End-to-End Feasible Optimization Proxies for Large-Scale Economic Dispatch

Ilay Menachem

January 2026

Motivation

- **Trend:** Deep Learning for grid management and optimization
- **Challenge:** Respecting problem structure and constraints
- **Benefit:** Networks that respect constraints learn more efficiently and achieve better performance

The paper's contribution

A new closed-form method to make neural networks respect economic dispatch constraints *by construction*

Previous Works: Optimization Proxies

Supervised Learning Approaches

- Successfully applied to DC-OPF and AC-OPF
- **Limitation:** Rely on fixed grid topologies
- Changes require expensive retraining and data re-generation

Graph Neural Networks

- Can handle topology changes
- **Limitation:** Shown to be unstable on large-scale networks

Previous Works: Ensuring Feasibility

Problem: ML proxies often violate physical constraints

Existing Strategies:

- **Active Set Learning:** Predict active constraints of the optimal solution (incorrect classifications $\rightarrow$ infeasibility)
- **Physics-Informed Models:** Add penalty terms for non-physical outputs (no feasibility guarantee)
- **Post-Processing:** Projection or power flow solvers to repair non-physical outputs (adds computational time)
- **End-to-End Layers:** Embed feasibility restoration (current methods fail to guarantee full feasibility or rely on restrictive assumptions)

Previous Works: Scalability Gap

Academic vs. Industrial Reality

- Most research: small test systems (< 300 buses)
- Real-world grids: tens of thousands of buses

The proposed approach

Test on large-scale systems up to 30,000 buses

The Economic Dispatch Problem

Objective

Minimize total production cost plus penalty for thermal limit violations:

$$\min_{\boldsymbol{p},\boldsymbol{r},\boldsymbol{\xi}_{th}} \; c(\boldsymbol{p}) + M_{th}||\boldsymbol{\xi}_{th}||_1$$

Subject to:

- $e^T \boldsymbol{p} = e^T \boldsymbol{d}$ (global power balance)
- $e^T \boldsymbol{r} \geq R$ (minimum reserve requirement)
- $\boldsymbol{p} + \boldsymbol{r} \leq \overline{\boldsymbol{p}}$ (generator output limits)
- $0 \leq \boldsymbol{p} \leq \overline{\boldsymbol{p}}$, $0 \leq \boldsymbol{r} \leq \overline{\boldsymbol{r}}$ (generator and reserve limits)
- $\underline{\boldsymbol{f}} - \boldsymbol{\xi}_{th} \leq \boldsymbol{\Phi}(\boldsymbol{p} - \boldsymbol{d}) \leq \overline{\boldsymbol{f}} + \boldsymbol{\xi}_{th}$ (thermal limits)
- $\boldsymbol{\xi}_{th} \geq 0$ (thermal limit violations non-negativity)

