

---

# Smoothness Errors in Dynamics Models and How to Avoid Them

---

Anonymous Authors<sup>1</sup>

## 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 over-constraining. 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 both PDEs such as the heat and wave equations over complex meshes and on weather forecasting, we find that our method outperforms several strong baselines such as mesh-aware transformers and equivariant neural networks. Our code is available [here](#).

## 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

<sup>1</sup>Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.



*Figure 1.* Qualitative comparison of model performance for the heat equation on the armadillo mesh at timestep  $T = 190$ . 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. A more complete comparison over several timesteps is in Sec. C.4, Tab. 5.

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 networks 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) exhibit a tendency towards oversmoothing ([Li et al., 2018](#)), where adjacent node features become increasingly similar over successive iterations of message passing. The phenomenon of oversmoothing has been proven to occur 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 limits the approximation power of 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 show 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. Unitary graph convolutions thus often result in undersmoothing and do not give a full solution to the problem of smoothness errors.

Our first contribution is therefore to theoretically characterize the limitations of unitary convolutions for these dynamics problems. In particular, we derive lower bound on the approximation error of unitary functions and shows that unitary functions are overconstrained for dynamical systems where the solution’s norm has high angular dependence. To address this limitation, we propose instead *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:

- 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](#)).
- Introduce relaxed unitary convolutions that balance accuracy with smoothness preservation, and extend both the Rayleigh quotient and unitary convolution framework to meshes ([Sec. 5](#)).
- 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.** Our work quantifies the effect of neural networks on the Rayleigh quotient ([Chung, 1997](#)) of a graph, an approach similarly employed by [Kiani et al. \(2024\)](#). In addition, [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, our work illustrates theoretically and empirically how this property can be overconstraining in GNNs. Similar 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 no previous works have used the Rayleigh quotient as we do for dynamics models.

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 model-

ing specifically. Similarly, [Li et al. \(2018\)](#) point out that GCNs ([Kipf & Welling, 2017](#)) can be understood as a special case of Laplacian smoothing and 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 out information throughout the network. Despite these findings, there is comparatively less work studying the role of smoothness in spatio-temporal modeling tasks. Only [Marisca et al. \(2025\)](#) have studied issues with message-passing based GNNs for spatio-temporal modeling, and while [Marisca et al. \(2025\)](#) focus on *oversquashing*, our work is novel in our focus on *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 on 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; Mauzzi 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](#)), there is comparatively less work done to understand the smoothness. The performance of deep dynamics models are typically measured either via quantitative error metrics

110 against the ground truth or their preservation of underlying  
 111 physical laws, such as spectral energy errors (Wang et al.,  
 112 2021) or equivariance errors (Wang et al., 2021; 2022a;b).  
 113 Our work is novel in our application of the Rayleigh quotient  
 114 in quantifying the smoothing effect of trained GNN  
 115 deep dynamics models. Furthermore, we are among the first  
 116 to design architectures with inductive biases that encourage  
 117 the model to match the Rayleigh quotient of the labeled  
 118 graphs. Other works have explored using the Rayleigh quotient  
 119 as an auxiliary loss (Rowan et al., 2025) or positional  
 120 encodings (Dong et al., 2024), and works such as Kiani et al.  
 121 (2024) have developed a constrained model that preserves  
 122 the Rayleigh quotient regardless of the true labels. In con-  
 123 trast, only our work and the work of Shao et al. (2024) tries  
 124 to match the smoothness of labeled graphs via inductive  
 125 biases in the network architecture, and only our work as-  
 126 sseses how well the true smoothness of dynamical systems  
 127 are recovered in evaluation.

### 3. Background

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

#### 3.1. Rayleigh quotient

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

142 **Definition 1** (Rayleigh quotient, (Chung, 1997)). Given  
 143 an undirected graph  $\mathcal{G} = (V, E)$  with  $|V| = n$  nodes and  
 144 adjacency matrix  $\mathbf{A} \in \{0, 1\}^{n \times n}$ , let  $\mathbf{D} \in \mathbb{R}^{n \times n}$  be a  
 145 diagonal matrix where the  $i$ -th entry  $\mathbf{D}_{ii} = d_i$  the degree  
 146 of node  $i$ . Let  $s: V \rightarrow \mathbb{C}^d$  be a function from nodes to  
 147 features. Denote by  $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$  the normalized  
 148 adjacency matrix and  $\mathbf{X} \in \mathbb{C}^{n \times d}$  a matrix with the  $i$ -th row  
 149 set to feature vector  $s(i)$ . The Rayleigh quotient is

$$R_{\mathcal{G}}(\mathbf{X}) = \frac{1}{2} \frac{\sum_{(u,v) \in E} \left\| \frac{s(u)}{\sqrt{d_u}} - \frac{s(v)}{\sqrt{d_v}} \right\|^2}{\sum_{w \in V} \|s(w)\|^2} \quad (1)$$

