Assessing Explanations of Graph Neural Networks for Dynamic Stability Assessment of Power Grids

Martin Sadric
Karlsruhe Institute of Technology
Karlsruhe, Germany
martin.sadric@kit.edu

Sebastian Pütz
Karlsruhe Institute of Technology
Karlsruhe, Germany
sebastian.puetz@kit.edu

Christian Nauck
Potsdam Institute for Climate Impact Research
Potsdam, Germany
christian.nauck@pik-potsdam.de

Veit Hagenmeyer
Karlsruhe Institute of Technology (KIT)
Karlsruhe, Germany
veit.hagenmeyer@kit.edu

Frank Hellmann
Potsdam Institute for Climate Impact Research
Potsdam, Germany
frank.hellmann@pik-potsdam.de

Benjamin Schäfer
Karlsruhe Institute of Technology
Karlsruhe, Germany
benjamin.schaefer@kit.edu

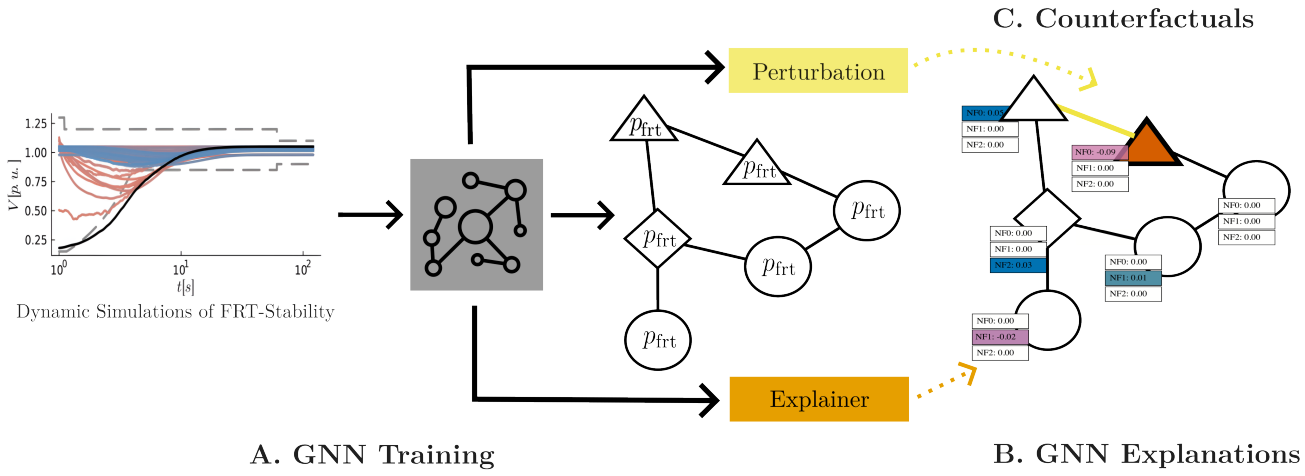


Figure 1: Graphical overview: We assess explainability of GNN-based dynamic stability predictions. (A) A graph neural network is trained on dynamic simulations (the FRT-Stability dataset). (B) Post-hoc explanation methods attribute feature importance to individual node features. (C) Counterfactual edge perturbations identify topology changes that improve predicted stability.

Abstract

As Graph Neural Networks (GNNs) are increasingly deployed for power grid stability predictions, understanding whether their learned behavior aligns with physical grid dynamics becomes essential for safe deployment. We present a systematic analysis addressing two questions: do post-hoc explanations accurately reflect GNN model behavior, and does that behavior align with the physics of grid dynamics? We apply gradient-based methods (five variants including Saliency, InputXGradient, and Integrated Gradients) and a game-theory-based method (Shapley Value Sampling) to a Dirac–Bianconi Graph Neural Network (DBGNN) achieving a skill score of 0.903 on a fault-ride-through (FRT) probability prediction task. Our analysis reveals that the model primarily relies on node type-level information, learning to distinguish inverters (and subtypes via the Normal Form (NF) parameter) from loads, rather than continuous power flow and network features within node types. Nevertheless, it incorporates meaningful topological context: neighborhood composition modulates predictions in physically intuitive ways, with influence decaying with graph distance. We validate the explanation methods on the IEEE39-AC dataset with known ground truth. Beyond explanation, we present a proof-of-concept counterfactual analysis: edge removals improve predicted stability by 45% on average at the most critical nodes while preserving global grid stability, reducing the search space for physics-based validation.

ACM Sustainability Week Companion '26, Banff, AB, Canada

ACM ISBN 979-8-4007-2199-1/26/06
https://doi.org/10.1145/3765611.3815508

CCS Concepts

• **Computing methodologies** → **Artificial intelligence**; **Machine learning**; • **Applied computing** → *Physical sciences and engineering*.

Keywords

Explainable AI, Graph Neural Networks, Power Grid Stability, Dynamic Stability Assessment, Feature Attribution, Counterfactual Analysis

ACM Reference Format:

Martin Sadric, Sebastian Pütz, Christian Nauck, Veit Hagenmeyer, Frank Hellmann, and Benjamin Schäfer. 2026. Assessing Explanations of Graph Neural Networks for Dynamic Stability Assessment of Power Grids. In *ACM Sustainability Week 2026 (ACM Sustainability Week Companion '26), June 22–25, 2026, Banff, AB, Canada.* ACM, New York, NY, USA, 17 pages. https://doi.org/10.1145/3765611.3815508

1 Introduction

Climate change is driving the transition to sustainable and carbon-neutral energy systems [40]. While many studies focus on steady-state analysis, the growing penetration of distributed energy resources such as wind and solar introduces significant operational challenges. Lower system inertia and volatile feed-in accelerate grid dynamics, requiring active monitoring and interventions to maintain stability [31]. Ensuring stable operation requires comprehensive dynamic stability assessments through non-linear simulations [51]. Probabilistic stability analyses using Monte Carlo simulations [14] are increasingly employed to evaluate large ensembles of disturbances. However, these simulations are too expensive to cover all potential grid configurations, leaving some instabilities undetected. This poses particular challenges for transmission system operators (TSOs) who require efficient, scalable methods to identify critical lines and anticipate control needs.

To address these challenges, Graph Neural Networks (GNNs) have emerged as a powerful tool capable of efficiently modeling the high-dimensional, interconnected, and non-Euclidean nature of power systems [24]. GNNs encode local and global grid properties in learned embeddings, providing a computationally efficient alternative to traditional simulations for rapid stability assessment [36]. Their speed enables earlier instability detection and preventive action.

However, despite their computational advantages and high predictive accuracy, GNNs remain black boxes. Their adoption in critical infrastructure requires not just performance but understanding: Do these models learn how continuous power flow features affect stability, or do they rely on node type patterns? Explainable Artificial Intelligence (XAI) addresses this by revealing what information models use and whether those patterns align with domain knowledge or fail under distribution shift [1]. As we demonstrate, this assumption can fail: our model relies primarily on node type, an abstract feature summarizing component dynamics, rather than continuous features like conductance and susceptance. Explainability is therefore not just a final validation step, but a diagnostic tool that may prompt model refinement or rejection.

While recent work [32] demonstrates that GNNs can predict stability metrics with strong performance, how models use graph structure and whether their learned representations reflect physical causality remain unexplored.

This paper presents a systematic study of GNN explainability for power grid dynamic stability prediction, focusing on two distinct questions: do post-hoc explanations accurately reflect model behavior, and does that behavior align with physical principles? We first validate our explanation methods on a synthetic dataset, IEEE39-AC [38], with known ground-truth physics, then apply them to a Fault-Ride-Through Stability (FRT-Stability) dataset, which captures the probability that grid components maintain operation within acceptable bounds following transient disturbances. Using a DBGNN [33] (skill score 0.903), we investigate which features the model relies on and whether they reflect physical principles. Finally, we show how the trained GNN can be repurposed for counterfactual explainability analysis, identifying edge removals that may improve stability at vulnerable nodes.

Contributions

(1) GNNs outperform non-graph methods (skill score 0.903 vs. $\leq$ 0.03) on FRT-Stability; architectural analysis shows accurate prediction requires enlarged spatial context and explicit edge features.
(2) Three explanation methods validated on IEEE39-AC with known ground truth reveal that on FRT-Stability node type (load vs. inverter) emerges as the primary predictive signal, with continuous power flow features playing a secondary role. Ablation confirms this: type-only achieves $R^2 = 0.90$; continuous features alone yield $R^2 \approx 0.02$.
(3) GNNs nevertheless capture meaningful topological context: neighborhood composition modulates predictions in physically intuitive ways, with influence decaying with graph distance.
(4) As a proof-of-concept, a counterfactual edge-removal framework identifies candidate interventions that boost *predicted* stability by 45% on average at critical nodes while preserving global grid integrity, without requiring full re-simulation.

2 Related Work

GNNs have been applied to various power system tasks, including stability assessment, fault diagnosis, and reliability analysis. We review three relevant areas: GNNs for dynamic stability prediction, explainability methods for such models, and the broader landscape of explainable GNNs in power systems.

GNNs for Dynamic Stability Assessment. Recent work demonstrates that GNNs effectively predict dynamic stability in power systems by learning from network topology and operational states. This work includes predicting single-node basin stability (SNBS) with generalization across network sizes [35] and identifying vulnerable nodes across synthetic and real-world topologies [34]. Other architectures address specific stability challenges: Khamis et al. [20] model inverter dynamics with graph convolutional networks, while Liu et al. [26, 28] incorporate temporal dependencies and physics-informed constraints for transient stability assessment. Despite this predictive success, understanding how GNNs make stability predictions remains largely unexplored.

Explainability for Dynamic Stability Prediction. Raum et al. [39] provide the first explainability analysis of GNN-based stability prediction, applying Layer-wise Relevance Propagation (LRP) to basin stability on Kuramoto-modeled grids. However, the Kuramoto model assumes homogeneous coupling and omits voltage dynamics, reactive power, and control systems, limiting applicability to operational power systems. Their key finding is that neighborhoods beyond direct connections strongly influence predictions, though their analysis lacks validation against ground-truth failure mechanisms. We extend this by validating multiple explanation methods on more realistic grid models.

GNN Explainability in Power Systems. Existing work on GNN explainability in power systems follows three main approaches. First, physics-informed methods embed domain knowledge directly into the model, for instance, through Kirchhoff's law constraints [55] or by reconstructing physical parameters [37], achieving interpretability by design rather than post-hoc analysis. Second, attention-based architectures use learned edge weights as explanations [30, 56], though these reflect model attention rather than feature importance. Third, post-hoc explanation methods such as GNNExplainer [9], SHAP [19], and LRP [27] have been applied to tasks including fault diagnosis and cascading failure analysis, though often without systematic validation against ground truth. Varbella et al. [50] address this gap with PowerGraph, a benchmark providing ground-truth explanations for cascading failures, finding that gradient-based methods achieve highest faithfulness. Our work follows a similar validation-first approach: we compare multiple explanation methods on a synthetic benchmark with known physics (IEEE39-AC dataset) before applying them to a stability prediction task.

3 Methods

3.1 Datasets

We evaluate our explainability methods on two datasets: a synthetic dataset based on the IEEE39-AC benchmark system for validation, and the FRT-Stability dataset for dynamic stability prediction.

3.1.1 IEEE39-AC Synthetic Dataset. The dataset comprises 1,000 samples from the IEEE39-AC benchmark [38], generated with pandapower [49] by uniformly perturbing active/reactive loads and generator outputs by ±30% before solving AC power flow equations.