End-to-End Learning with Feasibility Guarantee

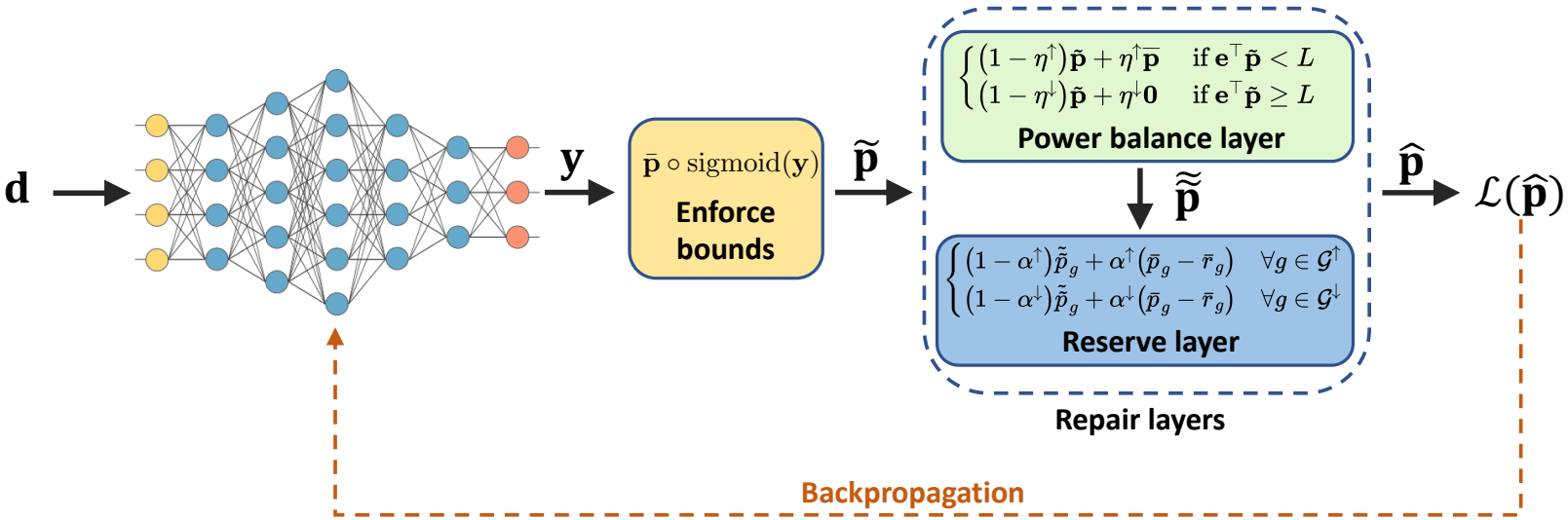
Key Idea: Add specialized layers that guarantee feasibility by construction

Architecture Components

1. Neural Network (predicts initial solution)
2. Sigmoid Layer (enforces generator limits)
3. Power Balance Repair Layer (enforces power balance)
4. Reserve Repair Layer (enforces reserve requirements)