150 or  $\text{Tr}(\mathbf{X}^\dagger (\mathbf{I} - \tilde{\mathbf{A}}) \mathbf{X}) \cdot \|\mathbf{X}\|_F^{-2}$  in matrix form. We will often  
 151 abbreviate the Laplacian as  $\mathbf{L} = (\mathbf{I} - \tilde{\mathbf{A}})$ .

152 Intuitively, the Rayleigh quotient measures the average dif-  
 153 ference in node features for nodes connected by an edge. A  
 154 graph with identical node features has a Rayleigh quotient  
 155 of zero.

#### 3.2. Unitary Convolution

156 Kiani et al. (2024) define two different models that preserve

110 the Rayleigh quotient using unitary functions, which sat-  
 111 isfy  $\mathbf{U}^\dagger \mathbf{U} = \mathbf{I}$ . In particular, they define separable unitary  
 112 convolution

$$f_{\text{UniConv}}^{\text{sep}}(\mathbf{X}; \mathbf{A}) = \exp(\mathbf{i} \mathbf{At}) \mathbf{X} \mathbf{U}, \quad \mathbf{U}^\dagger \mathbf{U} = \mathbf{I} \quad (2)$$

113 and Lie unitary convolution

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

114 where  $\exp(\cdot)$  denotes the matrix exponential. We provide  
 115 further background material on the matrix exponential and  
 116 its relationship to unitary matrices in Sec. A.1. The au-  
 117 thors show that unitary convolutions are mathematically  
 118 constrained to preserve the Rayleigh quotient:

**Proposition 1** (Invariance of Rayleigh quotient, Propo-  
 119 sition 6 in Kiani et al. (2024)). *Given an undirected  
 120 graph  $\mathcal{G}$  on  $n$  nodes with normalized adjacency matrix  
 121  $\mathbf{A} = \mathbf{D}^{-1/2} \mathbf{AD}^{-1/2}$ , the Rayleigh quotient is invariant  
 122 under normalized unitary or orthogonal graph convolution,  
 123 i.e.  $R_{\mathcal{G}}(\mathbf{X}) = R_{\mathcal{G}}(f_{\text{UniConv}}(\mathbf{X}))$  where  $f_{\text{UniConv}}$  is either  
 124 separable or Lie.*

#### 3.3. Mesh Data

125 A (triangular) mesh  $\mathcal{M}$  consists of a set  $(\mathcal{V}, \mathcal{E}, \mathcal{F})$ , where  $\mathcal{V}$   
 126 is a set of vertices,  $\mathcal{E} = \{(i, j)\}$  is a set of ordered vertex  
 127 indices  $i, j$  connected by an edge, and  $\mathcal{F} = \{(i, j, k)\}$  is the  
 128 set of ordered vertex indices  $i, j, k$  connected by a triangular  
 129 face. The mesh generalizes graphs by including high order  
 130 connectivity information via the inclusion of faces. We as-  
 131 sume that the mesh is a 2-dimensional manifold embedded  
 132 in  $\mathbb{R}^3$ , i.e. a manifold mesh. We recall the definition of the  
 133 manifold condition for a mesh in Definition 9 (Sec. A.7).

### 4. Theory: Unitary Models are Overconstrained

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

#### 4.1. Preliminaries

140 Let  $Z = \mathbb{C}^n$  be a domain with data probability density  
 141  $p: Z \rightarrow \mathbb{R}$ . Let  $u: \mathbb{C}^n \rightarrow \mathbb{C}^n$  be a unitary function and let  
 142  $f: \mathbb{C}^n \rightarrow \mathbb{C}^n$  be the target function. Denote the regression  
 143 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  
 144 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  $Gz = \{gz: g \in G\}$  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,  $Z = \bigcup_{g \in G} gF$ , where the conjugate is defined  $gF = \{gz: z \in F\}$ . Denote the integrated density on an orbit by  $p(Gz) = \int_{Gz} p(z) dz$ . Finally, denote the 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 2** (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 2 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 2 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 over-constrained when the norm of the ground truth function has a high angular dependence. Recall the definition of  $SU(n)$ , the group of rotations in  $\mathbb{C}^n$ :

