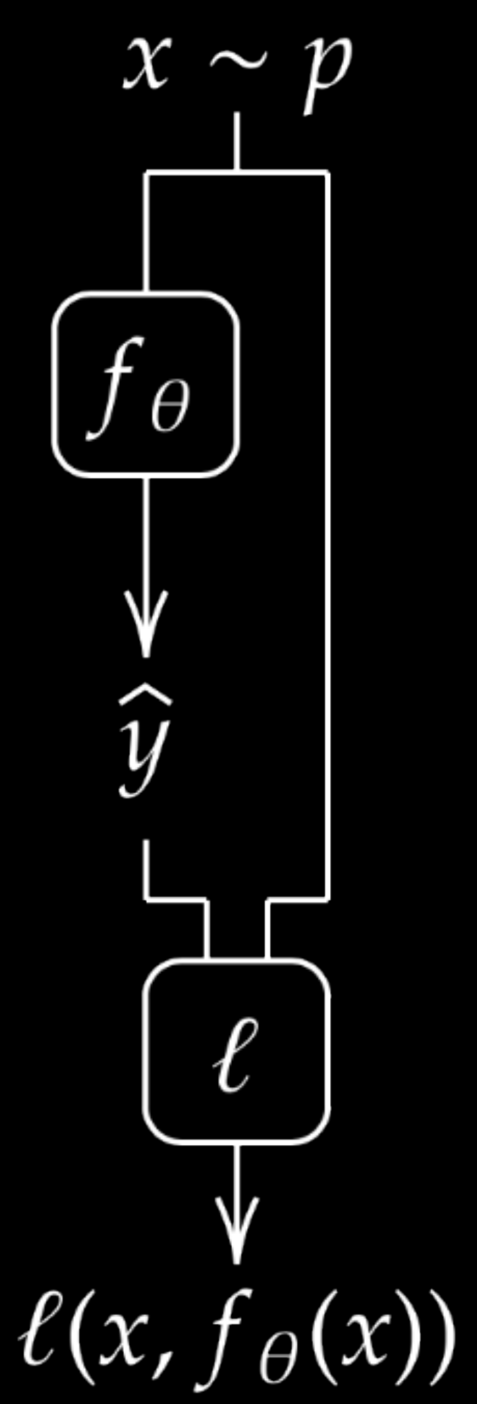


Statistical Solver Problems

Statistical Conversion



Proxy Approach



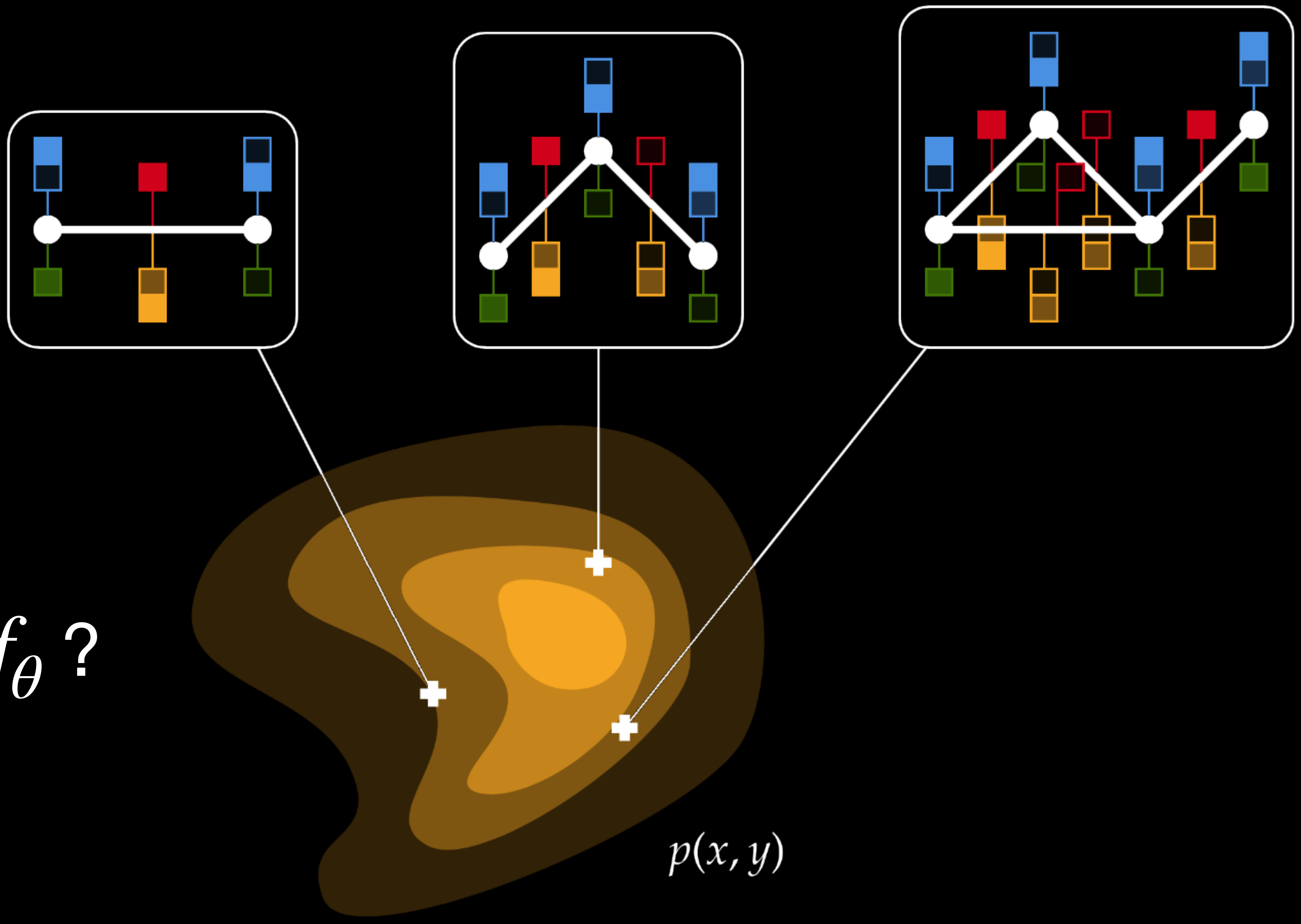
Statistical Solver

Statistical Solver Problems

Statistical Conversion

The holy grail :

$$\theta^* = \arg\min_{\theta} \mathbb{E}_{x \sim p}\left[\ell(x, f_\theta(x))\right]$$

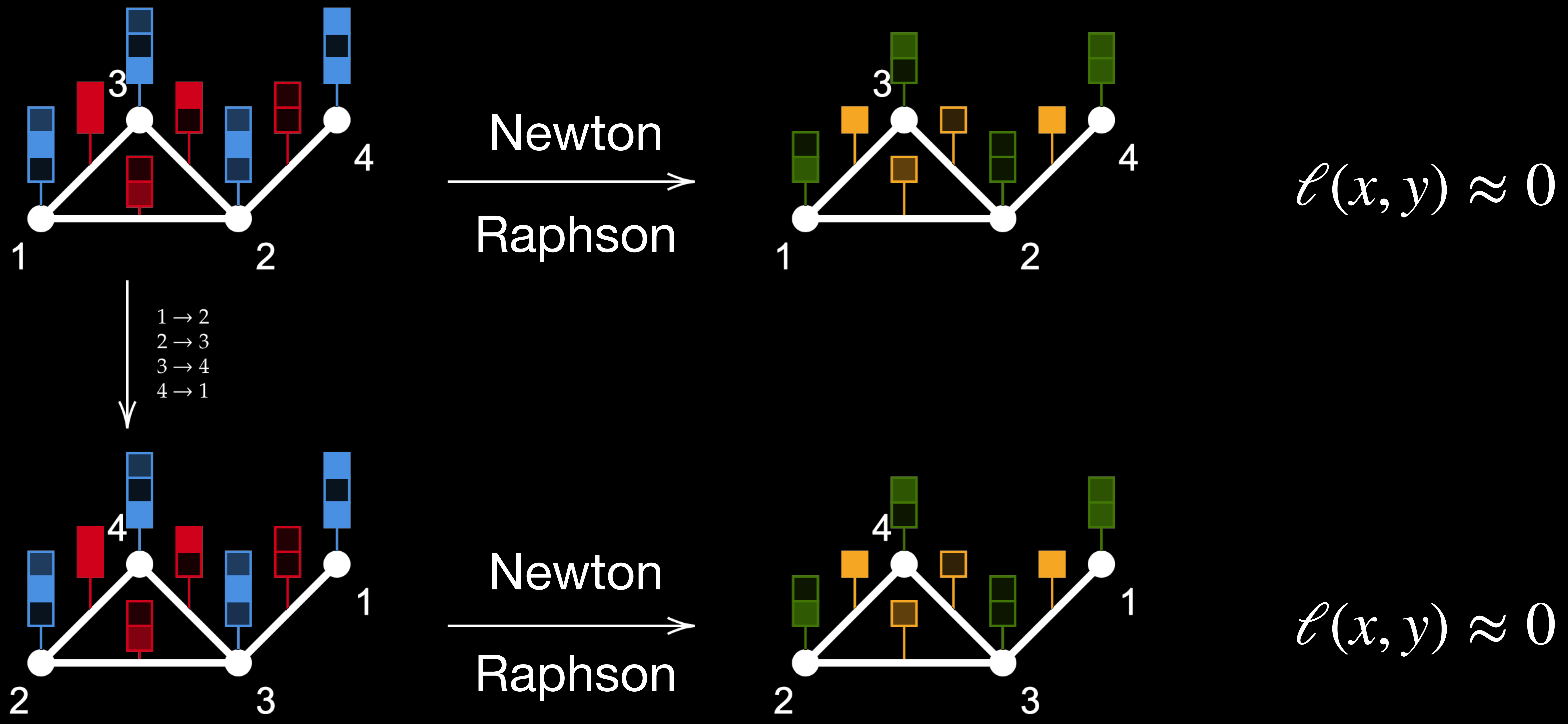


What kind of function should we use for f_θ ?

- Standard Neural Networks ?