Physical Foundation. The target variable is active power P_i, governed by the steady-state AC power flow equations:

$$P_i = \sum_{j \in \mathcal{N}_i} V_i V_j \left(G_{ij} \cos(\theta_i - \theta_j) + B_{ij} \sin(\theta_i - \theta_j) \right), \qquad (1)$$

where V_i and θ_i are voltage magnitude and phase angle at bus i, and G_{ij}, B_{ij} are conductance and susceptance from the admittance matrix. Node features (Q, V, θ) are corrupted with Gaussian noise $\mathcal{N}(0, \sigma^2)$ to simulate measurement uncertainty, where $\sigma = 0.01$ for voltage and angle, and $\sigma = 0.05$ for reactive power.

Ground Truth Validation. This dataset provides ground truth explanations defined by power flow physics. According to Eq. (1), P_i depends causally only on its own voltage magnitude, voltage angle, and the same properties of direct neighbors (1-hop). Reactive power Q does not appear in Eq. (1), though it may correlate with other variables. To test causal recovery, we include a dummy feature: reactive power values shuffled randomly across samples. A faithful explanation method should assign zero importance to this dummy feature while correctly identifying voltage angle differences as dominant for predicting active power.

3.1.2 FRT-Stability Dataset. The Fault-Ride-Through Stability (FRT-Stability) dataset [32] captures dynamic grid behavior following transient disturbances in inverter-dominated power grids. The dataset comprises 1,000 synthetic grid topologies with 70–80 buses each, generated using the framework described in [7]. Each grid contains 100% renewables, split equally between grid-forming and grid-following inverters. Grid-forming inverters are modeled using the normal-form model [22], validated through power-hardware-in-the-loop measurements [8].

Target Variable. The fault-ride-through (FRT) probability p_{frt} measures the likelihood that all grid components remain within operational bounds after a fault is cleared. The probability is computed as:

$$p_{\text{frt}} = \frac{V^*}{V_t}, \qquad (2)$$

where V^* is the number of post-clearance states for which all buses remain within ride-through curves, and $V_t = 1{,}000$ is the total number of states. The target ranges from 0 (certain failure) to 1 (certain survival). Sampling details are provided in Appendix A.1.

The distribution of p_{frt} across node types is shown in Figure 10 (see Appendix E.3).

Inverter Parameterization. Three grid-forming inverter parameterizations (NF1–NF3) differ only in the active-power low-pass filter time constant τ_P; buses with larger τ_P exhibit higher stability. Full parameterization and line model details are given in Appendix A.1.

Node and Edge Features. Each node is described by nine features (power flow variables, NF parameter, aggregated line properties, and node-type indicators) and each edge by five transmission line properties; see Table 3 (see Appendix A.1).

3.2 Graph Neural Network Architectures

Graph Neural Networks operate on graph-structured data $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ through message-passing, where node embeddings are iteratively updated by aggregating information from neighbors [17]. Building on the training code of [32], we retrain DBGNN on the FRT-Stability dataset and extend the comparison to three additional architectures (GCN, TAG, GATv2) representing distinct design principles. GCN [21] uses fixed aggregation weights from the normalized graph Laplacian with a receptive field limited by layer depth. TAG [12] extends the receptive field via learnable polynomial filters aggregating K-hop neighborhoods ($K = 3$) without stacking layers, but operates exclusively on node features. GATv2 [4] introduces attention-based neighbor weighting and incorporates edge features.

DBGNN [33] differs from these diffusive architectures. Based on the topological Dirac operator, it couples nodes and edges as joint

dynamical entities, enabling wave-like propagation:

$$\begin{aligned} \mathbf{x}_i(t+1) &= \mathbf{x}_i(t) + \mathbf{W}^{ne} \sum_{j \in \mathcal{N}_i} \mathbf{e}_{ij}(t) + \mathbf{W}^n_\beta \mathbf{x}_i(t), \\ \mathbf{e}_{ij}(t+1) &= \mathbf{e}_{ij}(t) + \mathbf{W}^{en}(\mathbf{x}_i(t) - \mathbf{x}_j(t)) - \mathbf{W}^e_\beta \mathbf{e}_{ij}(t), \end{aligned} \tag{3}$$

where $\mathbf{x}_i$ are node embeddings, $\mathbf{e}_{ij}$ are edge embeddings, $\mathbf{W}^{ne}$ and $\mathbf{W}^{en}$ couple nodes and edges, and $\mathbf{W}^n_\beta$, $\mathbf{W}^e_\beta$ are mass matrices. This wave-like propagation preserves signal structure over longer distances, suited to transient grid dynamics where disturbances propagate as electromagnetic waves.

Non-graph Models. For comparison, we evaluate non-graph methods using only local node features: a Multi-Layer Perceptron (MLP) and Gradient Boosted Trees (GBT) using XGBoost [10]. We also train a linear regression baseline on the same node features and splits; this baseline defines the skill score used in our primary performance tables.

3.3 Evaluation Metrics and Baselines

We report model performance using the skill score, which measures improvement relative to a linear regression baseline:

$$SS = \frac{R^2_{\text{model}} - R^2_{\text{baseline}}}{1 - R^2_{\text{baseline}}}. \tag{4}$$

Here R^2_{baseline} is the coefficient of determination of the linear regression baseline trained on identical features, and R^2_{model} is the model's R^2. A skill score of 1 indicates perfect prediction, 0 indicates no improvement over baseline, and negative values indicate degradation. R^2, MAE, and MSE are reported in Appendix D.1. Hyperparameters, data preprocessing steps, and additional dataset details are provided in Appendix B.2.

3.4 Explainability Methods

Post-hoc explainability methods identify which input features most influence GNN predictions. We employ methods from two paradigms: gradient-based and game-theory-based; detailed formulations are given in Appendix C.

Gradient-Based Attribution. Gradient-based methods assess feature importance via backpropagation. We evaluated five variants from the Captum library [23]: *Saliency* [45], *InputXGradient*, *GuidedBackpropagation* [47], *Deconvolution* [54], and *Integrated Gradients* [48]. All five produce near-identical feature rankings on our data (see Figure 8 in the Appendix). We select Integrated Gradients as our primary gradient-based method because it satisfies the completeness and sensitivity axioms: attributions sum to the prediction difference and relevant features receive non-zero attribution. We use zeros as the baseline. As shown in Section 4.2, its results are consistent with Shapley Value Sampling while being substantially faster.

Shapley Value Sampling. Shapley values [44] distribute predictions among features based on marginal contributions across all possible feature coalitions.

Explanation Scope. We focus on *model explanations* (what features the model uses) rather than *phenomenon explanations* (true physical causality). Ground truth explanations are only available for IEEE39-AC; FRT-Stability lacks analytical ground truth due to complex transient dynamics. All methods use default parameters from PyTorch Geometric [15, 16] and Captum [23] unless otherwise specified.

3.5 Counterfactual Intervention Framework

Counterfactual reasoning complements attribution methods: instead of explaining why a prediction was made, it identifies what minimal change would alter it. In power grid applications, this allows operators to use the GNN directly for evaluating intervention strategies.

Given a baseline grid $\mathcal{G}$ with GNN-predicted stability scores $f_\theta(\mathcal{G})_v$ for each node v, we identify the critical node

$$v^* = \arg\min_v f_\theta(\mathcal{G})_v$$

with lowest predicted stability. We then generate counterfactual topologies $\{\mathcal{G}'_e\}_{e \in \mathcal{E}}$ by removing each transmission line e and re-evaluating the GNN. A counterfactual topology $\mathcal{G}'_e$ is valid only if it satisfies three conditions:

1. **Connectivity**: The graph remains fully connected after edge removal.
2. **Global Floor**: The new system-wide minimum stability is not worse than baseline: $\min f_\theta(\mathcal{G}'_e) \geq \min f_\theta(\mathcal{G})$.
3. **Local Stability**: No individual node experiences a stability drop exceeding tolerance $\delta = 0.05$: $\forall v,\ f_\theta(\mathcal{G}'_e)_v \geq f_\theta(\mathcal{G})_v - \delta$.

From the set of valid interventions, we select

$$e^* = \arg\max_e f_\theta(\mathcal{G}'_e)_{v^*}$$

that maximizes the stability gain at the critical node. This approach exploits Braess's Paradox [43], where removing network capacity can counterintuitively improve stability by mitigating congestion or preventing instability propagation.

Unlike traditional dynamic simulation (potentially hours per scenario), GNN re-evaluation requires only milliseconds per forward pass. For a grid with $|\mathcal{E}|$ edges, exhaustive counterfactual search requires $|\mathcal{E}|$ GNN evaluations, completing in seconds on modern hardware. This enables rapid screening of candidate interventions before physics-based validation.

All models are trained across five random seeds to assess robustness, with results reported as mean ± standard deviation.

4 Results and Discussion

4.1 Graph and Non-Graph ML Performance

Transient phenomena in power grids propagate along network connections, creating dependencies between distant nodes. We compare models with increasing access to topology information to identify which inputs are essential for stability prediction.

In Table 1 we compare model skill scores (Eq. 4) between non-graph machine learning models and different GNN architectures (corresponding R^2 scores are reported in Table 7). On the IEEE39-AC dataset, non-graph methods already achieve moderate skill

scores of 0.62–0.63, reflecting the simpler nature of static power flow prediction where dependencies are more local.

Table 1: Model skill scores (mean ± std over 5 seeds) on IEEE39-AC (static power flow) and FRT-Stability (dynamic stability), measuring improvement over a linear regression baseline.

Architecture	IEEE39-AC (Skill)	FRT-Stability (Skill)
Non-Graph		
MLP	0.621 ± 0.007	0.016 ± 0.002
GBT	0.633 ± 0.003	0.022 ± 0.001
Graph-Based		
GCN	0.920 ± 0.007	0.606 ± 0.022
TAG	0.957 ± 0.003	0.700 ± 0.004
GATv2	0.863 ± 0.022	0.766 ± 0.012
DBGNN	0.952 ± 0.017	**0.903** ± 0.002

For FRT-Stability prediction, there is a substantial performance gap between architectures. Non-graph models (MLP, GBT) receive only node features as input and predict stability for each node independently. These models achieve skill scores below 0.03. Local node features alone provide no improvement beyond linear regression.

Incorporating topology information via GCN raises the skill score to 0.606, but GCN's limited receptive field and uniform treatment of neighbors constrain further gains. Increasing network depth is not a solution: deeper GNNs suffer from oversmoothing [41], so we limit all models to a maximum of three layers.

TAG overcomes this limitation without adding layers by aggregating across a k-hop neighborhood ($k = 3$), improving the skill score to 0.700. This confirms that longer-range spatial context benefits stability prediction. However, TAG operates exclusively on node features and does not incorporate edge attributes.

GATv2 adds edge features and achieves a skill score of 0.766, indicating that transmission line properties (conductance, susceptance) carry predictive value beyond node characteristics alone. Like GCN, it is limited to three layers to prevent oversmoothing.

DBGNN achieves the highest skill score (0.903) by combining expanded receptive fields, edge features, and wave-like propagation that better matches power flow physics than diffusive schemes.

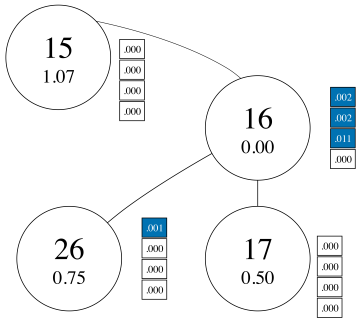
In the following sections, we focus explanation analysis on node features and topology, leaving edge feature importance for future work.

