
Smoothness Discrepancies in Dynamics Models and How to Avoid Them

Edward Berman

Department of Mathematics
Northeastern University
berman.ed@northeastern.edu

Luisa Li

Khoury College of Computer Science
Northeastern University
li.tao@northeastern.edu

Abstract

A defining limitation of modern graph neural networks (GNNs) is the phenomenon of over smoothing, wherein node features of a graph become increasingly close to the node features of their neighbors. While recent architectures have proposed various inductive biases to reduce over smoothing, these biases are often designed independently of tasks where a small amount of smoothing is actually desirable, e.g. in dynamics modeling. This paper studies over smoothing in GNNs on tasks where the ground truth smoothing factor is known. In doing so, we elucidate when inductive biases that aim to preserve smoothness are useful and when they are over constraining. In the process, we propose methods to relax smoothness preserving constraints and show their efficacy on 2D heat diffusion. We also clarify the general smoothing behaviors of different GNN architectures by studying their ability to solve partial differential equations on mesh manifolds, finding that the more expressive attention and message passing based models are the best at capturing smoothness. We supplement our experiments with an approximation error lower bound that clarifies the generalization limits of smoothness preserving (unitary) functions. We release all code at https://github.com/EdwardBerman/rayleigh_analysis

1 Introduction

A defining limitation of Graph Neural Networks (GNNs) is their tendency to exhibit over smoothing [LHW18], wherein adjacent node features become increasingly similar over successive iterations of message passing. This over smoothing phenomena has been proven to occur in a variety of settings [CW20; BDC+22; Ker22; RBM23; Bal23; KFW24; AGG+25; SW25] and prevents GNNs from being able to scale with the number of layers in a network. In addition, over smoothing becomes highly problematic on long range benchmarks where the relationship between nodes separated by a high number of edges is relevant to the task.

[KFW24] employ the Rayleigh quotient, the average distance between node features connected by an edge, as a measure for over smoothing. They then prove that their proposed solution of unitary convolution prevents over smoothing, as it preserves the Rayleigh quotient. However, this poses a new problem: many tasks commonly solved using GNNs require *some* amount of smoothing. For example, heat diffusion on graphs and meshes naturally smooth the input node features (see Tab. 1).

This paper asks: how smooth is too smooth? We explore when inductive biases that preserve smoothness in GNNs are useful and when they are over constraining. We start with a synthetic dataset of graphs where the task is to auto-regressively predict the heat distribution at the next time step. We show that unitary convolution is over constrained and propose a

constraint relaxation that is able to improve performance over both the unitary convolution network and an unconstrained baseline. We also explore the sensitivity of smoothness preserving networks such as unitary convolution to Taylor series truncation errors. In doing so, we echo a theme similar to [GFG+23] that practitioners should be more thorough in evaluating when numerical approximations break strict theoretical guarantees. We follow our synthetic experiment with a series of more challenging tasks involving Partial Differential Equation (PDE) solving on meshes and underscore which architectures are the best at capturing the local smoothness of the manifold. We contextualize our experiments with an approximation error lower bound for unitary functions that clarify the generalization limits of these smoothness preserving functions.

In summary, our contributions are the following:

1. An approximation error lower bound for smoothness preserving (unitary) functions and a novel link between unitary and group invariant approximation theory (Sec. 4).
2. Numerical experiments that illustrate when unitary convolution is over constraining and when it is beneficial for learning and smoothness (Sec. 5).
3. An architectural constraint relaxation for unitary convolution (Sec. 5).
4. A generalization of the Rayleigh quotient to mesh datums and a comparison of the smoothing behavior of different mesh-GNN architectures (Sec. 6).

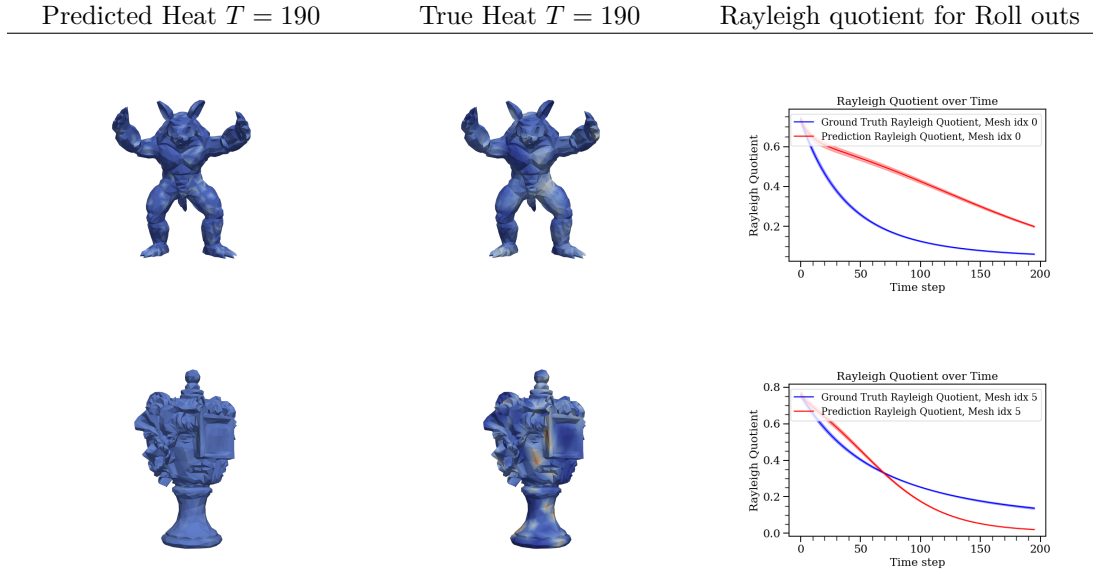


Table 1: Two graph neural networks [BGV+22; PWW23] are tasked to auto-regressively predict the heat distribution at time t . **Top:** The Hermes [PWW23] prediction is rougher than the true solution. Accordingly, the Rayleigh quotient is lower for the ground truth for each step of the roll out. This indicates *under* smoothing. **Bottom:** The EMAN [BGV+22] predicted heat distribution starts to become too smooth over time. Accordingly, the Rayleigh quotient is lower for the prediction during the latter half of the roll out. This indicates *over* smoothing.

2 Related Works

Over and under smoothing. This work quantifies the effect of neural networks on the Rayleigh quotient [Chu97] of a graph, an approach similarly employed by [KFW24].¹

¹The Rayleigh quotient is the normalized form of the Dirichlet form, which is also commonly referred to as the Dirichlet energy, local variance, and Laplacian quadratic form.

[KFW24] prove that unitary functions, and in particular the unitary convolution network, strictly preserve the Rayleigh quotient and therefore the smoothness of input graphs. This work illustrates how this property can be over constraining in GNNs. In particular, we provide a unitary function approximation error lower bound by reducing the approximation error to a group invariant problem. We then use invariant approximation error bounds established in [WZP+23], providing a novel connection between unitary and group invariant approximation. We also explore this thread experimentally: a unique feature of this work is that we use the Rayleigh quotient to empirically quantify the generalization limits of various graph neural network architectures. Similar approaches have studied the problem of over and under smoothing in PDE solutions using the Matérn kernel [BAT+21; DHM25] or fitting decay rate exponents [KBT25], but no previous works have used the Rayleigh quotient as we do for dynamics models.

