# Physics-Guided Self-Supervised Graph Neural Networks for Power Grid Analysis

Anonymous Authors
For Review

***Abstract*—Modern power grid operations require rapid computational methods for decision support, but traditional solvers face severe bottlenecks. We present a physics-guided self-supervised learning framework for graph neural networks that addresses labeled data scarcity in power system analysis. Our approach embeds admittance-weighted message passing into the encoder architecture and develops grid-specific pretext tasks (masked injection and parameter reconstruction) enabling representation learning from unlabeled operational data. Evaluated on the PowerGraph benchmark, our method achieves substantial improvements in low-label regimes: 29.1% power flow error reduction, 26.4% line flow error reduction, and 6.8% F1-score improvement for cascading failure prediction—all at 10% labeled data availability. On the IEEE 118-bus system under severe class imbalance, self-supervised pretraining dramatically reduces training instability (variance reduction from ±0.243 to ±0.051) while improving mean performance ($\Delta$F1 = +0.61). We achieve 0.93 AUC-ROC explainability fidelity via Integrated Gradients, addressing a recognized gap in interpretable cascade prediction.**

***Index Terms*—Power systems, graph neural networks, self-supervised learning, cascading failures, power flow, explainability**

## I. Introduction

Modern power grid operations demand rapid decision-making capabilities that traditional computational methods struggle to provide. Optimal power flow (OPF) solvers—used to determine economically efficient generator dispatch while respecting physical and operational constraints—can require minutes to solve on utility-scale networks with thousands of buses [1], rendering them impractical for real-time control and contingency analysis. Cascading failure risk assessment presents an even greater challenge: comprehensive N-k security analysis requires evaluating thousands or millions of contingency scenarios, each necessitating a full power flow calculation [2]. These computational bottlenecks have motivated a shift toward machine learning surrogates that approximate complex power system computations at dramatically reduced cost. Recent supervised learning approaches have demonstrated remarkable speedups—ranging from 100× to 10,000× faster than conventional solvers [1], [3]—while maintaining solution quality within 0.2% of optimality. However, these methods rely fundamentally on labeled training data: typical implementations require 5,000 to 60,000 labeled samples per network [3], [4], where each label is itself the output of the computationally expensive solver the surrogate aims to replace. This chicken-and-egg problem—needing expensive simulations to train models meant to avoid expensive simulations—severely limits the practical deployment of learning-based power system analysis, particularly for networks with limited historical data or frequently changing topologies.

Self-supervised learning (SSL) offers a compelling solution to the labeled data bottleneck by leveraging abundant unlabeled operational data. In the broader machine learning community, graph SSL methods have demonstrated substantial performance gains in low-label regimes: GraphMAE2 achieves a 5.46 percentage point accuracy improvement using only 1% labeled data on molecular property prediction benchmarks [5], while contrastive learning frameworks enable effective transfer across diverse graph domains [6]. These successes suggest that self-supervised pretraining could similarly benefit power system applications, where operational measurements (bus voltages, line flows, injection patterns) are continuously recorded but corresponding "labels" (optimal solutions, failure classifications) are costly to obtain. However, a critical gap exists: among the extensive body of graph SSL research spanning molecular graphs [7], social networks [6], and traffic systems [8], applications to power grids remain remarkably scarce. Our comprehensive literature review identified only five to six papers applying SSL to power systems [9], [10], and critically, *none* of these methods incorporate physics-informed pretext tasks that leverage the fundamental electrical relationships governing grid behavior. This represents a significant missed opportunity: power systems are governed by well-understood physical laws—Kirchhoff's current and voltage laws, power flow equations, admittance relationships—that could serve as powerful self-supervisory signals for representation learning.

The graph structure of electrical networks naturally aligns with graph neural network (GNN) architectures, where message passing along edges can mirror the physical propagation of power flows along transmission lines [11]. Recent work has begun embedding power system physics into GNN designs: impedance-weighted aggregation schemes [12], complex-valued representations preserving phase relationships [**?**], and hard Kirchhoff's law constraints enforced through architectural projections [**?**]. These physics-informed GNNs demonstrate improved accuracy and generalization compared to topology-agnostic baselines, particularly for out-of-distribution scenarios and N-1 contingencies [**?**]. However, existing physics-guided approaches remain tethered to supervised learning paradigms, requiring labeled power flow solutions or optimal dispatch setpoints for training. *No prior work has combined* *physics-guided GNN architectures with self-supervised pretraining*, leaving unexplored the question of whether physics-informed message passing can enable effective representation learning from unlabeled data alone.

This paper introduces a physics-guided self-supervised learning framework for power grid analysis that addresses these gaps. We make the following contributions:

