NP-NDS: A Nature-Powered Nonlinear Dynamical System for Power Grid Forecasting

Chunshu Wu[1], **Ruibing Song**[1], **Chuan Liu**[1], **Yuqing Wang**[5], **Yousu Chen**[2], **Ang Li**[2], **Dongfang Liu**[3], **Ying Nian Wu**[4], **Michael Huang**[1], **Tony (Tong) Geng**[1]

[1]University of Rochester,
[2]Pacific Northwest National Laboratory,
[3]Rochester Institute of Technology,
[4]University of California, Los Angeles,
[5]Tufts University

Abstract

The power grid is a critical dynamical system that forms the backbone of modern society, powering everything from household appliances to complex industrial machinery. However, this essential system is not without vulnerabilities – as electricity travels at lightspeed, unanticipated failures can cause catastrophic consequences such as country-wide blackouts in a cascading manner. In response to such threats, we introduce NP-NDS, a nature-powered nonlinear dynamical system designed to accurately and rapidly predict power grids as macroscopic dynamical systems in the real world. In particular, NP-NDS is established through a Hamiltonian-Hardware co-design: First, NP-NDS employs a hardware-friendly serial-additive Hamiltonian based on Chebyshev series for accurately capturing highly nonlinear interactions among power grid nodes, coupled with node-relation-aware training for high accuracy. Second, NP-NDS features a fully CMOS-based hardware dynamical system governed by the proposed Hamiltonian, facilitating inferences with "speed of electrons". Results show that NP-NDS achieves, on average, 2.3×10^3 speedup and $10^5\times$ energy reduction versus GNNs on GPU with 23.6% and 28.2% decrease in MAE and RMSE compared to GNNs on power grid forecasting datasets.

1 Introduction

"If it weren't for electricity, we'd all be watching television by candlelight."

– George Gobel

The domestic use of electricity dates back more than a century ago, gradually transforming the lifestyle of generations – one may be too comfortable to realize that television is powered by electricity. To enable the modern lifestyle, power grids play a critical role as the vein, transporting electricity from power plants to millions of households. The vein is also growing actively and consistently: over the past decades, power grids have undergone significant enhancements in both scope and complexity ($3\times$ global electricity generation in 40 years as Figure 1 highlights), establishing the lifeline of modern economy, technology, and even the entire civilization.

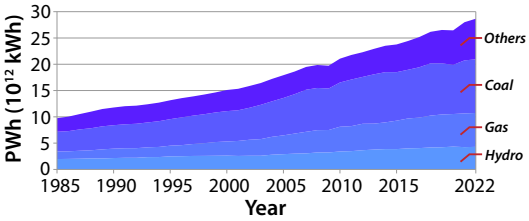

Figure 1: Global electricity generated by year [1].

However, despite years of development, power grids are still far from flawless. As a dynamical system of electricity, the behavior of a local area can have immediate global impacts as electricity

C. Wu et al., NP-NDS: A Nature-Powered Nonlinear Dynamical System for Power Grid Forecasting. *Proceedings of the Third Learning on Graphs Conference (LoG 2024)*, PMLR 269, Virtual Event, November 26–29, 2024.

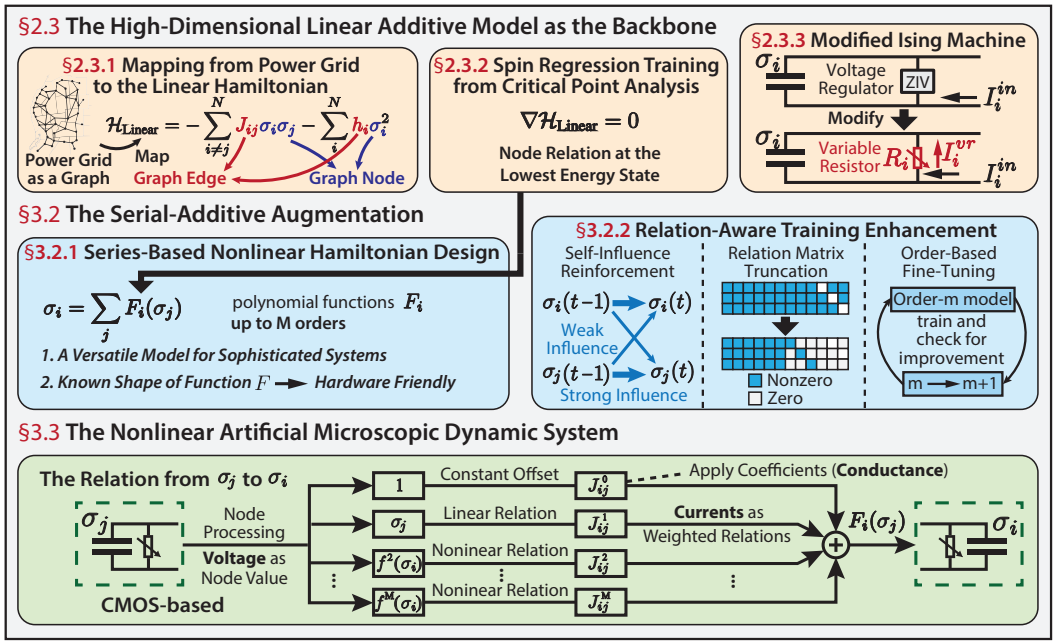

Figure 2: The overview of NP-NDS, a graph learning paradigm utilizing microscopic artificial dynamical systems to predict macroscopic real-world dynamical systems such as power grids.

propagates. In addition to performance fluctuations that can damage electrical devices, detrimental consequences can be triggered. For example, the failure of a single power line in a power grid can cause nearby lines to overload, causing cascading failures [2] that lead to severe total blackouts. To name a few on a mass scale: Northern India in 2012 for 14 hours [3], South Australia in 2016 for up to a week [4], and Southeast South America in 2019 for half a day [5]. Formidably, such events are too fast to respond, as damage is done in only a few seconds [6], vaporizing billions of dollars (in USD) [7]. With power grids constantly evolving in scale and density, this situation escalates, posing threats of significant losses in terms of finance, and even human lives. To minimize the losses inflicted, it becomes imminent to develop an approach to predict the behavior of power grids with high accuracy and ultra-low latency.

Recently, a new Graph Learning (GL) framework has emerged [8–12], achieving over $1000\times$ speedup compared to SOTA GNNs on Nvidia A100 GPUs, while maintaining comparable or even superior accuracy in graph prediction tasks. Within the framework, graph predictions are carried out through rapid natural processes on a dynamical system of electrons. Similar to how water freezes into ice, the electronic system evolves by seeking equilibrium at the lowest energy state, forming an organized pattern in electric charge distribution. By mapping the observed data to the lowest energy state through training, desired results are automatically obtained through a spontaneous decrease in energy function (Hamiltonian). The physical embodiment of the artificial dynamical system is derived from a CMOS-compatible Ising machine [13], which is based on the Ising model rooted in physics, featuring strong connectivity and long-range information propagation in a cascading fashion. Observing the remarkable results, we are motivated to investigate the following opportunity: Can we leverage such artificial microscopic dynamical systems to efficiently model and predict the behavior of the macroscopic dynamical systems – power grids, that demand extremely rapid responses?

