Attention-Enhanced Graph Convolutional Network for Multi-Period Optimal Power Flow

Liheng Lyu
Department of Electrical and Computer Engineering
Missouri University of Science & Technology
Rolla, MO, USA
llfgr@mst.edu

Rui Bo
Department of Electrical and Computer Engineering
Missouri University of Science & Technology
Rolla, MO, USA
rbo@mst.edu

***Abstract*—The power system optimal power flow (OPF) determines generator dispatch that minimizes generation cost while satisfying load demands and operating limits. Recent machine-learning methods approximate DC-OPF for faster solutions; however, their accuracy remains limited, especially under line congestion and temporal coupling across consecutive periods. Moreover, most existing studies do not estimate locational marginal prices (LMPs). This paper proposes an attention-enhanced graph convolutional network for multi-period OPF (MP-OPF). The GNN transforms nodal demand into node features that encode both load distribution and grid topological dependencies, and the attention mechanism models temporal dependencies to jointly estimate generator outputs, congestion, and LMPs. The proposed model is evaluated on both the IEEE 118-bus system and its modified version with real MISO load data. The results show that the proposed method achieves high accuracy in estimating MP-OPF and LMP outcomes with orders of magnitude speedup, demonstrating its potential for both operation and planning applications.**

***Index Terms*—Deep learning, graph neural network, optimal power flow, locational marginal pricing, electricity market.**

I. Introduction

As electricity demand and intermittent energy resources continue to grow, power systems face increasing uncertainty and challenges in maintaining secure and economic operation. The optimal power flow (OPF) problem, which determines the most economical generator dispatch that meets load demand while satisfying network and operational constraints is an essential tool to ensure secure and cost-effective operation [1]. Among its variants, the direct current OPF (DC-OPF), with lower computational complexity and better tractability, is widely used in industry [2].

Under time-varying load conditions, generator schedules must satisfy temporal constraints such as ramping limits, giving rise to the multi-period OPF (MP-OPF) problem [3], [4]. Load fluctuations also alter power flows and may drive lines toward congestion, increasing operational costs and causing non-smooth variations in locational marginal prices (LMPs). These spatio-temporal interactions highlight the need for accurate joint estimation of OPF outcomes and LMPs.

For OPF, the existing methods can be divided into two categories: optimization-based and learning-based. The first category formulates the OPF as a mathematical optimization problem—often using linear programming—subject to power balance equality and network operational constraints [5]–[7]. Its computational complexity increases significantly with system size and the number of study scenarios, making it challenging for real-time DC-OPF applications [8].

In contrast, the second category employs deep neural networks (DNNs) to directly learn the mapping from load demand to generator dispatch, achieving near-optimal solutions with reduced computation time. For instance, in [9], a DNN is trained to map load demand to generator power and voltage, achieving faster computation with acceptable accuracy. To reduce computational burden and improve estimation accuracy, convolutional neural networks (CNNs) are employed in [10] to learn the mapping between power injections and optimal objectives. Subsequently, graph neural networks (GNNs) [11], [12] are introduced to incorporate grid topology and capture spatial correlations, further improving prediction performance.

Despite these advances, most existing studies focus on single-period OPF and neglect temporal coupling, which degrades estimation accuracy. In [13], a long short-term memory (LSTM) network is introduced to model multi-period time-series data; however, LSTM-based methods suffer from gradient vanishing and fail to represent the long-term operational patterns. Moreover, in OPF, LMP is a byproduct of optimization, while congestion directly affects both the feasible region of generator dispatch and LMP. Generator dispatch, congestion, and LMP are strongly coupled, yet most prior works focus only on estimating generator dispatch, ignoring this interdependence and thus limiting accuracy.

To address these limitations, we propose an attention-enhanced graph convolutional network (AE-GCN) in Section II, which captures spatio-temporal dependencies across multiple periods and jointly predicts generator dispatch, congestion, and LMP through multi-task learning. Section III evaluates the proposed method, demonstrating high accuracy and Section IV concludes the paper.

II. Methodology

A. Problem Formulation

Let $\mathcal{I} = \{1, 2, \ldots, I\}$ denote the set of buses, and $\mathcal{T} = \{1, 2, \ldots, T\}$ the set of time steps. Let $\mathcal{D} = \{1, \ldots, D\} \subseteq \mathcal{I}$

This work was supported by the National Science Foundation under Award No. 2339956.

