Quantum Positional Encodings for Graph Neural Networks

Slimane Thabet [1 2] **Mehdi Djellabi** [1] **Igor O. Sokolov** [1] **Sachin Kasture** [1] **Louis-Paul Henry** [1] **Loïc Henriet** [1]

Abstract

In this work, we propose novel families of positional encodings tailored to graph neural networks obtained with quantum computers. These encodings leverage the long-range correlations inherent in quantum systems that arise from mapping the topology of a graph onto interactions between qubits in a quantum computer. Our inspiration stems from the recent advancements in quantum processing units, which offer computational capabilities beyond the reach of classical hardware. We prove that some of these quantum features are theoretically more expressive for certain graphs than the commonly used relative random walk probabilities. Empirically, we show that the performance of state-of-the-art models can be improved on standard benchmarks and large-scale datasets by computing tractable versions of quantum features. Our findings highlight the potential of leveraging quantum computing capabilities to enhance the performance of transformers in handling graph data.

1. Introduction

Graph machine learning (GML) is an expanding field of research with applications in chemistry (Gilmer et al., 2017), biology (Zitnik et al., 2018), drug design (Konaklieva, 2014), social networks (Scott, 2011), computer vision (Harchaoui & Bach, 2007) and science (Sanchez-Gonzalez et al., 2020; Xu et al., 2018). In the past few years, significant effort has been put into the design of Graph Neural Networks (GNNs) (Hamilton). The objective is to learn suitable representations that enable efficient solutions to the original problem.

Message Passing Neural Networks (MPNN) (Kipf & Welling, 2016; Hamilton et al., 2018; Veličković et al., 2018; Gilmer et al., 2017) is the first and most common approach to build GNNs. This method exhibits several recognized limitations (Zhu et al., 2020; Chen et al., 2020; Topping et al., 2021), that the research community is actively exploring in order to find solutions. The key idea is to expand aggregation beyond neighbouring nodes by incorporating information related to the entire graph or a more extensive portion of it. Graph Transformers were created according to these requirements, showing success on standard benchmarks (Ying et al., 2021; Rampášek et al., 2022). Among the myriad of proposed architectures, the Graph Inductive Bias Transformer (GRIT) (Ma et al., 2023) stands out for its impressive generalization capacity. Similarly to their counterparts in natural language processing, graph transformers compute positional encodings (PE) that are concatenated to node features. The most commonly used PEs are the eigenvectors of the laplacian matrix (Rampášek et al., 2022; Kreuzer et al., 2021) and the random walk probabilities (Ma et al., 2023; He et al., 2023; Dwivedi et al., 2022).

The goal of this work is to leverage new types of structural features as positional encodings emerging from quantum physics that can be obtained using quantum computers. The rapid development of quantum computers during the previous years indeed provides the opportunity to compute features that would be otherwise intractable. These features contain complex topological characteristics of the graph, and their inclusion has the potential to enhance the model's quality, reduce training or inference time, and decrease energy consumption. The idea of using quantum states containing topological features about the graphs has already been explored theoretically in (Verdon et al., 2019; Schuld et al., 2020; Henry et al., 2021; Thabet et al., 2022), and experimentally implemented in (Albrecht et al., 2023). Most of these works rely on the idea to prepare a parameterized quantum state by evolving a hamiltonian which encodes the topology of the graph. Measurements on this quantum state are used as graph features in a classical algorithm like Support Vector Machine. The majority of these works propose a method where most of the computation is done on a quantum processing unit. In contrast, (Thabet et al., 2022) proposes a hybrid algorithm where the attention matrix in conventional graph transformers is replaced by the the correlation matrix of the quantum state. The advantage of this method is to better leverage the capabilities of classical graph neural

[1]Pasqal, Massy, France [2]Sorbonne University, Paris, France. Correspondence to: Slimane Thabet <slimane.thabet@pasqal.com>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

networks (Xu et al., 2018; Hamilton et al., 2017), and to improve the expressivity of the model by jointly training classical parameters and quantum parameters. However it comes with significant difficulties in training quantum circuits (e.g., expensive computation of quantum gradients, noise-induced barren plateaus, only to name a few (Cerezo et al., 2022)). In this work, we propose a method similar to (Thabet et al., 2022) that circumvents these issues. In short, we propose to use the quantum features as positional encodings for graph transformers. In contrary to (Thabet et al., 2022), it separates the parts done on quantum and classical processors which improves the scalablitiy to large datasets.

The paper is organized as follows:

- In Sec. 2, we provide a concise overview of the existing research on graph transformers, along with references to the latest developments in quantum graph machine learning, and position our contributions in this landscape.
- Sec. 3 describes the core methodology that we propose. It details ways to construct a quantum state from a graph and explains how positional encodings can be extracted from that quantum state.
- In Sec. 4, we detail the properties of quantum positional encodings and show that they are superior to most other classical methods on SRGs.
- Sec. 5 presents the outcomes of our numerical experiments and includes discussions of the results. The code to run all the experiments is available at `https://github.com/pasqal-io/quantum-encodings-gnn`.

2. Related works and contributions

2.1. Graph Transformers

Several efforts have been made by the machine learning community to go beyond MPNNs due to several issues including the theoretical limitations of expressivity by the Weisfeiler-Lehman (WL) (Morris et al., 2019) test, oversmoothing (Chen et al., 2020), oversquashing (Topping et al., 2021), and difficutlties on heterophilic data (Zhu et al., 2020). Inspired by the success of transformers in natural language processing (Vaswani et al., 2017; Alayrac et al., 2022), new architectures of GNNs have been proposed to allow an all-to-all aggregation between the nodes of the graphs, the so-called graph transformers (GT) (Dwivedi & Bresson, 2020; Dwivedi et al., 2021; Rampášek et al., 2022; Kreuzer et al., 2021; Zhang et al., 2023; Ma et al., 2023). However, due to the quadratic cost of computing the attention process, they are not applicable to large-scale graphs of millions of nodes and more. It has been shown that GTs that include graph inductive biases such as MP modules perform better than those that do not (Rampášek et al., 2022; Ma et al., 2023).

2.2. Positional and Structural Encoding

Positional or structural embeddings are features computed from the graph that are concatenated to original node or edge features to enrich GNN architectures (either MPNN or GT). These two terms are used interchangeably in the literature and we denote them as "positional encodings" (PEs) in the rest of this work. PEs can include random walk probabilities (Rampášek et al., 2022; Ma et al., 2023), spectral information (Dwivedi et al., 2020; Rampášek et al., 2022; Kreuzer et al., 2021), shortest path distances (Li et al., 2018), or heat kernels (Mialon et al., 2021). They can also be learned (Dwivedi et al., 2021). We detail below the most common PEs used in the literature.

Laplacian Eigenvectors. The spectral information of the graph can be used as PE, more precisely the eigenvectors of the Laplacian matrix with the smallest eigenvalues, or laplacian eigenvectors (LE). For a line graph, the laplacian eigenvectors almost correspond to positional embeddings in the transformer architecture for sequences (Vaswani et al., 2017). The main issue of this encoding is to ensure that the model remains invariant by changing the sign of eigenvectors, and a solution has been proposed by (Lim et al., 2022).

Relative Random Walk Probabilities (RRWP). The authors of (Ma et al., 2023) introduced the RRWP with which they initialize their model. For a graph $\mathcal{G}$, let A be the adjacency matrix and D the degree matrix. Let P be a 3 dimensional tensor such that $P_{k,i,j} = (M^k)_{ij}$ with $M = D^{-1}A$. For each pair of node (i, j), we associate the vector $P_{:,i,j}$, i.e., the concatenation of the probabilities for all k to get from node i to node j in k steps in a random walk. $P_{:,i,i}$ is the same as the Random Walk Structural Encodings (RWSE) defined in (Rampášek et al., 2022). The authors of (Ma et al., 2023) highlight the benefits of RRWP. They prove that the Generalized Distance WL (GD-WL) test introduced by (Zhang et al., 2023) with RRWP is strictly more powerful than GD-WL test with the shortest path distance, and they prove universal approximation results of multi-layer perceptrons (MLP) initialized with RRWP. They also achieve state of the art results on most of benchmark datasets.

2.3. Quantum Computing for Graph Machine Learning

In recent years, quantum machine learning has seen a fast development with both theoretical and experimental advances (Huang et al., 2022; Cerezo et al., 2022). Using

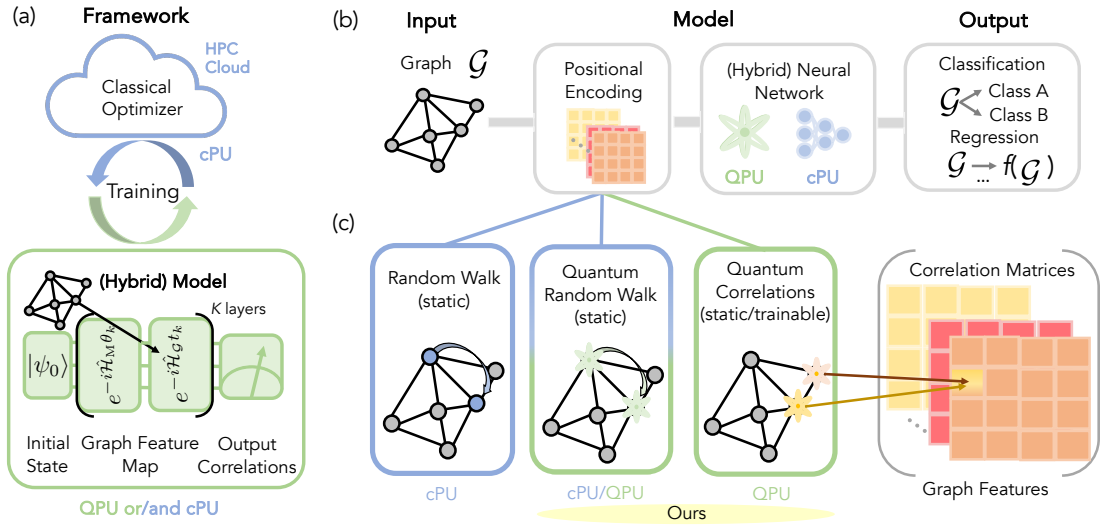

Figure 1. Summary of our method. **(a)** Our hybrid quantum-classical framework utilizes a classical computer for parameter optimization (if required) and employs a hybrid model using a Quantum Processing Unit (QPU) and a CPU and/or GPU, denoted as classical Processing Unit (cPU). In our quantum graph NN, we initialize QPU at a quantum state $|\psi_0\rangle$, apply a mixing Hamiltonian $\hat{\mathcal{H}}_{\mathrm{M}}$ evolution for a duration θ, and utilize a Hamiltonian $\hat{\mathcal{H}}_{\mathcal{G}}$ evolution for the graph feature map with a duration t. K layers are used to obtain a sufficiently expressive quantum model. Finally, the output is obtained by measuring correlators, e.g., $\langle Z_i Z_j \rangle$. See Section 3.1 for details. **(b)** Static or trainable PE is constructed for a graph $\mathcal{G}$ via **(c)** (quantum) random walk (static PE) or a quantum graph NN (static/trainable PE), which computes quantum correlations. Note that our PEs are not restricted to classical models (such as the transformer studied in this work) but are also applicable to all quantum models.

quantum computing for machine learning on graphs has already been proposed in several works, as reviewed in (Tang et al., 2022). The authors of (Verdon et al., 2019) realized learning tasks by using a parameterized quantum circuit depending on a Hamiltonian whose interactions share the topology of an input graph. Comparable ideas were used to build graph kernels from the output of quantum procedures, for photonic (Schuld et al., 2020) as well as neutral atom quantum processors (Henry et al., 2021). The latter was successfully implemented on quantum hardware (Albrecht et al., 2023). The architectures proposed in these papers are entirely quantum and only rely on classical computing for the optimization of variational parameters. Furthermore, these approaches can only be applied to graph-level tasks, whereas many applications are node-level or edge-level.

(Thabet et al., 2022) proposed a new hybrid architecture that uses the correlation matrix of quantum dynamics containing the information about the graph in the aggregation phase of a larger, entirely classical GNN architecture. Such a hybrid model presents the advantage of gaining access to hard-to-compute graph topological features through quantum dynamics while benefiting from the power of well-known existing classical architectures. Furthermore, the method presents the advantage to be applicable to all types of tasks, graph-level, node-level or edge-level. However, this approach presents some drawbacks. First, the model is difficult to train in simulation because the optimizer of quantum parameters needs to be adjusted separately from the optimizer of classical parameters. Secondly, it has been shown that methods to compute the gradients on a quantum computer are hardly scalable. (Abbas et al., 2023). Moreover, there were little theoretical indications about the power of the architecture compared to classical methods. The experiments have only been made on small datasets compared to the biggest ones available (a few 1000s of graphs and maximum 20 nodes)

2.4. Contributions

In light of the related works described above, we propose a method similar to (Thabet et al., 2022) that combines correlations from quantum dynamics and classical GNNs. Instead of parameterizing the quantum dynamics and training it as in (Thabet et al., 2022), we either use random parameterized states or well-chosen quantum states (such as ground states) and use the correlations as positional encoding for graph transformers. This approach removes the difficulty of training the quantum states, and can be included in most of classical architectures, which facilitates benchmarking. It keeps all the advantages of combining quantum and classical methods while being applicable to any graph machine learning tasks. In this work, we also give detailed theoretical properties of our positional encoding and show that they are strictly more powerful on some instances of graphs than random walks and spectral methods. Finally, we provide

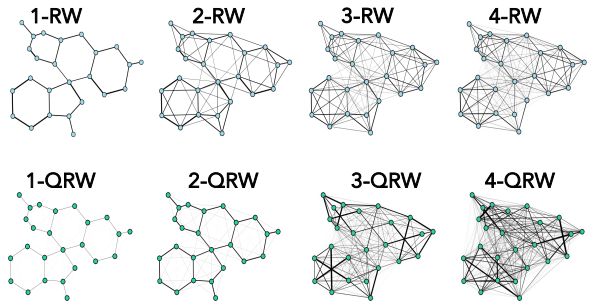

Figure 2. Visualization of RRWP (RW) and CQRW (QRW) steps for the fluorescein molecule that highlights different substructures (edge thickness). Note stronger long-range correlations (thicker edges between distant nodes) in 4th step of QRW versus RW.

benchmarks on large-scale datasets ($\sim$ 3 millions graphs).

3. Methods

In this section, we outline the process of mapping graphs to a quantum state of a QPU as used in (Henry et al., 2021; Thabet et al., 2022; Verdon et al., 2019). To extract graph features, we introduce correlators and define the concept of the ground state for a quantum graph representation. Finally, we explore an alternative approach for extracting graph features using quantum random walks (QRW) and their advantages over classical analogues. Figure 1 shows the summary of our hybrid approach in extracting positional encodings from graphs. We discuss the various steps in more detail below.

3.1. Quantum Graph Machine Learning

The graph as a quantum state. We explain in this subsection how to create a quantum state that contains relevant information about the graph. More details about quantum information processing can be found in (Nielsen & Chuang, 2002). We associate a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, to a quantum state $|\psi_\mathcal{G}\rangle$ of $|\mathcal{V}|$ qubits containing information about $\mathcal{G}$ via a hamiltonian $\hat{\mathcal{H}}_\mathcal{G}$ of the form

$$\hat{\mathcal{H}}_\mathcal{G} = \sum_{(i,j)\in\mathcal{E}} \hat{\mathcal{H}}_{ij} \tag{1}$$

where $\hat{\mathcal{H}}_{ij}$ is an Pauli string acting non-trivially on i and j only. We will be focusing on the Ising hamiltonian

$$\hat{\mathcal{H}}^I = \sum_{(i,j)\in\mathcal{E}} Z_i Z_j \tag{2}$$

and the XY hamiltonian

$$\hat{\mathcal{H}}^{XY} = \sum_{(i,j)\in\mathcal{E}} X_i X_j + Y_i Y_j. \tag{3}$$

We will note $|0\rangle$ and $|1\rangle$ the two eigenstates (or eigenvectors) of Z with respective eigenvalues 1 and -1, and we will use $\left\{|\mathbf{b}\rangle = \bigotimes_{i=1}^N |b_i\rangle\right\}_{\mathbf{b}\in\{0,1\}^N}$ as a basis of the 2^N-dimensional space of quantum states. We consider the quantum state obtained by alternated action of p layers of $\hat{\mathcal{H}}_\mathcal{G}$ and a *mixing* hamiltonian $\hat{\mathcal{H}}_M$ (that doesn't commute with $\hat{\mathcal{H}}_\mathcal{G}$, for instance $\hat{\mathcal{H}}_M \propto \sum_i Y_i$)

$$|\psi_\mathcal{G}(\boldsymbol{\theta})\rangle = \prod_{k=1}^{p} \left(\mathrm{e}^{-i\hat{\mathcal{H}}_M\theta_k}\mathrm{e}^{-i\hat{\mathcal{H}}_\mathcal{G}t_k}\right)\mathrm{e}^{-i\hat{\mathcal{H}}_M\theta_0}|\psi_0\rangle, \tag{4}$$

where $\boldsymbol{\theta} = (\theta_0, t_0, \theta_1, t_1, \dots \theta_p)$ is a real vector of parameters. The choice of these states is motivated by their similarity with the *Trotterized* dynamics of several quantum systems(Suzuki, 1976). Other states, including ones generated with analog evolution of quantum systems (Henriet et al., 2020) can also be considered.

Correlation. The correlations (or *correlators*) C_{ij} of local operators $\hat{\mathcal{O}}_i$ and $\hat{\mathcal{O}}_j$ acting respectively on qubits i and j can be defined either as the expectation value of their product $\langle\hat{\mathcal{O}}_i\hat{\mathcal{O}}_j\rangle$, or their covariance $\langle\hat{\mathcal{O}}_i\hat{\mathcal{O}}_j\rangle - \langle\hat{\mathcal{O}}_i\rangle\langle\hat{\mathcal{O}}_j\rangle$ (note that the orders matters if $\hat{\mathcal{O}}_i$ and $\hat{\mathcal{O}}_j$ don't commute). In the rest of the paper, we will indifferently call correlation the two former expressions, and give precisions when necessary. We will be focusing on the case where $\hat{\mathcal{O}}_i$ is a Pauli string of length 1 (i.e., X_i, Y_i or Z_i).

Ground state. The ground state (GS) of a system is defined as the lowest-energy eigenstate of its hamiltonian (when it is degenerate, one considers the *ground state manifold* $\mathbb{H}_{GS}$). GSs are in the general case hard to compute classically (Schollwöck et al., 2008). With that in mind, we call *ground state of the graph* the state

$$|\psi_{GS}\rangle = \frac{1}{\sqrt{|\mathbb{H}_{GS}|}} \sum_{\mathbf{b}\in\mathbb{H}_{GS}} |\mathbf{b}\rangle. \tag{5}$$

More details can be found in the appendix A.3.

Classical and Quantum Walks. Quantum walks, as introduced by (Aharonov et al., 1993), differ fundamentally from classical random walks by evolving through unitary processes, allowing for interference between different trajectories. This difference, related to the evolution of (real-valued classical) probabilities and (complex-valued quantum) amplitudes, leads to significant differences between dynamics of classical and quantum walks that can provide novel graph features which can be used as positional encodings (see Fig. 2). Quantum walks manifest in two primary types: *continuous-time quantum walks* (CQRW) (Farhi & Gutmann, 1998; Rossi et al., 2017) and *discrete-time quantum walks* (DQRW) (Lovett et al., 2010). In CQRW, the evolution of amplitudes $\langle i|\psi_\mathcal{G}(t)\rangle$ is given by

$$i\frac{\mathrm{d}}{\mathrm{d}t}\langle i|\psi_\mathcal{G}(t)\rangle = \sum_j \langle i|\hat{\mathcal{H}}_\mathcal{G}|j\rangle\langle j|\psi_\mathcal{G}(t)\rangle \tag{6}$$

where $|i\rangle$ is a computational basis state. The connection to a classical CRW can be made essentially by replacing the quantum amplitudes with classical probabilities, which we detail in the first section of the appendix. For an overview, refer to (Kempe, 2003). In Sec. 3.2.2, we define the particular cases of quantum walks studied in this work. More details can be found in the appendix A.1.