Despite the framework's promising characteristics, two significant obstacles remain, fundamentally limiting the effectiveness of this approach when applied to power grids modeled as graphs. First, the Hamiltonian, or the energy function described in the framework only leads to a linear model, capturing mere linear relations between graph nodes. As a result, the expressivity and versatility of the framework are fundamentally restricted, considering that power grids are dynamical systems with sophisticated inter-node relations. Second, although intricate extensions could enable nonlinearity in the machine, they must be carefully designed to ensure the practical complexity of the Hamiltonian, given the constraints imposed by hardware feasibility.

In this paper, we address these problems by proposing NP-NDS, a nature-powered nonlinear dynamical system that accurately predicts real-world graph behaviors such as power grids with lightweight

CMOS-based hardware, as depicted in Figure 2. To highlight, NP-NDS can capture intricate and nonlinear connections among nodes, demonstrating sufficient expressivity and versatility for such a complex system. Meanwhile, it harnesses the inherent computational power within dynamical systems through a Hamiltonian-hardware co-design approach that bridges Hamiltonian characteristics with hardware implementation. **On the Hamiltonian side**, taking the linear model in previous work as the backbone, we further enhance the Hamiltonian's expressivity by integrating the Chebyshev polynomials to capture nonlinear inter-node relations. Consequently, a highly nonlinear Hamiltonian with serial-additive augmentation is constructed, unleashing the power of NP-NDS for complex real-world GL problems such as power grid prediction. To achieve high accuracy, node-relation-aware training enhancement methods are incorporated, including self-influence reinforcement, relation matrix truncation, and order-based fine-tuning. **On the hardware side**, we develop a CMOS-based programmable dynamical system governed by the proposed Hamiltonian following the serial-additive structure. Specifically, the inter-node influences on a node are realized by combining electrical currents modeled after Chebyshev polynomials, allowing the system to efficiently seek the lowest energy state through electron movement. As a result, the hardware dynamical system, embodied as a nature-powered processor, automatically evolves toward the desired results at the "speed of electrons", enabling ns-level inference on NP-NDS.

To the best of our knowledge, NP-NDS is the first work that exploits nature's power within a nonlinear dynamical system to solve real-world graph learning problems, outperforming SOTA GNNs. Our contributions are summarized below.

- We introduce NP-NDS, a CMOS-compatible artificial dynamical system designed to predict the behaviors of complex real-world graphs. By co-designing the Hamiltonian and hardware of the dynamical systems, NP-NDS outperforms state-of-the-art GNN methods.
- We design hardware-friendly Hamiltonians with high expressivity and versatility for a nonlinear dynamical system through serial-additive augmentation, ensuring high accuracy in prediction.
- We develop a microscopic dynamical system that is governed by the proposed Hamiltonian, unleashing the inherent computational power within the system to solve power grid prediction problems at the "speed of electrons".
- Using power grids as representative cases, experimental results demonstrate that NP-NDS delivers 2.3×10^3 speedup (vs. GPU), $10^5\times$ energy saving, and higher accuracy compared to GNNs.

2 Background

2.1 Ising Model

The Ising model is a model originally proposed to study ferromagnetism in statistical physics with only binary variables as "spins" σ. The model is characterized by its Hamiltonian:

$$\mathcal{H}_{\text{Ising}} = -\sum_{i \neq j}^{N} J_{ij}\sigma_i\sigma_j - \sum_{i}^{N} h_i\sigma_i \; ; \;\; \sigma_i \in \{-1, +1\} \tag{1}$$

The Hamiltonian indicates that the pairwise relation between two spins σ_i and σ_j is denoted as J_{ij}, and the self-response of spin σ_i to an external magnetic field is indicated by h_i. In practice, the self-response term is equivalent to a coupling term with an extra spin fixed to "+1", namely, virtual spin. In the binary domain, despite the simplicity of the model, it provides high expressivity due to the vast number of discrete values (2^N, with N being the total number of spins) that the Hamiltonian can take. For conciseness, real-valued variables are also denoted as spins from now on.

2.2 Ising Machine

Ising machines are physical embodiments of the Ising model, which are essentially artificial dynamical systems governed by the Hamiltonian described in the Ising model. Through carefully designed spin dynamics, the systems are capable of spontaneously evolving toward the lowest energy state, offering orders of magnitude speedup compared to conventional digital computers, particularly for tasks such as combinatorial optimization problems [14].

To date, myriad Ising machines have been developed, such as D-Wave the quantum annealer [15], optical coherent Ising machines [16], and electrical coupled oscillators [17]. However, the Ising machines listed above typically pose significant challenges in modification, due to their strict operating conditions, sophisticated manufacturing, or the lack of high-quality hardware components. Consequently, in this work, we choose the CMOS-compatible Ising machine [13] as our basic hardware

substrate, due to its ability to function at room temperature, mature manufacturing technology, and easily controllable hardware components such as resistors and capacitors.

To achieve spontaneous energy decrease, a Lyapunov analysis is performed to design the circuit that defines the spin dynamics. Specifically, the inequality of Hamiltonian $\mathcal{H}$ needs to be satisfied:

$$\frac{d\mathcal{H}}{dt} = \sum_i (\frac{\partial \mathcal{H}}{\partial \sigma_i} \frac{d\sigma_i}{dt}) \leq 0 \tag{2}$$

Reasonably, the spin dynamics can be designed following:

$$\frac{d\sigma_i}{dt} = -\frac{1}{C} \frac{\partial \mathcal{H}}{\partial \sigma_i} \tag{3}$$

with C being a positive constant. The summation in the inequality becomes $\sum_i \frac{-1}{C}(d\sigma_i/dt)^2$, naturally satisfying the inequality with its quadratic shape.

2.3 The Linear Additive Model

The Linear Additive Model (LAM) used in prior work [8, 9] is extended from the Ising model, enabling real-value support and utilizing the spontaneous energy decrease in the selected Ising machine with slight modifications. In this paper, this linear work is taken as the foundation and is therefore briefly introduced here.