4.2 Global Level Explanations

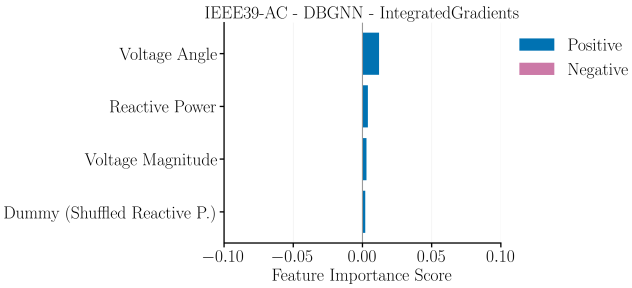
Validating Explanations on Static Power Flow. We use the synthetic IEEE39-AC dataset (Section 3.1.1) to validate against known AC power flow physics (Eq. 1). Reactive power Q_i does not appear in the equation for active power P_i, though it may correlate with other variables. IEEE39-AC serves as a controlled validation environment with analytical ground truth from Eq. 1. The full explanation pipeline is applied only to FRT-Stability, where no closed-form ground truth exists.

Figure 2 validates Integrated Gradients on IEEE39-AC at multiple scales. Locally (node 16), voltage angle is correctly identified as the dominant feature while the dummy shuffled-Q feature receives zero importance, confirming the method distinguishes causal from spurious features. Spatially, attributions are confined to the 1-hop neighborhood, matching the locality of the power flow equations.

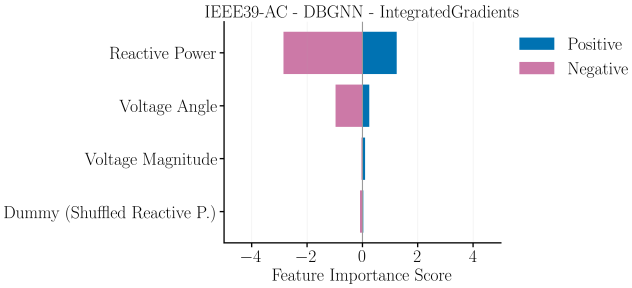
Aggregated across all nodes (Figure 2c), reactive power dominates due to its strong correlation ($r = 0.722$) with active power, illustrating how local and aggregated explanations can differ.



(a) Spatial explanation for node 16 showing 1-hop locality.



(b) Local feature importance for node 16.



(c) Aggregated feature importance across all nodes.

Figure 2: Validation of Integrated Gradients on IEEE39-AC. Local explanations (a, b) correctly identify voltage angle as dominant and assign zero importance to the dummy feature, matching power flow physics with 1-hop spatial locality. Aggregated explanations (c) reveal the model exploits the correlation between reactive power and active power, illustrating that local and aggregated explanations provide complementary perspectives.

As illustrated in Figure 2, this distinction between local and aggregated explanations is critical for interpreting the FRT-Stability results below.

Node Feature Importance on FRT-Stability. Despite different architectures and explanation methods, GBT and DBGNN reveal the same pattern (Figures 4 and 3): node type dominates predictions. GBT-SHAP ranks "Is Inverter Node" highest; Integrated Gradients and Shapley Value Sampling on DBGNN both assign strong positive attribution to "Is Load Node" and strong negative to "Is Inverter Node", confirming that load nodes correlate with stability while inverter nodes correlate with instability. The NF Parameter receives moderate importance in both methods, consistent with its causal role: larger τ_P corresponds to higher stability (Section 3.1.2).

Continuous power flow features (Aggregated Conductance, Susceptance, Shunt Susceptance, Active and Reactive Power) all receive minimal importance across all explanation methods. For DBGNN, the low importance of aggregated line properties is expected: the model receives unaggregated edge features directly and can learn its own aggregation. Even Active and Reactive Power contribute negligibly: within-type variation is less predictive than between-type differences.

The explanations reveal that both GBT and DBGNN rely primarily on type-level information: the model learns to distinguish inverters from loads and further differentiates inverter subtypes via the NF Parameter. While DBGNN incorporates graph structure, its explanations show the same type-level distinction as the non-graph GBT, paralleling the IEEE39-AC finding where correlation drives aggregated attributions. The agreement between GBT-SHAP and DBGNN explanations is telling: GBT uses no graph structure and shares no architectural assumptions with DBGNN, so the consistent dominance of node type confirms it reflects the dataset and physics, not a GNN artifact. Ablation results are provided in Appendix Table 5. The explanations show that node type is the primary signal; continuous power flow and network features play a secondary role because within-type variation is less predictive than between-type differences. This demonstrates that the explainers correctly distinguish between predictively useful features and features the model effectively ignores.

Node-type dominance reflects genuine physical differences: inverters and loads obey fundamentally different dynamic equations, with inverters acting as active control elements and loads as passive components. The NF Parameter importance within inverter subtypes provides causal confirmation: τ_P directly governs stability (Section 3.1.2), and recovering this within-type relationship confirms the result is not an artifact. Section 4.3 provides further evidence: when type is a poor predictor, the model shifts to neighborhood-based features, showing that topological context emerges where type alone is insufficient. These results highlight a fundamental distinction in XAI: post-hoc explanations are faithful to what the model has learned, not necessarily to what domain experts would prioritize. Varbella et al. [50] reach the same conclusion: even when explanation ground truth is available, explanations reflect model decision-making rather than human causal intuition. This is not a failure of the explainer; it is a diagnostic about what the model has learned.

Table 2: Comparison of skill scores (mean ± std over 5 seeds) across different GNN models on FRT-Stability, broken down by node type (corresponding R^2 scores in Table 8).

Model	Overall Skill	Inverter Skill	Load Skill
GCN	0.606 ± 0.022	0.710 ± 0.032	0.523 ± 0.014
TAG	0.700 ± 0.004	0.791 ± 0.006	0.626 ± 0.009
GATv2	0.766 ± 0.012	0.823 ± 0.005	0.721 ± 0.023
DBGNN	0.903 ± 0.002	0.919 ± 0.005	0.890 ± 0.003

The agreement between Integrated Gradients and Shapley Value Sampling, two methods with different theoretical foundations, validates both sets of attributions. We focus on Integrated Gradients in subsequent analyses: it is substantially faster, and its results are consistent with all other gradient-based variants (Saliency, InputX-Gradient, GuidedBackpropagation, Deconvolution; see Figure 8 in the Appendix).

4.3 Local Level Explanations

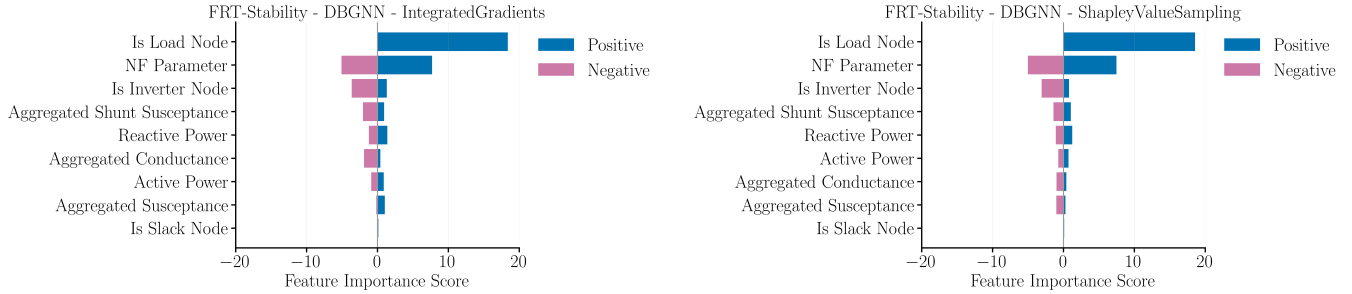
Detailed Node Feature Importance on FRT-Stability. Table 2 shows inverter dynamics are learned with higher skill than load dynamics across all models. For instance, DBGNN achieves 0.919 for inverters vs. 0.890 for loads, while GCN shows a larger gap (0.710 vs. 0.523). This is attributable to the feature space: inverters possess an additional categorical feature, the NF parameter, distinguishing three operational types, which is absent for loads.

To understand node-level decision-making, we analyze extreme outliers: the most stable inverter and least stable load from each test graph. Predicting these outliers is substantially harder than predicting typical nodes, with MSE ratios ranging from 7× to 33× depending on model and node type (detailed breakdown in Table 4).

Figure 5 shows a k-hop Integrated Gradients analysis that reveals the mechanism behind outlier predictions. While typical nodes show self-dominance (approximately 45% at 0-hop), extreme outliers shift toward neighbor-dominance. In these cases, the 1-hop neighborhood influence exceeds the node's own features: when a node deviates from its class norm, the model compensates by relying more on local topology. Impact decays rapidly beyond 1-hop, confirming that the DBGNN captures the localized nature of transient grid disturbances. The topological composition of each neighborhood (Table 6; see Appendix E) explains this behavior: least stable loads are surrounded by 84% inverter neighbors, while most stable inverters are embedded among 77% load neighbors. The model thus uses 1-hop topology to detect local stability while applying a penalty based on the inverter class's generally high instability probability.

4.4 Counterfactual Reasoning with Post-hoc Evaluation

In the previous section, we observed that GNN models capture distinct explanation patterns for extreme stability states. However, attribution methods are *passive*. They reveal which features influence a prediction but not how to modify the system toward a desired outcome.



(a) Integrated Gradients **(b) Shapley Value Sampling**

Figure 3: Aggregated node feature importance for the DBGNN on FRT-Stability. Both methods consistently identify node type as the dominant feature; continuous power flow features receive minimal importance.

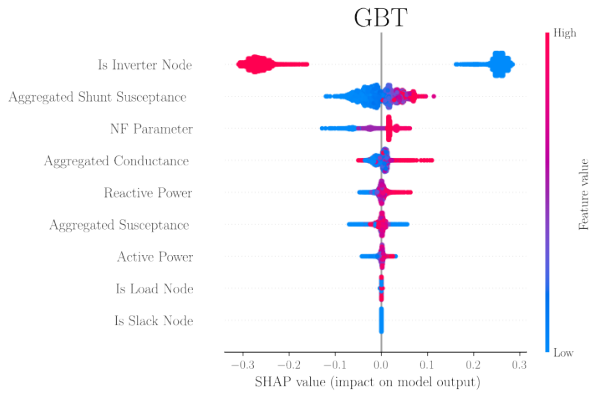


Figure 4: SHAP explanation for the GBT model on FRT-Stability, showing "Is Inverter Node" as the dominant feature.

To our knowledge, this is the first GNN-guided counterfactual topology screening for stability prediction. We frame it explicitly as a *candidate generation* tool: its purpose is to reduce thousands of potential interventions to a small set of locally consistent candidates in seconds, providing a tractable starting point for subsequent physics-based validation rather than replacing it. The 1-hop locality of spatial attributions (Section 4.2) motivates this local search strategy.

The model's generalization to unseen topologies justifies applying the framework described in Section 3.5. We evaluated edge removal strategies across 150 FRT-Stability test grids. The framework identifies the critical node v^* (minimum predicted stability) and searches for valid edge removals that improve its stability while satisfying connectivity, global floor, and local stability constraints ($\delta \leq 0.05$) as introduced in 3.5.

Effects of Edge Removal on the FRT-Stability Prediction. The results of this counterfactual search are visualized in Figure 6. Across the test cases, the framework consistently identifies edge removal interventions that increase the predicted stability of the most vulnerable node, occasionally exceeding ground-truth stability. The mean stability score increases by approximately 45% over the baseline prediction. Large error bars reflect topology-dependent variance.

Figure 7 demonstrates that our GNN-guided interventions are highly local. The impact of removing an edge decays exponentially with graph distance: over 80% of the total prediction change is concentrated within the 0-hop (incident nodes) and 1-hop radius. Nodes at 2+ hops show negligible deviation. This aligns with the locality bias observed in our post-hoc explanation analysis (Section 4.2), providing further evidence of localized model representations.

