SMOOTHNESS ERRORS IN DYNAMICS MODELS AND HOW TO AVOID THEM

Edward Berman,* **Luisa Li***, **Jung Yeon Park**, **Robin Walters**
Geometric Learning Lab, Northeastern University
berman.ed@northeastern.edu

ABSTRACT

Modern neural networks have shown promise for solving partial differential equations over surfaces, often by discretizing the surface as a mesh and learning with a mesh-aware graph neural network. However, graph neural networks suffer from oversmoothing, where a node's features become increasingly similar to those of its neighbors. Unitary graph convolutions, which are mathematically constrained to preserve smoothness, have been proposed to address this issue. Despite this, in many physical systems, such as diffusion processes, smoothness naturally increases and unitarity may be overconstraining. In this paper, we systematically study the smoothing effects of different GNNs for dynamics modeling and prove that unitary convolutions hurt performance for such tasks. We propose relaxed unitary convolutions that balance smoothness preservation with the natural smoothing required for physical systems. We also generalize unitary and relaxed unitary convolutions from graphs to meshes. In experiments on PDEs such as the heat and wave equations over complex meshes and on weather forecasting, we find that our method outperforms several strong baselines, including mesh-aware transformers and equivariant neural networks. Our code is available at https://github.com/EdwardBerman/rayleigh_analysis.

1 INTRODUCTION

Solving partial differential equations (PDEs) is crucial across many scientific and engineering domains, including acoustics, fluid dynamics, and electrodynamics. Recently, neural networks have been explored as alternatives to analytic and traditional numerical methods for solving PDEs. Neural networks offer faster inference (Cui et al., 2024), discretization free solutions (Li et al., 2021), better robustness to partial observability (Schlaginhaufen et al., 2021; Huang et al., 2024; Morel et al., 2025), and synergy with existing finite element methods (Gupta & Lermusiaux, 2023).

However, neural network models often have architectural biases that hurt their ability to model certain dynamics. In particular, many deep learning methods solve PDEs by discretizing the domain into a grid or mesh and modeling the solution using a graph neural network (e.g., Janny et al., 2023; Park et al., 2023). Unfortunately, graph neural networks (GNNs) tend to oversmooth (Li et al., 2018), where adjacent node features become increasingly similar over successive iterations of message passing. The phenomenon of oversmoothing occurs in a variety of settings (Cai & Wang, 2020; Bodnar et al., 2022; Keriven, 2022; Rusch et al., 2023; Balla, 2023; Kiani et al., 2024; Arroyo et al., 2025; Su & Wu, 2025; Mishayev et al., 2025) and hampers the performance of deep GNNs.

To address oversmoothing, Kiani et al. (2024) propose using unitary graph convolutions, which constrain weight matrices to be unitary. This ensures that the linear transformations preserve norms and remain invertible, improving network stability. They also prove that unitary convolutions prevent oversmoothing by preserving the Rayleigh quotient, a measure of graph smoothness. However, this poses a new problem: many dynamics problems commonly solved using GNNs require *some* amount of smoothing. For example, heat diffusion on graphs and meshes naturally smooths the input node features. Using unitary graph convolutions in such problems would result in undersmoothing and does not give a complete solution to smoothness errors.

*First two authors contributed equally.

In this work, we first theoretically characterize the limitations of unitary convolutions for dynamics problems. In particular, we derive a lower bound on the approximation error of unitary functions and show that unitary functions are overconstrained for dynamical systems where the solution's norm has high angular dependence. To address this issue, we propose *relaxed unitary convolutions*, which balance smoothness preservation with modeling fidelity, outperforming existing methods on dynamic systems that require natural smoothing. We also generalize both the Rayleigh quotient and unitary convolution framework from graphs to meshes so that relaxations can be applied in this setting. Finally, we systematically investigate smoothness tendencies of different mesh-GNN architectures and find that our mechanism for approximately preserving smoothness is key to successful modeling, providing equal or greater improvement to other inductive biases such as equivariance.

In summary, our contributions are the following: **(i)** Derive a lower bound on the approximation error of unitary functions, demonstrating that they are overly restrictive when predicting dynamics with high angular dependence in the solution's norm (Sec. 4). **(ii)** Introduce relaxed unitary convolutions that balance accuracy with smoothness preservation, and extend both the Rayleigh quotient and unitary convolution framework to meshes (Sec. 5). **(iii)** Empirically analyze the smoothness behavior of various GNN architectures on complex dynamical systems, showing that controlling smoothness can match or outperform strong baselines (Sec. 6).

2 Related Works

Oversmoothing and undersmoothing in GNNs. Like Kiani et al. (2024), our work quantifies the effect of neural networks on the Rayleigh quotient (Chung, 1997) of a graph. Kiani et al. (2024) prove that unitary functions, and in particular the unitary convolution network, strictly preserve the Rayleigh quotient and therefore the smoothness of input graphs. However, we show both theoretically and empirically how this property can be overconstraining in GNNs. Other approaches to quantifying smoothness in PDE solutions have used the Matérn kernel (Borovitskiy et al., 2021; Daniels et al., 2025) or decay rate exponents (Kulick et al., 2025), but none have considered the Rayleigh quotient for dynamics models as we do.

Our work is perhaps most similar to Keriven (2022), who also point out that some smoothing can be useful for certain regression tasks but do not consider dynamics modeling specifically. Similarly, Li et al. (2018) point out that GCNs (Kipf & Welling, 2017) can be understood as a special case of Laplacian smoothing, which is a key reason why GCNs work at all. In fact, Kipf & Welling (2017) argue that their architecture can be understood as a differentiable and parameterized generalization of the 1-dim Weisfeiler-Lehman algorithm (Leman & Weisfeiler, 1968), indicating that even randomly initialized GCNs can be performant due to the way they smooth information throughout the network. Despite these findings, there is comparatively less work studying the role of smoothness in spatio-temporal modeling tasks. While Marisca et al. (2025) study issues with message-passing based GNNs for spatio-temporal modeling, they focus on *oversquashing*, where information fails to propagate to distant nodes, whereas our work addresses *oversmoothing* and *undersmoothing*.

Dynamics modeling over graphs and meshes. Our work focuses on dynamics modeling where PDE solutions are discretized as signals on graphs and meshes through the lens of smoothness. Many physical systems, such as wave propagation (d'Alembert, 1747), heat diffusion (baron de Fourier, 1822), phase fields (Cahn & Hilliard, 1958; Li et al., 2024), fluid flows (Constantin & Foiaş, 1988; Anandkumar et al., 2020), and climate systems (Ghil & Simonnet, 2020) can be described by systems of PDEs. Deep learning based approaches are increasingly used to solve these PDEs in these domains where numerical solving is difficult (Wang et al., 2020; Cranmer et al., 2020; Anandkumar et al., 2020; Li et al., 2021; Mustafa et al., 2021; Cai et al., 2021; Maurizi et al., 2022; Park et al., 2023; Liu et al., 2024; Yu & Wang, 2024; Daniels & Rigollet, 2025). For PDE solving on meshes, these dynamics can be formulated extrinsically by embedding the manifold into Euclidean space (Satorras et al., 2021; Pfaff et al., 2021), or intrinsically by defining evolution directly in the coordinates of local tangent spaces (Cohen et al., 2019; de Haan et al., 2021; Mitchel et al., 2021; 2022; Basu et al., 2022; Park et al., 2023; Suk et al., 2024; Mitchel et al., 2024). While Cohen et al. (2019), Pfaff et al. (2021), and Suk et al. (2024) contain isolated experiments related to dynamics modeling, only our work and Park et al. (2023) study how the choice of Euclidean versus locally defined coordinate representations in the network can affect convergence to PDE solutions. Furthermore, our work is

distinct from Park et al. (2023) in that only we directly assess how these design choices affect neural network *smoothing behavior*.

Benchmarking PDE Surrogate Models. While the physical symmetries of many dynamical systems are well understood (Olver, 1993; Wang et al., 2021; Borovitskiy et al., 2021), smoothness has received less attention. The performance of deep dynamics models is typically measured either via quantitative error metrics against the ground truth or their preservation of underlying physical laws, such as spectral energy errors (Wang et al., 2021) or equivariance errors (Wang et al., 2021; 2022a;b). Our work is novel in our application of the Rayleigh quotient in quantifying the smoothing effect of trained GNN dynamics models. Furthermore, we are among the first to design architectures with inductive biases that encourage the model to match the Rayleigh quotient of the labeled graphs. Other works have explored using the Rayleigh quotient as an auxiliary loss (Rowan et al., 2025), as positional encodings (Dong et al., 2024), and as a hard constraint to preserve smoothness regardless of the true labels Kiani et al. (2024). In contrast, only our work and the work of Shao et al. (2024) use architectural inductive biases to match the smoothness of labeled graphs, and only our work evaluates how well the true smoothness of dynamical systems is recovered.

3 Background

We first recall the definition of the Rayleigh quotient, a measure of smoothness on graphs, and provide background on unitary convolutions and their invariance to the Rayleigh quotient. We also introduce the mesh data type, which we later use to extend the unitary convolution framework from graphs to meshes.

3.1 Rayleigh quotient

To measure the smoothness of a signal on a graph, we use the Rayleigh quotient as defined in Chung (1997).

Definition 1 (Rayleigh quotient, (Chung, 1997))**.** Given an undirected graph $\mathcal{G} = (V, E)$ with $|V| = n$ nodes and 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 $\mathbf{D}_{ii} = d_i$ the degree of node i. Let $s\colon V \to \mathbb{C}^d$ be a function from nodes to features. Denote by $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ the normalized adjacency matrix and $\mathbf{X} \in \mathbb{C}^{n\times d}$ a matrix with the i-th row set to feature vector $s(i)$. The Rayleigh quotient is

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

We will often abbreviate the Laplacian as $\mathbf{L} = (\mathbf{I} - \tilde{\mathbf{A}})$.

Intuitively, the Rayleigh quotient measures the mean difference in node features for adjacent nodes. A graph with identical node features has a Rayleigh quotient of zero.

3.2 Unitary Convolution

Kiani et al. (2024) define two different models that preserve the Rayleigh quotient using unitary functions, which satisfy $\mathbf{U}^\dagger\mathbf{U} = \mathbf{I}$. In particular, they define the separable unitary convolution

$$f^{\text{sep}}_{\text{UniConv}}(\mathbf{X}; \mathbf{A}) = \exp(\mathbf{i}\mathbf{A}\mathbf{t})\mathbf{X}\mathbf{U}, \quad \mathbf{U}^\dagger\mathbf{U} = \mathbf{I} \tag{2}$$

and the Lie unitary convolution

$$f^{\text{Lie}}_{\text{UniConv}}(\mathbf{X}; \mathbf{A}) = \exp(\mathbf{A}\mathbf{X}\mathbf{W}), \quad \mathbf{W} = -\mathbf{W}^\dagger \tag{3}$$

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 unitary convolutions are mathematically constrained to preserve the Rayleigh quotient, i.e., $R_{\mathcal{G}}(\mathbf{X}) = R_{\mathcal{G}}(f_{\text{UniConv}}(\mathbf{X}))$ where f_{UniConv} is either seperable or Lie. Please see Sec. A.2 for details.

3.3 MESH DATA

A (triangular) mesh $\mathcal{M}$ 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 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. The definition of the manifold condition for a mesh is given in Definition 9 (Sec. A.7).

4 THEORY: UNITARY MODELS ARE OVERCONSTRAINED

While unitary models on graphs can be useful because they preserve the Rayleigh quotient, this section illustrates how unitary models can be *overly* constrained. In particular, we derive an approximation error lower bound that clarifies the approximation limits of unitary models. We start by establishing our unitary approximation learning framework.

4.1 PRELIMINARIES

Let $Z = \mathbb{C}^n$ be a domain with data probability density $p\colon Z \to \mathbb{R}$. Let $u\colon \mathbb{C}^n \to \mathbb{C}^n$ be a unitary function and let $f\colon \mathbb{C}^n \to \mathbb{C}^n$ be the target function. Denote the regression error by

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

Group Invariance. Our main result relies on the theory of approximation error for group invariant functions h. We review these concepts in detail in Sec. A.4 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 $Gz = \{gz\colon g \in G\}$ be the orbit of z. A *fundamental domain* F of a group G in Z is a set of orbit representatives. The domain Z can be written as the union of conjugates, $Z = \cup_{g\in G} gF$, where the conjugate is defined $gF = \{gz\colon z \in F\}$. Denote the integrated density on an orbit by $p(Gz) = \int_{Gz} p(z)dz$. Finally, denote the variance of a function f on an orbit Gz by $\mathbb{V}_{Gz}[f]$. The approximation error lower bound for an invariant function is given by the following proposition.

Proposition 1 (Theorem 4.8 in Wang et al. (2023))**.** *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 1 was initially stated for real-valued functions in Wang et al. (2023), but can be applied to complex valued functions without loss of generality. Furthermore, Wang et al. (2023) provide numerical evidence that Proposition 1 is a tight bound.

4.2 UNITARY APPROXIMATION ERROR LOWER BOUND

In this subsection, we state our main theoretical result, which demonstrates that unitary neural networks are overconstrained when the norm of the ground truth function has a high angular dependence. Recall the definition of $\text{SU}(n)$, the group of rotations in $\mathbb{C}^n$:

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

We can now give an approximation error lower bound for unitary models. See Sec. A.5 for the proof.