[KFW24] and [AGG+25] point out that the problem of over smoothing is related to the problem of vanishing gradients. The unitary convolution approach itself borrows architectural innovations from Recurrent Neural Networks (RNNs) that were developed to better handle long context sequences [WPH+16; ASB16; MHR+17; JSD+17; HWY18; LM19]. This work illustrates that unitary convolution networks can fail to fully preserve the Rayleigh quotient due to Taylor series truncation error, and we therefore hypothesize that the success of these networks is due to the way they propagate gradients instead of their ability to preserve smoothness. This supports the initial motivation for using unitary operations as a way of combating vanishing gradients in RNNs.

Our work is perhaps most similar to [Ker22], who point out that some over smoothing can be useful for certain classification and regression tasks. Similarly, [LHW18] point out that GCNs [KW17] can be understood as a special case of Laplacian smoothing and is a key reason why GCNs work at all. In fact, [KW17] compare their architecture to hashing and the Weisfeiler-Leman 1 test [LW68], indicating that even randomly initialized GCNs can be performant due to the way they smooth out information throughout the network. Only [MBA+25] studied these over smoothing behaviors in the context of spatio-temporal modeling, and does not consider dynamics modeling specifically. Our work builds off of these studies by further clarifying when some amount of smoothing is and is not useful for spatio-temporal dynamics modeling.

Dynamics modeling over graphs and meshes. Many physical systems such as fluid flows, heat diffusion, and wave propagation, can be expressed as PDEs. Deep learning based approaches are increasingly used to solve these complex PDEs where numerical solving is difficult [WKM+20; AAB+20; MGB22; CWW+21]. These dynamics are commonly discretized as signals on graphs and meshes, and accordingly are solved with graph neural networks that propagate information locally. In the case of meshes, these dynamics can be formulated extrinsically by embedding the manifold into euclidean space [SHW21; PFS+21], or intrinsically by defining evolution directly on coordinates of local tangent spaces [CWK+19; HWC+21; BGV+22; PWW23]. In both cases, the underlying physical processes often smooth out variations in the node features. The physics of these smoothing processes are well understood, especially in the cases of heat and fluid flow [Olv93; WWY21; BAT+21]. Despite this, model performance in these domains is typically measured either via quantitative error metrics against the ground truth (mean squared error, temporal roll out error, etc) or via their preservation of underlying physical laws, such as spectral energy errors [WWY21]. Our work is novel in our application of the Rayleigh quotient in quantifying the smoothing effect of trained GNN surrogate models for PDEs. Furthermore, we are among the first to design architectures with inductive biases that encourage the model to exactly match the Rayleigh quotient of the labeled graphs. Other works have explored using the Rayleigh quotient as an auxiliary loss [REM+25] or positional encodings [DZW24], and works such as [KFW24] have developed a constrained model that preserves the Rayleigh quotient regardless of the true labels. In contrast, only our work and the work of [SSH+24] tries to match the smoothness of labeled graphs via inductive biases in the network architecture, and only our work assesses how well the true smoothness of the physical dynamics are recovered in evaluation.

3 Background

3.1 Rayleigh quotient

To measure smoothness of a graph, we use the Rayleigh quotient as defined in [Chu97]. Consistent with [KFW24], we use the following notation:

Definition 1 (Rayleigh quotient, [Chu97], also Definition 5 in [KFW24]). Given an undirected graph $\mathcal{G} = (V, E)$ on $|V| = n$ nodes with adjacency matrix $\mathbf{A} \in \{0, 1\}^{n \times n}$, let $\mathbf{D} \in \mathbb{R}^{n \times n}$ be a diagonal matrix where the i -th entry $\mathbf{D}_{ii} = d_i$ and d_i is the degree of node i . Let $f : V \rightarrow \mathbb{C}^d$ be a function from nodes to features. Then the Rayleigh quotient $R_{\mathcal{G}}(\mathbf{X})$ is equal to

$$R_{\mathcal{G}}(\mathbf{X}) = \frac{1}{2} \frac{\sum_{(u,v) \in E} \left\| \frac{f(u)}{\sqrt{d_u}} - \frac{f(v)}{\sqrt{d_v}} \right\|^2}{\sum_{w \in V} \|f(w)\|^2} = \frac{\text{Tr}(\mathbf{X}^\dagger (\mathbf{I} - \tilde{\mathbf{A}}) \mathbf{X})}{\|\mathbf{X}\|_F^2}, \quad (1)$$

where $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ is the normalized adjacency matrix and $\mathbf{X} \in \mathbb{C}^{n \times d}$ is a matrix with the i -th row set to feature vector $f(i)$. We will often abbreviate the normalized graph Laplacian as $\mathbf{L} = (\mathbf{I} - \tilde{\mathbf{A}})$.

Intuitively, the Rayleigh quotient measures the average difference in node features of nodes connected by an edge. A graph with uniform node features has a Rayleigh quotient of zero.

3.2 Unitary Convolution

[KFW24] contribute two models that preserve the Rayleigh quotient using unitary operations.

$$\text{(Separable Unitary Convolution)} \quad f_{\text{Uconv}}(\mathbf{X}) = \exp(\mathbf{iAt}) \mathbf{X} \mathbf{U}, \quad \mathbf{U}^\dagger \mathbf{U} = \mathbf{I} \quad (2)$$

$$\text{(Lie Unitary Convolution)} \quad f_{\text{Uconv}}(\mathbf{X}) = \exp(\mathbf{AXW}), \quad \mathbf{W} = -\mathbf{W}^\dagger \quad (3)$$

where $\exp(\cdot)$ denotes the matrix exponential. We provide further background material on the matrix exponential and its relationship to unitary matrices in Sec. A.1. The authors show that networks constructed from unitary convolutions are architecturally constrained to preserve the Rayleigh quotient:

Proposition 1 (Invariance of Rayleigh quotient, Proposition 6 in [KFW24]). *Given an undirected graph \mathcal{G} on n nodes with normalized adjacency matrix $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$, the Rayleigh quotient $R_{\mathcal{G}}(\mathbf{X}) = R_{\mathcal{G}}(f_{\text{Uconv}}(\mathbf{X}))$ is invariant under normalized unitary or orthogonal graph convolution (see Eq. (2) and Eq. (3)).*

Crucially, separable unitary convolution is often relaxed by dropping the constraint that $\mathbf{U}^\dagger \mathbf{U} = \mathbf{I}$, meaning that Proposition 1 is not strictly satisfied. Additionally, Taylor series truncation errors in the matrix exponential can cause both separable and Lie unitary convolution to violate Proposition 1. [KFW24] also show that more conventional architectures such as GCNs [KW17] are likely to exhibit over smoothing. We include the proposition in Sec. A.2 for completeness.