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

We 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  $SU(n)$  in  $Z$ , e.g.  $F = \{te: t \in \mathbb{R}_+\}$  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 2. 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 over constraining when the ground truth is

not perfectly smoothness preserving, this section describes two methods for relaxing unitary convolution and details how to extend these architectures from graphs to meshes. We name these methods the Taylor truncation method and the encoder-decoder method. The Taylor truncation method has the practical advantage that it gives the ability to control the extent of the unitary relaxation, whereas with the encoder-decoder method it is easier to scale with the number of parameters in the network. 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  $f_{\text{Layer}}$  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

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

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  $T \rightarrow \infty$  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 . This method is especially useful compared to the encoder-decoder method in situations where the true smoothness is known a priori, in which case theoretical results from the literature (see Sec. A.8) can inform what  $\mathbf{T}_{\max}$  is needed to achieve enough relaxation.

## 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  $z: \mathbb{R}^{n \times d_{in}} \rightarrow \mathbb{R}^{n \times d_{out}}$  by  $\mathbf{X} \mapsto \mathbf{X} \oplus \mathbf{0}$ , where  $\mathbf{0}$  is the  $\mathbb{R}^{n \times (d_{out} - d_{in})}$  zero matrix. We define our  $k$ -layer encoder  $E$  by

$$E = f_{\text{UniConv}}^{(k)} \circ \dots \circ f_{\text{UniConv}}^{(1)}(z(\mathbf{X}), \mathbf{A}) \quad (5)$$

where  $f_{\text{UniConv}}$  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. A comparison between the encoder-decoder and Taylor truncation methods is shown in Fig. 2.

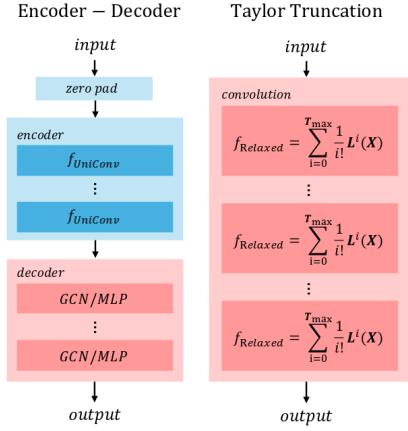


Figure 2. **Left:** After zero padding, individual unitary blocks are stacked and the output is fed into an unconstrained decoder. **Right:** Each block uses Taylor truncated unitary convolution. Smoothness preserving blocks are in blue.

## 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 discretizes. For a scalar function  $s$  defined on nodes,

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

where  $\mathcal{N}(i)$  denotes the adjacent vertices of  $i$ ,  $\alpha_{ij}$  and  $\beta_{ij}$  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  $\alpha_{ij}$  and  $\beta_{ij}$ ,  $\alpha_{ij} + \beta_{ij} \leq \pi$ .

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

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

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  $\mathcal{E}' = \{(u, v) : \mathcal{W}_{uv} \neq 0\}$ . Let  $s$  and  $\mathbf{X}$  be the same as in [Definition 1](#). The mesh Rayleigh quotient is defined

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

For further details and illustrations on the various Laplacians and Delaunay criterion, see Sec. A.7.

**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  $\tilde{\mathbf{A}}$ . In order to prove that incorporating these weights preserves the Rayleigh quotient given by Definition 3, we assume the mesh is already Delaunay without edge rewiring.

**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 in practice 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

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

and Lie unitary mesh convolution as

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

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 1). *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 on meshes 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  $H: \mathbb{R}_+ \rightarrow \mathbb{R}^n$  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  $\tau$  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  $T_{\max} = 3$  for the relaxed model. The mean Rayleigh quotient error is given by

$$\text{MRE}(f) = |\overline{R_{\mathcal{G}}(f(\mathbf{X}))} - \overline{R_{\mathcal{G}}(\mathbf{Y})}|.$$

**Results.** We see in Tab. 1 and Fig. 3 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 competitive with strong

Model	MSE ( $\downarrow$ )	MRE ( $\downarrow$ )
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)	<b><math>0.11 \cdot 10^{-2}</math></b>	<b><math>2.07 \cdot 10^{-2}</math></b>

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.

