# Solve Large-scale Unit Commitment Problems by Physics-informed Graph Learning

Dr. Nanpeng Yu
Full Professor and Vice Chair
Department of Electrical & Computer engineering
Director of Energy, Economics, and Environment Research Center
Email: nyu@ece.ucr.edu

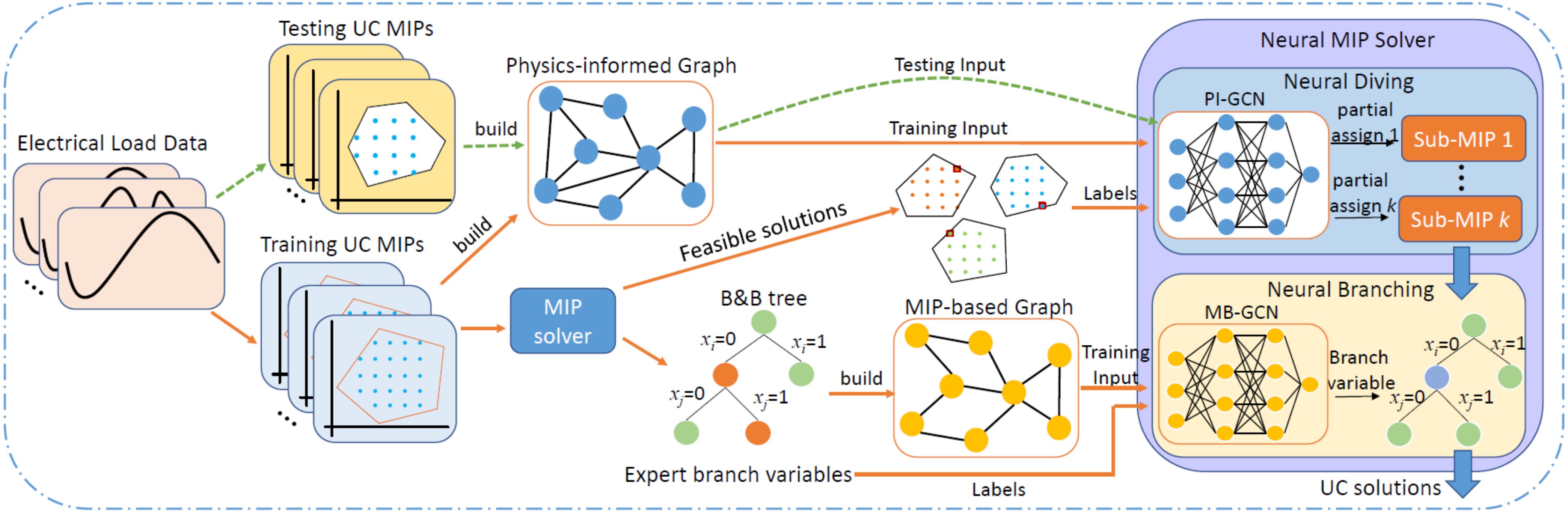
# Outline

- Motivation and Overall Framework
- Preliminaries
- MIP model-based GCN for Neural Diving and Branching
- Physics-informed GCN for Neural Diving
- Numerical Studies
- Conclusion and Future Work

# Motivation

- Increasing aggregated DERs & uncertainties in renewables made it difficult for ISOs to obtain high quality solutions to unit commitment (UC) problems in a timely manner.
- UC problems are formulated as mixed-integer programming (MIP). MIP solvers (e.g. Gurobi, CPLEX) play a key role in obtaining high-quality solutions.
- Traditional paradigm: on each day, the day-ahead UC task is treated as a brand new optimization problem.
- Insight: The previous UC problems' solutions provide useful information that can be used to improve the solution quality of similar UC problems.
- Recent advancements in Graph Neural Networks (GNN) make it a good tool to improve the performance of MIP solvers.