3.2. Positional encodings with quantum features

In this section, we detail our proposals to incorporate quantum features in GNN models and discuss the potential benefits and drawbacks. We focus on two types of encoding: the first uses the ground state of the graph as defined in section 3.1, the second uses the XY hamiltonian on the k-particles subspace (detailed in 3.2.2). Both methods use a quantum state that is difficult to prepare in the general case, so we expect to obtain features that are not available with classical approaches. For details of the corresponding algorithms, see appendix A.2.

3.2.1. Eigenvectors of the correlation on the ground state.

We propose to use the correlation matrix $C_{ij} = \langle Z_i Z_j \rangle$ on the ground state of the graph defined in Sec. 3.1. Since this matrix is symmetric with nonnegative eigenvalues, it can formally be used in the same place as the Laplacian matrix in graph learning models. Hence, we use the eigenvectors of this correlation matrix in the same way Laplacian eigenvectors (LE) are used in other architectures of graph transformers. Instead of taking the eigenvectors with the lowest eigenvalues as for the Laplacian eigenmaps, we take the ones with highest eigenvalues, since they are the ones in which most of the information about the correlation matrix is contained. We expect to face the same challenges due to the sign ambiguity (Dwivedi et al., 2021; Kreuzer et al., 2021), and to implement the same techniques to alleviate them (Lim et al., 2022).

3.2.2. k-particles quantum random walks (k-QRW).

In this subsection, we introduce the k-particles (or walkers) random walk positional encoding that can be obtained using $\hat{\mathcal{H}}^{XY}$. We denote by$\mathbb{H}_k$ the k-particles subspace (*i.e.* the Hilbert space obtained as the span of states $|\mathbf{b}\rangle$ of Hamming weight k, noted $|i_1 \dots i_k\rangle$, parameterized by k integers $i_1 \dots i_k \in \{0,1\}^k$). It is a well-known property that $\hat{\mathcal{H}}^{XY}$ stabilizes each of the $\mathbb{H}_k$s and we denote by $\hat{\mathcal{H}}_k^{XY}$ the XY hamiltonian restricted to $\mathbb{H}_k$ (Henry et al., 2021).
$\hat{\mathcal{H}}_k^{XY}$ can be seen as the adjacency matrix of a graph called the k *occupation graph*. Therefore, a quantum evolution of $\hat{\mathcal{H}}_k^{XY}$ can be seen as a quantum walk on the k occupation graph. We use the hamiltonian $\hat{\mathcal{H}}_k^{XY}$ to prepare a quantum state as in equation 4 that will represent a superposition over all k-tuples of nodes, and we measure observables for each pair of nodes which will give edge features. A complete simulation of the XY hamiltonian evolution is impractical for graphs of more than 20 nodes, therefore in this work we restrict ourselves to the simulation of $\hat{\mathcal{H}}_1^{XY}$ and $\hat{\mathcal{H}}_2^{XY}$. In the following, we give details about the features that were implemented in the experiments.

Continuous quantum random walk. For a 1-particle QRW, we calculate the probability

$$[X^{(1)}(t)]_{ij} = |\langle j| e^{-i\hat{\mathcal{H}}_1^{XY} t} |i\rangle|^2 \tag{7}$$

to find particle at node j coming from node i after time t. Similarly for a 2-particle QRW, we calculate

$$[X^{(2)}(t)]_{ij} = |\langle ij| e^{-i\hat{\mathcal{H}}_2^{XY} t} |\psi_i\rangle|^2 \tag{8}$$

where $|i,j\rangle \in \hat{\mathcal{H}}_2^{XY}$ is the state with walkers at nodes i and j and $|\psi_i\rangle \in \hat{\mathcal{H}}_2^{XY}$ the initial state. As choices for the initial distribution, we propose to use some localised state $|\psi_{\text{init}}\rangle \propto |ij\rangle$, or the uniform distribution over all pairs of nodes $|\psi_{\text{init}}\rangle \propto \sum_{(i,j)\in\mathcal{V}^2|i\neq j} |ij\rangle$, or the uniform distribution over the edges of the original graph $|\psi_{\text{init}}\rangle \propto \sum_{(i,j)\in\mathcal{E}} |ij\rangle$. From these we obtain the positional encodings using

$$\mathbf{P}_{ij} = [I, X^{(n_w)}(t_1), X^{(n_w)}(t_2) \dots X^{(n_w)}(t_K)]_{ij} \tag{9}$$

where $n_w = 1, 2$ is the number of walkers.

Quantum-inspired random walks. We propose a discrete version of the quantum features described above, where we consider powers of the hamiltonian $(\hat{\mathcal{H}}^{XY})^p$ for integers p instead of continuous evolutions as explained in the previous paragraph. The discrete powers are not implemented natively on a quantum computer, hence the name *quantum-inspired*. They are however directly comparable to the RRWP scheme of (Ma et al., 2023), and they are cheaper to compute than the continuous quantum random walks. We consider a *discrete* 2-particle quantum-inspired RW (2-QiRW) encoding that reads

$$\mathbf{P}_{ij} = \left[\langle ij| ((D_2^{XY})^{-1}\hat{\mathcal{H}}_2^{XY})^k |\psi_{\text{init}}\rangle \, |k \in [0, K]\right]_{ij} \tag{10}$$

where D_2^{XY} is the diagonal matrix sum of the rows of $\hat{\mathcal{H}}_2^{XY}$.

4. Theory

We present below our main theoretical results. The proofs for all the original statements made in this section can be found in appendix B. We mean by $X \sqsubset Y$ that algorithm X is weaker than algorithm Y or that Y can obtain a more refined node partition than X in the context of node coloring. $X \not\sqsubset Y$ means that there exists graphs for which X is stronger than Y and vice-versa in distinguishing ability.

4.1. QWs and their relationship with the WL test

In this section, we present some elements that help us to lay the groundwork for understanding the theoretical expressiveness of the approach developed in this paper. Since the seminal work of (Morris et al., 2019), it has become standard practice to bound this theoretical expressiveness by the various variants of the WL test. We propose the same here to better understand the advantages and limitations of quantum computation in such context. First, we position our current results on Hamiltonian evolution by the XY model in the landscape of the WL test and its variants (propositions 4.1, proposition 4.3 & theorem 4.2). We then focus on strongly regular graphs (SRGs) (propositions 4.6 and 4.7), which, as we describe next, constitute a relevant data set for the analysis of expressiveness.

Proposition 4.1. *When the input state of the unitary is a uniform superposition of delocalized states, the following holds :* 1-*QW* $\sqsubset$ *1-WL.*

This result is mainly due to the fact that we start from a delocalized state (uniform superposition), and we can actually show that it is more efficient to start from a localized state:

Proposition 4.2. *When the input state of the unitary is a uniform superposition or localized states the following holds:*

1. 1-*QW* $\sqsubset$ *2-WL*
2. 1-*QW* $\not\sqsubset$ *1-WL*

Both results presented so far are limited to a single quantum walker, but it's possible – and arguably no more expensive on quantum computers– to consider walks involving a larger number of walkers. For that, we simply need to prepare the initial state in a different configuration that allows such an analogy (Henry et al., 2021). In such case, it is possible to link it to the δ-k-LWL test, a variant of the WL test presented in (Morris et al., 2020):

Proposition 4.3. *When the input state of the unitary is a superposition state, the following holds :* k-*QW* $\sqsubset$ k-δ-*LWL.*

However, this result does not allow us to make a conclusive comparison between k-QW and k-WL. In a related paper (Morris et al., 2020), the authors also introduce an augmented version of k-δ-LWL,denoted k-δ-LWL$^+$, of which the k-δ-LWL is a particular case. They then show that both k-δ-WL and k-WL are less powerful than k-δ-LWL$^+$. The definition of a hierarchical rule to precisely position the k-QW in comparison with the k-WL is not provided here and is part of our future work. We do, however, have a series of experimental results for a further comparison, in the context of a particular type of graphs, the strongly regular graphs, that we define next.

4.2. QWs on SRGs

Definition 4.4. A strongly regular graph (SRG), noted $srg(\nu, k, \lambda, \mu)$, is a graph with ν vertices of fixed degree k, such that every pair of adjacent vertices have a fixed number λ of common neighbors, and every pair of non-adjacent vertices have a fixed number μ of common neighbors.

Each tuple (ν, k, λ, μ) defines a family of SRGs, and it is possible to find multiple non isomorphic graphs within the same family (Spence, 2024). We chose to work on these particular graphs because their regularity makes them especially difficult to distinguish within the same family. For instance, (Zhang et al., 2023) provide a worst-case analysis for the GD-WL test, a provably more powerful version of 1-WL , in the case of distance-regular graphs. Their examples include Rook's and Shrikhande graphs, both of diameter 2. We know on the other hand that a SRG is a distance-regular graph with a diameter 2 when $\mu \neq 0$ (Biggs, 1993). In (Ma et al., 2023), proposition 3.2., the authors show that GD-WL with RRWP distance is strictly more powerful than GD-WL with shortest path distance. They test their approach on 2 distance regular graphs. In the following, we propose to analyze the distinguishability of SRGs through the GD-WL test with RRWP distance, as well as a test involving a particular case of Hamiltonian evolution by the Ising model, constituting one of the rare configurations where it is possible to extract generic formulas without any costly simulation or quantum computing.

Proposition 4.5. *(Bodnar et al., 2021) It requires a* 3-*WL test or higher to distinguish two non-isomorphic strongly regular graphs from the same family.*

This result highlights the difficulty of the task, as a 3-WL test requires overlapping information from all the triplets in the graph, and is therefore costly in terms of memory and time as the size of the graph analyzed increases. We continue in the same line with the following result:

Proposition 4.6. *GD-WL with RRWP distance cannot, even with eigen-decomposition of the distance matrix, distinguish non isomorphic SRGs from the same family.*

This extends to the RRWP distance the results recovered in (Zhang et al., 2023), in which they show the same for shortest path distance and resistance distance.We provide in the next section a set of experiments as empirical evidence that show that in some cases, a GD-WL with correlations on 2-QW distinguishes SRGs.

4.3. Empirical study : Ising and XY models for the distinguishability of SRGs

So far we only focused on the XY hamiltonian, as it offers a nice basis for theoretical analysis. This is not the case for another widely used model in many-body physics : the Ising

model *cf.* section 3.1. As described in (Henry et al., 2021), the XY hamiltonian preserves the space of the states with the same number of occupied states, which allows the analogy between classical and quantum walks. Such an invariant is not available for the Ising model, and the counterpart for augmenting the number of walkers for higher expressiveness in the case of the Ising model comes from raising the value of p in equation 4, which can be seen as piling a larger number of "layers" in the hamiltonian evolution described there (Verdon et al., 2019). In this framework, we limit ourselves to the case of 1 and 2 bodies observables for many practical reasons. First they allow us to recover one and two dimensional tensors to describe graphs, which is convenient for learning tasks. They also allow us to construct the covariance matrix of such observables. This matrix can then be seen as the Gram matrix of a quantum graph kernel, which makes it suitable for comparison with the different variants of the GD-WL test. Finally, it is possible to derive a formal expression for one and two body observables resulting from an evolution for $p = 1$ in the general case, but not for larger values of p, which requires costly simulations or a quantum computer.

Proposition 4.7. *For any quantum state representing a strongly regular graph following an evolution from a local, uniform-field Ising Hamiltonian, the total occupation observable, as well as the linear (one body) and quadratic (two bodies) local occupation observables do not allow to distinguish two non isomorphic SRGs from the same family.*

This result shows that the case in which it is possible to derive formal expressions actually fails to distinguish non isomorphic SRGs. Figure 3 shows the results of the distinguishability in a set of 2 families, with 25 and 26 nodes. We compute both Ising and XY-hamiltonian evolutions, for $p = 2$ in the former and for a 2-QW in the latter. We use the following permutation invariant measure to compute the distance between a pair of graphs $d(G_i, G_j) = \frac{1}{2}\sum_{k=1}^{\nu^2} ||S(C(G_i)) - S(C(G_j)||_k$ where C is the correlation matrix, $S(M)$ a function that receives a matrix $M \in \mathbb{R}^{N\times N}$ as input and returns a vector $m \in \mathbb{R}^{N^2}$ containing the sorted elements of M. Using this formula, a non- zero value implies that the two graphs are not isomorphic, but the opposite is not necessarily true. We can see from this figure that it is possible to distinguish these graphs. We also ran experiments to verify that the distance between any of these graphs and a set of 5 randomly selected isomorphic counterparts is zero as expected.This shows empirically that for certain data sets, a two layers Ising evolution and a 2-QW are strictly more powerful than 2-WL. We also run the GD-WL test with the edge features $S(M)$ and we obtain that all pairs of graphs are successfully distinguished. Finally, it's important to point out that the classical k-WL test

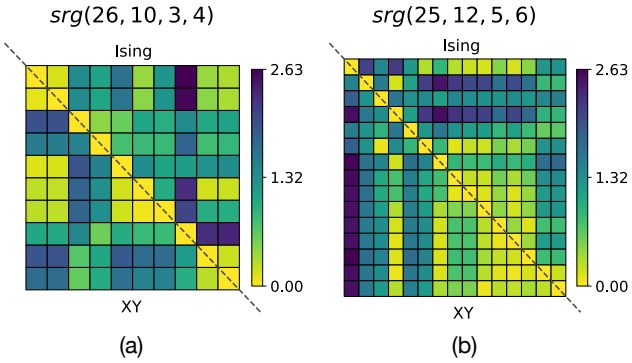

Figure 3. Normalized values of Ising correlations (upper triangular part) and XY model correlations (lower triangular part) for (a) the family $srg(26, 10, 3, 4)$ that includes 10 non-isomorphic graphs, and (b) the family of $srg(25, 12, 5, 6)$ that includes 15 non-isomorphic graphs.

requires comparisons between pairs of subsets of nodes of size k, rendering its complexity to at least N^k, N being the size of the graph. On the other hand, for quantum evolutions, the complexity of the algorithm is characterized by the number of shots that need to be measured in order to reconstruct the distribution of the desired observable. The number of shots is $\mathcal{O}(\frac{1}{\epsilon^2})$ for a precision up to ϵ (Huang et al., 2020). This number does not increase with the number of layers in the Ising hamiltonian case, where we only have a linear increase in the evolution time (which is short in practice), or with the number of quantum walkers that only depends on the initial input state preparation. This gives our approach an attractive potential for quantum advantage, albeit limited to datasets on which large values of k in the k-WL test are relevant. This also assumes that the k-QW as well as the k-layers Ising, are both strictly more powerful than the k-WL for any values of k. This property has been demonstrated in some cases for $k = 1$ and observed for $k = 2$ in the case of SRGs, but not yet demonstrated in the general case and for any value of k.

5. Experiments

5.1. Experiments on RW models

In this subsection, we test concatenating the QRW encodings to the RRWP in the GRIT model (Ma et al., 2023). We compute the (continuous) 1-CQRW for K random times and the discrete 2-QiRW for K steps. Those encodings are computed numerically since they are still tractable for graphs below 200 nodes compared to the higher order k-QiRW ones. We benchmark our method on 7 datasets from (Dwivedi et al., 2020), following the experimental setup of (Rampášek et al., 2022) and (Ma et al., 2023). Our method is compared to many other architectures and the results directly taken from (Ma et al., 2023). We do not perform

an extensive hyperparameter search for each architecture and only run ourselves the GRIT model by taking the same hyperparameters as the authors. The experiments are done by building on the codebase of (Ma et al., 2023) which is itself built on (Rampášek et al., 2022). More details about the protocol can be found in C.1, and more details about the datasets can be found in Appendix D. The results are included in Table 1. Our methods performs better on ZINC, MNIST and CIFAR10 than all others, and comes second for PATTERN and CLUSTER. We also benchmark our methods on large-scale datasets, ZINC-full (a bigger version of ZINC (Irwin et al., 2012)) and PCQM4MV2 (Hu et al., 2021). For these datasets, we run a variety of models (GINE, GatedGCN, and GRIT) with different position encodings (LE, RRWP, 2-QiRW, and a mix of RRWP and 2-QiRW. The results are reported figure 4 and the full numbers are table 4 in the appendix. Quantum features perform better for all models in the case of ZINC-full and for some models of PCQM4Mv2. All hyperparameters for this sections are reported in the appendix in tables 2 and 3.

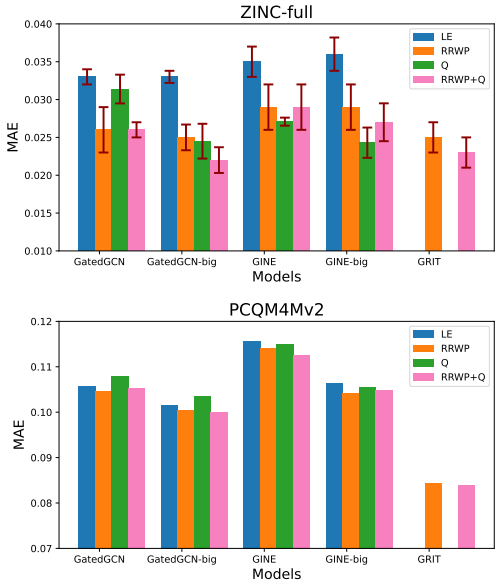

Figure 4. Test performance (mean absolute error) of different models with different positional encodings on large scale datasets. Q: 2-QiQRW features, RRWP+Q: RRWP and 2-QiQRW concatenated. Top : ZINC-full . Bottom: PCQM4Mv2. For ZINC-full, we show the mean and s.d of 4 runs with different random seeds. For PCQM4Mv2 we show the output of a single run.GRIT has 500k parameters for ZINC-full and 11.8M for PCQM4MV2. Normal models have about 200k parameters and big models about 700M.

5.2. Synthetic experiments

In this section, we provide one example of dataset with a binary graph classification task for which the use of the correlation matrix on the ground state as defined in 3.2.1 is more powerful than other commonly used features like the laplacian eigenvectors or RRWP. The idea is to construct graphs that will exhibit very different Ising ground states but similar spectral properties or random walk transition probabilities. We illustrate the differences between the encodings in Appendix C.2. We train classical models like GINE, GatedGCN, and GRIT on these datasets with LEs and RRWP as node features and edge features, and we compare it to a simple GCN model with the eigenvectors of the correlation matrix as node features. More details on the protocols can be found in C.2 We also benchmark the GRIT model with RRWP. The results are shown in table 5. The quantum encoding models achieve 100% accuracy whereas all other classical models achieve 45% accuracy.

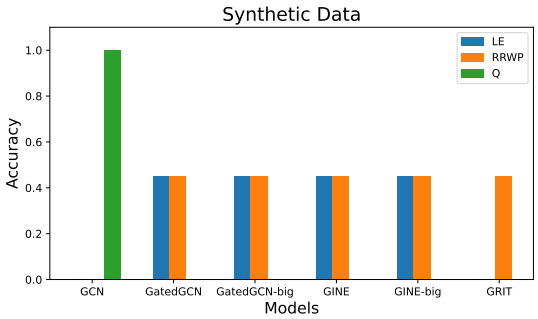

Figure 5. Results on synthetic data. We show the accuracy on the test set. Q: eigenvectors of the correlation on the ground state. GCN model has about 4k parameters, GRIT model has 500k parameters, other normal models have about 10k parameters, big models have about 60k parameters.

