Bol. Soc. Paran. Mat.
©SPM – E-ISSN-2175-1188
SPM: www.spm.uem.br/bspm

(3s.) **v. 2026 (44) 7** : 1–20.
ISSN-0037-8712
doi:10.5269/bspm.81615

Quantum Graph-Based Modelling of Multi-Step Phase Dynamics in Coupled Oscillator Networks

Surbhi Sharma, Rashmi Bhardwaj and Debabrata Datta

ABSTRACT: Complex networked dynamical systems modeling and prediction are paramount in many fields, including neuroscience, power grids, social and ecological network modeling and prediction. Conventional graph neural networks (GNNs) are not optimal at the high dimensional and oscillatory dynamics of such systems. Quantum computing offers an alternative conceptualization of the study of such complex dynamics because quantum circuits are highly expressive and phases are encoded naturally. This paper introduces a Quantum Graph Neural Oscillator (QGNO) model that is used to simulate the multi-step phase dynamics of Kuramoto oscillator networks modeled as graphs. The framework combines parametrized quantum circuits (PQCs), phase regression (using circular mean squared error) with a decaying hybrid teacher forcing rollout process to ensure stability in long-term predictions. Experiments on a 3-node Kuramoto network demonstrate that QGNO achieves low training loss and accurately tracks phase trajectories over multiple steps, highlighting its potential for scalable quantum neural modelling of complex network dynamics.

Key Words: GNN, QGNO, Kuramoto network, quantum neural.

Contents

2020 *Mathematics Subject Classification:* 81P99.
Submitted January 21, 2026. Published April 11, 2026

1. Introduction

Networked dynamical systems in the form of analysis, prediction and control are a basic problem in a broad scientific and engineering spectrum, spanning interestingly statistical physics, systems biology, control engineering, information and communication systems and the social sciences [1-4]. A wide variety of many systems of practical strategic significance in the real world, including neuronal assemblies, power transmission networks, coupled mechanical oscillators, sensor networks and interacting social agents are naturally modeled in the form of a graph, where nodes represent dynamical systems and edges provide interaction between them [2,5-8]. The collective behavior of the system is due to the interaction between the nonlinear local dynamics and the global network topology and hence the case of predictive modeling of temporal evolution is one of the key areas of concern in the study of complex networks and dynamic systems [3,4,9].

Coupled oscillator networks have a particularly niche position in the list of networked dynamical systems since they are capable of capturing such phenomena as synchronization and coordination in both natural and engineered systems [10-13]. These kinds of networks are canonical prototypes of a variety

of systems, such as circadian rhythms in the biology, frequency coordination in electrical grids, phase coherence in laser arrays, rhythmic activity in neuronal networks. One mathematical model of these systems in wide use is the Kuramoto model, which models the time-dependent dynamics of oscillator phases due to nonlinear coupling on the network structure by the sinusoidal [14-16]. The Kuramoto model has a rich repertoire of emergent behavior under a variety of conditions, including complete and incomplete synchronization, phase clustering, multistability and chimera states, coupling strength, the distribution of intrinsic frequencies, and the topology of underlying graphs [15-18].

Mathematically and computationally, the multi-step temporal dynamics of such a system have a number of inherent challenges associated with predicting them. First, it is highly nonlinear in nature and most of the time sensitive to initial conditions and therefore rapidly accumulates errors when performing long-horizon prediction [19-21]. Second, the state variables are angular quantities, defined on a compact manifold rather than a Euclidean space [22,23]. This circular geometry introduces discontinuities at the phase boundaries - π and π, rendering standard regression losses such as mean squared error mathematically inappropriate [23-25]. Third, the evolution of every node is not only determined by the state of the node itself but also by the overall structure of the entire network and is highly dimensional and globally coupled, which makes it difficult to approximate with the standard learning architectures.

Graph neural networks (GNNs) have become an influential representation learning tool in structured data defined on graphs in the past years [26-28]. Aggregation of information between adjacent nodes means that GNNs offer a principled approach to integrating network topology into machine learning models and have been found to be successful in a range of activities across data science, signal processing, and artificial intelligence. It has been suggested to extend GNNs to dynamical environments, such as recurrent and continuous-time graph-based models [29-31]. Nevertheless, classical GNN design has significant drawbacks when used with oscillatory systems and phase systems. More specifically, the rollout predictions on long time scales are usually subject to instability in time, time drift, and compounding errors, particularly in regimes of high nonlinearity or weak synchronization [32-34]. Besides, classical neural networks lack the ability to naturally admit the periodic geometry of the phase variables, which is frequently required to be corrected heuristically at the cost of physical understanding.

In this regard, quantum computing presents a radically new, and potentially revolutionary methodology of modeling complex dynamical systems. Quantum information processing is performed in exponentially large Hilbert spaces, which can represent high-dimensional states and correlations in compact form. Of special importance are parametrized quantum circuits (PQCs), parametrizable quantum neural network analogs [34-37]. It has been demonstrated that PQCs are expressive enough to approximate a broad range of nonlinear functions, and as of lately, are being considered in the larger domains of data science, learning systems and information processing [36-38].

The unique strength of PQCs is that they can take advantage of quantum superposition and entanglement where nonlocal correlations can be represented in a natural and efficient manner[38-39]. This property is particularly useful to dynamical systems which are networked, in which collective behavior at the global scale due to distributed interactions throughout the graph. Moreover, quantum measurement provides continuous expectation values, which makes PQCs the best choice to be used when solving regression and time-series prediction problems with real-valued observables.

On this basis, this contribution proposes a new model called the Quantum Graph Neural Oscillator (QGNO). The suggested method combines ideas of complex networks, neural learning systems and quantum information processing to simulate and emulate the time dependence of the phase dynamics on arbitrary graphs. Within the QGNO model, the phases of nodes are represented as quantum states, the network topology is modeled as entangling operations on the circuit and a learnable PQC learns the discrete-time evolution operator of the system dynamics.

To address the specific challenges associated with oscillatory systems, several methodological innovations are incorporated. First, a circular mean squared error (circular MSE) loss function is employed to ensure that phase differences are evaluated on modulo 2π, thereby respecting the intrinsic geometry of angular variables and avoiding artificial discontinuities. Second, decaying hybrid-teacher forcing approach is put forward to stabilize multi-step prediction in training. This is an error-reducing forecasting method that prevents errors by slowly shifting the ground-truth inputs to autoregressive predictions, eliminating error build-up while maintaining long-term stability. Third, explicit phase wrapping errors are used both

at training and evaluation, which guarantees physical consistency and does not allow the development of spurious phase jumps.

The QGNO architecture is a powerful, scalable, and topology-conscious networked oscillatory dynamics learning architecture built by incorporating these building blocks. Notably, the suggested technique is not limited to the size and structure of a particular graph and can be generalized to weighted, directed, or bigger networks.

The remainder of this paper is organized as follows. Section 2 reviews the theoretical background on coupled oscillator models and parametrized quantum circuits. Section 3 surveys related work in graph neural networks and quantum machine learning. Section 4 outlines the motivation and scope of the proposed approach. Section 5 presents the detailed methodology of the QGNO framework. Section 6 describes the experimental setup and Section 7 discusses the numerical results obtained on Kuramoto networks. Finally, Section 8 concludes the paper with insights into generalization, limitations and future research directions.