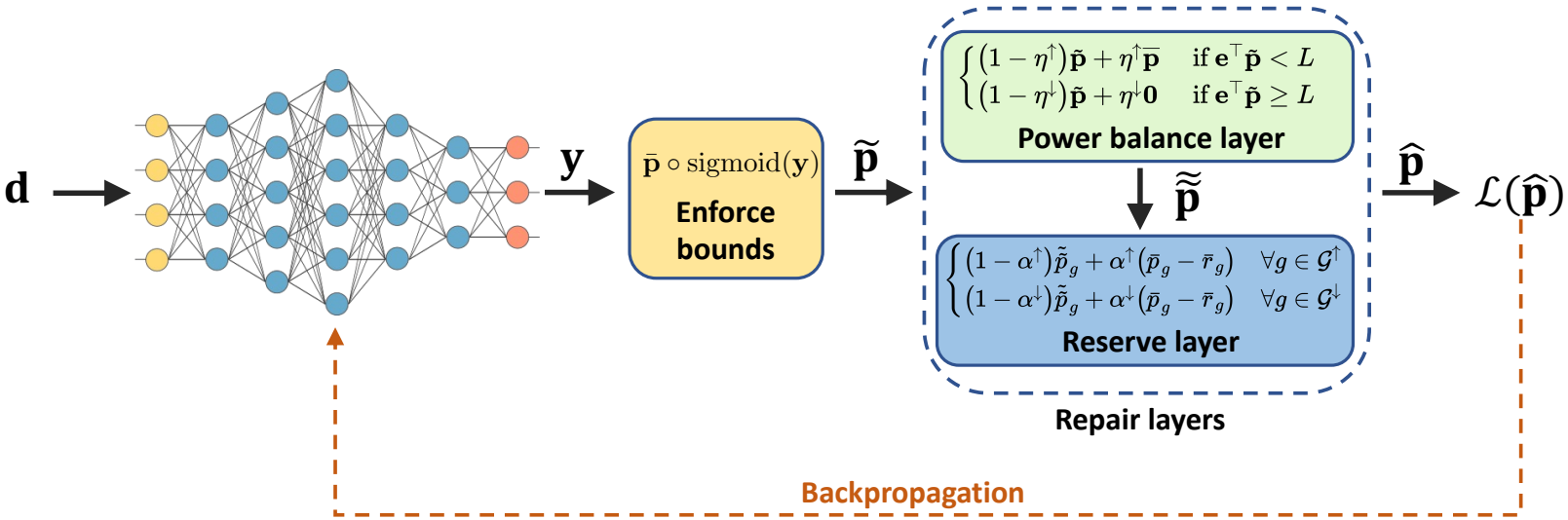
Guarantee

If a feasible solution exists, the repair layers will output a feasible solution



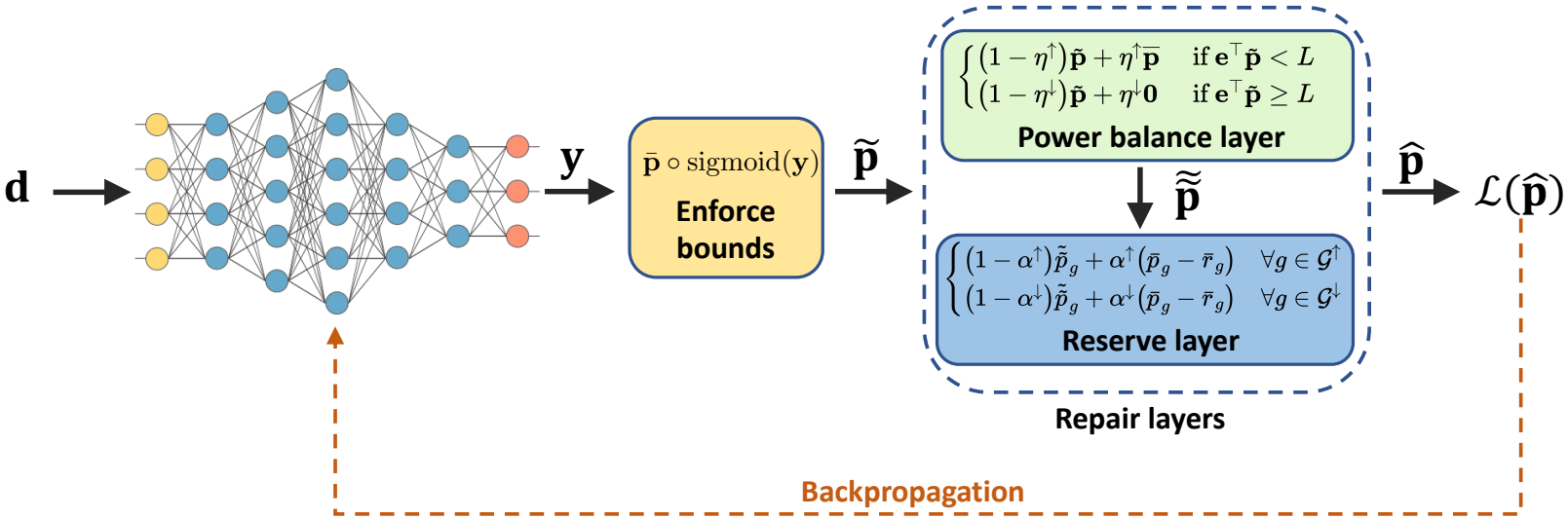
Purpose

The feed-forward architecture is used to decide the $\boldsymbol{p}$ given the data.