979-8-3315-8138-1

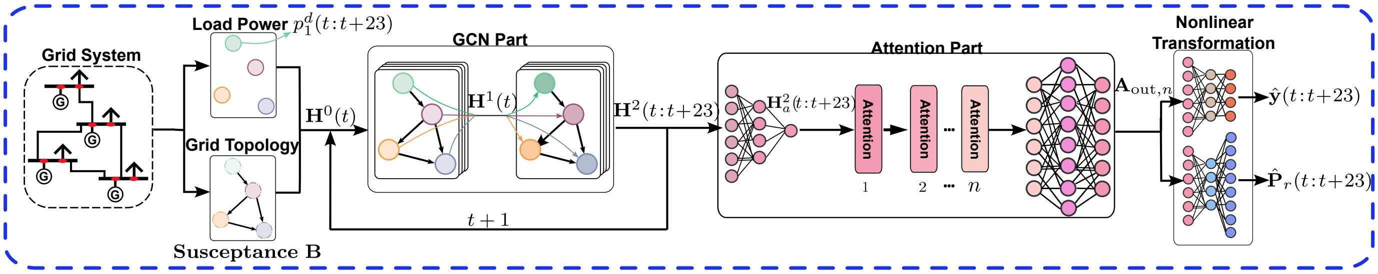

Fig. 1. Key components and data flows of the AE-GCN architecture for the case $I = 4$

and $\mathcal{G} = \{1, \ldots, G\} \subseteq \mathcal{I}$ represent the subsets of load and generator buses, respectively. Denote by $\mathcal{L} = \{\, l_{i,j} \mid (i,j) \in \mathcal{I} \times \mathcal{I} \,\}$ the set of transmission lines connecting bus i and j.

For each load bus $i \in \mathcal{D}$, the demand profile over time is $\mathbf{p}_i^d = [p_i^d(1), p_i^d(2), \ldots, p_i^d(T)]$, and for each generator bus $i \in \mathcal{G}$, the generation output profile is $\mathbf{p}_i^g = [p_i^g(1), p_i^g(2), \ldots, p_i^g(T)]$.

Then, the multi-period optimal power flow (MP-OPF) problem can be formulated as

$$\min_{\{p_i^g(t),\, \theta_i(t)\}} \sum_{t=1}^{T} \sum_{i \in \mathcal{G}} \left(c_{2,i} (p_i^g(t))^2 + c_{1,i} p_i^g(t) + c_{0,i} \right), \quad (1)$$

subject to

$$p_i^g(t) - p_i^d(t) = \sum_{j \in \mathcal{U}_i} B_{ij} \big(\theta_i(t) - \theta_j(t) \big), \quad \forall i \in \mathcal{I}, \forall t \in \mathcal{T}, \quad (2)$$

$$p_i^{g,\min} \le p_i^g(t) \le p_i^{g,\max}, \quad \forall i \in \mathcal{G}, \forall t \in \mathcal{T}, \quad (3)$$

$$-f_{ij}^{\max} \le B_{ij} \big(\theta_i(t) - \theta_j(t) \big) \le f_{ij}^{\max}, \quad \forall (i,j) \in \mathcal{L}, \forall t \in \mathcal{T}, \quad (4)$$

$$-r_i^- \le p_i^g(t+1) - p_i^g(t) \le r_i^+, \quad \forall i \in \mathcal{G}, t = 1, \ldots, T-1. \quad (5)$$

where $c_{2,i}$, $c_{1,i}$, and $c_{0,i}$ denote the quadratic, linear, and constant cost coefficients of generator i, respectively. θ_i represents the voltage phase angle at bus i, and $\mathcal{U}_i$ denotes the set of neighboring buses connected to bus i. B_{ij} is the susceptance of line (i,j), and $f_{ij}^{\max}$ is its thermal limit. r_i^- and r_i^+ denote the downward and upward ramping limits of generator i, respectively. Without loss of generality, bus 1 is selected as the reference bus, and its voltage angle is fixed to $\theta_1(t) = 0$ for all $t \in \mathcal{T}$.

B. Architecture of AE-GCN

The proposed AE-GCN architecture is illustrated in Fig. 1. It consists of three parts: a GCN for extracting load features and capturing the spatial information of the grid topology, an attention mechanism to model spatial correlations across multiple periods, and two non-linear transformation layers for estimating generator dispatch, congestion, and LMPs.