Theorem 1. *Let F be a fundamental domain of* $\text{SU}(n)$ *in Z, e.g. $F = \{te\colon t \in \mathbb{R}_+\}$ where e is a standard basis vector of $\mathbb{C}^n$. The approximation error of u of f has lower bound*

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

The proof of Theorem 1 uses the reverse triangle inequality before applying Proposition 1. 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 where the norm of u is constant. Our result suggests that unitary functions can be particularly overly constraining when the norm of f has a high angular dependence.

5 UNITARY CONVOLUTION CONSTRAINT RELAXATION

Since Theorem 1 informs us that a unitary convolution network may be overconstraining when the ground truth is not perfectly smoothness preserving, we propose two methods for relaxing unitary convolutions and describe how to extend these architectures from graphs to meshes. We name these methods Taylor truncation and encoder-decoder. The Taylor truncation method allows precise control of the extent of relaxation, whereas the encoder-decoder method scales more easily with the number of parameters in the network. The Taylor truncation method is especially useful in cases where the true smoothness is known a priori, in which case theoretical results from the literature (see Sec. A.8) can inform what truncation order $\mathbf{T}_{\max}$ is needed to achieve enough relaxation. We will use the first relaxation for a motivating experiment in Sec. 6.1 and the second for a more challenging set of tasks in Sec. 6.2. We will use f_{Layer} to define individual layers and SMALL CAPS TEXT to define architectures constructed from those layers.

5.1 RELAXED UNITARY CONVOLUTION VIA TAYLOR TRUNCATION

We relax Lie unitary convolution using early truncation of the Taylor series approximations used in Eq. 3. We note that Kiani et al. (2024) propose their own constraint relaxation by allowing $\mathbf{U}$ to be unconstrained in Eq. 2, and that our approach instead relaxes Eq. 3. This allows us to isolate the architectural component that alters the Rayleigh quotient. In contrast, the relaxation in Kiani et al. (2024) can be achieved via two different mechanisms: early Taylor series truncation of the matrix exponential and letting $\mathbf{U}$ remain unconstrained in Eq. 2. The downside of this approach is that the relative contributions of each are difficult to measure and it is therefore harder to tune the extent of the relaxation. Our relaxation is simply Taylor series truncation of Lie unitary convolution (Eq. 3). Instead of approximating the matrix exponential using enough Taylor series terms so that the truncation error is negligible, we truncate at some $T = \mathbf{T}_{\max}$ where $\mathbf{T}_{\max}$ controls the extent of the relaxation. Our Taylor-relaxed unitary convolution layer is defined

$$f_{\text{Relaxed}}(\mathbf{X}; \mathbf{A}, \mathbf{T}_{\max}) = \sum_{i=0}^{\mathbf{T}_{\max}} \frac{1}{i!} \mathbf{L}^i(\mathbf{X}) \tag{4}$$

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 $T \to \infty$ we recover standard Lie unitary convolution in Eq. 3.

Motivated by the desire to find an appropriate $\mathbf{T}_{\max}$ that applies only a small perturbation to the Rayleigh quotients of input graphs, in Sec. B.2 we conduct a sensitivity analysis of the Rayleigh quotient to different Taylor series truncations. Consistent with, we find that $\mathbf{T}_{\max} = 10$ is sufficient to preserve the Rayleigh quotient. We will refer to models constructed from relaxed layers in Eq. 4 as R-UNIGRAPH .

5.2 RELAXED UNITARY CONVOLUTIONAL MODELS VIA ENCODER-DECODER

In this section, we note a limitation of relaxed unitary convolution via Taylor truncation that makes it difficult to scale, and propose an alternative relaxation method that addresses this. Since unitary layers preserve the channel dimension of the node features, the only way to increase the number of parameters in the network is to increase the number of unitary layers. This means scaling unitary convolutional models requires making the model very deep which can make training unstable. As an alternative, we propose an encoder-decoder method which first zero-pads the input node features to the desired hidden dimension. This allows us to increase parameter count without increasing depth. Zero padding also trivially preserves the Rayleigh quotient since it preserves norms. Concretely, we define our zero padding function $f_{\text{pad}} \colon \mathbb{R}^{n \times d_{\text{in}}} \to \mathbb{R}^{n \times d_{\text{out}}}$ by $\mathbf{X} \mapsto \mathbf{X} \oplus 0$, where 0 is the $\mathbb{R}^{n \times (d_{\text{out}} - d_{\text{in}})}$ zero matrix. We define our k-layer encoder E by

$$E = f^{(k)}_{\text{UniConv}} \circ \cdots \circ f^{(1)}_{\text{UniConv}}(f_{\text{pad}}(\mathbf{X}), \mathbf{A}) \tag{5}$$

where f_{UniConv} is either separable or Lie. The decoder layer D then serves two purposes: (**i**) map to the target node feature dimension and (**ii**) break the unitary constraint. The decoder layer can be any network, such as an MLP or GCN.

5.3 Relaxed Unitary Convolution on Meshes

We now generalize unitary convolutional models, relaxed unitary convolutional models, and the mathematical definition of Rayleigh quotient from graphs to meshes. This enables us to solve dynamics problems on manifolds by discretizing them as meshes, such as testing the thermal stability of mechanical parts or weather forecasting on the Earth. In particular, we prove that under modest assumptions on the mesh triangulation, unitary convolution with a weighted adjacency matrix preserves the Rayleigh quotient on meshes (Definition 3); enabling generalization of unitary and relaxed unitary models to meshes.

Mesh Rayleigh Quotient. We first generalize the Rayleigh quotient from graphs to meshes by using the mesh Laplacian instead of the graph Laplacian. The Laplacian on a mesh is typically defined as the symmetric cotangent Laplacian (Reuter et al., 2009) given in Eq. 6, which approximates the Laplace-Beltrami operator for the continuous manifold which the mesh discritizes. For a scalar function s defined on nodes,

$$(\tilde{\mathbf{L}}(s))_i = \frac{1}{2A_i} \sum_{j \in \mathcal{N}(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(s_j - s_i) \tag{6}$$

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. We use the barycentric cell area for A_i. We note that it is invalid to define the mesh Rayleigh quotient by replacing $\mathbf{L}$ in Eq. 1 with the symmetric cotangent Laplacian $\tilde{\mathbf{L}}$ in Eq. 6. The cotangent weights in Eq. 6 may be negative, which means that the Rayleigh quotient is no longer a valid measure of smoothness (Definition 1, Rusch et al., 2023). To address this, we use the *Robust Laplacian* (Sharp & Crane, 2020), which performs a minimal edge rewiring of the mesh so that the cotangent weights obey the *intrinsic Delaunay criterion*.

Definition 2 (Intrinsic Delaunay Criterion, (Bobenko & Springborn, 2007))**.** For all faces connected by an edge (i, j) with opposite angles α_{ij} and β_{ij}, $\alpha_{ij} + \beta_{ij} \leq \pi$.

Concretely, this means that our Laplacian weights are both symmetric and the off-diagonals are nonnegative. Denote by $\mathcal{W}$ the cotangent weights

$$\mathcal{W}_{ij} = \begin{cases} \frac{1}{2}(\cot \alpha_{ij} + \cot \beta_{ij}), & j \in \mathcal{N}(i) \\ -\sum_{k \in \mathcal{N}(i)} \mathcal{W}_{ik}, & i = j \\ 0, & \text{Otherwise.} \end{cases} \tag{7}$$

We define a novel Rayleigh quotient for meshes as follows.

Definition 3 (Mesh Rayleigh Quotient)**.** Let $\mathcal{M} = (\mathcal{V}, \mathcal{E}, \mathcal{F})$ be a mesh with $|V| = n$ nodes. Denote by $\mathcal{W}$ the cotangent weights corresponding to the Robust Laplacian $\tilde{\mathbf{L}}$. Denote by $\mathcal{E}'$ the rewired edge set given by $\mathcal{E}' = \{(u, v) \colon \mathcal{W}_{uv} \neq 0\}$. Let s and $\mathbf{X}$ be the same as in Definition 1. The mesh Rayleigh quotient is defined

$$R_{\mathcal{M}}(\mathbf{X}) = \frac{1}{2} \frac{\sum_{(u,v) \in \mathcal{E}'} \mathcal{W}_{uv} \left\| \frac{s(u)}{\sqrt{d_u}} - \frac{s(v)}{\sqrt{d_v}} \right\|^2}{\sum_{w \in V} \|s(w)\|^2} = \frac{\operatorname{Tr}(\mathbf{X}^\dagger \tilde{\mathbf{L}} \mathbf{X})}{||\mathbf{X}||_F^2}.$$

Unitary Mesh Convolution. We now make similar modifications to generalize unitary convolution from graphs to meshes. Specifically, we modify the functions in Eq. 2 and Eq. 3 by incorporating the cotangent weights (Eq. 7) into the normalized adjacency matrix $\tilde{\mathbf{A}}$. This edge weighting been shown to improve PDE solving numerically (Crane et al., 2017; Sharp & Crane, 2020), but has not been used to augment deep dynamics models. In order to prove that incorporating these weights preserves the Rayleigh quotient given by Definition 3, we assume the mesh already satisfies the Delaunay criterion.

Assumption 1 (Mesh Weights Obey the Delaunay Criterion)**.** For a mesh $\mathcal{M}$, the mesh is manifold and all angles obey the Delaunay Criterion given by Definition 2.

We note that there are existing triangulation strategies that a practitioner can use to ensure that mesh edges satisfy this criterion as a standard data preprocessing step (Huang et al., 2018; Sharp & Crane, 2020), see Sec. A.7 for details. With this assumption, we will now define unitary mesh convolution.

Let $\mathbf{D}$ be the degree matrix defined by $\mathbf{D}_{ii} = \sum_{i \neq j} \mathcal{W}_{ij}$. Let $\odot$ represent the Hadamard product which performs element-wise matrix multiplication. Let $\tilde{\mathbf{A}}$ be the normalized adjacency matrix $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2} (\mathcal{W} \odot \mathbf{A}) \mathbf{D}^{-1/2}$. We define separable unitary mesh convolution as

$$f^{\text{Sep}}_{\text{UniMeshConv}}(\mathbf{X}; \mathbf{A}, \mathcal{W}) = \exp(i\tilde{\mathbf{A}})\mathbf{X}\mathbf{U} \tag{8}$$

and Lie unitary mesh convolution as

$$f^{\text{Lie}}_{\text{UniMeshConv}}(\mathbf{X}; \mathbf{A}, \mathcal{W}) = \exp(\tilde{\mathbf{A}}\mathbf{X}\mathbf{W}) \tag{9}$$

where $\mathbf{U}\mathbf{U}^{\dagger} = \mathbf{I}$ and $\mathbf{W} + \mathbf{W}^{\dagger} = \mathbf{0}$. The following Corollary (proven in Sec. A.6) states that Eq. 8 and Eq. 9 preserve the Rayleigh quotient on meshes.

Corollary 1 (Corollary to Proposition 3)**.** *Given a mesh $\mathcal{M}$ with normalized adjacency matrix $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}(\mathcal{W} \odot \mathbf{A})\mathbf{D}^{-1/2}$ that satisfies Assumption 1, the mesh Rayleigh quotient is invariant under normalized unitary or orthogonal graph convolution, i.e. $R_{\mathcal{M}}(\mathbf{X}) = R_{\mathcal{M}}(f_{\text{UniMeshConv}}(\mathbf{X}))$ where $f_{\text{UniMeshConv}}$ is either separable or Lie.*

Relaxed Unitary Mesh Convolution. We create a network architecture by coupling the relaxation in Sec. 5.2 with Lie unitary mesh convolution in Eq. 9. Concretely, nodes are first zero padded. An encoder $E^{(k)}$ (Eq. 5) is constructed from k layers of Lie unitary mesh convolution (Eq. 9), and a MLP or GCN decoder layer D maps to the target. We name our relaxed model R-UNIMESH. R-UNIMESH uses the norm preserving GroupSort activation from Anil et al. (2019) to introduce nonlinearity. R-UNIMESH also uses *orthogonal* weights, since our modeling tasks in Sec. 6 are real valued.

6 EXPERIMENTS

6.1 MOTIVATING EXPERIMENT: HEAT FLOW ON GRID GRAPHS

In this section, we motivate the use of relaxed unitary models by showing that R-UNIGRAPH is able to outperform both normal and Lie unitary graph convolution on predicting heat diffusion. The Taylor truncation method is key to balancing the smoothness preservation of unitary models with the flexibility to capture the true smoothness of the target heat graph.