The reported stability gains are *model-predicted improvements*. Confirming whether these translate to true FRT outcomes requires full dynamic re-simulation, which takes several hours per grid versus seconds for GNN re-evaluation. Simulation-in-the-loop validation is therefore outside the scope of this work by design: the framework is not intended as a replacement for physics-based analysis, but as a filter that makes such analysis tractable by identifying the most promising candidates first. The current strategy targets the least stable node, which in FRT-Stability datasets is typically an inverter; future work should explore alternative target selection criteria and intervention types (e.g., node-level actions or feature modifications), and quantify the translation rate between predicted and simulated stability gains on a held-out subset.

5 Conclusion

This paper investigates GNN explainability for dynamic stability prediction, a setting where node type is the dominant predictive signal. This reflects a physically meaningful distinction, as inverters and loads obey fundamentally different dynamics, while continuous features such as power flow parameters play a secondary role. Systematic explainability analysis addresses two questions: do post-hoc explanations reflect model behavior, and does that behavior align with physics? This determines when AI predictions can guide operations and when physics-based validation is necessary.

The DBGNN achieves a skill score of 0.903, far outperforming non-graph methods (skill $\leq$ 0.03). Post-hoc explanation analysis, validated on IEEE39-AC and applied to FRT-Stability, reveals that performance stems primarily from type-level information: the model learns to distinguish inverters from loads, while continuous features play a secondary role. Yet this does not invalidate operational utility. GNNs exploit topological context unavailable to non-graph models: the same node type yields different predictions depending on neighborhood composition. For extreme cases, the

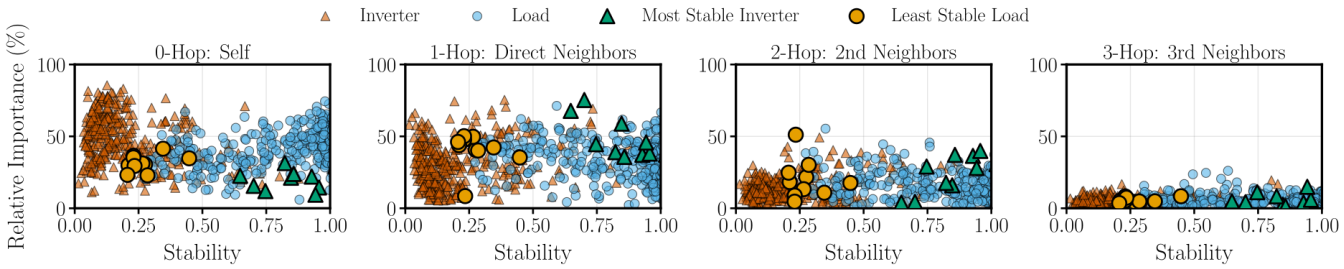


Figure 5: Integrated Gradients k-hop influence analysis for the DBGNN on FRT-Stability. Extreme outliers (most stable inverters, least stable loads) shift from self-dominance toward neighbor-dominance at 1-hop, while influence decays to negligible beyond 2 hops.

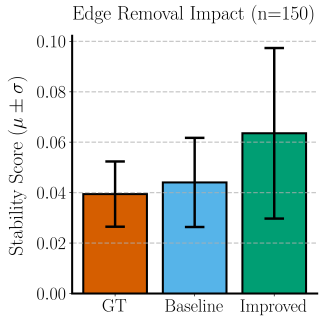


Figure 6: Edge removal impact on FRT-Stability prediction across 150 test grids. The Improved bar shows predicted stability of the critical node after the optimal intervention; high variance reflects grid-specific effectiveness.

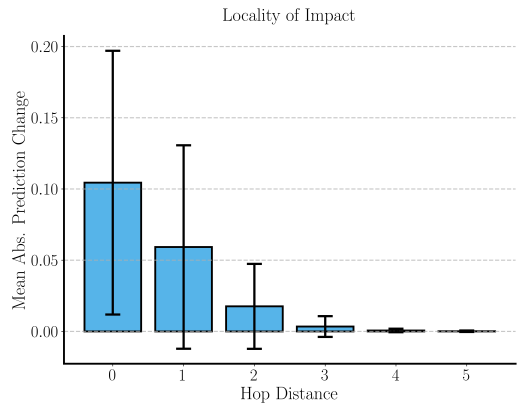


Figure 7: Mean absolute change in predicted p_{frt} by hop distance from the removed edge (DBGNN, FRT-Stability). Over 80% of the effect is concentrated within hops 0–1.

1-hop neighborhood contributes more than the node's own features, reflecting how local topology affects disturbance propagation. Influence decays rapidly beyond 1-hop, consistent with the localized nature of transient disturbances. This reframes deployment from "Does the model work?" to "When can we trust it?"

Our architectural comparison reveals that topology information is essential: GNNs require expanded spatial context, explicit edge features, and wave-like message propagation to capture transient dynamics. The counterfactual framework shows that explainability can guide optimization: edge removal interventions improve predicted stability at vulnerable nodes by 45% on average while maintaining safety constraints. The highly localized effects (82% within 0–1 hops) align with the attribution analysis, confirming localized model representations.

Limitations and Future Work. Our analysis relies on synthetic datasets where ground truth derives from simulation rather than operational measurements. While FRT-Stability provides realistic topologies, the scarcity of real-world fault data limits validation under actual conditions. To our knowledge, FRT-Stability is the most realistic publicly available benchmark for GNN-based dynamic stability prediction in inverter-dominated grids. The deeper challenge is that labels alone cannot distinguish whether the model exploits type patterns or physical causality.

Future work should pursue three directions: first, benchmark datasets with verified explanation ground truth and fidelity metrics for cases where it is unavailable, as adapting frameworks such as GraphFramEx [3] to node-level regression remains an open problem; second, complete explanations covering node, edge, and graph-level attributions with physics-informed architectures; and third, counterfactual interventions extended to node-level actions and feature modifications.

Data and Code Availability. Full code is available at https://github.com/KIT-IAI-DRACOS/explanations-of-gnns-for-dynamic-stability-assessment. Dataset and trained models are available at https://doi.org/10.5281/zenodo.20128791 and https://doi.org/10.5281/zenodo.20128840.

Acknowledgments

We gratefully acknowledge funding from the Helmholtz Association Initiative and Networking Fund through Helmholtz AI (grant no. VH-NG-1727) and the Federal Ministry for Economic Affairs and Energy (BMWE, FKZ 03EI1092C). Computational resources were provided by the state of Baden-Württemberg through bwHPC and the Helmholtz Association via the HAICORE@KIT partition. We thank Dr. Xiao Li for valuable discussions.

References

[1] Amina Adadi and Mohammed Berrada. 2018. Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI). *IEEE Access* 6 (2018), 52138–52160. doi:10.1109/ACCESS.2018.2870052

[2] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, Anchorage AK USA, 2623–2631. doi:10.1145/3292500.3330701

[3] Kenza Amara, Rex Ying, Zitao Zhang, Zhihao Han, Yinan Shan, Ulrik Brandes, Sebastian Schemm, and Ce Zhang. 2024. GraphFramEx: Towards Systematic Evaluation of Explainability Methods for Graph Neural Networks. doi:10.48550/arXiv.2206.09677

[4] Shaked Brody, Uri Alon, and Eran Yahav. 2022. How Attentive are Graph Attention Networks? doi:10.48550/arXiv.2105.14491

[5] Margarita Bugueño, Russa Biswas, and Gerard de Melo. 2024. Graph-based explainable AI: a comprehensive survey. (2024). https://hal.science/hal-04660442v1

[6] Anna Büttner, Jürgen Kurths, and Frank Hellmann. 2022. Ambient forcing: sampling local perturbations in constrained phase spaces. *New Journal of Physics* 24, 5 (May 2022), 053019. doi:10.1088/1367-2630/ac6822

[7] Anna Büttner, Anton Plietzsch, Mehrnaz Anvari, and Frank Hellmann. 2023. A framework for synthetic power system dynamics. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 33, 8 (Aug. 2023), 083120. doi:10.1063/5.0155971

[8] Anna Büttner, Hans Würfel, Sebastian Liemann, Johannes Schiffer, and Frank Hellmann. 2025. Complex-Phase, Data-Driven Identification of Grid-Forming Inverter Dynamics. doi:10.48550/arXiv.2409.17132

[9] Dibaloke Chanda and Nasim Yahya Soltani. 2023. A Heterogeneous Graph-Based Multi-Task Learning for Fault Event Diagnosis in Smart Grid. http://arxiv.org/abs/2309.09921

[10] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, San Francisco California USA, 785–794. doi:10.1145/2939672.2939785

[11] Deutsche Energie-Agentur GmbH (dena). 2012. Ausbau- und Innovationsbedarf der Stromverteilnetze in Deutschland bis 2030 (kurz: dena-Verteilnetzstudie). https://www.dena.de/fileadmin/dena/Dokumente/Pdf/9100_dena-Verteilnetzstudie_Abschlussbericht.pdf

[12] Jian Du, Shanghang Zhang, Guanhang Wu, Jose M. F. Moura, and Soummya Kar. 2018. Topology Adaptive Graph Convolutional Networks. doi:10.48550/arXiv.1710.10370

[13] Jonas Egerer and DIW Berlin Deutsches Institut für Wirtschaftsforschung. 2016. *Open Source Electricity Model for Germany (ELMOD-DE)*. Text. DIW Berlin. https://www.diw.de/de/diw_01.c.528929.de/publikationen/data_documentation/2016_0083/open_source_electricity_model_for_germany__elmod-de.html

[14] ENTSO-E. 2018. *All continental europe and nordic tsos' proposal for assumptions and a cost benefit analysis methodology in accordance with article 156(11) of the commission regulation (EU) 2017/1485 of 2 august 2017 establishing a guideline on electricity transmission system operation*. Technical Report. European Network of Transmission System Operators for Electricity, Brussels, Belgium. https://www.entsoe.eu

[15] Matthias Fey and Jan E. Lenssen. 2019. Fast graph representation learning with PyTorch Geometric. In *ICLR workshop on representation learning on graphs and manifolds*.

[16] Matthias Fey, Jinu Sunil, Akihiro Nitta, Rishi Puri, Manan Shah, Blaž Stojanovič, Ramona Bendias, Alexandria Barghi, Vid Kocijan, Zecheng Zhang, Xinwei He, Jan Eric Lenssen, and Jure Leskovec. 2025. PyG 2.0: Scalable Learning on Real World Graphs. doi:10.48550/arXiv.2507.16991 Version Number: 2.

[17] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. 2017. Neural Message Passing for Quantum Chemistry. In *Proceedings of the 34th International Conference on Machine Learning*. PMLR, 1263–1272. tex.eventtitle: International Conference on Machine Learning.

[18] C. Grigg, P. Wong, P. Albrecht, R. Allan, M. Bhavaraju, R. Billinton, Q. Chen, C. Fong, S. Haddad, S. Kuruganty, W. Li, R. Mukerji, D. Patton, N. Rau, D. Reppen, A. Schneider, M. Shahidehpour, and C. Singh. 1999. The IEEE Reliability Test System-1996. A report prepared by the Reliability Test System Task Force of the Application of Probability Methods Subcommittee. *IEEE Transactions on Power Systems* 14, 3 (Aug. 1999), 1010–1020. doi:10.1109/59.780914

[19] S. Gu, J. Qiao, W. Shi, F. Yang, X. Zhou, and Z. Zhao. 2023. Multi-task transient stability assessment of power system based on graph neural network with interpretable attribution analysis. *Energy Reports* 9 (Oct. 2023), 930–942. doi:10.1016/j.egyr.2023.05.159

[20] A.K. Khamis and M. Agamy. 2023. Three-Phase Inverter Dynamics Predictions Using GNN-based Regression Model. In *IEEE Energy Convers. Congr. Expo., ECCE*. Institute of Electrical and Electronics Engineers Inc., 6289–6294. doi:10.1109/ECCE53617.2023.10362156

[21] Thomas N. Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. tex.eventtitle: International Conference on Learning Representations.