To address long-term variability, the one-year dataset is partitioned into daily sequences, where each sequence corresponds to a 24-hour load vector. For each bus $i \in \mathcal{I}$, the one-year load profile can be represented as $\mathbf{p}_i^d = \big\{ \tilde{\mathbf{p}}_i^d(t : t{+}23) \;\big|\; t = 1, 25, 49, \ldots, 8760 \big\}$, where each 24-hour load vector is defined as

$$\tilde{\mathbf{p}}_i^d(t : t{+}23) = \begin{cases} [\, p_i^d(t), \, p_i^d(t{+}1), \, \ldots, \, p_i^d(t{+}23) \,]^\top, & i \in \mathcal{D}, \\ \mathbf{0}_{24}, & i \notin \mathcal{D}. \end{cases}$$

and the total input matrix for the entire system is given by

$$\mathbf{P}^d(t : t{+}23) = \big[\, \tilde{\mathbf{p}}_1^d(t : t{+}23), \, \ldots, \, \tilde{\mathbf{p}}_I^d(t : t{+}23) \,\big] \in \mathbb{R}^{24 \times I}.$$

This design ensures consistency with the complete network topology used by the GCN. For buses without load, zero vectors are used as placeholders.

1) GCN for Load Feature Extraction: For the input load power sequence $\mathbf{P}^d(t : t + 23)$, a GCN [14] is employed to extract load features while capturing the topological dependencies of the power grid. Unlike a CNN, which focuses on local feature extraction from individual load inputs, the GCN incorporates electrical connectivity among buses through a normalized adjacency matrix, providing a more physically consistent representation of grid operation. Specifically, the GCN consists of M layers indexed by $m = \{0, 1, \ldots, M-1\}$, and each layer updates node features at each time step t individually as:

$$\mathbf{H}^{m+1}(t) = \sigma\Big(\tilde{\mathbf{Q}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{Q}}^{-\frac{1}{2}} \mathbf{H}^m(t) \mathbf{W}^m \Big), \quad (6)$$

where $\mathbf{H}^m(t) \in \mathbb{R}^{I \times F_m}$ denotes the node feature matrix with F_m channels at layer m. For the input layer ($l = 0$), $\mathbf{H}^0(t) = \mathbf{P}^d(t)$. The adjacency matrix $\tilde{\mathbf{A}} \in \mathbb{R}^{I \times I}$ represents grid connectivity. In this work, we employ the susceptance matrix $\mathbf{B} \in \mathbb{R}^{I \times I}$ instead, so that each element b_{ij} reflects not only the connectivity but also the electrical coupling strength between buses. The degree matrix $\tilde{\mathbf{Q}} = \operatorname{diag}\Big(\sum_j \tilde{A}_{ij} \Big)$ normalizes the adjacency information to mitigate the dominance of highly connected nodes. $\mathbf{W}^m$ denotes the trainable weight matrix, and $\sigma(\cdot)$ is the ReLU activation function [15]. After processing the 24-hour sequence, we obtain the node features $\mathbf{H}^M(t{:}t{+}23) = \{\mathbf{H}^M(t), \, \ldots, \, \mathbf{H}^M(t{+}23)\} \in \mathbb{R}^{24 \times (I \times F_M)}$. Each $\mathbf{H}^M(t)$ contains only the spatial features extracted from its time step, without temporal correlations.

2) Attention Mechanism: These features are fed into the attention mechanism to capture temporal correlations across consecutive hours. Different hours exhibit distinct load levels, with morning and evening periods showing higher demand. The attention mechanism assigns adaptive weights to each time

979-8-3315-8138-1

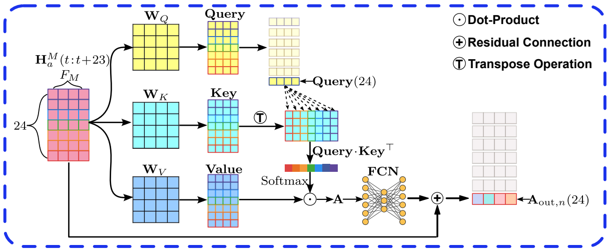

Fig. 2. Architecture of attention mechanism

step [16], enabling the model to focus on the most influential periods for generation dispatch decisions.