- **Physics-Guided Graph Neural Network Architecture:** We design a message-passing encoder that incorporates admittance-weighted aggregation, where line conductance and susceptance values directly control information flow between connected buses, embedding the structure of the power system admittance matrix into the neural network computation graph.
- **Self-Supervised Pretraining Objective:** We develop grid-specific pretext tasks—masked injection reconstruction (predicting active and reactive power injections at randomly masked buses) and masked parameter reconstruction (predicting line impedance parameters at randomly masked edges)—that enable representation learning from unlabeled grid operational data without requiring solutions from conventional solvers. Critically, we ensure no label leakage by pretraining exclusively on the training partition, with validation and test sets never exposed during self-supervised learning.
- **Multi-Task Transfer Learning Evaluation:** We demonstrate that representations learned via physics-guided SSL transfer effectively to three downstream tasks: (1) power flow prediction (bus voltage magnitudes), (2) line flow prediction (active and reactive power flows on transmission lines), and (3) cascading failure classification (graph-level binary prediction of cascade occurrence). On the PowerGraph benchmark [13], our approach achieves 29.1% mean absolute error reduction for power flow, 26.4% reduction for line flow, and 6.8% F1-score improvement for cascade prediction—all measured at 10% labeled data availability relative to scratch training baselines.
- **Scalability and Variance Reduction:** On the larger IEEE 118-bus system under severe class imbalance conditions (5% cascade rate), self-supervised pretraining not only improves mean performance ($\Delta$F1 = +0.61) but dramatically reduces training instability: scratch training exhibits ±0.243 F1 variance across random seeds, while SSL-pretrained models achieve ±0.051 variance—a stabilization effect critical for reliable deployment.
- **Explainability Validation:** Using ground-truth edge importance masks from the PowerGraph benchmark, we quantitatively evaluate explanation fidelity via Integrated Gradients, achieving 0.93 AUC-ROC compared to 0.72 for heuristic baselines. This addresses a recognized gap in cascading failure prediction, where current explainable AI methods have been reported to perform "suboptimally" [13].
- **Robustness Under Distribution Shift:** We evaluate model behavior under load stress conditions (1.0× to 1.3× nominal loading), demonstrating that SSL-pretrained representations exhibit superior robustness, maintaining 22% higher performance advantage at 1.3× load compared to scratch training.

The remainder of this paper is organized as follows. Section II reviews related work in power system machine learning, graph neural networks, self-supervised learning, physics-informed methods, and cascading failure prediction. Section III formulates the graph representation of power grids and defines the three downstream tasks. Section IV details our physics-guided encoder architecture and self-supervised pretraining objective. Section V describes the experimental setup, including datasets, baselines, and training protocols. Section VI presents comprehensive results across all tasks and grid scales, including ablation studies and robustness analysis. Section VII discusses implications for operational deployment, limitations, and future directions. Section VIII concludes.

## II. Related Work

Traditional power flow and optimal power flow computations rely on iterative numerical methods such as Newton-Raphson and interior-point algorithms, which can require seconds to minutes per solve on large-scale grids [3]. To enable near-real-time decision support, researchers have developed machine learning surrogates that approximate these complex mappings from grid conditions (load demands, generation capacities, network topology) to power flow solutions or optimal dispatch setpoints. Fully-connected deep neural networks have demonstrated impressive results: Pan et al.'s DeepOPF framework achieves feasible OPF solutions with less than 0.2% optimality loss and up to 100× speedup over conventional solvers on benchmark IEEE test systems [3], while Huang et al.'s DeepOPF-V extends this approach to AC-OPF with reported speedups exceeding 10,000× on 2,000-bus networks [1]. Graph neural networks have emerged as a particularly promising architecture due to their ability to exploit grid topology and achieve greater scalability [14], [15]. For instance, PowerFlowNet demonstrated successful scaling to a 6,470-bus French transmission network with voltage magnitude prediction errors below 0.001 per-unit [14], while heterogeneous message-passing neural networks maintain constant parameter counts across grid sizes ranging from 14 to 2,000+ buses [**?**]. However, these supervised approaches face a critical limitation: they require extensive labeled training data. Typical implementations demand 5,000–60,000 OPF solutions per network [3], [4], and generating this labeled data via conventional solvers is computationally expensive—precisely the bottleneck these methods aim to circumvent. This data scarcity challenge motivates unsupervised and self-supervised learning approaches that can leverage abundant unlabeled operational data.

The natural graph structure of electrical networks—with buses as nodes and transmission lines as edges—makes graph neural networks an ideal architecture for power system analysis. GNNs encode topological relationships through message passing, where each node aggregates information from its neighbors according to the grid's physical connectivity [11].

This inductive bias enables GNNs to handle topology changes (such as N-1 contingencies) without retraining, a critical advantage over fully-connected networks that treat grid states as fixed-dimensional vectors [?]. Empirical studies demonstrate that GNN-based approaches achieve substantially lower prediction errors than traditional neural networks: for example, an electrical-model-guided GNN for distribution system state estimation attained an order-of-magnitude lower error than standard feedforward networks, even with missing sensor measurements [16]. Recent work has begun incorporating power system physics more deeply into GNN architectures. Meta-PIGACN integrates impedance-weighted edge aggregation, where line admittance values directly control message-passing strength [12], while complex-valued spatial-temporal GCNs represent voltage phasors and impedance in their native complex form to preserve phase relationships inherent to AC power flow [?]. KCLNet enforces Kirchhoff's Current Law as hard architectural constraints via differentiable hyperplane projections, guaranteeing zero KCL violations by construction [?], and PINCO achieves zero inequality constraint violations through physics-informed hard constraints in an unsupervised learning framework [?]. The PowerGraph benchmark [13] provides standardized evaluation across GCN, GAT, GraphSAGE, and Graph Transformer architectures, establishing that topology-aware message passing consistently outperforms topology-agnostic baselines on both node-level (power flow, voltage estimation) and graph-level (cascading failure prediction) tasks.