Heat Diffusion Setup. We use `PyGSP` (Defferrard et al., 2017) to simulate heat diffusion on $10,000$ two-dimensional grids, each initialized with 20 randomly placed heat sources. Denote by $H : \mathbb{R}_+ \to \mathbb{R}^n$ a function that maps time t to the heat distribution of the graph with n nodes. In other words, the heat field on the graph is represented by a feature vector in $\mathbb{R}^{n \times 1}$. In particular, $H(t) = e^{-\tau t \mathbf{L}} H(0)$ where τ is a diffusivity constant, $\mathbf{L}$ is the graph Laplacian, and $H(0)$ is the initial heat values across the graph. The task is to predict the heat distribution on the graph at time $t = 4$ given the heat distribution at time $t = 3$. We denote the target heat field as $\mathbf{Y} = H(4)$. See Sec. B.1 for further dataset details. We compare the performance in terms of MSE loss and mean Rayleigh quotient error for three models: GCN, R-UNIGRAPH, and a Lie unitary model. We use $\mathbf{T}_{\max} = 3$ for the relaxed model. The mean Rayleigh quotient error is given by $\text{MRE}(f) = |\overline{R_{\mathcal{G}}(f(\mathbf{X}))} - \overline{R_{\mathcal{G}}(\mathbf{Y})}|$.

Model	MSE (↓)	MRE (↓)
GCN	$1.08 \cdot 10^{-2}$	$5.99 \cdot 10^{-2}$
Lie Uni	$0.14 \cdot 10^{-2}$	$8.86 \cdot 10^{-2}$
R-UNIGRAPH **(Ours)**	$\mathbf{0.11 \cdot 10^{-2}}$	$\mathbf{2.07 \cdot 10^{-2}}$

Table 1: Best MSE and Rayleigh quotient error over an ensemble of models for a GCN, a Lie unitary convolution network, and R-UNIMESH. The best performance is bold.

Results. We see in Tab. 1 and Fig. 1 that the relaxed model significantly outperforms the GCN and also outperforms the Lie unitary model. Moreover, the relaxed model is best able to produce graphs whose smoothness matches that of the true labels. Our results are predicted by Proposition 7 in Kiani et al. (2024), which is provided in Sec. A.2. The proposition demonstrates that GCNs with standard weight initialization are likely to increase the smoothness of input signals on the graph, as measured by the Rayleigh quotient. The nuance exhibited by our experiment is that not only do GCNs smooth the input signal, but they *oversmooth* compared to the target signal. This is true even when some smoothing is desirable, i.e. the output signal is smoother than the input. In contrast, R-UNIMESH is

often initialized in a state of being *undersmooth* and is able to learn how to increase the smoothness of input signals during training to match the smoothness of the target signal. We will see in Sec. 6.2 that these insights will hold for heat flow on more intricate mesh datasets.

6.2 Dynamics on Mesh Manifolds

We now consider a set of more challenging and realistic tasks and show that R-UniMesh is competitve with strong baselines and outperforms all other models on diffusive dynamics. Specifically, our experiment reveals the following practical conclusions: **(i)** R-UniMesh performs as well as strong baselines such as mesh-aware transformers and equivariant neural networks. **(ii)** R-UniMesh is especially strong on heat diffusion tasks. **(iii)** More geometric inductive biases such as unitarity or equivariance are required for strong performance on unseen meshes with complex geometries. We support these conclusions with datasets for dynamics modeling on mesh manifolds, including PDE solving on the `PyVista` (Sullivan & Kaszynski, 2019) meshes from Park et al. (2023) and weather forecasting on the Earth mesh from WeatherBench2 (WB2) (Rasp et al., 2024). Details on baselines are provided in Sec. C. We compare these baselines with R-UniMesh.

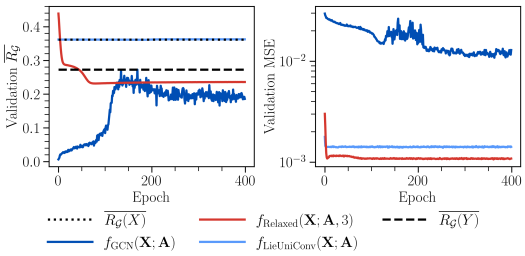


Figure 1: **Left:** The dotted lines indicate the mean Rayleigh quotient for input heat graphs at $T = 3$ and target graphs at $T = 4$. We also show the mean Rayleigh quotient for the best performing GCN, R-UniGraph, and Lie unitary models. R-UniGraph is best at capturing the true smoothness. **Right:** Validation MSE of the same three models. R-UniGraph has the best performance.

Datasets. The first task is to auto-regressively predict the solution to the heat, wave, and Cahn-Hilliard equations on test meshes given an initial condition. These equations are defined formally in Sec. D.1. We use the same `PyVista` meshes generated in Park et al. (2023) as data. These meshes are highly intricate and test the models' ability to handle nonlinear dynamics on complicated geometries. We use five different initializations for each test mesh. Further dataset and training details are provided in Sec. D.1 and Sec. D.2. We will refer to this dataset as MeshPDE in the remainder of the paper.

Metric	GCN	GemCNN	R-UniMesh (Ours)	EMAN	Transformer	MPNN	EGNN	Hermes
Heat ($\alpha = 1$)								
NRMSE (↓)	–	–	$\mathbf{51.9 \pm 3.6}$	73.50 ± 3.8	92.5 ± 5.6	99.45 ± 4.8	344.25 ± 110.5	73.02 ± 4.7
SMAPE (↓)	–	375.4 ± 0.53	$\mathbf{79.7 \pm 5.6}$	110.9 ± 13.3	213.9 ± 2.7	223.6 ± 1.5	319.33 ± 7.59	107.6 ± 7.4
RE (↓)	–	52.21 ± 9.4	$\mathbf{9.1 \pm 7.4}$	14.2 ± 1.4	46.0 ± 3.7	76.06 ± 3.6	81.5 ± 8.77	39.76 ± 4.7
Wave ($c = 1$)								
NRMSE (↓)	–	–	$\mathbf{236.5 \pm 6.4}$	281.3 ± 15.5	864.9 ± 184.9	563.6 ± 7.75	2280.1 ± 559.9	458.5 ± 13.0
SMAPE (↓)	–	318.8 ± 3.9	385.2 ± 1.2	$\mathbf{301.0 \pm 4.2}$	327.0 ± 4.4	318.0 ± 2.8	354.3 ± 11.0	316.4 ± 4.5
RE (↓)	–	107.9 ± 3.158	93.5 ± 25.4	73.57 ± 6.5	$\mathbf{48.0 \pm 7.9}$	139.3 ± 10.1	157.2 ± 14.8	70.03 ± 6.1
Cahn-Hilliard								
NRMSE (↓)	–	$\mathbf{121.2 \pm 1.8}$	123.9 ± 2.6	137.5 ± 0.69	144.4 ± 0.8	147.4 ± 11.36	1001.04 ± 5.73	122.0 ± 7.8
SMAPE (↓)	–	204.3 ± 2.4	167.3 ± 10.6	$\mathbf{143.7 \pm 2.5}$	191.7 ± 2.0	201.22 ± 32.79	336.5 ± 2.777	173.3 ± 4.3
RE (↓)	–	$\mathbf{10.68 \pm 3.3}$	18.9 ± 10.4	48.57 ± 3.49	27.42 ± 2.87	23.98 ± 6.51	41.8 ± 1.997	14.38 ± 11.5

Table 2: NRMSE, SMAPE, and RE averaged over all rollouts on all test meshes for the heat, wave, and Cahn-Hilliard. equations. Errors and standard deviations are reported over all test meshes and all initializations. Cells with a dash (–) correspond to models which do not converge for a given metric. Errors are scaled by $\times 196$, the number of rollout timesteps. R-UniMesh is competitive across all task and excels at solving the heat equation on unseen meshes.

The second task is weather forecasting using WB2 (Rasp et al., 2024), a widely used benchmark for data-driven global weather forecasting based on historic data. Specifically, we auto-regressively predict future weather conditions on Earth given an initial condition. A formal problem statement is in Sec. E.1. We train and evaluate our models on the ERA5 dataset from WB2, which is the curated

version of the ERA5 reanalysis data provided by the European Centre for Medium-Range Weather Forecasts (ECMWF) (Hersbach et al., 2020). We use 1.5 (240×120) degree spatial resolution data with a 6 hour temporal resolution, consistent with the evaluation performed in WB2. Further details on mesh construction can be found in Sec. E.3. We evaluate on two variables, temperature at pressure level 850 (T850) and geopotential at pressure level 500 (Z500). We take data from 2013-01-01 to 2019-12-31 UTC as training data. We use a smaller subset of the ERA5 data that is commonly used for other large scale data-based weather models due to compute constraints, but remain consistent to WB2 in evaluating on data from 2020-01-01 to 2020-12-31.

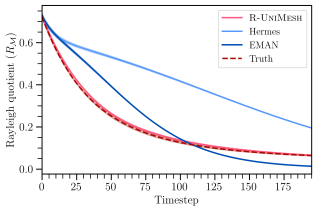


Figure 2: The Rayleigh quotient for each timestep on an unseen mesh for Hermes, EMAN, and R-UniMesh models. The R-UniMesh is the best at capturing the true smoothness for heat.

Evaluation. For MeshPDE, our metrics include normalized root mean squared error (NRMSE), symmetric mean absolute percentage error (SMAPE), and Rayleigh quotient errors aggregated over all time-steps. In particular, the Rayleigh error (RE) is given by:

$$\mathrm{RE}(f) = \frac{1}{\mathbf{T}_{\max}} \sum_{t}^{\mathbf{T}_{\max}} |R_{\mathcal{M}}(\mathbf{Y_t}) - R_{\mathcal{M}}(f(\mathbf{X_t}))|.$$

Further details on these metrics are provided in Sec. D.3 and we compare RE with more global smoothness metrics in Sec. D.5. We supplement these metrics with qualitative diagnostic figures in Sec. D.4 and videos in the supplementary material, showing in particular that Rayleigh errors are consistent with visual assessments of smoothness. For WB2, we report the root mean squared error (RMSE) and the anomaly correlation coefficient (ACC), both latitude weighted as recommended by the benchmark authors. RMSE measures forecast accuracy, while ACC is the Pearson correlation coefficient between forecast anomalies and ground-truth anomalies relative to a climatological baseline. Precise definitions are provided in Sec. E.

Results. Our main result is that our R-UniMesh model outperforms all baselines at solving highly diffusive PDEs and capturing the true smoothness while remaining competitive across all other tasks. For MeshPDE, this is indicated by the NRMSE, SMAPE, and RE for each of the models, which is given in Tab. 2. R-UniMesh performs particularly well on heat modeling, where it achieves the lowest error on all three metrics. On heat, R-UniMesh matches the true smoothness almost exactly, as seen in Fig. 2. The ability to match smoothness at each timestep illustrates that R-UniMesh is best at capturing the underlying differential structure of the PDE solution. We also show that the convergence and smoothness errors reported by our metrics agree visually with our qualitative diagnostics in Sec. D.4. Another important conclusion is that nearly all models are able to perform comparably well on the Cahn-Hilliard equation where the test mesh (toroid) is simple. The only models that perform poorly on this task are the GCN and EGNN models, which also struggle across all other tasks. This suggests that stronger geometric inductive biases such as unitarity or gauge equivariance are necessary for strong performance on unseen meshes with complex geometries. This is further supported by our results on WB2. As seen in Fig. 8 (Sec. E.4), the equivariant and unitary models show no significant advantages in this setting, where there is no cross mesh generalization. We also note that, despite restricting our training set size and model paramaeters due to compute limitations, our best performing models in Fig. 8 are comparable with the state of the art (Rasp et al., 2024) according to RMSE and ACC. In terms of RMSE, the best performing models are within a couple of degrees of SOTA even at 10 lead days. We show in Sec. E.4 that R-UniMesh is among the best models for the geopotential prediction task, though all models are below SOTA. Finally, we show in Sec. E.5 that R-UniMesh is competitive in terms of RE on both WB2 variables.

7 Conclusion

Our work clarifies the approximation limits of smoothness preserving (unitary) functions and unitary convolution networks and shows how constraint relaxations can aid performance on various dynamics modeling tasks on graphs and meshes. We contribute R-UniGraph and R-UniMesh, which provide SOTA performance on diffusive dynamics problems and excel at capturing the true smoothness of the system. Future work will explore using approximately unitary networks for solving PDEs under partial observability by using them as backbones for generative models.

REFERENCES

Anima Anandkumar, Kamyar Azizzadenesheli, Kaushik Bhattacharya, Nikola Kovachki, Zongyi Li, Burigede Liu, and Andrew Stuart. Neural operator: Graph kernel network for partial differential equations. In *ICLR 2020 workshop on integration of deep neural models and differential equations*, 2020.

Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International conference on machine learning*, pp. 291–301. PMLR, 2019.

Alvaro Arroyo, Alessio Gravina, Benjamin Gutteridge, Federico Barbero, Claudio Gallicchio, Xiaowen Dong, Michael M. Bronstein, and Pierre Vandergheynst. 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.

Michael Artin. *Algebra*. Birkhäuser, 1998.

Julia Balla. Over-squashing in riemannian graph neural networks. In *The Second Learning on Graphs Conference*, 2023. URL https://openreview.net/forum?id=UUnYi0yLcM.

Jean Baptiste Joseph baron de Fourier. *Théorie analytique de la chaleur*. Firmin Didot, 1822.

Sourya Basu, Jose Gallego-Posada, Francesco Viganò, James Rowbottom, and Taco Cohen. Equivariant mesh attention networks. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL https://openreview.net/forum?id=3IqqJh2Ycy. Expert Certification.