5.3. Discussion

We performed several experiments comparing the quantum encodings to the classical ones. Including the quantum walk features into state-of-the-art models improves their performances on most of the datasets tested. It is not surprising that the method works well for datasets such as ZINC for which random walks are known to provide relevant features (Rampášek et al., 2022). We limited ourselves to versions of quantum features that are efficiently computable and we were able to show a small gain in performance compared to state-of-the-art models. It is then plausible that using quantum features that cannot be classically accessible could lead to a greater improvement of models, if quantum hardware can be made widely available. We were able to engineer an artificial dataset for which classical approaches fail to perform the associated binary classification tasks and our quantum encoding perfectly realizes it, even with only 4k parameters where classical models have between 10k and 500k.

Table 1. Test performance in five benchmarks from (Dwivedi et al., 2020). We show the mean ± s.d. of 4 runs with different random seeds as in (Ma et al., 2023). Highlighted are the top first, second, and third results. Models are restricted to $\sim 500K$ parameters for ZINC, PATTERN, CLUSTER $\sim 100K$ for MNIST and CIFAR10. We compare our model to our run of GRIT and indicate the results obtained by the authors for information. Figures other than the last 3 lines are taken from (Ma et al., 2023). Models in **bold** are our models.

Model	ZINC	MNIST	CIFAR10	PATTERN	CLUSTER
	MAE↓	Accuracy↑	Accuracy↑	Accuracy↑	Accuracy↑
GIN	0.526 ± 0.051	96.485 ± 0.252	55.255 ± 1.527	85.387 ± 0.136	64.716 ± 1.553
GatedGCN	0.282 ± 0.015	97.340 ± 0.143	67.312 ± 0.311	85.568 ± 0.088	73.840 ± 0.326
EGT	0.108 ± 0.009	$\mathbf{98.173 \pm 0.087}$	68.702 ± 0.409	86.821 ± 0.020	$\mathbf{79.232 \pm 0.348}$
GPS	0.070 ± 0.004	98.051 ± 0.126	72.298 ± 0.356	86.685 ± 0.059	78.016 ± 0.180
GRIT (our run)	$\mathbf{0.060 \pm 0.002}$	$\mathbf{98.164 \pm 0.054}$	$\mathbf{76.198 \pm 0.744}$	$\mathbf{90.405 \pm 0.232}$	$\mathbf{79.856 \pm 0.156}$
GRIT 1-CQRW	$\mathbf{0.058 \pm 0.002}$	98.108 ± 0.111	$\mathbf{76.347 \pm 0.704}$	$\mathbf{87.205 \pm 0.040}$	78.895 ± 0.1145
GRIT 2-QiRW	$\mathbf{0.059 \pm 0.004}$	$\mathbf{98.204 \pm 0.048}$	$\mathbf{76.442 \pm 1.07}$	$\mathbf{90.165 \pm 0.446}$	$\mathbf{79.777 \pm 0.171}$

6. Conclusion

In this paper, we have investigated how quantum computing architectures can be used to construct new families of positional encodings for graph neural networks. This study involved measuring observables like correlations and probabilties for a quantum system with a hamiltonian that has the same topology as the graph of interest. We then integrated these observables as positional encodings and used them in different classical graph neural network architectures. We proved that some positional encodings that use quantum features are theoretically more expressive than ones based on simple random walk and laplacian eigenvectors, on strongly regular graphs. Our experiments show that state-of-the-art models can already be enhanced with restricted quantum features that are classically efficient to compute. This study provides strong indications that the full leverage of quantum hardware can lead to the development of high-performance architectures for certain tasks. Quantum simulation and analog quantum computing platforms are particularly suited to the type of time-dependent Hamiltonians we described here. In particular, Neutral Atom quantum hardware (Henriet et al., 2020; Wurtz et al., 2023) implement these natively (including the XY model (Chen et al., 2023)) and they are becoming usable through cloud access. Furthermore, we can create artificial classification tasks that are easily solvable with quantum-enhanced models while classical models fail. While it remains unclear whether an actual performance improvement will be observed on real-world use cases, and whether this improvement will justify the cost of using quantum hardware in the future., the results we obtain show that quantum-enhanced GNNs are a promising family of models that could be fully exploited with near-term quantum hardware.

Impact statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Acknowledgements

We thank Romain Fouilland and Shenyang Huang for useful discussions. We thank Shaheen Acheche for useful advice for writing the manuscript.

References

Abbas, A., King, R., Huang, H.-Y., Huggins, W. J., Movassagh, R., Gilboa, D., and McClean, J. R. On quantum backpropagation, information reuse, and cheating measurement collapse. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HF6bnhfSqH.

Aharonov, Y., Davidovich, L., and Zagury, N. Quantum random walks. *Phys. Rev. A*, 48(2):1687, 1993.

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

Albrecht, B., Dalyac, C., Leclerc, L., Ortiz-Gutiérrez, L., Thabet, S., D'Arcangelo, M., Cline, J. R., Elfving, V. E., Lassablière, L., Silvério, H., et al. Quantum feature maps for graph machine learning on a neutral atom quantum processor. *Physical Review A*, 107(4):042615, 2023.

Balasubramanian, S., Li, T., and Harrow, A. Exponential speedups for quantum walks in random hierarchical graphs, 2023.

Biggs, N. *Algebraic graph theory*. Cambridge Univ Pr, 1993.

Bodnar, C., Frasca, F., Wang, Y., Otter, N., Montufar, G. F., Lió, P., and Bronstein, M. Weisfeiler and lehman go topological: Message passing simplicial

networks. In Meila, M. and Zhang, T. (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 1026–1037. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/bodnar21a.html.

Cerezo, M., Verdon, G., Huang, H.-Y., Cincio, L., and Coles, P. J. Challenges and opportunities in quantum machine learning. *Nature Computational Science*, 2(9): 567–576, Sep 2022. ISSN 2662-8457. doi: 10.1038/s43588-022-00311-3. URL https://doi.org/10.1038/s43588-022-00311-3.

Chen, C., Bornet, G., Bintz, M., Emperauger, G., Leclerc, L., Liu, V. S., Scholl, P., Barredo, D., Hauschild, J., Chatterjee, S., Schuler, M., Läuchli, A. M., Zaletel, M. P., Lahaye, T., Yao, N. Y., and Browaeys, A. Continuous symmetry breaking in a two-dimensional rydberg array. *Nature*, 616(7958): 691–695, February 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-05859-2. URL http://dx.doi.org/10.1038/s41586-023-05859-2.

Chen, D., Lin, Y., Li, W., Li, P., Zhou, J., and Sun, X. Measuring and relieving the over-smoothing problem for graph neural networks from the topological view. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3438–3445, 2020.

Childs, A. M., Farhi, E., and Gutmann, S. An example of the difference between quantum and classical random walks. *Quantum Information Processing*, 1:35–43, 2002.

Childs, A. M., Cleve, R., Deotto, E., Farhi, E., Gutmann, S., and Spielman, D. A. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*. ACM, jun 2003. doi: 10.1145/780542.780552. URL https://doi.org/10.1145%2F780542.780552.

Das, A. and Chakrabarti, B. K. Colloquium: Quantum annealing and analog quantum computation. *Reviews of Modern Physics*, 80(3):1061, 2008.

Dwivedi, V. P. and Bresson, X. A generalization of transformer networks to graphs. *arXiv preprint arXiv:2012.09699*, 2020.

Dwivedi, V. P., Joshi, C. K., Laurent, T., Bengio, Y., and Bresson, X. Benchmarking graph neural networks. 2020.

Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. Graph neural networks with learnable structural and positional representations. *arXiv preprint arXiv:2110.07875*, 2021.

Dwivedi, V. P., Luu, A. T., Laurent, T., Bengio, Y., and Bresson, X. Graph neural networks with learnable structural and positional representations. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. URL https://openreview.net/forum?id=wTTjnvGphYj.

Fabila-Monroy, R., Flores-Peñaloza, D., Huemer, C., Hurtado, F., Urrutia, J., and Wood, D. R. Token graphs. *Graphs and Combinatorics*, 28(3):365–380, May 2011. ISSN 1435-5914. doi: 10.1007/s00373-011-1055-9. URL http://dx.doi.org/10.1007/s00373-011-1055-9.

Farhi, E. and Gutmann, S. Quantum computation and decision trees. *Phys. Rev. A*, 58(2):915, 1998.

Fürer, M. On the power of combinatorial and spectral invariants. *Linear Algebra and its Applications*, 432(9):2373–2380, 2010. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2009.07.019. URL https://www.sciencedirect.com/science/article/pii/S0024379509003620. Special Issue devoted to Selected Papers presented at the Workshop on Spectral Graph Theory with Applications on Computer Science, Combinatorial Optimization and Chemistry (Rio de Janeiro, 2008).

Gamble, J. K., Friesen, M., Zhou, D., Joynt, R., and Coppersmith, S. Two-particle quantum walks applied to the graph isomorphism problem. *Physical Review A*, 81(5): 052313, 2010.

Garey, M. R. and Johnson, D. S. *Computers And Intractability: A Guide To The Theory Of NP-Completeness*. 1979. ISBN 0-7167-1045-5.

Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. Neural message passing for quantum chemistry, 2017.

Hamilton, W., Ying, Z., and Leskovec, J. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Hamilton, W. L. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159.

Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs, 2018.

Harchaoui, Z. and Bach, F. Image classification with segmentation graph kernels. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–8. IEEE, 2007.

He, X., Hooi, B., Laurent, T., Perold, A., LeCun, Y., and Bresson, X. A generalization of vit/mlp-mixer to graphs. In Krause, A., Brunskill, E., Cho, K., Engelhardt, B., Sabato, S., and Scarlett, J. (eds.), *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pp. 12724–12745. PMLR, 2023. URL https://proceedings.mlr.press/v202/he23a.html.

Henriet, L., Beguin, L., Signoles, A., Lahaye, T., Browaeys, A., Reymond, G.-O., and Jurczak, C. Quantum computing with neutral atoms. *Quantum*, 4:327, 2020.

Henry, L.-P., Thabet, S., Dalyac, C., and Henriet, L. Quantum evolution kernel: Machine learning on graphs with programmable arrays of qubits. *Phys. Rev. A*, 104:032416, Sep 2021. doi: 10.1103/PhysRevA.104.032416. URL https://link.aps.org/doi/10.1103/PhysRevA.104.032416.

Hu, W., Fey, M., Ren, H., Nakata, M., Dong, Y., and Leskovec, J. Ogb-lsc: A large-scale challenge for machine learning on graphs. *arXiv preprint arXiv:2103.09430*, 2021.

Huang, H.-Y., Kueng, R., and Preskill, J. Predicting many properties of a quantum system from very few measurements. *Nature Physics*, 16(10):1050–1057, 2020.

Huang, H.-Y., Broughton, M., Cotler, J., Chen, S., Li, J., Mohseni, M., Neven, H., Babbush, R., Kueng, R., Preskill, J., and McClean, J. R. Quantum advantage in learning from experiments. *Science*, 376(6598):1182–1186, 2022. doi: 10.1126/science.abn7293. URL https://www.science.org/doi/abs/10.1126/science.abn7293.

Irwin, J. J., Sterling, T., Mysinger, M. M., Bolstad, E. S., and Coleman, R. G. Zinc: a free tool to discover chemistry for biology. *Journal of chemical information and modeling*, 52(7):1757–1768, 2012.

Kempe, J. Quantum random walks hit exponentially faster, 2002.

Kempe, J. Quantum random walks: an introductory overview. *Contemporary Physics*, 44(4):307–327, 2003.

Kipf, T. N. and Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016. doi: 10.48550/ARXIV.1609.02907. URL https://arxiv.org/abs/1609.02907.

Konaklieva, M. I. Molecular targets of β-lactam-based antimicrobials: beyond the usual suspects. *Antibiotics*, 3 (2):128–142, 2014.

Kreuzer, D., Beaini, D., Hamilton, W., Létourneau, V., and Tossou, P. Rethinking graph transformers with spectral attention. *Advances in Neural Information Processing Systems*, 34:21618–21629, 2021.

Li, Q., Han, Z., and Wu, X.-M. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Lim, D., Robinson, J., Zhao, L., Smidt, T., Sra, S., Maron, H., and Jegelka, S. Sign and basis invariant networks for spectral graph representation learning. *arXiv preprint arXiv:2202.13013*, 2022.

Lovett, N. B., Cooper, S., Everitt, M., Trevers, M., and Kendon, V. Universal quantum computation using the discrete-time quantum walk. *Phys. Rev. A*, 81(4):042330, 2010.

Ma, L., Lin, C., Lim, D., Romero-Soriano, A., Dokania, P. K., Coates, M., Torr, P., and Lim, S.-N. Graph inductive biases in transformers without message passing. *arXiv preprint arXiv:2305.17589*, 2023.

Mialon, G., Chen, D., Selosse, M., and Mairal, J. Graphit: Encoding graph structure in transformers. *arXiv preprint arXiv:2106.05667*, 2021.

Morris, C., Ritzert, M., Fey, M., Hamilton, W. L., Lenssen, J. E., Rattan, G., and Grohe, M. Weisfeiler and leman go neural: Higher-order graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pp. 4602–4609, 2019.

Morris, C., Rattan, G., and Mutzel, P. Weisfeiler and leman go sparse: towards scalable higher-order graph embeddings. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Nielsen, M. A. and Chuang, I. Quantum computation and quantum information, 2002.

Powers, D. L. and Sulaiman, M. M. The walk partition and colorations of a graph. *Linear Algebra and its Applications*, 48:145–159, 1982. ISSN 0024-3795. doi: https://doi.org/10.1016/0024-3795(82)90104-5. URL https://www.sciencedirect.com/science/article/pii/0024379582901045.

Rampášek, L., Galkin, M., Dwivedi, V. P., Luu, A. T., Wolf, G., and Beaini, D. Recipe for a general, powerful, scalable graph transformer. *arXiv preprint arXiv:2205.12454*, 2022.

Rattan, G. and Seppelt, T. *Weisfeiler-Leman and Graph Spectra*, pp. 2268–2285. doi: 10.1137/1.9781611977554.ch87. URL https://epubs.siam.org/doi/abs/10.1137/1.9781611977554.ch87.

Rossi, M. A., Benedetti, C., Borrelli, M., Maniscalco, S., and Paris, M. G. Continuous-time quantum walks on spatially correlated noisy lattices. *Phys. Rev. A*, 96(4): 040301, 2017.

Sanchez-Gonzalez, A., Godwin, J., Pfaff, T., Ying, R., Leskovec, J., and Battaglia, P. Learning to simulate complex physics with graph networks. In *International Conference on Machine Learning*, pp. 8459–8468. PMLR, 2020.

Schollwöck, U., Richter, J., Farnell, D. J., and Bishop, R. F. *Quantum magnetism*, volume 645. Springer, 2008.

Schuld, M., Brádler, K., Israel, R., Su, D., and Gupt, B. Measuring the similarity of graphs with a gaussian boson sampler. *Phys. Rev. A*, 101:032314, Mar 2020. doi: 10.1103/PhysRevA.101.032314. URL https://link.aps.org/doi/10.1103/PhysRevA.101.032314.

Scott, J. Social network analysis: developments, advances, and prospects. *Social network analysis and mining*, 1(1): 21–26, 2011.

Shiau, S.-y. and Joynt, R. Physically-motivated dynamical algorithms for the graph isomorphism problem. *arXiv preprint quant-ph/0312170*, 2003.

Spence, E., 2024. URL https://www.maths.gla.ac.uk/~es/srgraphs.php. Accessed on 31st January 2024.

Suzuki, M. Generalized trotter's formula and systematic approximants of exponential operators and inner derivations with applications to many-body problems. *Communications in Mathematical Physics*, 51(2):183–190, 1976.

Tang, Y., Yan, J., and Edwin, H. From Quantum Graph Computing to Quantum Graph Learning: A Survey. *arXiv e-prints*, art. arXiv:2202.09506, February 2022.

Thabet, S., Fouilland, R., and Henriet, L. Extending graph transformers with quantum computed aggregation. *arXiv preprint arXiv:2210.10610*, 2022.

Topping, J., Di Giovanni, F., Chamberlain, B. P., Dong, X., and Bronstein, M. M. Understanding over-squashing and bottlenecks on graphs via curvature. *arXiv preprint arXiv:2111.14522*, 2021.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., and Bengio, Y. Graph attention networks, 2018.

Verdon, G., McCourt, T., Luzhnica, E., Singh, V., Leichenauer, S., and Hidary, J. Quantum Graph Neural Networks. *arXiv e-prints*, art. arXiv:1909.12264, September 2019.

Wurtz, J., Bylinskii, A., Braverman, B., Amato-Grill, J., Cantu, S. H., Huber, F., Lukin, A., Liu, F., Weinberg, P., Long, J., Wang, S.-T., Gemelke, N., and Keesling, A. Aquila: Quera's 256-qubit neutral-atom quantum computer, 2023.

Xu, K., Hu, W., Leskovec, J., and Jegelka, S. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.

Ying, C., Cai, T., Luo, S., Zheng, S., Ke, G., He, D., Shen, Y., and Liu, T. Do transformers really perform badly for graph representation? In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 28877–28888, 2021.

Zhang, B., Luo, S., Wang, L., and He, D. Rethinking the expressive power of gnns via graph biconnectivity. *arXiv preprint arXiv:2301.09505*, 2023.

Zhu, J., Yan, Y., Zhao, L., Heimann, M., Akoglu, L., and Koutra, D. Beyond homophily in graph neural networks: Current limitations and effective designs. *Advances in Neural Information Processing Systems*, 33:7793–7804, 2020.

Zitnik, M., Agrawal, M., and Leskovec, J. Modeling polypharmacy side effects with graph convolutional networks. *Bioinformatics*, 34(13):i457–i466, 2018.

A. Methods

In this section, we present the details about classical and quantum random walks.

A.1. Differences between classical and quantum random walks

In this section, we make the connection between (continuous) classical and quantum walks explicit. In CRWs, the probability of a walker being at vertex i and time t is represented as $p_i(t)$, which follows the differential equation $\frac{\mathrm{d}}{\mathrm{d}t}p_i(t) = -\sum_j G_{ij}p_j(t)$. Here, the infinitesimal generator $G_{ij} = -\gamma$ if an edge exists between nodes i and j, and 0 otherwise, with diagonal elements $G_{ii} = k_i\gamma$ determined by the node degree k_i. Considering now a quantum evolution with a graph Hamiltonian $\hat{\mathcal{H}}_{\mathcal{G}}$, given a 2^N-dimensional Hilbert space of N qubits, the Schrödinger equation which governs the evolution of a quantum state $|\psi_{\mathcal{G}}\rangle$ when projected onto a state $|i\rangle$ is given as

$$\underbrace{i\frac{\mathrm{d}}{\mathrm{d}t}\langle i|\psi_{\mathcal{G}}(t)\rangle = \sum_j \langle i|\hat{\mathcal{H}}_{\mathcal{G}}|j\rangle\langle j|\psi_{\mathcal{G}}(t)\rangle}_{\text{Quantum}} \longleftrightarrow \underbrace{\frac{\mathrm{d}}{\mathrm{d}t}p_i(t) = -\sum_j G_{ij}p_j(t).}_{\text{Classical}} \tag{11}$$

Note the similarity between the differential equations of CQRW and CRW. A quantum analogue of CRW can be obtained by taking $\langle i|\hat{\mathcal{H}}_{\mathcal{G}}|j\rangle = G_{ij}$. The probabilities are preserved as the sum of amplitude squared, $\sum_i |\langle i|\psi_{\mathcal{G}}(t)\rangle|^2 = 1$, in the quantum case, instead of $\sum_i p_i(t) = 1$ in the classical case. Using this formalism, any quantum evolution can be thought of as a CQRW (Childs et al., 2002). Notably, quantum walks have demonstrated exponential hitting time advantage for graphs like hypercubes (Kempe, 2002) and glued binary trees (Childs et al., 2003). These results have been recently extended for more general hierarchical graphs (Balasubramanian et al., 2023). For an overview, refer to (Kempe, 2003).