2.3.1 The Modified Hamiltonian and Graph Mapping

The Ising Hamiltonian in Eq. 1 is only suitable for discrete spins due to the lack of convexity. If the spin values are naively extended to real values, the spontaneous energy decrease leads to infinitely low energy, leading to divergent spin values. A naive approach is to replace the linear terms in the Ising Hamiltonian (Eq. 1) by quadratic terms:

$$\mathcal{H}_{\text{Linear}} = -\sum_{i \neq j}^{N} J_{ij}\sigma_i\sigma_j - \sum_{i}^{N} h_i\sigma_i^2 \; ; \;\; \sigma_i \in \mathbb{R} \tag{4}$$

With negative parameters $\mathbf{h}$, the Hamiltonian is granted a global minimum, allowing the dynamical system to evolve towards the lowest energy state.

The inter-spin relations and the self-response depicted in the model are particularly compatible with graph concepts. It is logical to consider the values of nodes in a graph as analogous to spin values in the model, where the edges and self-loops of the graph correspond to the parameters $\mathbf{J}$ and $\mathbf{h}$, respectively. In Ising Graph Learning (IGL), these parameters $\mathbf{J}$ and $\mathbf{h}$ are trainable based on the historical data of the graph nodes, and are then used to forecast future values of the nodes. Figure 3 provides an example of the predictive mechanism, in which the node values in the next time step are estimated from their values in the current time step. As depicted in the figure, the graph consists of N nodes, with each node in the graph linked to a pair of spins in the Ising model, representing the node's state at time t and its subsequent state at time $t+1$.

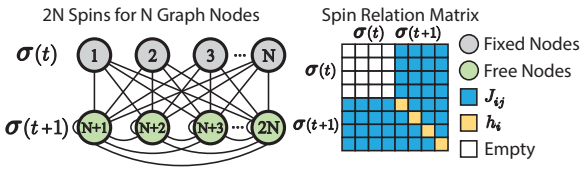

Figure 3: Example: predicting node values at time t+1 based on the data at time t in IGL paradigm.

During the inference process, the spins related to the known node values are kept constant, whereas their influences propagate through the spins to be predicted. As the spins evolve, the Hamiltonian of the dynamical system naturally decreases, moving towards the lowest energy state. To escape local minima, IGL is usually accompanied by an annealing process. Eventually, the lowest energy state is reached, with the predicted spin values obtained as the inference result.

2.3.2 Training with Spin Regression

The spin relations at the lowest energy state are obtained by taking the gradient of the Hamiltonian.

$$\frac{\partial \mathcal{H}_{\text{Linear}}}{\partial \sigma_i} = 0 \;\; \Rightarrow \;\; \sigma_i = \frac{-1}{2h_i} \sum_{i \neq j}^{N} (J_{ij} + J_{ji})\sigma_j \tag{5}$$

It indicates that any spin σ_i follows a linear additive relation at the lowest energy state. Since σ_i is clearly modeled in the equation, a direct training process can be carried out by minimizing the difference between the modeled σ_i and the ground truth, effectively mapping the desired result to the lowest energy state, allowing the dynamical system to perform inference.

2.3.3 The Modified Ising Machine

In Figure 2, the primary modification in the hardware of the vanilla Ising machine is to replace the voltage regulator with a variable resistor. In the original setup [13], the voltage regulator, designated as "ZIV," confines the capacitor's voltage (or spin value) to either +1 or -1, suitable for binary applications. With the variable resistor, the voltage is now influenced by the the resistor, the electric current I_i^{vr} flowing through the resistor, and the total input current I_i^{in}. The spin dynamics satisfy:

$$\frac{d\sigma_i}{dt} = \frac{1}{C}(I_i^{in} - I_i^{rv}) = \frac{1}{C}(\sum_{i \neq j}^{N}(J_{ij} + J_{ji})\sigma_j - 2h_i\sigma_i) \tag{6}$$

where $2h_i$ is the inverse of resistance R_i, which essentially leads to the term "$2h_i\sigma_i$" in the equation, effectively corresponds to the quadratic term in the Hamiltonian as described in the model (See Eq. 4, note that $\nabla(h_i\sigma_i^2) = 2h_i\sigma_i$).

3 Methods

3.1 Overview

Despite the real-value support and the use of a rapidly evolving dynamical system, the complexity of LAM's Hamiltonian (described and discussed in Sec. 2.3) is compromised for hardware feasibility, causing the lack of nonlinearity. Consequently, its ability to accurately represent sophisticated relations between real-world observables is hampered. To this end, we propose a Hamiltonian-hardware co-design paradigm that bridges hardware with Hamiltonians of high complexity, breaking the trade-off between model complexity and hardware feasibility with the following steps.

(1) We design the spin relations by modifying the gradient of the LAM's Hamiltonian, incorporating nonlinear relations in the designed Hamiltonian. (2) To ensure the versatility of the model to represent various relations, the nonlinear relations are approximated using series, taking the coefficients of the series as trainable parameters. (3) Three training enhancement methods are adopted to improve the self-influence of nodes in power grids, to eliminate redundant relations that may induce significant error, and to fine-tune the model by gradually adding higher-order terms in the series. (4) We demonstrate that the Hamiltonian design is highly compatible with hardware since the dynamics of spins on hardware are also derived from the gradient of the Hamiltonian.

3.2 The Serial-Additive Augmentation

3.2.1 Series-Based Hamiltonian Design

The Hamiltonian design methodology is derived from Eq. 5. Rather than starting from an analytical Hamiltonian such as $\mathcal{H}_{\text{Linear}}$, we begin with the equation on the right hand side, which essentially represents the spin relations at the lowest energy state. As a result, the gradient of the Hamiltonian can be directly designed with a general form:

$$\frac{\partial \mathcal{H}}{\partial \sigma_i} = h_i\sigma_i + \sum_{j \neq k}^{N}(P_{jk}f(\sigma_j) + Q_{jk}g(\sigma_j) + ...) \tag{7}$$

where P_{jk} and Q_{jk} are parameters to be trained, $f(\sigma_j)$ and $g(\sigma_j)$ are functions of σ_j. Compared to Eq. 5, two notable improvements are achieved. First, the original linear relations are generalized into flexible functions, allowing for custom model complexity. Second, the symmetric relation $(J_{ij} + J_{ji})$ between two spins is no longer compulsory, enabling more realistic asymmetric relations.

To properly capture the spin relations for a wide range of applications, functions such as $f(\sigma_j)$ and $g(\sigma_j)$ must be carefully designed. In the following discussions, a guideline is provided to design the functions in a systematic manner.