3.3 The Mesh Datum

A (triangular) mesh M consists of a set $(\mathcal{V}, \mathcal{E}, \mathcal{F})$, where \mathcal{V} is a set of vertices, $\mathcal{E} = \{(i, j)\}$ is a set of ordered vertex indices i, j connected by an edge, and $\mathcal{F} = \{(i, j, k)\}$ is the set of ordered vertex indices i, j, k connected by a triangular face. The mesh datum generalizes graphs by including high order connectivity information via the inclusion of faces. We assume that the mesh is a 2-dimensional manifold embedded in \mathbb{R}^3 , i.e. a manifold mesh. For a complete description of the local tangent space structures on mesh we defer to [PWW23]. Crucial to this work is that we employ models that work in the coordinates of local tangent planes to the manifold mesh as well as models that use the global \mathbb{R}^3 embedding coordinates.

4 Theory on Overly Constrained Unitary Functions

While unitary functions on graphs can be useful because they preserve the Rayleigh quotient, this section illustrates how unitary functions can be *overly* constrained. In particular, we

derive an approximation error lower bound that clarifies the generalization limits of unitary functions. We start by establishing our unitary approximation learning framework.

Unitary Approximation Learning Framework. Let $Z = \mathbb{C}^n$ be a domain. Let $p : Z \rightarrow \mathbb{R}$ be the density on Z . Let $u : \mathbb{C}^n \rightarrow \mathbb{C}^n$ be a unitary function and let $f : \mathbb{C}^n \rightarrow \mathbb{C}^n$ be an arbitrary function. Denote the regression error by

$$\text{err}_{\text{reg}}(u) = \int_Z p(z) \|u(z) - f(z)\|_2^2 dz. \quad (4)$$

We assume no covariate shift during testing, i.e., $p(z)$ is always the underlying data distribution. Our result in [Theorem 1](#) will use concepts from the approximation error of group invariant functions h . We review these concepts in detail in [Sec. A.3](#) and provide informal definitions in the paragraph that follows.

A group invariant function h satisfies $h(z) = h(gz)$ for all $g \in G, z \in Z$. Let $Gz = \{gz : g \in G\}$ be the orbit of z . A *fundamental domain* F of a group G in Z is a set of orbit representatives. Z can be written as the union of conjugates, $Z = \cup_{g \in G} gF$, where the conjugate is defined $gF = \{gz : z \in F\}$. Denote the integrated density on an orbit by $p(Gz) = \int_{Gz} p(z) dz$. Finally, denote the average and variance of a function f on an orbit Gz by $\mathbb{E}_{Gz}[f]$ and $\mathbb{V}_{Gz}[f]$ respectively. The approximation error lower bound for an invariant function is given by the following proposition.

Proposition 2 (Theorem 4.8 in [\[WZP+23\]](#)). *For a G -invariant function h , the regression error is bounded below by $\text{err} \geq \int_F p(Gz) \mathbb{V}_{Gz}[f] dz$.*

[Proposition 2](#) was initially stated for real valued functions in [\[WZP+23\]](#), but can be applied to complex valued functions without loss of generality.

Incorrect Unitary Approximation Error Lower Bound. Recall the definition of $SU(n)$, the group of rotations in \mathbb{C}^n :

$$SU(n) = \{U \in \mathbb{C}^{n \times n} : \det(U) = 1\}.$$

We now have the necessary machinery to prove our approximation error lower bound in the case of an incorrect unitary constraint.

Theorem 1. *Let F be a fundamental domain of $SU(n)$ in Z . In particular, $F = \{te, t \in \mathbb{R}_+\}$ where e is a standard basis vector of \mathbb{C}^n . The approximation error lower bound can be expressed as*

$$\int_Z p(z) \|u(z) - f(z)\|_2^2 dz \geq \int_F p(\|te\|) \mathbb{V}_{Gz}[\|f\|] dz.$$

Proof. By the reverse triangle inequality,

$$\int_Z p(z) \|u(z) - f(z)\|_2^2 dz \geq \int_Z p(z) (\|u(z)\| - \|f(z)\|)^2 dz.$$

Notice that $\|u(z)\|$ is invariant under the action of $SU(n)$ on the sphere S^{2n-1} with radius $\|te\|$ and recall that $SU(n)$ acts transitively on the sphere. Thus, $F = \{te, t \in \mathbb{R}_+\}$ is a valid fundamental domain that indexes each orbit Gz , the spheres with radii $\|te\|$. Our theorem then follows from [Proposition 2](#). \square

Intuitively, the fundamental domain enumerates all concentric spheres S^{2n-1} embedded in \mathbb{C}^n . Unitary functions are complex valued rotations and reflections that preserve the norm of data points that live on each sphere. The error lower bound is given by the variance of the norm of f averaged over each concentric sphere wherein the norm of u is constant.

5 Unitary Convolution Constraint Relaxation

In this section, we prescribe a method for relaxed unitary convolution and show that it significantly outperforms normal graph convolution and marginally outperforms standard

Lie Unitary Convolution on a synthetic heat diffusion task. We attribute the relative success of the method not only to the constraint relaxation but also the tendencies of GCNs at initialization. We provide evidence for this by studying the training curves of an ensemble of runs for each model. Our method for constraint relaxation depends not on separating the exponential function from the weight updates as in Eq. (2), but in truncating the number of Taylor series terms in the matrix exponential in Eq. (3) so that the Rayleigh quotient is not strictly preserved.

5.1 Relaxed Unitary Convolution

Since Theorem 1 informs us that a unitary convolution network may be over constraining, this section proposes a constraint relaxation to the architecture. We note that the authors [KFW24] propose their own constraint relaxation by allowing \mathbf{U} to be unconstrained in Eq. (2), and that our approach instead relaxes Eq. (3). We do this because it allows us to isolate the architectural component that alters the Rayleigh quotient. In contrast, the relaxation in [KFW24] can be achieved via two different mechanisms and the relative contributions of each are difficult to measure. Our relaxation is also simple, instead of approximating the the matrix exponential using enough Taylor series terms so that the truncation error is vanishingly small, we truncate at some $T = \mathbf{T}_{\max}$ where \mathbf{T}_{\max} controls the extent of the relaxation:

$$f_{\text{Relaxed}}(\mathbf{X}; \mathbf{A}, \mathbf{T}_{\max}) = \sum_{i=0}^{\mathbf{T}_{\max}} \frac{1}{i!} \mathbf{L}^i(\mathbf{X}) \quad (5)$$

where $\mathbf{L}(\mathbf{X}) = \mathbf{A}\mathbf{X}\mathbf{W}$, $\mathbf{W} = -\mathbf{W}^\dagger$. This approach does not preserve the Rayleigh quotient for small \mathbf{T}_{\max} . In the limit as $T \rightarrow \infty$ we recover standard Lie Unitary convolution in Eq. (3). Empirically, we find that $T = 10$ is sufficient to preserve the Rayleigh quotient, which is consistent with what is employed in [KFW24]. Accordingly, we will use $f_{\text{Relaxed}}(\mathbf{X}; \mathbf{A}, 10)$ interchangeably with $f_{\text{LieUniConv}}(\mathbf{X}; \mathbf{A})$.