Self-supervised learning has emerged as a powerful paradigm for learning graph representations without labeled data, with two dominant approaches: contrastive learning and generative masked reconstruction. Contrastive methods, exemplified by Deep Graph Infomax [17], InfoGraph [18], and GraphCL [19], maximize agreement between different augmented views of graphs through node dropping, edge perturbation, or subgraph sampling. More recent frameworks have reduced the complexity of these approaches: BGRL eliminates negative sampling through bootstrapped representation learning [20], achieving 2–10× memory reduction while matching state-of-the-art performance, and SimGRACE dispenses with manual graph augmentation entirely by perturbing encoder parameters to generate contrastive views [21]. In parallel, generative approaches have gained traction: GraphMAE employs masked feature reconstruction with a scaled cosine error loss and dedicated decoder architecture [22], demonstrating that carefully designed autoencoders can match or exceed contrastive methods (84.2% accuracy on Cora versus 82.7% for BGRL). GraphMAE2 extends this with multi-view random re-masking and latent representation prediction [5], scaling to graphs with over 100 million nodes. These graph SSL methods show substantial benefits in low-label regimes: GraphMAE2 achieves a 5.46 percentage point accuracy improvement with only 1% labeled data on large-scale molecular property prediction benchmarks [5], while GCC's cross-domain pretraining on diverse network types enables effective transfer to new graph tasks with minimal fine-tuning [6]. Applications to physical systems have proven successful in molecular graphs [7] and traffic networks [8], where graph structure naturally encodes physical relationships. However, a significant research gap exists for power grid applications: while SafePowerGraph [23] and recent work by Zhu et al. [?] introduced hybrid supervised-SSL approaches for power systems, no pure graph SSL method has been designed specifically for electrical networks with physics-informed constraints such as power flow equations, Kirchhoff's laws, or voltage-angle relationships.

Physics-informed neural networks embed domain knowledge—governing equations, conservation laws, or known constraints—into machine learning models to improve generalization and sample efficiency. The canonical PINN framework [?] encodes partial differential equations as soft regularization terms in the loss function, minimizing both data mismatch and PDE residuals at collocation points. This approach has proven effective for solving forward and inverse problems in fluid dynamics, solid mechanics, and heat transfer [?]. Alternative strategies include hard constraint enforcement, where physical laws are satisfied by construction through architectural design: Beucler et al. demonstrated that architecture-constrained networks can achieve energy and mass conservation to machine precision without loss penalties [?], while Hamiltonian Neural Networks embed symplectic structure to guarantee exact energy conservation in dynamical systems [?]. In power systems specifically, physics-informed approaches have addressed swing equation dynamics for transient stability [?], state estimation with admittance matrix constraints [?], and power flow prediction with Kirchhoff's law regularization [24]. These methods demonstrate compelling benefits: physics constraints act as inductive biases that limit the hypothesis space and improve out-of-distribution generalization, with physics-informed GNNs showing zero-shot generalizability to systems an order of magnitude larger than training configurations [?]. However, a critical gap remains: the vast majority of PINN research targets continuous PDE-governed domains where automatic differentiation naturally computes spatial and temporal derivatives. Discrete, graph-structured infrastructure networks like power grids present fundamentally different challenges—physical laws (KCL, KVL, power balance) operate directly on graph edges and nodes rather than as discretized continuous fields, and topological changes require handling discrete switching dynamics. While GraPhyR [?] demonstrated physics-informed GNNs with gated message passing for power system reconfiguration, graph-based power system surrogates that combine self-supervised pretraining with physics-guided architectures remain unexplored.

Cascading failures—sequences of component outages that can escalate to widespread blackouts—represent a critical threat to power grid resilience, as evidenced by major disruptions including the 2003 Northeast blackout and the 2021 Texas winter storm [25]. Traditional simulation models such as OPA [26], DCSIMSEP [27], and the Manchester model [28] capture cascading dynamics through iterative power flow calculations coupled with protection system logic, revealing selforganized criticality in grid behavior. However, these physics-based simulators face computational limitations: DC cascade models provide two orders of magnitude speedup over AC models but sacrifice accuracy by ignoring voltage collapse mechanisms [**?**], while AC models are approximately 7× more computationally expensive and suffer convergence issues under high stress conditions. Machine learning approaches have achieved dramatic acceleration: deep convolutional neural networks enable 100× faster N-1 contingency screening [2], while graph neural networks leverage network topology to predict cascade outcomes with over 96% accuracy [**?**]. GNN-based methods demonstrate transfer learning capability across different grid topologies and operating conditions, addressing a key limitation of classical simulation approaches. However, explainability remains a significant gap. Post-mortem analyses of major blackouts still rely on manual timeline reconstruction and root cause analysis [25], and recent benchmarking reveals that current explainable AI methods perform "suboptimally" on cascade explanation tasks [13]. The PowerGraph benchmark explicitly identifies this gap, noting that "given the crucial role of explainability for power grid operators, this underscores the ongoing need for dedicated research and development in this field" [13]. While XAI techniques including SHAP, LIME, and attention mechanisms have been successfully applied to other power system tasks such as frequency stability prediction, cascading failure prediction lacks robust explainability frameworks that can identify critical transmission pathways and quantify failure propagation mechanisms in a way that operators can trust and act upon.