To design spin relations following the methodology above, we aim to achieve two key objectives. First, for practicality, the model should be hardware-friendly, preferably analytical, to guide the

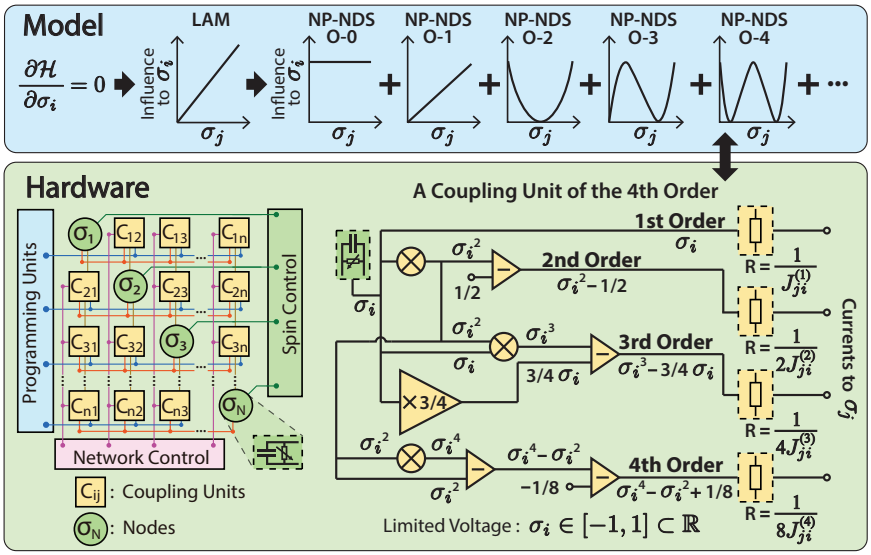

Figure 4: The model and hardware implementation schematics based on the Chebyshev series.

hardware design. Second, a versatile "one-for-all" model is desired. This not only provides a general model but also eliminates the need for expensive hardware customization for various spin relations.

With these objectives in mind, our approach is to approximate the spin relations with series. Specifically, the spin relations at $\nabla\mathcal{H} = 0$ are generally designed as:

$$\sigma_i = -\frac{1}{2h_i}\sum_{m}^{M}\sum_{i\neq j}^{N} J_{ij}^{m} f^{m}(\sigma_j) \tag{8}$$

It shows that the relation between σ_i and σ_j is represented as M-order series, with $f^m(\sigma_j)$ being the m'th term in the series. Similar to the original Ising model, the 0'th order relation is represented by a virtual spin and therefore omitted in the following discussions. It is worth noting that this approach not only shows promising practicality and versatility but also demonstrates high interpretability. By decomposing the inter-spin relations into separate components with trainable coefficients J_{ij}^m, the strength of a particular component is quantified, thereby offering a clearer understanding of the underlying mechanisms governing real-world phenomena.

With the general series form of inter-spin relations established, an immediate follow-up question is: What series to choose? In this work, the Chebyshev polynomials of the first kind are selected for the following features: (1) Chebyshev polynomials are typically used for series expansion, as it is often used to mitigate Runge's phenomenon in high-order polynomials. (2) Compared to trigonometric series such as the Fourier series, the hardware implementation for polynomials is considerably straightforward. While inductors can be used to directly achieve periodicity in trigonometric series, the lack of high-quality inductors and their large size pose significant challenges. Conversely, polynomials offer a simpler implementation path, as they require just multiplication and addition operations. (3) Every term $f^m(\sigma_j)$ in the polynomial is confined between -1 and 1 for $\sigma_j \in [-1, 1]$. As σ_j and $f^m(\sigma_j)$ are represented by voltages, these non-divergent terms are exceptionally valuable for practical implementation. To provide an intuitive view, the leading terms of the Chebyshev polynomials are shown on the Model side in Figure 4. Conveniently, the 1st order of the polynomials (NP-NDS O-1) is a linear term, preserving the linear relations from the backbone model.

3.2.2 Training Enhancement Methods

In training, we inherit the straightforward yet effective regression method for LAM introduced in Sec. 2.3.2. With the serial additive model established, we focus on improving its training by implementing three enhancement methods for graph prediction as shown in Figure 5, where the original relation matrix depicts the general relation from the spin series of $\boldsymbol{\sigma}(t)$ and $\boldsymbol{\sigma}(t+1)$ to the spins $\boldsymbol{\sigma}(t+1)$, where $\boldsymbol{\sigma}(t+1)$ are the spins to be predicted.

Self-Influence Reinforcement. Based on empirical observations, the relation of a power grid node with its historical values outweighs its relation with other nodes. To differentiate the relations, the

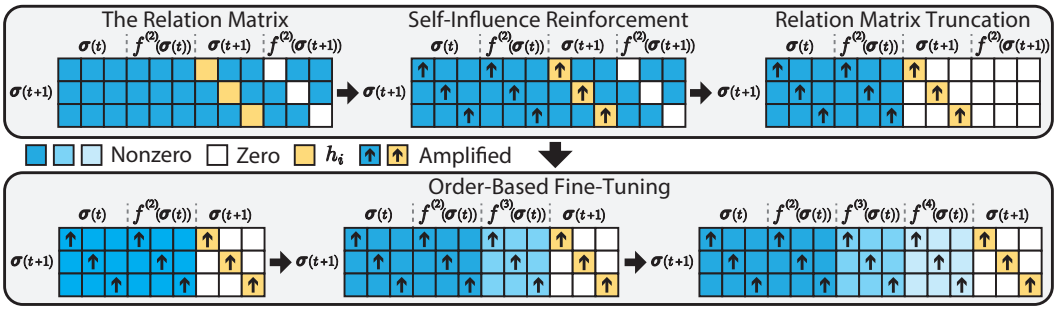

Figure 5: The three training enhancement methods. Each square represents a coefficient J_{ij}^m.

self-influence, or the diagonal lines in the square sub-matrices in Figure 5 are amplified, as indicated by the arrows. The amplification is performed by multiplying a constant factor, which is considered a hyperparameter. Note that the self-response parameter h_i is also amplified, as the summations in Eq. 8 typically require a sufficiently large h_i for a limited range of σ_i.

Relation Matrix Truncation. This approach deliberately removes the off-diagonal elements in the sub-matrices describing the relations between spins to be predicted $\boldsymbol{\sigma}(t+1)$ and themselves. Despite that the relations contribute to the model's expressivity, errors are introduced and amplified through spin evolution following the relations, causing a destructive influence on the accuracy. To this end, the error sources are removed to preserve clean spin relations.

Order-Based Fine-Tuning. The complexity of the model apparently increases as more terms of the series are included. However, more terms do not always lead to high accuracy, as they may cause disruptive influences on the actual significant terms during training. Based on this observation, without prior knowledge of the optimal number of series terms, we first train a basic model using only the lower-order terms to capture the relatively simple relations, which are usually predominant. Subsequently, we increase the model size by adding higher-order terms as corrections while preserving the previously trained parameters, and initiate a fine-tuning process to include higher-order relations until the maximum order is reached, or the validation accuracy stops improving. As illustrated in Figure 5, the number of parameters scales linearly with the number of series terms, demonstrating good scalability in model sizes.