# Overall Framework[1]

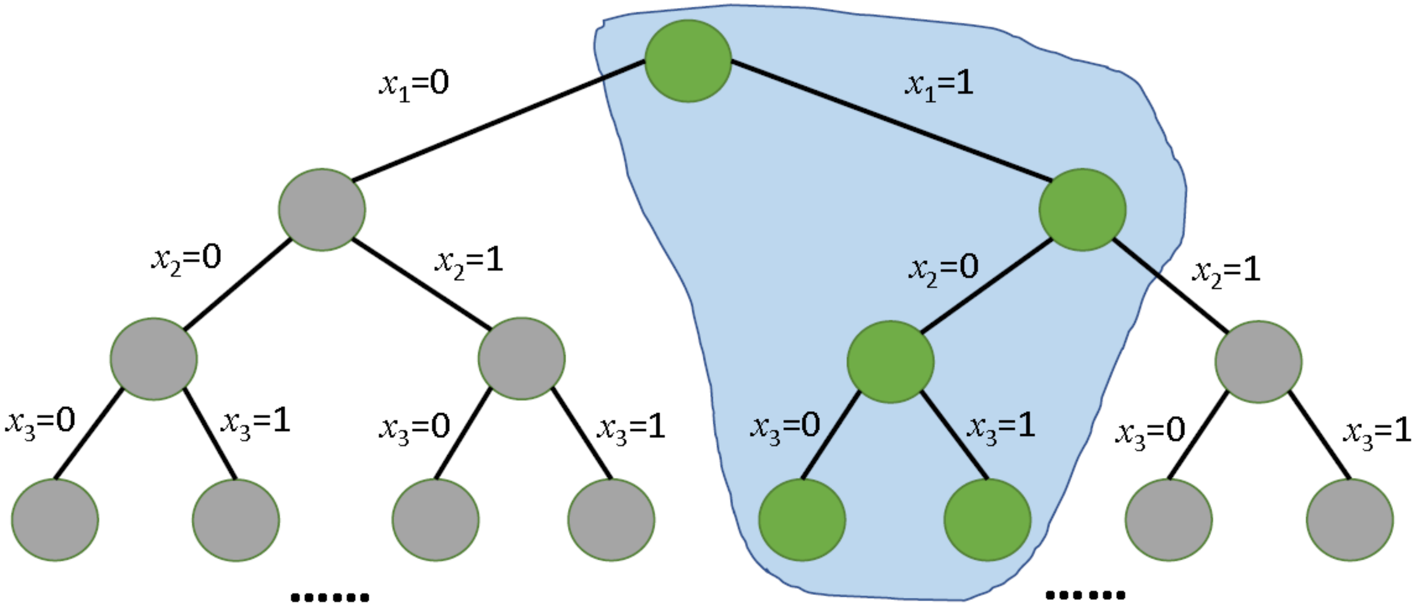



- Neural Diving: trained to find high-quality variable assignments
  - Input: physics-informed spatio-temporal graph
  - Output: decision variable assignments
  - Label: high-quality feasible solutions
- Neural Branching: trained to select the ideal variables for branching
  - Input: MIP-based bipartite graph
  - Output: branching scores of variables on each node
  - Label: expert branching variables (e.g. strong branching)

[1] Jingtao Qin and Nanpeng Yu, "Solve Large-scale Unit Commitment Problems by Physics-informed Graph Learning, " https://arxiv.org/pdf/2311.15216.pdf.

# Preliminaries: Diving

- Diving is a set of primal heuristics of exploring the branch-and-bound tree in a depth-first manner.
- It involves sequentially fixing integer variables until either a leaf node is reached or the linear programming (LP) problem becomes infeasible.
- The diving approach in our work is performed only on the root node.



# Preliminaries: Strong Branching

- Brief Introduction to Strong Branching
  - The measurement for the quality of branching on a variable is the improvement of the dual bound.
  - The strong branching (SB) score of branching on variable $x_i$ can be calculated as:

$$SB_i = max\{\hat{z}_i^- - \hat{z}, \varepsilon\} \times max\{\hat{z}_i^+ - \hat{z}, \varepsilon\}$$

$\hat{z}$: LP value of the node

$\hat{z}_i^-$ , $\hat{z}_i^+$: LP values of children nodes

$\varepsilon$: a small constant (e.g. $10^{-6}$)

  - Pros: Can identify critical variables that have a significant improvement of the dual bound.
  - Cons: Computationally expensive

# MIP model-based GCN for Neural Diving & Branching

## ❑ Graph Definition & Features

Formulate a Generic MIP as a Bipartite Graph[1]