[22] Raphael Kogler, Anton Plietzsch, Paul Schultz, and Frank Hellmann. 2022. Normal Form for Grid-Forming Power Grid Actors. *PRX Energy* 1, 1 (June 2022), 013008. doi:10.1103/PRXEnergy.1.013008

[23] Narine Kokhlikyan, Vivek Miglani, Miguel Martin, Edward Wang, Bilal Alsallakh, Jonathan Reynolds, Alexander Melnikov, Natalia Kliushkina, Carlos Araya, Siqi Yan, and Orion Reblitz-Richardson. 2020. Captum: A unified and generic model interpretability library for PyTorch. doi:10.48550/arXiv.2009.07896

[24] W. Liao, B. Bak-Jensen, J.R. Pillai, Y. Wang, and Y. Wang. 2022. A Review of Graph Neural Networks and Their Applications in Power Systems. *Journal of Modern Power Systems and Clean Energy* 10, 2 (2022), 345–360. doi:10.35833/MPCE.2021.000058

[25] Sebastian Liemann, Lia Strenge, Paul Schultz, Holm Hinners, Johannis Porst, Marcel Sarstedt, and Frank Hellmann. 2021. Probabilistic Stability Assessment for Active Distribution Grids. In *2021 IEEE Madrid PowerTech*. IEEE, Institute of Electrical and Electronics Engineers, Madrid, 1–6. doi:10.1109/PowerTech46648.2021.9494855

[26] J. Liu, C. Yao, and L. Chen. 2022. Time-Adaptive Transient Stability Assessment Based on the Gating Spatiotemporal Graph Neural Network and Gated Recurrent Unit. *Frontiers in Energy Research* 10 (2022). doi:10.3389/fenrg.2022.885673

[27] Yuxiao Liu, Ning Zhang, Dan Wu, Audun Botterud, Rui Yao, and Chongqing Kang. 2020. Guiding Cascading Failure Search with Interpretable Graph Convolutional Network. http://arxiv.org/abs/2001.11553

[28] Zhao Liu, Zhenhuan Ding, Xiaoge Huang, and Pei Zhang. 2023. An online power system transient stability assessment method based on graph neural network and central moment discrepancy. *Frontiers in Energy Research* 11 (Feb. 2023), 1082534. doi:10.3389/fenrg.2023.1082534

[29] Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. doi:10.48550/arXiv.1711.05101

[30] Zhe Lv, Bin Wang, Qinglai Guo, Haotian Zhao, Zhengcheng Wang, and Hongbin Sun. 2025. Short-Term Voltage Stability Assessment Based on Heterogeneous Edge-Integrated Graph Attention Network. *IEEE Transactions on Power Systems* 40, 1 (Jan. 2025), 421–434. doi:10.1109/TPWRS.2024.3401742

[31] Federico Milano, Florian Dörfler, Gabriela Hug, David J Hill, and Gregor Verbič. 2018. Foundations and challenges of low-inertia systems. In *2018 power systems computation conference (PSCC)*. IEEE, 1–25.

[32] Christian Nauck, Anna Büttner, Sebastian Liemann, Frank Hellmann, and Michael Lindner. 2026. Predicting the Fault-Ride-Through Probability of Inverter-Dominated Power Grids Using Machine Learning. *IET Generation, Transmission & Distribution* 20, 1 (2026), e70264. doi:10.1049/gtd2.70264

[33] Christian Nauck, Rohan Gorantla, Michael Lindner, Konstantin Schürholt, Antonia S. J. S. Mey, and Frank Hellmann. 2024. Dirac–Bianconi Graph Neural Networks – Enabling Non-Diffusive Long-Range Graph Predictions. http://arxiv.org/abs/2407.12419

[34] Christian Nauck, Michael Lindner, Konstantin Schürholt, and Frank Hellmann. 2023. Toward dynamic stability assessment of power grid topologies using graph neural networks. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 33, 10 (Oct. 2023), 103103. doi:10.1063/5.0160915

[35] Christian Nauck, Michael Lindner, Konstantin Schürholt, Haoming Zhang, Paul Schultz, Jürgen Kurths, Ingrid Isenhardt, and Frank Hellmann. 2022. Predicting basin stability of power grids using graph neural networks. *New Journal of Physics* 24, 4 (April 2022), 043041. doi:10.1088/1367-2630/ac54c9

[36] Damian Owerko, Fernando Gama, and Alejandro Ribeiro. 2022. Unsupervised Optimal Power Flow Using Graph Neural Networks. doi:10.48550/arXiv.2210.09277 Version Number: 1.

[37] Laurent Pagnier and Michael Chertkov. 2021. Physics-Informed Graphical Neural Network for Parameter & State Estimations in Power Systems. doi:10.48550/arXiv.2102.06349 Version Number: 1.

[38] M. A. Pai. 2012. *Energy function analysis for power system stability*. Springer Science & Business Media.

[39] H. Raum, T. Schnake, F. Hellmann, J. Kurths, and C. Nauck. 2025. Explainable AI for analyzing the decision of GNNs at predicting dynamic stability of complex oscillator networks. *Chaos* 35, 11 (2025). doi:10.1063/5.0278469

[40] Joeri Rogelj, Gunnar Luderer, Robert C. Pietzcker, Elmar Kriegler, Michiel Schaeffer, Volker Krey, and Keywan Riahi. 2015. Energy system transformations for limiting end-of-century warming to below 1.5 °C. *Nature Climate Change* 5, 6 (June 2015), 519–527. doi:10.1038/nclimate2572

[41] T. Konstantin Rusch, Michael M. Bronstein, and Siddhartha Mishra. 2023. A Survey on Oversmoothing in Graph Neural Networks. doi:10.48550/arXiv.2303.10993 Version Number: 1.

[42] Johannes Schiffer, Romeo Ortega, Alessandro Astolfi, Jörg Raisch, and Tevfik Sezi. 2014. Conditions for stability of droop-controlled inverter-based microgrids. *Automatica* 50, 10 (Oct. 2014), 2457–2469. doi:10.1016/j.automatica.2014.08.009

[43] Benjamin Schäfer, Thiemo Pesch, Debsankha Manik, Julian Gollenstede, Guosong Lin, Hans-Peter Beck, Dirk Witthaut, and Marc Timme. 2022. Understanding Braess' Paradox in power grids. *Nature Communications* 13, 1 (Sept. 2022), 5396. doi:10.1038/s41467-022-32917-6

[44] Lloyd S. Shapley. 1953. A value for n-person games. In *Contributions to the theory of games*, H. W. Kuhn and A. W. Tucker (Eds.). Vol. 2. Princeton University Press, Santa Monica, CA, 307–317.
[45] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. doi:10.48550/arXiv.1312.6034
[46] I.M Sobol'. 1967. On the distribution of points in a cube and the approximate evaluation of integrals. *U. S. S. R. Comput. Math. and Math. Phys.* 7, 4 (Jan. 1967), 86–112. doi:10.1016/0041-5553(67)90144-9
[47] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. 2015. Striving for Simplicity: The All Convolutional Net. doi:10.48550/arXiv.1412.6806
[48] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. 2017. Axiomatic Attribution for Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*. PMLR, Sydney, Australia, 3319–3328. https://proceedings.mlr.press/v70/sundararajan17a.html
[49] Leon Thurner, Alexander Scheidler, Florian Schäfer, Jan-Hendrik Menke, Julian Dollichon, Friederike Meier, Steffen Meinecke, and Martin Braun. 2018. pandapower - an Open Source Python Tool for Convenient Modeling, Analysis and Optimization of Electric Power Systems. *IEEE Transactions on Power Systems* 33, 6 (Nov. 2018), 6510–6521. doi:10.1109/TPWRS.2018.2829021
[50] A. Varbella, K. Amara, B. Gjorgiev, M. El-Assady, and G. Sansavini. 2024. PowerGraph: A power grid benchmark dataset for graph neural networks. In *Adv. neural inf. proces. syst.*, Globerson A., Mackey L., Belgrave D., Fan A., Paquet U., Tomczak J., and Zhang C. (Eds.), Vol. 37. Neural information processing systems foundation. https://www.scopus.com/inward/record.uri?eid=2-s2.0-105000523867&partnerID=40&md5=db6fe0678e8d508cd69f6f0bad2675ed
[51] Dirk Witthaut, Frank Hellmann, Jürgen Kurths, Stefan Kettemann, Hildegard Meyer-Ortmanns, and Marc Timme. 2022. Collective nonlinear dynamics and self-organization in decentralized power grids. *Reviews of Modern Physics* 94, 1 (Feb. 2022), 015005. doi:10.1103/RevModPhys.94.015005
[52] Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. 2019. GNNExplainer: Generating explanations for graph neural networks. In *Advances in neural information processing systems*, Vol. 32. Curran Associates, Inc., Vancouver, Canada, 9240–9251.
[53] Hao Yuan, Haiyang Yu, Shurui Gui, and Shuiwang Ji. 2023. Explainability in Graph Neural Networks: A Taxonomic Survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 45, 5 (May 2023), 5782–5799. doi:10.1109/TPAMI.2022.3204236
[54] Matthew D. Zeiler and Rob Fergus. 2013. Visualizing and Understanding Convolutional Networks. doi:10.48550/arXiv.1311.2901
[55] Hai-Feng Zhang, Xin-Long Lu, Xiao Ding, and Xiao-Ming Zhang. 2024. Physics-informed line graph neural network for power flow calculation. *Chaos: An Interdisciplinary Journal of Nonlinear Science* 34, 11 (Nov. 2024), 113123. doi:10.1063/5.0235301
[56] Ke Zhang, Peidong Xu, Tianlu Gao, and Jun Zhang. 2021. A Trustworthy Framework of Artificial Intelligence for Power Grid Dispatching Systems. In *2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI)*. IEEE, Beijing, China, 418–421. doi:10.1109/DTPI52967.2021.9540198

A Dataset Details

A.1 FRT-Stability Dataset

Data Splitting and Batching. The 1,000 synthetic grids were divided into training (grids 1–700, 70%), validation (grids 701–850, 15%), and test sets (grids 851–1,000, 15%). This graph-level split ensures that test performance reflects generalization to unseen network topologies rather than merely unseen operating points. Training samples were shuffled between epochs (batch size 128), while validation and test sets maintained fixed ordering (batch size 500) to ensure reproducible evaluation.

Dynamic Simulations. Dynamic simulations use higher-order implicit differential-algebraic equation solvers with error tolerances ensuring standard errors of ±0.02. The complete dataset required approximately 50,000 CPU hours to generate.

Ride-Through Curves. Voltage ride-through curves follow [25] for under-voltages and German transmission system requirements [13] for over-voltages. Frequency thresholds are symmetric at $[-2, +2]$ Hz. While post-clearance states are evaluated per bus, ride-through criteria apply to *all* buses simultaneously, making p_{frt} a system-wide stability measure.

Post-Clearance State Sampling. For each bus, 1,000 post-clearance states are generated using Sobol sequences [46] and the ambient forcing algorithm [6]. Voltage magnitudes are sampled uniformly in $[0, 1]$ p.u., phase angles in $[0, 2\pi]$, and frequency deviations in $[-1, +1]$ Hz for grid-forming components.