3.3 Hardware Implementation

In accordance with the model design, the hardware design is also closely related to the Hamiltonian's gradient, which directly corresponds to the spin dynamics described by $d\sigma_i/dt$ as designed in Eq. 3. The architecture of a spin shown in Figure 4 indicates that the spin dynamics is determined by the current I_i^{vr} flowing through the variable resistor and the total input current I_i^{in} from the system. Since the gradient is already defined in the model (Eq. 7), our task is to match the behavior of currents with the gradient. Note that the value of I_i^{vr} obtained from LAM already fulfills the condition $I_i^{vr} = 2h_i\sigma_i$ as stated by Ohm's Law, our attention is thus directed towards I_i^{in}. Specifically, $I_i^{in} = \sum_m^M \sum_{i\neq j}^N J_{ij}^m f^m(\sigma_j)$. This implies that the inter-spin relations are physically interpreted as the superposition of electrical currents. Originally, a coupling unit combines $(J_{ij} + J_{ji})$ and σ_j in the Ising model and LAM, producing partial input currents to the corresponding spins. In NP-NDS, the coupling units are upgraded to produce multiple current components according to the series terms. During the process, to prevent voltages from exceeding limits, the series terms are scaled down by a factor (e.g., the second order $2\sigma_i^2 - 1$ becomes $\sigma_i^2 - 1/2$), leaving the factor embedded in the downstream resistor. Eventually, the current components are merged as I_j^{in}, and delivered to the j'th spin. This work employs dynamical systems with up to the 4th order of the Chebyshev polynomial.

4 Evaluation

4.1 Experimental Setup

Datasets. We evaluate NP-NDS with three datasets listed below, covering different aspects of power grids. The datasets are divided into 70% for training, 20% for validation, and 10% for testing.

Table 1: Accuracy comparison of NDS-GL variances: the lower the better - best results are in bold.

Dataset	PMU-PA		PMU-V		PowerFlow	
Metric	MAE	RMSE	MAE	RMSE	MAE	RMSE
NDS-O1	1.88e-2	2.31e-2	5.20e-2	6.78e-2	0.396	0.687
NDS-O2	1.41e-2	1.77e-2	5.15e-2	6.66e-2	0.395	0.690
NDS-O3	1.33e-2	1.70e-2	4.97e-2	6.61e-2	0.391	0.685
NDS-O4	1.26e-2	1.68e-2	4.75e-2	6.46e-2	0.394	0.690
NDS-FT-O2	1.27e-2	1.71e-2	5.05e-2	6.71e-2	0.387	0.681
NDS-FT-O3	1.16e-2	1.54e-2	4.61e-2	6.21e-2	0.379	**0.662**
NDS-FT-O4	**1.08e-2**	**1.43e-2**	**4.27e-2**	**6.02e-2**	**0.377**	0.664

- *PMU-PA* – Second-level Phase Angle prediction of Phasor Measurement Unit (PMU) data on a synthetic network in Texas [18].
- *PMU-V* – Second-level Voltage data (in mV) prediction on a synthetic network in Texas [18].
- *PowerFlow* – Prediction of hourly power flow in MW on a synthetic network in Texas [19].

Baselines. We compare the accuracy and inference of NP-NDS with four SOTA spatial-temporal GNNs that are capable of capturing a graph's logical topology for fair evaluations. These GNNs include *Graph WaveNet* [20], *MTGNN* [21], *DDGCRN* [22], and *MegaCRN* [23]. Their hyperparameters are set based on their released code.

Platforms. The inference latency of the baseline GNNs is measured using an NVIDIA A100-40GB GPU and an 18-core, 36-thread Intel Xeon Gold 6140 CPU. The dynamical system used for NP-NDS is adapted from the current SOTA Ising machine, BRIM [13], based on the Cadence Analog Design Environment.

4.2 Accuracy Evaluation

Figure 6 depicts the MAE and RMSE results of predicting the $(t+1)$'th snapshot based on the (t)'th snapshot. The comparison involves LAM, NP-NDS, and four SOTA GNNs that are capable of extracting the logical topology of graphs. Specifically, NDS-best shows the best NP-NDS result of up to the 4th series order with order-based fine-tuning taken into consideration. In the LAM and NP-NDS implementations, L1 loss is used to mitigate the effect of outliers. The results demonstrate that the NDS-best, incorporated with nonlinearity and custom model complexity, is generally much more accurate than the high-dimensional linear model LAM, and significantly outperforms the best GNN results, with 23.6% reduced MAE and 28.2% reduced RMSE averaged over all three power grid datasets.

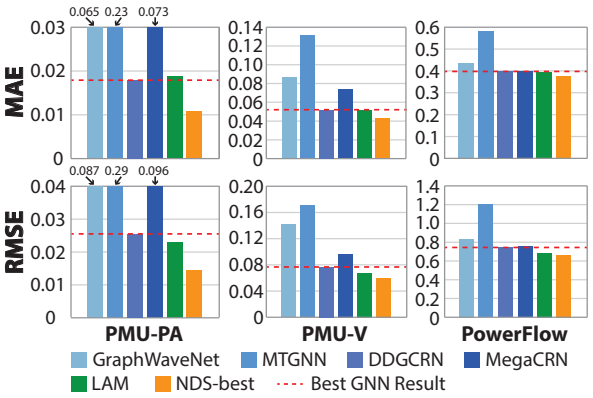

Figure 6: Accuracy comparison with SOTA GNNs across three datasets in MAE and RMSE. Lower is better.

4.3 Latency Evaluation

Figure 7 shows the inference latency results. Compared to the average latency of SOTA GNNs, NP-NDS delivers 3.2×10^4 speedup over CPU and 2.3×10^3 over GPU for all three datasets. In addition, the power consumption of the dynamical system is mere $\sim$2 W, more than 50$\times$ more efficient in Watts than modern CPUs and GPUs, which ultimately contributes to over $10^5\times$ reduction in energy consumption compared to GNNs on CPUs and GPUs.

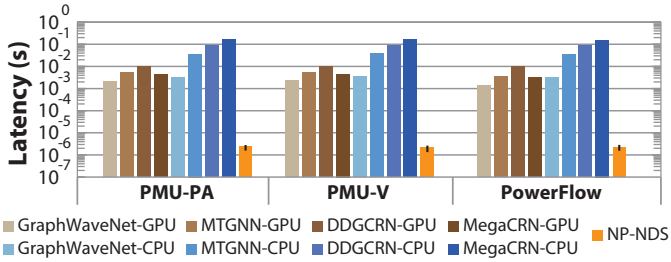

Figure 7: Latency comparison: NP-NDS vs SOTA GNNs.