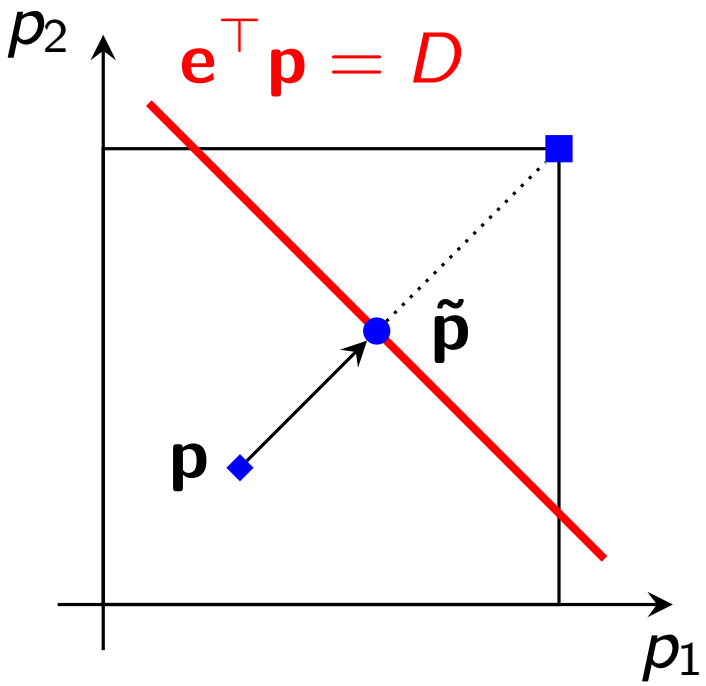
Purpose

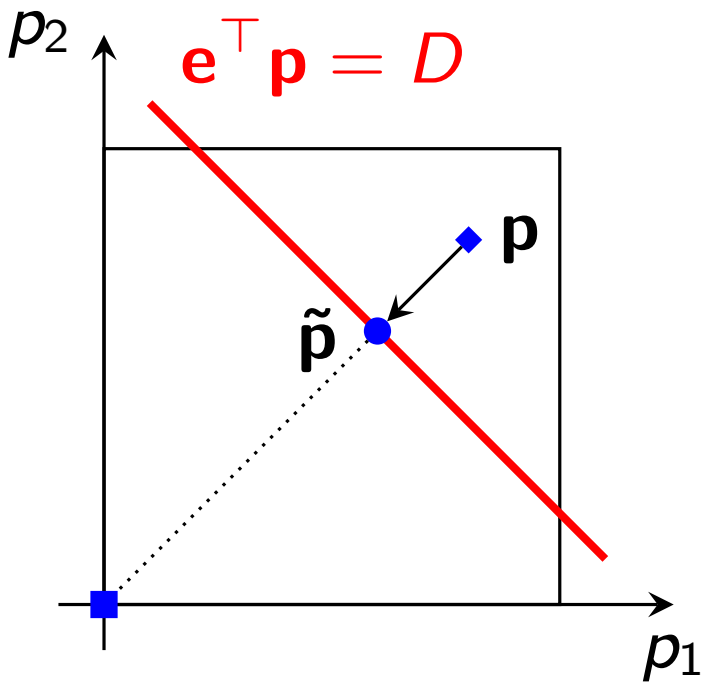
The sigmoid layer is used to enforce the generator limits $0 \leq \boldsymbol{p} \leq \overline{\boldsymbol{p}}$.

