Prediction of Cascading Failure Size Using a Graph Neural Network with Dynamic Weights

Tingwei Chen, Zhenping Guo, Peigong Wang, Kai Sun, *Fellow, IEEE*,
Jin Dong, *Member, IEEE*, and Boming Liu, *Member, IEEE.*

***Abstract*—There has been growth in the applications of Graph Neural Network (GNN) models in various subfields of power systems, such as residential load prediction and power flow approximation, where the load remains relatively stable without significant fluctuations. However, there is a lack of studies on predicting power outage size using GNNs for catastrophic cascading failures. To address this gap, we introduce a two-layer neural network model named DW-GNN, which consists solely of an encoding layer and a GNN layer. By leveraging Dynamic Weighting adjustment and Multi-task Learning mechanisms, our model achieves state-of-the-art performance on the heavy loading condition in PowerCascade dataset.**

***Index Terms*—Cascading failure, Power outages, Failure size, Graph Neural Network, Dynamic Weighting-enhanced GNN, Multi-task learning.**

I. Introduction

In recent years, the application of Graph Neural Networks (GNN) models, such as residential load prediction [1] and power flow approximation [2]–[5], has garnered widespread attention due to their increased efficiency, compared to traditional simulation methods [6], in task processing and better interpretability in topological maps compared to traditional methods. These works span a wide range of topics and continue to contribute to various power system needs [7], [8]. Notably, the tasks of fault scenario application in power system have gained significant attention and remarkable achievements. For example, in the field of transformer fault diagnosis, reference [9] proposed a spectrum-based GNN model to enhance the accuracy of transformer fault detection. In the context of fault location, paper [10] introduced a graph-based Gate Recurrent Unit (GRU) method that automatically pinpoints distribution network faults, distinguishing it from traditional approaches based on voltage sags or impedance. Similarly, paper [11] developed a spectrum-based Graph Convolutional Network (GCN) model to integrate comprehensive information from multiple measurement units and capture spatial correlations between buses. Additionally, GNNs have seen extensive application in fault detection and isolation. Reference [12] modeled connected components in power systems as weighted undirected graph structures, addressing the limitations of traditional methods [13] that focus only on individual components and their respective features for fault identification and isolation. In power outage prediction, GNN models have also been applied successfully. By using a GNN-based model to process meteorological data, the work in [14] significantly improved outage prediction accuracy.

Despite the remarkable achievement, previous studies overlook the importance of cascading failure study that plays a pivotal role in power system tasks. On one hand, cascading failures in power grids can lead to widespread blackouts, causing substantial economic losses across industrial, commercial, and service sectors, while also triggering social panic and threatening public safety [15], [16]. On the other hand, understanding the propagation mechanisms of cascading failures can strengthen the grid's fault tolerance and rapid recovery capabilities. This insight enables more effective identification of vulnerable segments and high-risk areas within the grid, allowing for the development of preventive strategies that reduce the likelihood of failure events [17], [18].

Although some studies [19], [20] have successfully introduced GNN into cascading failure analysis in power system, limitations remain in the design of these tasks. Current research primarily focuses on predicting whether cascading failures will occur based on given initial fault conditions. While these models have achieved promising results, they fall short in assisting grid operators in predicting potential fault locations and severity levels under current grid conditions. These works have demonstrated the effectiveness of applying GNN into cascading failure analysis in power system, highlighting the urgent need for a comprehensive task that integrates them to prompt further advancements of GNN application in cascading failure analysis.

However, the application of GNN in cascading failure analysis posed several challenges. **C1. Difficulty in controlling dataset size:** In reality, large-scale cascading failures in power grids occur only a few times [16], limiting available data. To achieve high performance, some researchers intentionally use simulation platforms to construct large datasets, which may not accurately reflect real-world conditions. **C2. Lack of sufficient baseline models for comparison:** As discussed earlier, although some studies apply GNN to analyze cascading failures in power system, they primarily focus on predicting whether cascading failures will occur based on given initial fault conditions. However, there is limited research on predicting likely fault locations and fault severity, leaving a gap in this

This work was supported in part by NSF grant ECCS-2329924 and in part by the UT-Oak Ridge Innovation Institute.