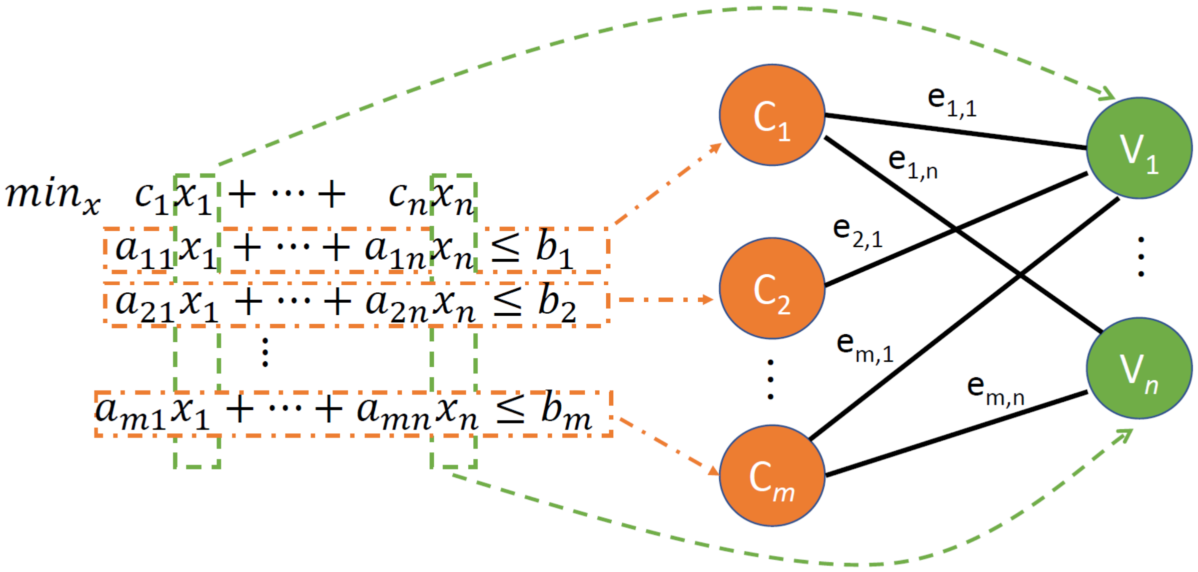



Table: Features of different nodes in the bipartite graph

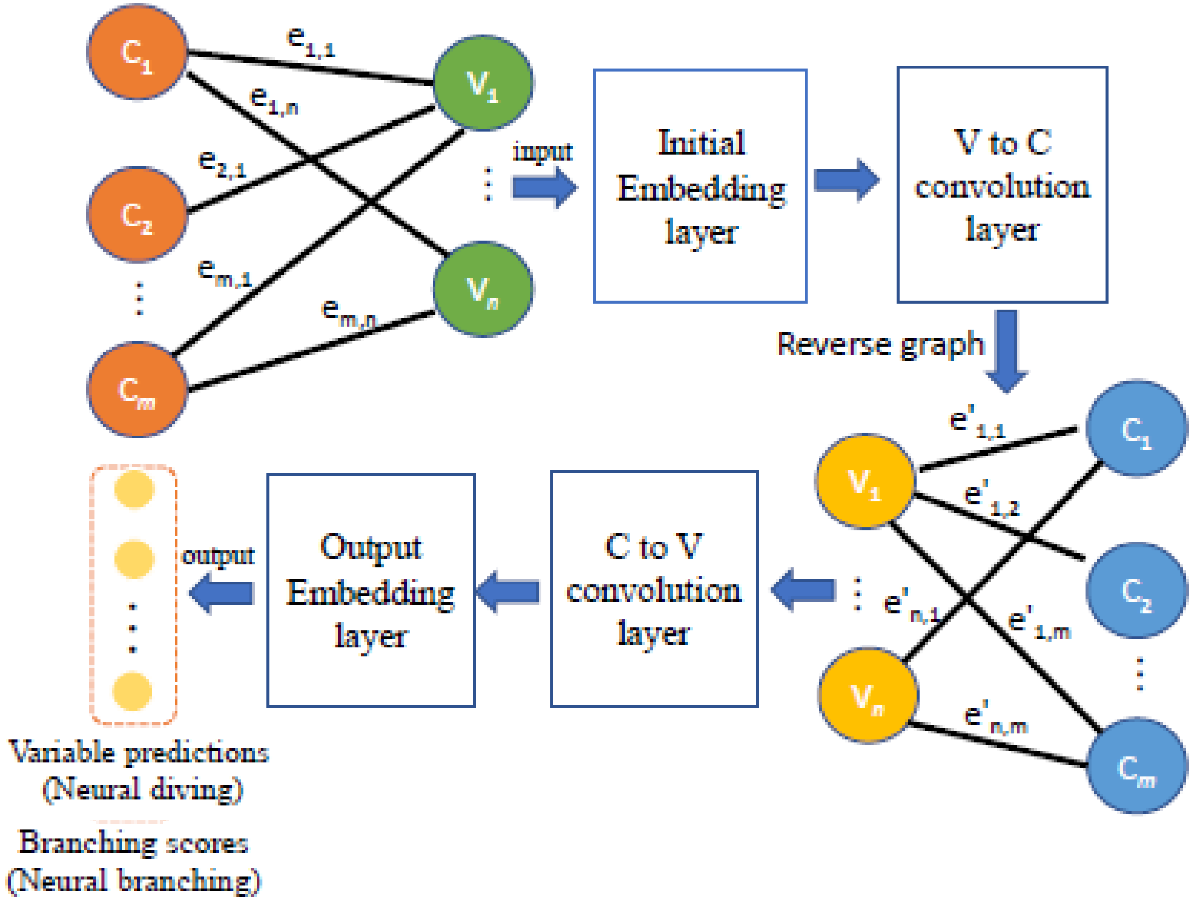
| Set | Feature | Description |
|---|---|---|
| C | obj_cos_sim | Cosine similarity with objective |
| | bias | Bias value, normalized with constraint coefficients |
| | is_tight | Tightness indicator in LP solution |
| | dual_sol_val | Dual solution value, normalized |
| | age | LP age, normalized with the total number of LPs |
| E | coef | Constraint coefficient, normalized per constraint |
| V | type | Type (binary, integer, impl. integer, continuous) as a one-hot encoding. |
| | coef | Objective coefficient, normalized |
| | has_lb | Lower bound indicator |
| | has_ub | Upper bound indicator |
| | sol_is_at_lb | Solution value equals lower bound |
| | sol_is_at_ub | Solution value equals upper bound |
| | sol_frac | Solution value fractionality |
| | basis_status | Simplex basis status (lower, basic, upper, zero) as a one-hot encoding |
| | reduced_cost | Reduced cost, normalized |
| | age | LP age, normalized |
| | sol_val | Solution value |
| | inc_val | Value in incumbent |
| | avg_inc_val | Average value in incumbents |

- ❑ The bipartite graph is defined as: $\mathcal{G} = (C, E, V)$
  - ✓ $C$: the set of constraint nodes. $E$: the set of edges. $V$: the set of variable nodes

[1] Deep Mind, Google Research "Solving mixed integer programs using neural networks." arXiv preprint arXiv:2012.13349 (2020).

# MIP model-based GCN for Neural Diving & Branching

### Network Architecture of Baseline Model




- Two graph convolution layers have the same network structure but are initialized with different weights.