Inverter Parameterization. Three parameter sets for grid-forming inverters vary only in the time constant τ_P of the active power low-pass filter in droop control: NF1 ($\tau_P = 0.5$ s), NF2 ($\tau_P = 1.0$ s), and NF3 ($\tau_P = 5.0$ s) [6, 22, 42]. Buses with larger time constants are expected to have higher stability due to their faster response during the critical first milliseconds following fault clearance. Grid-following inverters and loads are modeled as PQ-buses, while transmission lines use the Pi-model with parameters for 380 kV systems from the German energy agency [11].

Table 3: Node and edge features for the FRT-Stability dataset.

Type	Feature	Description
Node	P	Active power
Node	Q	Reactive power
Node	τ_P	NF parameter (inverter time constant; 0 for loads/slack)
Node	G_k	Aggregated line conductance $\sum_m G_{km}$
Node	B_k	Aggregated line susceptance $\sum_m B_{km}$
Node	$B_{k,\text{sh}}$	Aggregated shunt susceptance $\sum_m B_{km,\text{sh}}$
Node	`Is Inverter`	Categorical node-type indicator
Node	`Is Slack`	Categorical node-type indicator
Node	`Is Load`	Categorical node-type indicator
Edge	G_{ij}	Line conductance
Edge	B_{ij}	Line susceptance
Edge	$B_{ij,\text{sh}}$	Line shunt susceptance
Edge	P_{ij}	Line active power
Edge	Q_{ij}	Line reactive power

Node and Edge Features.

A.2 IEEE39-AC Dataset Generation

A.2.1 Physical Foundation: Steady-State AC Power Flow. The target variable, active power P_i, is governed by the non-linear AC power flow equations (Eq. 1), where V_i and θ_i denote voltage magnitude and phase angle at bus i, and G_{ij} and B_{ij} are conductance and susceptance from the admittance matrix $\mathbf{Y}$.

A.2.2 Stochastic Parameterization. To sample diverse operating conditions, base active power was randomized:

$$P_i^{(0)} \sim \mathcal{U}\left(P_i^{\text{base}} - \Delta P,\ P_i^{\text{base}} + \Delta P\right), \tag{5}$$

where P_i^{base} is the nominal active power and ΔP controls perturbation magnitude.

A.2.3 Newton–Raphson Power Flow. For each randomized configuration, AC power flow was solved using Newton–Raphson:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \mathbf{J}^{-1}(\mathbf{x}^{(k)})\,\mathbf{F}(\mathbf{x}^{(k)}), \tag{6}$$

where $\mathbf{x} = [\boldsymbol{\theta}, \mathbf{V}]^\top$ is the state vector, $\mathbf{F}(\cdot)$ denotes power flow mismatch equations, and $\mathbf{J}$ is the Jacobian matrix.

A.2.4 Feature Mapping and Normalization. Converged state variables were mapped to nodal feature vectors $\mathbf{h}_i$ with additive Gaussian noise:

$$\tilde{\mathbf{h}}_i = \mathbf{h}_i + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \sigma^2\mathbf{I}). \tag{7}$$

Features were standardized following the procedure described in Appendix B.1.

B Model and Training Details

B.1 Training Procedure

Feature Preprocessing. All features were standardized using Z-score normalization (zero mean, unit variance) computed on the training set and applied to validation and test sets. For GBT, raw features were used without scaling as tree-based methods are invariant to monotonic transformations.

Optimization. All models were trained using the AdamW optimizer [29] with momentum 0.9 and model-specific weight decay ($\sim 10^{-7}$). Gradient clipping with maximum norm and value of 100 prevented gradient explosion. The loss function is mean squared error (MSE), evaluated on all nodes except slack buses, as slack buses serve as voltage references and have trivially stable behavior ($p_{\text{frt}} \approx 1$).

Training proceeded for a maximum of 100 epochs with early stopping based on validation performance. Early stopping used patience of 50 epochs (no improvement threshold: 0.001) and a grace period of 50 epochs to prevent premature termination. Training was conducted on NVIDIA GPUs with mixed precision (float32). Each model was trained across five random seeds (seeds 1–5) to quantify variance.

B.2 Hyperparameter Optimization

Model-specific hyperparameters were optimized on the FRT-Stability dataset using Optuna [2] with 100 trials per architecture. The same configurations were applied to IEEE39-AC, where they achieved strong performance without further tuning (Table 7). The search space included learning rates (log-uniform 10^{-5} to 10^{-2}), hidden dimensions ($\{32, 64, 128, 256\}$), dropout rates (uniform 0 to 0.5), number of layers ($\{2, 3, 4\}$ for GCN/GATv2, fixed at 2 for DBGNN), and architecture-specific parameters (attention heads for GATv2, polynomial order K for TAG, time steps for DBGNN). The optimization objective was validation R^2 score (equivalently, skill score since the baseline is fixed).

The final optimized configurations are:

- **DBGNN**: 2 layers, 120 hidden dimensions (nodes and edges), 30 time steps per layer, learning rate 10^{-5}, OneCycleLR scheduler (cosine annealing, max LR 0.01, division factor 25, final division factor 10,000), dropout 0.015 (nodes) and 0.002 (edges), LeakyReLU activation, skip connections enabled. Total parameters: 147,961.
- **GATv2**: 3 layers, 64 hidden dimensions, 6 attention heads, learning rate 7.8×10^{-4}, StepLR scheduler (step size 20, $\gamma = 0.5945$), dropout 0.2499 (nodes) and 0.1349 (edges), ELU activation (nodes), GELU activation (edges).
- **GCN**: 3 layers, 64 hidden dimensions, learning rate 1.96×10^{-3}, ReduceLROnPlateau scheduler (patience 20, factor 0.5066, step size 40), dropout 0.1276 (nodes) and 0.1 (edges), LeakyReLU activation (nodes), ReLU activation (edges).
- **TAG**: 3 layers, 64 hidden dimensions, $k = 3$ hop neighborhoods, learning rate 1.96×10^{-3}, ReduceLROnPlateau scheduler (patience 20, factor 0.5066), dropout 0.1276 (nodes) and 0.1 (edges), LeakyReLU activation.

B.3 GNN Architecture Details

B.3.1 Message Passing Framework. The message-passing framework [17] provides the foundation for GNN architectures. A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of nodes $\mathcal{V}$ and edges $\mathcal{E}$. Node features $\mathbf{x}_v \in \mathbb{R}^{n_f}$ and topology (adjacency matrix $\mathbf{A}$) form the input. Let $\mathcal{N}_v$ denote the neighborhood of node v. Node embeddings are updated via:

$$\mathbf{h}_v^{(l+1)} = \sigma\left(\mathbf{W}^{(l)} \cdot \frac{1}{|\mathcal{N}_v|} \sum_{u \in \mathcal{N}_v} \mathbf{h}_u^{(l)} + \mathbf{B}^{(l)} \cdot \mathbf{h}_v^{(l)}\right), \tag{8}$$

where $\mathbf{h}_v^{(l)}$ is the embedding at layer l, $\mathbf{W}^{(l)}, \mathbf{B}^{(l)}$ are learnable weights, and σ is a non-linear activation.

B.3.2 DBGNN Dynamics. The Dirac-Bianconi Graph Neural Network (DBGNN) [33] differs from conventional message-passing neural networks in its approach to information propagation. Instead of relying on diffusive mechanisms similar to the heat equation, DBGNNs emphasize long-range interactions through the topological Dirac equation.

The update equations for node and edge features are given in Eq. (3), where:

- $\mathbf{x}_i(t) \in \mathbb{R}^{d_n}$: Node feature vector of node i at time step t
- $\mathbf{e}_{ij}(t) \in \mathbb{R}^{d_e}$: Edge feature vector for edge (i, j) at time step t
- $\mathcal{N}_i$: The set of neighbors of node i
- $\mathbf{W}^{ne} \in \mathbb{R}^{d_n \times d_e}$: Coupling matrix mapping edge features to node features
- $\mathbf{W}^{en} \in \mathbb{R}^{d_e \times d_n}$: Coupling matrix mapping node features to edge features
- $\mathbf{W}_\beta^n, \mathbf{W}_\beta^e$: Mass matrices regulating updates

For comparison, a simple MPNN with edge features follows:

$$\begin{aligned} \mathbf{x}_i(t+1) &= \mathbf{x}_i(t) + \mathbf{W}_n^{\text{MPNN}} \sum_{j \in \mathcal{N}_i} \mathbf{e}_{ij}(t) + \beta_n \mathbf{x}_i(t), \\ \mathbf{e}_{ij}(t) &= \mathbf{W}_e^{\text{MPNN}}\big(\mathbf{x}_i(t) - \mathbf{x}_j(t)\big). \end{aligned} \tag{9}$$

Unlike DBGNNs, this formulation lacks temporal dynamics in edge features and does not incorporate a feedback mechanism between node and edge states.

C Explainability Method Details

Graph explanation methods can be categorized based on their underlying approach [5, 53]: **Perturbation-Based Methods** evaluate feature importance by modifying the input graph and observing

changes in model predictions; **Gradient-Based Methods** leverage backpropagation to assess how infinitesimal changes to input features affect model predictions; and **Game Theory-Based Methods** treat each input feature as a player contributing to the final prediction, distributing the prediction fairly among features based on marginal contributions across all possible feature coalitions.

C.1 Gradient-Based Methods

We evaluated five gradient-based attribution methods from the Captum library: *Integrated Gradients* [48], *Saliency* [45], *InputXGradient*, *GuidedBackpropagation* [47], and *Deconvolution* [54]. Figure 8 shows aggregated feature attributions for all five methods alongside Shapley Value Sampling.

Integrated Gradients. First introduced by [48], Integrated Gradients compute the integral of gradients with respect to inputs along a path from a given baseline to the actual input. Formally, given a function $F : \mathbb{R}^n \to [0, 1]$ representing a deep network, where $x \in \mathbb{R}^n$ is the input and $x' \in \mathbb{R}^n$ is the baseline, attributions are computed relative to this baseline. Attribution methods rely on sensitivity analysis by perturbing inputs or network parameters and observing changes in predictions. The integrated gradient for the i^{th} feature of an input x and baseline x' is:

$$\text{IG}_i(x) = \left(x_i - x_i'\right) \cdot \int_{\alpha=0}^{1} \frac{\partial F\left(x' + \alpha\left(x - x'\right)\right)}{\partial x_i} d\alpha, \tag{10}$$

where $\frac{\partial F(x)}{\partial x_i}$ is the gradient of F along the i^{th} dimension at x, and α is the scaling coefficient. This integral is approximated using Riemann summation or Gauss-Legendre quadrature (50 steps in our implementation).

Saliency. Saliency maps, first introduced in [45], are a simple explainability technique originally developed for deep convolutional networks. This method computes input attributions by taking the gradient of the output with respect to the input. Based on a first-order Taylor expansion of the model at the input, the gradient coefficients represent the linearized model's feature importance. The absolute values of these coefficients can be interpreted as feature importance scores.

InputXGradient. InputXGradient extends the saliency approach by multiplying the gradient of the output with respect to the input by the input feature values. The intuition is derived from linear models, where gradients correspond to feature coefficients. The product of an input feature and its gradient represents the total contribution of that feature to the model's output.

Guided Backpropagation. Guided Backpropagation [47] computes the gradient of the target output with respect to the input while modifying the backpropagation of ReLU activations. Specifically, only non-negative gradients are propagated, ensuring that only positive influences are considered. Originally developed for convolutional networks, this method can be applied generically, similar to *Deconvolution*.

Deconvolution. Deconvolution [54] is similar to Guided Backpropagation but differs in how ReLU functions are handled. Instead of applying the ReLU operation to input gradients, it is applied to output gradients before backpropagation. This alternative approach allows for a different perspective on feature importance and visualization.