To improve computational efficiency, a fully connected layer (FCN) is applied to compress the full-grid feature dimension from $\mathbf{H}^M(t\!:\!t\!+\!23) \in \mathbb{R}^{24\times(I\times F_M)}$ to $\mathbf{H}_a^M(t\!:\!t\!+\!23) \in \mathbb{R}^{24\times F_M}$, yielding the compressed global representations.

$$\mathbf{H}_a^M(t\!:\!t\!+\!23) = \mathrm{ReLU}\big(\mathbf{H}^M(t\!:\!t\!+\!23)\mathbf{W}^a + \mathbf{b}^a\big)\,. \tag{7}$$

Then, $\mathbf{H}_a^M(t:t+23)$ is fed into the attention mechanism, which consists of a stack of n identical layers. Each layer contains an attention block followed by a fully connected layer, as illustrated in Fig. 2. The attention block uses three learnable parameter matrices, $\mathbf{W}_Q$, $\mathbf{W}_K$, and $\mathbf{W}_V$ (all in $\mathbb{R}^{F_M\times F_M}$), to linearly transform the input into the feature matrices **Query**, **Key**, and **Value** (each in $\mathbb{R}^{24\times F_M}$), enabling the model to capture temporal patterns across nodes.

$$\mathbf{Query} = \mathbf{H}_a^M(t\!:\!t\!+\!23)\mathbf{W}_Q, \tag{8}$$
$$\mathbf{Value} = \mathbf{H}_a^M(t\!:\!t\!+\!23)\mathbf{W}_V, \tag{9}$$
$$\mathbf{Key} = \mathbf{H}_a^M(t\!:\!t\!+\!23)\mathbf{W}_K. \tag{10}$$

The attention weights $\mathbf{A} \in \mathbb{R}^{24\times F_M}$ are then computed by (11), allocating temporal correlations to diverse nodal representations, where the scaling factor $\sqrt{F_M}$ prevents the dot-product magnitude from becoming excessively large.

$$\mathbf{A} = \mathrm{Softmax}\left(\frac{\mathbf{Query}\cdot\mathbf{Key}^\top}{\sqrt{F_M}}\right)\cdot\mathbf{Value} \tag{11}$$

Finally, $\mathbf{A}$ is fed into a two-layer FCN parameterized by $\mathbf{W}_p$ and $\mathbf{b}_p$ to further couple its spatio-temporal features, yielding the first block output $\mathbf{A}_{\text{out},1} \in \mathbb{R}^{24\times F_M}$. The output is then sequentially passed through the subsequent blocks, repeating the aforementioned process. After n stacked layers, the model produces an enriched representation $\mathbf{A}_{\text{out},n} \in \mathbb{R}^{24\times F_M}$, where each block has independently learned parameters.

3) Nonlinear Transformation for Multi-task Estimation: To simultaneously obtain generation dispatch, LMP, and line shadow prices (regression tasks), as well as the congestion status of each line (classification task), two independent two-layer FCNs are built, both sharing the same input $\mathbf{A}_{\text{out},n}$.

For the regression tasks, the transformation is defined as

$$\hat{\mathbf{P}}_r(t:t+23) = \mathrm{ReLU}(\mathrm{ReLU}(\mathbf{A}_{\text{out},n}\mathbf{W}_1^r + \mathbf{b}_1^r)\mathbf{W}_2^r + \mathbf{b}_2^r)\,, \tag{12}$$

where $\hat{\mathbf{p}}_r(t\!:\!t\!+\!23) = [\hat{\mathbf{p}}^g(t\!:\!t\!+\!23),\ \hat{\boldsymbol{\lambda}}(t\!:\!t\!+\!23),\ \hat{\boldsymbol{u}}(t\!:\!t\!+\!23)] \in \mathbb{R}^{24\times(2I+|\mathcal{L}|)}$ denotes the regression outputs, including the estimated generation dispatch $\hat{\mathbf{p}}^g$ for all generators, LMPs $\hat{\boldsymbol{\lambda}}$ for all buses, and the line shadow prices $\hat{\boldsymbol{u}}$ for all transmission lines.