This work addresses the identified gaps by uniquely combining physics-guided graph neural network architectures with self-supervised pretraining for power grid analysis. Our physics-guided encoder embeds admittance-weighted message passing directly into the network architecture—drawing on the success of Meta-PIGACN [12] and KCLNet [**?**] but extending to a self-supervised learning paradigm. Unlike existing SSL approaches for graphs that ignore domain physics [20], [22], we design grid-specific pretext tasks (masked injection reconstruction, masked parameter reconstruction) that leverage power system structure without requiring labeled solutions from conventional solvers. This addresses the data scarcity challenge identified in supervised power flow learning [3], [4], enabling effective representation learning from the abundant unlabeled operational data available in modern power grids. We demonstrate transfer learning across multiple tasks (power flow, line flow prediction, cascading failure classification) and grid scales (IEEE 24-bus and 118-bus systems), validating the generalizability of learned representations. Finally, we provide quantitative explainability evaluation using ground-truth explanation masks from the PowerGraph benchmark [13], achieving 0.93 AUC-ROC fidelity for edge importance attribution via Integrated Gradients—addressing the explainability gap in cascading failure prediction and providing operators with interpretable risk assessments.

TABLE I
TASK SPECIFICATIONS, INPUTS, OUTPUTS, AND EVALUATION METRICS

| **Task** | **Output** | **Metric** | **Unit** |
|---|---|---|---|
| Cascade | Binary graph label | F1 Score | [0,1] |
| Power Flow | Bus voltages $V_i$ | MAE | p.u. |
| Line Flow | Line flows ($P_{ij}, Q_{ij}$) | MAE | p.u. |

## III. Problem Formulation

### *A. Graph Representation of Power Grids*

We represent a power grid as an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where nodes $v_i \in \mathcal{V}$ correspond to buses (electrical connection points) and edges $e_{ij} \in \mathcal{E}$ correspond to transmission lines and transformers connecting buses $i$ and $j$. Each node is associated with a feature vector $\mathbf{x}_i \in \mathbb{R}^{d_{\text{node}}}$ encoding the electrical state at bus $i$, and each edge is associated with a feature vector $\mathbf{e}_{ij} \in \mathbb{R}^{d_{\text{edge}}}$ capturing line impedance and capacity parameters.

**Node features** ($d_{\text{node}} = 3$ for cascade and line flow tasks, $d_{\text{node}} = 2$ for power flow task):

- $P_{\text{net},i}$: Net active power injection (generation minus load) at bus $i$
- $S_{\text{net},i}$: Net apparent power magnitude at bus $i$
- $V_i$: Voltage magnitude at bus $i$ (excluded for power flow prediction to avoid trivial leakage)

**Edge features** ($d_{\text{edge}} = 4$):

- $g_{ij}$: Conductance of line $(i, j)$
- $b_{ij}$: Susceptance of line $(i, j)$
- $x_{ij}$: Reactance of line $(i, j)$
- $S_{\max,ij}$: Thermal rating (maximum apparent power capacity) of line $(i, j)$

All electrical quantities are normalized to per-unit values with system base $S_{\text{base}} = 100$ MVA, ensuring dimensionless features in the range $[-1, 1]$ for injections and $[0.9, 1.1]$ for voltages under normal operating conditions. The admittance $Y_{ij} = g_{ij} + jb_{ij}$ relates voltage difference to power flow via the AC power flow equations, providing the physical coupling we leverage in our message-passing design.

### *B. Task Definitions and Evaluation Metrics*

We consider three downstream prediction tasks, each addressing a critical operational need in power system analysis. Table I summarizes the input-output specifications and evaluation metrics for each task.

**Cascading Failure Prediction (Graph-Level Classification):** Given the pre-outage state of a power grid, predict whether an N-k contingency (simultaneous outage of $k$ components) will trigger a cascading failure. We define a cascade as occurring when the total demand not served (DNS) exceeds zero: DNS $= \sum_i(\text{load}_i - \text{served}_i) > 0$ MW. This is formulated as binary graph-level classification, where the model produces a single prediction $\hat{y} \in \{0, 1\}$ per graph. Performance is evaluated using F1-score computed over test graphs, which balances precision and recall—critical for imbalanced datasets where cascades are rare events (5–20% positive class rate depending on grid size and simulation parameters).

**Power Flow Prediction (Node-Level Regression):** Predict bus voltage magnitudes $\{V_i\}_{i=1}^{|\mathcal{V}|}$ given load injections and grid topology. This approximates the solution to the AC power flow equations without iterative numerical methods. The model output is a vector $\hat{\mathbf{V}} \in \mathbb{R}^{|\mathcal{V}|}$ of predicted voltage magnitudes. Performance is measured by mean absolute error (MAE) in per-unit: MAE $= \frac{1}{|\mathcal{V}|}\sum_{i=1}^{|\mathcal{V}|}|V_i - \hat{V}_i|$, averaged over all buses and test samples. Typical operational voltage bounds are $0.95 \leq V_i \leq 1.05$ per-unit, making MAE values on the order of $10^{-3}$ per-unit operationally acceptable.

**Line Flow Prediction (Edge-Level Regression):** Predict active and reactive power flows $\{(P_{ij}, Q_{ij})\}_{(i,j)\in\mathcal{E}}$ on all transmission lines given bus states and topology. This enables rapid screening of line loading for contingency analysis. The model outputs two scalars per directed edge: $(\hat{P}_{ij}, \hat{Q}_{ij})$. MAE is computed separately for active and reactive components, then averaged: MAE $= \frac{1}{2|\mathcal{E}|}\sum_{(i,j)\in\mathcal{E}}(|P_{ij}-\hat{P}_{ij}|+|Q_{ij}-\hat{Q}_{ij}|)$. Accurate line flow prediction is essential for identifying thermal overloads that could initiate cascades.