4.4 Ablation Study

In this section, we investigate how prediction is impacted by series order and fine-tuning. To this end, we examine the results of NP-NDS with different orders, ranging from 1st to 4th order (NDS-O1 to O4), as well as the results with fine-tuning for orders 2 to 4 (NDS-FT-O2 to O4). These results are demonstrated in Table 1, where the higher-ordered implementations generally yield better results. However, when fine-tuning is not applied, the MAE/RMSE may increase for higher orders. Conversely, the order-based fine-tuning approach tends to maintain the MAE/RMSE achieved in the lower-ordered models, while further enhancing the accuracy with higher-order corrections.

5 Related Work

5.1 Graph Neural Networks

GNNs have become a cornerstone in machine learning for graph-structured data, enabling effective predictions and classifications. Their ability to model complex relational data has led to widespread applications across various domains, including physics [24], social science [25], bioinformatics [26], and chemistry [27]. In GNNs, information is propagated through message passing among nodes – typically, each node collects and updates information from neighboring nodes, effectively capturing the relations between graph embeddings and target outputs. The major areas of GNN research include, but are not limited to, the following fields:

Graph Convolutional Networks (GCNs) [28–31] extend convolutional neural networks to graphs, enabling efficient learning on non-Euclidean data structures. GCNs have been widely adopted in, e.g., e-commerce [32, 33], natural language processing [34], internet of things [35], chemical reactivity [36], material sciences [37] and pharmaceutical sciences [38]. Limited by the latency and throughput requirements by those applications, GCN accelerators have also emerged [28, 29, 39–45].

Graph Attention Networks (GATs) [46] enhance GNNs by integrating attention mechanisms that allow nodes to assign different importance weights. Subsequent advancements[47] have extended GATs to support heterogeneous graphs containing multiple types of nodes and links. Further developments have introduced dynamic attention mechanisms [48], enhancing the adaptability and expressiveness. GATs have also been widely adopted across various research areas [49–52].

GraphSAGE [53] enables efficient inductive learning by leveraging node feature information to generate embeddings for unseen data, eliminating the need to retrain the entire model as required by traditional transductive methods. Subsequent works include: FastGCN [54] addresses sampling inefficiency by applying importance sampling; Cluster-GCN [55] improves scalability by clustering the graph and performing mini-batch training on these clusters; GraphSAINT [56] introduces a graph sampling-based inductive learning method that reduces computation and memory consumption.

5.2 General Additive Models

Generalized additive models [57] are a statistical modeling technique that incorporates the relations between variables. The related work dates back to a decade ago: GAM [58] employs decision trees for regression and classification, achieving outstanding model interpretability. Its follow-up work GA^{2}M [59] further includes pairwise coupling interactions to improve model complexity. Recently, neural network based additive models have also emerged. In NAM [60], individual neural networks are employed to capture complex relationships between variables. Furthermore, this research highlights the extensive influence of additive models. For example, scientific insights can be obtained from such interpretable models and utilized to guide studies in natural sciences. Our work is clearly distinguished from others, as we exploit the inherent power of dynamical systems to address real-world dynamical system problems.

6 Conclusion

In this work, NP-NDS is proposed to address real-world graph learning problems with a physical dynamical system. incorporating complicated nonlinear graph node relations through a Hamiltonian-hardware co-design. As a result, the inherent computing power in the dynamical system is unleashed. In particular, an average 2.3×10^3 speedup is achieved compared to a Nvidia A100 GPU on various GNNs, with 23.6% reduced MAE and 28.2% reduced RMSE averaged over three datasets for power grids, as well as approximately $10^5\times$ power saving compared to modern CPUs and GPUs.

Acknowledgement

This work is supported by the U.S. DOE Office of Electricity through its Advanced Grid Modeling (AGM) program and by ComPort: Rigorous Testing Methods to Safeguard Software Porting, the U.S. DOE, Office of Science, Office of Advanced Scientific Computing Research, under Award Number 78284. This work is also supported by NSF under Awards SHF-2326494. The Pacific Northwest National Laboratory is operated by Battelle for the U.S. DOE under Contract DE-AC05-76RL01830.

References

[1] Hannah Ritchie and Pablo Rosado. Electricity mix. *Our World in Data*, 2020. https://ourworldindata.org/electricity-mix.

[2] Vaiman, Bell, Chen, Chowdhury, Dobson, Hines, Papic, Miller, and Zhang. Risk assessment of cascading outages: Methodologies and challenges. *IEEE Transactions on Power Systems*, 27 (2):631–641, 2012. doi: 10.1109/TPWRS.2011.2177868.

[3] Central Electricity Regulatory Commission et al. Report on the grid disturbances of 30th july and 31st july 2012. *New Delhi: Central Electricity Regulatory Commission*, 2012.

[4] Ruifeng Yan, Tapan Kumar Saha, Feifei Bai, Huajie Gu, et al. The anatomy of the 2016 south australia blackout: A catastrophic event in a high renewable network. *IEEE Transactions on Power Systems*, 33(5):5374–5388, 2018.

[5] Gonzalo E. Alvarez. A novel strategy to restore power systems after a great blackout. the argentinean case. *Energy Strategy Reviews*, 37:100685, 2021. ISSN 2211-467X. doi: https://doi.org/10.1016/j.esr.2021.100685. URL https://www.sciencedirect.com/science/article/pii/S2211467X21000717.

[6] Benjamin Schäfer, Dirk Witthaut, Marc Timme, and Vito Latora. Dynamically induced cascading failures in power grids. *Nature communications*, 9(1):1975, 2018.

[7] Electricity Consumers Resource Council. The economic impacts of the august 2003 blackout. *Washington, DC*, 2004.

[8] Chunshu Wu, Ruibing Song, Chuan Liu, Yunan Yang, Ang Li, Michael Huang, and Tong Geng. NP-GL: Extending power of nature from binary problems to real-world graph learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=qT7DXUmX7j.

[9] Ruibing Song, Chunshu Wu, Chuan Liu, Ang Li, Michael Huang, and Tong Geng. DS-GL: Advancing graph learning via harnessing the power of nature within dynamic systems. In *Proceedings of the 51st Annual International Symposium on Computer Architecture*. Association for Computing Machinery, 2024.

[10] Liu Chuan, Wu Chunshu, Song Ruibing, Chen Yousu, Li Ang, Huang Michael, and Geng Tony. Nature-gl: A revolutionary learning paradigm unleashing nature's power in real-world spatial-temporal graph learning. In *the 30th Asia and South Pacific Design Automation Conference*, 2025.

[11] Zhenyu Pan, Anshujit Sharma, Jerry Yao-Chieh Hu, Zhuo Liu, Ang Li, Han Liu, Michael Huang, and Tony Geng. Ising-traffic: Using ising machine learning to predict traffic congestion under uncertainty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9354–9363, 2023.