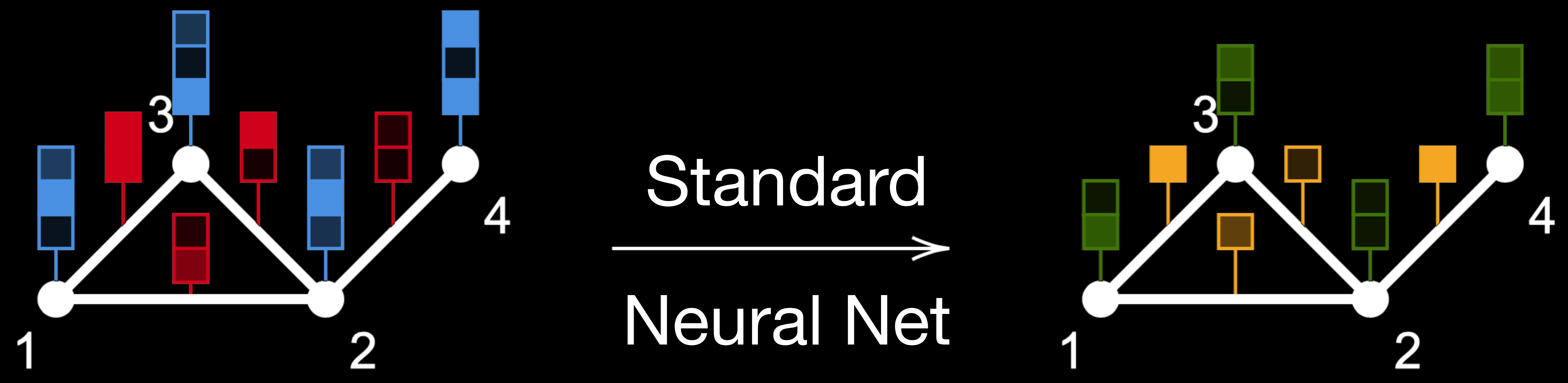
Statistical Solver Problems

Limits of Standard Neural Networks



Statistical Solver Problems

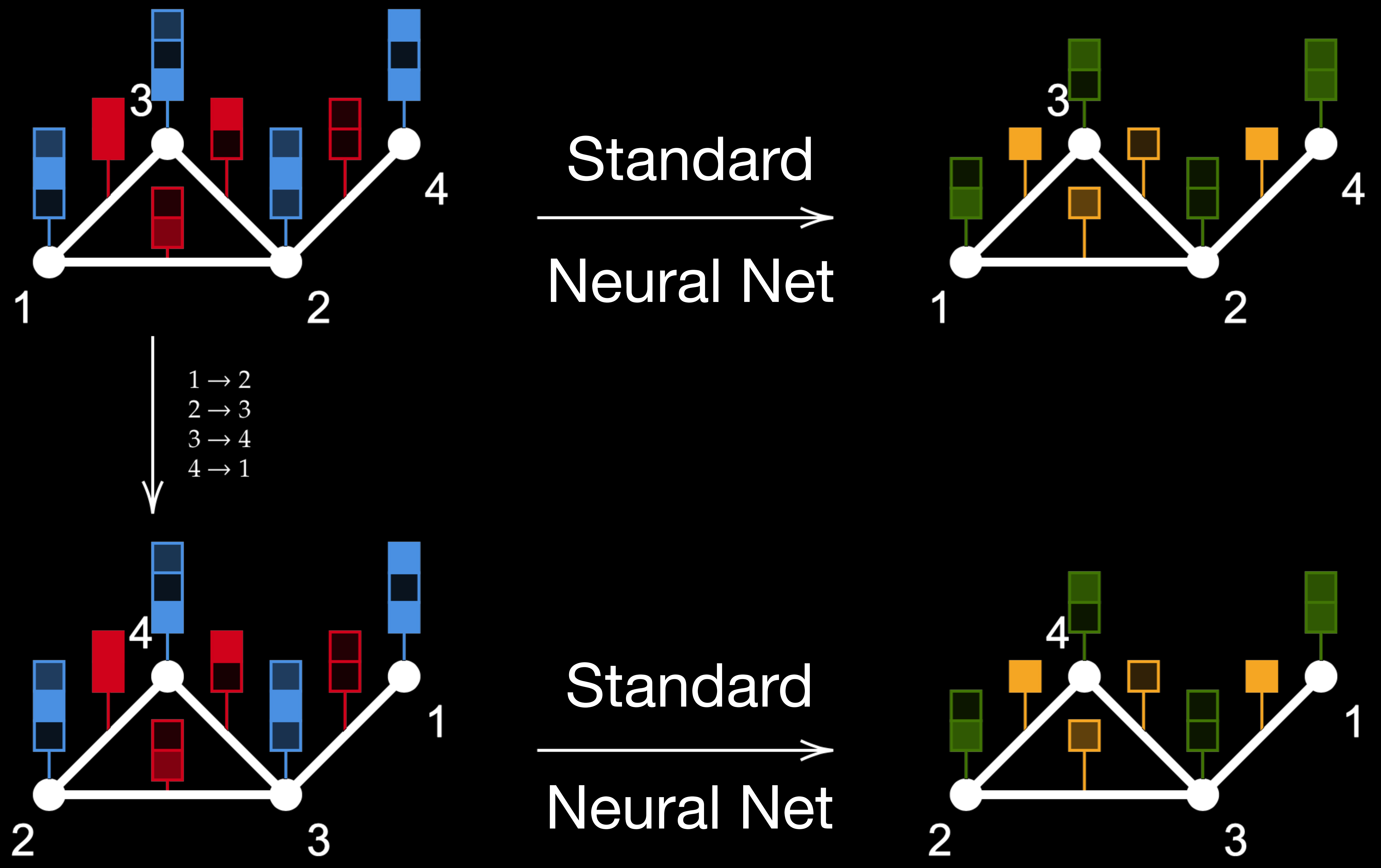
Limits of Standard Neural Networks



$$\ell(x, y) \approx 0$$

Statistical Solver Problems

Limits of Standard Neural Networks



$\ell(x, y) \approx 0$

$\ell(x, y) \gg 0$

Standard Neural Networks are not permutation-equivariant.

Graph Neural Networks

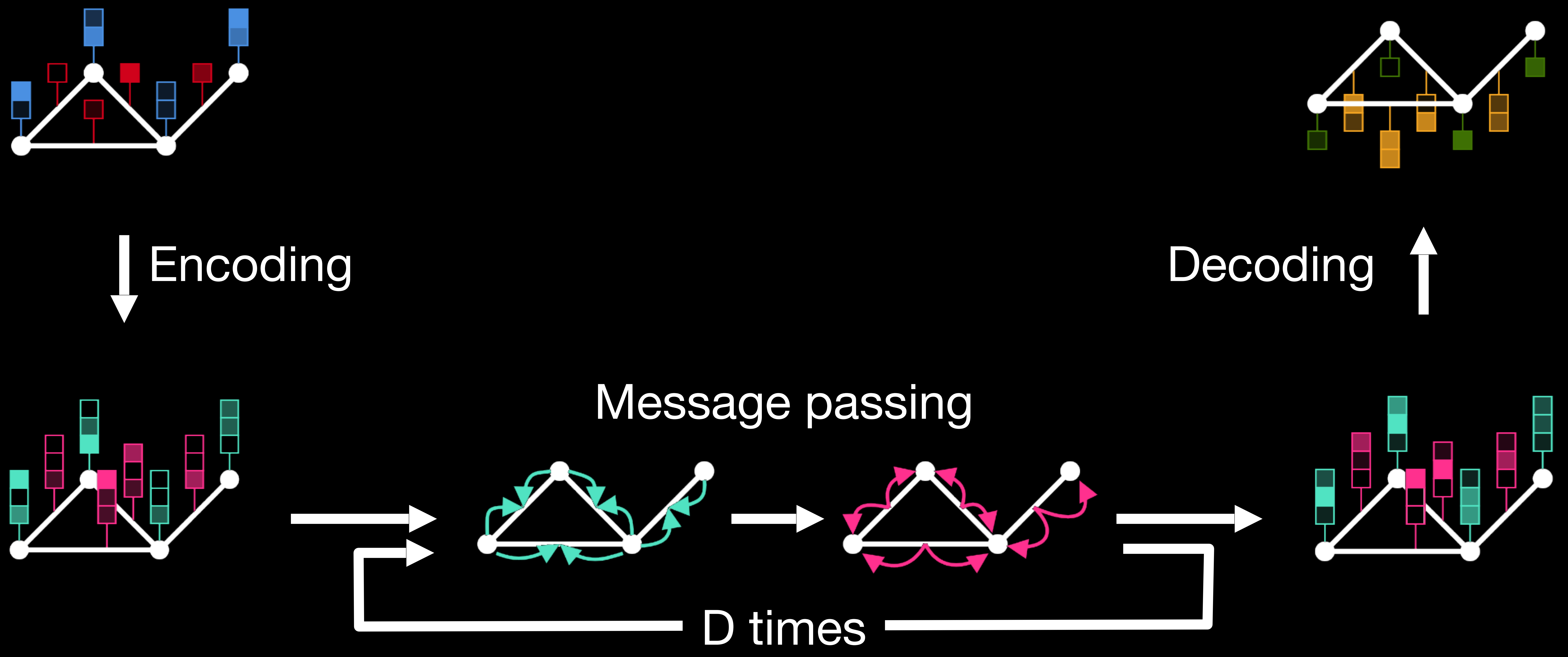
Graph Neural Networks

Message Passing GNN

- Message Passing Graph Neural Networks :
  - Iterative process
  - Propagates information locally