- For V to C convolution layer, nodal information are propagated using the operator below:

$$x_i' = W_C x_i + \sum_{j \in \mathcal{N}(i)} W_V x_j + \sum_{j \in \mathcal{N}(i)} W_E e_{i,j}$$

$W_C$, $W_V$, and $W_E$ are weight matrices

$\mathcal{N}(i)$: the set of indices of the neighbor nodes of node $i$.

# MIP model-based GCN for Neural Diving & Branching

- **Loss Function for Neural Diving**: the closer the predictions of binary variables to true values, the smaller the loss

$$L(\theta) = -\sum_{i=1}^{n}\sum_{j=1}^{m_i} \omega_{ij} log p_{\theta}(x^{i,j}|\mathbb{G}_i)$$

sum over # of graphs $n$

sum over # of feasible solutions by solving each graph instance

weights used to reduce the sample bias

$$p_{\theta}(x|\mathcal{G}) = \prod_{d\in D} p_{\theta}(x_d|\mathcal{G})$$

assume all integer variables are independent

✓ Use the acquired probability distribution $p_{\theta}(x|\mathcal{G})$ to stochastically set the value of binary variables to 0 or 1, and then solve the remaining sub-MIP with a MIP solver.

- **Loss Function for Neural Branching**: maximize output scores of the binary variables selected by strong branching

Cross-entropy function loss $$L(\theta) = -\log\left(\frac{\exp(y_c)}{\sum_{i\in V_I}\exp(y_i)}\right)$$

$y$ is the output vector of the graph neural network

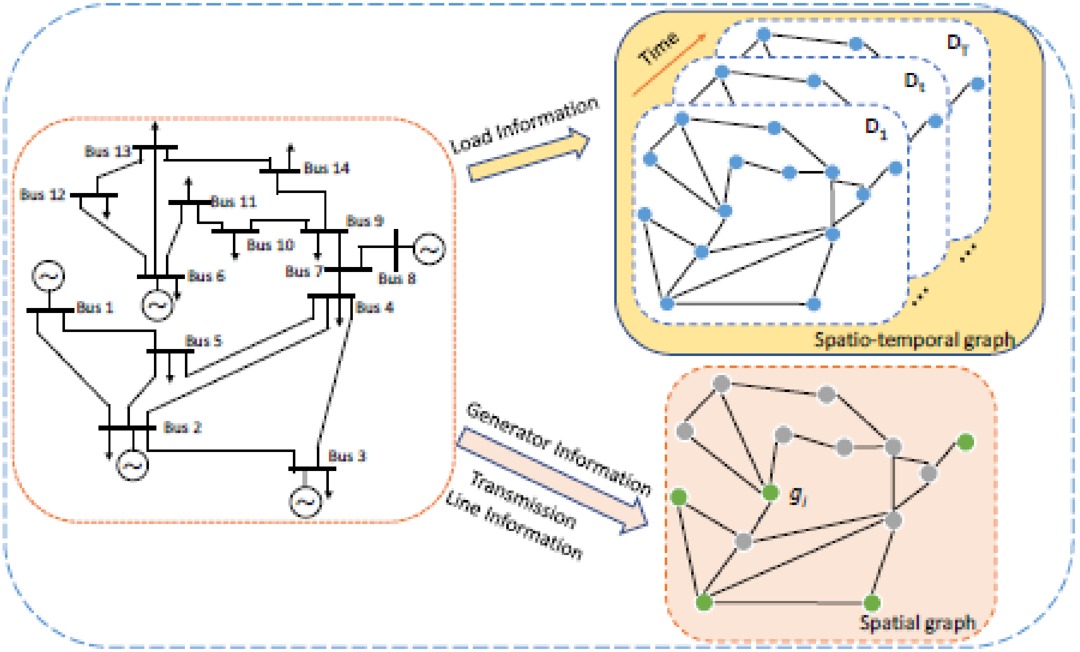
$c$ is the index of the candidate variable selected by strong branching

$V_I$ is the set of binary variables.