A.2. Algorithms

In this appendix, we detail the algorithms for the approach described in this paper.
Firstly, we give a generic algorithm that allows the simulation of quantum features on a classical computer, according to an arbitrary choice of Hamiltonian and quantum observables.

Algorithm 1 Positional encoding with a generic classical simulation of a Hamiltonian evolution

Input : a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, a hyper-parameter set $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_K\}$[1] and a set of quantum observables $\{\hat{\mathcal{O}}_i\}_{i\in\mathcal{V}}$ [2]
Output : the matrices of node pairs positional encoding $\mathbf{P_E} \in \mathbb{R}^{d_e\times K}$, and single nodes positional encoding $\mathbf{P_N} \in \mathbb{R}^{N\times K}$ where $d_e = \binom{N}{2}$, and N the number of nodes in the graph.

for $k \in \{1, 2, ..., K\}$ **do**
  Compute $|\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)\rangle$ according to equation (4)[3]
  $(\mathbf{P_E})_{ij,k} = \langle\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)|\hat{\mathcal{O}}_i\hat{\mathcal{O}}_j|\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)\rangle - \langle\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)|\hat{\mathcal{O}}_i|\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)\rangle\,\langle\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)|\hat{\mathcal{O}}_j|\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)\rangle$
  $(\mathbf{P_N})_{i,k} = \langle\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)|\hat{\mathcal{O}}_i^2|\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)\rangle - \langle\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)|\hat{\mathcal{O}}_i|\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)\rangle^2$
end for
return $\mathbf{P_E}$ and $\mathbf{P_N}$.

[1] Here each $\boldsymbol{\theta}_i = (\theta_{i,0}, t_{i,0}, \theta_{i,1}, t_{i,1}, \ldots, \theta_{i,p_i})$ with p_i fixed by the user as the number of "layers" in the i-th Hamiltonian evolution, and potentially (but not necessarily) $p_i \neq p_j$ for $i \neq j$.

[2] e.g $\mathcal{O}_i = \frac{1}{2}(\mathbb{I} - Z_i)$ in this paper, and $\hat{\mathcal{O}}_i = \bigotimes_{j<i}\mathbb{I} \otimes \hat{\mathcal{O}} \otimes \bigotimes_{j>i}\mathbb{I}$ in general

[3] This approach implies that we can store and reuse the quantum state $|\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)\rangle$, which is true when running simulations on classical computers but not in the case where we do it on quantum computers, where we need to run a certain number of measurements in order to obtain an estimation of the desired quantities, as showcased in algorithm 3

Next we provide an efficient classical algorithm for the simulation of the 2 particles quantum walk (which also applies to the 1 particle random walk, if one applies the following on the original adjacency matrix of the graph).

Algorithm 2 Positional encoding with a classically simulated 2 particles QW

Input : a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, a hyper-parameter set $\{t_1, t_2, ..., t_K\}$

Output : the matrices of node pairs positional encoding $\mathbf{P_E} \in \mathbb{R}^{d_e \times K}$, and single nodes positional encoding $\mathbf{P_N} \in \mathbb{R}^{N \times K}$ where $d_e = \binom{N}{2}$, and N the number of nodes in the graph.

- Compute the adjacency matrix $\boldsymbol{A}_2$ of the (2 particles) product graph : we draw an edge in the latter if the symmetric difference $\{i_1, j_1\}\Delta\{i_2, j_2\}$ is a pair of adjacent nodes in the input graph[4].
- Define $\hat{\mathcal{H}}_1^{XY}$ as in equation (3) on the edges of the product graph, indexed as $(i_1 j_1, i_2 j_2)$

for $k \in \{1, 2, ..., K\}$ **do**
  if Continuous random walk scheme **then**
    $(\mathbf{P_E})_{ij,k} = |\langle ij| e^{-i.t_k.\hat{\mathcal{H}}_1^{XY}} |\psi_0\rangle|^2 \;\; \forall i \in \{1, .., N\}$ and $\forall j \in \{i+1, ..., N\}$ [5]
    $(\mathbf{P_N})_{i,k} = |\langle ii| e^{-i.t_k.\hat{\mathcal{H}}_1^{XY}} |\psi_0\rangle|^2 \;\; \forall i \in \{1, .., N\}$
  else if Quantum-inspired random walk scheme **then**
    $(\mathbf{P_E})_{ij,k} = [(\boldsymbol{D}_2^{-1}.\boldsymbol{A}_2)^k.\psi_0]_{ij} \;\; \forall i \in \{1, .., N\}$ and $\forall j \in \{i+1, ..., N\}$
    $(\mathbf{P_N})_{i,k} = [(\boldsymbol{D}_2^{-1}.\boldsymbol{A}_2)^k.\psi_0]_{ii} \;\; \forall i \in \{1, .., N\}$[6]
  end if
end for
return $\mathbf{P_E}$ and $\mathbf{P_N}$.

Finally, we present an algorithm for computing our quantum features using a quantum computer.

Algorithm 3 Positional encoding with a generic Hamiltonian evolution on a quantum computer

Input : a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, a hyper-parameter set $\{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, ..., \boldsymbol{\theta}_K\}$, the number of measurements N_m for the estimation of the average of quantum observable $\hat{\mathcal{O}}_i$

Output : the matrices of node pairs positional encoding $\mathbf{P_E} \in \mathbb{R}^{d_e \times K}$, and single nodes positional encoding $\mathbf{P_N} \in \mathbb{R}^{N \times K}$ where $d_e = \binom{N}{2}$, and N the number of nodes in the graph.

for $k \in \{1, 2, ..., K\}$ **do**
  for $m \in \{1, 2, ..., N_m\}$ **do**
    Let the system evolve according to equation (4) to prepare the quantum state $|\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)\rangle$
    Perform a measurement of the observables[7] $\{\hat{\mathcal{O}}_i\}_{i \in \mathcal{V}}$
  end for

  $(\mathbf{P_E})_{ij,k} = \langle\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)|\hat{\mathcal{O}}_i\hat{\mathcal{O}}_j|\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)\rangle - \langle\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)|\hat{\mathcal{O}}_i|\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)\rangle \langle\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)|\hat{\mathcal{O}}_j|\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)\rangle$ estimated from the measurements
  $(\mathbf{P_N})_{i,k} = \langle\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)|\hat{\mathcal{O}}_i^2|\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)\rangle - \langle\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)|\hat{\mathcal{O}}_i|\psi_{\mathcal{G}}(\boldsymbol{\theta}_k)\rangle^2$ estimated from the measurments
end for
return $\mathbf{P_E}$ and $\mathbf{P_N}$.

A.3. Ground state.

The ground state of a system is defined as the lowest-energy eigenstate of its hamiltonian (when it is degenerate, one considers the *ground state manifold* $\mathbb{H}_{GS}$). Ground state properties are widely studied in many-body physics and their

[4]This construction is specific to the case of 2 particles quantum walks, for more details see (Fabila-Monroy et al., 2011; Henry et al., 2021)

[5]$|\psi_0\rangle$ could be for example a zero state or a uniform superposition state

[6]Here $\boldsymbol{D}_2$ is the diagonal degree matrix of the product graph, and $\psi_0 \in \mathbb{R}^{d_e}$ is a vector that represents the uniform superposition state over edges *i.e* contains ones in the index pairs ij that are actually linked in the input graph, or alternatively a vector of ones which extends the previous property to all pairs in the original graph.

[7]All the observables in the set are measured simultaneously which prevents large computational overheads

properties depend on the topology of the graph. Preparing this state is the purpose of quantum annealing (Das & Chakrabarti, 2008). When using neutral atom quantum processors (Henriet et al., 2020), one can natively address hamiltonians of the form

$$\hat{\mathcal{H}}_{\mathcal{G}} = \sum_{(i,j)\in\mathcal{E}} J_{ij}(Z_i - \alpha_i I)(Z_j - \alpha_j I) \tag{12}$$

with α_i real coefficients. Its eigenstates are the basis states $|\mathbf{b}\rangle$ described above. In the case where $\alpha_i = 1 - \delta/(2z_i)$ with $z_i = \sum_{j|(i,j)\in\mathcal{E}} J_{ij}$ and $J_{ij} = 1/4$, the eigenenergies (or eigenvalues) are

$$E(\mathbf{b}) = \sum_{i,j\in\mathcal{E}} b_i b_j - \delta \sum_{i=1}^{N} b_i. \tag{13}$$

When $0 < \delta < 1$, this is the cost function associated with the maximum independent set problem, a NP-hard problem (Garey & Johnson, 1979). In the absence of degeneracy-lifting or symmetry-breaking effects, a quantum annealing scheme would prepare a symmetric, equal-weight superposition of all maximum independent sets. With that in mind, we will call *ground state of the graph* the state

$$|\psi_{GS}\rangle = \frac{1}{\sqrt{|\mathbb{H}_{GS}|}} \sum_{\mathbf{b}\in\mathbb{H}_{GS}} |\mathbf{b}\rangle. \tag{14}$$

B. Theory

In this section, we present the proofs relating the expressiveness of QW with well-known WL tests and other graph invariants such as the Furer Invariant (Fürer, 2010).

B.1. Proof of proposition 4.1

1-QW $\sqsubset$ 1-WL when the input state of the unitary is a uniform superposition state.
We show that beginning with an initial uniform superposition state, the action of $\mathcal{H}_{\mathcal{XY}}$ evolution is similar to WL coloring iterations. We consider a graph $\mathcal{G}$, with vertex set $\mathcal{V}$ and consisting of N nodes. Suppose we begin with 1 particle QW. The action of $\mathcal{H}_{\mathcal{XY}}$ restricted to 1-particle subspace is then equivalent to action of the adjacency matrix A. Therefore we have :

$$\begin{aligned} e^{\mathbf{i}\mathcal{H}_{\mathcal{XY}}t} &= \sum_{k=0}^{\infty} \frac{(\mathbf{i}.t)^k}{k!} (H_{XY})^k \\ &\equiv 1 + \mathbf{i}At + \mathbf{i}^2 \frac{1}{2} A^2 t^2 + .. \end{aligned} \tag{15}$$

The action of A on the identity vector is simply the vector of degrees of the nodes, which is the first WL iteration. Further operating with A leads to a vector where we add degrees of the neighbours of a node. Further iterations simply lead to vectors where two entries are equal if they have same degrees and equal number of neighbours also of same degrees, which corresponds to stable WL coloring. This scheme corresponds exactly to a 1-WL test with the sum operation as the update function. To see this, suppose we define $|I\rangle$ as the normalized vector of uniform superposition. Each entry of the vector is $\frac{1}{\sqrt{N}}$. After the action of adjacency matrix A we obtain the vector $|I^{(1)}\rangle$, which will be:

$$|I^{(1)}\rangle = \frac{1}{\sqrt{N}} \begin{bmatrix} d_1 \\ d_2 \\ .. \\ .. \\ d_N \end{bmatrix} \tag{16}$$

where $d_1, d_2, ..., d_N$ are the degrees of the nodes. This can be interpreted as the first step of WL coloring where each node is assigned the color depending on the degree. Again applying A to vector $|I^{(1)}\rangle$ we obtain the vector $|I^{(2)}\rangle$, which will be:

$$|I^{(1)}\rangle = \frac{1}{\sqrt{N}} \begin{bmatrix} \sum_{i\in\mathcal{N}(1)} d_i \\ \sum_{i\in\mathcal{N}(2)} d_i \\ .. \\ .. \\ \sum_{i\in\mathcal{N}((N)} d_i \end{bmatrix} \tag{17}$$

where $\mathcal{N}(i)$ is the set of neighbours of node i. Thus in this iteration, the degrees of the neighbours of a node are aggregated [8]. Further steps simply lead to aggregation of neighbour properties for a given node until stable coloring is reached i.e same colored neighbours will have same entry in the vector. It is known that i^{th} entry of $A^k|\mathbf{1}\rangle$ will correspond to the number of walks of length k beginning from node i. Every k^{th} iteration induces a refinement on the previous step until convergence. It has been shown that stable WL coloring is a refinement of the partition obtained at convergence for some k for $A^k|\mathbf{1}\rangle$ and is in fact equal for a large number of graphs (Powers & Sulaiman, 1982). Since the evolution due to $\mathcal{H}_{\mathcal{XY}}$ is simply the weighted summation of these vectors, it follows that that the output vector for 1-QW with input superposition $\sqsubset$ 1-WL.

B.2. Proof of proposition 4.2

1-QW $\not\sqsubseteq$ 1-WL $\sqsubseteq$ 2-WL when the input state of the unitary is a localized state.
The complete proof relies on this two following statements :

1. 1-QW $\not\sqsubseteq$ 1-WL when the input state of the unitary is a general input state.
   We can see that in fact for a localized input state a single-particle, QW can be more expressive than a 1-WL test for some graphs. For example, beginning with a particle localized at one of the nodes, the action of the $\mathcal{H}_{\mathcal{XY}}$ as seen above has contributions from different powers of A. Beginning with a localized input state, the kth power of A tell about the number of paths of length k between a set of nodes. If 2 regular graphs differ in these quantities, the set of probabilities $p_{ij} = p(i \to j)$ or probability of a particle to go from node i to j will be different. Thus, a single particle QW can distinguish 2 such regular graphs. However, we know that a 1-WL cannot distinguish regular graphs. Hence 1-QW $\not\sqsubseteq$ 1-WL for a general input.

2. 1-QW $\sqsubset$ F.I when the input state of the unitary is a localized state.
   For a general input and output state, the time averaged transition probability for evolution with $\mathcal{H}_{\mathcal{XY}}$ to go from $|\psi\rangle$ to $|\phi\rangle$ is given by:

$$p(|\phi\rangle \to |\psi\rangle) = lim_{T\to\infty} \int_0^T dt |\langle\phi|e^{i\mathcal{H}_{\mathcal{XY}}t}|\psi\rangle|^2 dt \tag{18}$$

   This can be further written as

$$p_{|\psi\rangle\to|\phi\rangle}(\infty) = \sum_{\lambda\in Sp(\mathcal{H}_{\mathcal{XY}})} |\langle\phi|P^\lambda|\psi\rangle|^2 \tag{19}$$

   where $\mathcal{H}_{\mathcal{XY}} = \sum_{\lambda\in Sp(\mathcal{H}_{\mathcal{XY}})} \lambda P^\lambda$, $Sp(\mathcal{H}_{\mathcal{XY}})$ is the set of eigenvalues of $\mathcal{H}_{\mathcal{XY}}$. For a single particle QW, this set is exactly the set of eigenvalues of A. P^λ are the projectors into the eigenspace spanned by the corresponding eigenvectors. (Fürer, 2010) defines the spectral invariant based on projectors P^λ. We refer to this as the Furer Invariant (F.I.). The Furer invariant is a spectral invariant defined in terms of projectors P^λ defined above. The projector P^λ by definition can be written as $|e_\lambda\rangle\langle e_\lambda|$, where $|e_\lambda\rangle$ is the eigenvector corresponding to the eigenvalue λ. In matrix form this can be written as

$$P^\lambda = \begin{bmatrix} p^\lambda_{11} & p^\lambda_{11} & .. & p^\lambda_{1n} \\ p^\lambda_{21} & p^\lambda_{22} & .. & p^\lambda_{2n} \\ . & . & . & \\ . & . & . & \\ p^\lambda_{n1} & p^\lambda_{n2} & .. & p^\lambda_{nn} \end{bmatrix} \tag{20}$$

   The quantities $\langle b_i|P^\lambda|b_j\rangle$ and $\langle b_i|P^\lambda|b_i\rangle$ represent the non-diagonal and the diagonal entries of the matrix P^λ respectively. The $|b_i\rangle$ represent the coordinate basis vectors where a vector $|0_1, 0_2.., 1_i, 0.., 0\rangle$ would represent the node i. The quantity $\langle b_i|P^\lambda|b_j\rangle$ represents the product of angles made by the coordinate basis vectors corresponding to node i, j with the eigenspace defined by λ. Similarly $\langle b_i|P^\lambda|b_i\rangle$ represents the angle made by the coordinate basis vector of node i with the eigenspace defined by λ. These quantities along with the spectra of the adjacency matrix define the Furer invariant. Expanding each of the vectors $|\phi\rangle$ and $|\psi\rangle$ in terms of the eigenvectors, it can be seen that

[8] Again, we are here in a case where the aggregation function is the sum operation, which is not an injective function. This is the main reason for which the $1-QW$ is less powerful than the $1-WL$, which allows any sort of aggregation functions and most notably injective functions.

the probability $p(i \to j)$ can be computed using the Furer invariant. Thus, the set of probabilities is weaker than the Furer invariant.

Thus, we see that 1-QW $\sqsubset$ F.I. In (Rattan & Seppelt), the authors prove that F.I $\sqsubset$ 2-WL. Therefore 1-QW$\sqsubset$ 2-WL. These inequalities along with Proposition 1.1 tell us that 1-QW $\not\sqsubseteq$ 1-WL and 1-QW $\sqsubseteq$ 2-WL.

B.3. Proof of proposition 4.3

k-QW $\sqsubset$ to k-δ-LWL for uniform input superposition state.
For a general k-particle case, the WL coloring for a k-particle occupation graph is the same as k-δ-LWL coloring as defined in (Morris et al., 2020). This is because the colors of the k-tuples are updated only based on the local neighbourhood, as is the case for the k-δ-LWL test. The k-particle occupancy graph G^k, where every node now labelled by a k-tuple represents the presence of k particles at that node, is constructed in the following way.: We begin with a k-tuple , $(v_1, v_2, ..v_i, v_{i+1}, ..v_k)$. An adjacent k-tuple is found by replacing v_i with a node w_i such that $w_i \in \mathcal{N}(v_i)$. In a k-δ-LWL test, colors of a particular k-tuple are updated based on colors of the *local* neighborhood, in which two k-tuples $(v_1, v_2, ..v_i, v_{i+1}, ..v_k)$ and $(v_1, v_2, ..w_i, v_{i+1}, ..v_k)$ are local neighbours only if $w_i \in \mathcal{N}(v_i)$ in the original graph. Thus the updates for both the cases are equivalent. Analogously to the 1-QW case, we see therefore that k-QW $\sqsubset$ to k-δ-LWL.

B.4. Proof of proposition 4.6

One can show (Shiau & Joynt, 2003; Gamble et al., 2010) that for strongly regular graphs, the powers of the adjacency matrix A can be expressed as

$$A^n = \alpha_n I + \beta_n J + \gamma_n A$$

where $\alpha_n, \beta_n, \gamma_n$ only depend on N, k, λ, μ. I is the identity matrix, J is the matrix full of 1s.
The degree matrix D is also equal to kI, then $(D^{-1}A)^n = A^n/k^n$. Hence the information about distance contained in $\mathbf{P}_{uv}$ for strongly regular graphs is the same as in their adjacency matrices. Therefore, for strongly regular graphs, the GD-WL with RRWP test is equivalent to the WL test. The F.I. consists of eigenvalues and eigenvectors. It has been shown that F.I is stronger than 1-WL, but weaker than 2-WL (Rattan & Seppelt). Therefore, GD-WL along with eigenvectors cannot distinguish SRGs, which require 3-WL or higher.