T. Chen, Z. Guo, P. Wang and K. Sun are with the Department of EECS, University of Tennessee, Knoxville, TN, USA (e-mail: tchen39@vols.utk.edu, zguo19@vols.utk.edu, pwang29@vols.utk.edu, kaisun@utk.edu).

J. Dong and B. Liu are with the Electrification and Energy Infrastructures Division, Oak Ridge National Laboratory, Oak Ridge, TN, USA (e-mail: dongj@ornl.gov, liub@ornl.gov)

979-8-3315-8138-1

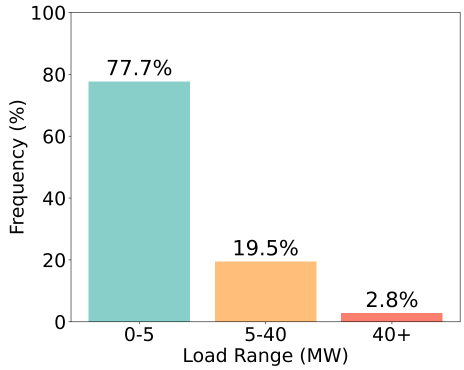

Fig. 1. Frequency Distribution of Load Change

critical area. **C3. Imbalanced data distribution:** Cascading failure data is highly imbalanced, as severe load losses occur with much lower probability than mild or moderate events [21]. As shown in Fig. 1, buses with large load changes constitute only a small portion of the overall data, making it challenging to capture these rare events effectively. Specially, we categorized the load changes on the bus into three classes based on the potential hazards they might cause. Class 0, ranging from 0 MW to 5 MW, is considered to have minimal impact on the system's operation, allowing it to function almost normally without requiring special attention from the operators. Class 1, ranging from 5 MW to 40 MW, indicates that there might be minor faults in the power grid. Class 2, with changes greater than 40 MW, signifies severe faults in the power grid that demand immediate and special attention from the operators.

In this paper, we introduce a Dynamic Weighting-enhanced GNN (namely DW-GNN).The heavy loading condition in PowerCascade dataset [22] was selected as the foundational dataset, upon which the load information was consolidated. In designing the model, we drew inspiration from the application of GNNs in other areas of computer science [23], [24] and incorporated the specific data distribution characteristics of cascading failures by combining the Dynamic Weighting strategy and the Multi-task Learning classifier. As a result, our model achieves state-of-the-art performance on current task. This paper aims to leverage advanced techniques to predict the scale of cascading failures, enabling utility companies to issue early warnings and take preventive operational actions before such failures propagate through the power system.

II. Model Design

This section introduces our DW-GNN model, whose architecture is illustrated in Fig. 2. To align with the topology of the power grid, we first encode the feature data from buses and branches separately, then input these encoded features jointly into the graph neural network. To mitigate overfitting, particularly due to limited data, we apply a dropout layer to the network output. The model's two-dimensional output is then normalized using the softmax function.

To address the challenge posed by imbalanced data distributions, where the model may struggle to learn the features of underrepresented classes, we implement a Dynamic Weighting strategy that balances class weights in the loss function. Additionally, to improve accuracy in the classification of three classes, particularly the middle class where similar features might lead to misclassification, we employ a Multi-task Learning-based classifier. This classifier allows us to output class predictions with shared representational information, which enhances the model's ability to capture fine-grained distinctions.

By combining the Dynamic Weighting strategy and the Multi-task Learning classifier, our DW-GNN model achieves state-of-the-art performance, effectively balancing precision across diverse data categories and handling complex class boundaries with greater accuracy.