**Improvement Metric Convention:** When comparing self-supervised pretraining (SSL) against scratch training, we define improvement as $(SSL - Scratch)/Scratch \times 100\%$ for metrics where higher is better (F1-score), and $(Scratch - SSL)/Scratch \times 100\%$ for metrics where lower is better (MAE). This convention ensures positive improvement percentages consistently indicate SSL outperforming scratch training.

## IV. Methodology

### A. Architecture Overview

Our framework follows a shared-encoder paradigm: a single physics-guided graph neural network encoder learns representations from grid topology and electrical states, which are then specialized for downstream tasks via lightweight task-specific heads. This design enables effective transfer learning—representations learned during self-supervised pretraining on unlabeled data transfer to supervised fine-tuning on labeled samples, with the encoder weights providing a strong initialization that accelerates convergence and improves sample efficiency.

The overall pipeline consists of three stages: (1) **Self-supervised pretraining** on unlabeled grid operational data using masked reconstruction objectives, (2) **Encoder transfer** where pretrained encoder weights initialize downstream models, and (3) **Supervised fine-tuning** on task-specific labeled data with frozen or fine-tuned encoder parameters. We implement all models using PyTorch Geometric [**?**], leveraging its efficient sparse message-passing primitives for scalability to large grids.

### B. Physics-Guided Message Passing

Traditional graph convolutional networks aggregate neighbor information uniformly or via learned attention weights, ignoring the physical laws governing power flow. We embed power system physics directly into the message-passing structure through admittance-weighted aggregation.

**Standard message passing** updates node $i$'s representation $\mathbf{h}_i^{(\ell)}$ at layer $\ell$ via:

$$\mathbf{h}_i^{(\ell+1)} = \sigma\left(\mathbf{W}^{(\ell)}\mathbf{h}_i^{(\ell)} + \sum_{j\in\mathcal{N}(i)} \mathbf{M}^{(\ell)}(\mathbf{h}_j^{(\ell)}, \mathbf{e}_{ij})\right) \tag{1}$$

where $\mathcal{N}(i)$ is the neighborhood of node $i$, $\mathbf{M}^{(\ell)}$ is a message function, $\mathbf{W}^{(\ell)}$ is a learnable weight matrix, and $\sigma$ is a nonlinear activation.

**Physics-guided message passing** modifies this by weighting messages according to line admittance magnitude $|\mathbf{Y}_{ij}| = \sqrt{g_{ij}^2 + b_{ij}^2}$, which physically determines the strength of electrical coupling between buses:

$$\mathbf{h}_i^{(\ell+1)} = \sigma\left(\mathbf{W}^{(\ell)}\mathbf{h}_i^{(\ell)} + \sum_{j\in\mathcal{N}(i)} \frac{|\mathbf{Y}_{ij}|}{\sqrt{|\mathcal{N}(i)|}} \cdot \mathbf{M}^{(\ell)}(\mathbf{h}_j^{(\ell)}, \mathbf{e}_{ij})\right) \tag{2}$$

The normalization by $\sqrt{|\mathcal{N}(i)|}$ prevents representation magnitudes from growing with node degree, similar to spectral graph convolution normalization [11]. The message function $\mathbf{M}^{(\ell)}$ is implemented as:

$$\mathbf{M}^{(\ell)}(\mathbf{h}_j, \mathbf{e}_{ij}) = \mathbf{W}_{\text{msg}}^{(\ell)} \cdot [\mathbf{h}_j \,||\, \phi(\mathbf{e}_{ij})] \tag{3}$$

where $||$ denotes concatenation, $\phi$ is an edge feature embedding network, and $\mathbf{W}_{\text{msg}}^{(\ell)}$ are learnable parameters.

This design encodes a key physical intuition: power flows preferentially through low-impedance (high-admittance) paths, analogous to current following least-resistance paths in electrical circuits. By making admittance a structural prior rather than a learned weight, we constrain the model to respect grid physics even before observing labeled data.

### C. Encoder Architecture

The **PhysicsGuidedEncoder** consists of $L = 4$ stacked physics-guided convolutional layers (Eq. 2) with hidden dimension $d_h = 128$, ReLU activations, and dropout (rate $p = 0.1$) for regularization. Input node features are first projected from $\mathbb{R}^{d_{\text{node}}}$ to $\mathbb{R}^{d_h}$ via a learnable linear layer, and edge features are similarly projected from $\mathbb{R}^{d_{\text{edge}}}$ to $\mathbb{R}^{d_h}$. After $L$ layers of message passing, each node $i$ has learned a representation $\mathbf{h}_i \in \mathbb{R}^{d_h}$ capturing both local electrical state and global topological context via multi-hop aggregation.

For graph-level tasks (cascading failure prediction), node representations are aggregated into a graph embedding $\mathbf{h}_\mathcal{G} \in \mathbb{R}^{d_h}$ via:

$$\mathbf{h}_\mathcal{G} = \text{READOUT}(\{\mathbf{h}_i\}_{i\in\mathcal{V}}) = \frac{1}{|\mathcal{V}|}\sum_{i\in\mathcal{V}} \mathbf{h}_i \,||\, \max_{i\in\mathcal{V}} \mathbf{h}_i \tag{4}$$

concatenating mean and max pooling to capture both distributional and extreme-value information—relevant for cascades where a single critical bus can trigger system-wide failure.