B.5. proof of proposition 4.7

In this proof, we provide the full calculation for

- The average value of the global occupation observable $\langle \hat{n}(t) \rangle_{\mathcal{G}} = \langle \sum_{v \in \mathcal{V}} \hat{n}_v(t) \rangle_{\mathcal{G}}$
- the correlation matrix for the local occupation observables $C_{uv}(t) = \langle \hat{n}_u(t).\hat{n}_v(t) \rangle_{\mathcal{G}} - \langle \hat{n}_u(t) \rangle_{\mathcal{G}} \langle \hat{n}_v(t) \rangle_{\mathcal{G}}$

We provide for each of these cases a formulation involving the corresponding graph quantities. In each of these steps, the goal is to show that it is not a powerful tool to distinguish graphs, and more specifically, that it fails to distinguish non isomorphic strongly regular graphs from the same family.

We start by defining the following quantities :

$$\hat{n}_v = \frac{1}{2}(\mathbb{I} - \hat{Z}_v) \tag{21}$$

and

$$\hat{Z}_v = \bigotimes_{u<v} \mathbb{I} \otimes \hat{Z} \otimes \bigotimes_{u>v} \mathbb{I} \tag{22}$$

We also define

$$\langle \hat{\mathcal{O}} \rangle_{\mathcal{G}} = \langle \psi_{\mathcal{G}} | \hat{\mathcal{O}} | \psi_{\mathcal{G}} \rangle \tag{23}$$

where

$$|\psi_{\mathcal{G}}\rangle = \mathcal{U}_{\mathcal{G}} |\psi_0\rangle = \mathcal{U}_{\mathcal{G}} \bigotimes_{v=1}^{n} |0\rangle \tag{24}$$

For our case, the evolution operator $\mathcal{U}_{\mathcal{G}}$ can be written as :

$$\mathcal{U}_{\mathcal{G}}(\vartheta, t) = \mathrm{e}^{i\hat{\mathcal{H}}_M \vartheta} \mathrm{e}^{-i\hat{\mathcal{H}}_I t} \mathrm{e}^{-i\hat{\mathcal{H}}_M \vartheta} \tag{25}$$

where, for a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ we have

$$\hat{\mathcal{H}}_I = \sum_{(v,u)\in\mathcal{E}} \hat{n}_v \hat{n}_u + \sum_{v\in\mathcal{V}} \hat{n}_v \tag{26}$$

and [9]

$$\hat{\mathcal{H}}_M = \sum_{v\in\mathcal{V}} \hat{Y}_v \tag{27}$$

This corresponds to a single layer Ising evolution, preceded and followed by an evolution through the mixing hamiltonian $\hat{\mathcal{H}}_M$ with opposite signs. The motivation for switching signs in the mixing hamiltonian parameter ϑ is to allow us to write $\mathcal{U}_{\mathcal{G}}(\vartheta, t) = \mathcal{U}_M^\dagger(\vartheta) \cdot \mathcal{U}_I(t) \cdot \mathcal{U}_M(\vartheta)$ which is equal to the identity when $t = 0$.
Now let us compute the quantities $\langle \hat{n}_v \hat{n}_u \rangle_{\mathcal{G}}$ and $\langle \hat{n}_v \rangle_{\mathcal{G}}$

We start by defining the following pulse operator, that will be useful later in this calculation :

$$\hat{P}_\vartheta^\dagger = \exp\left(-\mathbf{i}\,\vartheta \sum_v \hat{Y}_v\right) = \prod_v \underbrace{\left[\cos\theta\,\mathbb{I} + \sin\theta\left(\hat{\sigma}_v^+ - \sigma_v^-\right)\right]}_{\hat{P}_{\vartheta,v}^\dagger} \tag{28}$$

where we introduce the spin operators $\hat{\sigma}^+$ and $\hat{\sigma}^-$, such that

$$\begin{aligned} \hat{\sigma}^+|0\rangle = |1\rangle, \quad \hat{\sigma}^+|1\rangle = 0, \\ \hat{\sigma}^-|0\rangle = 0, \quad \hat{\sigma}^-|1\rangle = |0\rangle, \end{aligned} \tag{29}$$

such that $\hat{X} = (\hat{\sigma}^+ + \hat{\sigma}^-)/2, \hat{Y} = -\mathbf{i}(\hat{\sigma}^+ - \hat{\sigma}^-)/2$ and $\hat{Z} = \hat{\sigma}^+\hat{\sigma}^-$.

B.5.1. State preparation

1. We start with the system in

$$|\mathbf{0}\rangle = \bigotimes_v |0\rangle. \tag{30}$$

2. We then apply a pulse of angle ϑ to get the system in the state

$$|\psi_\vartheta(0)\rangle = \hat{P}_\vartheta^\dagger|\mathbf{0}\rangle = (\cos\theta)^N \sum_{\sigma\in\{0,1\}^N} (\tan\theta)^{n_\sigma}|\sigma\rangle. \tag{31}$$

3. We then let the system evolve with the Ising Hamiltonian, so that

$$|\psi_\vartheta(t)\rangle = \mathrm{e}^{-\mathbf{i}\hat{\mathcal{H}}_I t}|\psi_\vartheta(0)\rangle = (\cos\theta)^N \sum_{\sigma\in\{0,1\}^N} (\tan\theta)^{n_\sigma} \mathrm{e}^{-\mathbf{i}E_\sigma^{(I)}}|\sigma\rangle. \tag{32}$$

4. Finally, the inverse pulse is then applied, to get

$$|\psi_{f,\vartheta}(t)\rangle = \hat{P}_\vartheta|\psi_\vartheta(t)\rangle = (\cos\theta)^N \sum_{\sigma\in\{0,1\}^N} (\tan\theta)^{n_\sigma} \mathrm{e}^{-\mathbf{i}E_\sigma^{(I)}t} \left(\prod_{v=1}^N \hat{P}_{\vartheta,v}\right)|\sigma\rangle. \tag{33}$$

[9] Physically, this describes the same phenomenon as if we had chosen a mixing hamiltonian with operators X_i, up to a gauge change. Which leads to similar results that do not affect the statement in the corresponding proposition.

B.5.2. Computation of the total occupation

We first start by providing a full expression, in terms of graph quantities, for a (graph level) observable that we call the global occupation :

$$\hat{n} = \sum_v \hat{n}_v = \sum_v \hat{\sigma}_v^+ \hat{\sigma}_v^- , \tag{34}$$

Such that, when transposed in the expression provided in equations 33 and 23 we have :

$$\langle \hat{n}(t) \rangle = (\cos\theta)^{2N} \sum_{\sigma,\sigma' \in \{0,1\}^N} \mathrm{e}^{-\mathbf{i}\left(E_\sigma^{(I)} - E_{\sigma'}^{(I)}\right)t} \underbrace{(\tan\theta)^{n_\sigma + n_{\sigma'}} \langle \sigma' | \left(\prod_{v'=1}^N \hat{P}_{\vartheta,v'}^\dagger \right) \hat{n} \left(\prod_{v=1}^N \hat{P}_{\vartheta,v} \right) | \sigma \rangle}_{:= f_{\sigma\sigma'}(\vartheta)} . \tag{35}$$

We then insert an identity $\mathbb{I} = \sum_{\mu \in \{0,1\}^N} |\mu\rangle\langle\mu|$, so that

$$\langle \hat{n}(t) \rangle = (\cos\theta)^{2N} \sum_{\sigma,\sigma' \in \{0,1\}^N} \mathrm{e}^{-\mathbf{i}\left(E_\sigma^{(I)} - E_{\sigma'}^{(I)}\right)t} (\tan\theta)^{n_\sigma + n_{\sigma'}} \sum_{\mu \in \{0,1\}^N} n_\mu \underbrace{\langle \sigma' | \left(\prod_{v'=1}^N \hat{P}_{\vartheta,v'}^\dagger \right) | \mu \rangle \langle \mu | \left(\prod_{v=1}^N \hat{P}_{\vartheta,v} \right) | \sigma \rangle}_{= \prod_{v=1}^N \langle \sigma'_v | \hat{P}_{\vartheta,v}^\dagger | \mu_v \rangle \langle \mu_v | \hat{P}_{\vartheta,v} | \sigma_v \rangle} . \tag{36}$$

Let us now compute an expression for the matrix elements appearing in (36) :

$$\begin{aligned} \lambda_{\mu_v \sigma_v}(\vartheta) &:= \langle \mu_v | \hat{P}_{\vartheta,v} | \sigma_v \rangle \\ &= \cos\theta \, \langle \mu_v | \sigma_v \rangle + \sin\theta \, \langle \mu_v | \hat{\sigma}_v^- | \sigma_v \rangle - \sin\theta \, \langle \mu_v | \hat{\sigma}_v^+ | \sigma_v \rangle \\ &= \cos\theta \, \delta_{\mu_v,\sigma_v} + \sin\theta \, \left(\delta_{\mu_v,\sigma_v - 1} - \delta_{\mu_v,\sigma_v + 1} \right) \end{aligned} \tag{37}$$

The following table gives the possible values for the product $\lambda^*_{\mu_v \sigma'_v}(\vartheta) \lambda_{\mu_v \sigma_v}(\vartheta)$:

$(\mu_v, \sigma_v, \sigma'_v)$	$\lambda^*_{\mu_v \sigma'_v}(\vartheta) \lambda_{\mu_v \sigma_v}(\vartheta)$
$(0,0,0), (1,1,1)$	$\cos\theta^2$
$(0,1,1), (1,0,0)$	$\sin\theta^2$
$(0,1,0), (0,0,1)$	$\cos\theta \sin\theta$
$(1,1,0), (1,0,1)$	$-\cos\theta \sin\theta$

(38)

In order to further explicit (36), we split the set of all vertices into $\mathcal{O}_{\sigma\sigma'} = \{v | \sigma_v = \sigma'_v = 1\}$, $\mathcal{Z}_{\sigma\sigma'} = \{v | \sigma_v = \sigma'_v = 0\}$ and $\Delta_{\sigma\sigma'} = \{v | \sigma_v \neq \sigma'_v\}$, respectively containing n_0, n_1, and $n_\neq$ vertices, so that

$$\prod_{v=1}^N \lambda^*_{\mu_v \sigma'_v}(\vartheta) \lambda_{\mu_v \sigma_v}(\vartheta) = \prod_{v \in \mathcal{O}_{\sigma\sigma'}} \underbrace{\lambda^*_{\mu_v \sigma'_v}(\vartheta) \lambda_{\mu_v \sigma_v}(\vartheta)}_{=(\cos\theta)^{2\mu_v} (\sin\theta)^{2-2\mu_v}} \prod_{v \in \mathcal{Z}_{\sigma\sigma'}} \underbrace{\lambda^*_{\mu_v \sigma'_v}(\vartheta) \lambda_{\mu_v \sigma_v}(\vartheta)}_{=(\cos\theta)^{2-2\mu_v} (\sin\theta)^{2\mu_v}} \prod_{v \in \Delta_{\sigma\sigma'}} \underbrace{\lambda^*_{\mu_v \sigma'_v}(\vartheta) \lambda_{\mu_v \sigma_v}(\vartheta)}_{\cos\theta \sin\theta (-1)^{\mu_v}} . \tag{39}$$

We then split μ in $\mu = \mu^{(1)} \cup \mu^{(0)} \cup \mu^{(\neq)}$, where $\mu^{(1)} = \{\mu_v | v \in \mathcal{O}_{\sigma\sigma'}\}$, $\mu^{(0)} = \{\mu_v | v \in \mathcal{Z}_{\sigma\sigma'}\}$ and $\mu^{(\neq)} = \{\mu_v | v \in \Delta_{\sigma\sigma'}\}$. We can then write

$$f^\mu_{\sigma\sigma'}(\vartheta) = \prod_{v=1}^N \lambda^*_{\mu_v \sigma'_v}(\vartheta) \lambda_{\mu_v \sigma_v}(\vartheta) = \underbrace{(\sin\theta)^{2n_1} (\tan\theta)^{-2n_{\mu^{(1)}}} \times (\cos\theta)^{2n_0} (\tan\theta)^{2n_{\mu^{(0)}}} \times (\cos\theta)^{n_\neq} (\sin\theta)^{n_\neq} (-1)^{n_{\mu^{(\neq)}}}}_{(\cos\theta)^{2N} (\tan\theta)^{n_\neq + 2n_1} (\tan\theta)^{2(n_{\mu^{(0)}} - n_{\mu^{(1)}})} (-1)^{n_{\mu^{(\neq)}}}} .$$

Noting that $\sum_{\mu\in\{0,1\}^m} A(n_\mu) = \sum_{p=0}^{m} \binom{m}{p} A(p)$ we can write

$$\sum_{\mu\in\{0,1\}^N} n_\mu f^\mu_{\sigma\sigma'}(\vartheta) = \underbrace{(\cos\theta)^{2N}(\tan\theta)^{n_{\neq}+2n_1} \underbrace{\sum_{p_1=0}^{n_1}\binom{n_1}{p_1} \underbrace{\sum_{p_0=0}^{n_0}\binom{n_0}{p_0}(\tan\theta)^{2(p_0-p_1)} \underbrace{\sum_{p_{\neq}=0}^{n_{\neq}}\binom{n_{\neq}}{p_{\neq}}(p_1+p_0+p_{\neq})(-1)^{p_{\neq}}}_{=(p_1+p_0)\delta_{0,n_{\neq}}-\delta_{1,n_{\neq}}}}_{=[1+(\tan\theta)^2]^{n_0}\left[\delta_{0,n_{\neq}}\left(p_1+n_0\frac{(\tan\theta)^2}{1+(\tan\theta)^2}\right)-\delta_{1,n_{\neq}}\right]}}_{=[1+(\tan\theta)^2]^{n_0}[1+(\tan\theta)^{-2}]^{n_1}\left\{\delta_{0,n_{\neq}}\left[n_0(\sin\theta)^2+n_1\frac{(\tan\theta)^{-2}}{1+(\tan\theta)^{-2}}\right]-\delta_{1,n_{\neq}}\right\}}}_{(\cos\theta)^{2N}(\tan\theta)^{n_{\neq}}[1+(\tan\theta)^2]^{n_0+n_1}\left\{\delta_{0,n_{\neq}}[n_0(\sin\theta)^2+n_1(\cos\theta)^2]-\delta_{1,n_{\neq}}\right\}}$$

Since $n_0 + n_1 + n_{\neq} = N$,

$$(\cos\theta)^{2N}(\tan\theta)^{n_{\neq}}\left[1+(\tan\theta)^2\right]^{n_0+n_1} = \frac{(\tan\theta)^{n_{\neq}}}{[1+(\tan\theta)^2]^{n_{\neq}}} = (\cos\theta\sin\theta)^{n_{\neq}}, \tag{40}$$

so that

$$\sum_{\mu\in\{0,1\}^N} n_\mu f^\mu_{\sigma\sigma'}(\vartheta) = \delta_{0,n_{\neq}}\left[n_0(\sin\theta)^2 + n_1(\cos\theta)^2\right] - \delta_{1,n_{\neq}}\cos\theta\sin\theta \tag{41}$$

The sum in (36) then only runs on pairs of configuration (σ,σ') that differ by at most 1 element. Let us simplify those terms.

- $n_{\neq} = 0$
  Then $\sigma = \sigma'$, so that $n_\sigma = n_{\sigma'} = n_1 = N - n_0$.

$$f_{\sigma\sigma}(\vartheta) = (\tan\theta)^{2n_\sigma}\left[(N-n_\sigma)(\sin\theta)^2 + n_\sigma(\cos\theta)^2\right] \tag{42}$$

- $n_{\neq} = 1$
  - $n_{\sigma'} = n_\sigma + 1$
    Then $n_1 = n_\sigma = N - n_0 - 1$, so that

$$f_{\sigma\sigma'}(\vartheta) = -(\tan\theta)^{2n_\sigma+1}\cos\theta\sin\theta = -(\tan\theta)^{2n_\sigma}\sin^2\vartheta \tag{43}$$

  - $n_{\sigma'} = n_\sigma - 1$
    Then $n_1 = n_{\sigma'} = N - n_0 - 1$, so that

$$f_{\sigma\sigma'}(\vartheta) = -(\tan\theta)^{2n_\sigma-1}\cos\theta\sin\theta = -(\tan\theta)^{2n_\sigma}\cos^2\vartheta \tag{44}$$

Remark : Exchanging σ and σ' in either of the previous equations (43) or (44) yields the other one, ensuring that $\langle\hat{n}(t)\rangle$ is real. For example, if $n_{\sigma'} = n_\sigma + 1$

$$f_{\sigma'\sigma} = -(\tan\theta)^{2n'_\sigma}\cos^2\vartheta = -(\tan\theta)^{2n_\sigma+2}\cos^2\vartheta = -(\tan\theta)^{2n_\sigma}\sin^2\vartheta = f_{\sigma\sigma'} \tag{45}$$

For a given configuration σ, we note $\Delta_{\sigma,v} = E_\sigma - E_{\sigma'(v)}$ where $\sigma'(v)$ is the configuration obtain from σ by flipping σ_v. The total occupation is then

$$\langle\hat{n}(t)\rangle = \sum_{\sigma\in\{0,1\}^N}\left\{f_{\sigma\sigma}(\vartheta) + \sum_{v\in\mathcal{Z}_\sigma} \mathrm{e}^{-\mathbf{i}\Delta_{\sigma,v}t} f_{\sigma\sigma'(v)}(\vartheta) + \sum_{v\in\mathcal{O}_\sigma} \mathrm{e}^{-\mathbf{i}\Delta_{\sigma,v}t} f_{\sigma\sigma'(v)}(\vartheta)\right\} \tag{46}$$

Let us put this in a form that makes it more explicitly real. We start with

$$\sum_{\sigma\in\{0,1\}^N}\sum_{v\in\mathcal{Z}_\sigma} \mathrm{e}^{-\mathbf{i}\Delta_{\sigma,v}t} f_{\sigma\sigma'(v)}(\vartheta) \underbrace{=}_{\nu=\sigma'(v)} \sum_{\nu\in\{0,1\}^N}\sum_{v\in\mathcal{O}_\nu} \mathrm{e}^{\mathbf{i}\Delta_{\nu,v}t} f_{\nu'(v)\nu}(\vartheta) = \sum_{\nu\in\{0,1\}^N}\sum_{v\in\mathcal{O}_\nu} \mathrm{e}^{\mathbf{i}\Delta_{\nu,v}t} f_{\nu\nu'(v)}(\vartheta). \tag{47}$$

The total occupation becomes

$$\langle\hat{n}(t)\rangle = \sum_{\sigma\in\{0,1\}^N}\left\{ f_{\sigma\sigma}(\vartheta) + 2\sum_{v\in\mathcal{O}_\sigma}\cos\left(\Delta_{\sigma,v}t\right) f_{\sigma\sigma'(v)}(\vartheta)\right\}. \tag{48}$$

The first term can be rewritten as

$$\sum_{\sigma\in\{0,1\}^N} f_{\sigma\sigma}(\vartheta) = (\cos\theta)^{2N}\sum_{p=0}^{N}\binom{N}{p}(\tan\theta)^{2p}\left[(N-p)\sin^2\vartheta + p\cos^2\vartheta\right] = 2N\sin^2\vartheta\cos^2\vartheta \tag{49}$$

The total occupation becomes

$$\langle\hat{n}(t)\rangle = 2N\sin^2\vartheta\cos^2\vartheta - 2(\cos\theta)^{2N+2}\sum_{\sigma\in\{0,1\}^N}(\tan\theta)^{2n_\sigma}\sum_{v\in\mathcal{O}_\sigma}\cos\left(\Delta_{\sigma,v}t\right) \tag{50}$$

Remark : for $n_\sigma = 0$, the term in the sum vanishes $\left(\sum_{v\in\mathcal{O}_\sigma}\cos\left(\Delta_{\sigma,v}t\right) = 0\right)$.