A. Introduction of Graph Neural Networks

In this paper, we adopted the GATv2Conv model [25], which has demonstrated remarkable performance in learning node representations through an adaptive weighting mechanism based on attention coefficients. Original paper have also shown that the time complexity of GATv2Conv scales favorably with graph size, making them well-suited for large-scale applications in real-world systems. Compared to the original Graph Attention Networks model (GAT) [26], a special GNN model, GATv2Conv introduces significant improvements by relaxing the order-invariance assumption and enhancing flexibility in attention computation.

Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes ($\mathcal{V} = \{1, ..., n\}$) and edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. In our model, the input of nodes represent buses in the grid, where each node $i \in V$ has an input feature vector h_i representing the properties of the corresponding bus. Edges (j, k) represent branches connecting pairs of buses in the grid, and each edge $(j, k) \in \mathcal{E}$ is associated with an input feature vector e_{jk}, describing the branch properties.

For each edge $(j, k) \in \mathcal{E}$, we aggregate the features of the source node j (h_j), the target node k (h_k) and the edge itself (e_{jk}). This aggregation produces the feature x_{jk}, define as:

$$x_{jk} = h_j + h_k + e_{jk} \tag{1}$$

The aggregated feature x_{jk} is then passed through a non-linear activation function, specifically the LeakyReLU [27] used in the GATv2Conv model, yielding:

$$x'_{jk} = LeakyReLU(x_{jk}) \tag{2}$$

Here, x'_{jk} represents the attention score of edge (j, k). Treating j as the source node, attention scores are normalized across all its neighboring nodes $k \in N_j$ (where N_j is the set of neighbors of j) using the softmax function:

$$\alpha_{jk} = softmax_k(x'_{jk}) = \frac{exp(x'_{jk})}{\Sigma_{k' \in N_j} exp(x'_{jk'})} \tag{3}$$

979-8-3315-8138-1

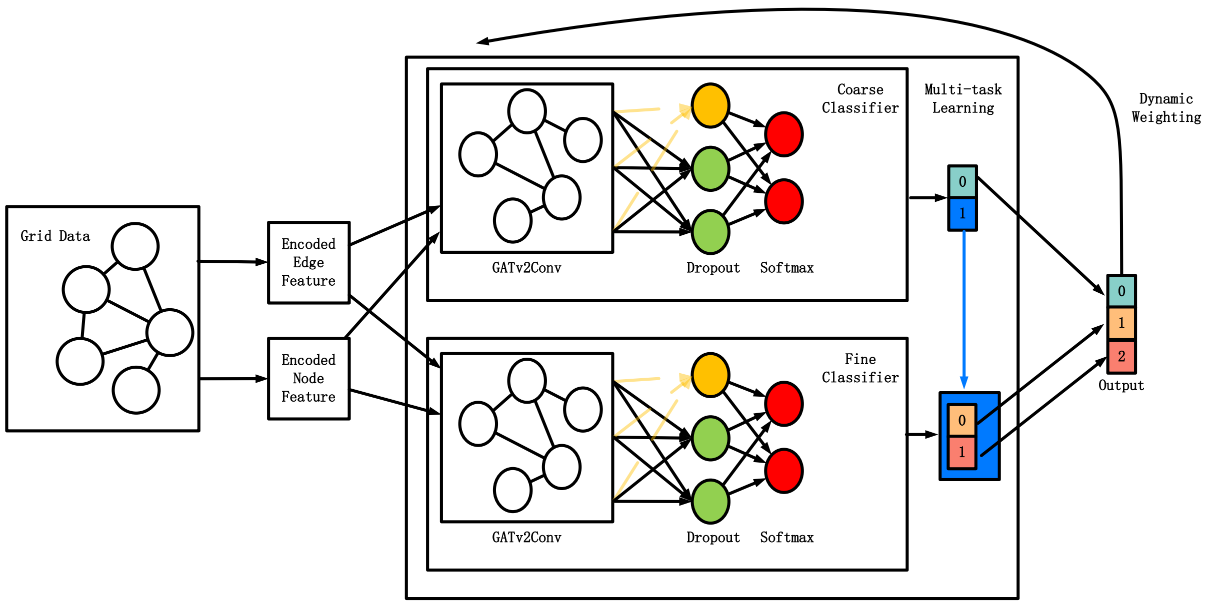

Fig. 2. DW-GNN architecture

With these attention scores, the representation of node k with respect to node j can be computed as:

$$r_{jk} = \alpha_{jk} \cdot (h_k + e_{jk}) \tag{4}$$

Finally, the output representation of node j after passing through a GATv2Conv layer is obtained by aggregating the representations r_{jk} of all its neighbors $k \in N_j$:

$$o_j = \Sigma_{k \in N_j} r_{jk} \tag{5}$$

B. Design of Dynamic Weighting

Given the imbalanced data distribution, we adopted a class-sensitive approach inspired by long-tail learning in computer science [28] to enhance the model's learning of features for underrepresented classes. Specifically, we rebalanced the training loss by reweighting each class, aiming to counteract the positive gradient imbalance between classes with different sample sizes.

Initially, we tried setting class weights as the inverse of sample counts to amplify the impact of classes with fewer samples. However, we observed that this adjustment was insufficient; minority classes often required weights even larger than the inverse of their sample sizes. Yet, we found that such high weights introduced a new challenge: classes with abundant samples, now assigned very low weights, suffered in performance during testing.

To address this, we implemented a Dynamic Weight adjustment strategy that gradually optimizes class balance during training, which is shown as Fig. 3. At the beginning, we assigned a larger weight to minority classes. Then, each time an improved model was obtained during training, we incrementally increased the weight for majority classes. In out experiment, we first initialized the loss weights according to the empirical imbalance of each subtasks: [1,5.5] for the coarse task and [1,13.5] for the fine task. During training, whenever the total loss decreased, we slightly increased the positive class-weight of each task. This adaptive approach allowed us to achieve an optimal balance, ensuring strong model performance across both minority and majority classes.

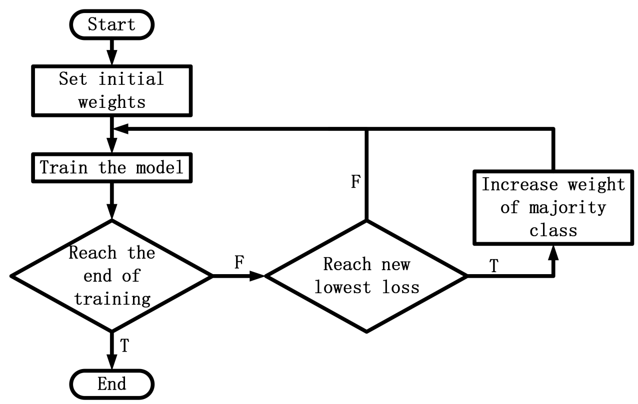

Fig. 3. Flowchart of Dynamic Weighting

C. Design of Multi-task Learning

Since the expected load changes are divided into three classes, directly predicting them is challenging—especially the intermediate class, whose features often overlap with adjacent categories. However, this middle category is the most critical in practice: small load losses are negligible, and large cascading failures are rare, making accurate identification of intermediate events the operator's primary concern. To address this difficulty, we adopt a Multi-task Learning framework inspired by transfer learning, which enables shared representations across related tasks and improves generalization compared with single-task training.

We design a two-level Multi-task Learning module consisting of coarse and fine classifiers to address data imbalance and subtle inter-class differences. The coarse classifier first assigns samples to broad categories, while the fine classifier further refines predictions within high-variation regions, improving recognition of the intermediate class.