### *D. Task-Specific Heads*

**Power Flow Head (Node-Level):** Predicts voltage magnitude $\hat{V}_i$ for each bus via a two-layer multilayer perceptron (MLP) applied independently to each node embedding: $\hat{V}_i = \text{MLP}_{\text{PF}}(\mathbf{h}_i)$. The MLP has hidden dimension 64 with ReLU activation and outputs a single scalar constrained to $[0.8, 1.2]$ via sigmoid scaling to respect physical voltage limits.

**Line Flow Head (Edge-Level):** Predicts active and reactive power flows $(\hat{P}_{ij}, \hat{Q}_{ij})$ by concatenating source and target node embeddings and applying an edge-level MLP: $(\hat{P}_{ij}, \hat{Q}_{ij}) = \text{MLP}_{\text{LF}}([\mathbf{h}_i \,||\, \mathbf{h}_j])$. This design captures bidirectional electrical coupling: power flow from bus $i$ to $j$ depends on both buses' states.

**Cascading Failure Head (Graph-Level):** A two-layer MLP maps the graph embedding $\mathbf{h}_{\mathcal{G}}$ to a cascade probability: $\hat{p}_{\text{cascade}} = \sigma(\text{MLP}_{\text{CF}}(\mathbf{h}_{\mathcal{G}}))$, where $\sigma$ is the sigmoid function. Training uses binary cross-entropy loss with optional class weighting to handle imbalanced datasets.

### *E. Self-Supervised Pretraining*

We design a graph-specific self-supervised learning objective that exploits the abundant unlabeled operational measurements available in modern power grids (continuously recorded bus injections, line parameters, voltage samples) without requiring expensive labels from OPF solvers or cascade simulations.

**Masked Reconstruction Objective:** Inspired by masked language modeling in BERT [?] and recent graph autoencoders [22], we randomly mask 15% of node features and 15% of edge features in each training graph, then train the encoder to reconstruct the original masked values. This forces the model to learn how electrical quantities relate through grid topology and physics.

**Masking Strategy:** For each selected node/edge, we apply one of three transformations with specified probabilities: (1) Replace with a learnable mask token (80%), (2) Replace with random noise sampled from the feature distribution (10%), (3) Keep unchanged (10%). This prevents the model from trivially identifying masked positions and encourages robust representations.

**Reconstruction Architecture:** Masked node features are reconstructed via an MLP decoder: $\hat{\mathbf{x}}_i = \text{MLP}_{\text{node}}(\mathbf{h}_i)$. Masked edge features are reconstructed by concatenating source and target node embeddings: $\hat{\mathbf{e}}_{ij} = \text{MLP}_{\text{edge}}([\mathbf{h}_i \,||\, \mathbf{h}_j])$. Loss is computed as mean squared error over *masked positions only*:

$$\mathcal{L}_{\text{SSL}} = \frac{1}{|\mathcal{M}_{\mathcal{V}}|} \sum_{i \in \mathcal{M}_{\mathcal{V}}} \|\mathbf{x}_i - \hat{\mathbf{x}}_i\|^2 + \frac{1}{|\mathcal{M}_{\mathcal{E}}|} \sum_{(i,j) \in \mathcal{M}_{\mathcal{E}}} \|\mathbf{e}_{ij} - \hat{\mathbf{e}}_{ij}\|^2 \tag{5}$$

where $\mathcal{M}_{\mathcal{V}}$ and $\mathcal{M}_{\mathcal{E}}$ are the sets of masked nodes and edges respectively.

**Critical Leakage Prevention:** We strictly ensure no label leakage during pretraining. For power flow tasks, voltage magnitude $V_i$ is the prediction target and is *excluded* from node features during both SSL pretraining and fine-tuning. For line flow tasks, edge power flows $(P_{ij}, Q_{ij})$ are excluded from edge features. Additionally, SSL pretraining uses *only* the training partition (80% of data)—validation and test sets are never exposed during unsupervised learning, ensuring that representations transfer fairly to held-out evaluation.

**Algorithm 1** Physics-Guided SSL Pipeline

```
Input: Unlabeled graphs $\{\mathcal{G}_i\}_{i=1}^N$, labeled data
    $\{(\mathcal{G}_j, y_j)\}_{j=1}^M$, masking ratio $r = 0.15$
Output: Task-specific model $f_\theta$
// Phase 1: Self-Supervised Pretraining
Initialize encoder $E_\phi$ randomly
for epoch $= 1$ to $T_{\text{pretrain}}$ do
    for each batch $\mathcal{B}$ from $\{\mathcal{G}_i\}_{i=1}^N$ (train only) do
        $\tilde{\mathcal{B}} \leftarrow$ Mask($\mathcal{B}$, $r$)          // Mask nodes and edges
        $\{\mathbf{h}_i\} \leftarrow E_\phi(\tilde{\mathcal{B}})$                // Encode
        $\hat{\mathbf{x}}, \hat{\mathbf{e}} \leftarrow$ Decode($\{\mathbf{h}_i\}$)         // Reconstruct
        $\mathcal{L} \leftarrow$ MSE($\mathbf{x}_{\text{masked}}$, $\hat{\mathbf{x}}$) + MSE($\mathbf{e}_{\text{masked}}$, $\hat{\mathbf{e}}$)
        Update $\phi$ via gradient descent on $\mathcal{L}$
    end for
end for
// Phase 2: Supervised Fine-Tuning
Initialize task head $H_\psi$ randomly
Initialize encoder from pretrained: $E_{\phi'} \leftarrow E_\phi$
for epoch $= 1$ to $T_{\text{finetune}}$ do
    for each batch $\mathcal{B}$ from $\{(\mathcal{G}_j, y_j)\}_{j=1}^M$ do
        $\{\mathbf{h}_i\} \leftarrow E_{\phi'}(\mathcal{B})$                  // Encode
        $\hat{y} \leftarrow H_\psi(\{\mathbf{h}_i\})$                  // Task prediction
        $\mathcal{L}_{\text{task}} \leftarrow$ TaskLoss($\hat{y}$, $y$)
        Update $\phi'$, $\psi$ via gradient descent on $\mathcal{L}_{\text{task}}$
    end for
end for
return $f_\theta = H_\psi \circ E_{\phi'}$
```