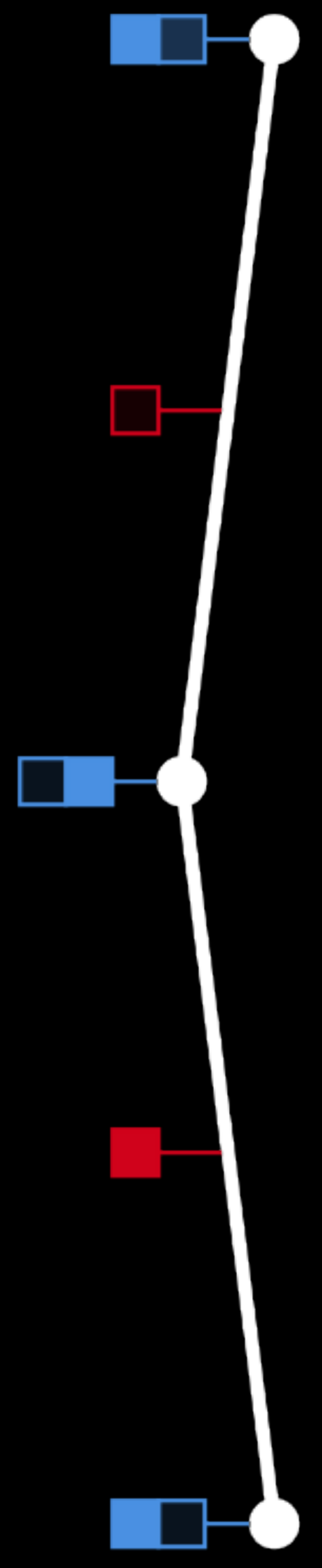
Graph Neural Networks

Message Passing GNN



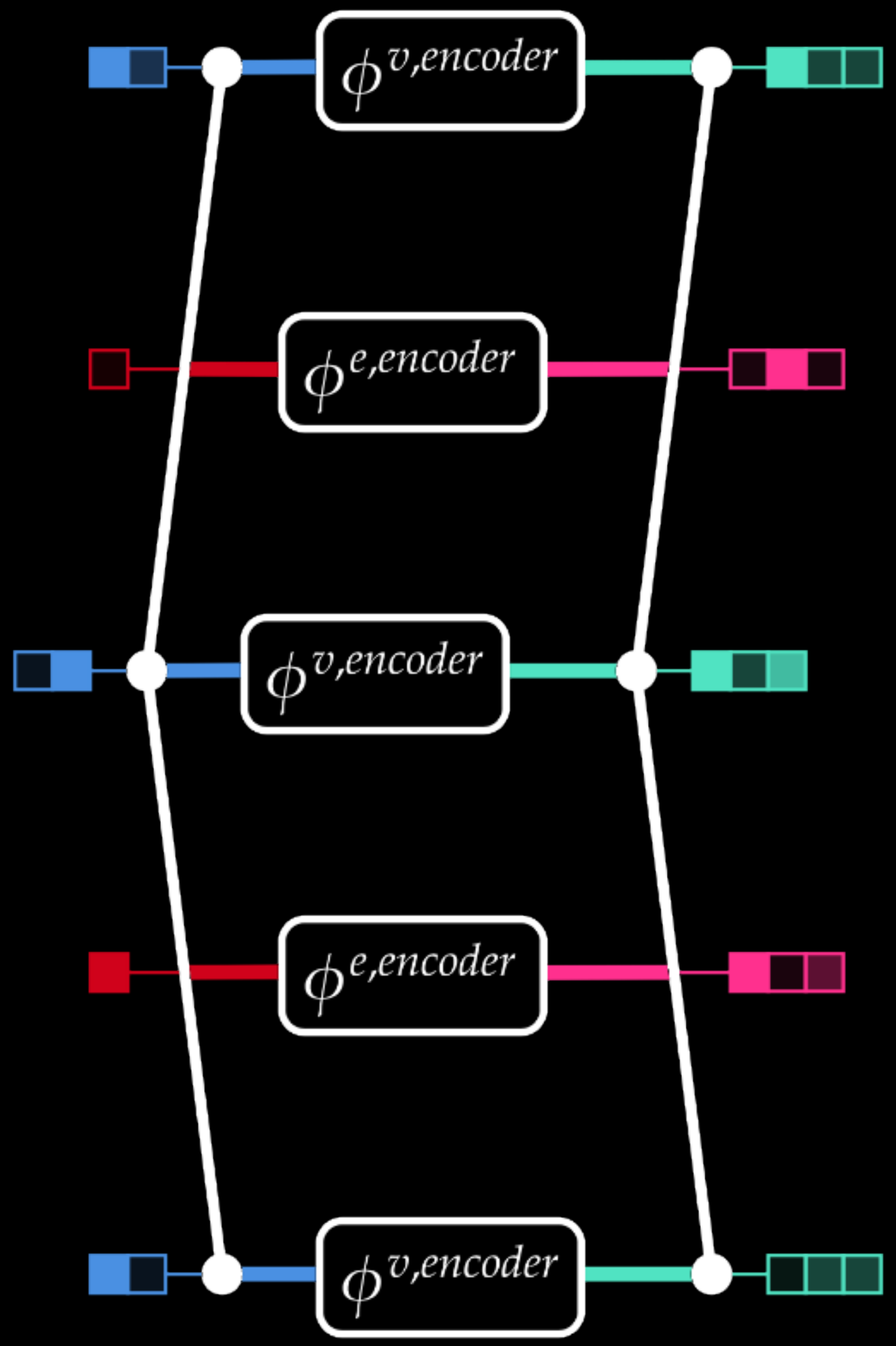
Graph Neural Networks

Example



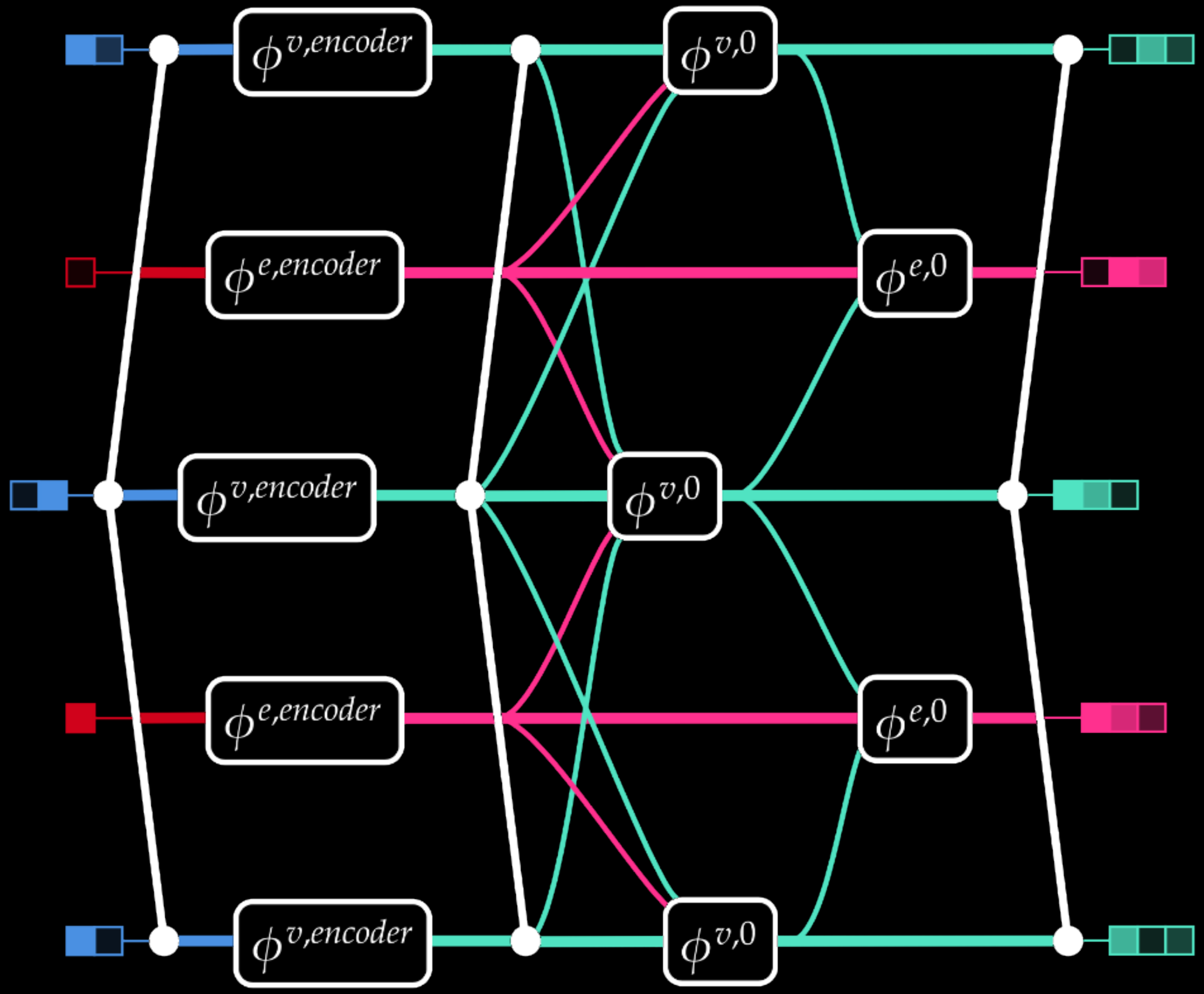
Graph Neural Networks

Example



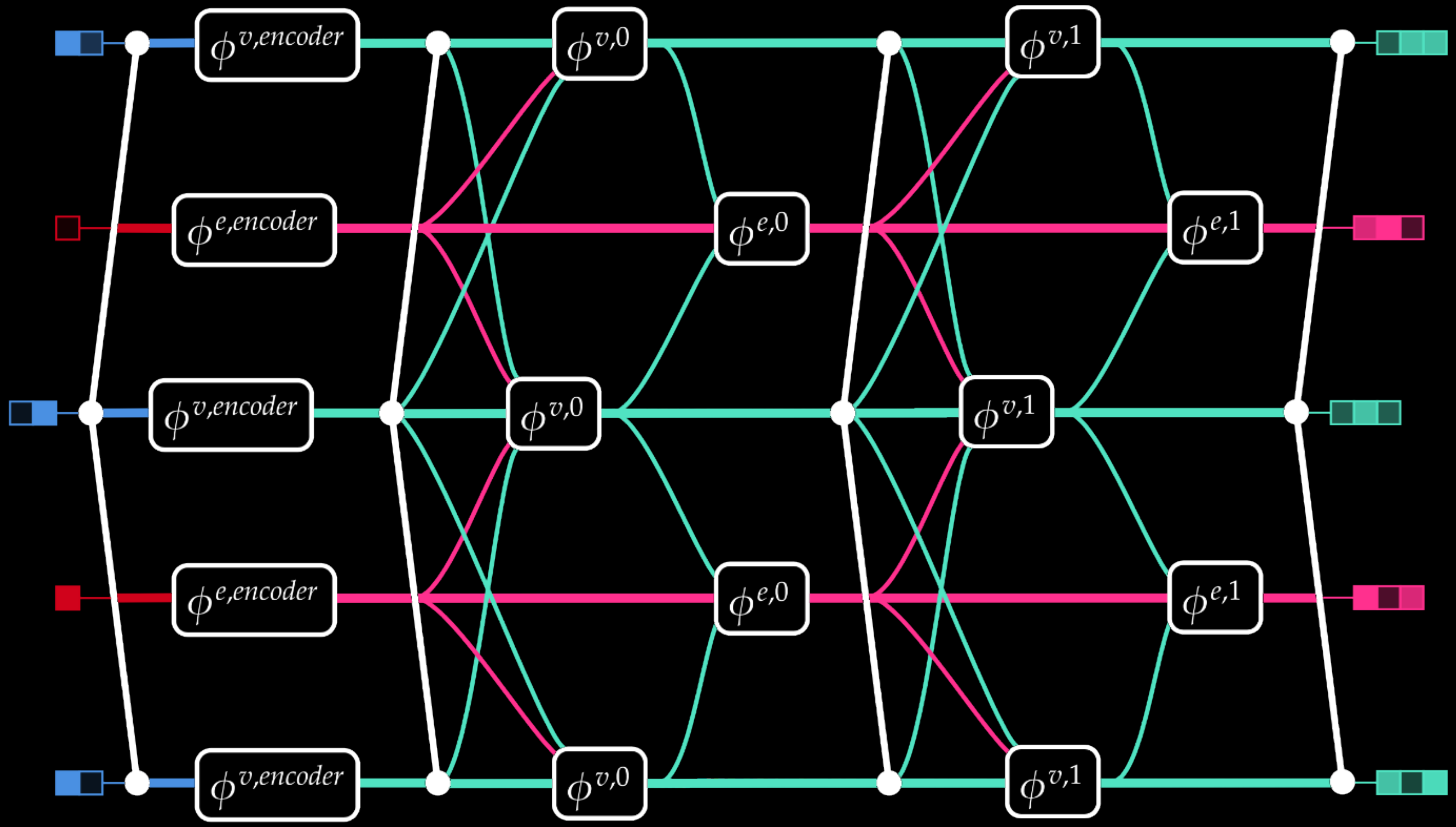
Graph Neural Networks

Example



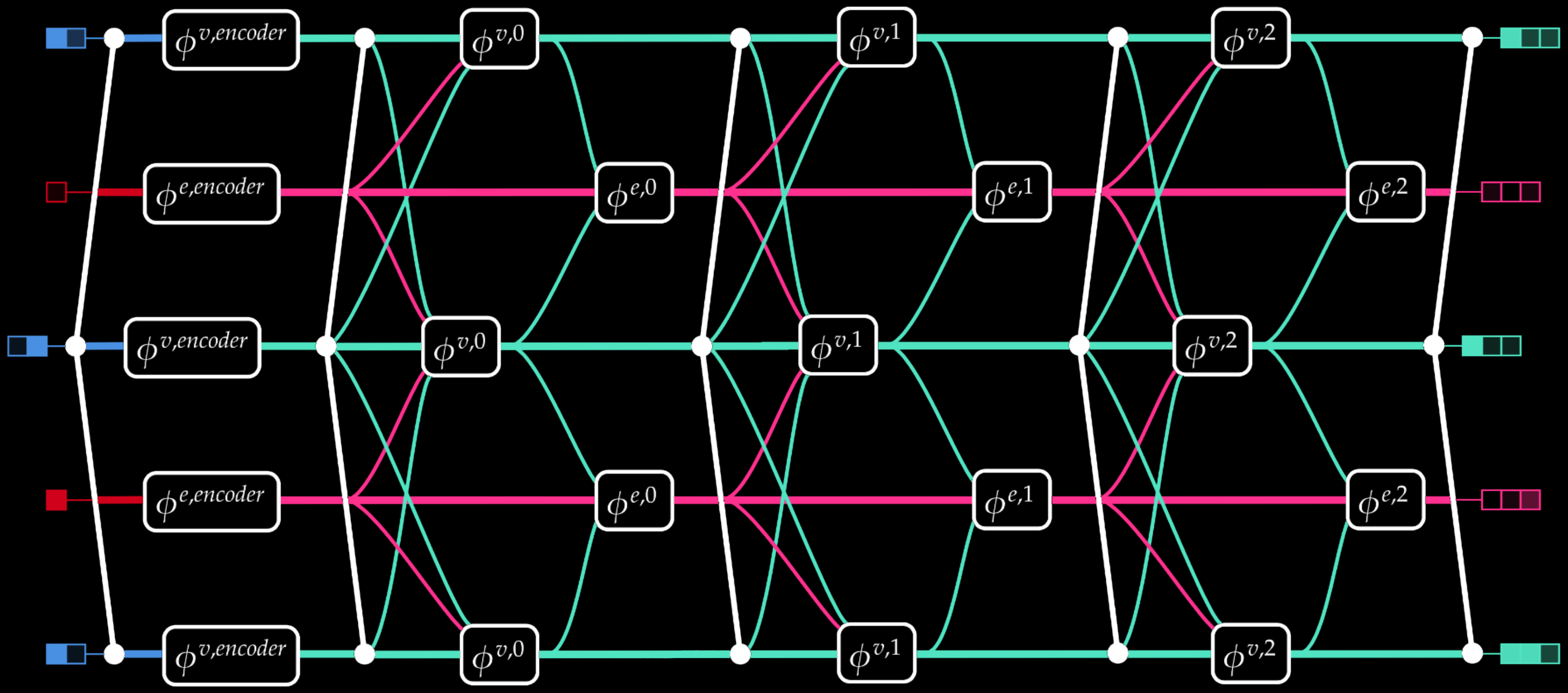
Graph Neural Networks

Example



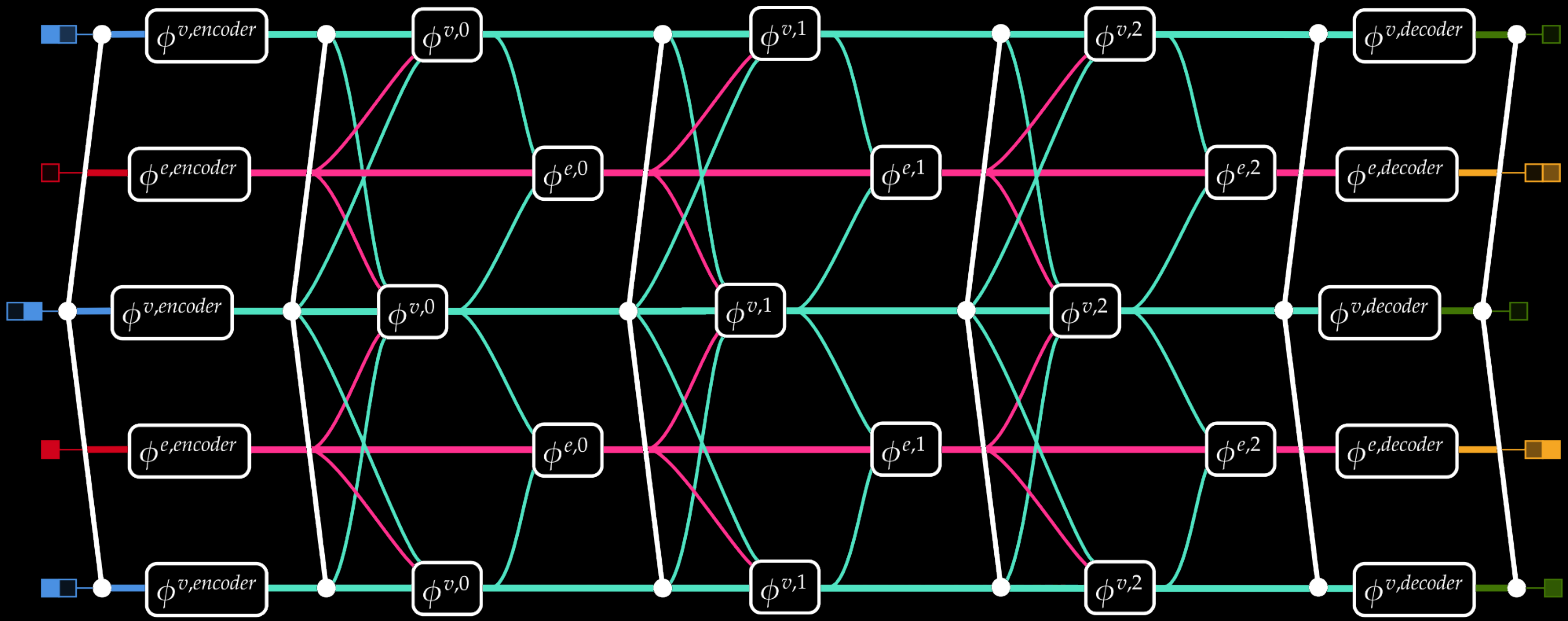
Graph Neural Networks

Example



Graph Neural Networks

Example



Graph Neural Networks

Properties

- Only local operations.
- Neural network blocks are trained jointly.
- Permutation equivariant.

Graph Neural Networks
Deep Statistical Solver

- A Deep Statistical Solver is a Graph Neural Network f_θ trained so as to minimize

$$\mathbb{E}_{x \sim p}\left[\ell(x, f_\theta(x))\right]$$

Universal Approximation Theorem

Universal Approximation Theorem

Summary

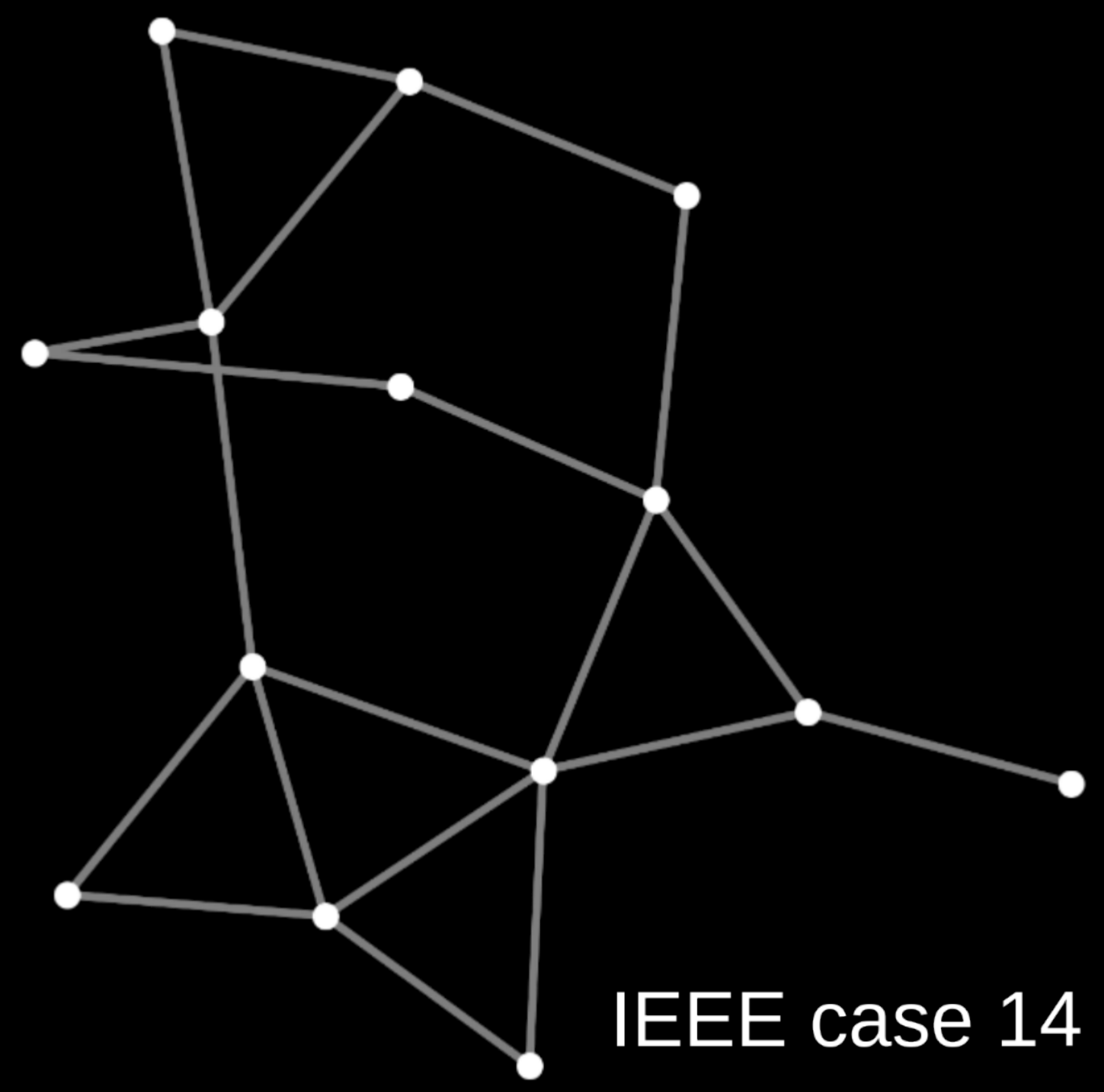
- GNNs can approximate an actual Newton-Raphson solver to an arbitrary precision.