At $t = 0$, this becomes

$$n(0) = 2N\sin^2\vartheta\cos^2\vartheta - 2(\cos\theta)^{2N+2}\sum_{\sigma\in\{0,1\}^N}(\tan\theta)^{2n_\sigma} n_\sigma = 2N\sin^2\vartheta\cos^2\vartheta - 2(\cos\theta)^2 N\frac{\tan^2\vartheta}{1+\tan^2\vartheta} = 0 \tag{51}$$

B.5.3. Expression in terms of graph

For an induced subgraph $\mathcal{G}_\sigma$ of $\mathcal{G}$ and a vertex $v\in\mathcal{V}_\sigma$, we note $\kappa_\sigma(v)$ the degree of v in $\mathcal{G}_\sigma$, and $\kappa(v)$ its degree in $\mathcal{G}$. The energy difference between the Ising configurations corresponding to $\mathcal{G}_\sigma$ and to $\mathcal{G}_\sigma\setminus v$ is $\kappa(v)$. The total occupation can then be expressed as

$$\langle\hat{n}(t)\rangle_{\mathcal{G}} = 2N\sin^2\vartheta\cos^2\vartheta - 2(\cos\theta)^{2N+2}\sum_{\mathcal{G}_\sigma\subset\mathcal{G}}(\tan\theta)^{2|\mathcal{V}_\sigma|}\sum_{v\in\mathcal{V}_\sigma}\cos\left(\kappa_\sigma(v)t\right). \tag{52}$$

We can also express the sum as

$$\sum_{\mathcal{G}_\sigma\subset\mathcal{G}}(\tan\theta)^{2|\mathcal{V}_\sigma|}\sum_{v\in\mathcal{V}_\sigma}\cos\left(\kappa_\sigma(v)t\right) = \sum_{v\in\mathcal{V}}\sum_{\mathcal{V}_\sigma\ni v}(\tan\theta)^{2|\mathcal{V}_\sigma|}\cos\left(\kappa_\sigma(v)t\right). \tag{53}$$

We note $V(v) = \{v'\in\mathcal{V}|(v,v')\in\mathcal{E}\}$ the set of all neighbours of v in $\mathcal{G}$, as well as $V_\sigma(v) = \{v'\in\mathcal{V}_\sigma|(v,v')\in\mathcal{E}_\sigma\}$ the set of all neighbours of v in $\mathcal{G}_\sigma$, so that[10]

$$\sum_{v\in\mathcal{V}}\sum_{\mathcal{V}_\sigma\ni v}(\tan\theta)^{2|\mathcal{V}_\sigma|}\cos\left(\kappa_\sigma(v)t\right) = \sum_{v\in\mathcal{V}}\sum_{\omega=0}^{\kappa(v)}\sum_{\substack{\mathcal{V}_\sigma\ni v|\\ \kappa_\sigma(v)=\omega}}(\tan\theta)^{2|\mathcal{V}_\sigma|}\cos\left(\omega t\right). \tag{54}$$

[10] $|V_\sigma(v)| = \kappa_\sigma(v)$

For a given degree $\omega \leq \kappa(v)$, and a given integer $n \geq 0$, there are $\binom{N-1-\kappa(v)}{n}\binom{\kappa(v)}{\omega}$ induced subgraphs $\mathcal{G}_\sigma$ of size $n+\omega+1$, containing v, and in which v has degree ω. We can then write

$$\sum_{v\in\mathcal{V}}\sum_{\omega=0}^{\kappa(v)}\sum_{\substack{\mathcal{V}_\sigma\ni v\\ \kappa_\sigma(v)=\omega}}(\tan\theta)^{2|\mathcal{V}_\sigma|}\cos(\omega t)=\sum_{v\in\mathcal{V}}\underbrace{\sum_{\omega=0}^{\kappa(v)}\binom{\kappa(v)}{\omega}\cos(\omega t)(\tan\theta)^{2\omega+2}}_{(\tan\theta)^2\Re\left\{(1+\mathrm{e}^{\mathrm{i}t}(\tan\theta)^2)^{\kappa(v)}\right\}}\underbrace{\sum_{n=0}^{N-1-\kappa(v)}\binom{N-1-\kappa(v)}{n}(\tan\theta)^{2n}}_{(1+(\tan\theta)^2)^{N-1-\kappa(v)}} \tag{55}$$

The sum in $\langle\hat{n}(t)\rangle$ then becomes

$$2(\cos\theta)^{2N+2}\sum_{\mathcal{G}_\sigma\subset\mathcal{G}}(\tan\theta)^{2|\mathcal{V}_\sigma|}\sum_{v\in\mathcal{V}_\sigma}\cos(\kappa_\sigma(v)t)=2\sin^2\vartheta\sum_{v\in\mathcal{V}}\frac{\Re\left\{\left(1+\mathrm{e}^{\mathrm{i}t}\tan^2\vartheta\right)^{\kappa(v)}\right\}}{\left(1+\tan^2\vartheta\right)^{1+\kappa(v)}}. \tag{56}$$

If we note $m_{\mathcal{G}}(\kappa)$ the number of vertices of degree κ in $\mathcal{G}$, and $\kappa_{\max}(\mathcal{G})$ the maximum degree of $\mathcal{G}$, we can express the occupation as

$$\langle\hat{n}(t)\rangle_{\mathcal{G}}=2\sin^2\vartheta\cos^2\vartheta\sum_{\kappa=0}^{\kappa_{\max}(\mathcal{G})}m_{\mathcal{G}}(\kappa)\,\Re\left\{1-\left(\cos^2\vartheta+\mathrm{e}^{\mathrm{i}t}\sin^2\vartheta\right)^{\kappa}\right\}. \tag{57}$$

This expression is checked in Fig. 6, on a random graph of 10 vertices.

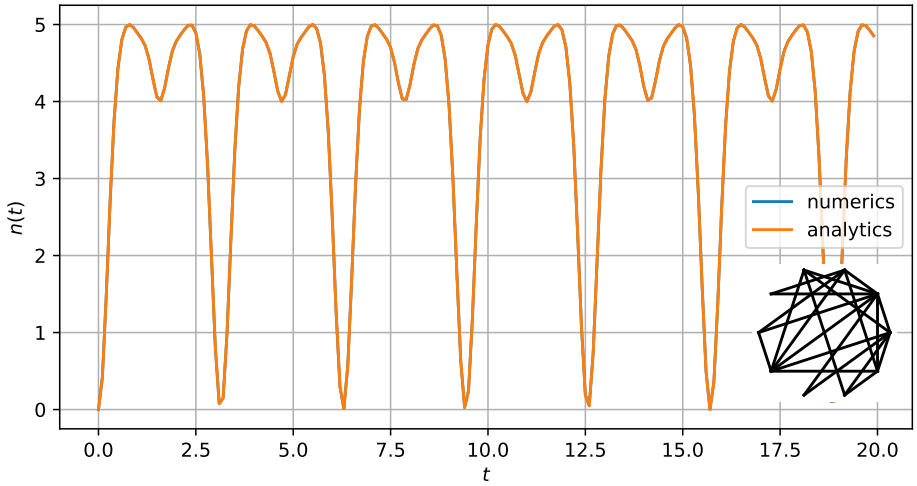

Figure 6. Comparison between (57) and numerical simulation of the dynamics, in the case of a density-density hamiltonian, on the graph shown in the inset. Both curves collapse on top of each other.

Computation for a generic Ising model

We now consider the case of a generic Ising model, defined by the Hamiltonian

$$\hat{\mathcal{H}}_{I_{gen}}=\sum_{(i,j)\in\mathcal{E}}J_{ij}\hat{n}_i\hat{n}_j+\sum_{v\in\mathcal{V}}h_v\hat{n}_v. \tag{58}$$

In this case, (50) becomes

$$\langle\hat{n}(t)\rangle=2N\sin^2\vartheta\cos^2\vartheta-2(\cos\theta)^{2N+2}\sum_{v\in[0,N-1]}\sum_{\substack{\sigma\in\{0,1\}^N\\ \sigma_v=1}}(\tan\theta)^{2n_\sigma}\cos(\Delta_{\sigma,v}t)\,. \tag{59}$$

We note $\mathcal{N}_{\mathcal{G}}(v) = \{v' \in \mathcal{V} | (v, v') \in \mathcal{E}\}$ the neighbourhood of v in $\mathcal{G}$, and split the Ising configuration σ into $\sigma = \underbrace{\{\sigma_{v'} | v' \in \mathcal{N}_{\mathcal{G}}(v)\}}_{\sigma_{\mathcal{N}_{\mathcal{G}}(v)}} \cup \{\sigma_v\} \cup \underbrace{\{\sigma_{v'} | v' \notin \mathcal{N}_{\mathcal{G}}(v) \cup \{v\}\}}_{\bar{\sigma}_{\mathcal{N}_{\mathcal{G}}(v)}}$, so that

$$\sum_{\substack{\sigma \in \mathcal{V} \\ \sigma_v = 1}} (\tan\theta)^{2n_\sigma} \cos(\Delta_{\sigma,v} t) = \sum_{\sigma_{\mathcal{N}_{\mathcal{G}}(v)} \in \{0,1\}^{\kappa(v)}} (\tan\theta)^{2(1+n_{\sigma_{\mathcal{N}_{\mathcal{G}}(v)}})} \cos(\Delta_{\sigma,v} t) \underbrace{\sum_{\sigma_{\mathcal{N}_{\mathcal{G}}(v)} \in \{0,1\}^{N-1-\kappa(v)}} (\tan\theta)^{2n_{\bar{\sigma}_{\mathcal{N}_{\mathcal{G}}(v)}}}}_{(1+\tan^2\vartheta)^{N-\kappa(v)-1}} . \tag{60}$$

We can then write

$$2(\cos\theta)^{2N+2} \sum_{v \in \mathcal{V}} \sum_{\substack{\sigma \in \{0,1\}^N \\ \sigma_v = 1}} (\tan\theta)^{2n_\sigma} \cos(\Delta_{\sigma,v} t) = 2\cos^2\vartheta \sin^2\vartheta \sum_{v \in \mathcal{V}} \sum_{\sigma \in \{0,1\}^{\kappa(v)}} \frac{(\tan\theta)^{2n_\sigma}}{\left(1+\tan^2\vartheta\right)^{\kappa(v)}} \cos(\Delta_{\sigma,v} t) . \tag{61}$$

Finally, we can further simplify the second sum into

$$\sum_{\sigma \in \{0,1\}^{\kappa(v)}} \frac{(\tan\theta)^{2\sum_{i=1}^{\kappa(v)} \sigma_i}}{\left(1+\tan^2\vartheta\right)^{\kappa(v)}} \cos\left[(h_v + \sum_{i=1}^{\kappa(v)} J_{v\tilde{v}_i}\sigma_i)t\right] = \sum_{\kappa=0}^{\kappa(v)} \frac{(\tan\theta)^{2\kappa}}{\left(1+\tan^2\vartheta\right)^{\kappa(v)}} \sum_{\substack{\tilde{v} \in \mathcal{N}_{\mathcal{G}}(v)^\kappa \\ \tilde{v}_i \neq \tilde{v}_j}} \cos\left[(h_v + \sum_{i=1}^{\kappa} J_{v\tilde{v}_i})t\right] . \tag{62}$$

Or,

$$\sum_{\sigma \in \{0,1\}^{\kappa(v)}} \frac{(\tan\theta)^{2\sum_{i=1}^{\kappa(v)} \sigma_i}}{\left(1+\tan^2\vartheta\right)^{\kappa(v)}} \cos\left[(h_v + \sum_{i=1}^{\kappa(v)} J_{v\tilde{v}_i}\sigma_i)t\right] = \Re\left\{ \mathrm{e}^{\mathbf{i}h_v t} \prod_{\tilde{v} \in \mathcal{N}_{\mathcal{G}}(v)} \left(\frac{1+\tan^2\vartheta \mathrm{e}^{\mathbf{i}J_{v\tilde{v}}t}}{1+\tan^2\vartheta} \right) \right\} . \tag{63}$$

The density $\varrho_{\mathcal{G}}(t) = n(t)/N$ then becomes

$$\varrho_{\mathcal{G}}(t) = 2\sin^2\vartheta\cos^2\vartheta \sum_{v \in \mathcal{V}} \Re\left\{ 1 - \mathrm{e}^{\mathbf{i}h_v t} \prod_{\tilde{v} \in \mathcal{N}_{\mathcal{G}}(v)} \left(\cos^2\vartheta + \sin^2\vartheta\, \mathrm{e}^{\mathbf{i}J_{v\tilde{v}}t}\right) \right\} \tag{64}$$

In this form, it is very easy to check that $n(0) = 0$.

More generic pulse

We consider here the case where the system is first subject to a ϑ pulse, then evolves freely for a duration t and finally is subject to a φ pulse. In this case the average number of excitations is

$$\langle \hat{n}(t) \rangle_{\mathcal{G}} = \sin\vartheta\cos\vartheta\sin\varphi\cos\varphi \sum_{v \in \mathcal{V}} \Re\left\{ \frac{\tan\vartheta}{\tan\varphi} + \frac{\tan\varphi}{\tan\vartheta} + 2\, \mathrm{e}^{\mathbf{i}h_v t} \prod_{\tilde{v} \in \mathcal{V}} \left(\cos^2\vartheta + \sin^2\vartheta\, \mathrm{e}^{\mathbf{i}J_{v\tilde{v}}t}\right) \right\} \tag{65}$$

If we consider the case of the uniform local Ising model, *i.e.* $h_v = 1 \forall v \in \mathcal{V}$ and J_{v_1,v_2} is the adjacency matrix of the graph, then we can write :

$$\langle \hat{n}(t) \rangle_{\mathcal{G}} = \sin\vartheta\cos\vartheta\sin\varphi\cos\varphi \sum_{v \in \mathcal{V}} \Re\left\{ \frac{\tan\vartheta}{\tan\varphi} + \frac{\tan\varphi}{\tan\vartheta} + 2\, \mathrm{e}^{\mathbf{i}t} \left(\cos^2\vartheta + \sin^2\vartheta\, \mathrm{e}^{\mathbf{i}t}\right)^{d_v} \right\} \tag{66}$$

where d_v is the degree of node v. We can see that this quantity is not enough to distinguish two graphs with the same degree histograms, which is the case of two non-isomorphic SRGs from the same family. this completes the first part of the proof.

B.5.4. COMPUTATION OF THE LOCAL OBSERVABLES

As we have just shown, the total occupation seems to too simple of an observable to extract meaningful informations from the graph. To this end, we express the expectation value of a generic quadratic function of the *bitstring* μ

$$\varphi(\mu) = \sum_{v\in\mathcal{V}} h_v\mu_v + \sum_{(v,v')\in\mathcal{E}} J_{vv'}\mu_v\mu_{v'}. \tag{67}$$

We start from (36), and replace n_μ by $\varphi(\mu)$

$$\begin{aligned}\langle\varphi\rangle(t) = (\cos\theta)^{2N+3}(\sin\theta)^3 \sum_{\mu\in\{0,1\}^N}\varphi(\mu)\sum_{\sigma,\sigma'\in\{0,1\}^N} \mathrm{e}^{-\mathbf{i}\Delta E_{\sigma\sigma'}t}(\tan\theta)^{n_\sigma+n_{\sigma'}} \\ \times\prod_{v\in\mathcal{O}_{\sigma\sigma'}}(\tan\theta)^{-2\mu_v}\prod_{v\in\mathcal{Z}_{\sigma\sigma'}}(\tan\theta)^{2\mu_v}\prod_{v\in\Delta_{\sigma\sigma'}}(-1)^{\mu_v}.\end{aligned} \tag{68}$$

Linear term

Let us focus on a given linear term and find an expression for $\langle\mu_v\rangle$. We start from (68).

We note $\mathcal{O}_v = \left\{\sigma\in\{0,1\}^N|\sigma_v=1\right\}$ and $\mathcal{Z}_v = \left\{\sigma\in\{0,1\}^N|\sigma_v=0\right\}$ then expand $\sum_{\sigma,\sigma'\in\{0,1\}^N}$ into

The term for $v\in\mathcal{O}_{\sigma\sigma'}$

$$\begin{aligned}\sum_{\mu\in\mathcal{O}_v} f^\mu_{\sigma\sigma'}(\vartheta) &= (\cos\theta)^{2N}(\tan\theta)^{n_{\neq}+2n_1}\sum_{p_1=0}^{\mathbf{n_1-1}}\binom{n_1-1}{p_1}(\tan\theta)^{-2(p_1+1)}\sum_{p_0=0}^{n_0}\binom{n_0}{p_0}(\tan\theta)^{2p_0}\sum_{p_{\neq}=0}^{n_{\neq}}(-1)^{p_{\neq}} \\ &= (\cos\theta)^{2N}\delta_{\sigma\sigma'}(\tan\theta)^{2(n_1-1)}(1+\tan^{-2}\vartheta)^{n_1-1}(1+\tan^2\vartheta)^{n_0} \\ &= (\cos\theta)^{2N}\delta_{\sigma\sigma'}(\tan\theta)^{2(n_1-1)}(\sin\theta)^{-2(n_1-1)}(\cos\theta)^{-2n_0} \\ &= \delta_{\sigma\sigma'}\cos^2\vartheta\end{aligned} \tag{69}$$

The term for $v\in\mathcal{Z}_{\sigma\sigma'}$

$$\sum_{\mu\in\mathcal{O}_v} f^\mu_{\sigma\sigma'}(\vartheta) = (\cos\theta)^{2N}(\tan\theta)^{n_{\neq}+2n_1}\sum_{p_1=0}^{n_1}\binom{n_1}{p_1}(\tan\theta)^{-2p_1)} \quad = \delta_{\sigma\sigma'}\sin^2\vartheta \tag{70}$$

The term for $v\in\Delta_{\sigma\sigma'}$

$$\begin{aligned}\sum_{\mu\in\mathcal{O}_v} f^\mu_{\sigma\sigma'}(\vartheta) &= (\cos\theta)^{2N}(\tan\theta)^{n_{\neq}+2n_1}\sum_{p_1=0}^{n_1}\binom{n_1}{p_1}(\tan\theta)^{-2p_1}\sum_{p_0=0}^{n_0}\binom{n_0}{p_0}(\tan\theta)^{2p_0}\sum_{p_{\neq}=0}^{n_{\neq}-1}(-1)^{p_{\neq}+1} \\ &= -(\cos\theta)^{2N}\delta_{n_{\neq}-1}(\tan\theta)^{1+2n_1}(\sin\theta)^{-2n_1}(\cos\theta)^{-2n_0} \\ &= -\delta_{n_{\neq}-1}\sin\vartheta\cos\vartheta\end{aligned} \tag{71}$$

The sum of all terms : $\langle n_v(t)\rangle_{\mathcal{G}}$

$$\begin{aligned}\langle n_v(t)\rangle_{\mathcal{G}} = (\cos\theta)^{2N}\Bigg[&\cos^2\vartheta\sum_{\sigma\in\mathcal{O}_v}\tan^{2n_\sigma}\vartheta + \sin^2\vartheta\sum_{\sigma\in\mathcal{Z}_v}\tan^{2n_\sigma}\vartheta \\ &-\sin\vartheta\cos\vartheta\sum_{\sigma\in\mathcal{O}_v}(\tan\theta)^{2n_\sigma-1}\mathrm{e}^{\mathbf{i}\Delta E_{\sigma\tilde{\sigma}^v}t} - \sin\vartheta\cos\vartheta\mathrm{e}^{\mathbf{i}h_v t}\sum_{\sigma\in\mathcal{Z}_v}(\tan\theta)^{2n_\sigma+1}\mathrm{e}^{-\mathbf{i}\Delta E_{\sigma\tilde{\sigma}^v}t}\Bigg]\end{aligned} \tag{72}$$

We have introduced here $\tilde{\sigma}^v$ the configuration obtained from σ by flipping σ_v.