Alexander I Bobenko and Boris A Springborn. A discrete laplace–beltrami operator for simplicial surfaces. *Discrete & Computational Geometry*, 38(4):740–756, 2007.

Cristian Bodnar, Francesco Di Giovanni, Benjamin Chamberlain, Pietro Lio, and Michael Bronstein. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. *Advances in Neural Information Processing Systems*, 35:18527–18541, 2022.

Viacheslav Borovitskiy, Iskander Azangulov, Alexander Terenin, Peter Mostowsky, Marc Deisenroth, and Nicolas Durrande. Matérn gaussian processes on graphs. In *International Conference on Artificial Intelligence and Statistics*, pp. 2593–2601. PMLR, 2021.

John W Cahn and John E Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *The Journal of chemical physics*, 28(2):258–267, 1958.

Chen Cai and Yusu Wang. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020.

Shengze Cai, Zhicheng Wang, Sifan Wang, Paris Perdikaris, and George Em Karniadakis. Physics-informed neural networks for heat transfer problems. *Journal of Heat Transfer*, 143(6):060801, 2021.

Fan RK Chung. *Spectral graph theory*, volume 92. American Mathematical Soc., 1997.

Taco Cohen, Maurice Weiler, Berkay Kicanaoglu, and Max Welling. Gauge equivariant convolutional networks and the icosahedral cnn. In *International conference on Machine learning*, pp. 1321–1330. PMLR, 2019.

Peter Constantin and Ciprian Foiaş. *Navier-stokes equations*. University of Chicago press, 1988.

Keenan Crane, Fernando de Goes, Mathieu Desbrun, and Peter Schröder. Digital geometry processing with discrete exterior calculus. In *ACM SIGGRAPH 2013 Courses*, New York, NY, USA, 2013. ACM.

Keenan Crane, Clarisse Weischedel, and Max Wardetzky. The heat method for distance computation. *Communications of the ACM*, 60(11):90–99, 2017.

Miles Cranmer, Sam Greydanus, Stephan Hoyer, Peter Battaglia, David Spergel, and Shirley Ho. Lagrangian neural networks. In *ICLR 2020 Workshop on Integration of Deep Neural Models and Differential Equations*, 2020.

Qingjie Cui, Meina Zhang, Min Xiao, and Guoxi Ni. Transformer based deep learning accelerated numerical simulation for incompressible flow. *Physics of Fluids*, 36(12), 2024.

Jean le Rond d'Alembert. Recherches sur la courbe que forme une corde tenduë mise en vibration. https://www.exhibit.xavier.edu/oresme_2017Sept/4, 1747. Presented at the ORESME Reading Group Meeting, September 30, 2017.

Mara Daniels and Philippe Rigollet. Splat regression models. *arXiv preprint arXiv:2511.14042*, 2025.

Mara Daniels, Liam Hodgkinson, and Michael Mahoney. Uncertainty-aware diagnostics for physics-informed machine learning. *arXiv preprint arXiv:2510.26121*, 2025.

Pim de Haan, Maurice Weiler, Taco Cohen, and Max Welling. Gauge equivariant mesh cnns: Anisotropic convolutions on geometric graphs. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=Jnspzp-oIZE.

Michaël Defferrard, Lionel Martin, Rodrigo Pena, and Nathanaël Perraudin. Pygsp: Graph signal processing in python, 2017. URL https://github.com/epfl-lts2/pygsp/.

Xiangyu Dong, Xingyi Zhang, and Sibo 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.

Carlos Esteves. Theoretical aspects of group equivariant neural networks. *arXiv preprint arXiv:2004.05154*, 2020.

Giulia Ferrandi and Michiel E Hochstenbach. A homogeneous rayleigh quotient with applications in gradient methods. *Journal of Computational and Applied Mathematics*, 437:115440, 2024.

Wenhan Gao, Ruichen Xu, Yuefan Deng, and Yi Liu. Discretization-invariance? on the discretization mismatch errors in neural operators. In *The Thirteenth International Conference on Learning Representations*, 2025.

Michael Ghil and Eric Simonnet. Geophysical fluid dynamics, nonautonomous dynamical systems, and the climate sciences. In *Mathematical Approach to Climate Change and its Impacts: MAC2I*, pp. 3–81. Springer, 2020.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. Neural message passing for quantum chemistry. In *International conference on machine learning*, pp. 1263–1272. Pmlr, 2017.

Nate Gruver, Marc Anton Finzi, Micah Goldblum, and Andrew Gordon Wilson. The lie derivative for measuring learned equivariance. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=JL7Va5Vy15J.

Abhinav Gupta and Pierre FJ Lermusiaux. Generalized neural closure models with interpretability. *Scientific Reports*, 13(1):10634, 2023.

Brian C Hall. Lie groups, lie algebras, and representations. In *Quantum Theory for Mathematicians*, pp. 333–366. Springer, 2013.

Hans Hersbach, Bill Bell, Paul Berrisford, Shoji Hirahara, András Horányi, Joaquín Muñoz-Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Dinand Schepers, Adrian Simmons, Cornel Soci, Saleh Abdalla, Xavier Abellan, Gianpaolo Balsamo, Peter Bechtold, Gionata Biavati, Jean Bidlot, Massimo Bonavita, Giovanna De Chiara, Per Dahlgren, Dick Dee, Michail Diamantakis, Rossana Dragani, Johannes Flemming, Richard Forbes, Manuel Fuentes, Alan Geer, Leo Haimberger, Sean Healy, Robin J. Hogan, Elías Hólm, Marta Janisková, Sarah Keeley, Patrick Laloyaux, Philippe Lopez, Cristina Lupu, Gabor Radnoti, Patricia de Rosnay, Iryna Rozum, Freja Vamborg, Sebastien Villaume, and Jean-Noël Thépaut. The era5 global reanalysis. *Quarterly Journal of the Royal*

Meteorological Society, 146(730):1999–2049, 2020. doi: https://doi.org/10.1002/qj.3803. URL https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3803.

Jiahe Huang, Guandao Yang, Zichen Wang, and Jeong Joon Park. Diffusionpde: Generative pde-solving under partial observation. *Advances in Neural Information Processing Systems*, 37: 130291–130323, 2024.

Jingwei Huang, Hao Su, and Leonidas Guibas. Robust watertight manifold surface generation method for shapenet models. *arXiv preprint arXiv:1802.01698*, 2018.

Steeven Janny, Aurélien Bénéteau, Madiha Nadri, Julie Digne, Nicolas Thome, and Christian Wolf. EAGLE: Large-scale learning of turbulent fluid dynamics with mesh transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=mfIX4QpsARJ.

Mike Jarvis, Gary Bernstein, and Bhuvnesh Jain. The skewness of the aperture mass statistic. *Monthly Notices of the Royal Astronomical Society*, 352(1):338–352, 2004.

Nicolas Keriven. Not too little, not too much: a theoretical analysis of graph (over) smoothing. *Advances in Neural Information Processing Systems*, 35:2268–2281, 2022.

Bobak Kiani, Lukas Fesser, and Melanie Weber. Unitary convolutions for learning on graphs and groups. *Advances in Neural Information Processing Systems*, 37:136922–136961, 2024.

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.

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.

Remi Lam, Alvaro Sanchez-Gonzalez, Matthew Willson, Peter Wirnsberger, Meire Fortunato, Ferran Alet, Suman Ravuri, Timo Ewalds, Zach Eaton-Rosen, Weihua Hu, Alexander Merose, Stephan Hoyer, George Holland, Oriol Vinyals, Jacklynn Stott, Alexander Pritzel, Shakir Mohamed, and Peter Battaglia. Graphcast: Learning skillful medium-range global weather forecasting, 2023. URL https://arxiv.org/abs/2212.12794.

Andrei Leman and Boris Weisfeiler. A reduction of a graph to a canonical form and an algebra arising during this reduction. *Nauchno-Technicheskaya Informatsiya*, 2(9):12–16, 1968.

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*, volume 32, 2018.

Wei Li, Ruqing Fang, Junning Jiao, Georgios N Vassilakis, and Juner Zhu. Tutorials: Physics-informed machine learning methods of computing 1d phase-field models. *APL Machine Learning*, 2(3), 2024.

Zongyi Li, Nikola Borislavov Kovachki, Kamyar Azizzadenesheli, Burigede liu, Kaushik Bhattacharya, Andrew Stuart, and Anima Anandkumar. Fourier neural operator for parametric partial differential equations. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=c8P9NQVtmnO.

Ziming Liu, Yixuan Wang, Sachin Vaidya, Fabian Ruehle, James Halverson, Marin Soljacic, Thomas Y Hou, and Max Tegmark. Kan: Kolmogorov–arnold networks. In *The Thirteenth International Conference on Learning Representations*, 2024.

Ivan Marisca, Jacob Bamberger, Cesare Alippi, and Michael M Bronstein. Over-squashing in spatiotemporal graph neural networks. *arXiv preprint arXiv:2506.15507*, 2025.

Marco Maurizi, Chao Gao, and Filippo Berto. Predicting stress, strain and deformation fields in materials and structures with graph neural networks. *Scientific reports*, 12(1):21834, 2022.

Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H Barr. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and mathematics III*, pp. 35–57. Springer, 2003.

Yaaqov Mishayev, Yonatan Sverdlov, Tal Amir, and Nadav Dym. Short-range oversquashing. In *The Fourth Learning on Graphs Conference*, 2025.

Thomas W Mitchel, Vladimir G Kim, and Michael Kazhdan. Field convolutions for surface cnns. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10001–10011, 2021.

Thomas W Mitchel, Noam Aigerman, Vladimir G Kim, and Michael Kazhdan. Möbius convolutions for spherical cnns. In *ACM SIGGRAPH 2022 Conference Proceedings*, pp. 1–9, 2022.

Thomas W Mitchel, Carlos Esteves, and Ameesh Makadia. Single mesh diffusion models with field latents for texture generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7953–7963, 2024.

Rudy Morel, Francesco Pio Ramunno, Jeff Shen, Alberto Bietti, Kyunghyun Cho, Miles Cranmer, Siavash Golkar, OLEXANDR GUGNIN, Geraud Krawezik, Tanya Marwah, et al. Predicting partially observable dynamical systems via diffusion models with a multiscale inference scheme. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

M Mustafa, JL Wu, C Jiang, R Wang, et al. Physics-informed machine learning: case studies for weather and climate modelling. *Philosophical Transactions of the Royal Society A*, 379(2194): 20200093, 2021.

NVIDIA. Nvidia h200 tensor core gpu datasheet, 2025. URL https://resources.nvidia.com/en-us-data-center-overview-mc/en-us-data-center-overview/hpc-datasheet-sc23-h200. Retrieved from NVIDIA website.

Peter J Olver. *Applications of Lie groups to differential equations*, volume 107. Springer Science & Business Media, 1993.

Sneh Pandya, Yuanyuan Yang, Nicholas Van Alfen, Jonathan Blazek, and Robin Walters. Iaemu: Learning Galaxy Intrinsic Alignment Correlations. *The Open Journal of Astrophysics*, 8, dec 2 2025. doi: 10.33232/001c.151749.

Jung Yeon Park, Lawson Wong, and Robin Walters. Modeling dynamics over meshes with gauge equivariant nonlinear message passing. *Advances in Neural Information Processing Systems*, 36: 15277–15302, 2023.

Tobias Pfaff, Meire Fortunato, Alvaro Sanchez-Gonzalez, and Peter Battaglia. Learning mesh-based simulation with graph networks. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=roNqYL0_XP.

Stephan Rasp, Stephan Hoyer, Alexander Merose, Ian Langmore, Peter Battaglia, Tyler Russell, Alvaro Sanchez-Gonzalez, Vivian Yang, Rob Carver, Shreya Agrawal, et al. Weatherbench 2: A benchmark for the next generation of data-driven global weather models. *Journal of Advances in Modeling Earth Systems*, 16(6):e2023MS004019, 2024.

Martin Reuter, Silvia Biasotti, Daniela Giorgi, Giuseppe Patanè, and Michela Spagnuolo. Discrete laplace–beltrami operators for shape analysis and segmentation. *Computers & Graphics*, 33(3): 381–390, 2009.

Conor Rowan, Alireza Doostan, Kurt Maute, and John Evans. Solving engineering eigenvalue problems with neural networks using the rayleigh quotient. *International Journal for Numerical Methods in Engineering*, 126(24):e70209, 2025.

T Konstantin Rusch, Michael M Bronstein, and Siddhartha Mishra. A survey on oversmoothing in graph neural networks. *arXiv preprint arXiv:2303.10993*, 2023.

Vıctor Garcia Satorras, Emiel Hoogeboom, and Max Welling. E (n) equivariant graph neural networks. In *International conference on machine learning*, pp. 9323–9332. PMLR, 2021.

Andreas Schlaginhaufen, Philippe Wenk, Andreas Krause, and Florian Dörfler. Learning stable deep dynamics models for partially observed or delayed dynamical systems. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=u8HmtBBSVJS.

Peter Schneider. Weak gravitational lensing. In *Gravitational lensing: strong, weak and micro*, pp. 269–451. Springer, 2006.