- It requires more propagation steps than the largest graph diameter.

Power Systems Experiments

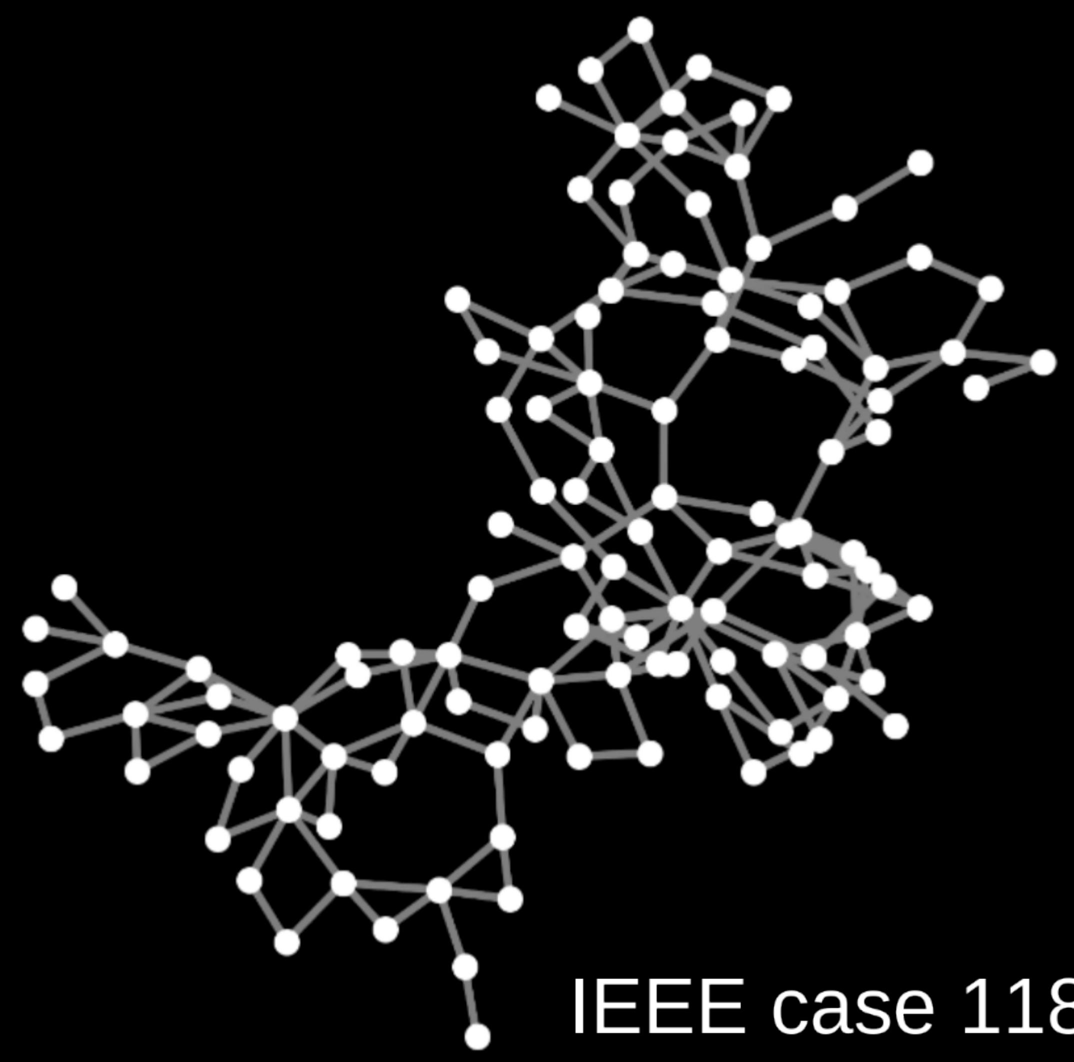
Power Systems Experiments

Setup

- Random injections
- Random line disconnections



IEEE case 14



IEEE case 118

Power Systems Experiments

Results

Dataset		IEEE 14 nodes			IEEE 118 nodes		
Method		DSS	Proxy	NR	DSS	Proxy	NR
Corr. w/ NR	v_i	.9993	$>$ .9999	-	.9979	$>$ .9999	-
	ϑ_i	.9986	$>$ .9999	-	.8131	$>$ .9999	-
	p_{ij}	$>$ .9999	$>$ .9999	-	$>$ .9999	$>$ .9999	-
	q_{ij}	$>$ .9999	$>$ .9999	-	$>$ .9999	$>$ .9999	-
nRMSE w/ NR	v_i	2.0e-3	4.9e-4	-	1.4e-3	1.2e-3	-
	ϑ_i	7.1e-3	1.7e-3	-	5.7e-2	4.5e-3	-
	p_{ij}	6.2e-4	2.6e-4	-	1.0e-3	3.9e-4	-
	q_{ij}	4.2e-4	2.0e-4	-	1.1e-4	1.7e-4	-
Loss 10^{th} p.		4.2e-6	2.3e-5	1e-12	1.3e-6	6.2e-6	3e-14
Loss 50^{th} p.		1.0e-5	4.0e-5	2e-12	1.7e-6	8.3e-6	4e-14
Loss 90^{th} p.		4.4e-5	1.2e-4	3e-12	2.5e-6	1.3e-5	6e-14

Our trained DSS models are highly correlated with the Newton-Raphson solutions.