**Physics-Informed Pretext Tasks:** Unlike generic graph SSL that might mask arbitrary features, our approach targets power-relevant quantities: bus injections $(P_{\text{net}}, S_{\text{net}})$ and line impedances $(g, b, x)$. Reconstructing masked injections requires understanding how power balances across the network (Kirchhoff's Current Law), while reconstructing masked impedances requires inferring electrical distances from voltage/power patterns (Ohm's Law for AC circuits). This makes the pretext task *physically meaningful* rather than a purely statistical pattern-matching exercise.

### *F. Training Procedure*

Algorithm 1 summarizes the complete training pipeline.

**Pretraining Phase:** We train the SSL model for 50 epochs using AdamW optimizer (learning rate $10^{-3}$, weight decay $10^{-4}$) with cosine annealing learning rate schedule. Batch size is 64 graphs. The pretrained encoder weights are saved when validation reconstruction loss is minimized.

**Fine-Tuning Phase:** Task-specific heads are randomly initialized, and the encoder is initialized from pretrained weights. We fine-tune both encoder and head jointly for 50–100 epochs depending on task complexity, using the same optimizer configuration. Early stopping monitors validation task metric (F1-score for classification, MAE for regression) with patience of 20 epochs. For low-label experiments, we randomly sample the specified fraction (10%, 20%, 50%, or 100%) of labeled training data, repeating across 5 random seeds (42, 123, 456, 789, 1011) to assess statistical significance.

### G. Explainability via Integrated Gradients

To provide interpretable predictions for cascading failure risk, we employ Integrated Gradients [**?**] to attribute prediction scores to individual transmission lines. For a cascade prediction $f(\mathcal{G})$, the importance of edge $(i, j)$ is computed by integrating gradients along a straight path from a baseline (zero edge features) to the actual edge features:

$$\mathrm{IG}_{ij} = (\mathbf{e}_{ij} - \mathbf{e}^{\text{baseline}}) \cdot \int_{\alpha=0}^{1} \frac{\partial f(\mathcal{G}_\alpha)}{\partial \mathbf{e}_{ij}} d\alpha \tag{6}$$

where $\mathcal{G}_\alpha$ is the graph with edge features interpolated as $\mathbf{e}_{ij}^{(\alpha)} = \mathbf{e}^{\text{baseline}} + \alpha(\mathbf{e}_{ij} - \mathbf{e}^{\text{baseline}})$. The integral is approximated via Riemann sum with 50 steps. We evaluate explanation fidelity by comparing $\{\mathrm{IG}_{ij}\}$ against ground-truth edge importance masks provided in the PowerGraph benchmark using AUC-ROC, which measures the method's ability to rank truly critical edges above non-critical ones.

## V. Experimental Setup

[To be drafted in Day 4]

## VI. Results

[To be drafted in Day 5]

## VII. Discussion

[To be drafted in Day 6]

## VIII. Conclusion

[To be drafted in Day 6]

## References

[1] W. Huang, X. Pan, M. Chen, and S. H. Low, "DeepOPF-V: Solving AC-OPF problems efficiently," *IEEE Transactions on Power Systems*, vol. 37, no. 1, pp. 800–803, 2022, extends DeepOPF to AC-OPF with voltage-constrained approach; achieves 10,000x speedup on 2000-bus systems.

[2] Y. Du, F. Li, J. Li, and T. Zheng, "Achieving 100x acceleration for N-1 contingency screening with uncertain scenarios using deep convolutional neural network," *IEEE Transactions on Power Systems*, vol. 34, no. 4, pp. 3303–3305, 2019.

[3] X. Pan, T. Zhao, M. Chen, and S. Zhang, "DeepOPF: A deep neural network approach for security-constrained DC optimal power flow," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 1725–1735, 2021, achieves ¡0.2% optimality loss with 100x speedup on IEEE 30/118/300-bus systems using predict-and-reconstruct DNN approach.

[4] S. Mohammadi, V.-H. Bui, W. Su, and B. Wang, "Surrogate modeling for solving OPF: A review," *Sustainability*, vol. 16, no. 22, p. 9851, 2024.

[5] Z. Hou, Y. He, Y. Cen, X. Liu, Y. Dong, E. Kharlamov, and J. Tang, "GraphMAE2: A decoding-enhanced masked self-supervised graph learner," in *Proceedings of the ACM Web Conference 2023*. ACM, 2023, pp. 737–746.

[6] J. Qiu, Q. Chen, Y. Dong, J. Zhang, H. Yang, M. Ding, K. Wang, and J. Tang, "GCC: Graph contrastive coding for graph neural network pre-training," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2020, pp. 1150–1160.