To integrate outputs from both classifiers, we adopted a weighted joint probability strategy, balancing the importance of coarse and fine classifications to ensure robust decision-making in boundary cases. Additionally, we introduced a dynamic adjustment mechanism that modulates the weights of coarse and fine classifiers in real time based on each sample's confidence across both tasks, enhancing performance on long-tail data. This design not only improves overall classification accuracy but also significantly alleviates the issue of class imbalance.

III. Experiment

In this section, we present the experimental setup to evaluate the performance of our proposed model. We begin by describing the dataset used for our analysis, followed by a discussion of the evaluation metrics applied to measure model effectiveness. Finally, we report the experimental results, highlighting the advantages of our approach across various metrics and providing insights into its performance on key tasks.

A. Dataset

For our experiments, we chose heavy loading condition in PowerCascade dataset [22], containing 7,500 scenarios, which considered load variation during dataset generation, as our base dataset. Utilizing the similar setup in [21], PowerCascade dataset conduct cascading failure simulations [29] on the Northeastern Power Coordinating Council (NPCC) 140-bus system. In each scenario, they documented each stage by recording input features such as the current bus type, load values, branch parameters, and the open/closed status of each line.

For the consolidation of load value, we categorized the load change in the subsequent stage into three classes: 0 MW to 5 MW (class 0), 5 MW to 40 MW (class 1), and above 40 MW (class 2). These classifications serve as the prediction targets, providing a structured dataset to evaluate the model's ability to capture varying load shifts under cascading failure conditions. And in the training-testing task, we divide the dataset in the ratio of 8:1:1 for training set, validation set and testing set

B. Evaluation Metrics

Since we divide the overall data changes into three classes, we calculate the accuracy for each class i based on the total count of instances $Count_i$ in class i and the model's correctly predicted instances for that class, $Pred_i$. Based on these, we can construct the accuracy for each class.

$$Acc_i = \frac{Pred_i}{Count_i} \tag{6}$$

We then define two evaluation metrics, the $WeightedScore$ and the $BalancedScore$, to assess model performance across these classes.

$$WeightedScore = 25\%Acc_0 + 40\%Acc_1 + 35\%Acc_2 \tag{7}$$

$$BalancedScore = \Sigma_{i=0}^{2} Acc_i \tag{8}$$

The $WeightedScore$ uses varying weights to emphasize performance in specific classes of particular interest to our analysis. Specifically, We designed it with class-specific weights of 0.25, 0.4, and 0.35, reflecting the practical impact of each class on power grid operations. We assigned a lower weight to Class 0 because its occurrences have minimal impact on grid performance. This ensures the model is less sensitive to predictions of negligible importance, focusing its learning capacity on more critical scenarios. For Class 2, although these cases warrant high vigilance due to their severity, their rarity in real-world scenarios necessitated a balanced approach. Considering both their potential harm and low frequency, we assigned a weight slightly above an equal distribution to emphasize their importance without overshadowing other categories. Class 1, on the other hand, represents events with moderate risk but higher frequency in real-world power grids. Consequently, we assigned it the highest weight, as accurately predicting this class is crucial for mitigating frequent and potentially disruptive grid events. This weighted design aligns with practical operational priorities, allowing the model to achieve a balanced focus across different risk levels while emphasizing scenarios most critical to grid reliability.

In contrast, the $BalancedScore$ applies uniform weights across all classes, allowing us to evaluate whether the model maintains accuracy across all categories without compromising the performance of less emphasized classes to boost results in the prioritized ones. Together, these scores provide a comprehensive view of the model's accuracy, both in targeted performance and overall balance.

C. Input and Output Specifications of the Model

In this paper, as discussed earlier, our input consists of various data from the power system. The topological structure of the power system is used as the connectivity information of the graph. Additionally, properties of individual buses, such as power consumption, are provided as node features, while properties of each branch, such as the maximum branch load, are used as edge features.

For the output, the model predicts one of three values—0, 1, or 2—for each node. To calculate the model's error, we use the classified categories of the expected changes in node load as the reference y values.