Summary

PhD Summary
Main Contributions

We have designed a full-fledged optimization method that

- is trained by direct minimization of physical laws;
- aims at solving a distribution of problem instances;
- respects the permutation-invariance of the problem;
- relies on the training of a graph neural network heuristics;
- is theoretically grounded by a Universal Approximation Theorem.

GNNs for Voltage Management

Post Doctoral Research Project

Project Overview

Project Overview

Team

- Collaboration between **RTE R&D** and **Institut Montefiore** (Liège Université).

Louis Wehenkel
Professor - ULiège

François Cubelier
PhD student - ULiège

Efthymios Karangelos
Postdoc - ULiège

Laure Crochepierre
Researcher - RTE

Balthazar Donon
Postdoc - ULiège

Project Overview

Industrial Motivations

- Increasing amount of high voltage events.
- In the future, it may become even more complex.
  - We want to anticipate and develop real-time methods for voltage management, based on Graph Neural Networks (GNNs).

Project Overview
Industrial Motivations

- To ensure that voltages are admissible, dispatchers can adjust :
  - Voltage set points at pilot buses (secondary voltage control)
  - The opening / closing of transmission lines
  - Activate / deactivate shunts
  - Activate / deactivate synchronous condensers
  - Set points of static VAR compensators

Project Overview

Research Motivations

- Prove the viability of GNNs for real-life industrial problems
- GNN-based control policy for voltage management

Project Overview

Research Motivations

x - Power Grid Snapshot

$\downarrow$

$\boxed{\Pi_\theta \text{ - Control Policy}}$

$\downarrow$

y - Action

Project Overview
Research Motivations

x - Power Grid Snapshot

Π_θ - Control Policy

y - Action

AC Power Flow Simulator

r - Reward

Project Overview

Research Motivations

x - Power Grid Snapshot

Π_θ - Control Policy

y - Action

AC Power Flow Simulator

Non-differentiable

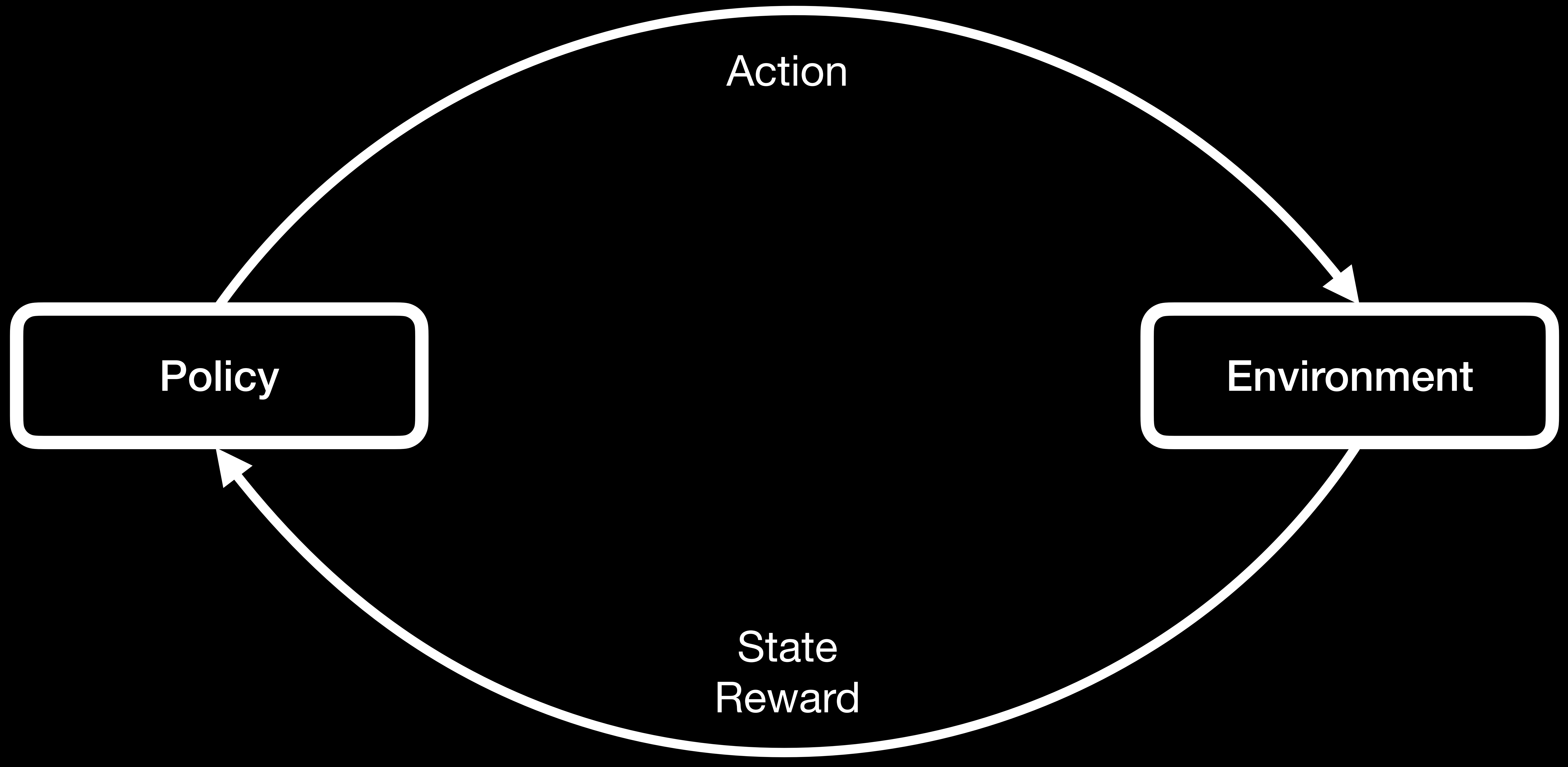
r - Reward

Project Overview
Simulator

- Non-differentiable simulator
  - Discrete control variables
  - Divergence issues
  - Easier to replace

- Need to use gradient-free methods (Reinforcement Learning literature)

Project Overview
Simulator



Project Overview
Reward

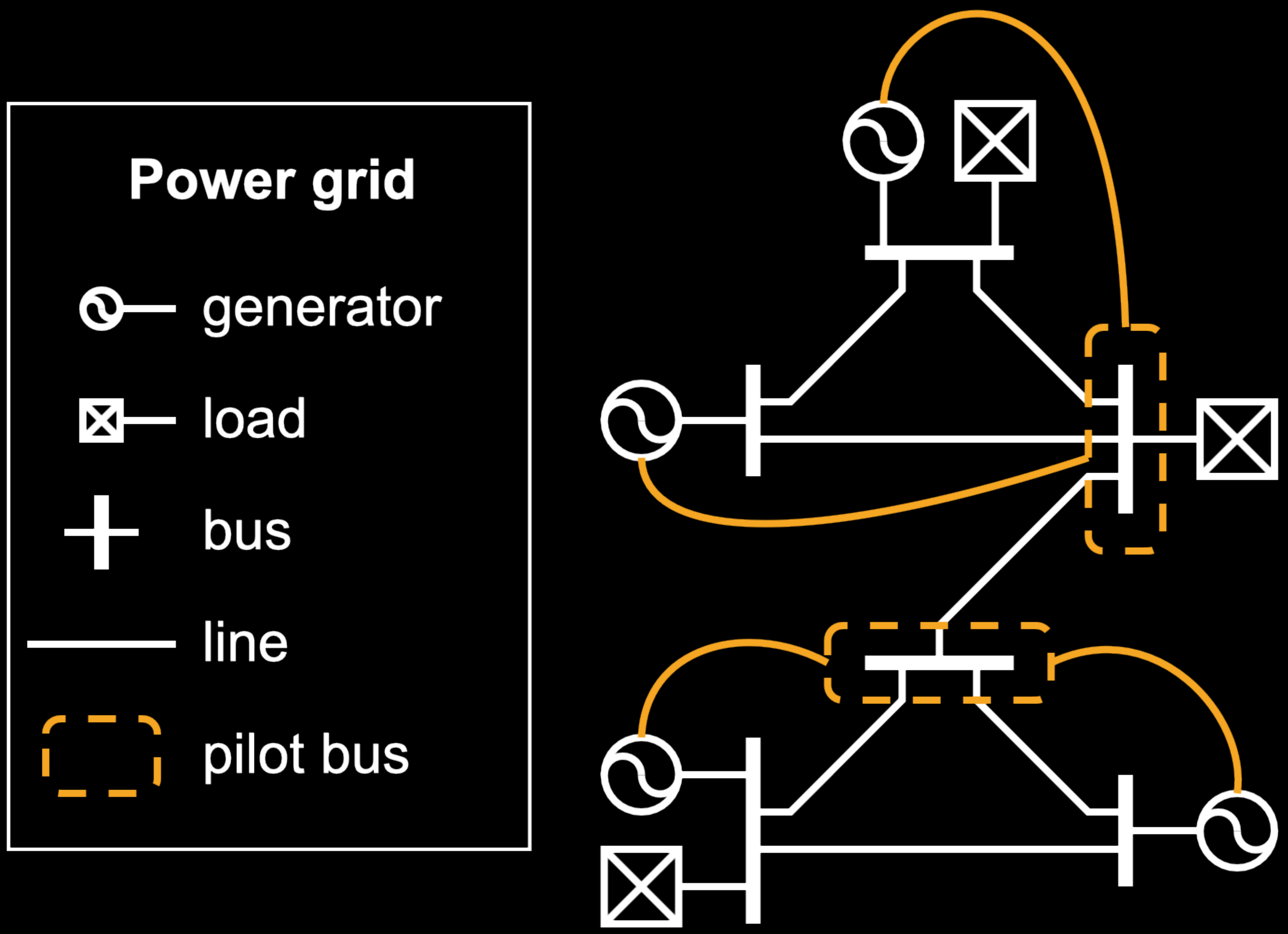
- Primary objective : Joule losses
- Relaxed constraints :
  - Voltage in admissible range
  - No line overflow
  - Generator reactive margins

Can we train a GNN-based policy to solve real-life voltage management problems in an unsupervised and gradient-free way ?