2. Theoretical Foundations

This section provides the theoretical foundations underpinning the proposed Quantum Graph Neural Oscillator (QGNO) framework. First, we review the mathematical formulation of coupled oscillator networks, with emphasis on the Kuramoto model and its graph-based generalizations. We then introduce the formal structure of parametrized quantum circuits (PQCs), which serve as the computational substrate for learning dynamical evolution in the proposed approach.

2.1. Coupled Oscillator Networks

Coupled oscillators constitute a canonical class of nonlinear dynamical systems used to study collective behaviour emerging from local interactions. Consider a network of N oscillators represented by a graph [3,4]

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}),$$

where $\mathcal{V} = \{1, 2, \ldots, N\}$ denotes the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ denotes the set of edges. The connectivity of the network is encoded by an adjacency matrix

$$A = [A_{ij}], \quad A_{ij} = \begin{cases} 1, & \text{if } (i,j) \in \mathcal{E}, \\ 0, & \text{otherwise.} \end{cases}$$

Each node i is associated with a phase variable

$$\theta_i(t) \in \mathbb{S}^1,$$

where $\mathbb{S}^1$ denotes the unit circle. The dynamics of the system evolve on the compact manifold $\left(\mathbb{S}^1\right)^N$, rather than in a Euclidean space [3].

2.2. Kuramoto Model on Graphs

The Kuramoto model provides a paradigmatic description of phase synchronization in networks of weakly coupled oscillators [3,5]. On a general graph, the continuous-time dynamics are given by

$$\frac{d\theta_i(t)}{dt} = \omega_i + K \sum_{j=1}^{N} A_{ij} \sin(\theta_j(t) - \theta_i(t)), \quad i = 1, \ldots, N, \tag{2.1}$$

where:

- $\omega_i \in \mathbb{R}$ is the intrinsic frequency of oscillator i,
- $K > 0$ is the global coupling strength,
- A_{ij} encodes the network topology.

For weighted graphs, A_{ij} may be replaced by a weight matrix $W_{ij} \in \mathbb{R}^+$, allowing heterogeneous coupling strengths. Equation (2.1) defines a nonlinear system of ordinary differential equations whose qualitative behaviour depends sensitively on K, the distribution of $\{\omega_i\}$, and the spectral properties of the graph Laplacian [6,7]

$$L = D - A,$$

where D is the degree matrix with entries $D_{ii} = \sum_j A_{ij}$.

2.3. Synchronization and Order Parameters

A central concept in the analysis of the Kuramoto model is the order parameter, defined as

$$re^{i\psi} = \frac{1}{N}\sum_{j=1}^{N} e^{i\theta_j}, \tag{2.2}$$

where:

- $r \in [0,1]$ measures the degree of phase coherence,
- ψ denotes the mean phase.

Values of $r \approx 0$ correspond to incoherent dynamics, while $r \approx 1$ indicates strong synchronization [3,5]. In networked settings, partial synchronization and cluster formation may occur, particularly for heterogeneous graphs [6].

2.4. Discrete-Time Formulation and Prediction Problem

For learning-based approaches, the continuous-time dynamics in Eq. (1) are typically discretized using a time step Δt, yielding

$$\theta_i^{(t+1)} = \theta_i^{(t)} + \Delta t\,\omega_i + K\sum_{j=1}^{N} A_{ij}\sin\Big(\theta_j^{(t)} - \theta_i^{(t)}\Big) \mod 2\pi. \tag{2.3}$$

Let

$$\boldsymbol{\theta}^{(t)} = \Big(\theta_1^{(t)}, \ldots, \theta_N^{(t)}\Big)^\top$$

denote the system state at time step t. The objective of multi-step prediction is to approximate the unknown nonlinear mapping

$$\boldsymbol{\theta}^{(t+1)} = \mathcal{F}\Big(\boldsymbol{\theta}^{(t)}, \mathcal{G}\Big), \tag{2.4}$$

where $\mathcal{F}$ is implicitly defined by the underlying physical dynamics and network structure. Learning an accurate approximation of $\mathcal{F}$ over long horizons constitute a central challenge addressed in this work.

2.5. Parametrized Quantum Circuits

Parametrized quantum circuits (PQCs) form the computational backbone of many quantum machine learning models [11-13]. A PQC consists of a sequence of unitary operations acting on an n-qubit quantum state, where certain gates depend on trainable parameters.

Let $|0\rangle^{\otimes n}$ denote the initial quantum state. A PQC implements a unitary transformation

$$|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta})|0\rangle^{\otimes n}, \tag{2.5}$$

where

$$U(\boldsymbol{\theta}) = \prod_{\ell=1}^{L} U_\ell(\boldsymbol{\theta}_\ell),$$

and each layer U_ℓ comprises parameterized single-qubit rotations and fixed entangling gates.

2.6. Quantum Encoding of Classical Data

Classical inputs, such as oscillator phases, must be embedded into quantum states. A common strategy is angle encoding, in which a scalar input x modulates a rotation gate [14,15]:

$$R_y(x) = \exp\Big(-i\frac{x}{2}\sigma_y\Big), \qquad R_z(x) = \exp\Big(-i\frac{x}{2}\sigma_z\Big), \tag{2.6}$$

where σ_y and σ_z are Pauli matrices. For a vector of phases $\boldsymbol{\theta}$, the encoding unitary takes the form

$$U_{\text{enc}}(\boldsymbol{\theta}) = \bigotimes_{i=1}^{n} R_y(\theta_i), \tag{2.7}$$

assuming one qubit per node.

2.7. Measurement and Output Mapping

After applying the PQC, observable quantities are obtained by measuring expectation values of Hermitian operators. For instance, the expectation value of the Pauli-Z operator on qubit i is

$$\langle Z_i \rangle = \langle \psi(\boldsymbol{\theta}) | Z_i | \psi(\boldsymbol{\theta}) \rangle \in [-1, 1]. \tag{2.8}$$

These expectation values are mapped back to classical outputs, such as predicted phase increments:

$$\theta_i^{(t+1)} = \theta_i^{(t)} + \alpha \langle Z_i \rangle, \tag{2.9}$$

where α is a scaling factor.

2.8. Variational Training of PQCs

The trainable parameters $\boldsymbol{\theta}$ of the PQC are optimized by minimizing a classical loss function $\mathcal{L}$. Parameter updates are computed using gradient-based methods, with gradients obtained via the parameter-shift rule [16]:

$$\frac{\partial \langle O \rangle}{\partial \theta_k} = \frac{1}{2}\left[\langle O \rangle_{\theta_k + \frac{\pi}{2}} - \langle O \rangle_{\theta_k - \frac{\pi}{2}}\right]. \tag{2.10}$$

This hybrid quantum-classical optimization loop enables efficient training of PQCs on classical hardware augmented by quantum processors or simulators.