5.2 Rayleigh Quotient Sensitivity

Motivated by the desire to find an appropriate \mathbf{T}_{\max} that applies only a small perturbation to the Rayleigh quotients of input graphs, we conduct a sensitivity analysis of the Rayleigh quotient to different Taylor series truncations. For completeness, we also compare with standard GCNs and Separable Unitary networks. We start by describing our procedure for creating a synthetic heat diffusion dataset before outlining our experimental setup. We choose to examine heat diffusion so that we can study the tendencies of the different architectures at initialization in a domain of interest rather than on randomly initialized graphs. Moreover, we will use this dataset for the experiment in Sec. 5.3 as well.

Heat Diffusion Data Curation. We use [DMP+] to generate 10,000 two-dimensional grids with 20 randomly placed heat sources and use [DMP+] to model the true heat diffusion. Our setup mirrors the common task of modeling heat flow on a hot plate. Denote by $f : \mathbb{R}_+ \rightarrow \mathbb{R}^n$ a function that maps time t to the heat distribution of the graph. In particular, $t \mapsto e^{-\tau t} \mathbf{L} f(0)$ where τ is a diffusivity constant, \mathbf{L} is the graph Laplacian, and $f(0)$ is the initial heat graph. The exponential term is sometimes referred to as the Green’s function of the graph diffusion equation [BAT+21]. The first three time steps are designated as training and the fourth as validation.

Experimental Setup. We use time step 3 to conduct the sensitivity analysis, and evaluate on the models f_{GCN} , $f_{\text{SepUniConv}}$, and $f_{\text{LieUniConv}}$. For each model f and truncation length $\mathbf{T}_{\max} \in \{1, \dots, 10\}$, we compute the Rayleigh quotients $R_{\mathcal{G}}(\mathbf{X})$ and $R_{\mathcal{G}}(f(\mathbf{X}))$ for all graph mini batches \mathbf{X} . We denote the distribution of Rayleigh quotients before applying the model by $P_{\mathbf{X}}$ and after applying the model by $P_{f(\mathbf{X})}$. To quantify the deviation between these distributions, we compute the KL divergence $D_{\text{KL}}(P_{\mathbf{X}} \parallel P_{f(\mathbf{X})})$, which measures the change in the distribution of Rayleigh quotients caused by the model at initialization.

Results. We see in Fig. 1 the effect of Taylor series truncation on the unitarity of the network. In particular, we observe that the KL divergence between the two distributions decreases exponentially with the number of terms. This is to be expected, we know from Taylor’s theorem that a truncation at term t gives truncation error $\mathcal{O}([\mathbf{A}\mathbf{X}\mathbf{W}]^{t+1})$, and furthermore, works such as [FH24; DZW24] show theoretically that small truncation errors will not compound into large deviations in the Rayleigh quotient. For details on the relevant propositions from [FH24; DZW24], see Sec. A.4.

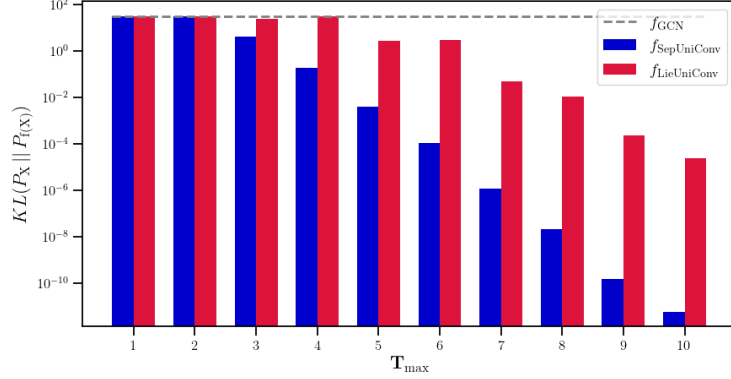


Figure 1: KL divergence between distribution of Rayleigh quotients before and after applying the model. Results are averaged over 10 runs.

5.3 Heat Diffusion Performance and Smoothing Comparison

Having established tendencies of the different architectures at initialization, we now show how this manifests in different observed training dynamics and that the relaxed unitary convolution network in Eq. (5) provides the best performance.

Experimental Setup. We use the same dataset and objective as in Sec. 5.2. The network must predict the heat distribution on the graph at time t given the heat distribution at time $t - 1$. We denote the true node feature labels (heat) as Y . We compare the performance in terms of MSE loss and mean Rayleigh quotient for three models: f_{GCN} , f_{Relaxed} , and $f_{\text{LieUniConv}}$. We use $T_{\max} = 3$ for the relaxed model. In order to understand how tendencies at initialization manifest in different training dynamics, we examine training curves over multiple different runs. Further details are given in Sec. B.

Results. We see in Fig. 2 that the relaxed model significantly outperforms the GCN and marginally outperforms the Lie Unitary model. Moreover, the relaxed model is best able to produce graphs whose smoothness matches that of the true labels. We note that while only 5 runs are shown for visual clarity, extensive ablations confirmed the behavior over a much larger set of runs. Our results are grounded theoretically by [KFW24, Proposition 7], which is provided in Sec. A.2 for convenience. The proposition demonstrates that GCNs with weights commonly seen at initialization are likely to exhibit smoothing. The nuance exhibited by our experiment is that this tendency at initialization makes GCNs perform poorly relative to other architectures even in a toy task where one of the baselines is overly constrained.

6 Smoothness on Meshes

Having illustrated when inductive biases that preserve smoothness is both helpful and harmful for learning on graphs, we now turn our attention towards architectures developed for the more intricate mesh datum. In particular, we study the problem of solving PDEs on PyVista [SK19] meshes using Gauge Equivariant GNNs.

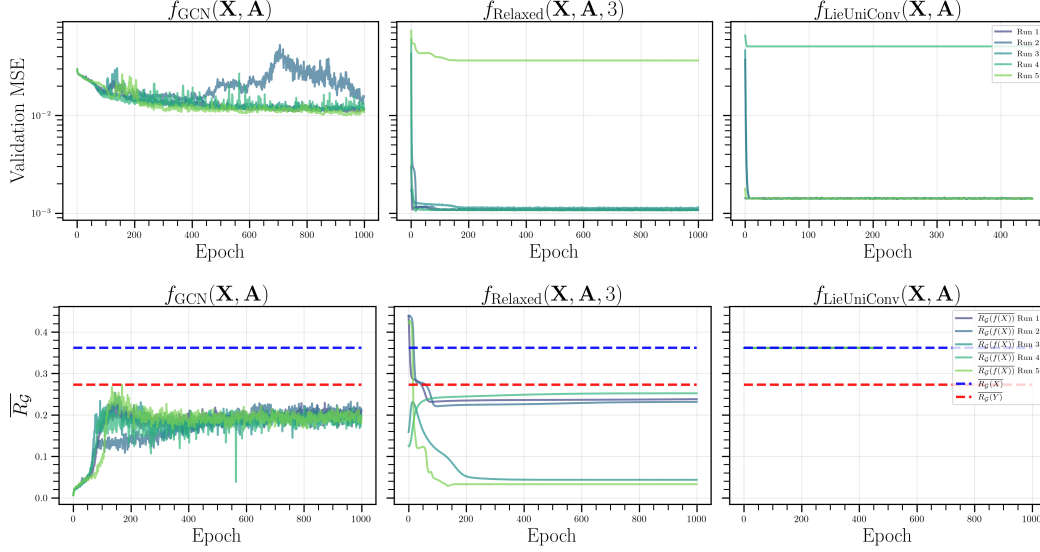


Figure 2: **Top:** Validation MSE for an ensemble of 5 runs for a GCN (left), a Lie Unitary Convolution network with 3 truncation terms (middle), and a Lie Unitary Convolution network with 10 truncation terms (right). The 3 truncation Lie Unitary network significantly outperforms the GCN and marginally outperforms the 10 truncation term Lie Unitary network. **Bottom:** The average Rayleigh quotient over all graphs for an ensemble of 5 runs for the same models. The GCN is under constrained and biased towards over smoothing at initialization. The 3 truncation term Lie Unitary network is able to roughly match the true smoothness of the labeled graphs. The 10 truncation term Lie Unitary network is over constrained and can not model the Rayleigh quotient of the labels because it is forced to preserve the Rayleigh quotient of the input graphs.