Data & Neural Networks

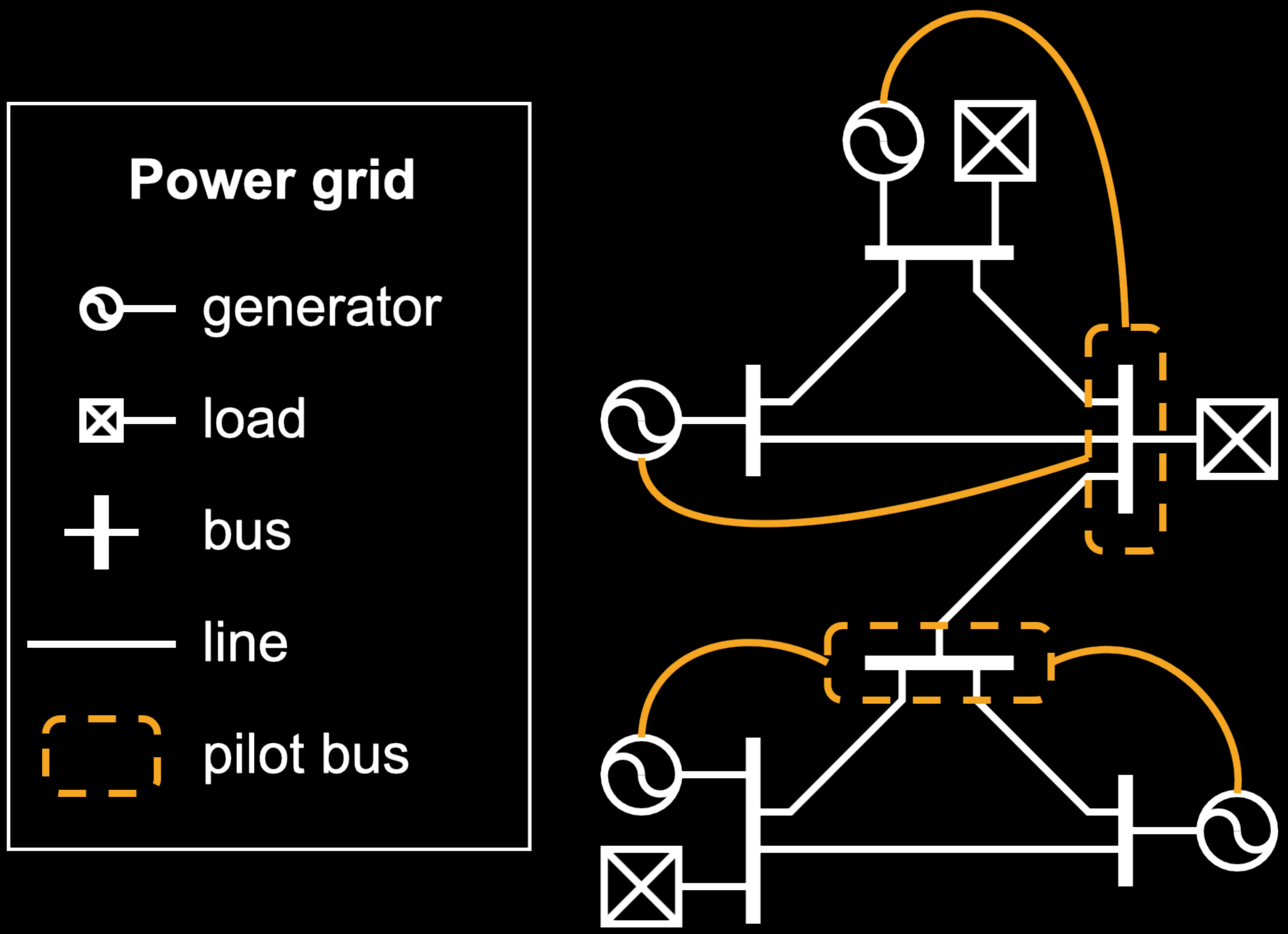
Data & Neural Networks

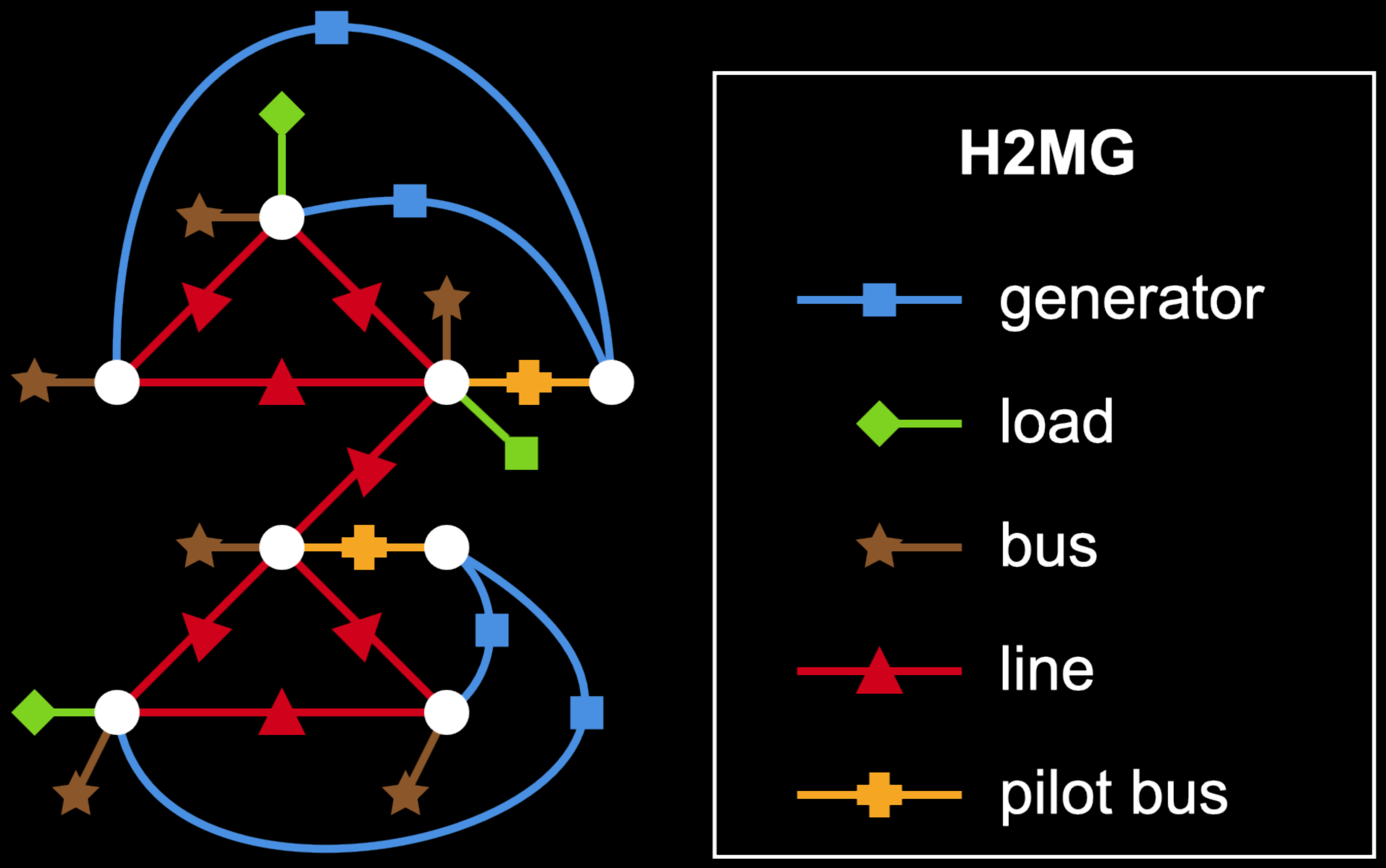
Power Grid vs. Hyper Heterogeneous Multi Graph (H2MG)



Data & Neural Networks

Power Grid vs. Hyper Heterogeneous Multi Graph (H2MG)

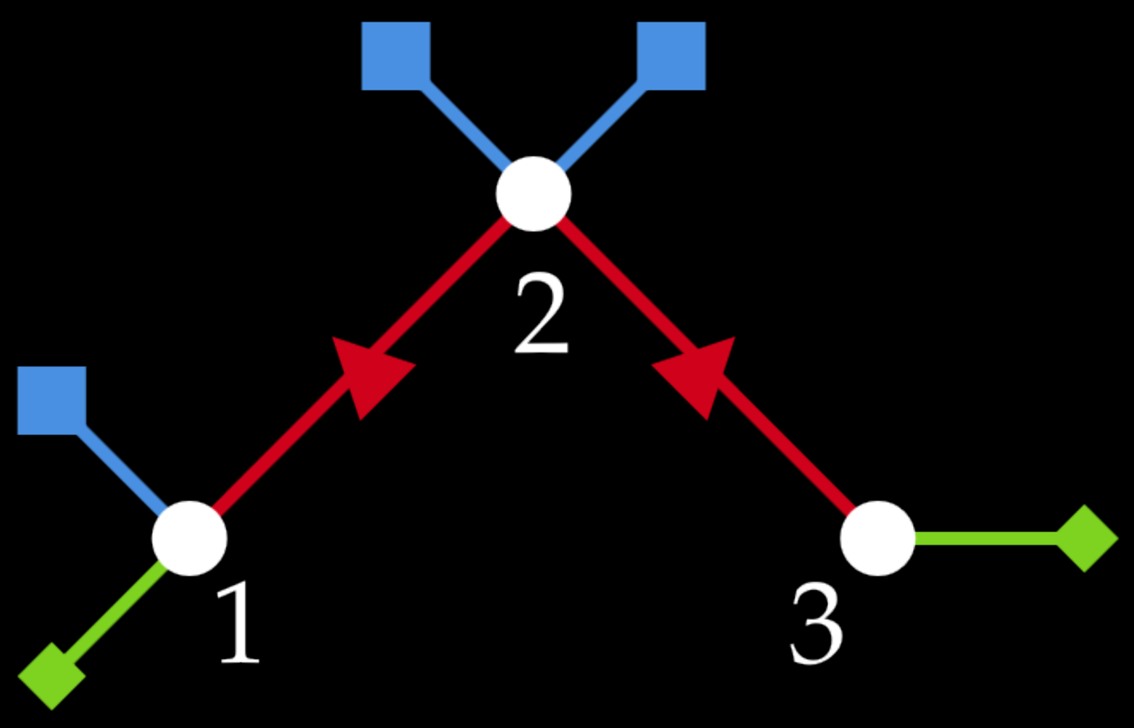




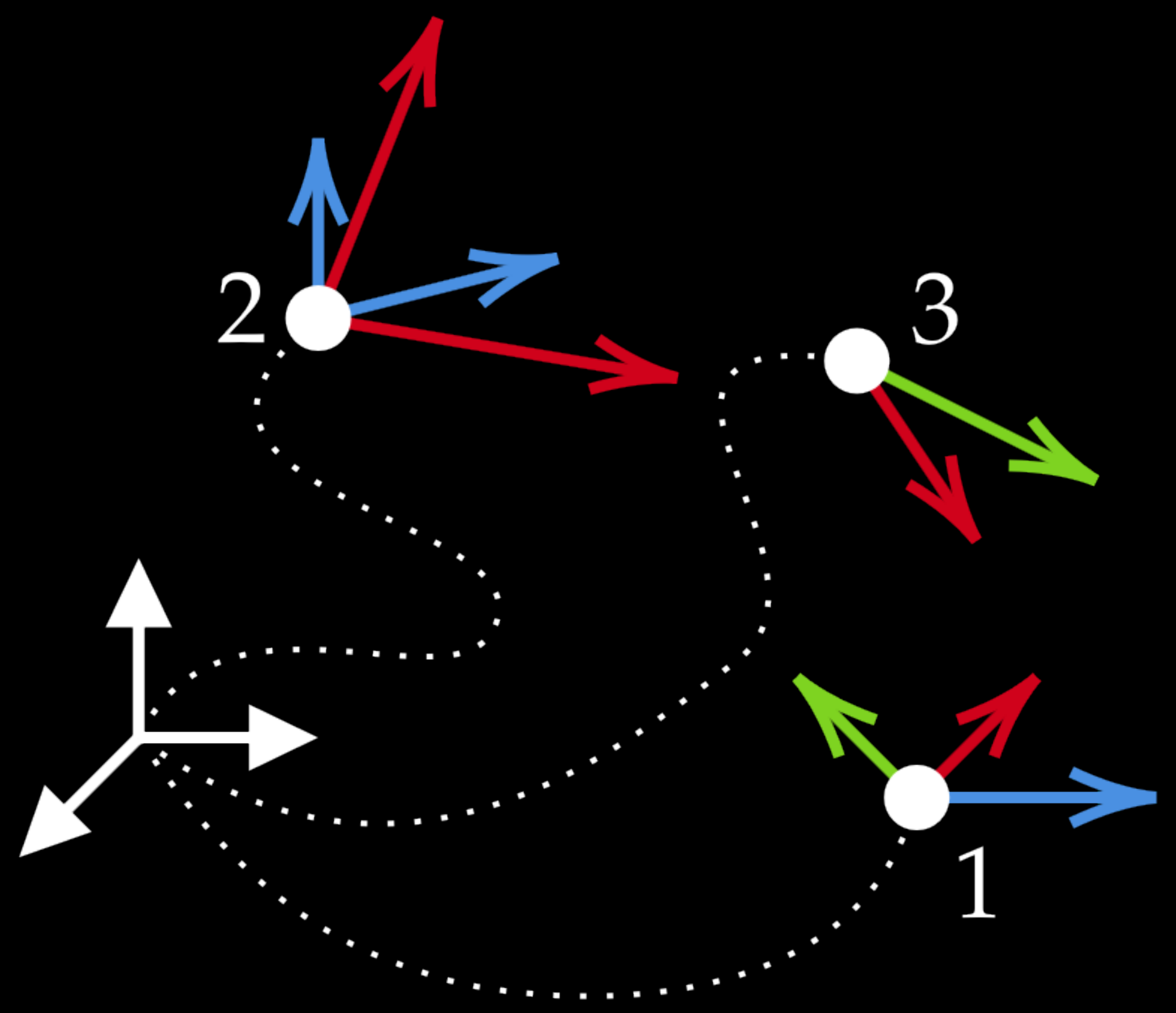
Data & Neural Networks

Example

Input data



Latent trajectories



$$\frac{dh_i}{ds} = \tanh\left[\sum_{(c,e,o)\in\mathcal{N}(i)} \Phi_\theta^{c,o}\left(x^g, x_e^c, h^g, \left(h_{o(e)}\right)_{o\in\mathcal{O}^c}, t\right)\right]$$

Data & Neural Networks

Trajectories

Data & Neural Networks

Main Novelties

- Hyper Heterogeneous Multi Graph (H2MG)
  - Seamless representation of power grids.
  - Cyber-physical graph structure.
- Hyper Heterogeneous Multi Graph Neural Ordinary Differential Equation (H2MGNODE)
  - Dynamical system defined over H2MGs.