2.9. Relevance to Networked Dynamical Systems

The expressive capacity of PQCs, combined with their ability to model nonlocal correlations through entanglement, makes them particularly suitable for approximating the nonlinear evolution operator $\mathcal{F}$ in Eq. (4). Through a combination of embracing network topology into the entangling structure of the circuit, PQCs can be innately used as quantum counterparts to graph neural networks [17,18], which underlies the theoretical framework of the QGNO used in this work.

3. Related Work

The section reviews the existing literature in the field of the current study, namely (i) graph neural networks of dynamical systems, (ii) learning oscillatory and phase-based dynamics, and (iii) quantum machine learning models, especially parametrized quantum circuits on structured data. The review identifies the successes and shortcomings of the current strategies, which is a strong incentive to the proposed Quantum Graph Neural Oscillator (QGNO) framework.

3.1. Graph Neural Networks for Networked Systems

Graph neural networks (GNNs) have emerged as a powerful paradigm for learning representations of structured data defined over graphs [19,20]. Early formulations, such as spectral graph convolutional networks, relied on the eigen decomposition of the graph Laplacian L and defined convolutional operators in the graph Fourier domain:

$$\mathbf{H}^{(l+1)} = \sigma\Big(U g_\theta(\Lambda) U^\top \mathbf{H}^{(l)}\Big), \tag{3.1}$$

where $U\Lambda U^\top$ is the eigen decomposition of L, g_θ is a learnable spectral filter, and $\sigma(\cdot)$ denotes a nonlinear activation function. While mathematically elegant, spectral methods suffer from scalability issues and poor generalization across graph topologies [21].

To address these limitations, spatial GNNs were introduced, defining message-passing operations directly in the vertex domain. A generic message-passing GNN layer takes the form

$$\mathbf{h}_i^{(l+1)} = \phi^{(l)}\left(\mathbf{h}_i^{(l)}, \bigoplus_{j\in\mathcal{N}(i)} \psi^{(l)}\Big(\mathbf{h}_i^{(l)}, \mathbf{h}_j^{(l)}, e_{ij}\Big)\right), \tag{3.2}$$

where $\mathbf{h}_i^{(l)}$ denotes the embedding of node i at layer l, $\mathcal{N}(i)$ is its neighborhood, e_{ij} encodes edge attributes, and $\phi^{(l)}, \psi^{(l)}$ are learnable functions.

GNNs have been successfully applied to static graph tasks such as node classification, link prediction, and graph-level regression. Extensions to dynamic graphs and temporal processes include spatio-temporal GNNs, which integrate recurrent neural networks or temporal convolutions to model time evolution. However, such approaches typically operate in Euclidean feature spaces and do not explicitly respect the underlying geometry of phase-valued variables.

3.2. Learning Dynamical Systems on Graphs

The problem of learning graph-based dynamical systems has attracted considerable interest, particularly in the context of physical and biological networks. Neural ordinary differential equations (Neural ODEs) and graph neural ODEs attempt to learn continuous-time dynamics [9,22] of the form

$$\frac{d\mathbf{x}(t)}{dt} = f_\theta(\mathbf{x}(t), \mathcal{G}), \tag{3.3}$$

where $\mathbf{x}(t) \in \mathbb{R}^{N\times d}$ represents node states and f_θ is parameterized by a neural network respecting graph structure.

While such methods provide a principled framework for modeling dynamics, they often struggle with stiff systems, long-term stability, and highly oscillatory behavior [23]. Moreover, when applied to angular variables $\theta_i \in \mathbb{S}^1$, naive Euclidean loss functions fail to account for periodicity, leading to discontinuities near phase boundaries.

Fourier neural operators and Koopman operator-based models are special methods for oscillatory dynamics, which aim at linearizing nonlinear dynamics in a lifted space. However, these approaches usually presuppose global linearizability or demand large latent dimensions, and thus are only applicable to limited classes of complex networked oscillators.

3.3. Phase-Based and Circular Data Learning

Learning from circular or angular data introduces unique challenges due to the topology of the state space. For phase variables, the natural distance metric is defined on the unit circle:

$$d_{\mathbb{S}^1}(\theta, \hat{\theta}) = \min_{k\in\mathbb{Z}} \Big|\theta - \hat{\theta} + 2\pi k\Big|. \tag{3.4}$$

Several works have proposed circular loss functions, such as the circular mean squared error

$$\mathcal{L}_{\text{circ}} = \frac{1}{N}\sum_{i=1}^{N}\Big[1 - \cos\Big(\theta_i - \hat{\theta}_i\Big)\Big], \tag{3.5}$$

which is invariant under phase wrapping. Despite their theoretical soundness, such losses are rarely integrated into GNN-based dynamical prediction frameworks, especially in multi-step rollout settings.

3.4. Parametrized Quantum Circuits in Machine Learning

Quantum machine learning (QML) leverages the expressive power of quantum systems to perform learning tasks. Parametrized quantum circuits (PQCs) form the core of variational quantum algorithms, implementing mappings of the form

$$f_\theta(\mathbf{x}) = \langle 0 \mid U^\dagger(\mathbf{x},\theta)\, O\, U(\mathbf{x},\theta) \mid 0 \rangle, \tag{3.6}$$

where $U(\mathbf{x},\theta)$ is a data-encoding and parameterized unitary, and O is a measurement observable.

PQCs have been shown to be universal function approximators under suitable conditions and to exhibit rich expressivity due to entanglement and nonlinearity induced by measurement [26,27]. Recent studies have explored their application to regression, classification, and time-series prediction. However, many such works focus on vector-valued data and do not exploit structured relational information.

3.5. Quantum Graph Neural Networks

Quantum extensions of GNNs have been proposed in various forms, including quantum random walks on graphs, quantum convolutional networks, and hybrid quantum-classical message passing architectures [17,28]. These models typically encode graph structure into entangling gates or Hamiltonians of the form

$$H = \sum_{(i,j)\in\mathcal{E}} J_{ij}\, \sigma_i^z \sigma_j^z + \sum_i h_i \sigma_i^x, \tag{3.7}$$

where $\sigma_i^{x,z}$ are Pauli operators acting on qubit i.

While such Hamiltonian-based models capture graph connectivity, most existing quantum graph models are designed for static inference tasks and do not address time-dependent dynamical prediction. Furthermore, the majority of QGNN studies evaluate performance on small synthetic datasets, with limited consideration of stability under long-horizon rollouts.

3.6. Limitations of Existing Approaches

Despite substantial progress, several gaps remain in the literature [10,23,29]:

- **Topology–Dynamics Mismatch:** Classical GNNs are not inherently designed to handle dynamics evolving on non-Euclidean manifolds such as $\mathbb{S}^1$.
- **Long-Horizon Instability:** Multi-step prediction of oscillatory systems suffers from error accumulation and phase drift.
- **Underutilization of Quantum Expressivity:** Existing QML models rarely exploit quantum correlations for structured dynamical systems.
- **Lack of Generalization Across Graphs:** Many approaches are tailored to fixed graph sizes and topologies.

3.7. Positioning of the Present Work

The QGNO framework proposed overcomes these drawbacks by incorporating graph structure, circular geometry, and quantum variational learning into a single framework. The method is based on the joint application of circular loss functions, decaying teacher forcing, and entanglement-aware PQCs, and improves upon existing classical and quantum models of networked oscillator dynamics.