Models and Data. We benchmark all existing flavors of Gauge Equivariant GNNs, including convolutional with GemCNN [HWC+21], attentional with EMAN [BGV+22], and message passing with Hermes [PWW23]. We also consider a state of the art mesh transformer [JBN+23] and a state of the art $E(3)$ -equivariant model [SHW21]. The task is to auto-regressively predicting the solution to the heat and wave equations on meshes given an initial condition (Eq. (7)) and we use the same PyVista meshes generated in [PWW23] as data. Let α be the thermal diffusivity, c a constant, and $\tilde{\mathbf{L}}$ the symmetric cotangent Laplacian [RBG+09]. In particular, for a scalar function u , we have

$$(\tilde{\mathbf{L}}(u))_i = \frac{1}{2A_i} \sum_{j \in \mathcal{N}(i)} (\cot \alpha_{ij} + \cot \beta_{ij}) (u_j - u_i) \quad (6)$$

where $\mathcal{N}(i)$ denotes the adjacent vertices of i , α_{ij} and β_{ij} are the angles opposite edge (i, j) , and A_i is the vertex area of i , where we use the barycentric cell area. The heat and wave equations on the mesh are then given by

$$\frac{\partial u}{\partial t} = \alpha \tilde{\mathbf{L}} u, (\text{Heat}) \quad \frac{\partial^2 u}{\partial t^2} = c^2 \tilde{\mathbf{L}} u, (\text{Wave}). \quad (7)$$

Our task is to approximate these PDEs by auto-regressively predicting the spatial distribution of u at each time step t . We use roll outs on unseen meshes as validation. Each unseen mesh has five different initializations to test on. Further training details are given in Sec. C.

Evaluation. We now generalize our metrics such that they apply to mesh datums and multi-step temporal roll outs. First, we note that the Rayleigh quotient in Eq. (1) uses the Laplacian $\mathbf{L} = \mathbf{I} - \mathbf{A}$. While it is tempting to simply replace \mathbf{L} with the symmetric cotangent Laplacian $\tilde{\mathbf{L}}$ in Eq. (6), this does not work. The cotangent weights in Eq. (6) may be negative, which in turn means that the Rayleigh quotient is no longer a valid measure of

over smoothing [RBM23, Definition 1]. Instead, we use the “robust Laplacian” [SC20], which rewires the mesh so that the cotangent weights obey the *Delaunay criterion*. Concretely, this means that our Laplacian weights are both symmetric and the off-diagonals are less than or equal to zero².

Next, in order to aggregate discrepancies between the smoothness of the labeled graph with the one produced by the model over all time steps, we introduce a new metric. Define the Integrated Rayleigh Error (IRE) by $\int_0^\infty |R_{\mathcal{G}}(\mathbf{Y}_t) - R_{\mathcal{G}}(f(\mathbf{X}_t))| dt$. In practice we approximate this by summing over the time steps where we are able to perform inference:

$$\text{IRE}(f) = \sum_t |R_{\mathcal{G}}(\mathbf{Y}_t) - R_{\mathcal{G}}(f(\mathbf{X}_t))|.$$

Following [JBN+23; PYV+25], we also consider the scale invariant metrics NRMSE and SMAPE integrated over the entire roll out:

$$\begin{aligned} \text{INRMSE}(f) &= \sum_t \sqrt{\frac{\frac{1}{n} \sum_{i=1}^n (f(\mathbf{X}_t)_i - (\mathbf{Y}_t)_i)^2}{\frac{1}{n} \sum_{i=1}^n (\mathbf{Y}_t)_i^2}} \\ \text{ISMAPE}(f) &= \sum_t \frac{1}{n} \sum_{i=1}^n \frac{2|(\mathbf{Y}_t)_i - f(\mathbf{X}_t)_i|}{|(\mathbf{Y}_t)_i| + |f(\mathbf{X}_t)_i| + \varepsilon} \end{aligned}$$

where $\varepsilon = 10^{-8}$ is a stability constant. SMAPE is generally more robust than NMRSE in that it is less sensitive to outliers, but it is also more sensitive to small values. The scale invariant property of these metrics is crucial especially for heat diffusivity because solutions tend to decrease proportionally to e^{-t} . Thus, we need to consider deviations across several orders of magnitude in order to see how accurately we are modeling the decay.

Results. The message passing and attentional models perform better than convolutional both in terms of raw performance and smoothness. The EMAN and Hermes gauge equivariant methods tend to perform better on the heat equation, whereas the EMAN and the Mesh Transformer show the best results on the wave equation. This is indicated by the INRMSE, ISMAPE, and IRE for each of the models, which is given in Tab. 2. Our results indicate that while most models are able to fit the function with comparable accuracy, there are more nuanced tradeoffs between each model in their ability to capture the underlying smoothness.

Table 2: INRMSE, ISMAPE, and IRE averaged over all roll outs on all test meshes for the heat and wave equations. Best performing model is indicated with bold text and the second best is indicated with an underline.

Heat ($\alpha = 1$)			
Model	INRMSE (\downarrow)	ISMAPE (\downarrow)	IRE (\downarrow)
GemCNN [HWC+21]	$2.28 \times 10^{18} \pm 4.1 \times 10^{17}$	375.4 ± 0.53	52.21 ± 9.4
EMAN [BGV+22]	<u>73.50 ± 3.8</u>	<u>110.9 ± 13.3</u>	14.2 ± 1.4
Hermes [PWW23]	73.02 ± 4.7	107.6 ± 7.4	39.76 ± 4.7
Mesh Transformer [JBN+23]	93.02 ± 13.46	133.9 ± 5.4	<u>38.69 ± 7.6</u>
EGNN [SHW21]	344.25 ± 110.5	319.33 ± 7.59	<u>81.5 ± 8.77</u>
Wave ($c = 1$)			
GemCNN [HWC+21]	$1.17 \times 10^7 \pm 1.17 \times 10^7$	318.8 ± 3.9	107.9 ± 3.158
EMAN [BGV+22]	281.3 ± 15.5	301.0 ± 4.2	73.57 ± 6.5
Hermes [PWW23]	458.5 ± 13.0	<u>316.4 ± 4.5</u>	<u>70.03 ± 6.1</u>
Mesh Transformer [JBN+23]	323.8 ± 32.9	296.5 ± 2.5	12.14 ± 1.8
EGNN [SHW21]	2280.1 ± 559.9	354.3 ± 11.02	157.2 ± 14.8

²If the opposite $\tilde{\mathbf{L}}$ sign convention is taken, then the constraint is that the weights are nonnegative.