Zhiqi Shao, Dai Shi, Andi Han, Yi Guo, Qibin Zhao, and Junbin Gao. Unifying over-smoothing and over-squashing in graph neural networks: A physics informed approach and beyond, 2024. URL https://openreview.net/forum?id=swPf2hwKl8.

Nicholas Sharp and Keenan Crane. A laplacian for nonmanifold triangle meshes. In *Computer Graphics Forum*, volume 39, pp. 69–80. Wiley Online Library, 2020.

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.

Julian Suk, Pim de Haan, Phillip Lippe, Christoph Brune, and Jelmer M Wolterink. Mesh neural networks for se (3)-equivariant hemodynamics estimation on the artery wall. *Computers in biology and medicine*, 173:108328, 2024.

C Sullivan and Alexander Kaszynski. Pyvista: 3d plotting and mesh analysis through a streamlined interface for the visualization toolkit (vtk). *Journal of Open Source Software*, 4(37):1450, 2019.

Jan Tönshoff, Martin Ritzert, Eran Rosenbluth, and Martin Grohe. Where did the gap go? reassessing the long-range graph benchmark. In *The Second Learning on Graphs Conference*, 2023.

Jan Tönshoff, Martin Ritzert, Eran Rosenbluth, and Martin Grohe. Where did the gap go? reassessing the long-range graph benchmark. *Transactions on Machine Learning Research*, 2024.

Dian Wang, Xupeng Zhu, Jung Yeon Park, Mingxi Jia, Guanang Su, Robert Platt, and Robin Walters. A general theory of correct, incorrect, and extrinsic equivariance. *Advances in Neural Information Processing Systems*, 36:40006–40029, 2023.

Rui Wang, Karthik Kashinath, Mustafa Mustafa, Adrian Albert, and Rose Yu. 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, pp. 1457–1466, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403198. URL https://doi.org/10.1145/3394486.3403198.

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.

Rui Wang, Robin Walters, and Rose Yu. Approximately equivariant networks for imperfectly symmetric dynamics. In *International Conference on Machine Learning*, pp. 23078–23091. PMLR, 2022a.

Rui Wang, Robin Walters, and Rose Yu. Data augmentation vs. equivariant networks: A theory of generalization on dynamics forecasting. *International Conference on Machine Learning (ICML) Principles of Distribution Shift Workshop*, 2022b.

Rose Yu and Rui Wang. Learning dynamical systems from data: An introduction to physics-guided deep learning. *Proceedings of the National Academy of Sciences*, 121(27):e2311808121, 2024. doi: 10.1073/pnas.2311808121. URL https://www.pnas.org/doi/abs/10.1073/pnas.2311808121.

A Deferred Theory

This section provides both theoretical background and deferred proofs from the main text.

A.1 Lie Algebras and the Exponential Map

In this section we review the formalism behind 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. Our work encounters the orthogonal and unitary Lie groups

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

as well as the special unitary group

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

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

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

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:

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

Applying the exponential map to a linear operator is given by

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

In the case of Eq. 3, $\mathbf{L}$ is graph convolution, $\mathbf{L}(\mathbf{X}) = \mathbf{AXW}$. Further background on group theory and abstract algebra can be found in Artin (1998), Hall (2013), and Esteves (2020).

A.2 Convolutional smoothing

This section provides results from Kiani et al. (2024) which establishes that Graph Convolution Networks (Kipf & Welling, 2017) have a high probability to exhibit smoothing and unitary convolution preserves smoothness.

Proposition 2 (Proposition 7 in Kiani et al. (2024))**.** *Given a simple undirected graph $\mathcal{G}$ on n nodes with normalized adjacency matrix $\widetilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$ 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 $f_{conv}(\mathbf{X}) = \widetilde{\mathbf{A}}\mathbf{X}\mathbf{W}$ denote convolution with orthogonal feature transformation matrix $\mathbf{W} \in O(d)$. Then, the event below holds with probability $1 - \exp(-\Omega(\sqrt{n}))$:*

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

Proposition 3 (Invariance of Rayleigh quotient, Proposition 6 in Kiani et al. (2024))**.** *Given an undirected graph $\mathcal{G}$ on n nodes with normalized adjacency matrix $\widetilde{\mathbf{A}} = \mathbf{D}^{-1/2}\mathbf{A}\mathbf{D}^{-1/2}$, the Rayleigh quotient is invariant under normalized unitary or orthogonal graph convolution, i.e. $R_{\mathcal{G}}(\mathbf{X}) = R_{\mathcal{G}}(f_{\mathrm{UniConv}}(\mathbf{X}))$ where f_{UniConv} is either seperable or Lie.*

A.3 Gauge and Euclidean Equivariance

In this section, we introduce the necessary background and formal definitions for the equivariance constraints commonly applied to tasks defined on meshes. While working with arbitrary meshes, many commonly used network architectures compute distances between node positions. One has

the option of computing these distances in either global Cartesian coordinates or in local tangent spaces of the mesh. In both cases, we may exploit the symmetry of these coordinate systems by enforcing equivariance with respect to transformations from a certain symmetry group into the network architecture, which allows the network to automatically generalize across orbits.

We now give precise definitions of equivariance and invariance.

Definition 4. Let $f : \mathcal{X} \to \mathcal{Y}$ be a map between input and output vector spaces $\mathcal{X}$ and $\mathcal{Y}$. Let G be a group with representations $\rho^{\mathcal{X}}$ and $\rho^{\mathcal{Y}}$ which transform vectors in $\mathcal{X}$ and $\mathcal{Y}$ respectively. Representations are group homomorphisms which map group elements to invertible linear transformations. The map $f : \mathcal{X} \to \mathcal{Y}$ is *equivariant* if

$$\rho^{\mathcal{Y}}(g)f(x) = f(\rho^{\mathcal{X}}(g)x)\ , \text{ for all } g \in G, x \in \mathcal{X}\ .$$

Invariance is a special case of equivariance in which $\rho^{\mathcal{Y}} = \mathrm{Id}^{\mathcal{Y}}$ for all $g \in G$. With an invariant operator, the output of f is unaffected by the transformations applied to the input.

Definition 5. A map $f : \mathcal{X} \to \mathcal{Y}$ is *invariant* if

$$f(x) = f(\rho^{\mathcal{X}}(g)x)\ , \text{ for all } g \in G, x \in \mathcal{X}.$$

A.3.1 Euclidean Equivariance

For a mesh defined over a global coordinate system, a common choice of symmetry constraint is equivariance to the Euclidean group in n dimensions, $E(n)$. In this setting, the mesh is treated as a graph with positional encodings, and the equivariance constraint ensures generalization to different roto-translations of the mesh.

Definition 6. Let $t \in \mathbb{R}^n$ be a translation vector and $Q \in \mathbb{R}^{n\times n}$ an orthogonal matrix representing a rotation or reflection. A function f is equivariant to the Euclidean group $E(n)$ if for any $t \in \mathbb{R}^n$ and $Q \in \mathbb{R}^{n\times n}$ we have

$$f(Qx + t) = Qf(x) + t.$$

A.3.2 Gauge Equivariance

We may also choose to embed coordinates locally, using coordinates that are intrinsic to the 2D mesh rather than the extrinsic 3D coordinates of the embedding space. This approach arises from the desire for a general convolution-like operator over arbitrary manifolds discretized as a mesh. To encode data over a mesh it is still necessary to make a choice of local coordinate frame at each vertex. In order to guarantee the equivalence of the features resulting from different choices of reference frames, the model should be invariant to change of coordinates frame at each vertex, i.e. gauge equivariant.

We specifically adapt the strategy described in de Haan et al. (2021) and define the local coordinate frame at each vertex in terms of a reference neighboring vertex. Denote v_a as the reference neighbor for gauge A, in which the neighbors have angles θ_A, and denote v_b as the reference neighbor for gauge B with angles θ_B. Comparing the two gauges, we see that they are related by a rotation of angle ϕ, so that $\theta_B = \theta_A - \phi$. This change of gauge is called a gauge transformation of angle $g := \phi$.

Definition 7 (Equations 3 and 4 in de Haan et al. (2021))**.** Let ρ_{in} and ρ_{out} be input and output types with dimensions C_{in} and C_{out}. Let $K_{\text{self}} \in \mathbb{R}^{\mathbb{C}_{\text{out}}\times\mathbb{C}_{\text{in}}}$ and $K_{\text{neigh}} \colon [0, 2\pi) \to \mathbb{R}^{C_{\text{out}}\times C_{in}}$ be two kernels. We say the kernels are *gauge equivariant* if for any gauge transformation $g \in [0, 2\pi)$ and for any angle $\theta \in [0, 2\pi)$ we have

$$K_{\text{neigh}}(\theta - g) = \rho_{\text{out}}(-g)K_{\text{neigh}}(\theta)\rho_{\text{in}}(g), \qquad K_{\text{self}} = \rho_{\text{out}}(-g)K_{\text{self}}\,\rho_{\text{in}}(g).$$

Finally, as features at different nodes live in different tangent spaces and thus have different gauges, it is invalid to sum them directly. Let f_p and f_q be node features of a pair of neighboring nodes p and q. Before performing gauge equivariant convolution, we must parallel transport each f_q to T_pM along the mesh edge that connects the two vertices for them to be in the same gauge. For more details, we refer the reader to de Haan et al. (2021).

A.4 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 8 (Fundamental Domain, Definition 4.1 in Wang et al. (2023))**.** 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 Wang et al. (2023):

Assumption 2 (Integrability Hypothesis, Sec. A in Wang et al. (2023))**.** 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 1. 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

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

A.5 Proof of Main Theorem

We now provide proof of our main theoretical result in the main text. We repeat the theorem here for convenience.

Theorem 1. Let F be a fundamental domain of $\mathrm{SU}(n)$ in Z. In particular, $F = \{te \colon t \in \mathbb{R}_+\}$ 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 of Theorem 1. By the reverse triangle inequality,

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

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

In the following example, we show how this bound may be computed.

Example 1 (Variance on the Unit Disk)**.** Denote by D_2 the unit disk $D_2 = \{(x, y) \colon x^2 + y^2 \leq 1\}$. Let $Z = \mathbb{R}^2$ be a domain with density

$$p = \begin{cases} \frac{1}{\pi}, & (x, y) \in D_2 \\ 0, & \text{Otherwise.} \end{cases}$$

Denote by f a function in polar coordinates given by

$$\begin{aligned} f \colon (\theta, r) &\to \mathbb{R}^2 \\ \theta &\mapsto (\sin\theta + r, \cos\theta + r). \end{aligned}$$

On each orbit, we compute

$$\mathbb{E}_{G_z}[f] = (r, r), \qquad \mathbb{V}_{G_z}[f] = 1, \qquad p(r) = \begin{cases} 2r, & r \leq 1 \\ 0, & \text{otherwise.} \end{cases}$$

The approximation error bound of a unitary function u of f is then

$$\int_Z p(z)\|u(z) - f(z)\|_2^2 dz \geq \int_F p(r)\mathbb{V}_{G_z}[f]dr = \int_0^1 2r(1)dr = 1.$$

A.6 Unitary Convolution on Meshes

In this section, we prove Corollary 1 stated in Sec. 5.3 and repeated here for convenience.

Corollary 1 (Corollary to Proposition 3)**.** Given a mesh $\mathcal{M}$ with normalized adjacency matrix $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2}(\mathcal{W} \odot \mathbf{A})\mathbf{D}^{-1/2}$ that satisfies Assumption 1, the mesh Rayleigh quotient is invariant under normalized unitary or orthogonal graph convolution, i.e. $R_{\mathcal{M}}(\mathbf{X}) = R_{\mathcal{M}}(f_{\text{UniMeshConv}}(\mathbf{X}))$ where $f_{\text{UniMeshConv}}$ is either separable or Lie.

Our proof follows the same structure as the proof of Proposition 3 in Kiani et al. (2024) with modifications to account for the weighted adjacency matrix. Namely, we invoke Assumption 1 which ensures that $f_{\text{UniMeshConv}}$ is norm preserving and therefore the the strategy in Kiani et al. (2024) still holds.

Proof. We first prove invariance for Eq. 8. By the circulant property of the trace,

$$\text{Tr}\left(\left(\exp(i\tilde{\mathbf{A}})\mathbf{X}\mathbf{U}\right)^{\dagger}\left(\mathbf{I}-\tilde{\mathbf{A}}\right)\left(\exp(i\tilde{\mathbf{A}})\mathbf{X}\mathbf{U}\right)\right) = \text{Tr}\left(\mathbf{X}^{\dagger}\exp(-i\tilde{\mathbf{A}})\left(\mathbf{I}-\tilde{\mathbf{A}}\right)\exp(i\tilde{\mathbf{A}})\mathbf{X}\right).$$

Because $\exp(-i\tilde{\mathbf{A}})$, $\exp(i\tilde{\mathbf{A}})$, and $(\mathbf{I}-\tilde{\mathbf{A}})$ share an eigenbasis, they commute, so

$$\text{Tr}\left(\left(\exp(i\tilde{\mathbf{A}})\mathbf{X}\mathbf{U}\right)^{\dagger}\left(\mathbf{I}-\tilde{\mathbf{A}}\right)\left(\exp(i\tilde{\mathbf{A}})\mathbf{X}\mathbf{U}\right)\right) = \text{Tr}\left(\mathbf{X}^{\dagger}\left(\mathbf{I}-\tilde{\mathbf{A}}\right)\mathbf{X}\right).$$

For the denominator, we need to show that $\|\exp(i\tilde{\mathbf{A}})\mathbf{X}\mathbf{U}\|_F^2 = \|\mathbf{X}\|_F^2$. By Assumption 1 we have that $\mathcal{W}$ is symmetric. Because $\mathbf{A}$ is also symmetric, we have that $i\tilde{\mathbf{A}}$ is skew-symmetric and therefore $\exp(i\tilde{\mathbf{A}}) \in SU(n)$. Thus, $\|\exp(i\tilde{\mathbf{A}})\mathbf{X}\mathbf{U}\|_F^2 = \|\mathbf{X}\|_F^2$ and finally $R_{\mathcal{M}}(\mathbf{X}) = R_{\mathcal{M}}(f_{\text{UniMeshConv}}(\mathbf{X}))$.