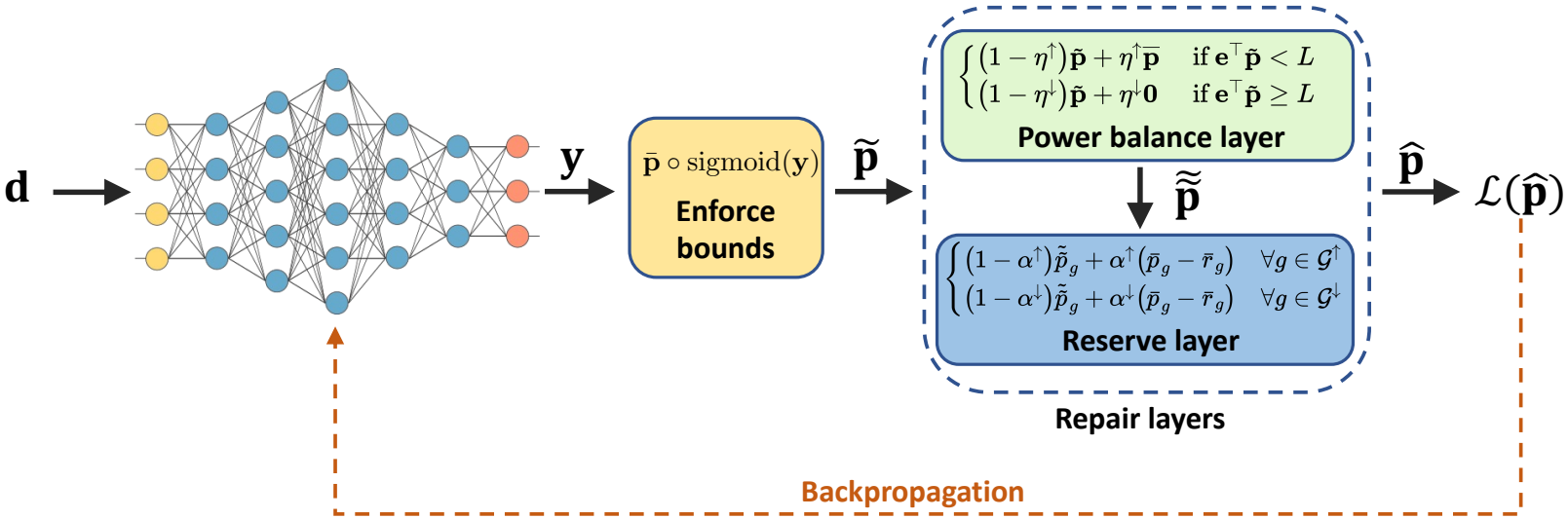


Purpose

Assuming that the generator limits are respected, the power balance layer is used to enforce the power balance constraint $e^T \boldsymbol{p} = e^T \boldsymbol{d}$.

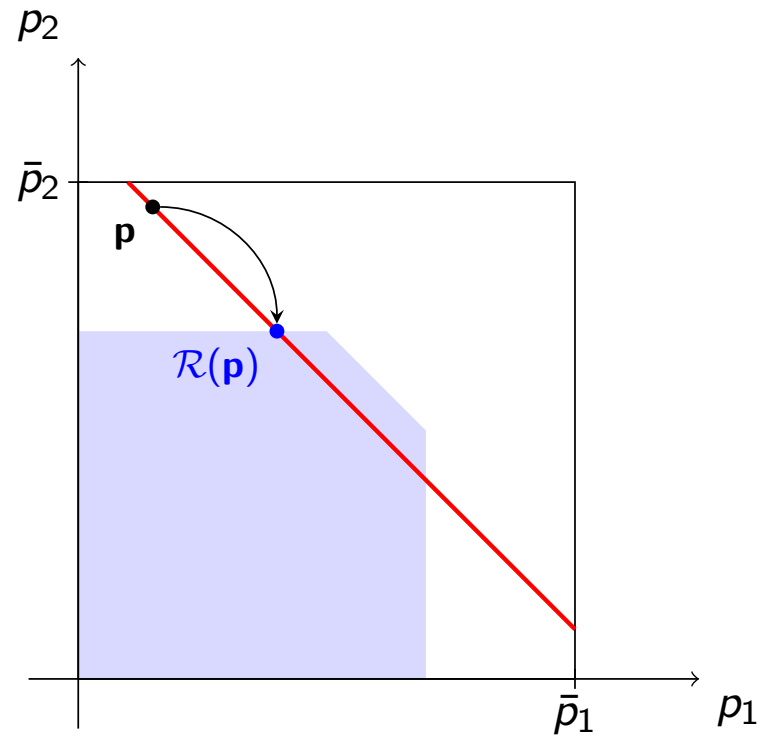

p_2
$\mathbf{e}^\top \mathbf{p} = D$
$\tilde{\mathbf{p}}$
$\mathbf{p}$
p_1


p_2
$\mathbf{e}^\top \mathbf{p} = D$
$\mathbf{p}$
$\tilde{\mathbf{p}}$
p_1



Purpose

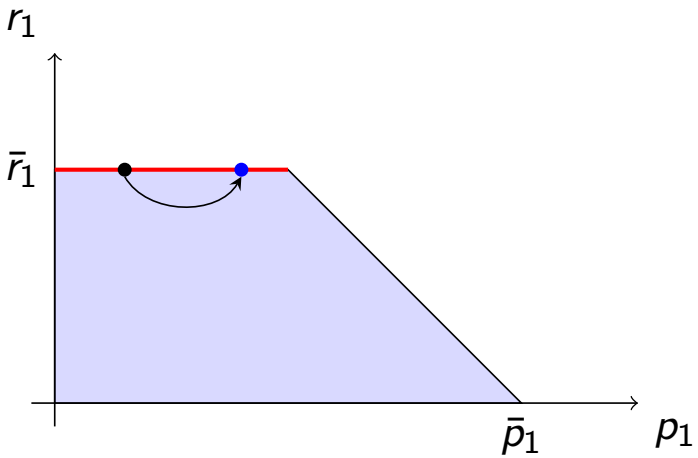
Assuming that the power balance and generator limits are respected, the reserve layer is used to enforce the reserve requirement $e^T \boldsymbol{r} \geq R$.