Artificial Data

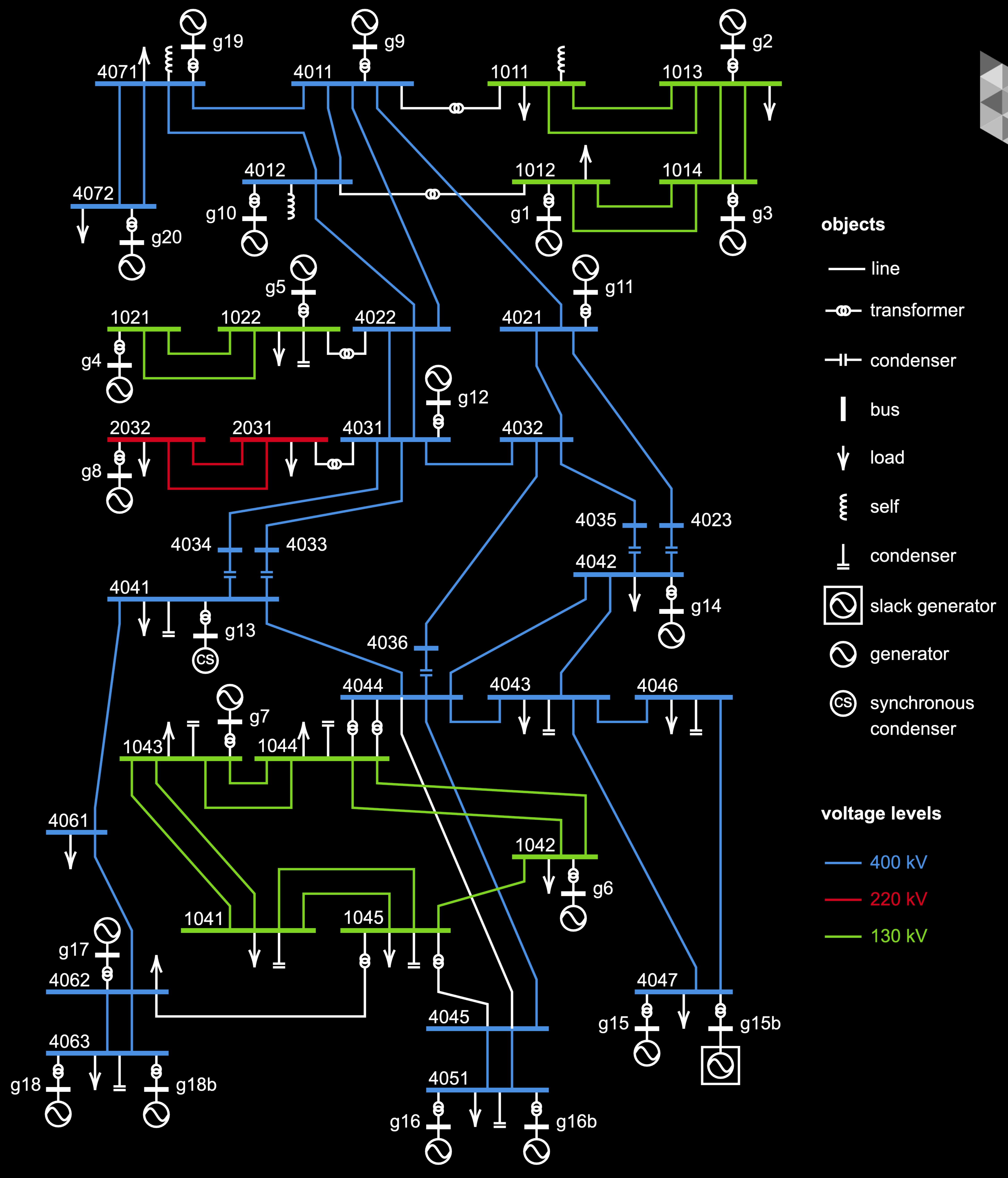
Artificial Data
Summary

- Swedish transmission system.
- Stems from and extends the Nordic32 test case (CIGRE Task Force 38.02.08).
- Enriched first by the research group of Thierry Van Cutsem, and then by Florin Capitanescu.
- 60 buses, 23 generators, 57 lines, 31 transformers, 22 loads, 12 shunts.

Suppressing ineffective control actions in optimal power flow problems, F. Capitanescu, IET Generation, Transmission & Distribution, 14 (13), pp. 2520-2527, 2020. doi: 10.1049/iet-gtd.2019.1783

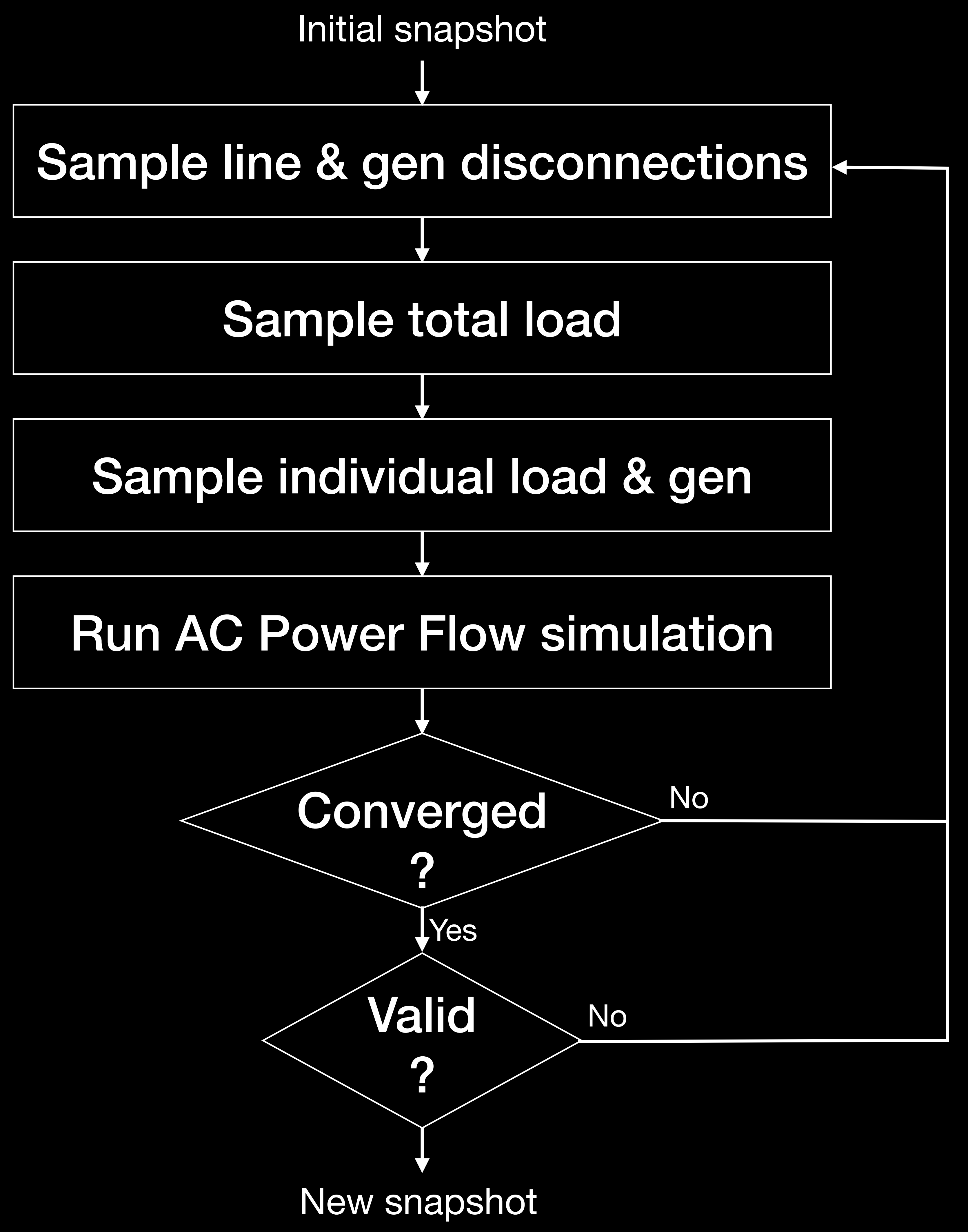
Artificial Data

Diagram



Artificial Data

Overview



github.com/bdonon/powerdatagen

Cétautomatix

Cétautomatix

OPF Solver

- Knitro (RTE License)
- 2 modes :
  - Constrained (if no feasible solution, solve the Relaxed version)
  - Relaxed (quadratic penalization of constraint violations)

Idéfix

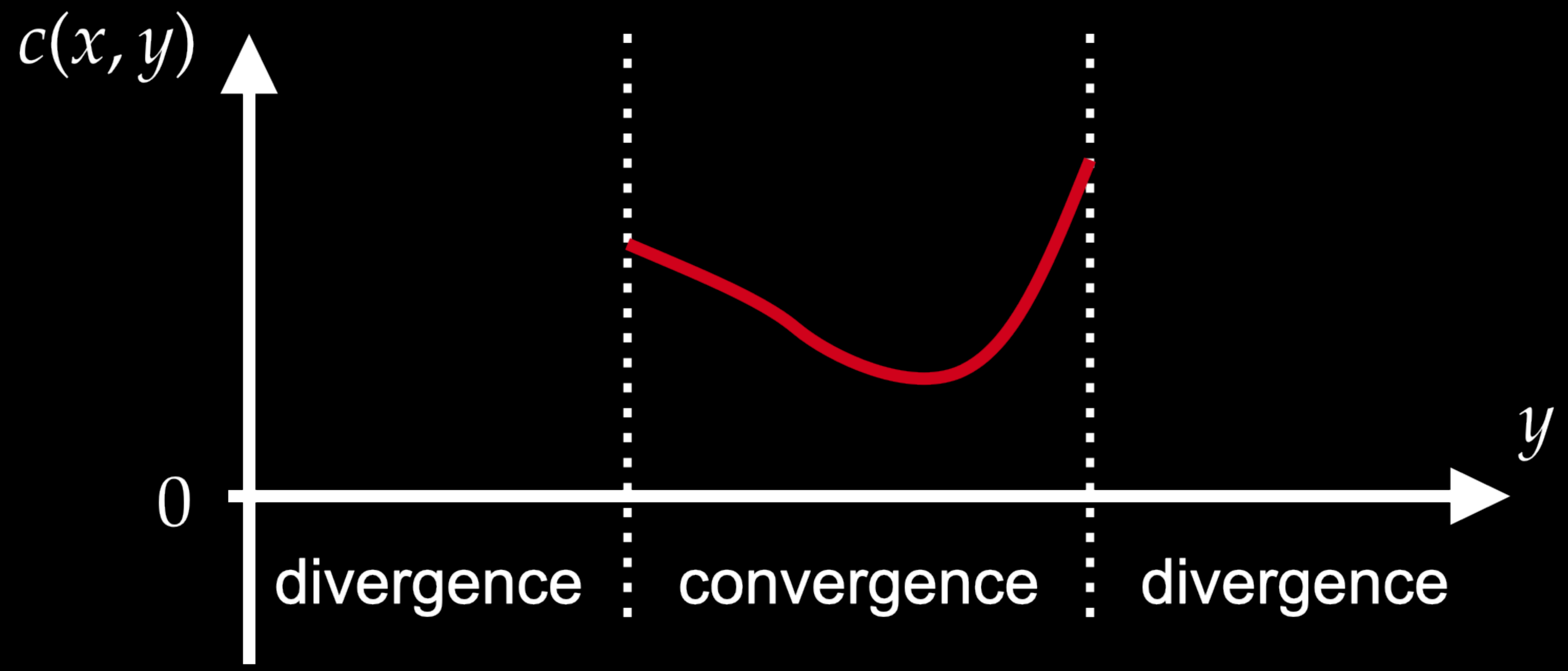
Idéfix

Early Experiments

- Only continuous action variables:
    - Generator Voltage Setpoints
- Reward : Joule losses + quadratic penalization of constraint violations.
    - Requires to run a power flow simulation.