4. Motivation and Scope of the Proposed Approach

The growing widespread use of complex networked systems in science and engineering has focused on the necessity of structurally aware and dynamically expressive modeling frameworks. Although notable advances have been achieved in the field of graph-based learning and nonlinear dynamical systems theory, the correct prediction of long-horizon dynamics of coupled, oscillatory dynamical systems on arbitrary graphs has remained an open problem. In this section, the reasons that prompted the formulation of the proposed Quantum Graph Neural Oscillator (QGNO) framework are described, and the area of its applicability is outlined.

4.1. Motivation: Challenges in Learning Networked Oscillatory Dynamics

Networks of coupled oscillators exhibit several characteristics that complicate their modeling using conventional data-driven methods. First, the dynamics are inherently nonlinear, governed by trigonometric interactions that cannot be well approximated by linear or locally linear models. Second, the state variables are phase-valued and evolve on the compact manifold $\mathbb{S}^1$ rather than in Euclidean space. Consequently, standard regression losses of the form

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{N}\sum_{i=1}^{N}\left(\theta_i - \hat{\theta}_i\right)^2 \tag{4.1}$$

are fundamentally mismatched to the geometry of the problem, as they fail to respect periodicity and introduce artificial discontinuities near $\pm\pi$.

A further complication arises in multi-step prediction, where errors at each step compound over time. Given a learned transition operator $\mathcal{F}_\theta$,

$$\hat{\boldsymbol{\theta}}(t+1) = \mathcal{F}_\theta(\boldsymbol{\theta}(t))\,, \tag{4.2}$$

small phase errors at early time steps can accumulate, leading to phase drift, loss of synchronization structure, and eventual divergence from physically plausible trajectories. This phenomenon is especially pronounced in oscillatory systems due to their sensitivity to relative phase relationships rather than absolute magnitudes.

4.2. Limitations of Classical Graph Learning Paradigms

Classical graph neural networks are primarily designed for relational inference or short-term prediction in Euclidean feature spaces. Although recurrent or attention-based temporal extensions have been proposed, they typically rely on discrete-time approximations and large parameter counts to approximate continuous nonlinear dynamics. When applied to oscillatory systems, such approaches suffer from several limitations:

- **Geometric Incompatibility:** Phase variables evolve on a circular manifold, whereas most GNN architectures assume embeddings in $\mathbb{R}^d$.
- **Expressivity Constraints:** Capturing higher-order correlations among interacting oscillators often requires deep networks with extensive message passing, increasing computational cost.
- **Stability Problems:** Long-horizon rollouts are prone to compounding approximation errors, leading to instability.

These issues motivate the exploration of alternative computational paradigms capable of compactly representing nonlinear correlations while respecting the geometry of the state space.

4.3. Quantum Motivation: Leveraging Hilbert Space Expressivity

Quantum computing offers a fundamentally different representational framework based on complex Hilbert spaces. An n-qubit quantum system evolves in a 2^n-dimensional state space, enabling the compact

encoding of high-order correlations through entanglement. From a learning perspective, parametrized quantum circuits (PQCs) implement hypothesis classes of the form

$$f_\theta : \mathbb{R}^d \to \mathbb{R}^m, \qquad f_\theta(\mathbf{x}) = \langle \psi(\mathbf{x}) \mid U^\dagger(\theta)\, O\, U(\theta) \mid \psi(\mathbf{x}) \rangle, \tag{4.3}$$

where $|\psi(\mathbf{x})\rangle$ denotes a data-encoded quantum state and $U(\theta)$ is a trainable unitary transformation.

PQCs are particularly attractive for networked dynamics for two reasons. First, quantum entanglement naturally mirrors graph connectivity, allowing inter-node interactions to be encoded directly via entangling gates. Second, the nonlinear measurement process provides expressive nonlinear mappings without requiring deep circuit architectures. These properties suggest that PQCs can serve as compact and stable alternatives to deep classical networks for learning complex graph-based dynamics.

4.4. Motivation for Hybrid Quantum–Classical Training

Despite their expressivity, near-term quantum devices are constrained by noise, limited qubit counts, and shallow circuit depths. As a result, fully quantum end-to-end learning remains impractical for most real-world applications. The proposed QGNO framework therefore adopts a hybrid quantum–classical approach, in which quantum circuits act as learnable function approximators while classical optimizers perform parameter updates.

Formally, the learning problem is posed as

$$\min_\theta \; \mathbb{E}_t \left[\mathcal{L}_{\text{circ}}\big(\boldsymbol{\theta}(t+1), \mathcal{F}_\theta(\boldsymbol{\theta}(t))\big) \right], \tag{4.4}$$

where gradients $\nabla_\theta \mathcal{L}$ are computed via parameter-shift rules and optimized using classical gradient-based methods. This hybrid strategy enables scalability while remaining compatible with both simulators and emerging quantum hardware.

4.5. Scope of the Proposed Framework

The proposed QGNO framework is deliberately broad and applies to a wide class of networked dynamical systems with the following properties:

- **Graph-Structured Interactions:** Systems defined on arbitrary, possibly weighted and irregular graphs.
- **Phase or Angle-Valued States:** Dynamics evolving on compact manifolds such as $\mathbb{S}^1$.
- **Nonlinear Coupling Mechanisms:** Interactions that cannot be accurately captured by linear operators.
- **Multi-Step Prediction Requirements:** Applications demanding stable long-horizon forecasts.

Although the present work focuses on the Kuramoto model as a canonical testbed, the methodology readily generalizes to other oscillator models, including Sakaguchi–Kuramoto systems, power grid swing equations, and phase-reduced neuron models.

4.6. Research Objectives and Contributions

Guided by the above motivations, the objectives of this work are:

- To develop a quantum-enhanced graph neural architecture for oscillatory dynamics.
- To enforce geometric consistency and phase-wrapping constraints via circular loss functions.
- To stabilize long-horizon prediction using a decaying hybrid teacher-forcing strategy.
- To demonstrate the feasibility of the approach on graph-based oscillator networks.

Collectively, these contributions position the proposed QGNO framework as a mathematically principled and computationally innovative approach to learning networked dynamical systems.

5. Methodology: Quantum Graph Neural Oscillator (QGNO) Framework

This section presents the detailed methodology underlying the proposed Quantum Graph Neural Oscillator (QGNO) framework. The objective of QGNO is to learn a stable, graph-aware, and geometry-consistent approximation of the discrete-time flow map governing coupled oscillator dynamics. The framework integrates classical dynamical system modeling, quantum state encoding, parametrized quantum circuits, and hybrid quantum–classical optimization.

5.1. Problem Formulation

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ denote an undirected graph with node set $\mathcal{V} = \{1, \dots, N\}$, edge set $\mathcal{E}$, and adjacency matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$. Each node $i \in \mathcal{V}$ represents an oscillator characterized by a phase variable

$$\theta_i(t) \in \mathbb{S}^1 \cong (-\pi, \pi].$$

The continuous-time dynamics are governed by

$$\dot{\theta}_i(t) = \omega_i + \sum_{j=1}^{N} A_{ij}\, g(\theta_j(t) - \theta_i(t))\,, \tag{5.1}$$

where ω_i denotes the natural frequency of oscillator i, and $g(\cdot)$ is a nonlinear coupling function (e.g., $g(x) = \sin x$ in the Kuramoto model).