7 Conclusion

This work marks a step forward towards understanding smoothness discrepancies in GNNs for discrete dynamics modeling on graphs and meshes. In particular, we elucidate the approximation limits of unitary functions and unitary convolution networks, and show how constraint relaxations results in improved performance on 2D heat diffusion. We also discuss how various inductive biases are able to solve dynamical systems on the more intricate mesh setting, and paint a more nuanced picture regarding the model’s ability to capture the underlying smoothness.

Acknowledgments. E.B. thanks Noah Shinn for insightful discussions.

References

- [AAB+20] Anima Anandkumar et al. “Neural operator: Graph kernel network for partial differential equations”. In: *ICLR 2020 workshop on integration of deep neural models and differential equations*. 2020.
- [ASB16] Martin Arjovsky, Amar Shah, and Yoshua Bengio. “Unitary evolution recurrent neural networks”. In: *International conference on machine learning*. PMLR. 2016, pp. 1120–1128.
- [AGG+25] Alvaro Arroyo et al. “On Vanishing Gradients, Over-Smoothing, and Over-Squashing in GNNs: Bridging Recurrent and Graph Learning”. In: *The Thirty-ninth Annual Conference on Neural Information Processing Systems*. 2025. URL: <https://openreview.net/forum?id=N4cyRMuLyl>.
- [Bal23] Julia Balla. “Over-squashing in Riemannian Graph Neural Networks”. In: *The Second Learning on Graphs Conference*. 2023. URL: <https://openreview.net/forum?id=UUnYi0yLcM>.
- [BGV+22] Sourya Basu et al. “Equivariant Mesh Attention Networks”. In: *Transactions on Machine Learning Research* (2022). Expert Certification. ISSN: 2835-8856. URL: <https://openreview.net/forum?id=3IqqJh2Ycy>.
- [BDC+22] Cristian Bodnar et al. “Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnn’s”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 18527–18541.
- [BAT+21] Viacheslav Borovitskiy et al. “Matérn Gaussian processes on graphs”. In: *International Conference on Artificial Intelligence and Statistics*. PMLR. 2021, pp. 2593–2601.
- [CW20] Chen Cai and Yusu Wang. “A note on over-smoothing for graph neural networks”. In: *arXiv preprint arXiv:2006.13318* (2020).
- [CWW+21] Shengze Cai et al. “Physics-informed neural networks for heat transfer problems”. In: *Journal of Heat Transfer* 143.6 (2021), p. 060801.
- [Chu97] Fan RK Chung. *Spectral graph theory*. Vol. 92. American Mathematical Soc., 1997.
- [CWK+19] Taco Cohen et al. “Gauge equivariant convolutional networks and the icosahedral CNN”. In: *International conference on Machine learning*. PMLR. 2019, pp. 1321–1330.
- [DHM25] Mara Daniels, Liam Hodgkinson, and Michael Mahoney. “Uncertainty-Aware Diagnostics for Physics-Informed Machine Learning”. In: *arXiv preprint arXiv:2510.26121* (2025).
- [DMP+] Michaël Defferrard et al. *PyGSP: Graph Signal Processing in Python*. DOI: [10.5281/zenodo.1003157](https://doi.org/10.5281/zenodo.1003157). URL: <https://github.com/epfl-lts2/pygsp/>.
- [DZW24] Xiangyu Dong, Xingyi Zhang, and Sibor Wang. “Rayleigh Quotient Graph Neural Networks for Graph-level Anomaly Detection”. In: *The Twelfth International Conference on Learning Representations*. 2024. URL: <https://openreview.net/forum?id=4UIBysXjVq>.
- [FH24] Giulia Ferrandi and Michiel E Hochstenbach. “A homogeneous Rayleigh quotient with applications in gradient methods”. In: *Journal of Computational and Applied Mathematics* 437 (2024), p. 115440.
- [GFG+23] Nate Gruver et al. “The Lie Derivative for Measuring Learned Equivariance”. In: *The Eleventh International Conference on Learning Representations*. 2023. URL: <https://openreview.net/forum?id=JL7Va5Vy15J>.
- [HWC+21] Pim De Haan et al. “Gauge Equivariant Mesh {CNN}s: Anisotropic convolutions on geometric graphs”. In: *International Conference on Learning Representations*. 2021. URL: <https://openreview.net/forum?id=Jnspzp-oIZE>.
- [HWY18] Kyle Helfrich, Devin Willmott, and Qiang Ye. “Orthogonal recurrent neural networks with scaled Cayley transform”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1969–1978.
- [JBN+23] Steeven Janny et al. “Eagle: Large-scale learning of turbulent fluid dynamics with mesh transformers”. In: *arXiv preprint arXiv:2302.10803* (2023).

- [JSD+17] Li Jing et al. “Tunable efficient unitary neural networks (eunn) and their application to rnns”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 1733–1741.
- [Ker22] Nicolas Keriven. “Not too little, not too much: a theoretical analysis of graph (over) smoothing”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 2268–2281.
- [KFW24] Bobak Kiani, Lukas Fesser, and Melanie Weber. “Unitary convolutions for learning on graphs and groups”. In: *Advances in Neural Information Processing Systems* 37 (2024), pp. 136922–136961.
- [KW17] Thomas N. Kipf and Max Welling. “Semi-Supervised Classification with Graph Convolutional Networks”. In: *International Conference on Learning Representations*. 2017. URL: <https://openreview.net/forum?id=SJU4ayYgl>.
- [KBT25] Charles Kulick, Björn Birnir, and Sui Tang. “Investigating Zero-Shot Size Transfer of Graph Neural Differential Equations for Learning Graph Diffusion Dynamics”. In: *Topology, Algebra, and Geometry in Data Science*. 2025. URL: <https://openreview.net/forum?id=qgbyLknKXY>.
- [LW68] Andrei Leman and Boris Weisfeiler. “A reduction of a graph to a canonical form and an algebra arising during this reduction”. In: *Nauchno-Technicheskaya Informatsiya* 2.9 (1968), pp. 12–16.
- [LM19] Mario Lezcano-Casado and David Martinez-Rubio. “Cheap orthogonal constraints in neural networks: A simple parametrization of the orthogonal and unitary group”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 3794–3803.
- [LHW18] Qimai Li, Zhichao Han, and Xiao-Ming Wu. “Deeper insights into graph convolutional networks for semi-supervised learning”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.
- [MBA+25] Ivan Marisca et al. “Over-squashing in Spatiotemporal Graph Neural Networks”. In: *arXiv preprint arXiv:2506.15507* (2025).
- [MGB22] Marco Maurizi, Chao Gao, and Filippo Berto. “Predicting stress, strain and deformation fields in materials and structures with graph neural networks”. In: *Scientific reports* 12.1 (2022), p. 21834.
- [MHR+17] Zakaria Mhammedi et al. “Efficient orthogonal parametrisation of recurrent neural networks using householder reflections”. In: *International Conference on Machine Learning*. PMLR. 2017, pp. 2401–2409.
- [Olv93] Peter J Olver. *Applications of Lie groups to differential equations*. Vol. 107. Springer Science & Business Media, 1993.
- [PYV+25] Sneha Pandya et al. “IAEmu: Learning Galaxy Intrinsic Alignment Correlations”. In: *arXiv preprint arXiv:2504.05235* (2025).
- [PWW23] Jung Yeon Park, Lawson Wong, and Robin Walters. “Modeling dynamics over meshes with gauge equivariant nonlinear message passing”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 15277–15302.
- [PFS+21] Tobias Pfaff et al. “Learning Mesh-Based Simulation with Graph Networks”. In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=roNqYLO_XP.
- [RBG+09] Martin Reuter et al. “Discrete Laplace–Beltrami operators for shape analysis and segmentation”. In: *Computers & Graphics* 33.3 (2009), pp. 381–390.
- [REM+25] Conor Rowan et al. “Solving engineering eigenvalue problems with neural networks using the Rayleigh quotient”. In: *arXiv preprint arXiv:2506.04375* (2025).
- [RBM23] T Konstantin Rusch, Michael M Bronstein, and Siddhartha Mishra. “A survey on oversmoothing in graph neural networks”. In: *arXiv preprint arXiv:2303.10993* (2023).
- [SHW21] Victor Garcia Satorras, Emiel Hoogeboom, and Max Welling. “E (n) equivariant graph neural networks”. In: *International conference on machine learning*. PMLR. 2021, pp. 9323–9332.