We now show that Eq. 9 also preserves the Rayleigh quotient. First, we need to show that $\|\exp(\mathbf{A}\mathbf{X}\mathbf{W})\|_F^2 = \|\mathbf{X}\|_F^2$. To do this, we note that Eq. 9 can equivalently be viewed us a function that acts on a vector in $\mathbb{C}^{nd}$. By properties of the Kroneckor tensor product,

$$f_{\text{UniMeshConv}}(\mathbf{X};\mathbf{A}) = \exp(\mathbf{A}\mathbf{X}\mathbf{W}) \iff \text{vec}\left(f_{\text{UniMeshConv}}(\mathbf{X};\mathbf{A})\right) = \exp\left(\mathbf{A}\otimes\mathbf{W}^{\mathbf{T}}\right)\text{vec}(\mathbf{X}).$$

Since

$$\left(\mathbf{A}\otimes\mathbf{W}^{\mathbf{T}}\right) + \left(\mathbf{A}\otimes\mathbf{W}^{\mathbf{T}}\right)^{\dagger} = \mathbf{A}\otimes\left(\mathbf{W}+\mathbf{W}^{\dagger}\right)^{T} = 0,$$

we have that $\left(\mathbf{A}\otimes\mathbf{W}^{\mathbf{T}}\right)$ is in the lie algebra of the unitary group and therefore preserves the norm of $\text{vec}(\mathbf{X})$. This holds for any symmetric edge weighting $\tilde{\mathbf{A}} = \mathcal{W}\odot\mathbf{A}$, which is guaranteed by Assumption 1. Thus, $\|\exp(\mathbf{A}\mathbf{X}\mathbf{W})\|_F^2 = \|\mathbf{X}\|_F^2$. Next, note that $\exp\left(\tilde{\mathbf{A}}\otimes\mathbf{W}^{\mathbf{T}}\right)$ commutes with $(\tilde{\mathbf{A}}\otimes\mathbf{I})$. Thus,

$$\begin{aligned}
&\text{Tr}\left(f_{\text{UniMeshConv}}(\mathbf{X};\tilde{\mathbf{A}})^{\dagger}(\mathbf{I}-\tilde{\mathbf{A}})f_{\text{UniMeshConv}}(\mathbf{X};\tilde{\mathbf{A}})\right)\\
&= \text{vec}(\mathbf{X})^{\dagger}\exp(\tilde{\mathbf{A}}\otimes\mathbf{W}^{\mathbf{T}})^{\dagger}\left[(\mathbf{I}-\tilde{\mathbf{A}})\otimes\mathbf{I}\right]\exp(\tilde{\mathbf{A}}\otimes\mathbf{W}^{\mathbf{T}})\text{vec}(\mathbf{X})\\
&= \text{vec}(\mathbf{X})^{\dagger}\left[(\mathbf{I}-\tilde{\mathbf{A}})\otimes\mathbf{I}\right]\text{vec}(\mathbf{X}).
\end{aligned}$$

Multipliying the above by $\|\mathbf{X}\|_F^{-2}$ recovers $R_{\mathcal{M}}(\mathbf{X})$. We conclude that $R_{\mathcal{M}}(\mathbf{X}) = R_{\mathcal{M}}(f(\mathbf{X}))$. □

Remark 2. Corollary 1 was applied to convolution with the symmetric cotangent weights in Eq. 7, but the proof extends without loss of generality to any set of symmetric weights.

A.7 Discrete Differential Geometry

We provide further details and visualizations for concepts in discrete differential geometry, including the mesh manifold condition, the cotangent Laplacian, and Delaunay criterion. We also review results from the literature that suggests that the mesh edge rewiring algorithm used by the Robust Laplacian is a safe choice for the task of PDE solving. For additional reference on these topics, see Meyer et al. (2003) and Crane et al. (2013).

The Manifold Condition Our work makes use of the assumption that a given mesh is manifold. We recall the following definition of the mesh manifold condition:

Definition 9 (Manifold Condition, (Sharp & Crane, 2020))**.** An interior (or boundary) edge ij is manifold if it is contained in exactly two (or one) triangles; an interior (or boundary) vertex i is manifold if the boundary of all triangles incident on i forms a single loop (or path) of edges.

Alternative definitions for the manifold condition often state that a mesh is manifold if interior vertices have local neighborhoods that are homeomorphic to the unit disk and boundary vertices are homeomorphic to the half disk. We refer specifically to Definition 9 when we talk about a mesh being manifold in the paper.

Cotangent Laplacian Area Normalization. Recall that for a scalar function s on mesh vertices we the cotangent Laplacian is defined as

$$(\tilde{\mathbf{L}}(s))_i = \frac{1}{2A_i} \sum_{j \in \mathcal{N}(i)} (\cot \alpha_{ij} + \cot \beta_{ij})(s_j - s_i)$$

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 and that we use the barycentric cell area for A_i. In particular, let $\mathcal{A}_{abc}$ be the area of a triangular face with vertices abc and let $\mathcal{F}(i)$ be the set of faces containing vertex i. The barycentric cell area is defined

$$A_i = \sum_{abc \in \mathcal{F}(i)} \mathcal{A}_{abc}/3.$$

Normalization by the cell area was used for the dataset construction in Park et al. (2023), but it is not used in the definition of the Robust Laplacian (Sharp & Crane, 2020). In fact, the cell area introduces asymmetry in the edge weights. This is undesirable, as unitary mesh convolution depends on symmetric edge weights in order to preserve the Rayleigh quotient.

Cotangent Laplacian Edge Weights and Robust Rewiring. As noted in Sec. 5.3, an arbitrary mesh may have negative cotangent weights. These cotangent weights have the following geometric meaning. For vertices ij connected by an edge, we say that the edge is *primal*. For manifold meshes, for each primal edge there exists a *dual edge* that connects the triangle circumcenters for the two triangles that share a primal edge. The cotangent weights correspond to the ratio of the primal and dual edge lengths for vertices ij (Crane et al., 2013). The weights are positive when the angles $\alpha_{ij} + \beta_{ij} \leq \pi$. The Robust Laplacian applies two edge rewiring algorithms sequentially, the tufted cover algorithm and the Delaunay edge flip algorithm. The tufted cover algorithm ensures the mesh is manifold so that the Delaunay edge flip algorithm can be applied. The Delaunay edge flip algorithm then ensures that the mesh satisfies the intrinsic Delaunay criterion (Definition 2). A sample edge flip is illustrated in Fig. 3.

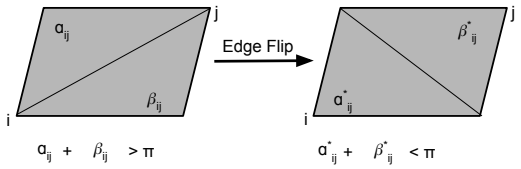


Figure 3: Illustration of an intrinsic Delaunay edge flip performed by the Robust Laplacian edge rewiring algorithm. This figure is a reproduced version of Figure 7, Sharp & Crane (2020).

Sharp & Crane (2020) note that from a finite element perspective, changing the triangulation via Delaunay edge-flipping effectively just provides a different set of linear basis functions for the same polyhedral domain. Thus, the only practical concern is whether the tufted cover algorithm, which ensures that the mesh is manifold, does not dramatically change the connectivity. Empirical results from Sharp & Crane (2020) find that the Robust Laplacian greatly *improves* performance computing geodesic distances with the heat method (Crane et al., 2017), which depends on numerical PDE solving on highly nonmanifold meshes with the cotangent Laplacian. This provides confidence that the tufted cover algorithm improves the edge-weighting scheme for unitary mesh convolution. See Crane et al. (2017) for additional reference on numerical PDE solving on nonmanifold meshes.

A.8 Rayleigh Quotient Sensitivity

We include results from Ferrandi & Hochstenbach (2024) and Dong et al. (2024) 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 Ferrandi & Hochstenbach (2024))**.** *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 Dong et al. (2024):

Proposition 5 (Theorem 1 in Dong et al. (2024))**.** *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 Dong et al. (2024))**.** *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 Dong et al. (2024) state fewer hypotheses than Ferrandi & Hochstenbach (2024). 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 Further Details

B.1 Simulated Heat Diffusion Dataset

This section details dataset generation specifications for our experiment in Sec. 6.1. We generate grid-graphs with an average of 10 nodes and a standard deviation of 2 nodes. On the grid we randomly set 20 nodes to be heat sources. They are given a heat value of 1 and all other nodes start at 0. Using `PyGSP`, We simulate heat flow on $10,000$ graphs for training, and the task is to predict the next time step given the previous one. The simulation proceeds until time $T = 10$ in increments of $\Delta T = 0.5$ time steps. A sample graph data point is given in Fig. 4.

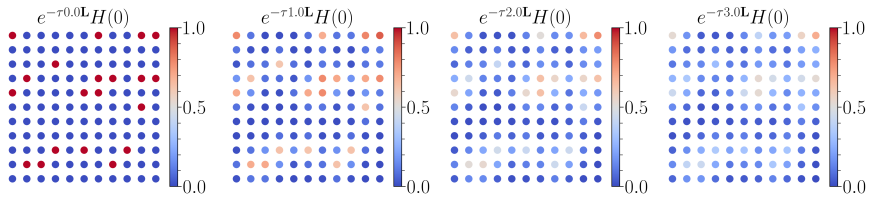


Figure 4: Sample heat diffusion process on a grid discretized as a graph. Node neighbors are the nodes that sit adjacent in the grid.

B.2 Taylor Series Sensitivity Analysis

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 study these tendencies at initialization for the heat diffusion dataset that is used for the experiment in Sec. 6.1, described further in Sec. B.1. Our analysis echos a theme similar to Gruver et al. (2023) and Gao et al. (2025) that practitioners should be more thorough in evaluating when numerical approximations break strict theoretical guarantees.

Experimental Setup. We simulate heat diffusion on a grid graph and use time step 3 to conduct the sensitivity analysis. We evaluate on the models f_{GCN}, $f_{\text{SepUniConv}}$, and $f_{\text{LieUniConv}}$. For each model f and truncation length $\mathbf{T}_{\max} \in \{1, \ldots, 10\}$, 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 $D_{\text{KL}}\left(P_{\mathbf{X}} \parallel P_{f(\mathbf{X})}\right)$, which measures the change in the distribution of Rayleigh quotients caused by the model at initialization.

Results. We see in Fig. 5 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 $\mathcal{O}\left(\frac{[\|\mathbf{AXW}\|_{\mathbf{O}}]^{t+1}\|\mathbf{X}\|_2}{(t+1)!}\right)$ where $\|\cdot\|_{\mathbf{O}}$ is the operator norm. Furthermore, works such as Ferrandi & Hochstenbach (2024) and Dong et al. (2024) show theoretically that small truncation errors will not compound into large deviations in the Rayleigh quotient. For details on the relevant propositions from Ferrandi & Hochstenbach (2024) and Dong et al. (2024), see Sec. A.8.

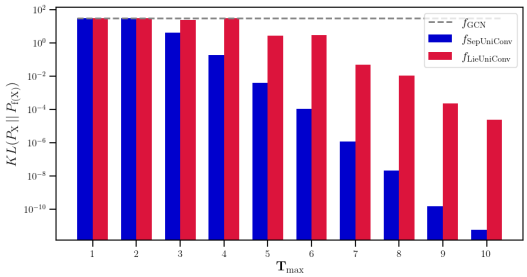


Figure 5: KL divergence between distribution of Rayleigh quotients before and after applying the model. Results are averaged over 10 runs.

In the supplementary material we include a video that shows the evolving Rayleigh quotient distribution as we increase $\mathbf{T}_{\max}$. Fig. 5 is also clickable and links to the same video in our anonymous artifact.

B.3 Heat Diffusion Model Ensembles

This section provides the results for Sec. 6.1 for a larger ensemble of models. In Fig. 6, we see that the training runs in Fig. 1 occur frequently, with only a couple runs diverging.