Given discretized observations $\{\boldsymbol{\theta}(t_k)\}_{k=0}^{T}$, the learning task is to approximate the discrete-time evolution operator

$$\boldsymbol{\theta}(t_{k+1}) = \mathcal{F}(\boldsymbol{\theta}(t_k))\,, \tag{5.2}$$

using a parametrized quantum model $\mathcal{F}_\Theta$ such that long-horizon predictions remain stable and physically consistent.

5.2. Quantum Encoding of Node States

Each node phase θ_i is encoded into a local quantum state using a trigonometric amplitude representation:

$$\mathbf{x}_i = \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \end{bmatrix}. \tag{5.3}$$

For a node allocated q qubits, the local Hilbert space dimension is 2^q. The vector $\mathbf{x}_i$ is zero-padded and normalized as

$$\widetilde{\mathbf{x}}_i = \frac{1}{\|\mathbf{z}_i\|} \begin{bmatrix} \cos\theta_i \\ \sin\theta_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} \in \mathbb{R}^{2^q}. \tag{5.4}$$

The global quantum input state is constructed via a tensor product over all nodes:

$$|\Psi_{\text{in}}\rangle = \bigotimes_{i=1}^{N} |\widetilde{\mathbf{x}}_i\rangle \in \mathbb{C}^{2^{Nq}}. \tag{5.5}$$

This encoding preserves the circular geometry of the phase variables and ensures compatibility with amplitude embedding.

5.3. Parameterized Quantum Circuit Architecture

The QGNO model employs a parameterized quantum circuit (PQC) composed of alternating local rotation layers and graph-informed entanglement layers.

5.3.1. Local Rotation Layers. Each qubit undergoes a sequence of trainable rotations:

$$U_{\text{rot}}(\Theta^{(\ell)}) = \prod_{w=1}^{Nq} R_Z\left(\theta_{w,3}^{(\ell)}\right) R_Y\left(\theta_{w,2}^{(\ell)}\right) R_X\left(\theta_{w,1}^{(\ell)}\right), \tag{5.6}$$

where ℓ indexes the circuit depth and $\Theta^{(\ell)} \in \mathbb{R}^{Nq\times 3}$ are trainable parameters.

5.3.2. Graph-Structured Entanglement. Inter-node interactions are encoded using controlled entangling operations aligned with the adjacency matrix $\mathbf{A}$. For each edge $(i, j) \in \mathcal{E}$, the following entangling block is applied:

$$U_{ij} = \text{CNOT}_{i\to j}\ R_Z(\phi)\ \text{CNOT}_{i\to j}. \tag{5.7}$$

This design ensures that information propagates only along graph edges, preserving the network topology.

5.3.3. Full Circuit Evolution. The full unitary transformation implemented by the PQC is

$$U(\Theta) = \prod_{\ell=1}^{L} U_{\text{ent}}\, U_{\text{rot}}\left(\Theta^{(\ell)}\right), \tag{5.8}$$

where L denotes the circuit depth.

5.4. Quantum Measurement and Phase Readout

For each node i, the phase is recovered by measuring the expectation values of Pauli operators:

$$x_i = \langle \sigma_x^{(i)} \rangle, \qquad y_i = \langle \sigma_y^{(i)} \rangle. \tag{5.9}$$

The predicted phase is obtained via

$$\hat{\theta}_i = \text{atan2}(y_i, x_i). \tag{5.10}$$

5.5. Circular Loss Function

To respect the topology of the phase space $\mathbb{S}^1$, a circular mean squared error (cMSE) loss is employed:

$$\mathcal{L}_{\text{circ}} = \frac{1}{N} \sum_{i=1}^{N} \left| \angle e^{j(\theta_i - \hat{\theta}_i)} \right|^2. \tag{5.11}$$

This loss penalizes angular discrepancies modulo 2π.

5.6. Multi-Step Training with Decaying Hybrid Teacher Forcing

At training step k, the input is defined as

$$\mathbf{x}_k = \begin{cases} \boldsymbol{\theta}_{\text{true}}(k), & \text{with probability } p_k, \\ \boldsymbol{\theta}_{\text{pred}}(k), & \text{with probability } 1 - p_k, \end{cases} \tag{5.12}$$

where

$$p_k = p_{\text{start}} - \frac{k}{K} \left(p_{\text{start}} - p_{\text{end}}\right). \tag{5.13}$$

5.7. Hybrid Optimization

Model parameters are optimized using a hybrid quantum–classical loop:

$$\Theta_{t+1} = \Theta_t - \eta \nabla_\Theta \mathcal{L}_{\text{circ}}, \tag{5.14}$$

where gradients are computed via the parameter-shift rule and optimized using Adam.

5.8. Phase Wrapping Enforcement

To ensure physical consistency, all predicted phases are wrapped into the principal interval:

$$\theta \leftarrow \big((\theta + \pi) \bmod 2\pi\big) - \pi. \tag{5.15}$$

5.9. Computational Complexity

For N nodes, q qubits per node, and circuit depth L, the parameter count scales as

$$\mathcal{O}(LNq), \tag{5.16}$$

while the Hilbert space dimension scales as $\mathcal{O}(2^{Nq})$. This exponential growth underscores the importance of shallow circuits and motivates future hardware-aware optimizations.

6. Experimental Setup

This section describes the experimental protocol used to evaluate the proposed Quantum Graph Neural Oscillator (QGNO) framework. The experiments are designed with three primary objectives: (i) to assess whether QGNO can learn graph-structured nonlinear oscillator dynamics, (ii) to evaluate the stability of long-horizon phase predictions, and (iii) to verify the compatibility of QGNO with near-term quantum simulation environments.

6.1. Dynamical System and Dataset Generation

To provide a controlled and interpretable testbed, experiments are conducted using the classical Kuramoto model defined on a finite graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with N nodes. The continuous-time dynamics are given by

$$\dot{\theta}_i(t) = \omega_i + \frac{K}{\max(1, d_i)} \sum_{j=1}^{N} A_{ij} \sin(\theta_j(t) - \theta_i(t)), \tag{6.1}$$

where $\theta_i(t)$ denotes the phase of oscillator i, $\omega_i \sim \mathcal{N}(0, \sigma^2)$ is the intrinsic frequency, K is the global coupling strength, $d_i = \sum_j A_{ij}$ is the degree of node i, and A_{ij} represents the adjacency matrix.

The system is numerically integrated using a fixed-step solver over a time interval $[0, T]$ with step size Δt. The resulting phase trajectories

$$\mathcal{D} = \{\boldsymbol{\theta}(t_k)\}_{k=0}^{K} \tag{6.2}$$

serve as supervised learning data for training and evaluation.

6.2. Graph Topology Construction

For the baseline experiments, random connected graphs are generated using the Erdős–Rényi model:

$$\mathbb{P}((i, j) \in \mathcal{E}) = p, \tag{6.3}$$

with $p = 0.6$. Graph connectivity is enforced to guarantee information flow across all nodes. The resulting adjacency matrix $\mathbf{A}$ is a binary symmetric matrix corresponding to an undirected interaction network. This setup enables the analysis of QGNO on irregular and nontrivial network structures representative of real-world systems.