- [SSH+24] Zhiqi Shao et al. *Unifying over-smoothing and over-squashing in graph neural networks: A physics informed approach and beyond*. 2024. URL: <https://openreview.net/forum?id=swPf2hwKl8>.
- [SC20] Nicholas Sharp and Keenan Crane. “A laplacian for nonmanifold triangle meshes”. In: *Computer Graphics Forum*. Vol. 39. 5. Wiley Online Library. 2020, pp. 69–80.
- [SW25] Junwei Su and Chuan Wu. “On the Interplay between Graph Structure and Learning Algorithms in Graph Neural Networks”. In: *Forty-second International Conference on Machine Learning*. 2025.
- [SK19] C Sullivan and Alexander Kaszynski. “PyVista: 3D plotting and mesh analysis through a streamlined interface for the Visualization Toolkit (VTK)”. In: *Journal of Open Source Software* 4.37 (2019), p. 1450.
- [WZP+23] Dian Wang et al. “A general theory of correct, incorrect, and extrinsic equivariance”. In: *Advances in Neural Information Processing Systems* 36 (2023), pp. 40006–40029.
- [WWY21] Rui Wang, Robin Walters, and Rose Yu. “Incorporating Symmetry into Deep Dynamics Models for Improved Generalization”. In: *International Conference on Learning Representations*. 2021. URL: https://openreview.net/forum?id=wta_8Hx2KD.
- [WKM+20] Rui Wang et al. “Towards Physics-informed Deep Learning for Turbulent Flow Prediction”. In: *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD ’20. Virtual Event, CA, USA: Association for Computing Machinery, 2020, pp. 1457–1466. ISBN: 9781450379984. DOI: [10.1145/3394486.3403198](https://doi.org/10.1145/3394486.3403198). URL: <https://doi.org/10.1145/3394486.3403198>.
- [WPH+16] Scott Wisdom et al. “Full-capacity unitary recurrent neural networks”. In: *Advances in neural information processing systems* 29 (2016).

A Deferred Theory

A.1 Lie Algebras and the Exponential Map

A group is a mathematical structure that formalizes what it means for something to be *symmetric*. We say that a group is a matrix *Lie group*, if it is a differentiable manifold and a subgroup of the set of invertible $n \times n$ matrices. Lie groups are equipped with a *lie algebra*, which is the tangent space at the identity element. This work encounters the orthogonal and unitary lie groups

$$O(n) = \{O \in \mathbb{R}^{n \times n} : OO^T = I\}, \quad U(n) = \{U \in \mathbb{C}^{n \times n} : UU^\dagger = I\}$$

as well as the special unitary group

$$SU(n) = \{U \in \mathbb{C}^{n \times n} : \det(U) = 1\}.$$

The associated lie algebras for $O(n)$ and $U(n)$ are given by

$$\mathfrak{o}(n) = \{M \in \mathbb{R}^{n \times n} : M + M^T = 0\}, \quad \mathfrak{u}(n) = \{M \in \mathbb{C}^{n \times n} : M + M^\dagger = 0\}.$$

The exponential map provides a mechanism of parameterizing lie groups with elements in the lie algebra. For matrix lie groups, the exponential map is simply the matrix exponential:

$$\exp(\mathbf{X}) = \sum_i \frac{1}{i!} \mathbf{X}^i.$$

Applying the exponential map to a linear operator is given by

$$\exp(\mathbf{L})(\mathbf{X}) = \sum_i \frac{1}{i!} \mathbf{L}^i(\mathbf{X}) = \mathbf{X} + \mathbf{L}(\mathbf{X}) + \frac{1}{2} \mathbf{L} \circ \mathbf{L}(\mathbf{X}) + \frac{1}{6} \mathbf{L} \circ \mathbf{L} \circ \mathbf{L}(\mathbf{X}) + \dots$$

In the case of [Eq. \(3\)](#), \mathbf{L} is graph convolution, $\mathbf{L}(\mathbf{X}) = \mathbf{A}\mathbf{X}\mathbf{W}$.

A.2 Convolutional Over smoothing

[\[KFW24\]](#) establish that Graph Convolution Networks [\[KW17\]](#) have a high probability to exhibit smoothing.

Proposition 3 (Proposition 7 in [\[KFW24\]](#)). *Given a simple undirected graph \mathcal{G} on n nodes with normalized adjacency matrix $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$ and node degree bounded by D , let $\mathbf{X} \in \mathbb{R}^{n \times d}$ have rows drawn i.i.d. from the uniform distribution on the hypersphere in dimension d . Let $f_{\text{conv}}(\mathbf{X}) = \tilde{\mathbf{A}} \mathbf{X} \mathbf{W}$ denote convolution with orthogonal feature transformation matrix $\mathbf{W} \in O(d)$. Then, the event below holds with probability $1 - \exp(-\Omega(\sqrt{n}))$:*

$$R_{\mathcal{G}}(\mathbf{X}) \geq 1 - O\left(\frac{1}{n^{1/4}}\right) \quad \text{and} \quad R_{\mathcal{G}}(f_{\text{conv}}(\mathbf{X})) \leq 1 - \frac{\text{Tr}(\tilde{\mathbf{A}}^3)}{\text{Tr}(\tilde{\mathbf{A}}^2)} + O\left(\frac{1}{n^{1/4}}\right).$$

A.3 Unitary Learning Framework

This section provides rigorous definitions for the mathematical tools used in the main text and additionally clarifies necessary hypotheses.

We start with the fundamental domain. Assume X has dimension n . Let d be the dimension of a generic orbit of G in X . Let ν be the $(n - d)$ dimensional Hausdorff measure in X .