[7] S. Liu, H. Wang, W. Liu, J. Lasenby, H. Guo, and J. Tang, "Pre-training molecular graph representation with 3d geometry," in *International Conference on Learning Representations (ICLR)*, 2022.

[8] J. Ji, J. Wang, C. Huang, J. Wu, B. Xu, Z. Wu, J. Zhang, and Y. Zheng, "Spatio-temporal self-supervised learning for traffic flow prediction," vol. 37, pp. 4356–4364, 2023.

[9] 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, self-supervised GNN minimizing Kirchhoff violations; no labeled data needed; 0.995 correlation with Newton-Raphson on IEEE 118-bus.

[10] S. Park and P. Van Hentenryck, "PDL-SCOPF: Self-supervised primal-dual learning for large-scale security-constrained DC optimal power flow," *arXiv preprint arXiv:2311.18072*, 2023, end-to-end primal-dual learning eliminating need for pre-computed optimal solutions; Augmented Lagrangian Method.

[11] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017.

[12] Z. Wu, Y. Xu, J. Wang, and C. Lyu, "Meta-learning enhanced physics-informed graph attention convolutional network for power system state estimation," *IEEE Transactions on Network Science and Engineering*, 2025, cited in Agent 2.

[13] A. Varbella, K. Amara, B. Gjorgiev, M. El-Assady, and G. Sansavini, "PowerGraph: A power grid benchmark dataset for graph neural networks," *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2024, benchmark with PF, OPF, cascading failure tasks; GAT best-performing architecture; single MPL layer optimal.

[14] N. Lin, S. Orfanoudakis, N. Ordonez Cardenas, J. S. Giraldo, and P. P. Vergara, "PowerFlowNet: Power flow approximation using message passing graph neural networks," *International Journal of Electrical Power & Energy Systems*, vol. 160, p. 110112, 2024, 145x faster than Newton-Raphson on French 6470-bus network; K-hop message passing; voltage MAE ¡0.001 p.u.

[15] S. Liu, C. Wu, and H. Zhu, "Topology-aware graph neural networks for learning feasible and adaptive AC-OPF solutions," *IEEE Transactions on Power Systems*, vol. 38, no. 5, pp. 4953–4966, 2023, exploits LMP locality; physics-aware flow feasibility regularization; efficient retraining for topology changes.

[16] H. Lin and Y. Sun, "EleGNN: Electrical-model-guided graph neural networks for power distribution system state estimation," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2022, pp. 5292–5298.

[17] P. Veličković, W. Fedus, W. L. Hamilton, P. Liò, Y. Bengio, and R. D. Hjelm, "Deep graph infomax," in *International Conference on Learning Representations*, 2019.

[18] F.-Y. Sun, J. Hoffmann, V. Verma, and J. Tang, "InfoGraph: Unsupervised and semi-supervised graph-level representation learning via mutual information maximization," in *International Conference on Learning Representations*, 2020.

[19] Y. You, T. Chen, Y. Sui, T. Chen, Z. Wang, and Y. Shen, "Graph contrastive learning with augmentations," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 5812–5823.

[20] S. Thakoor, C. Tallec, M. G. Azar, M. Azabou, E. L. Dyer, R. Munos, P. Veličković, and M. Valko, "Large-scale representation learning on graphs via bootstrapping," in *International Conference on Learning Representations*, 2022.

[21] J. Xia, L. Wu, J. Chen, B. Hu, and S. Z. Li, "SimGRACE: A simple framework for graph contrastive learning without data augmentation," in *Proceedings of the ACM Web Conference 2022*. ACM, 2022, pp. 1070–1079.

[22] Z. Hou, X. Liu, Y. Cen, Y. Dong, H. Yang, C. Wang, and J. Tang, "GraphMAE: Self-supervised masked graph autoencoders," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, 2022, pp. 594–604.

[23] M. Ringsquandl ***et al.***, "SafePowerGraph: Safety-aware evaluation of graph neural networks for transmission power grids," *arXiv preprint arXiv:2407.15929*, 2024.

[24] X. Hu, H. Hu, S. Verma, and Z.-L. Zhang, "Physics-guided deep neural networks for power flow analysis," *IEEE Transactions on Power Systems*, vol. 36, no. 3, pp. 2082–2092, 2021, autoEncoder-based DNN embedding Kirchhoff's laws; topology knowledge in weight matrices; tested on IEEE 118-bus.

[25] U.S.-Canada Power System Outage Task Force, “Final report on the august 14, 2003 blackout in the united states and canada: Causes and recommendations,” U.S. Dept. of Energy and Natural Resources Canada, Tech. Rep., 2004.

[26] B. A. Carreras, V. E. Lynch, I. Dobson, and D. E. Newman, “Critical points and transitions in an electric power transmission model for cascading failure blackouts,” *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 12, no. 4, pp. 985–994, 2002.

[27] M. J. Eppstein and P. D. H. Hines, “A “random chemistry” algorithm for identifying collections of multiple contingencies that initiate cascading failure,” *IEEE Transactions on Power Systems*, vol. 27, no. 3, pp. 1698–1705, 2012.

[28] D. P. Nedic, I. Dobson, D. S. Kirschen, B. A. Carreras, and V. E. Lynch, “Criticality in a cascading failure blackout model,” *International Journal of Electrical Power & Energy Systems*, vol. 28, no. 9, pp. 627–633, 2006.