6.3. Quantum Simulation Environment

All quantum computations are performed using the `PennyLane` framework, which supports hybrid quantum–classical optimization and automatic differentiation. The quantum circuits are executed on a state-vector simulator, providing exact expectation values and noiseless quantum evolution.

The total number of qubits is given by

$$n_{\text{qubits}} = N \times q, \tag{6.4}$$

where q denotes the number of qubits allocated per node.

6.4. Model Architecture and Hyperparameters

The QGNO architecture is parameterized as follows:

- Number of nodes: $N = 3$
- Number of edges: $|\mathcal{E}| = 2$
- Qubits per node: $q = 2$
- Total qubits: $n_{\text{qubits}} = 6$
- Circuit depth: $L = 4$

The total number of trainable parameters is

$$\#\Theta = 3 \times L \times N \times q = 72. \tag{6.5}$$

Trainable parameters are initialized using a zero-mean Gaussian distribution with small variance to ensure stable behavior during early training.

6.5. Training Protocol

The training dataset consists of consecutive phase pairs

$$(\boldsymbol{\theta}(t_k), \boldsymbol{\theta}(t_{k+1}))\,, \tag{6.6}$$

excluding the final rollout horizon.

The optimization objective is to minimize the circular mean squared error defined in Eq. (32). Parameter updates are performed using the Adam optimizer with learning rate $\eta = 0.05$. Training is conducted for 100 epochs. At each epoch e, the teacher forcing probability is updated according to

$$p_{\text{TF}}(e) = p_{\text{start}} - \frac{e}{E}\,(p_{\text{start}} - p_{\text{end}})\,, \tag{6.7}$$

where $p_{\text{start}} = 0.9$, $p_{\text{end}} = 0.1$, and $E = 100$.

6.6. Multi-Step Rollout Evaluation

To evaluate long-horizon predictive stability, the trained QGNO model is rolled out for $R = 20$ future time steps. At each rollout step, the input is selected using the hybrid teacher forcing rule described in Section 5.6. Predicted phases are wrapped into the principal interval $(-\pi, \pi]$ after each step to prevent phase drift.

6.7. Evaluation Metrics

Model performance is assessed using the following metrics:

Circular Mean Squared Error (cMSE).

$$\text{cMSE} = \frac{1}{NR}\sum_{k=1}^{R}\sum_{i=1}^{N}\left|\angle\left(e^{j(\hat{\theta}_i(k)-\theta_i(k))}\right)\right|^2. \tag{6.8}$$

Qualitative Trajectory Consistency. Visual comparison between predicted and ground-truth phase trajectories over the rollout horizon.

Final-Step Phase Error.

$$\varepsilon_{\text{final}} = \frac{1}{N}\sum_{i=1}^{N}\left|\angle\left(e^{j(\hat{\theta}_i(R)-\theta_i(R))}\right)\right|. \tag{6.9}$$

6.8. Reproducibility and Implementation Details

All experiments are conducted using Python 3.13. Graph generation, parameter initialization, and stochastic teacher forcing are controlled via fixed random seeds to ensure reproducibility. Numerical integration of the Kuramoto dynamics is performed using the `odeint` solver from the `SciPy` library. The implementation is fully compatible with quantum simulators and can be directly extended to quantum hardware backends supported by PennyLane.

7. Results and Discussion

This section reports and analyzes the numerical results obtained from applying the proposed Quantum Graph Neural Oscillator (QGNO) framework to Kuramoto oscillator networks. The discussion focuses on training behavior, predictive accuracy, long-horizon rollout stability, and implications for modeling graph-structured dynamical systems.

7.1. Training Convergence and Optimization Behavior

The evolution of the training loss across epochs demonstrates stable convergence of the QGNO model under the proposed circular loss and decaying hybrid teacher forcing strategy. Starting from an initial circular mean squared error of approximately 7.9×10^{-2}, the loss consistently decreases during training, reaching values on the order of 3.5×10^{-2} by epoch 100:

$$\mathcal{L}_{\text{circ}}^{(1)} \approx 0.0793, \qquad \mathcal{L}_{\text{circ}}^{(100)} \approx 0.0354. \tag{7.1}$$

This monotonic reduction, despite the gradual decrease in teacher forcing probability from 0.90 to 0.10, indicates that the quantum model successfully transitions from supervised guidance to autonomous prediction. Notably, no divergence or oscillatory behavior is observed in the loss trajectory, highlighting the stabilizing effect of the circular loss formulation and phase-wrapping constraint.

7.2. Effect of Decaying Hybrid Teacher Forcing

The decaying hybrid teacher forcing mechanism plays a crucial role in stabilizing the training process. High teacher forcing probability during early epochs exposes the model to physically consistent trajectories and prevents premature error accumulation. As training progresses, the gradual reduction in teacher forcing encourages the QGNO to increasingly rely on its own predictions, effectively training the quantum circuit to correct internal errors.

Formally, the teacher forcing schedule

$$p_{\text{TF}}(e) = 0.9 - 0.8\frac{e}{100} \tag{7.2}$$

ensures a smooth transition from teacher-driven learning to free-running prediction. The sustained low loss at later epochs demonstrates that QGNO learns the underlying dynamical flow map rather than merely memorizing one-step transitions.

7.3. Multi-Step Prediction Accuracy

After training, the QGNO model is evaluated under a 20-step rollout setting. The final-step predictions are compared against the ground-truth phases obtained from numerical integration of the Kuramoto model:

$$\boldsymbol{\theta}_{\text{true}}(R) = [-1.288,\ -0.772,\ -0.894]\,, \tag{7.3}$$

$$\boldsymbol{\theta}_{\text{pred}}(R) = [-1.085,\ -0.576,\ -0.698]\,. \tag{7.4}$$

The resulting angular discrepancies are modest and remain well within the principal interval $(-\pi, \pi]$. The average final-step circular error is computed as