[12] Zhuo Liu, Yunan Yang, Zhenyu Pan, Anshujit Sharma, Amit Hasan, Caiwen Ding, Ang Li, Michael Huang, and Tong Geng. Ising-cf: A pathbreaking collaborative filtering method through efficient ising machine learning. In *Proceedings of the 60th ACM/IEEE Design Automation Conference. of DAC*, 2023.

[13] Richard Afoakwa, Yiqiao Zhang, Uday Kumar Reddy Vengalam, Zeljko Ignjatovic, and Michael Huang. Brim: Bistable resistively-coupled ising machine. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 749–760. IEEE, 2021.

[14] Naeimeh Mohseni, Peter L McMahon, and Tim Byrnes. Ising machines as hardware solvers of combinatorial optimization problems. *Nature Reviews Physics*, 4(6):363–379, 2022.

[15] R. Harris, M. W. Johnson, T. Lanting, A. J. Berkley, J. Johansson, P. Bunyk, E. Tolkacheva, E. Ladizinsky, N. Ladizinsky, T. Oh, F. Cioata, I. Perminov, P. Spear, C. Enderud, C. Rich, S. Uchaikin, M. C. Thom, E. M. Chapple, J. Wang, B. Wilson, M. H. S. Amin, N. Dickson, K. Karimi, B. Macready, C. J. S. Truncik, and G. Rose. Experimental investigation of an eight-qubit unit cell in a superconducting optimization processor. *Phys. Rev. B*, 82:024511, Jul 2010. doi: 10.1103/PhysRevB.82.024511.

[16] Takahiro Inagaki, Yoshitaka Haribara, Koji Igarashi, Tomohiro Sonobe, Shuhei Tamate, Toshimori Honjo, Alireza Marandi, Peter L. McMahon, Takeshi Umeki, Koji Enbutsu, Osamu Tadanaga, Hirokazu Takenouchi, Kazuyuki Aihara, Ken ichi Kawarabayashi, Kyo Inoue, Shoko Utsunomiya, and Hiroki Takesue. A coherent ising machine for 2000-node optimization problems. *Science*, 354(6312):603–606, 2016. doi: 10.1126/science.aah4243.

[17] Tianshi Wang and Jaijeet Roychowdhury. Oim: Oscillator-based ising machines for solving combinatorial optimisation problems. In Ian McQuillan and Shinnosuke Seki, editors, *Unconventional Computation and Natural Computation*, pages 232–256, Cham, 2019. Springer International Publishing. ISBN 978-3-030-19311-9.

[18] Hanyue Li, Ashly L Bornsheuer, Ti Xu, Adam B Birchfield, and Thomas J Overbye. Load modeling in synthetic electric grids. In *2018 IEEE Texas Power and Energy Conference (TPEC)*, pages 1–6. IEEE, 2018.

[19] Ti Xu, Adam B Birchfield, Komal S Shetye, and Thomas J Overbye. Creation of synthetic electric grid models for transient stability studies. In *The 10th Bulk Power Systems Dynamics and Control Symposium (IREP 2017)*, pages 1–6, 2017.

[20] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, and Chengqi Zhang. Graph wavenet for deep spatial-temporal graph modeling, 2019.

[21] Zonghan Wu, Shirui Pan, Guodong Long, Jing Jiang, Xiaojun Chang, and Chengqi Zhang. Connecting the dots: Multivariate time series forecasting with graph neural networks, 2020.

[22] Wenchao Weng, Jin Fan, Huifeng Wu, Yujie Hu, Hao Tian, Fu Zhu, and Jia Wu. A decomposition dynamic graph convolutional recurrent network for traffic forecasting. *Pattern Recognition*, 142:109670, 2023. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2023.109670.

[23] Renhe Jiang, Zhaonan Wang, Jiawei Yong, Puneet Jeph, Quanjun Chen, Yasumasa Kobayashi, Xuan Song, Shintaro Fukushima, and Toyotaro Suzumura. Spatio-temporal meta-graph learning for traffic forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 8078–8086, 2023.

[24] Jonathan Shlomi, Peter Battaglia, and Jean-Roch Vlimant. Graph neural networks in particle physics. *Machine Learning: Science and Technology*, 2(2):021001, 2020.

[25] Liangwei Yang, Zhiwei Liu, Yingtong Dou, Jing Ma, and Philip S. Yu. Consisrec: Enhancing gnn for social recommendation via consistent neighbor aggregation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, page 2141–2145, 2021.

[26] Hai-Cheng Yi, Zhu-Hong You, De-Shuang Huang, and Chee Keong Kwoh. Graph representation learning in bioinformatics: trends, methods and applications. *Briefings in Bioinformatics*, 23(1): bbab340, 2021.

[27] Justin Gilmer, Samuel S. Schoenholz, Patrick F. Riley, Oriol Vinyals, and George E. Dahl. Neural message passing for quantum chemistry. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1263–1272. JMLR.org, 2017.

[28] Tong Geng, Ang Li, Runbin Shi, Chunshu Wu, Tianqi Wang, Yanfei Li, Pouya Haghi, Antonino Tumeo, Shuai Che, Steve Reinhardt, and Martin C. Herbordt. Awb-gcn: A graph convolutional network accelerator with runtime workload rebalancing. MICRO '20, .

[29] Tong Geng, Chunshu Wu, Yongan Zhang, Cheng Tan, Chenhao Xie, Haoran You, Martin Herbordt, Yingyan Lin, and Ang Li. I-gcn: A graph convolutional network accelerator with runtime locality enhancement through islandization. MICRO '21, .

[30] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2016.

[31] Haoran You, Tong Geng, Yongan Zhang, Ang Li, and Yingyan Lin. Gcod: Graph convolutional network acceleration via dedicated algorithm and accelerator co-design. In *2022 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 460–474. IEEE, 2022.

[32] Rong Zhu, Kun Zhao, Hongxia Yang, Wei Lin, Chang Zhou, Baole Ai, Yong Li, and Jingren Zhou. Aligraph: A comprehensive graph neural network platform. *arXiv preprint arXiv:1902.08730*, 2019.

[33] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L. Hamilton, and Jure Leskovec. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '18, page 974–983, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3219890. URL https://doi.org/10.1145/3219819.3219890.

[34] Jasmijn Bastings, Ivan Titov, Wilker Aziz, Diego Marcheggiani, and Khalil Sima'an. Graph convolutional encoders for syntax-aware neural machine translation. In Martha Palmer, Rebecca Hwa, and Sebastian Riedel, editors, *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1957–1967, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1209. URL https://aclanthology.org/D17-1209.

[35] Huy-Trung Nguyen, Quoc-Dung Ngo, and Van-Hoang Le. Iot botnet detection approach based on psi graph and dgcnn classifier. In *2018 IEEE international conference on information communication and signal processing (ICICSP)*, pages 118–122. IEEE, 2018.