# Physics-informed GCN for Neural Diving

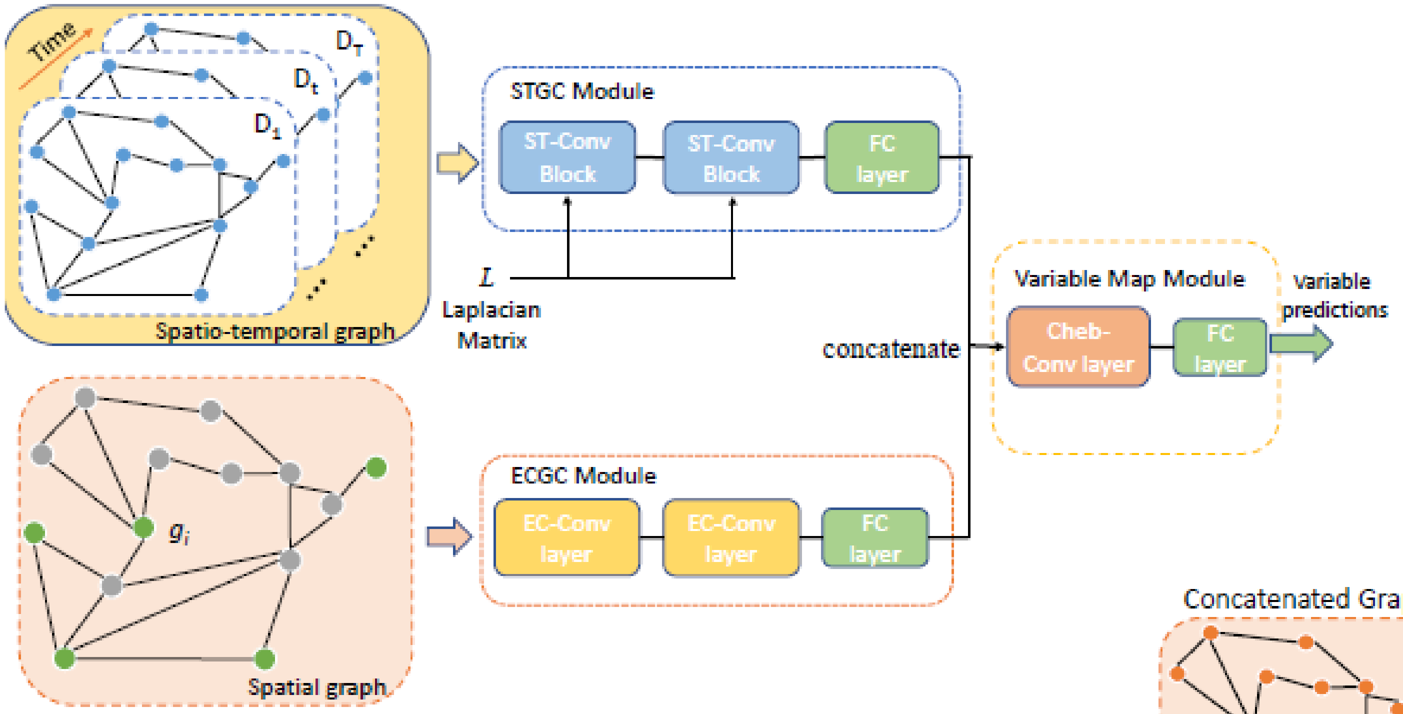
- Motivation for Using Physics-informed Graph
  - The baseline model is designed for a generic MIP. UC problems are MIP on a physical network.
  - When dealing with large-scale UC problems, the bipartite graph built from MIP will be massive and computationally prohibitive to implement (see numerical study results).
  - Physical information and constraints on power network can be used to speed up training and improve performance.

Physics-informed Graphs for UC Problems



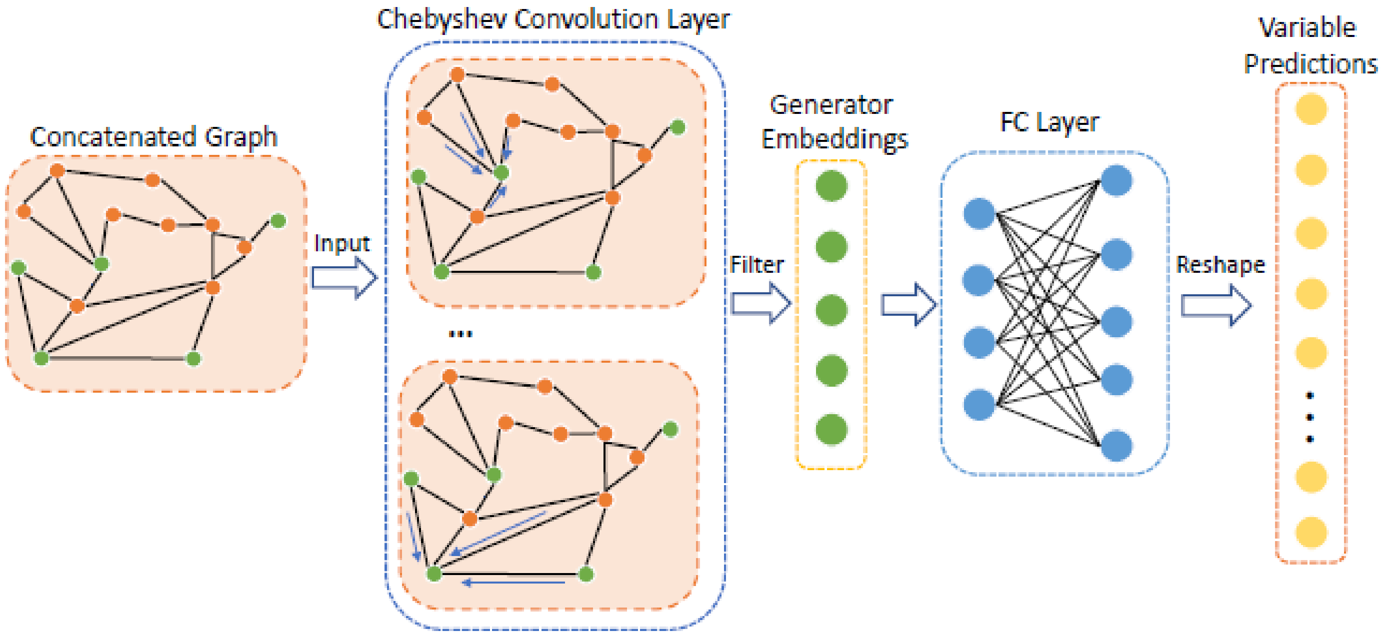

- Construction of Physics-informed Graph for UC Problems
  - Spatial Graph: Captures static information
    - ✓ Generator features (e.g., Pmin, Pmax, ramp rate, heat rates),
    - ✓ Transmission line features (e.g., impedance, thermal limits).
  - Spatio-temporal Graph: Embeds variables that change with location and time
    - ✓ Electric load
    - ✓ Renewable generation