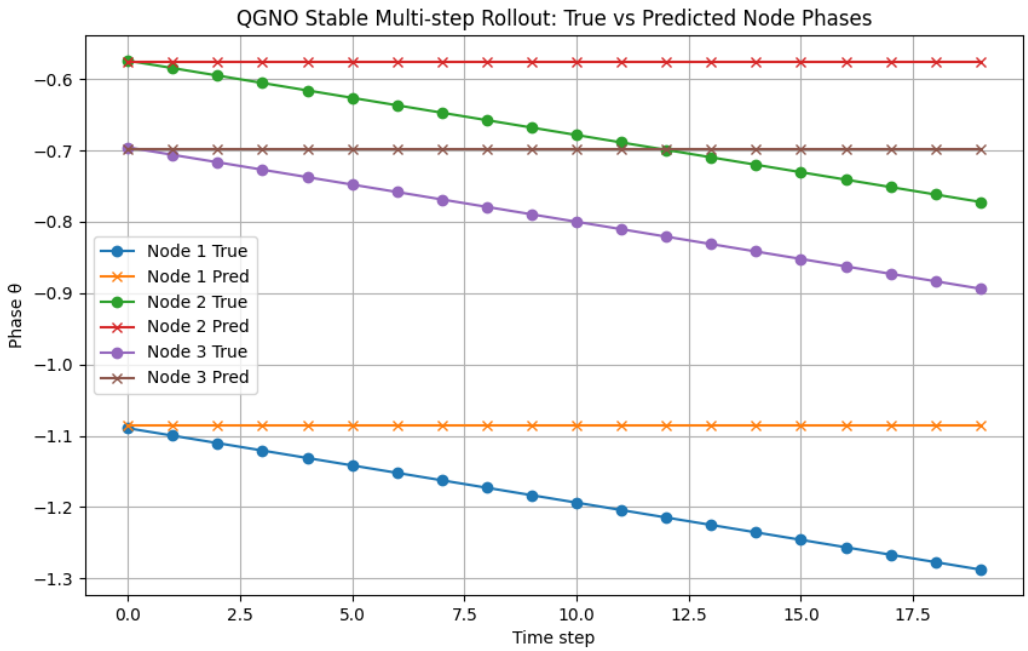
$$\varepsilon_{\text{final}} \approx \frac{1}{3}\sum_{i=1}^{3}\left|\angle\left(e^{j(\hat{\theta}_i - \theta_i)}\right)\right| \approx 0.19. \tag{7.5}$$

This level of accuracy is notable given the nonlinear coupling and multi-step nature of the task, particularly in light of the shallow circuit depth and limited number of qubits employed.

7.4. Trajectory-Level Consistency

Beyond pointwise prediction errors, qualitative consistency between predicted and ground-truth phase trajectories provides a strong indication of learned dynamical structure. The predicted trajectories closely follow the temporal evolution of each oscillator, preserve relative phase relationships, and avoid unphysical phase jumps.

From a dynamical systems perspective, this behavior indicates that QGNO approximates the global flow operator governing the network rather than merely fitting local one-step transitions. The preservation of smooth phase evolution suggests that the model captures essential properties of the underlying vector field on the phase manifold.



7.5. Role of Circular Loss and Phase Wrapping

The effectiveness of the circular mean squared error loss is particularly evident near the $\pm\pi$ boundaries. Standard Euclidean losses penalize small angular deviations near these boundaries excessively, leading to artificial discontinuities and unstable gradients. In contrast, the circular loss

$$\mathcal{L}_{\text{circ}} = \frac{1}{N}\sum_{i=1}^{N}\left|\angle\left(e^{j(\theta_i-\hat{\theta}_i)}\right)\right|^2 \tag{7.6}$$

ensures smooth gradients and stable optimization across the entire phase domain.

When combined with explicit phase wrapping after each prediction step, this formulation enforces geometric consistency and prevents long-term phase drift during multi-step rollouts.

7.6. Implications for Graph-Structured Learning

The experimental findings demonstrate that QGNO effectively incorporates graph topology into the learning process. Despite the small network size, the model preserves inter-node dependencies encoded in the adjacency matrix and reproduces collective behaviors induced by coupling.

The graph-structured entanglement layers play a crucial role in facilitating information flow among nodes. In this sense, they can be interpreted as a quantum analogue of message passing, enabling the circuit to encode relational dependencies in a compact and expressive manner.

7.7. Limitations and Observed Constraints

While the results are encouraging, several limitations are observed. The exponential growth of the Hilbert space restricts experiments to small graph sizes under current simulation capabilities. Moreover, all experiments are conducted in a noiseless simulation environment, and performance under realistic quantum noise remains an open question.

These constraints are inherent to current quantum hardware and simulators and naturally motivate future research directions, including noise-aware training, circuit compression, and scalable graph partitioning strategies.

8. Conclusion

This paper introduced the *Quantum Graph Neural Oscillator* (QGNO) framework, a hybrid quantum–classical learning paradigm designed for stable modeling and prediction of oscillatory dynamics on graphs. By integrating graph-structured parametrized quantum circuits with geometry-aware loss functions and stabilized multi-step training strategies, the proposed approach addresses several fundamental challenges associated with learning nonlinear dynamical systems evolving on complex networks.

8.1. Summary of Contributions

The primary contributions of this work can be summarized as follows:

- **Quantum-Enhanced Graph Representation:** The QGNO framework embeds adjacency-aligned entanglement directly into the quantum circuit architecture, enabling the principled and physically interpretable modeling of inter-node dependencies. This design introduces graph topology as an inductive bias at the quantum level.
- **Geometry-Consistent Learning of Phase Dynamics:** Oscillator phases are encoded using trigonometric amplitude representations, and learning is guided by a circular mean squared error loss. This combination respects the intrinsic geometry of the phase manifold $\mathbb{S}^1$, avoiding artificial discontinuities and gradient instabilities common to Euclidean loss formulations.
- **Stabilized Multi-Step Prediction:** A decaying hybrid teacher forcing strategy is employed to transition the model from supervised one-step learning to autonomous long-horizon rollout. This mechanism significantly reduces error accumulation and improves predictive stability over multiple time steps.
- **Empirical Validation on Kuramoto Networks:** The proposed framework is validated on graph-based Kuramoto oscillator systems, demonstrating accurate multi-step phase prediction, preservation of synchronization structure, and robustness despite shallow circuit depth and limited qubit resources.

8.2. Implications for Complex Networked Systems

From a dynamical systems perspective, the results indicate that QGNO effectively learns an approximation of the discrete-time flow operator governing coupled oscillator dynamics. Unlike conventional graph learning models that primarily focus on local message passing or short-term inference, QGNO captures global phase relationships and collective behaviors, including synchronization phenomena.

From a computational standpoint, the framework highlights the potential of parametrized quantum circuits as compact function approximators for nonlinear, high-dimensional graph mappings. The use of entanglement as a structural inductive bias suggests a novel paradigm for encoding relational information within non-classical learning architectures.

8.3. Generalization to Arbitrary Graphs and Dynamical Systems

Although the current study focuses on small-scale Kuramoto networks, the QGNO framework is inherently graph-agnostic. The adjacency matrix serves as the sole requirement for constructing graph-aligned entanglement layers, allowing straightforward extension to:

- larger networks with heterogeneous or non-uniform connectivity,
- directed and weighted graphs,
- time-varying network topologies.

Moreover, QGNO is not restricted to the Kuramoto model. The encoding strategy and circular loss formulation naturally extend to other classes of dynamical systems, including:

- Sakaguchi–Kuramoto and phase-lag oscillator models,
- power grid swing equations and coupled phase–frequency systems,
- neuronal and biological oscillator models under phase reduction.

In these contexts, QGNO may be viewed as a quantum analogue of a nonlinear graph operator learner capable of approximating a broad class of dynamical flows.