For the classification task, the transformation is

$$\hat{\mathbf{y}}(t\!:\!t\!+\!23) = \mathrm{Sigmoid}(\mathrm{ReLU}(\mathbf{A}_{\text{out},n}\mathbf{W}_1^c\!+\!\mathbf{b}_1^c)\,\mathbf{W}_2^c\!+\!\mathbf{b}_2^c)\,, \tag{13}$$

where $\hat{\mathbf{y}} \in \mathbb{R}^{N_{\text{line}}}$ denotes the congestion estimation of all lines. $\mathbf{W}_1$ and $\mathbf{W}_2$ are the weight matrices of the first and second layers with corresponding biases $\mathbf{b}_1$ and $\mathbf{b}_2$. The Sigmoid activation bounds $\hat{\mathbf{y}}$ within $[0,1]$, representing the estimated congestion probability of each line.

4) Parameter Determination: To obtain the network parameters, we define $\phi = \{\phi_p,\ \mathbf{W}^r,\ \mathbf{b}^r,\ \mathbf{W}^c,\ \mathbf{b}^c\}$, where ϕ_p denotes the parameters of the feature extraction module before the non-linear transformation for multi-task estimation. The regression task parameters are defined as $\phi_r = \{\phi_p,\ \mathbf{W}^r,\ \mathbf{b}^r\}$, while the classification task parameters are defined as $\phi_c = \{\phi_p,\ \mathbf{W}^c,\ \mathbf{b}^c\}$. The learning process is performed on a dataset containing the ground truth of $\{\mathbf{p}_i^d,\ \mathbf{p}_i^g,\ \boldsymbol{\lambda},\ \mathbf{y}\}$, for all $i \in \mathcal{I}$.

$$\begin{aligned}\mathcal{L}^{\text{regression}}{}_{\phi_r} = \sum_{t=1}^{24}\sum_{i\in\mathcal{I}}\Big(&\|\hat{p}_i^g(t) - p_i^g(t)\|_2^2 \\ &+ \alpha_\lambda\|\hat{\lambda}_i(t) - \lambda_i(t)\|_2^2 + \alpha_u\|\hat{u}(t) - u(t)\|_2^2\Big).\end{aligned} \tag{14}$$

$$\begin{aligned}\mathcal{L}_{\phi_c}^{\text{classification}} = -\sum_{t=1}^{24}\sum_{\ell=1}^{|\mathcal{L}|}\Big(&y_\ell(t)\log\hat{y}_\ell(t) \\ &+ (1 - y_\ell(t))\log(1 - \hat{y}_\ell(t))\Big).\end{aligned} \tag{15}$$

$$\mathcal{L} = \mathcal{L}^{\text{regression}} + \mathcal{L}^{\text{classification}} \tag{16}$$

where $\|\cdot\|_2$ denotes the ℓ_2-norm, and α_λ and α_u represent the weights of the respective task. The final loss function is the sum of $\mathcal{L}^{\text{regression}}$ and $\mathcal{L}^{\text{classification}}$ Since the optimization processes of the two tasks are coupled, sharing the parameters ϕ_p allows the proposed method to save storage space. Given the definitions in 14 and 15 and a learning rate α, the parameters ϕ_r and ϕ_c are updated at each iteration t as follows:

$$\phi_r \leftarrow \phi_r - \alpha\,\frac{\partial\mathcal{L}^{\text{regression}}(t)}{\partial\phi_r}, \tag{17}$$

$$\phi_c \leftarrow \phi_c - \alpha\,\frac{\partial\mathcal{L}^{\text{classification}}(t)}{\partial\phi_c}. \tag{18}$$

III. Numerical Evaluation

The numerical experiments are implemented using the PyTorch framework on a desktop computer equipped with an AMD Ryzen 5 5600H processor (3.3 GHz), 16 GB of RAM, and an NVIDIA GeForce RTX 3050 GPU.

A. Dataset Overview

The proposed method is first evaluated on the IEEE 118-bus system. The IEEE 118-bus system includes 118 buses, 53 generators, and 186 transmission lines [17], with all parameters taken from MATPOWER v8.1 [18]. For data generation, each bus load is uniformly sampled within $[90\%, 110\%]$ of its nominal value over 365 days, and generator ramp limits are set to 15% of their maximum capacity. Ground-truth OPF solutions are obtained using the Gurobi solver [19].

Secondly, the study is performed on a variant of 118-bus system with hourly load profile adapted from actual MISO daily total load data from January 1, 2024 to January 1, 2025. Since only system-level loads are available, each day's total load is normalized to $[0, 1]$ and redistributed across buses according to their nominal load proportions. All operational constraints remain identical to the IEEE 118-bus case.