We focus on the first two terms first.

$$\begin{aligned}\cos^2\vartheta \sum_{\sigma\in\mathcal{O}_v} \tan^{2n_\sigma}\vartheta + \sin^2\vartheta \sum_{\sigma\in\mathcal{Z}_v} \tan^{2n_\sigma}\vartheta &= \cos^2\vartheta \sum_{n=0}^{N-1}\binom{N-1}{n}\tan^{2(n+1)}\vartheta + \sin^2\vartheta\sum_{n=0}^{N-1}\binom{N-1}{n}\tan^{2n}\vartheta \\ &= 2\sin^2\vartheta\sum_{n=0}^{N-1}\binom{N-1}{n}\tan^{2n}\vartheta \\ &= 2\sin^2\vartheta\cos^2\vartheta(\cos\theta)^{-2N}.\end{aligned} \tag{73}$$

And now, the third term, using (58)

$$\begin{aligned}\cos^2\vartheta\sum_{\sigma\in\mathcal{O}_v}(\tan\theta)^{2n_\sigma}\mathrm{e}^{\mathbf{i}\Delta E_{\sigma\tilde{\sigma}^v}t} &= \sin^2\vartheta\,\mathrm{e}^{\mathbf{i}h_v t}\prod_{\tilde{v}\in\mathcal{N}_\mathcal{G}(v)}\left(1+\tan^2\vartheta\,\mathrm{e}^{\mathbf{i}J_{\tilde{v}}t}\right)\sum_{n=0}^{N-1-\kappa(v)}\binom{N-1-\kappa(v)}{n}(\tan\theta)^{2n} \\ &= \sin^2\vartheta\,\mathrm{e}^{\mathbf{i}h_v t}\prod_{\tilde{v}\in\mathcal{N}_\mathcal{G}(v)}\left(1+\tan^2\vartheta\,\mathrm{e}^{\mathbf{i}J_{v\tilde{v}}t}\right)(\cos\theta)^{2(1+\kappa(v)-N)} \\ &= \sin^2\vartheta\cos^2\vartheta(\cos\theta)^{-2N}\,\mathrm{e}^{\mathbf{i}h_v t}\prod_{\tilde{v}\in\mathcal{N}_\mathcal{G}(v)}\left(\cos^2\vartheta+\sin^2\vartheta\,\mathrm{e}^{\mathbf{i}J_{v\tilde{v}}t}\right).\end{aligned} \tag{74}$$

Similarly, the fourth terms is

$$\begin{aligned}\sin^2\vartheta\sum_{\sigma\in\mathcal{Z}_v}(\tan\theta)^{2n_\sigma}\mathrm{e}^{-\mathbf{i}\Delta E_{\sigma\tilde{\sigma}^v}t} &= \sin^2\vartheta\,\mathrm{e}^{-\mathbf{i}h_v t}\prod_{\tilde{v}\in\mathcal{N}_\mathcal{G}(v)}\left(1+\tan^2\vartheta\,\mathrm{e}^{-\mathbf{i}J_{\tilde{v}}t}\right)\sum_{n=0}^{N-1-\kappa(v)}\binom{N-1-\kappa(v)}{n}(\tan\theta)^{2n} \\ &= \sin^2\vartheta\,\mathrm{e}^{-\mathbf{i}h_v t}\prod_{\tilde{v}\in\mathcal{N}_\mathcal{G}(v)}\left(1+\tan^2\vartheta\,\mathrm{e}^{-\mathbf{i}J_{v\tilde{v}}t}\right)(\cos\theta)^{2(1+\kappa(v)-N)} \\ &= \sin^2\vartheta\cos^2\vartheta(\cos\theta)^{-2N}\,\mathrm{e}^{-\mathbf{i}h_v t}\prod_{\tilde{v}\in\mathcal{N}_\mathcal{G}(v)}\left(\cos^2\vartheta+\sin^2\vartheta\,\mathrm{e}^{-\mathbf{i}J_{v\tilde{v}}t}\right).\end{aligned} \tag{75}$$

Adding all those terms together, we get

$$\langle n_v(t)\rangle_\mathcal{G} = 2\sin^2\vartheta\cos^2\vartheta\,\Re\left\{1-\mathrm{e}^{\mathbf{i}h_v t}\prod_{\tilde{v}\in\mathcal{N}_\mathcal{G}(v)}\left(\cos^2\vartheta+\sin^2\vartheta\,\mathrm{e}^{\mathbf{i}J_{v\tilde{v}}t}\right)\right\} \tag{76}$$

Quadratic term

We no focus on a given quadratic term and find an expression for $\langle\mu_{v_1}\mu_{v_2}\rangle$, where $v_1\neq v_2$. We start from again from (68). $v_1, v_2\in\mathcal{O}_{\sigma\sigma'}$

$$\begin{aligned}\sum_{\mu\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}} f^\mu_{\sigma\sigma'}(\vartheta) &= (\cos\theta)^{2N}(\tan\theta)^{n_{\neq}+2n_1}\sum_{p_1=0}^{\mathbf{n_1-2}}\binom{n_1-2}{p_1}(\tan\theta)^{-2(p_1+2)}\sum_{p_0=0}^{n_0}\binom{n_0}{p_0}(\tan\theta)^{2p_0}\sum_{p_{\neq}=0}^{n_{\neq}}(-1)^{p_{\neq}} \\ &= \cos^4\vartheta\,\delta_{\sigma\sigma'}\end{aligned} \tag{77}$$

Note that here $\forall v, \sigma_v = \sigma'_v$, and

$$(\tan\theta)^{n_\sigma+n_{\sigma'}}\sum_{\mu\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}} f^\mu_{\sigma\sigma'}(\vartheta) = \cos^4\vartheta(\tan\theta)^{2n_\sigma}\,\delta_{\sigma\sigma'} \tag{78}$$

$v_1 \in \mathcal{O}_{\sigma\sigma'}, v_2 \in \mathcal{Z}_{\sigma\sigma'}$

$$\begin{aligned}\sum_{\mu\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}} f^{\mu}_{\sigma\sigma'}(\vartheta) &= (\cos\theta)^{2N}(\tan\theta)^{n_{\neq}+2n_1}\sum_{p_1=0}^{\mathbf{n_1-1}}\binom{n_1-1}{p_1}(\tan\theta)^{-2(p_1+1)}\sum_{p_0=0}^{\mathbf{n_0-1}}\binom{n_0-1}{p_0}(\tan\theta)^{2(p_0+1)}\sum_{p_{\neq}=0}^{n_{\neq}}(-1)^{p_{\neq}}\\ &= \sin^2\vartheta\cos^2\vartheta\,\delta_{\sigma\sigma'}\end{aligned} \tag{79}$$

Note that here $\forall v, \sigma_v = \sigma'_v$, and

$$(\tan\theta)^{n_\sigma+n_{\sigma'}}\sum_{\mu\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}} f^{\mu}_{\sigma\sigma'}(\vartheta) = \cos^4\vartheta(\tan\theta)^{2n_\sigma+1}\,\delta_{\sigma\sigma'} \tag{80}$$

$v_1 \in \mathcal{O}_{\sigma\sigma'}, v_2 \in \Delta_{\sigma\sigma'}$

$$\begin{aligned}\sum_{\mu\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}} f^{\mu}_{\sigma\sigma'}(\vartheta) &= (\cos\theta)^{2N}(\tan\theta)^{n_{\neq}+2n_1}\sum_{p_1=0}^{\mathbf{n_1-1}}\binom{n_1-1}{p_1}(\tan\theta)^{-2(p_1+1)}\sum_{p_0=0}^{n_0}\binom{n_0}{p_0}(\tan\theta)^{2p_0}\sum_{p_{\neq}=0}^{\mathbf{n_{\neq}-1}}(-1)^{p_{\neq}+1}\\ &= -\sin\theta\cos^3\vartheta\,\delta_{n_{\neq}-1}\end{aligned} \tag{81}$$

Note that here $\forall v \neq v_2, \sigma_v = \sigma'_v$, and $\sigma_{v_2} = 1-\sigma'_{v_2}$, and

$$(\tan\theta)^{n_\sigma+n_{\sigma'}}\sum_{\mu\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}} f^{\mu}_{\sigma\sigma'}(\vartheta) = -\cos^4\vartheta(\tan\theta)^{2(n_\sigma-\sigma_{v_2})}\,\delta_{n_{\neq}-1}\,\bar{\delta}_{\sigma_{v_2}\sigma'_{v_2}} \tag{82}$$

$v_1, v_2 \in \mathcal{Z}_{\sigma\sigma'}$

$$\begin{aligned}\sum_{\mu\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}} f^{\mu}_{\sigma\sigma'}(\vartheta) &= (\cos\theta)^{2N}(\tan\theta)^{n_{\neq}+2n_1}\sum_{p_1=0}^{n_1}\binom{n_1}{p_1}(\tan\theta)^{-2p_1}\sum_{p_0=0}^{\mathbf{n_0-2}}\binom{n_0-2}{p_0}(\tan\theta)^{2(p_0+2)}\sum_{p_{\neq}=0}^{n_{\neq}}(-1)^{p_{\neq}}\\ &= \sin^4\vartheta\,\delta_{\sigma\sigma'}\end{aligned} \tag{83}$$

Note that here $\forall v, \sigma_v = \sigma'_v$, and

$$(\tan\theta)^{n_\sigma+n_{\sigma'}}\sum_{\mu\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}} f^{\mu}_{\sigma\sigma'}(\vartheta) = \sin^4\vartheta(\tan\theta)^{2n_\sigma}\,\delta_{\sigma\sigma'} = \cos^4\vartheta(\tan\theta)^{2n_\sigma+4}\,\delta_{\sigma\sigma'} \tag{84}$$

$v_1 \in \mathcal{Z}_{\sigma\sigma'}, v_2 \in \Delta_{\sigma\sigma'}$

$$\begin{aligned}\sum_{\mu\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}} f^{\mu}_{\sigma\sigma'}(\vartheta) &= (\cos\theta)^{2N}(\tan\theta)^{n_{\neq}+2n_1}\sum_{p_1=0}^{n_1}\binom{n_1}{p_1}(\tan\theta)^{-2p_1}\sum_{p_0=0}^{\mathbf{n_0-1}}\binom{n_0-1}{p_0}(\tan\theta)^{2(p_0+1)}\sum_{p_{\neq}=0}^{\mathbf{n_{\neq}-1}}(-1)^{p_{\neq}+1}\\ &= -\sin^3\vartheta\cos\theta\,\delta_{n_{\neq}-1}\end{aligned} \tag{85}$$

Note that here $\forall v \neq v_2, \sigma_v = \sigma'_v$, and $\sigma_{v_2} = 1-\sigma'_{v_2}$, and

$$\begin{aligned}(\tan\theta)^{n_\sigma+n_{\sigma'}}\sum_{\mu\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}} f^{\mu}_{\sigma\sigma'}(\vartheta) &= -\sin^4\vartheta(\tan\theta)^{2(n_\sigma-\sigma_{v_2})}\,\delta_{n_{\neq}-1}\,\bar{\delta}_{\sigma_{v_2}\sigma'_{v_2}}\\ &= -\cos^4\vartheta(\tan\theta)^{2(n_\sigma+2-\sigma_{v_2})}\,\delta_{n_{\neq}-1}\,\bar{\delta}_{\sigma_{v_2}\sigma'_{v_2}}\end{aligned} \tag{86}$$

$v_1, v_2 \in \Delta_{\sigma\sigma'}$

$$\sum_{\mu\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}} f^{\mu}_{\sigma\sigma'}(\vartheta) = (\cos\theta)^{2N}(\tan\theta)^{n_{\neq}+2n_1}\sum_{p_1=0}^{n_1}\binom{n_1}{p_1}(\tan\theta)^{-2p_1}\sum_{p_0=0}^{n_0}\binom{n_0}{p_0}(\tan\theta)^{2p_0}\sum_{p_{\neq}=0}^{\mathbf{n}_{\neq}-\mathbf{2}}(-1)^{p_{\neq}+2}$$
$$= \sin^2\vartheta\cos^2\vartheta\,\delta_{n_{\neq}-2} \tag{87}$$

Note that here $\forall v \neq v_2, v_1, \sigma_v = \sigma'_v$, and $\sigma_{v_1} = 1-\sigma'_{v_1}, \sigma_{v_2} = 1-\sigma'_{v_2}$, and

$$(\tan\theta)^{n_\sigma+n_{\sigma'}}\sum_{\mu\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}} f^{\mu}_{\sigma\sigma'}(\vartheta) = \cos^4\vartheta(\tan\theta)^{2(n_\sigma-\sigma_{v_2}-\sigma_{v_2})}\,\delta_{n_{\neq}-2}\,\bar{\delta}_{\sigma_{v_2}\sigma'_{v_2}}\,\bar{\delta}_{\sigma_{v_1}\sigma'_{v_1}} \tag{88}$$

$\langle n_{v_1}n_{v_2}(t)\rangle_{\mathcal{G}}$

In the following, we will note $\tilde{\sigma}^i$ the configuration obtained from σ by flipping σ_{v_i}, and configuration obtained from σ by flipping both σ_{v_1} and σ_{v_2}. We then split the sum $\sum_{\sigma\in\{0,1\}^N}$ into four terms

$$\sum_{\sigma\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}} + \sum_{\sigma\in\mathcal{O}_{v_1}\cap\mathcal{Z}_{v_2}} + \sum_{\sigma\in\mathcal{Z}_{v_1}\cap\mathcal{O}_{v_2}} + \sum_{\sigma\in\mathcal{Z}_{v_1}\cap\mathcal{Z}_{v_2}} := \mathcal{S}_{11}+\mathcal{S}_{10}+\mathcal{S}_{01}+\mathcal{S}_{00}, \tag{89}$$

and focus on each of them separately.

$$\begin{aligned}\mathcal{S}_{11} &= \sum_{\sigma\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}}\sum_{\sigma'\in\{0,1\}^N}\mathrm{e}^{-\mathbf{i}\Delta E_{\sigma,\sigma'}t}(\tan\theta)^{n_\sigma+n_{\sigma'}}\sum_{\mu\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}} f^{\mu}_{\sigma\sigma'}(\vartheta)\\ &= \cos^4\vartheta\sum_{n=0}^{N-2}\binom{N-2}{n}(\tan\theta)^{2(n+2)}\\ &- \cos^4\vartheta\sum_{\sigma\in\mathcal{O}_{v_1}\cap\mathcal{O}_{v_2}}(\tan\theta)^{2(n_\sigma-1)}\left(\mathrm{e}^{-\mathbf{i}\Delta E_{\sigma,\tilde{\sigma}^1}t}+\mathrm{e}^{-\mathbf{i}\Delta E_{\sigma,\tilde{\sigma}^2}t}\right)\end{aligned} \tag{90}$$

We introduce $\mathcal{N}_{12} = \mathcal{N}_{\mathcal{G}}(v_1)\cap\mathcal{N}_{\mathcal{G}}(v_2)$, and $\mathcal{N}_1 = \mathcal{N}_{\mathcal{G}}(v_1)\backslash\mathcal{N}_{12}\backslash v_2$ (and $1\leftrightarrow 2$). Furthermore, we also introduce

$$w_\vartheta(\phi,t) = \left(\cos^2\vartheta+\sin^2\vartheta\,\mathrm{e}^{-\mathbf{i}\phi t}\right) \tag{91}$$

If $(v_1,v_2)\in\mathcal{E}$, then

$$\begin{aligned}\mathcal{S}_{11} = (\cos\theta)^{-2N}\sin^2\vartheta\cos^6\vartheta\Bigg[&\tan^2\vartheta - \mathrm{e}^{-\mathbf{i}h_{v_1}t}\prod_{\tilde{v}\in\mathcal{N}\backslash v_2} w_\vartheta(J_{v_1\tilde{v}},t) - \mathrm{e}^{-\mathbf{i}h_{v_2}t}\prod_{\tilde{v}\in\mathcal{N}\backslash v_1} w_\vartheta(J_{v_2\tilde{v}},t)\\ &+\tan^2\vartheta\,\mathrm{e}^{-\mathbf{i}(h_{v_1}+h_{v_2}+J_{v_1v_2})t}\prod_{\tilde{v}\in\mathcal{N}_1} w_\vartheta(J_{v_1\tilde{v}},t)\prod_{\tilde{v}\in\mathcal{N}_2} w_\vartheta(J_{v_2\tilde{v}},t)\prod_{\tilde{v}\in\mathcal{N}_{12}} w_\vartheta(J_{v_1\tilde{v}}+J_{v_2\tilde{v}},t)\Bigg]\end{aligned} \tag{92}$$

The generic expression is

$$\begin{aligned}\mathcal{S}_{11} = (\cos\theta)^{-2N}\sin^2\vartheta\cos^6\vartheta\Bigg[&\tan^2\vartheta - \mathrm{e}^{-\mathbf{i}h_{v_1}t}\prod_{\tilde{v}\neq v_2} w_\vartheta(J_{v_1\tilde{v}},t) - \mathrm{e}^{-\mathbf{i}h_{v_2}t}\prod_{\tilde{v}\neq v_1} w_\vartheta(J_{v_2\tilde{v}},t)\\ &+\tan^2\vartheta\,\mathrm{e}^{-\mathbf{i}(h_{v_1}+h_{v_2}+J_{v_1v_2})t}\prod_{\tilde{v}\neq v_1,v_2} w_\vartheta(J_{v_1\tilde{v}}+J_{v_2\tilde{v}},t)\Bigg]\end{aligned} \tag{93}$$

$$\begin{aligned}\mathcal{S}_{10} = (\cos\theta)^{-2N}\sin^2\vartheta\cos^6\vartheta\Bigg[&\tan\vartheta - \mathrm{e}^{\mathbf{i}h_{v_2}t}\prod_{\tilde{v}\neq v_1} w_\vartheta(-J_{v_2\tilde{v}},t)\\ &-\tan^{-1}\vartheta\mathrm{e}^{-\mathbf{i}h_{v_1}t}\prod_{\tilde{v}\neq v_2} w_\vartheta(J_{v_1\tilde{v}},t)\\ &-\tan^{-2}\vartheta\mathrm{e}^{-\mathbf{i}h_{v_1}t}\prod_{\tilde{v}\neq v_2} w_\vartheta(J_{v_1\tilde{v}},t)\\ &+\tan^2\vartheta\,\mathrm{e}^{-\mathbf{i}(h_{v_1}+h_{v_2}+J_{v_1v_2})t}\prod_{\tilde{v}\neq v_1,v_2} w_\vartheta(J_{v_1\tilde{v}}+J_{v_2\tilde{v}},t)\Bigg]\end{aligned} \tag{94}$$

$$\begin{aligned}\langle n_{v_1}n_{v_2}(t)\rangle_{\mathcal{G}} = 4\sin^4\vartheta\cos^4\vartheta\;\Re\Bigg\{&1-\mathrm{e}^{\mathbf{i}h_{v_1}t}\prod_{\tilde{v}\neq v_2}\left(\cos^2\vartheta+\sin^2\vartheta\;\mathrm{e}^{\mathbf{i}J_{v_1\tilde{v}}t}\right)-\mathrm{e}^{\mathbf{i}h_{v_2}t}\prod_{\tilde{v}\neq v_1}\left(\cos^2\vartheta+\sin^2\vartheta\;\mathrm{e}^{\mathbf{i}J_{v_2\tilde{v}}t}\right)\\ &+\frac{1}{2}\mathrm{e}^{\mathbf{i}(h_{v_1}+h_{v_2}+J_{v_1v_2})t}\prod_{\tilde{v}\neq v_1,v_2}\left(\cos^2\vartheta+\sin^2\vartheta\;\mathrm{e}^{\mathbf{i}(J_{v_1\tilde{v}}+J_{v_2\tilde{v}})t}\right)\\ &+\frac{1}{2}\mathrm{e}^{\mathbf{i}(h_{v_1}-h_{v_2})t}\prod_{\tilde{v}\neq v_1,v_2}\left(\cos^2\vartheta+\sin^2\vartheta\;\mathrm{e}^{\mathbf{i}(J_{v_1\tilde{v}}-J_{v_2\tilde{v}})t}\right)\Bigg\}\end{aligned} \tag{95}$$