All methods agree on the top features: Is Load Node dominates across every method, with NF Parameter and Is Inverter Node consistently ranking second and third. Power flow and network features (Reactive Power, Conductance, Susceptance) remain near zero throughout. The strongest consensus is between Integrated Gradients, InputXGradient, and Shapley Value Sampling, which produce nearly identical rankings and magnitudes.

Minor disagreements do not change the overall story. GuidedBackpropagation is the clearest outlier: it inverts the sign of Is Load Node and Is Inverter Node, a known artifact of that method caused by its modified backpropagation rule rather than a genuine feature attribution. Saliency shows a more pronounced negative score on Is Inverter Node, and Deconvolution assigns NF Parameter a positive rather than negative attribution; both still rank the same top three features. Shapley Value Sampling shows mixed positive and negative bars for NF Parameter, reflecting its different theoretical treatment of feature interactions, but its overall ranking matches the gradient consensus.

The same consistency holds on the IEEE39-AC dataset. Across all five gradient-based methods and Shapley Value Sampling, Reactive Power is unanimously the top feature (with some methods showing mixed positive and negative bars reflecting different node roles), Voltage Angle ranks second and is consistently negative, and Voltage Magnitude is negligible throughout. Critically, the dummy feature (shuffled Reactive Power) receives near-zero attribution in every method, confirming that the sanity check passes regardless of which attribution method is used.

Given the agreement across methods on both datasets, and given that Integrated Gradients satisfies the completeness and sensitivity axioms (attributions sum to the prediction difference and relevant features receive non-zero attribution), we use it as the primary explanation method throughout.

C.2 Game Theory-Based Methods

Shapley Value Sampling. Shapley values distribute model predictions among features based on marginal contributions:

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(n - |S| - 1)!}{n!} \left[F(S \cup \{i\}) - F(S)\right], \tag{11}$$

where N is the set of all features, S is a coalition (subset) of features, and $n = |N|$. Exact computation is intractable ($O(2^n)$ complexity). We use KernelSHAP, which approximates Shapley values by fitting a weighted linear model on feature coalitions.

C.3 GNNExplainer

GNNExplainer [52] identifies informative subgraphs and features by maximizing mutual information:

$$\max_{G_S, X_S} \text{MI}(Y, (G_S, X_S)) = H(Y) - H(Y \mid G = G_S, X = X_S), \tag{12}$$

where $G_S \subseteq G$ is a subgraph, X_S are masked features, Y is the prediction, and $H(\cdot)$ denotes entropy. We use edge mask temperature 1.0, feature mask temperature 1.0, and L_1 regularization coefficient 0.001. Results are consistent with other methods only when using

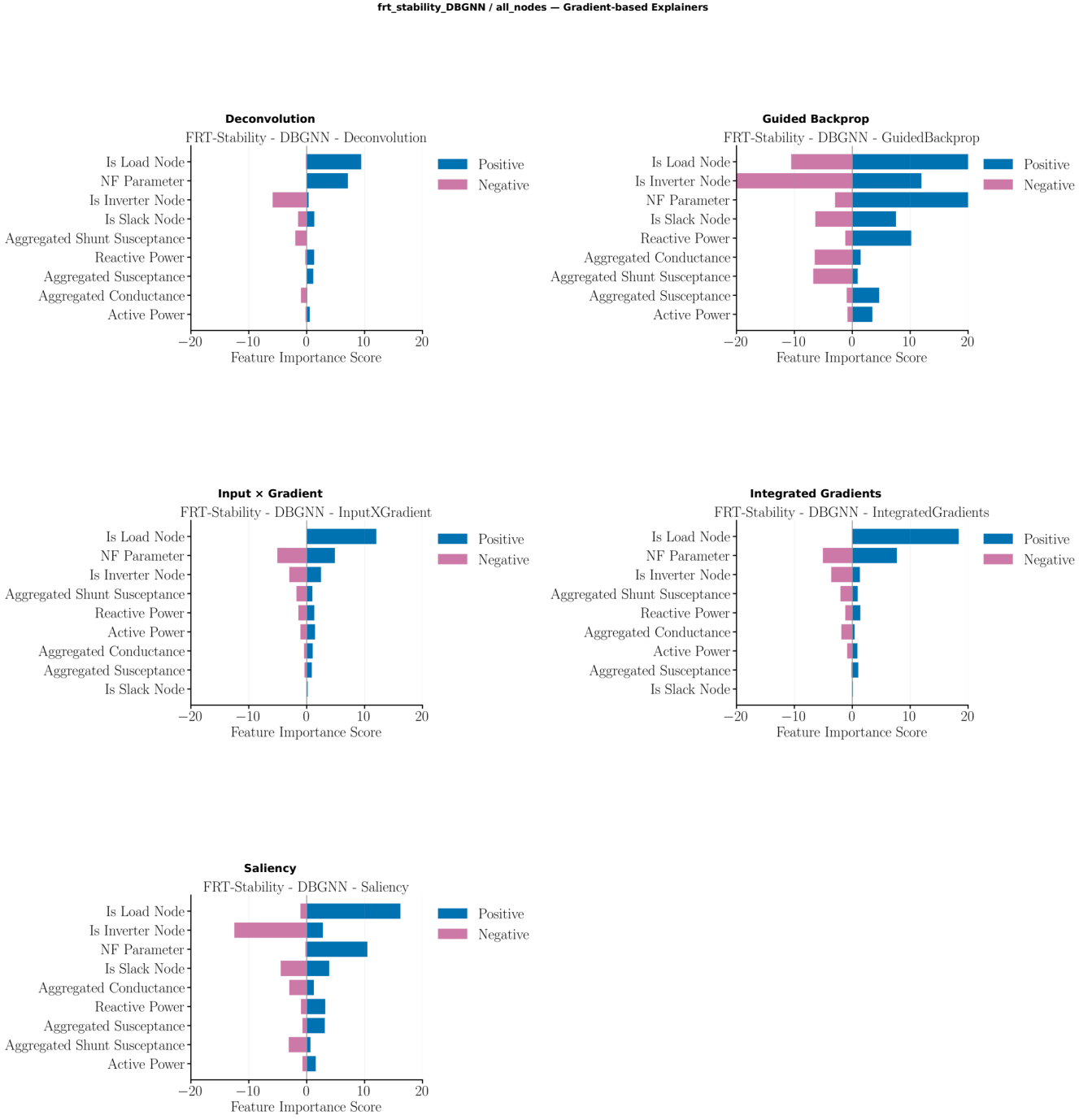


Figure 8: Aggregated node feature attributions on FRT-Stability (DBGNN, all nodes) for all five gradient-based methods and Shapley Value Sampling. The top three features (Is Load Node, NF Parameter, Is Inverter Node) are consistent across all methods. GuidedBackpropagation inverts signs due to a known artifact of its backpropagation rule. The remaining four gradient methods and Shapley Value Sampling tell the same story.

learning rate 0.01 and 200 epochs; varying these hyperparameters produces unstable attributions.

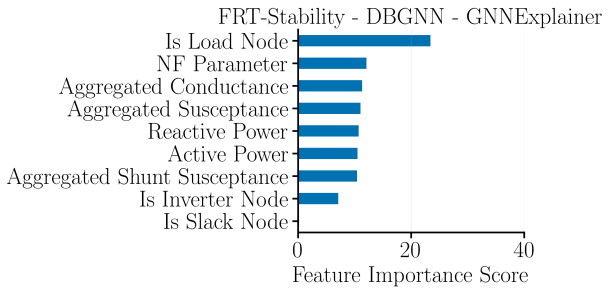


Figure 9: GNNExplainer aggregated node feature importance for DBGNN on FRT-Stability (learning rate 0.01, 200 epochs). Results are consistent with Integrated Gradients and Shapley Value Sampling under these specific hyperparameter settings.

GNNExplainer is the most widely used GNN explanation method and the only method evaluated here that theoretically provides edge feature attributions. However, it was originally designed for classification tasks, and its output in regression settings is sensitive to hyperparameter choices: results consistent with Integrated Gradients and Shapley Value Sampling are only obtained at learning rate 0.01 with 200 epochs; varying these settings produces unstable attributions. Computation time also scales with hyperparameter choices, unlike gradient-based methods. For these reasons, GNNExplainer is excluded from the main analyses; adapting it for node-level regression tasks is a promising direction for future work.

D Evaluation Details

D.1 Evaluation Metrics

Skill Score (Primary Metric). We report model performance using the skill score relative to the linear regression baseline (Eq. 4), which measures the fraction of *remaining* variance captured beyond a linear model trained on the same features and splits. A skill score of 1 indicates perfect prediction, 0 indicates no improvement over the baseline, and negative values indicate degradation.

R^2 (Reference). R^2 values are reported in Appendix E.2. The coefficient of determination measures the proportion of variance in the target variable (stability p_{frt} for FRT-Stability and active power for IEEE39-AC) explained by the model:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}, \tag{13}$$

where y_i are true values, $\hat{y}_i$ are predictions, and $\bar{y}$ is the mean.

Additionally, we report Mean Absolute Error (MAE) and Mean Squared Error (MSE) for detailed error characterization.

Out-of-Distribution Evaluation. To assess generalization of FRT-Stability models beyond the synthetic training distribution, we evaluate all models on the IEEE-96 Reliability Test System [18], modified to include grid-forming inverters with the same normal-form parameterization used in the synthetic grids. The IEEE-96 system (72 buses) features different topology, geographical constraints, and load distributions, representing a challenging out-of-distribution test. All four GNN architectures were trained and evaluated on this system following the same procedure as for the synthetic grids. Importantly, post-hoc explanations generated on the IEEE-96 graph using Integrated Gradients, Shapley Value Sampling, and GNNExplainer are consistent with those on the in-distribution test set: node type and NF Parameter remain the dominant features across all methods, suggesting that the model's learned representations generalize to unseen topologies. Note that this single test case provides qualitative evidence of generalization capability but lacks sufficient sample size for statistically robust conclusions.

D.2 Feature Selection Analysis

With a total of five edge features and nine node features, it would be computationally prohibitive to vary all combinations (2^{14} combinations). Instead, we present a selected subset of 24 cases in Table 5.

Table 4: GNN trained on FRT-Stability: MSE metrics across different models (mean ± std over 5 seeds), including performance on extreme outlier nodes.