[36] Connor W Coley, Wengong Jin, Luke Rogers, Timothy F Jamison, Tommi S Jaakkola, William H Green, Regina Barzilay, and Klavs F Jensen. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chemical science*, 10(2):370–377, 2019.

[37] Tian Xie and Jeffrey C Grossman. Crystal graph convolutional neural networks for an accurate and interpretable prediction of material properties. *Physical review letters*, 120(14):145301, 2018.

[38] Marinka Zitnik, Monica Agrawal, and Jure Leskovec. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 2018.

[39] Sergi Abadal, Akshay Jain, Robert Guirado, Jorge López-Alonso, and Eduard Alarcón. Computing graph neural networks: A survey from algorithms to accelerators. *ACM Computing Surveys (CSUR)*, 54(9):1–38, 2021.

[40] Adam Auten, Matthew Tomei, and Rakesh Kumar. Hardware acceleration of graph neural networks. In *2020 57th ACM/IEEE Design Automation Conference (DAC)*, pages 1–6. IEEE, 2020.

[41] Xiaobing Chen, Yuke Wang, Xinfeng Xie, Xing Hu, Abanti Basak, Ling Liang, Mingyu Yan, Lei Deng, Yufei Ding, Zidong Du, et al. Rubik: A hierarchical architecture for efficient graph learning. *arXiv preprint arXiv:2009.12495*, 2020.

[42] Kevin Kiningham, Philip Levis, and Christopher Ré. Grip: A graph neural network accelerator architecture. *IEEE Transactions on Computers*, 72(4):914–925, 2022.

[43] Shengwen Liang, Ying Wang, Cheng Liu, Lei He, LI Huawei, Dawen Xu, and Xiaowei Li. Engn: A high-throughput and energy-efficient accelerator for large graph neural networks. *IEEE Transactions on Computers*, 70(9):1511–1525, 2020.

[44] Mingyu Yan, Lei Deng, Xing Hu, Ling Liang, Yujing Feng, Xiaochun Ye, Zhimin Zhang, Dongrui Fan, and Yuan Xie. Hygcn: A gcn accelerator with hybrid architecture. In *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 15–29. IEEE, 2020.

[45] Yongan Zhang, Haoran You, Yonggan Fu, Tong Geng, Ang Li, and Yingyan Lin. G-cos: Gnn-accelerator co-search towards both better accuracy and efficiency. In *2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD)*, pages 1–9. IEEE, 2021.

[46] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[47] Xiao Wang, Houye Ji, Chuan Shi, Bai Wang, Yanfang Ye, Peng Cui, and Philip S Yu. Heterogeneous graph attention network. In *The World Wide Web Conference*, WWW '19, page 2022–2032, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366748. doi: 10.1145/3308558.3313562. URL https://doi.org/10.1145/3308558.3313562.

[48] Shaked Brody, Uri Alon, and Eran Yahav. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*, 2021.

[49] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. Kgat: Knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 950–958, 2019.

[50] Linjie Li, Zhe Gan, Yu Cheng, and Jingjing Liu. Relation-aware graph attention network for visual question answering. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10313–10322, 2019.

[51] Yuke Wang, Ling Lu, Yang Wu, and Yinong Chen. Polymorphic graph attention network for chinese ner. *Expert Systems with Applications*, 203:117467, 2022.

[52] Yanni Dong, Quanwei Liu, Bo Du, and Liangpei Zhang. Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification. *IEEE Transactions on Image Processing*, 31:1559–1572, 2022.

[53] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[54] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: fast learning with graph convolutional networks via importance sampling. *arXiv preprint arXiv:1801.10247*, 2018.

[55] Wei-Lin Chiang, Xuanqing Liu, Si Si, Yang Li, Samy Bengio, and Cho-Jui Hsieh. Cluster-gcn: An efficient algorithm for training deep and large graph convolutional networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 257–266, 2019.

[56] Hanqing Zeng, Hongkuan Zhou, Ajitesh Srivastava, Rajgopal Kannan, and Viktor Prasanna. Graphsaint: Graph sampling based inductive learning method. *arXiv preprint arXiv:1907.04931*, 2019.

[57] Trevor Hastie and Robert Tibshirani. Generalized additive models: Some applications. *Journal of the American Statistical Association*, 82(398):371–386, 1987. doi: 10.1080/01621459.1987.10478440.

[58] Yin Lou, Rich Caruana, and Johannes Gehrke. Intelligible models for classification and regression. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 150–158, 2012.

[59] Yin Lou, Rich Caruana, Johannes Gehrke, and Giles Hooker. Accurate intelligible models with pairwise interactions. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 623–631, 2013.

[60] Rishabh Agarwal, Levi Melnick, Nicholas Frosst, Xuezhou Zhang, Ben Lengerich, Rich Caruana, and Geoffrey E Hinton. Neural additive models: Interpretable machine learning with neural nets. *Advances in neural information processing systems*, 34:4699–4711, 2021.

A Implementation Details

The Rprop optimizer is utilized with a customized early-stop mechanism for all experiments conducted on the six datasets. The experiments employ the same set of hyperparameters, except for the Self-Influence Reinforcement Factors (SIRF), which are specified in Table 2.

The remaining hyperparameters are listed in Table 3. LR is the initial learning rate, Etas and Step Sizes are embedded in the Rprop optimizer. LR Decay and Step Decay are related to the early stop mechanism – if the loss does not decrease in 100 consecutive epochs, the learning rate is halved. If the learning rate reaches the early stop threshold of 1E-6, training is completed.

Table 2: Self-Influence Reinforcement Factors (SIRF) for Different Datasets.

Dataset	PMU-PA	PMU-V	PowerFlow
SIRF	1000	50	1000

Table 3: The universal hyperparameters used for all datasets.

Minibatch	LR	Etas	Step Sizes	LR Decay	Step Decay	Early Stop	Random Seed
64	0.01	(0.5, 1.2)	(1E-4, 50)	0.5	100	1E-6	123

B Social Impacts

The development of NP-NDS represents a significant stride in the realm of power grid forecasting, with implications extending beyond technological advancement. By addressing the pressing need for accurate and rapid predictions in power grid behaviors, NP-NDS holds the potential to mitigate the socio-economic repercussions of power grid failures. Such failures, as evidenced by past incidents in regions like Northern India, South Australia, and Southeast South America, not only disrupt daily activities but also incur substantial financial losses and jeopardize public safety. NP-NDS, with its ability to efficiently model complex inter-node relations and provide ultra-low latency predictions, offers a proactive approach to safeguarding against such disruptions. By enhancing the resilience and reliability of power grids, NP-NDS contributes to the stability of essential services, fosters economic productivity, and ultimately enhances the quality of life for communities reliant on electricity infrastructure.