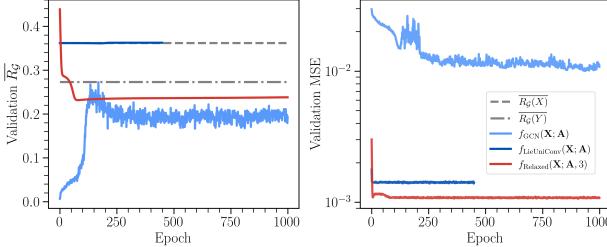


Figure 3. **Left:** The dotted lines indicate the mean Rayleigh quotient for input heat graphs at  $T = 3$  and target graphs at  $T = 4$ . We also show the mean Rayleigh quotient for the best performing GCN, R-UNIGRAPH , and Lie unitary models. R-UNIGRAPH is best at capturing the true smoothness. **Right:** Validation MSE fothe same three models. R-UNIGRAPH has the best performance. The unitary model is trained for less epochs due to compute limitations. Results for a larger ensemble of runs are provided in Sec. B.3.

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 a collection of challenging and realistic 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).

**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 .

**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. C.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. Sample meshes and initializations can be found in Sec. C.1. Training details are given in Sec. C.2. We will refer to this dataset as MeshPDE in the remainder of the paper.

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. D.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 ( $240 \times 120$ ) 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. D.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.

**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:

$$RE(f) = \frac{1}{T_{\max}} \sum_t^{T_{\max}} |R_{\mathcal{M}}(\mathbf{Y}_t) - R_{\mathcal{M}}(f(\mathbf{X}_t))|.$$

Further details on these metrics are provided in Sec. C.3 and we compare RE with more global smoothness metrics in Sec. C.5. We supplement these metrics with qualitative diagnostic figures in Sec. C.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

Metric	GCN	GemCNN	R-UNIMESH (Ours)	EMAN	Transformer	MPNN	EGNN	Hermes
Heat ( $c = 1$ )								
NRMSE ( $\downarrow$ )	-	-	<b>51.9 ± 3.6</b>	73.50 ± 3.8	92.5 ± 5.6	99.45 ± 4.8	344.25 ± 110.5	73.02 ± 4.7
SMAPE ( $\downarrow$ )	-	375.4 ± 0.53	<b>79.7 ± 5.6</b>	110.9 ± 13.3	213.9 ± 2.7	223.6 ± 1.5	319.33 ± 7.59	107.6 ± 7.4
RE ( $\downarrow$ )	-	52.21 ± 9.4	<b>9.1 ± 7.4</b>	14.2 ± 1.4	46.0 ± 3.7	76.06 ± 3.6	81.5 ± 8.77	39.76 ± 4.7
Wave ( $c = 1$ )								
NRMSE ( $\downarrow$ )	-	-	<b>236.5 ± 6.4</b>	<b>281.3 ± 15.5</b>	864.9 ± 184.9	563.6 ± 7.75	2280.1 ± 559.9	458.5 ± 13.0
SMAPE ( $\downarrow$ )	-	318.8 ± 3.9	385.2 ± 1.2	<b>301.0 ± 4.2</b>	327.0 ± 4.4	318.0 ± 2.8	354.3 ± 11.0	316.4 ± 4.5
RE ( $\downarrow$ )	-	107.9 ± 3.158	93.5 ± 25.4	73.57 ± 6.5	<b>48.0 ± 7.9</b>	139.3 ± 10.1	157.2 ± 14.8	70.03 ± 6.1
Cahn-Hilliard								
NRMSE ( $\downarrow$ )	-	<b>121.2 ± 1.8</b>	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 ( $\downarrow$ )	-	204.3 ± 2.4	<b>167.3 ± 10.6</b>	<b>143.7 ± 2.5</b>	191.7 ± 2.0	201.22 ± 32.79	336.5 ± 2.777	173.3 ± 4.3
RE ( $\downarrow$ )	-	<b>10.68 ± 3.3</b>	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.

coefficient between forecast anomalies and ground-truth anomalies relative to a climatological baseline. Precise definitions are provided in Sec. D.

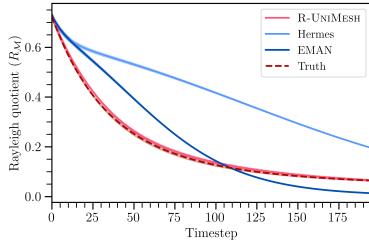


Figure 4. The Rayleigh quotient for each timestep on an unseen mesh for Hermes, EMAN, and R-UNIMESH models. The R-UNIMESH is the best at capturing the true smoothness for heat.