Both datasets are split into 80% training, 10% validation, and 10% testing. Input data are normalized by subtracting the training-set mean and dividing by its standard deviation.

B. Main Results

In Table I, the detailed model parameters of the proposed method are listed. The input channel dimension of the GCN corresponds to F_l. Based on empirical performance, the mini-batch size is set to 16, the maximum number of epochs to 400, and Adam is used as the optimizer with a learning rate of 1×10^{-4}.

TABLE I
DETAILS OF THE ATTENTION-ENHANCED GRAPH CONVOLUTIONAL NETWORK.

Module	Part	Layer	Channel	Data Size
GCN	GCN	1	$F_1 = 128$	$(I, 1) \rightarrow (I, 128)$
		2	$F_2 = 128$	$(I, 128) \rightarrow (I, 128)$
Attention	FCN	3	128	$(T,\ I \times 128) \rightarrow (T, 128)$
	Attention	4–6	\	$(T, 128) \rightarrow (T, 128)$
	FCN	7	512	$(T, 128) \rightarrow (T, 512)$
	FCN	8	128	$(T, 512) \rightarrow (T, 128)$
Nonlinear Transformation	FCN	9	512	$(T, 128) \rightarrow (T, 512)$
	FCN	10	n_{out}	$(T, 512) \rightarrow (T, n_{\text{out}})$

In the simulation process, to quantify the estimation accuracy, we adopt three evaluation metrics: the mean absolute error (MAE), the normalized root mean square error (NRMSE), and the F1-score. For each generator i, the MAE is defined as $\text{MAE} = \frac{1}{T}\sum_{t=1}^{T} \left|p_i^g(t) - \hat{p}_i^g(t)\right|$, and the NRMSE is given by $\text{NRMSE} = \frac{\sqrt{\frac{1}{T}\sum_{t=1}^{T}\left(p_i^g(t) - \hat{p}_i^g(t)\right)^2}}{\frac{1}{T}\sum_{t=1}^{T} p_i^g(t)}$. For the congestion classification task, the F1-score, defined as the harmonic mean of precision and recall, is calculated as $\text{F1} = \frac{2\,\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$, where $\text{Precision} = \frac{TP}{TP+FP}$ and $\text{Recall} = \frac{TP}{TP+FN}$. The evaluation results are summarized in Table II.

TABLE II
ACCURACY EVALUATION OF THE PROPOSED MODEL (IN MW).

Output	IEEE 118-bus System			118-bus Variant		
	MAE	NRMSE (%)	F1(%)	MAE	NRMSE (%)	F1(%)
Pg	0.040	0.160	\	0.028	0.050	\
Flow	0.089	0.361	\	0.074	0.220	\
LMP ($/MWh)	0.010	0.036	\	0.001	0.010	\
Congestion	\	\	0.994	\	\	0.9951

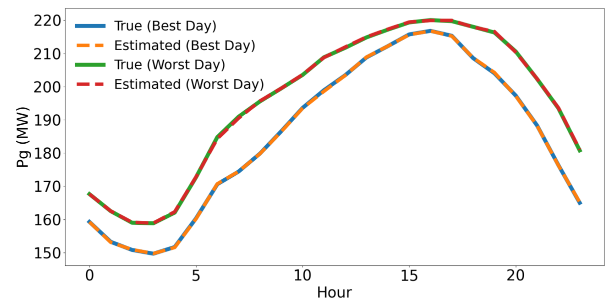

Fig. 3. Generator dispatch on the best and worst testing days.

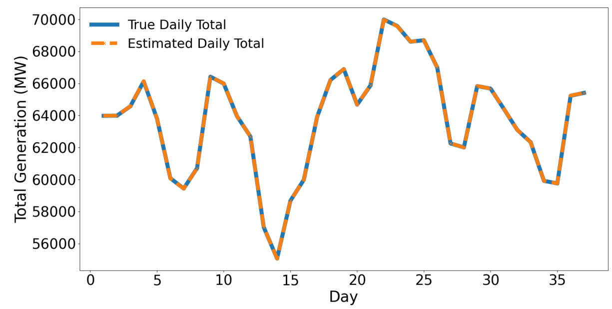

Fig. 4. Generator dispatch on all the testing days.