C Baselines

We include as baselines standard GNN models without any specific inductive biases for working on meshes, including a GCN (Kipf & Welling, 2017) and an MPNN (Gilmer et al., 2017). Additionally, we study symmetry preserving equivariant models, including gauge and Euclidean equivariance (formally defined in Sec. A.3). Informally, Euclidean equivariant models are invariant to roto-translations of the mesh in Cartesian coordinates and Gauge Equivariant GNNs are invariant to a choice of reference angle for models that work in local coordinates of the mesh-manifold. These models have been shown to be particularly strong for PDE solving on meshes (Park et al., 2023). We benchmark a state-of-the-art (SOTA) Euclidean equivariant model (Satorras et al., 2021) as well as different types of Gauge Equivariant GNNs, including convolutional with GemCNN (de Haan et al., 2021), attentional with EMAN (Basu et al., 2022), and message passing with Hermes (Park et al., 2023). We also consider a SOTA mesh transformer (Janny et al., 2023). We compare these baselines with R-UniMesh .

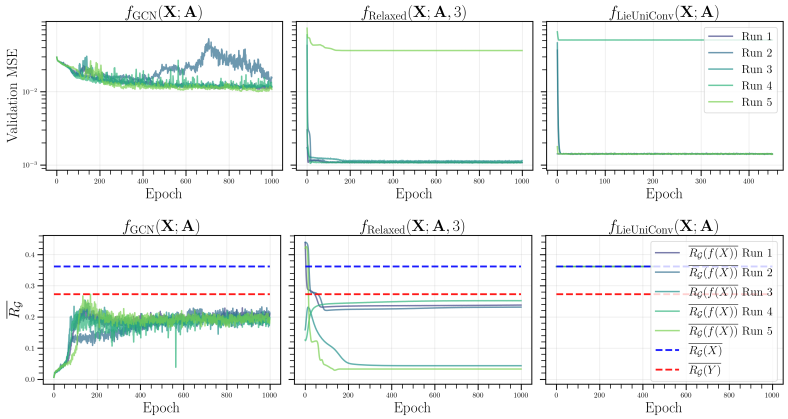


Figure 6: **Top:** Validation MSE for an ensemble of 5 runs for a GCN (left), R-UNIGRAPH (middle), and a Lie unitary convolution network (right) at timestep $t = 3$. R-UNIGRAPH significantly outperforms the GCN and also outperforms the Lie unitary network. **Bottom:** The average Rayleigh quotient over all graphs for an ensemble of 5 runs for the same models at timestep $t = 3$. The GCN is under constrained and biased towards oversmoothing at initialization. R-UNIGRAPH is able to roughly match the true smoothness of the labeled graphs. The Lie unitary network is overconstrained and can not model the Rayleigh quotient of the labels because it is forced to preserve the Rayleigh quotient of the input graphs.

D PYVISTA MESH TRAINING DETAILS, EVALUATION DETAILS, AND FURTHER EXPERIMENTS

This section provides extra experimental details and results for our dynamical systems modeling on `PyVista` meshes.

D.1 PYVISTA MESH DYNAMICAL SYSTEMS

This section provides details on the PDEs to be solved on the `PyVista` meshes. Before defining the PDEs to be solved, let us establish notation. Let α be the thermal diffusivity, and c a constant. The heat and wave equations on the mesh are then given by

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

We now define the Cahn-Hilliard equation (Cahn & Hilliard, 1958). Let c be the fluid concentration, M the diffusion coefficient, μ the chemical potential, f the double-free energy function, and λ a positive constant. The Cahn-Hilliard equation is often represented by the following two coupled second order equations:

$$\frac{\partial c}{\partial t} - M\tilde{\mathbf{L}}(\mu) = 0, \quad \mu - \frac{\partial f}{\partial c} + \lambda \tilde{\mathbf{L}}(c) = 0.$$

Here, the double-free energy function is given by $f(c) = 100c^2(1-c^2)$. Sample initial conditions for each equation on the `PyVista` meshes are shown in Tab. 3 and Tab. 4.

D.2 PYVISTA MESH TRAINING DETAILS

We extend the publicly available code base from Park et al. (2023) to train our baselines for the `PyVista` and WeatherBench2 datasets: `https://github.com/jypark0/hermes/`. For GemCNN, EMAN, and Hermes, we use the already available pretrained model checkpoints. For GCN, R-UNIMESH, Mesh Transformer, MPNN, and EGNN we train our own models. We use the same train / test split for the meshes as in Park et al. (2023). We performed ablations over learning rate and latent space sizes. Following Park et al. (2023) we keep models within a $\sim 40,000 - 50,000$ parameter budget. We note that this budget is relatively small, and that models that diverge in our

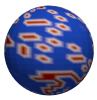
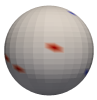
Mesh	Heat $T = 0$	Wave $T = 0$
Armadillo		
Bunny		
Lucy		
Sphere		
Spider		
Urn		
Woman		

Table 3: Sample initializations for the heat and wave equations on the `PyVista` meshes.

experiments could potentially perform better under a more forgiving budget. All runs were performed on a single H200 GPU (NVIDIA, 2025). We use the previous 5 time steps as input node feature vectors and backpropagate through 3 steps of auto-regressive inference.

Hyperparameters are given in our artifact. Defaults are taken from Park et al. (2023) if provided and otherwise optimized via grid search. Considered hyperparameters include learning rate, optimizer, training epochs, latent size, and skip connections. We also consider z-scoring of normed edge lengths for EGNN, different decoder heads for R-UNIMESH, and the number of clusters for the mesh transformer.

As observed in Park et al. (2023), we notice that residual connections can be key for performance with the gauge equivariant models. For R-UNIMESH, we found that using a MLP readout with

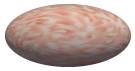
Mesh	Cahn-Hilliard $T = 0$
Bunny	
Ellipsoid	
Sphere	
Super Toroid	

Table 4: Sample initializations for the Cahn-Hilliard equation on the `PyVista` meshes.

sinusoidal activation functions was a key ingredient for strong performance on MeshPDE. This supports previous work on how to train GNNs for long range tasks (Tönshoff et al., 2023; 2024). However, the GCN decoder exhibited the best performance on WB2.

D.3 PyVista Evaluation Details

In order to aggregate smoothness errors over all time steps, we introduce a new metric. Define the Rayleigh Error (RE) by $\int_0^\infty |R_{\mathcal{M}}(\mathbf{Y_t}) - R_{\mathcal{M}}(f(\mathbf{X_t}))|dt$. In practice we approximate this by summing over the time steps where we are able to perform inference and normalize to the max timestep:

$$\text{RE}(f) = \frac{1}{\mathbf{T}_{\max}} \sum_{t}^{\mathbf{T}_{\max}} |R_{\mathcal{M}}(\mathbf{Y_t}) - R_{\mathcal{M}}(f(\mathbf{X_t}))|.$$

Following Janny et al. (2023) and Pandya et al. (2025), we also consider the scale invariant metrics NRMSE and SMAPE averaged over the entire rollout:

$$\text{NRMSE}(f) = \frac{1}{\mathbf{T}_{\max}} \sum_{t}^{\mathbf{T}_{\max}} \sqrt{\frac{\frac{1}{n}\sum_{i=1}^{n} \left(f(\mathbf{X_t})_i - (\mathbf{Y_t})_i\right)^2}{\frac{1}{n}\sum_{i=1}^{n} (\mathbf{Y_t})_i^2}}$$

$$\text{SMAPE}(f) = \frac{1}{\mathbf{T}_{\max}} \sum_{t}^{\mathbf{T}_{\max}} \frac{1}{n} \sum_{i=1}^{n} \frac{2|(\mathbf{Y_t})_i - f(\mathbf{X_t})_i|}{|(\mathbf{Y_t})_i| + |f(\mathbf{X_t})_i| + \varepsilon}$$

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.

D.4 PyVista Mesh Qualitative Diagnostics

In this section, we validate the superior performance of R-UniMesh on solving the heat equation with qualitative diagnostics. In Tab. 5 we show that R-UniMesh is the best at capturing the true smoothness of an unseen mesh during each step of the rollout. In the supplementary material we include a video corresponding to the rollout in Tab. 5 over all timesteps.

Time	Truth	R-UniMesh **(Ours)**	EMAN	Hermes
10				
50				
100				
150				
190				

Table 5: Qualitative comparison of model performance for the heat equation on the armadillo mesh. Our R-UniMesh model remains faithful to the ground truth during each step of the rollout, whereas the EMAN model over smooths and the Hermes model under smooths.

D.5 Beyond 1−hop Smoothness

Since the Rayleigh quotient is a 1−hop metric, this section performs additional comparisons with a more global smoothness metric and finds that our 1−hop smoothness tendencies also hold more generally for the gauge equivariant models we study. In particular, we define smoothness according to the 2−point correlation function. Let $\delta : \mathbb{R}^3 \to \mathbb{R}$ be a function that maps a point $\mathbf{x}$ on a mesh to the scalar solution $u(\mathbf{x})$ (or approximation thereof) to the PDE at that point. The smoothness is then defined by the 2−point correlation function ξ given in Eq. 10:

$$\xi(r; \delta) = \mathbb{E}\left[\delta(\mathbf{x})\delta(\mathbf{x} + r)\right]. \tag{10}$$

Intuitively, if node features are similar at a distance of r apart, the correlation will be high. This allows us to study smoothness beyond 1−hop neighbors by considering larger r. Fig. 7 shows an example correlation function for Hermes at a given time step. We note that this characterization of

smoothness is common in the weak gravitational lensing literature for point-cloud datasets (Schneider, 2006) and are easily computed with the `TreeCorr` library (Jarvis et al., 2004).

Let δ_{ij} be the scalar field for the ground truth on a mesh $\mathcal{M}_i$ at time step j and $\widehat{\delta_{ij}}$ be the approximation thereof. We define our smoothness error by

$$\text{err}_{\text{smooth}}(\widehat{\delta_{ij}}) = \frac{1}{r_{\text{bins}}} \frac{1}{\mathbf{T}_{\max}} \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{\mathbf{T}_{\max}} \sum_{k=1}^{r_{\text{bins}}} |\xi(r_k; \delta_{ij}) - \xi(r_k; \widehat{\delta_{ij}})|.$$

We note that the correlation function is related to the Fourier space power spectrum $P(k)$ by

$$\xi(r) = \frac{1}{2\pi^2} \int k^2 P(k) \frac{\sin(kr)}{kr} dk. \tag{11}$$

Thus, Eq. 11 informs us that our metric for smoothness as a function of r is related to traditional energy spectrum errors (e.g., Wang et al., 2021). We leave a more systematic comparison between the measures as an opportunity for future work.

As seen in Tab. 6, the more expressive attention and message passing based models are much better at capturing the underlying smoothness. The CNN model diverges for the heat and wave datasets, but performs reasonably well on Cahn-Hilliard. This is mirrored by our results in Tab. 2 in the main text for the Gauge Equivariant models.

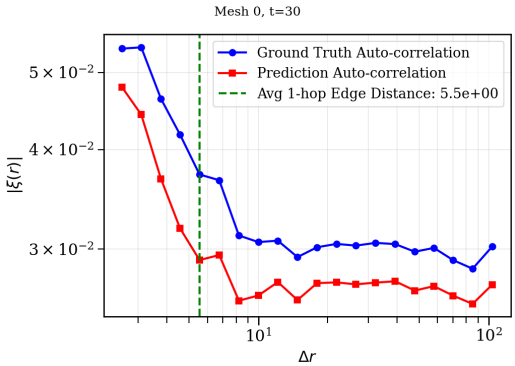


Figure 7: Smoothness for a Hermes model as measured by the $2-$point correlation function. The plot indicates undersmoothing in each radial bin.

E WeatherBench2 Further Details

We lay out the relevant training, evaluation, and dataset details for WB2.

E.1 Training Details and Problem Statement

Our WB2 problem statement is as follows. Given a dataset $\mathcal{D} = \{X_i\}_{i=1}^{N}$ of historical weather data, the task of weather forecasting is to predict future weather conditions $X_T \in \mathbb{R}^{V \times H \times W}$ given initial conditions $\{X_i\}_{i=1}^{K}, X_i \in \mathbb{R}^{V \times H \times W}$, where T is the target lead time, K is the number of input time steps to the model, V is the number of atmospheric variables, and $H \times W$ is the spatial resolution of the data, which depends on how densely we grid the globe. We follow the same training and hyper parameter optimization strategy as in Sec. D.2. The only difference is that we use the 3 previous time steps as input instead of 5. All models are given a consistent compute budget of 8 hours on an NVIDIA H200 GPU for up to 100 epochs.

E.2 Evaluation Details

Here we give precise definitions of the evaluation and metrics omitted in the main text. We begin by establishing some notation common to the subsections, and consistent with the notation used in (Rasp et al., 2024).

Heat ($\alpha = 1$)	
Model	$\text{err}_{\text{smooth}}$ ($\downarrow$)
GemCNN	–
EMAN	$\mathbf{4.04 \times 10^{-3}}$
Hermes	9.71×10^{-3}
Wave ($c = 1$)	
Model	$\text{err}_{\text{smooth}}$ ($\downarrow$)
GemCNN	–
EMAN	$\mathbf{1.78 \times 10^{-3}}$
Hermes	1.38×10^{-2}
Cahn–Hilliard	
Model	$\text{err}_{\text{smooth}}$ ($\downarrow$)
GemCNN	1.89×10^{-1}
EMAN	4.59×10^{-1}
Hermes	$\mathbf{9.61 \times 10^{-3}}$

Table 6: $\text{err}_{\text{smooth}}$ for Gauge Equivariant models on the `PyVista` Mesh datasets. Dashes (–) indicate non-convergence. Best performing model is indicated with **bold text**.