Definition 2 (Fundamental Domain, Definition 4.1 in [\[WZP+23\]](#)). A closed subset F of X is called a fundamental domain of G in X if X is the union of conjugates of F , i.e., $X = \cup_{g \in G} gF$, and the intersection of any two conjugates has measure 0 under ν .

Next, we note that our proof of [Theorem 1](#) satisfies the integrability assumption on the fundamental domain F and orbits Gz established in [\[WZP+23\]](#):

Assumption 2 (Integrability Hypothesis, Sec. A in [\[WZP+23\]](#)). The fundamental domain F and orbit Gx are differentiable manifolds and the union of all pairwise intersections $\cap_{g_1 \neq g_2} (g_1 F \cap g_2 F)$ has measure zero.

We now provide more formal definitions for $\mathbb{E}_{Gx}[f]$ and $\mathbb{V}_{Gx}[f]$ used in [Proposition 2](#). Denote by $q(z) = \frac{p(z)}{p(Gx)}$ the density of the orbit Gx so that $\int_{Gx} q(z) dz = 1$. The mean and variance of a function f on Gx are given by

$$\mathbb{E}_{Gx}[f] = \int_{Gx} q(z) f(z) dz, \quad \mathbb{V}_{Gx}[f] = \int_{Gx} q(z) \|\mathbb{E}_{Gx}[f] - f(z)\|_2^2 dz.$$

A.4 Rayleigh Quotient Sensitivity

We include results from [\[FH24; DZW24\]](#) that illustrate the sensitivity of the Rayleigh quotient to small perturbations of the input, such as Taylor series truncation errors. While the hypotheses are stronger than what we may actually see in practice, the following proposition provides an intuition for the Rayleigh quotient sensitivity.

Proposition 4 (Proposition 4 in [\[FH24\]](#)). *Suppose $\mathbf{u} = \mathbf{x} + \mathbf{e}$ is an approximate eigenvector corresponding to a simple eigenvalue $\lambda \neq 0$ of a symmetric A , with $\|\mathbf{x}\| = 1$, $\mathbf{e} \perp \mathbf{x}$, and $\varepsilon = \|\mathbf{e}\|$. Then, up to $\mathcal{O}(\varepsilon^4)$ -terms, for the sensitivity of the Rayleigh quotient (as a function of \mathbf{u}) it holds that*

$$\min_{\lambda_i \neq \lambda} \frac{|\lambda_i - \lambda|}{|\lambda_i|} \varepsilon^2 \lesssim \frac{|R_{\mathcal{G}}(\mathbf{u}) - \lambda|}{|\lambda|} \lesssim \max_{\lambda_i \neq \lambda} \frac{|\lambda_i - \lambda|}{|\lambda_i|} \varepsilon^2.$$

This indicates that the Rayleigh quotient sensitivity is quadratic in perturbations ε . For $\varepsilon < 1$, this means that the sensitivity of the Rayleigh quotient is even less than the truncation error. We also have the following results from [\[DZW24\]](#):

Proposition 5 (Theorem 1 in [\[DZW24\]](#)). *For any given graph G , if there exists a perturbation Δ on \mathbf{L} , the change of Rayleigh quotient can be bounded by $\|\Delta\|_2$.*

Proposition 6 (Theorem 2 in [\[DZW24\]](#)). *For any given graph G , if there exists a perturbation δ on \mathbf{x} , the change of Rayleigh quotient can be bounded by $2\mathbf{x}^T \mathbf{L} \delta + o(\delta)$. If δ is small enough, in which case $o(\delta)$ can be ignored, the change can be further bounded by $2\mathbf{x}^T \mathbf{L} \delta$.*

The results from [\[DZW24\]](#) state fewer hypotheses. [Proposition 5](#) outlines a bound similar to [Proposition 4](#) in that they are both related to the norm of the perturbing vector, and [Proposition 6](#) states an alternative bound related to the graph’s Laplacian.

B Simulated Heat Diffusion Dataset

A sample graph data point is given in [Fig. 3](#).

C Gauge Equivariant GNN Training Details

Peace and love

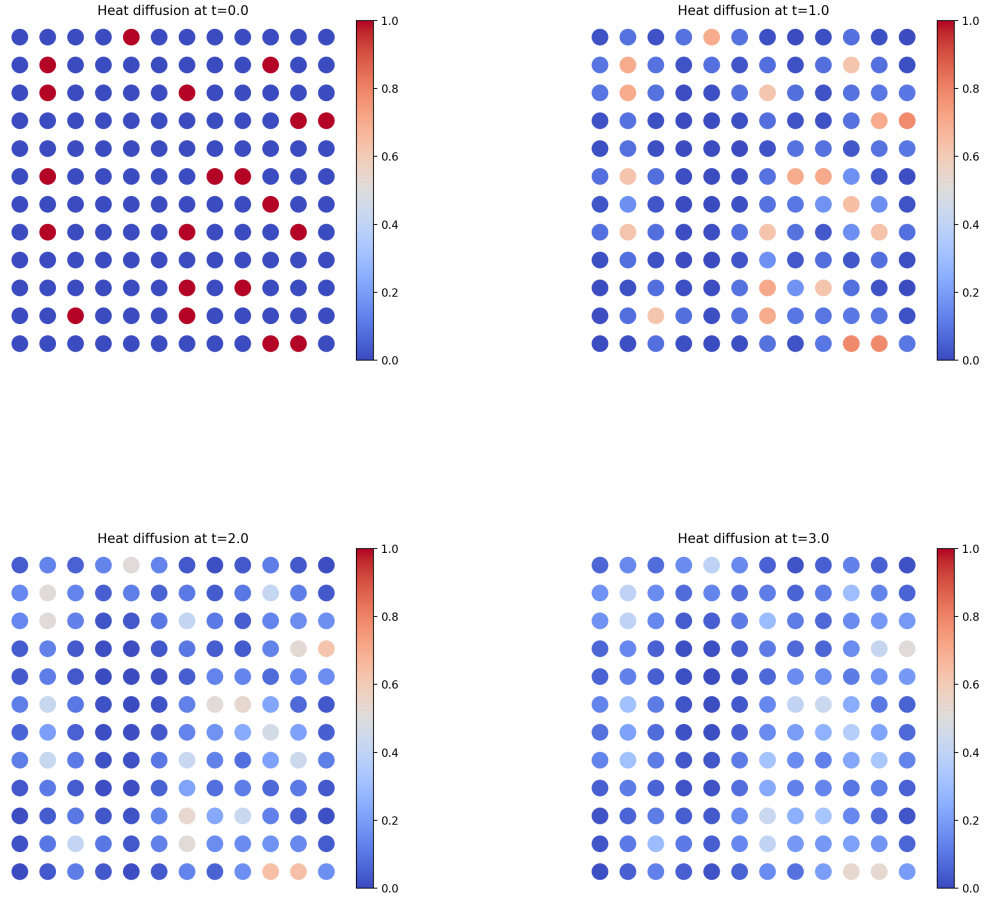


Figure 3: Sample heat diffusion process on a grid discretized as a graph. Node neighbors are the nodes that sit adjacent in the grid.