Model	Overall Loss	Inverter Loss	Most Stable Inv. Loss	Load Loss	Least Stable Load Loss
GCN	1.93e-02 ± 1.13e-03	1.24e-02 ± 1.05e-03	4.12e-01 ± 5.31e-02	2.64e-02 ± 1.23e-03	1.28e-01 ± 2.61e-02
TAG	1.46e-02 ± 1.95e-04	8.95e-03 ± 1.46e-04	6.41e-02 ± 1.76e-02	2.03e-02 ± 3.52e-04	7.34e-02 ± 4.07e-02
GATv2	1.10e-02 ± 3.27e-04	7.64e-03 ± 3.39e-04	2.30e-01 ± 2.76e-02	1.43e-02 ± 4.77e-04	1.74e-02 ± 8.52e-03
DBGNN	4.63e-03 ± 1.04e-04	3.50e-03 ± 1.61e-04	3.16e-02 ± 1.07e-02	5.77e-03 ± 9.07e-05	2.34e-03 ± 1.91e-03

Model Type	Feature Reduction	Test Case	Is Inverter Node	Is Load Node	Is Slack Node	Active Power	Reactive Power	NF Parameter	Aggregated Conductance	Aggregated Susceptance	Aggregated Shunt Susceptance	Best R^2	Edge Conductance	Edge Susceptance	Edge Shunt Susceptance	Edge Active Power	Edge Reactive Power
	14.3%	1	1	1	1	1	1	1	1	1	1	0.9688	1		1	1	
	7.1%	2	1	1	1		1	1	1	1	1	0.9682	1	1	1	1	1
	7.1%	3	1	1	1	1	1	1	1	1		0.9679	1	1	1	1	1
	28.6%	4	1			1	1	1	1			0.9677	1	1	1	1	1
	7.1%	5	1		1	1	1	1	1	1	1	0.9675	1	1	1	1	1
Full model	0.0%	6	1	1	1	1	1	1	1	1	1	0.9673	1	1	1	1	1
	7.1%	7	1	1	1	1		1	1	1	1	0.9670	1	1	1	1	1
	7.1%	8	1	1	1	1	1	1	1		1	0.9663	1	1	1	1	1
	28.6%	9	1	1	1	1	1	1	1	1	1	0.9658	1				
Reduced model	35.7%	10				1	1	1	1			0.9654	1	1	1	1	1
	21.4%	11	1	1	1	1	1	1	1	1	1	0.9645	1			1	
	35.7%	12	1				1	1	1			0.9643	1	1	1	1	1
Only NFP	57.1%	13						1				0.9520	1	1	1	1	1
	28.6%	14	1	1	1	1	1	1	1	1	1	0.9269				1	
Full model (no EF)	35.7%	15	1	1	1	1	1	1	1	1	1	0.9250					
Reduced model (no EF)	71.4%	16				1	1	1	1			0.9173					
Only node type	42.9%	17	1	1	1							0.9009	1	1	1	1	1
Removed NFP	7.1%	18	1	1	1	1	1		1	1	1	0.8981	1	1	1	1	1
Only NFP	57.1%	19	1									0.8967	1	1	1	1	1
No neighbor info.	57.1%	20	1	1	1	1	1	1				0.8837					
Only NFP (no EF)	92.9%	21						1				0.8387					
Only Is-Inverter (no EF)	92.9%	22	1									0.8061					
No categ. NF (no EF)	78.6%	23				1	1		1			0.0238					
No categ. NF	42.9%	24				1	1		1			0.0229	1	1	1	1	1

Table 5: Training DBGNN with different input information for the FRT-Stability dataset, sorting cases based on the best R^2 score. "1" means the feature is used for training. NFP = NF Parameter (inverter time constant τ_P), EF = Edge Features, categ. = categorical node type indicators.

Table 8: Comparison of R^2 scores (mean ± std over 5 seeds) across different GNN models on the FRT-Stability dataset, broken down by node type.

Model	Overall R^2	Inverter R^2	Load R^2
GCN	0.852 ± 0.009	0.730 ± 0.024	0.539 ± 0.022
TAG	0.889 ± 0.001	0.806 ± 0.004	0.646 ± 0.006
GATv2	0.916 ± 0.002	0.837 ± 0.005	0.747 ± 0.010
DBGNN	0.965 ± 0.001	0.925 ± 0.004	0.900 ± 0.002

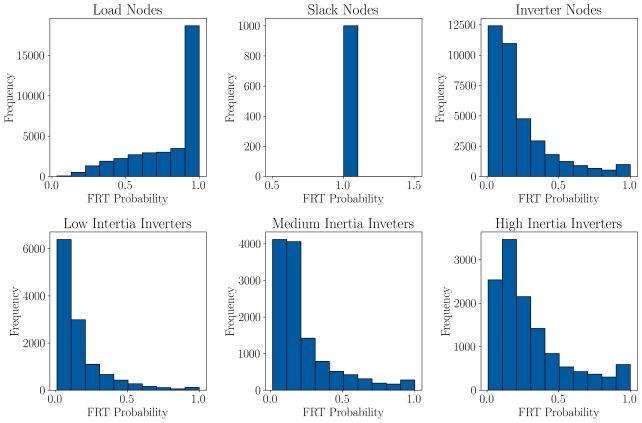


Figure 10: Distribution of FRT probability p_{frt} across all nodes in the 1,000 grid topologies, stratified by node type. Load and slack nodes are predominantly stable ($p_{\text{frt}} \approx 1$), while inverter nodes exhibit lower stability. The bottom row shows inverter subtypes: stability increases with the time constant τ_P (low to high inertia).

E Supplementary Results

E.1 Neighbor Composition

Table 6: Proportion of inverter and load neighbors for different node categories in the FRT-Stability test set.

Node Category	Inv. Neigh. (%)	Load Neigh. (%)
All Load Nodes	50	50
All Inverter Nodes	49	51
Least Stable Load Nodes	84	16
Most Stable Inverter Nodes	23	77

E.2 Performance Tables

E.3 Additional Figures

Table 7: Model performance (R^2 scores, mean ± std over 5 seeds) on IEEE39-AC (static power flow) and FRT-Stability (dynamic stability).

Model Architecture	IEEE39-AC (R^2)	FRT-Stability (R^2)
Non-Graph Methods		
MLP	0.857 ± 0.003	0.627 ± 0.001
GBT	0.861 ± 0.001	0.630 ± 0.000
Graph-Based Models		
GCN	0.971 ± 0.003	0.852 ± 0.009
TAG	0.985 ± 0.001	0.889 ± 0.001
GATv2	0.950 ± 0.008	0.916 ± 0.002
DBGNN	0.984 ± 0.005	**0.965** ± 0.001

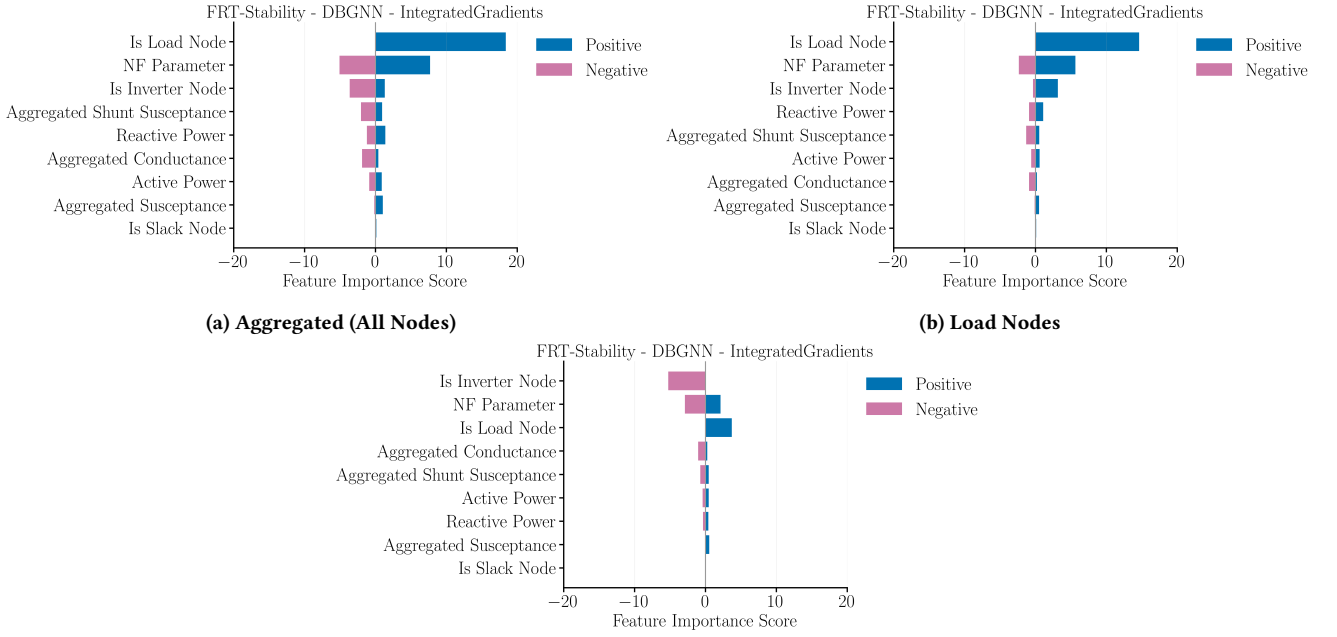


Figure 11: Integrated Gradients feature importance for the DBGNN on FRT-Stability. (a) Global importance aggregated across all nodes. (b) and (c) separate the analysis by node type (load vs. inverter).

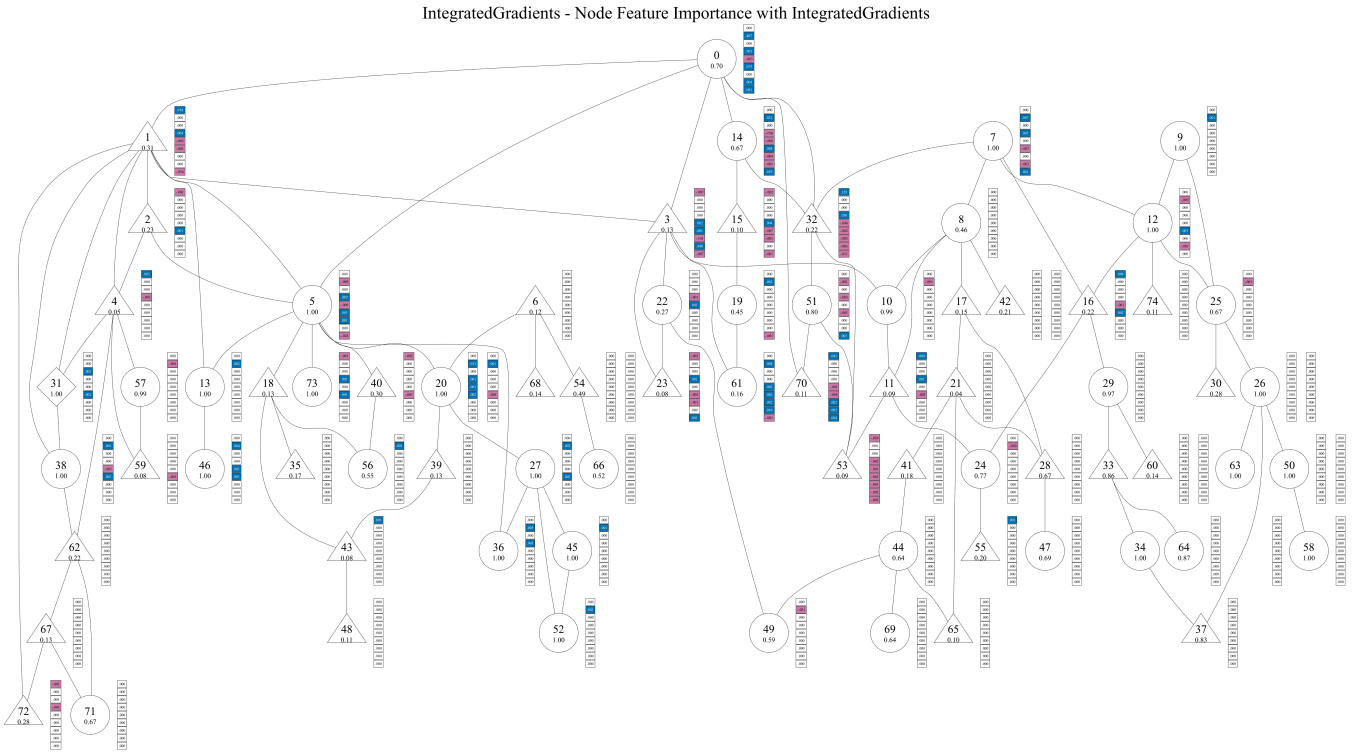


Figure 12: Integrated Gradients local feature importance for Node 0 (load type) in FRT-Stability Sample 0 (DBGNN). Node features: Active Power (P), Reactive Power (Q), NF Parameter (τ_P), Aggregated Conductance (G_k), Aggregated Susceptance (B_k), Aggregated Shunt Susceptance ($B_{k,\mathbf{sh}}$), Is Inverter, Is Slack, Is Load.