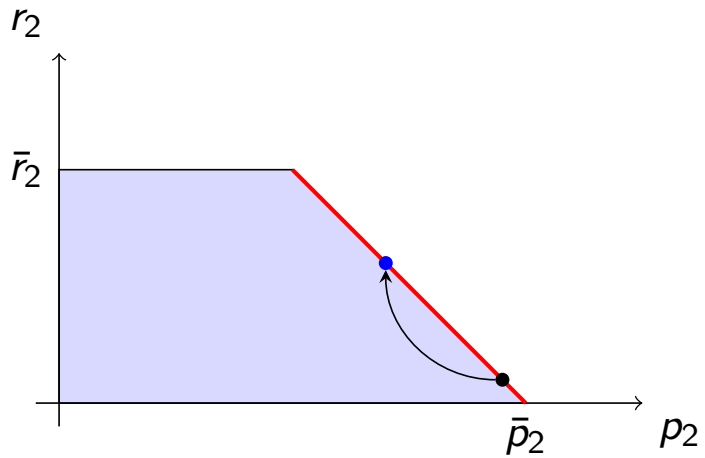


$\mathbf{0} \leq \mathbf{p} \leq \bar{\mathbf{p}}$

$\mathbf{e}^\top \mathbf{p} = D$

$\min\{\bar{r}_1, \bar{p}_1 - p_1\}$
$+ \min\{\bar{r}_2, \bar{p}_2 - p_2\} \geq R$





What we got?

what we got?

What we got?

what we got? as much one can ask for!

- ensure feasibility if there is a feasible solution
- informative gradients for the repair layers
- closed-form and simple $\rightarrow$ automatically differentiable
- closed-form and simple $\rightarrow$ fast

Datasets: Large-Scale Test Systems

System	Buses	Generators	Branches
ieee300	300	69	411
pegase1k	1,354	260	1,991
rte6470	6,470	761	9,005
pegase9k	9,241	1,445	16,049
pegase13k	13,659	4,092	20,467
goc30k	30,000	3,526	35,393

Datasets: Large-Scale Test Systems

System	Buses	Generators	Branches
ieee300	300	69	411
pegase1k	1,354	260	1,991
rte6470	6,470	761	9,005
pegase9k	9,241	1,445	16,049
pegase13k	13,659	4,092	20,467
goc30k	30,000	3,526	35,393

rte6470 is based on the full french grid!

Training: Supervised and Self-Supervised

Self-Supervised Learning

- Train to minimize objective function and constraint penalties directly
- Eliminates need for labeled data and offline optimization

Training: Supervised and Self-Supervised

Self-Supervised Learning

- Train to minimize objective function and constraint penalties directly
- Eliminates need for labeled data and offline optimization

Supervised Learning Loss

$$\mathcal{L}^{SL}(\hat{p}, p^*) = \underbrace{\frac{1}{|\mathcal{G}|}||\hat{p} - p^*||_1}_{\text{MAE}} + \underbrace{\mu M_{th}||\xi_{th}(\hat{p})||}_{\text{Thermal}} + \underbrace{\lambda\psi(\hat{p})}_{\text{Hard Constraints}}$$

Self-Supervised Learning Loss

$$\mathcal{L}^{SSL}(p^*) = \underbrace{c(p^*)}_{\text{Cost}} + \underbrace{M_{th}\xi_{th}(p^*)}_{\text{Thermal}} + \underbrace{\lambda\psi(\hat{p})}_{\text{Hard Constraints}}$$

- Hard constraints: $\psi(\hat{p}) = M_{pb}|e^T \boldsymbol{d} - e^T \hat{p}| + M_r \xi_r(\hat{p})$

Experimental Results

Key Findings

- **E2ELR model outperformed all baselines** on every dataset
- **Same inference time** as unconstrained networks

Impact

First method to guarantee full feasibility for large-scale economic dispatch while maintaining competitive speed and accuracy

End of Presentation

Thank you for your attention!

Questions?