# Physics-informed GCN for Neural Diving




**Network Architecture of Physics-informed GCN**

## Variable Mapping Module



# Numerical Studies

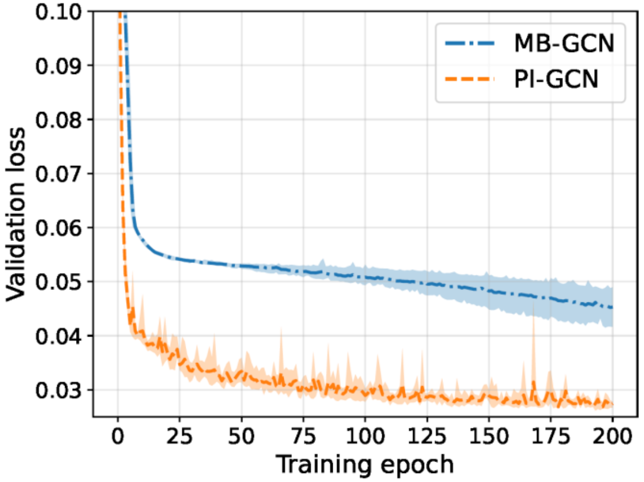
- Optimization horizon: 24-hours
- Test Systems: IEEE 1354-bus, 2383-bus, and 6515-bus systems
- Historical Load Dataset from CAISO for Neural Diving and Branching
  - ✓ Neural Diving: 07/01/2017 – 10/22/2022. Training (1000 days), validation (100 days), testing (100 days)
  - ✓ Neural Branching: 8/16/2021 – 09/03/2021. Training (100 days), validation (20 days), testing (20 days)
- State-of-the-art UC problem formulation: state transition model[1] with tight formulation[2].
- MIP Solvers: SCIP and Gurobi (30 minutes of execution time).

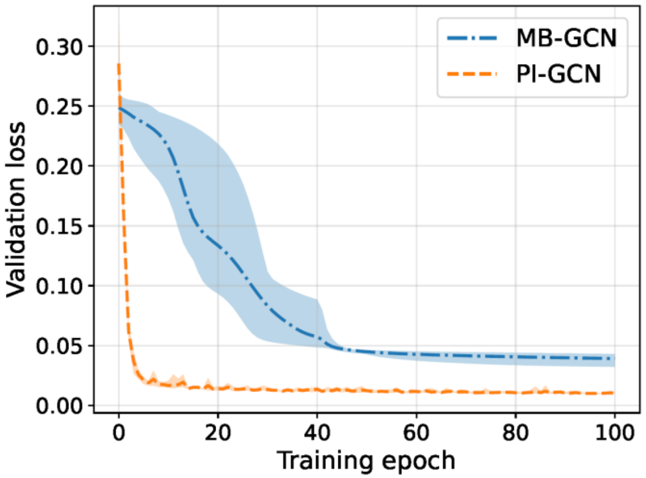
[1] Atakan Semih, Guglielmo Lulli, and Suvrajeet Sen, "A state transition MIP formulation for the unit commitment problem." *IEEE Transactions on Power Systems*, vol. 33, no. 1 (2017): 736-748.

[2] B. Knueven, J. Ostrowski, and J.-P. Watson, "On mixed integer programming formulations for the unit commitment problem," *INFORMS Journal on Computing*, vol. 32, no. 4, pp. 857–876, 2020.

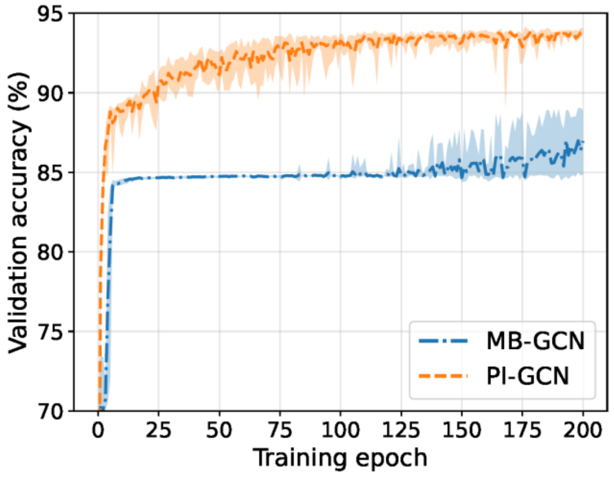
# Numerical Studies: Neural Diving

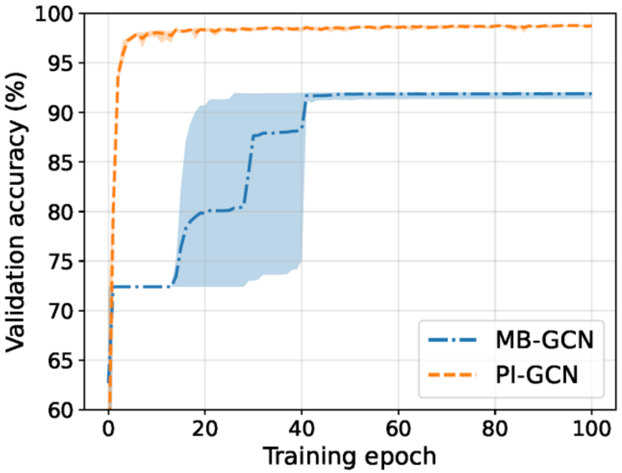
Validation Loss for Neural Diving Model: 1354- and 2383-bus systems