8.4. Future Research Directions

Several promising avenues for future research emerge from this work:

- **Scalability:** Developing strategies for scaling QGNO to larger graphs through circuit compression, graph partitioning, and hardware-aware design.
- **Hardware Deployment:** Evaluating performance on noisy intermediate-scale quantum (NISQ) devices and incorporating noise-aware training techniques.
- **Theoretical Expressivity Analysis:** Establishing formal guarantees on the representational capacity of graph-structured parametrized quantum circuits.
- **Hybrid Quantum–Classical Architectures:** Integrating QGNO with classical graph neural networks in hierarchical or multi-resolution frameworks.
- **Control and Inference Tasks:** Extending the framework to parameter estimation, control, and inverse problems in networked dynamical systems.
- **Beyond Phase Dynamics:** Generalizing the approach to amplitude–phase coupled systems and higher-dimensional state manifolds beyond $\mathbb{S}^1$.

In conclusion, the proposed Quantum Graph Neural Oscillator framework represents a significant step toward quantum-enhanced modeling of networked dynamical systems. By unifying graph structure, quantum expressivity, and geometric consistency within a single learning paradigm, this work demonstrates the feasibility and promise of quantum approaches for complex dynamical prediction tasks. As quantum hardware and algorithms continue to mature, QGNO-style frameworks may play a central role in the data-driven analysis, prediction, and control of complex systems across physics, engineering, and information sciences.

Acknowledgment

The authors thank Guru Gobind Singh Indraprastha University for providing the necessary computational resources.

Funding

This research received no external financial support.

Author Contributions

Rashmi Bhardwaj: Writing – review and editing, validation, supervision, methodology, formal analysis. Debabrata Datta: Writing – review and editing, validation, supervision, methodology, formal analysis. Surbhi Sharma: Writing – original draft, review and editing, visualization, methodology.

Conflict of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

1. Strogatz, S. H. *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, 2nd edn. (CRC Press, 2015).
2. Kuramoto, Y. *Chemical Oscillations, Waves, and Turbulence* (Springer, 1984).
3. Acebrón, J. A., Bonilla, L. L., Pérez Vicente, C. J., Ritort, F. & Spigler, R. The Kuramoto model: A simple paradigm for synchronization phenomena. *Rev. Mod. Phys.* **77**, 137–185 (2005).
4. Boccaletti, S., Latora, V., Moreno, Y., Chavez, M. & Hwang, D.-U. Complex networks: Structure and dynamics. *Phys. Rep.* **424**, 175–308 (2006).
5. Arenas, A., Díaz-Guilera, A., Kurths, J., Moreno, Y. & Zhou, C. Synchronization in complex networks. *Phys. Rep.* **469**, 93–153 (2008).
6. Newman, M. *Networks: An Introduction* (Oxford Univ. Press, 2010).
7. Nishikawa, T. & Motter, A. E. Comparative analysis of existing models for power-grid synchronization. *New J. Phys.* **17**, 015012 (2015).
8. Chung, F. & Lu, L. *Complex Graphs and Networks* (American Mathematical Society, 2006).
9. Kipf, T. N. & Welling, M. Semi-supervised classification with graph convolutional networks. In *Proc. Int. Conf. Learn. Represent.* (2017).
10. Hamilton, W. L., Ying, R. & Leskovec, J. Inductive representation learning on large graphs. In *Adv. Neural Inf. Process. Syst.* **30** (2017).
11. Veličković, P. et al. Graph attention networks. In *Proc. Int. Conf. Learn. Represent.* (2018).
12. Wu, Z. et al. A comprehensive survey on graph neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **32**, 4–24 (2021).
13. Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O. & Dahl, G. E. Neural message passing for quantum chemistry. In *Proc. Int. Conf. Mach. Learn.* **70**, 1263–1272 (2017).
14. Bronstein, M. M. et al. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. arXiv:2104.13478 (2021).
15. Battaglia, P. W. et al. Relational inductive biases, deep learning, and graph networks. arXiv:1806.01261 (2018).
16. Sanchez-Gonzalez, A. et al. Graph networks as learnable physics engines for inference and control. In *Proc. Int. Conf. Mach. Learn.* **80**, 4470–4479 (2018).
17. Greydanus, S., Dzamba, M. & Yosinski, J. Hamiltonian neural networks. In *Adv. Neural Inf. Process. Syst.* **32** (2019).
18. Nielsen, M. A. & Chuang, I. L. *Quantum Computation and Quantum Information* (Cambridge Univ. Press, 2010).
19. Preskill, J. Quantum computing in the NISQ era and beyond. *Quantum* **2**, 79 (2018).
20. Schuld, M. & Petruccione, F. *Supervised Learning with Quantum Computers* (Springer, 2018).
21. Farhi, E., Goldstone, J. & Gutmann, S. A quantum approximate optimization algorithm. arXiv:1411.4028 (2014).
22. Havlíček, V. et al. Supervised learning with quantum-enhanced feature spaces. *Nature* **567**, 209–212 (2019).
23. Benedetti, M., Lloyd, E., Sack, S. & Fiorentini, M. Parameterized quantum circuits as machine learning models. *Quantum Sci. Technol.* **4**, 043001 (2019).
24. Abbas, A. et al. The power of quantum neural networks. *Nat. Comput. Sci.* **1**, 403–409 (2021).
25. Tacchino, F. et al. Quantum implementations of graph neural networks. *Phys. Rev. A* **102**, 062414 (2020).
26. Wang, Y., Li, Y. & Peng, X. Variational quantum graph neural networks. arXiv:2106.06119 (2021).
27. Biamonte, J. et al. Quantum machine learning. *Nature* **549**, 195–202 (2017).
28. Cerezo, M. et al. Variational quantum algorithms. *Nat. Rev. Phys.* **3**, 625–644 (2021).
29. Skolik, A. et al. Quantum machine learning with graph-structured data. *Quantum* **6**, 799 (2022).
30. Mardia, K. V. & Jupp, P. E. *Directional Statistics* (Wiley, 2000).
31. Lathauwer, J.-P. Manifold-based learning for circular and periodic data. *Signal Process.* **160**, 113–123 (2019).
32. Bishop, C. M. *Pattern Recognition and Machine Learning* (Springer, 2006).
33. Kingma, D. P. & Ba, J. Adam: A method for stochastic optimization. In *Proc. Int. Conf. Learn. Represent.* (2015).

34. Pascanu, R., Mikolov, T. & Bengio, Y. On the difficulty of training recurrent neural networks. In *Proc. Int. Conf. Mach. Learn.* **28**, 1310–1318 (2013).

35. Bengio, S. et al. Scheduled sampling for sequence prediction with recurrent neural networks. In *Adv. Neural Inf. Process. Syst.* **28** (2015).

36. Gal, Y. & Ghahramani, Z. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proc. Int. Conf. Mach. Learn.* **48**, 1050–1059 (2016).

37. Dörfler, F. & Bullo, F. Synchronization in complex oscillator networks. *Automatica* **50**, 1539–1564 (2014).

38. Kundur, P. *Power System Stability and Control* (McGraw–Hill, 1994).

39. Izhikevich, E. M. *Dynamical Systems in Neuroscience: The Geometry of Excitability and Bursting* (MIT Press, 2007).

Surbhi Sharma and Rashmi Bhardwaj,
University School of Basic and Applied Sciences (USBAS),
Guru Gobind Singh Indraprastha University, Delhi, India.

and

Debabrata Datta,
Heritage Institute of Technology, Kolkata, India.
E-mail address: rashmib@ipu.ac.in