The testing results span 37 days and require only 0.023 seconds to generate all estimates. In contrast, the traditional Gurobi solver takes about 11.33 minutes for the testing. This demonstrates that our method offers a substantial speed advantage. As shown in Table II, the NRMSE for both generator dispatch and line flow is below 0.5%, confirming the high accuracy of the proposed method. Accurate line-flow estimation further improves LMP estimation, as both tasks share the same backbone before the nonlinear transformation layers. Likewise, congestion estimation influences LMP estimation, and this coupling highlights the effectiveness of the multi-task architecture in jointly learning correlated operational variables.

TABLE III
COMPUTATION TIME COMPARISON AND TESTING TIME SPEEDUP OVER GUROBI SOLVER.

Method	Training Time (min)	Testing Time (s)	Speedup
Gurobi Solver	26.0	679.800	\
AE-GCN (CPU)	18.54	0.219	3104
AE-GCN (GPU)	8.33	0.023	29556

To prevent the network outputs from violating system constraints and power balance, we follow the method in [13] and project all estimated variables that exceed operational limits back into their feasible ranges after inference. This post-processing step guarantees physical consistency without requiring a second OPF solve.

To further assess model performance, Fig. 3 and Fig. 4 compare the estimated generator dispatch with the ground truth. Fig. 3 presents the testing days with the lowest and highest NRMSE, where the estimated trajectories remain close to the true values even on the worst day. Fig. 4 further reports the total daily generation across all 37 testing days, where the estimated and true values exhibit strong consistency.

979-8-3315-8138-1

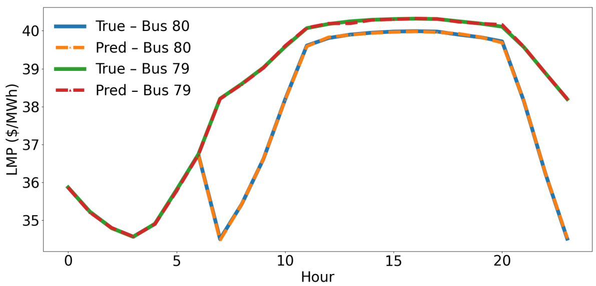

Fig. 5. LMP at buses exhibiting the largest deviation.

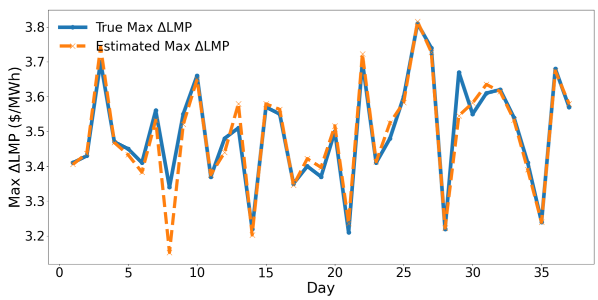

Fig. 6. LMP difference across all the testing days.

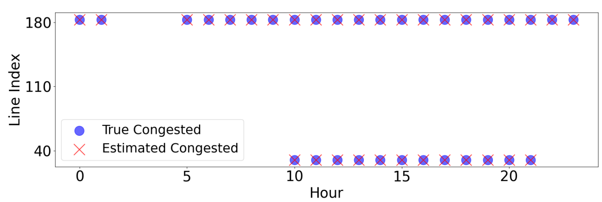

Fig. 7. Estimation for congestion.

These results confirm that the proposed method accurately estimates dispatch at both the single-day and multi-period scales, demonstrating its ability to capture the temporal and spatial characteristics of power system operations.

Furthermore, we identify the bus pair with the largest LMP divergence during testing and visualize their true and estimated LMP in Fig. 5. To analyze its variation over time, Fig. 6 presents the evolution of the maximum ΔLMP across all test days, which consistently arises from the same bus pair. The model maintains high accuracy on most days, with a noticeable deviation around Day 8, likely caused by an abrupt load change affecting congestion conditions.

In Fig. 7, we visualize the congestion estimation for the day with the most congestion. The comparison shows that the model successfully identifies all congested hours on both lines. This accuracy can be attributed to the shared network backbone—by accurately predicting the line flows, the model can naturally infer which lines are congested—thereby confirming its effectiveness in capturing branch-level congestion patterns.