Validation Accuracy for Neural Diving Model: 1354- and 2383-bus systems



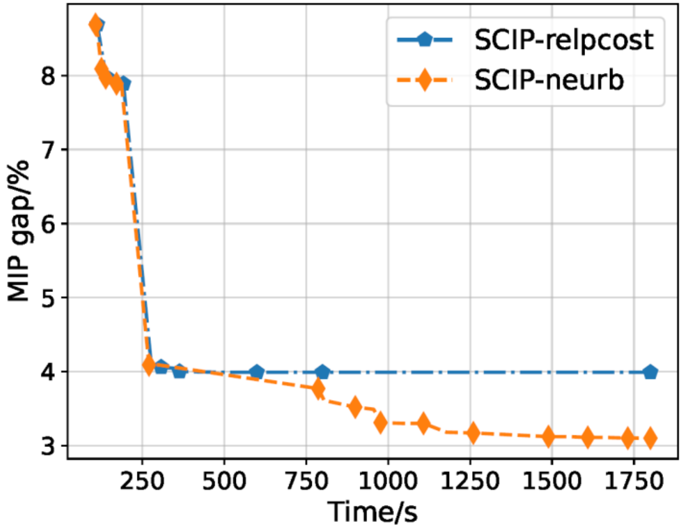

**Number of Binary Variables in Different Accuracy Intervals for Testing Dataset**

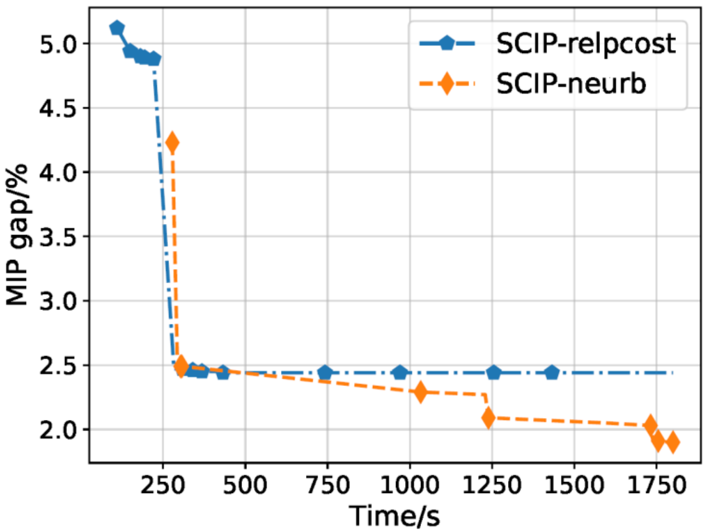
| Interval | 1354-bus | | 2383-bus | |
|---|---|---|---|---|
| | MB | PI | MB | PI |
| ≥ 80% | 4,582 | 4,846 | 7,031 | 7,598 |
| ≥ 90% | 4,253 | 4,533 | 6,744 | 7,176 |
| = 100% | 3,554 | 3,692 | 6,061 | 6,286 |
| Total | 6,240 | 6,240 | 7,656 | 7,656 |

- PI-GCN converges to a lower loss than MB-GCN
- PI-GCN achieves higher validation and testing accuracy in terms of binary decision variables prediction.

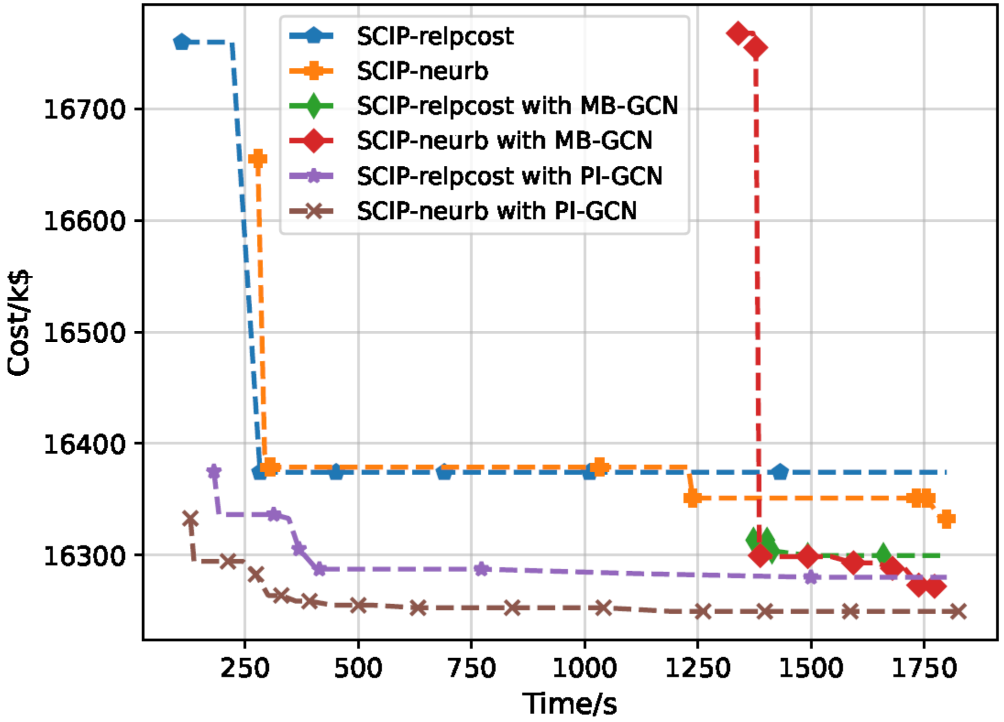
# Numerical Studies: Neural Branching & Diving