$\langle n_{v_1}n_{v_2}(t)\rangle_{\mathcal{G}} - \langle n_{v_1}(t)\rangle_{\mathcal{G}}\langle n_{v_2}(t)\rangle_{\mathcal{G}}$

In order to emphasise the structure of the expression, we introduce

$$w_{v\tilde{v}}(\vartheta,t) = \left(\cos^2\vartheta+\sin^2\vartheta\;\mathrm{e}^{\mathbf{i}J_{v\tilde{v}}t}\right) \tag{96}$$

and

$$\varrho_{\vartheta,v}(t) = \mathrm{e}^{\mathbf{i}h_{v_1}t}\prod_{\tilde{v}}\left(\cos^2\vartheta+\sin^2\vartheta\;\mathrm{e}^{\mathbf{i}J_{v\tilde{v}}t}\right) = \mathrm{e}^{\mathbf{i}h_{v_1}t}\prod_{\tilde{v}} w_{v\tilde{v}}(\vartheta,t), \tag{97}$$

so that

$$\langle n_v(t)\rangle_{\mathcal{G}} = 2\sin^2\vartheta\cos^2\vartheta\;\Re\left\{1-\varrho_{\vartheta,v}(t)\right\} = 2\sin^2\vartheta\cos^2\vartheta\left[1-\frac{1}{2}\left(\varrho_{\vartheta,v}(t)+\varrho^*_{\vartheta,v}(t)\right)\right], \tag{98}$$

and

$$\langle n_{v_1}n_{v_2}(t)\rangle_{\mathcal{G}} = 4\sin^4\vartheta\cos^4\vartheta\;\Re\left\{1-\left[\varrho_{\vartheta,v_1}(t)+\varrho_{\vartheta,v_2}(t)\right]+\frac{1}{2}\left[\varrho_{\vartheta,v_1}(t)\varrho_{\vartheta,v_2}(t)+\varrho_{\vartheta,v_1}(t)\varrho^*_{\vartheta,v_2}(t)\right]\right\}. \tag{99}$$

Similarly,

$$\begin{aligned}\langle n_{v_1} n_{v_2}(t)\rangle_{\mathcal{G}} = 4\sin^4\vartheta\cos^4\vartheta\ \Re\Bigg\{1 - \frac{\varrho_{\vartheta,v_1}(t)+\varrho_{\vartheta,v_2}(t)}{w_{v_1v_2}(\vartheta,t)} \\ + \frac{1}{2}\mathrm{e}^{\mathbf{i}J_{v_1v_2}t}\frac{\varrho_{\vartheta,v_1}(t)\varrho_{\vartheta,v_2}(t)}{\prod_{\tilde{v}\in\mathcal{N}_{12}\cup\{v_1,v_2\}}\cos^2\vartheta+\sin^2\vartheta\ \mathrm{e}^{\mathbf{i}(J_{v_1\tilde{v}}+J_{v_2\tilde{v}})t}} \\ + \frac{1}{2}\frac{\varrho_{\vartheta,v_1}(t)\varrho^*_{\vartheta,v_2}(t)}{\prod_{\tilde{v}\in\mathcal{N}_{12}\cup\{v_1,v_2\}}\cos^2\vartheta+\sin^2\vartheta\ \mathrm{e}^{\mathbf{i}(J_{v_1\tilde{v}}-J_{v_2\tilde{v}})t}}\Bigg\}.\end{aligned} \tag{100}$$

Introducing $w^{\pm}_{v1v_2}(\vartheta,t) = \prod_{\tilde{v}\in\mathcal{N}_{12}\cup\{v_1,v_2\}}\cos^2\vartheta+\sin^2\vartheta\ \mathrm{e}^{\mathbf{i}(J_{v_1\tilde{v}}\pm J_{v_2\tilde{v}})t}$, the correlation between densities at v_1 and v_2 can then be expressed as

$$\begin{aligned}\langle n_{v_1} n_{v_2}(t)\rangle_{\mathcal{G}} - \langle n_{v_1}(t)\rangle_{\mathcal{G}}\langle n_{v_2}(t)\rangle_{\mathcal{G}} = 4\sin^4\vartheta\cos^4\vartheta\ \Re\Big\{[\varrho_{\vartheta,v_1}(t)+\varrho_{\vartheta,v_2}(t)]\left[1-w_{v_1v_2}(\vartheta,t)^{-1}\right] \\ + \frac{1}{2}\left[1-\mathrm{e}^{\mathbf{i}J_{v_1v_2}t}w^{+}_{v_1v_2}(\vartheta,t)^{-1}\right]\varrho_{\vartheta,v_1}(t)\varrho_{\vartheta,v_2}(t) \\ + \frac{1}{2}\left[1-w^{-}_{v_1v_2}(\vartheta,t)^{-1}\right]\varrho_{\vartheta,v_1}(t)\varrho^*_{\vartheta,v_2}(t)\Big\}.\end{aligned} \tag{101}$$

Since we consider only the uniform Ising model for this proof, we have $h_v = 1\ \forall v\in\mathcal{V}$ and $J_{v_1,v_2} = 1$ if $(v_1,v_2)\in\mathcal{E}$ and 0 if not. This allows us to rewrite the elements recovered in the generic expression derived in equation 101. furthermore, if we write $a = (\cos^2\vartheta+\sin^2\vartheta e^{\mathbf{i}t})$ and $b = (\cos^2\vartheta+\sin^2\vartheta e^{2\mathbf{i}t})$ we can rewrite the previous quantities in terms of SRGs invariants (ν,k,λ,μ) as :

$$\varrho_{\vartheta,v}(t) = \mathrm{e}^{\mathbf{i}t}a^k \tag{102}$$

$$w^{+}_{v_1,v_2}(\vartheta,t) = \begin{cases} a^2.b^{\lambda}, & \text{if } (v_1,v_2)\in\mathcal{E} \\ b^{\mu} & \text{otherwise}\end{cases} \tag{103}$$

$$w^{-}_{v_1,v_2}(\vartheta,t) = \begin{cases} |a|^2, & \text{if } (v_1,v_2)\in\mathcal{E} \\ 1 & \text{otherwise}\end{cases} \tag{104}$$

this allows us to write the correlation matrix C as :

$$C = c_{k,\lambda}A + c_{k,\mu}(J-A) \tag{105}$$

with $c_{k,\lambda}$ & $c_{k,\mu}\in\mathbb{R}$ that only depend on the invariants, A the adjacency matrix of the graph, and J a matrix where all the entries are ones. It is then easy to verify that using this formula, invariant functions (ex : the distance introduced in section 4.2) are not enough to distinguish non isomorphic SRGs from the same family, neither can the GD-WL which is in this case equivalent to the 1-WL, and we know that the latter is not powerful enough to distinguish SRGs *c.f* proposition 4.5.

C. Experiments

C.1. Experiments on quantum random walk

In the same way as (Ma et al., 2023), we perform the experiments on the standard train/val/test splits. For each dataset, we perform 4 runs with the seeds 0, 1, 2, 3 and display the average of the scores and the standard deviation.

We do not perform an extensive hyperparameter search, and we only compute ourselves the GRIT model. We take the same hyperparameters as (Ma et al., 2023) that we remind in table 2.

Table 2. Hyperparameters for GRIT model five datasets from BenchmarkingGNNs (Dwivedi et al., 2020), ZINC-full (Irwin et al., 2012) and (Hu et al., 2021)

Hyperparameter	ZINC/ZINC-full	MNIST	CIFAR10	PATTERN	CLUSTER	PCQM4Mv2
# Transformer Layers	10	3	3	10	16	16
Hidden dim	64	52	52	64	48	256
# Heads	8	4	4	8	8	8
Dropout	0	0	0	0	0.01	0.1
Attention dropout	0.2	0.5	0.5	0.2	0.5	0.1
Graph pooling	sum	mean	mean	–	–	mean
PE dim (RW-steps)	21	18	18	21	32	16
PE encoder	linear	linear	linear	linear	linear	linear
QPE dim (1CQRW steps)	20	18	18	20	32	16
Max duration	π	π	π	π	π	π
Min duration	0.1	0.1	0.1	0.1	0.1	0.1
Initial distribution	local	local	local	local	local	local
QPE dim (2QiRW steps)	20	18	18	20	32	16
Initial distribution	adjacency	adjacency	adjacency	adjacency	adjacency	adjacency
Batch size	32/256	16	16	32	16	256
Learning Rate	0.001	0.001	0.001	0.0005	0.0005	0.0002
# Epochs	2000	200	200	100	100	150
# Warmup epochs	50	5	5	5	5	10
Weight decay	$1e-5$	$1e-5$	$1e-5$	$1e-5$	$1e-5$	0
# Parameters GRIT	473,473	102,138	99486	477,953	432,206	11.8M
# Parameters 2QiRW GRIT	476,033	104,010	101,358	480,513	434,742	11.8M

Table 3. Hyperparameters for non transformer base models large scale datasets , ZINC-full (Irwin et al., 2012) and PCQM4Mv2 (Hu et al., 2021). Each entry has to be read as the values for ZINC-full/PCQM4Mv2, when there is a single entry, the value is the same for both datasets. * : same as the first column. - : non applicable.

Hyperparameter	GINE	GINE-big	GatedGCN	GatedGCN-big
# Layers	5/3	*	*	*
Hidden dim	128	256	128	256
Dropout	0	*	*	*
Aggregation	mean	*	*	*
# Layers MLP postprocessing	3	*	*	*
PE encoder	linear	*	*	*
PE dim (RRWP)	21	40	21	40
PE dim (LE)	32	*	*	*
PE dim (Q)	21	40	20	40
Initial distribution	adjacency	*	*	*
PE dim RRWP(RRWP+Q)	20	*	*	*
PE dim Q(RRWP+Q)	20	*	*	*
Batch size	256	*	*	*
Learning Rate	$0.001/0.0002$	*	*	*
# Epochs	2000/150	*	*	*
# Warmup epochs	50/10	*	*	*
Weight decay	$1e-5/0$	*	*	*

C.2. Experiments on synthetic datasets

We train the GCN model for 200 epochs using the Adam optimizer, 0.001 learning rate, no weight decay. We split randomly the dataset on train/validation/test with a proportion 0.8/0.1/0.1, and we measure the test accuracy of the model having the highest validation accuracy. For the other models we use the same hyperparameters as in table 3, but with hidden dimensions of 32 for normal models and 64 for big models. We also use a dimension 20 for all positional encodings, and initialize with uniform node and edge features full of 1s.

Table 4. Test performance on ZINC-full and PCQM4MV2.

Method	Model	**ZINC-full** (MAE ↓)	**PCQM4MV2** (MAE ↓)
GatedGCN	LE	$.033 \pm .001$	.1056
	RRWP	$.026 \pm .003$	.1045
	Q	$.031 \pm .002$	.1079
	RRWP+Q	$.026 \pm .001$	.1052
GatedGCN-big	LE	$.033 \pm .0008$	.1016
	RRWP	$.025 \pm .0017$	.1005
	Q	$.025 \pm .0023$	.1035
	RRWP+Q	$.022 \pm .0017$	.0999
GINE	LE	$.035 \pm .002$	.1155
	RRWP	$.029 \pm .003$	.114
	Q	$.027 \pm .0005$	.1149
	RRWP+Q	$.029 \pm .003$	.1124
GINE-big	LE	$.036 \pm .0022$	.1063
	RRWP	$.029 \pm .003$	.1041
	Q	$.024 \pm .002$	.1054
	RRWP+Q	$.027 \pm .0025$	.1048
GRIT	RRWP	0.025 ± 0.002	.0842
	RRWP+Q	0.023 ± 0.002	.0838

D. Supplementary information about the datasets

D.1. Datasets used in experiments on quantum random walks

The datasets used for benchmarking the use of quantum random walks encodings are standard in the GNN community. The first five are from (Dwivedi et al., 2020), the last one is from (Hu et al., 2021). We reproduce the table of statistics 5 taken from (Ma et al., 2023), and we also refer the reader to (Rampášek et al., 2022) for more information about the datasets.

Table 5. Overview of the graph learning datasets involved in this work (Dwivedi et al., 2020), (Irwin et al., 2012), (Hu et al., 2021) .

Dataset	# Graphs	Avg. # nodes	Avg. # edges	Directed	Prediction level	Prediction task	Metric
ZINC(-full)	12,000 (250,000)	23.2	24.9	No	graph	regression	Mean Abs. Error
MNIST	70,000	70.6	564.5	Yes	graph	10-class classif.	Accuracy
CIFAR10	60,000	117.6	941.1	Yes	graph	10-class classif.	Accuracy
PATTERN	14,000	118.9	3,039.3	No	inductive node	binary classif.	Weighted Accuracy
CLUSTER	12,000	117.2	2,150.9	No	inductive node	6-class classif.	Accuracy
PCQM4Mv2	3,746,620	14.1	14.6	No	graph	regression	Mean Abs. Error

D.2. Synthetic Dataset

In this subsection, we explain how to construct our artificial dataset. Our building blocks are 3 types of graphs, called types 0, 1, 2. Each type is composed of one ladder graphs with crossings inserted at different places. All crossings are in the same fixed arbitrary direction. Type 0 graphs are plain ladder graphs and their Ising hamiltonian has two ground states. Type 1 graphs are type 0 graphs with crossings separated with an odd number of nodes. The crossings are located such that they have one possible Ising ground state which is one of the ground states of the type 0 associated graph. The crossings will effectively select one of the two possible ground states. Type 2 graphs are ladder graphs of odd length with crossings at the beginning and the end. An illustration of the types of graphs is provided figure 7.

We construct a graph given two graphs of same length but different types concatenated to each other. The first class is determined by graphs of type 0 and type 1 concatenated, and the second class is composed of graphs of type 0 concatenated to graphs of type 1. The concatenation is made by adding edges to continue the ladder, the process is illustrated figure 8. The ground state of the total graph is included in a union of the groundstates of the subgraph, so it can be efficiently computed. The length of graphs are taken between 100 and 400, our dataset consists of 400 graphs per class, so 800 in total.

In figure 8 we see that the RRWP features are very similar for the two classes whereas the correlations on the ground state are very different.

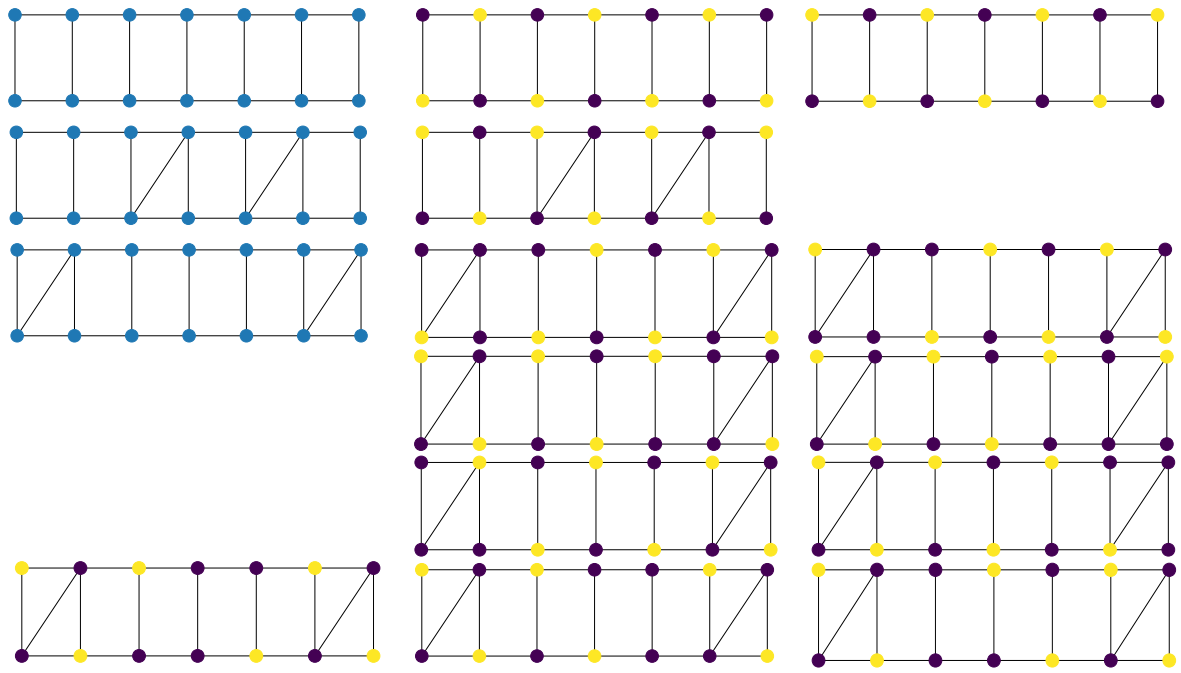

Figure 7. The base subgraphs (type 0, type 1, type 2) and their possible ground state. Top : type 0 graph of length 7, 2 possible ground states. Middle: type 1 graph of length 7, 1 possible ground state. Bottom : type 2 graph of length 7, 9 possible ground states.

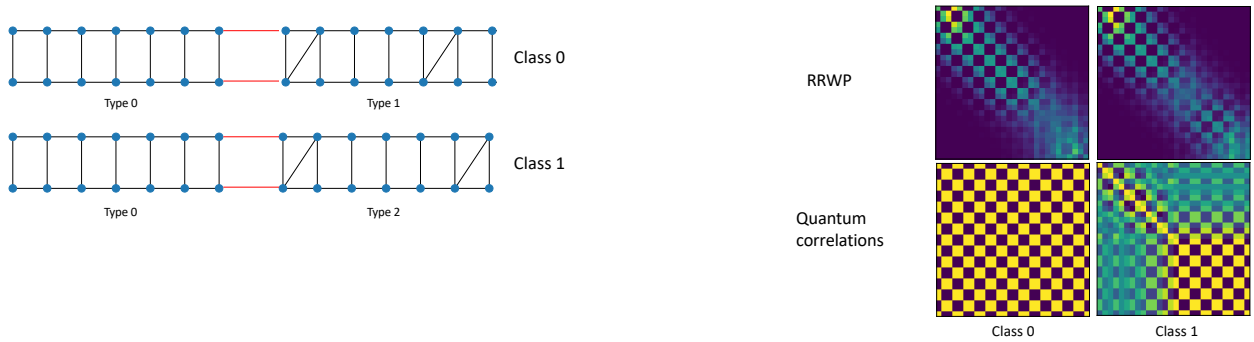

Figure 8. Left: construction of our artificial dataset. Right: RRWP and quantum features for each class of the dataset on a 40 nodes graph.