IV. Conclusion

We have developed an attention-enhanced graph convolutional network for improving the estimation accuracy of multi-period optimal power flow. By capturing spatio-temporal grid correlations, the proposed method achieves high prediction accuracy with strong computational efficiency.

Future work will extend the framework toward ACOPF by modeling additional AC variables (e.g., bus voltage magnitudes and reactive power) as intermediate physical states within a constraint-aware learning framework. Physics-informed loss terms or differentiable OPF layers will be explored to reduce constraint violations. The method will also be evaluated under stressed operating conditions, including extreme ramping events, high congestion, and near-binding ramp constraints.

References

[1] H. W. Dommel and W. F. Tinney, "Optimal power flow solutions," *IEEE Transactions on power apparatus and systems*, no. 10, pp. 1866–1876, 2007.

[2] B. Stott, J. Jardim, and O. Alsaç, "Dc power flow revisited," *IEEE Transactions on Power Systems*, vol. 24, no. 3, pp. 1290–1300, 2009.

[3] N. T. Nguyen, D. D. Le, C. Bovo, and A. Berizzi, "Optimal power flow with energy storage systems: Single-period model vs. multi-period model," in *2015 IEEE Eindhoven PowerTech*. IEEE, 2015, pp. 1–6.

[4] A. Agarwal and L. Pileggi, "Large scale multi-period optimal power flow with energy storage systems using differential dynamic programming," *IEEE Transactions on Power Systems*, vol. 37, no. 3, pp. 1750–1759, 2022.

[5] A. Rabiee and A. Soroudi, "Stochastic multiperiod opf model of power systems with hvdc-connected intermittent wind power generation," *IEEE Transactions on Power Delivery*, vol. 29, no. 1, pp. 336–344, 2014.

[6] M. Alkhraijah, M. Alowaifeer, S. Grijalva, and D. K. Molzahn, "Distributed multi-period dcopf via an auxiliary principle problem algorithm," in *2021 IEEE Texas Power and Energy Conference (TPEC)*. IEEE, 2021, pp. 1–6.

[7] ——, "Distributed multi-period dcopf via an auxiliary principle problem algorithm," in *2021 IEEE Texas Power and Energy Conference (TPEC)*, 2021, pp. 1–6.

[8] A. Hauswirth, S. Bolognani, G. Hug, and F. Dörfler, "Projected gradient descent on riemannian manifolds with applications to online power system optimization," in *2016 54th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2016, pp. 225–232.

[9] X. Pan, M. Chen, T. Zhao, and S. H. Low, "Deepopf: A feasibility-optimized deep neural network approach for ac optimal power flow problems," *IEEE Systems Journal*, vol. 17, no. 1, pp. 673–683, 2022.

[10] H. Wu and Z. Xu, "Fast dc optimal power flow based on deep convolutional neural network," in *2022 IEEE 5th International Electrical and Energy Conference (CIEEC)*, 2022, pp. 2508–2512.

[11] S. Liu, C. Wu, and H. Zhu, "Topology-aware graph neural networks for learning feasible and adaptive ac-opf solutions," *IEEE Transactions on Power Systems*, vol. 38, no. 6, pp. 5660–5670, 2022.

[12] S. Talebi and K. Zhou, "Graph neural networks for efficient ac power flow prediction in power grids," *arXiv preprint arXiv:2502.05702*, 2025.

[13] R. Zafar and I.-Y. Chung, "Data-driven multiperiod optimal power flow for power system scheduling considering renewable energy integration," *IEEE Access*, 2024.

[14] T. Kipf, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[15] A. F. Agarap, "Deep learning using rectified linear units (relu)," *arXiv preprint arXiv:1803.08375*, 2018.

[16] G. Liu, L. Lv, C. Wang, H. Wang, H. Wan, and L. Yang, "Attention-enhanced actor–critic learning for household nonintrusive load monitoring," *IEEE Transactions on Industrial Informatics*, vol. 20, no. 12, pp. 14 361–14 370, 2024.

[17] C. H. Liang, C. Y. Chung, K. P. Wong, and X. Z. Duan, "Parallel optimal reactive power flow based on cooperative co-evolutionary differential evolution and power system decomposition," *IEEE Transactions on Power Systems*, vol. 22, no. 1, pp. 249–257, 2007.

[18] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

[19] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2024. [Online]. Available: https://www.gurobi.com

979-8-3315-8138-1