**MIP Gap Versus Solving Time for 1354- and 2383-bus system: Branching Only**




**MIP Gap Versus Solving Time 2383-bus system: Branching + Diving**



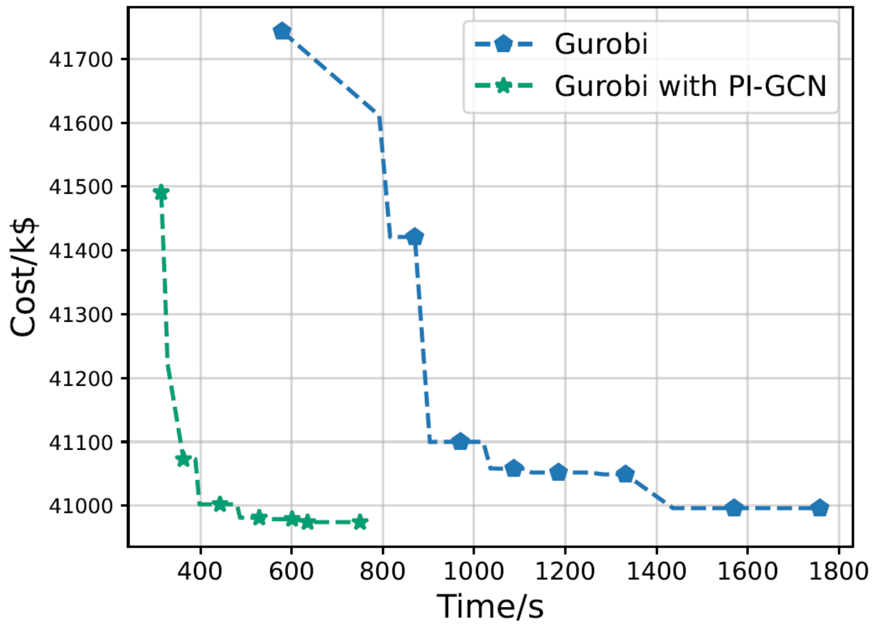

- Neural Branching helps reduce MIP Gap between 0.6% to 0.9% for the 1354- and 2383-bus system.
- Integrating neural branching with physics-informed neural diving achieves the best result on 2383-bus system.
- Within 250 seconds, our proposed method is able to find a very good feasible solution that can not be discovered by other methods in 30 minutes.

# Numerical Studies: Scalability

**GPU Memory Requirement of Different Approaches: Physics informed (PI) versus MIP Model-based (MB)**

| System | Max node input | | Max edge input | | Max allocated GPU memory (GB) | |
|---|---|---|---|---|---|---|
| | MB | PI | MB | PI | MB | PI |
| 1354-bus | 137,621 | 1,354 | 7,452,282 | 1,991 | 7.04 | 0.36 |
| 1888-bus | 155,303 | 1,888 | 13,030,995 | 2,531 | 12.36 | 0.57 |
| 2382-bus | 209,535 | 2,233 | 42,758,766 | 2,896 | 15.92 | 1.21 |
| 3012-bus | 216,110 | 3,012 | 52,104,887 | 3,572 | 19.59 | 1.52 |

**Operational cost versus solving time for a sample day on IEEE 6516-bus system**




- ❑ Our proposed approach reduces the maximum mode and edge inputs by at least 3 orders of magnitude and requires substantially less GPU memory.
- ❑ Our proposed approach (Gurobi + PI-GCN) significantly outperforms commercial MIP solver Gurobi by consistently maintaining lower operational costs in significantly less time.

# Conclusion & Future Work

- Developed a physics-informed GCN for neural diving that is trained to find high-quality variable assignments for unit commitment problems.
- Designed a hierarchical GCN to handle heterogeneous inputs and map inputs of UC problems on a graph to outputs at the decision variable level.
- Numerical studies show that our proposed physics-informed neural diving and neural branching easily beats commercial MIP solver (e.g. SCIP and Gurobi).
- The scalability of the proposed model is significantly enhanced by reducing graph size and the dependence on computing resources.
- Consider uncertainties of load and renewable energy outputs.
- Model contingencies and extend to AC power flow.

# Contact Information

**Nanpeng Yu. Email: nyu@ece.ucr.edu**

# Thank You

# Questions?

PhD Student: Jingtao Qin

Funding Support for the Project: DOE, NSF, CEC, UCOP.