D. Experiment Result

In our study, we selected classic models from machine learning and neural networks, including Decision Trees (DT) [30], Random Forests (RF) [31], Residual Networks (Resnet) [32], Graph Convolution Networks (GCN) [33], and Graph Attention Networks (GAT) [26], as baselines for comparison.

979-8-3315-8138-1

The comparison results are shown in Table I. We used bold text to indicate the best performance scores for each evaluation metric and underlined the second-best scores. The results demonstrate that our proposed model achieved state-of-the-art performance on both evaluation metrics.

TABLE I
EXPERIMENT RESULT

	Model Name	WeightedScore	BalancedScore
Machine Learning Model	DT [30]	0.5367	1.7083
	RF [31]	0.5416	1.7221
Neural Network Model	Resnet [32]	0.3442	1.2165
	GCN [33]	0.2516	0.9999
	GAT [26]	0.2500	1.0000
	DW-GNN (ours)	**0.7031**	**2.1278**

IV. CONCLUSION

This paper aims to leverage advanced techniques to predict the scale of cascading failures, enabling utility companies to issue early warnings and take preventive operational actions before such failures propagate through the power system. In this paper, we introduce a cascading failure size prediction model based on GNN named DW-GNN. which firstly applied GNN model into cascading failure-related problems that are helpful to reality. The proposed DW-GNN is composed of two main layers, *i.e.*, encoding layer and graph neural network layer. Utilizing the PowerCascade dataset, we consolidate the expected load changes into three classes based on the potential hazards they might cause. By testing on this dataset, we found that our model, with the help of Dynamic Weighting and Multi-task Learning, achieved state-of-the-art performance on both evaluation metrics.

REFERENCES

[1] C. Dinesh, S. Makonin, and I. V. Bajić, "Residential power forecasting using load identification and graph spectral clustering," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 11, pp. 1900–1904, 2019.

[2] V. Bolz, J. Rueß, and A. Zell, "Power flow approximation based on graph convolutional networks," in *2019 18th IEEE ICMLA*, pp. 1679–1686, IEEE, 2019.

[3] Q. Yang, A. Sadeghi, G. Wang, G. B. Giannakis, and J. Sun, "Power system state estimation using gauss-newton unrolled neural networks with trainable priors," in *2020 IEEE SmartGridComm*, pp. 1–6, 2020.

[4] D. Wang, K. Zheng, Q. Chen, G. Luo, and X. Zhang, "Probabilistic power flow solution with graph convolutional network," in *2020 IEEE PES ISGT-Europe*, pp. 650–654, IEEE, 2020.

[5] B. Donon, R. Clément, B. Donnot, A. Marot, I. Guyon, and M. Schoenauer, "Neural networks for power flow: Graph neural solver," *Electric Power Systems Research*, vol. 189, p. 106547, 2020.

[6] K. Huang, Y. Liu, K. Sun, and F. Qiu, "Pi-controlled variable time-step power system simulation using an adaptive order differential transformation method," *IEEE Transactions on Power Systems*, 2024.

[7] W. Liao, B. Bak-Jensen, J. R. Pillai, Y. Wang, and Y. Wang, "A review of graph neural networks and their applications in power systems," *MPCE*, vol. 10, no. 2, pp. 345–360, 2021.

[8] B. Liu, C. Zhang, J. Dong, P. Balaprakash, Y. Liu, and B. Eiffert, "Enhancing power distribution system resilience with fusion-gnn: A dynamic graph representation learning approach," in *IECON, Chicago, IL, November 3-6*, 2024.

[9] W. Liao, D. Yang, Y. Wang, and X. Ren, "Fault diagnosis of power transformers using graph convolutional network," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 2, pp. 241–249, 2020.

[10] J. T. de Freitas and F. G. F. Coelho, "Fault localization method for power distribution systems based on gated graph neural networks," *Electrical Engineering*, vol. 103, no. 5, pp. 2259–2266, 2021.