Let f denote the forecast, o the ground-truth observation, and c the climatology. Let $t \in \{1, \ldots, T\}$ denote the verification time, $l \in \{1, \ldots, L\}$ the lead time, $i \in \{1, \ldots, I\}$ the latitude index, and $j \in \{1, \ldots, J\}$ the longitude index. Forecasts are indexed as $f_{t,l,i,j}$, while observations and climatology are indexed by absolute time as $o_{t,i,j}$ and $c_{t,i,j}$.

E.2.1 Latitude weighting

In an equiangular latitude-longitude grid, grid cells at the poles have a much smaller area compared to grid cells at the equator. Weighting all cells equally in the computation of RMSE and ACC would result in an inordinate bias towards the polar regions. As a result both metrics are latitude-weighted with weights computed as follows:

$$w(i) = \frac{\sin \theta_i^u - \sin \theta_i^l}{\frac{1}{I} \sum_i^I (\sin \theta_t^u - \sin \theta_i^l)},$$

where θ_i^u and θ_i^l indicate upper and lower latitude bounds, respectively.

E.2.2 Climatology

The climatology c is a function of the day of year and time of day, it is computed by taking the mean of ERA5 data from 1990 to 2019 (inclusive) for each grid point. A sliding window of 61 days is used around each day of year and time of day combination with weights linearly decaying to zero from the center. For notational consistency, we also define the lead-time–indexed climatology $c_{t,l,i,j} := c_{t+l,i,j}$, corresponding to the climatology at the forecast valid time.

E.2.3 Root mean squared error (RMSE)

Following the WB2 convention, our work measures error in terms of RMSE. For each variable and level pair, the RMSE at lead time l is defined as:

$$\text{RMSE}_l = \sqrt{\frac{1}{TIJ} \sum_t^T \sum_i^I \sum_j^J w(i)(f_{t,l,i,j} - o_{t,i,j})^2}.$$

This choice is important for temperature forecasting, as we are invariant to choice of unit (e.g., temperature in terms of Kelvin and Celsius will have the same RMSE). Moreover, the change in scale over time is less dramatic as it was for the `PyVista` meshes, where we considered NRMSE.

E.2.4 ANOMALY CORRELATION COEFFICIENT (ACC)

The ACC is computed as the Pearson correlation coefficient of the anomalies with respect to the climatology c. Denote the differences between forecast and climatology and between observation and climatology by

$$f'_{t,l,i,j} = f_{t,l,i,j} - c_{t,l,i,j}; \quad o'_{t,i,j} = o_{t,i,j} - c_{t,i,j}.$$

The ACC at lead time l is then defined as

$$\mathrm{ACC}_l = \frac{1}{T}\sum_t^T \frac{\sum_i^I \sum_j^J w(i) f'_{t,l,i,j}\, o'_{t,i,j}}{\sqrt{\sum_i^I \sum_j^J w(i) {f'_{t,l,i,j}}^2 \sum_i^I \sum_j^J w(i) {o'_{t,i,j}}^2}}.$$

ACC ranges from 1, indicating perfect correlation, to -1, indicating perfect anti-correlation. The ECMWF states that when the ACC value falls below 0.6, it is considered that the positioning of synoptic scale features ceases to have value for forecasting purposes.

E.3 EARTH MESH DISCRETIZATION

We construct a spherical mesh of the Earth by directly projecting the latitude–longitude grid points onto the unit sphere, and define mesh connectivity according to the original grid neighborhood structure. In order to obtain triangular faces, we further subdivide each cell into two triangles. The resulting mesh has 29040 nodes, 57600 faces, and 86640 edges. We note that this mesh construction is simpler than those used in other graph-based models, such as GraphCast (Lam et al., 2023), which employs a subdivided icosahedron as the underlying mesh. However, our approach has the advantage that it operates directly on the native latitude–longitude grid and therefore does not require interpolation or regridding of the ERA5 data. Although more elaborate mesh constructions is likely to improve performance in real weather forecasting applications, our focus is on methodological experimentation rather than optimized weather prediction, and we therefore leave mesh optimization as an opportunity for future work.

E.4 TEMPERATURE AND GEOPOTENTIAL EXTENDED RESULTS

We report the RMSE and ACC curves for all lead times. We see in Fig. 9 that R-UNIMESH stays valid for the longest lead time in terms of ACC, and is rivaled only by GemCNN in terms of RMSE for geopotential. The results for temperature are given in Fig. 8. The results of Fig. 8 and Fig. 9 are also given as a table in Tab. 7.

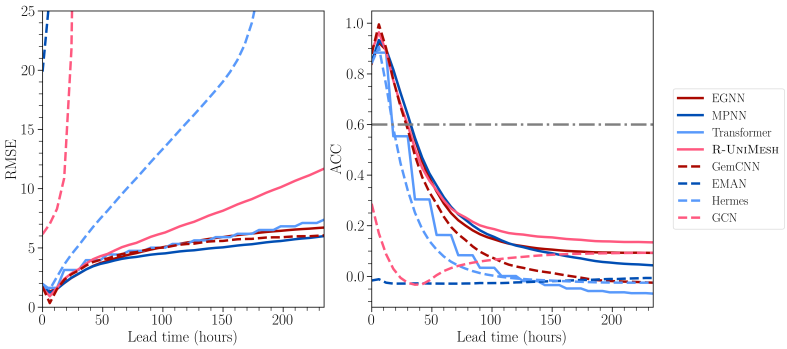


Figure 8: RMSE and ACC as a function of lead time for all models temperature prediction. R-UNIMESH has a competitive RMSE, especially at early lead time. R-UNIMESH also maintains viability for lead times of roughly 2 days according to the ECMWF baseline.

E.5 SMOOTHNESS EXTENDED RESULTS

We provide smoothness errors for all models on WB2 temperature and geopotential datasets. As seen in Tab. 8, R-UNIMESH is competitive across both temperature and geopotential, and is within statistical significance of the best performing EGNN model on geopotential.

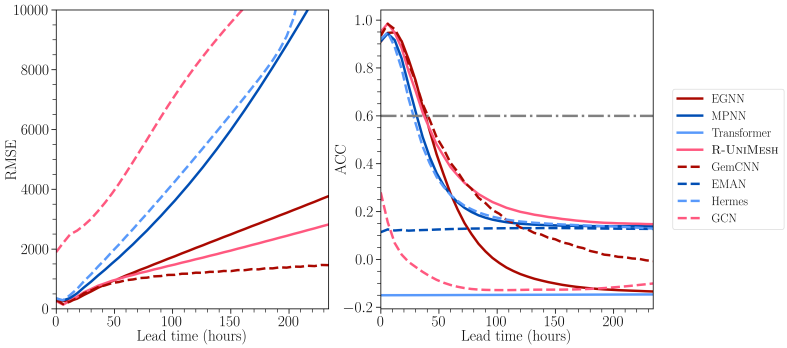


Figure 9: RMSE and ACC as a function of lead time for all models geopotential prediction. R-UniMesh has a competitive RMSE, especially at early lead time. R-UniMesh also maintains viability for lead times of roughly 2 days according to the ECMWF baseline.

Model	Metric	6h	12h	18h	24h	30h	36h	42h	48h	240h
EGNN	Z500 ACC	0.909	0.946	0.948	0.912	0.835	0.747	0.639	0.538	-0.134
	Z500 RMSE	347.83	266.96	261.09	340.43	467.68	581.20	706.01	812.70	3769.31
	T850 ACC	0.840	0.923	0.894	0.817	0.725	0.637	0.541	0.456	0.093
	T850 RMSE	1.94	1.32	1.54	2.02	2.46	2.81	3.17	3.47	6.71
EMAN	Z500 ACC	0.114	0.125	0.121	0.123	0.120	0.122	0.123	0.123	0.127
	Z500 RMSE	39530.01	52028.53	65037.68	89210.70	103179.47	119529.81	139314.85	156786.23	4898180.46
	T850 ACC	-0.016	-0.011	-0.025	-0.029	-0.029	-0.028	-0.028	-0.028	-0.007
	T850 RMSE	19.87	26.34	29.95	33.25	35.42	37.73	39.95	43.95	1097293305.30
GCN	Z500 ACC	0.280	0.155	0.068	0.020	-0.011	-0.032	-0.052	-0.070	-0.101
	Z500 RMSE	1897.27	2205.72	2491.37	2616.58	2798.23	3009.40	3272.80	3552.62	13147.85
	T850 ACC	0.287	0.170	0.087	0.030	-0.006	-0.025	-0.033	-0.031	0.093
	T850 RMSE	6.15	7.02	8.29	10.90	21.75	61.01	173.60	466.71	930146229837458.25
GemCNN	Z500 ACC	0.934	0.984	0.962	0.910	0.818	0.742	0.650	0.584	-0.007
	Z500 RMSE	304.11	156.00	242.86	364.36	521.75	615.11	724.60	785.92	1461.45
	T850 ACC	0.879	0.995	0.901	0.776	0.670	0.581	0.477	0.392	-0.025
	T850 RMSE	1.71	0.34	1.51	2.30	2.77	3.10	3.49	3.80	6.05
Hermes	Z500 ACC	0.919	0.948	0.888	0.795	0.670	0.568	0.472	0.402	0.129
	Z500 RMSE	339.23	289.93	463.62	667.90	927.73	1157.17	1416.43	1653.81	25644.40
	T850 ACC	0.842	0.906	0.756	0.586	0.447	0.340	0.252	0.187	-0.025
	T850 RMSE	1.96	1.50	2.55	3.59	4.45	5.20	5.99	6.76	293.82
MPNN	Z500 ACC	0.911	0.943	0.916	0.839	0.726	0.617	0.512	0.429	0.139
	Z500 RMSE	349.58	283.77	360.45	519.00	721.06	908.69	1116.36	1310.80	11191.46
	T850 ACC	0.846	0.933	0.898	0.817	0.724	0.638	0.547	0.469	0.043
	T850 RMSE	1.90	1.23	1.52	2.03	2.48	2.82	3.17	3.44	5.99
Transformer	Z500 ACC	-0.150	-0.150	-0.150	-0.150	-0.150	-0.150	-0.149	-0.149	-0.147
	Z500 RMSE	55368.55	55368.50	55368.45	55520.22	55520.17	55520.12	55558.25	55558.18	55570.73
	T850 ACC	0.884	0.884	0.884	0.553	0.553	0.553	0.304	0.304	-0.068
	T850 RMSE	1.62	1.62	1.62	3.13	3.13	3.13	3.96	3.96	7.37
R-UniMesh **(Ours)**	Z500 ACC	0.950	0.985	0.947	0.882	0.786	0.703	0.614	0.546	0.147
	Z500 RMSE	260.31	140.60	271.28	405.01	556.83	662.19	773.67	853.87	2821.03
	T850 ACC	0.888	0.964	0.889	0.774	0.679	0.598	0.504	0.428	0.134
	T850 RMSE	1.63	0.91	1.63	2.37	2.85	3.20	3.65	4.04	11.68

Table 7: Weather Forecasting Results, ACC and RMSE at Different Lead Times for Temperature and Geopotential.

Model	RE Temperature (↓)	RE Geopotential (↓)
GCN	$6.8 \cdot 10^{-2} \pm 1.7 \cdot 10^{-2}$	$1.2 \cdot 10^{-3} \pm 1.0 \cdot 10^{-3}$
MPNN	$3.0 \cdot 10^{-3} \pm 2.7 \cdot 10^{-3}$	$1.3 \cdot 10^{-3} \pm 8.5 \cdot 10^{-4}$
Hermes	$3.1 \cdot 10^{-1} \pm 7.5 \cdot 10^{-1}$	$1.0 \cdot 10^{-1} \pm 3.6 \cdot 10^{-1}$
GemCNN	$\mathbf{8.9 \cdot 10^{-4} \pm 7.0 \cdot 10^{-4}}$	$9.4 \cdot 10^{-4} \pm 6.1 \cdot 10^{-4}$
EMAN	$7.7 \cdot 10^{0} \pm 5.2 \cdot 10^{0}$	$4.1 \pm 2.6 \cdot 10^{-2}$
EGNN	$1.1 \cdot 10^{-3} \pm 7.3 \cdot 10^{-4}$	$\mathbf{8.3 \cdot 10^{-4} \pm 1.0 \cdot 10^{-3}}$
Transformer	$9.4 \cdot 10^{-4} \pm 8.2 \cdot 10^{-4}$	$2.0 \cdot 10^{-3} \pm 1.1 \cdot 10^{-3}$
R-UniMesh (Ours)	$2.2 \cdot 10^{-3} \pm 1.6 \cdot 10^{-3}$	$9.8 \cdot 10^{-4} \pm 7.2 \cdot 10^{-4}$

Table 8: Rayleigh error for all models for all initializations on WB2 temperature and geopotential. Best performing model is indicated with **bold text**. Errors scaled up by $\times 40$.