Idéfix

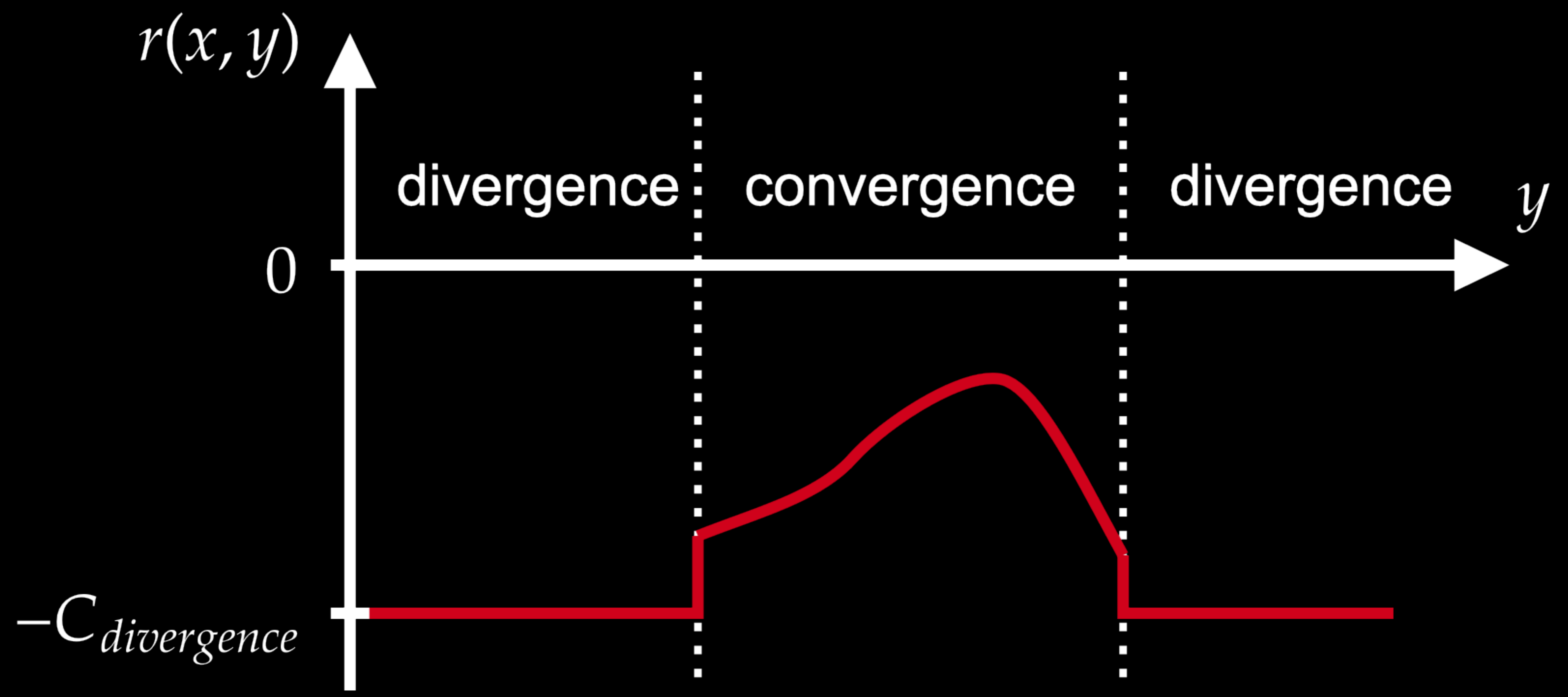
Reward Function



Idéfix

Reward Function - Opposite of Cost

$$r(x, y) = \begin{cases} -c(x, y) & \text{if } (x, y) \text{ converges} \\ -C_{\text{div}} & \text{otherwise} \end{cases}$$



Idéfix

Overall Idea

x - Power Grid Snapshot

Π_θ - Control Policy

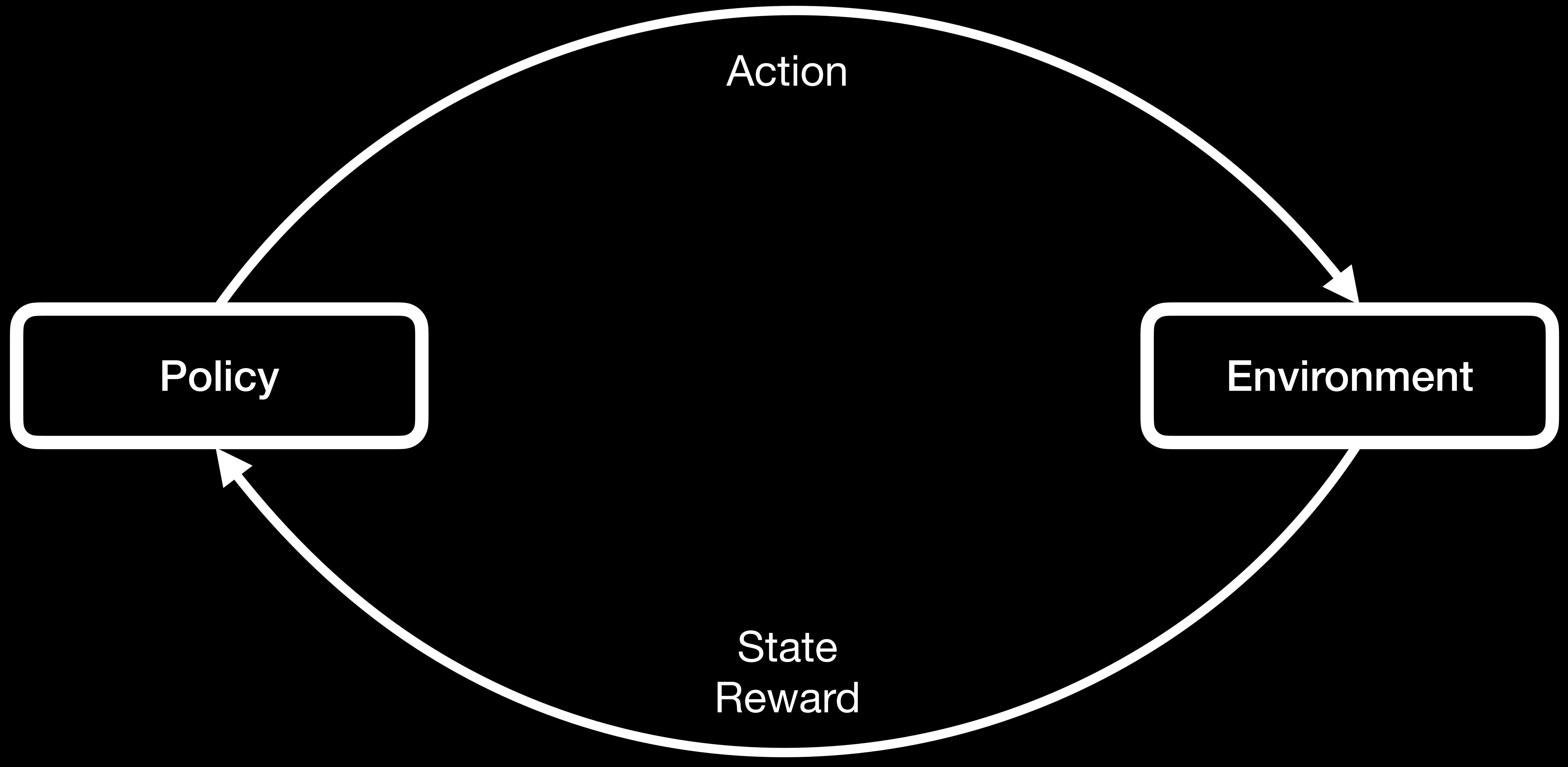
y - Action

AC Power Flow Simulator

r - Reward

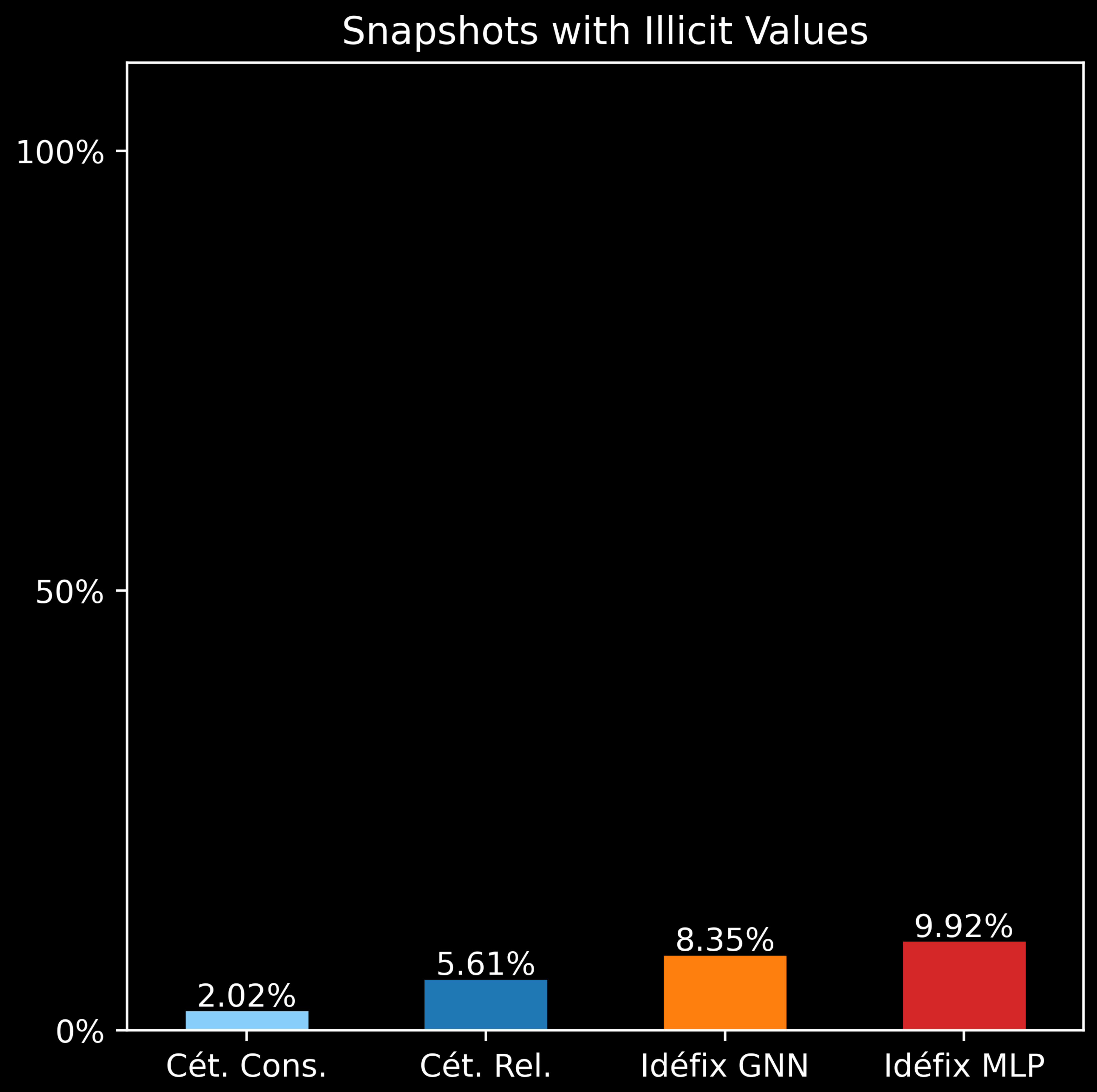
Idéfix

Overall Idea



Idéfix

Illicit Snapshots



Questions ?

Thank you !