[11] K. Chen, J. Hu, Y. Zhang, Z. Yu, and J. He, "Fault location in power distribution systems via deep graph convolutional networks," *IEEE JSAC*, vol. 38, no. 1, pp. 119–131, 2019.

[12] H. Khorasgani, A. Hasanzadeh, A. Farahat, and C. Gupta, "Fault detection and isolation in industrial networks using graph convolutional neural networks," in *2019 IEEE ICPHM*, pp. 1–7, IEEE, 2019.

[13] R. M. Ferrari, T. Parisini, and M. M. Polycarpou, "Distributed fault detection and isolation of large-scale discrete-time nonlinear systems: An adaptive approximation approach," *IEEE Transactions on Automatic Control*, vol. 57, no. 2, pp. 275–290, 2011.

[14] D. Owerko, F. Gama, and A. Ribeiro, "Predicting power outages using graph neural networks," in *2018 IEEE GlobalSIP*, pp. 743–747, 2018.

[15] K. Sun, ed., *Cascading Failures in Power Grids: Risk Assessment, Modeling, and Simulation*. Springer, 2024.

[16] Z. Guo, K. Sun, X. Su, and S. Simunovic, "A review on simulation models of cascading failures in power systems," *iEnergy*, vol. 2, no. 4, pp. 284–296, 2023.

[17] K. Sun, Y. Hou, W. Sun, and J. Qi, *Power system control under cascading failures: understanding, mitigation, and system restoration*. John Wiley & Sons, 2019.

[18] Z. Guo, X. Su, K. Sun, B. Park, and S. Simunovic, "Analysis and mitigation of cascading failures using a stochastic interaction graph with eigen-analysis," *IEEE Transactions on Power Systems*, vol. 40, no. 2, pp. 1675–1685, 2024.

[19] S. Chadaga, X. Wu, and E. Modiano, "Power failure cascade prediction using graph neural networks," in *2023 IEEE SmartGridComm*, pp. 1–7, IEEE, 2023.

[20] A. Varbella, B. Gjorgiev, and G. Sansavini, "Geometric deep learning for online prediction of cascading failures in power grids," *Reliability Engineering & System Safety*, vol. 237, p. 109341, 2023.

[21] Z. Guo, K. Sun, and S. Simunovic, "Comparing interaction graphs on cascading outages under different loading conditions," in *2024 IEEE PESGM*, pp. 1–5, 2024.

[22] T. Chen, Z. Guo, K. Huang, and K. Sun, "Descriptor: Power systems cascading failure analysis synthetic dataset (powercascade)," *IEEE Data Descriptions*, 2025.

[23] J. Chen, W. Zhan, H. Wang, Z. Jia, J. Gan, J. Zhang, J. Qi, T. Chen, L. Huang, M. Chen, *et al.*, "Metamatbench: Integrating heterogeneous data, computational tools, and visual interface for metamaterial discovery," in *ACM KDD*, pp. 5334–5344, 2025.

[24] T. Chen, J. Chen, and D. Zhou, "3d-fum: Benchmarking 3d molecule learning with functional groups," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 8635–8639, 8 2024.

[25] S. Brody, U. Alon, and E. Yahav, "How attentive are graph attention networks?," in *ICLR*, 2022.

[26] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[27] B. Xu, "Empirical evaluation of rectified activations in convolutional network," *arXiv preprint arXiv:1505.00853*, 2015.

[28] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, "Deep long-tailed learning: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 9, pp. 10795–10816, 2023.

[29] I. Dobson, B. A. Carreras, V. E. Lynch, D. E. Newman, *et al.*, "An initial model for complex dynamics in electric power system blackouts," in *hicss*, 2001.

[30] L. Breiman, J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*. Wadsworth International Group, 1984.

[31] L. Breiman, "Random forests," *Machine learning*, vol. 45, pp. 5–32, 2001.

[32] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[33] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

979-8-3315-8138-1