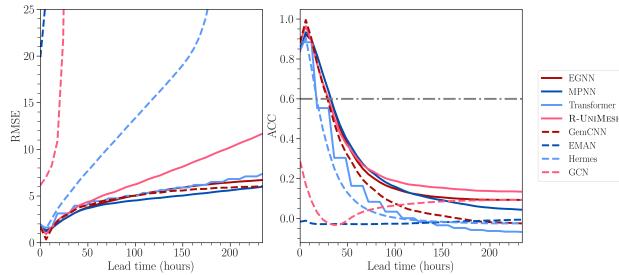


Figure 5. RMSE and ACC as a function of lead time for all models temperature prediction. R-UNIMESH has a competitive RMSE, especially at early lead time. R-UNIMESH also maintains viability for lead times of roughly 2 days according to the ECMWF baseline.

**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. 4. 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 Fig. 1 and in Sec. C.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. 5, 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 due to compute limitations, our best performing models in Fig. 5 are comparable with the state of the art (Figure 1, 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. D.4 that R-UNIMESH is among the best models for the geopotential prediction task, though all models are below the SOTA in Rasp et al. (2024). Finally, we see in Tab. 7 (Sec. D.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.

440  
441  
442  
443  
444  
445  
446  
447  
448  
449  
450  
451  
452  
453  
454  
455  
456  
457  
458  
459  
460  
461  
462  
463  
464  
465  
466  
467  
468  
469  
470  
471  
472  
473  
474  
475  
476  
477  
478  
479  
480  
481  
482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494**Impact Statement**

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

**References**

- Anandkumar, A., Azizzadenesheli, K., Bhattacharya, K., Kovachki, N., Li, Z., Liu, B., and Stuart, A. Neural operator: Graph kernel network for partial differential equations. In *ICLR 2020 workshop on integration of deep neural models and differential equations*, 2020.
- Anil, C., Lucas, J., and Grosse, R. Sorting out lipschitz function approximation. In *International conference on machine learning*, pp. 291–301. PMLR, 2019.
- Arroyo, A., Gravina, A., Gutteridge, B., Barbero, F., Gallicchio, C., Dong, X., Bronstein, M. M., and Vandergheynst, P. 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>.
- Artin, M. *Algebra*. Birkhäuser, 1998.
- Balla, J. Over-squashing in riemannian graph neural networks. In *The Second Learning on Graphs Conference*, 2023. URL <https://openreview.net/forum?id=UUnYi0yLcM>.
- baron de Fourier, J. B. J. *Théorie analytique de la chaleur*. Firmin Didot, 1822.
- Basu, S., Gallego-Posada, J., Viganò, F., Rowbottom, J., and Cohen, T. Equivariant mesh attention networks. *Transactions on Machine Learning Research*, 2022. ISSN 2835-8856. URL <https://openreview.net/forum?id=3IqqJh2Ycy>. Expert Certification.
- Bobenko, A. I. and Springborn, B. A. A discrete laplace–beltrami operator for simplicial surfaces. *Discrete & Computational Geometry*, 38(4):740–756, 2007.
- Bodnar, C., Di Giovanni, F., Chamberlain, B., Lio, P., and Bronstein, M. Neural sheaf diffusion: A topological perspective on heterophily and oversmoothing in gnns. *Advances in Neural Information Processing Systems*, 35: 18527–18541, 2022.
- Borovitskiy, V., Azangulov, I., Terenin, A., Mostowsky, P., Deisenroth, M., and Durrande, N. Matérn gaussian processes on graphs. In *International Conference on Artificial Intelligence and Statistics*, pp. 2593–2601. PMLR, 2021.
- Cahn, J. W. and Hilliard, J. E. Free energy of a nonuniform system. i. interfacial free energy. *The Journal of chemical physics*, 28(2):258–267, 1958.
- Cai, C. and Wang, Y. A note on over-smoothing for graph neural networks. *arXiv preprint arXiv:2006.13318*, 2020.

- 495 Cai, S., Wang, Z., Wang, S., Perdikaris, P., and Karniadakis,  
 496 G. E. Physics-informed neural networks for heat trans-  
 497 fer problems. *Journal of Heat Transfer*, 143(6):060801,  
 498 2021.
- 500 Chung, F. R. *Spectral graph theory*, volume 92. American  
 501 Mathematical Soc., 1997.
- 502 Cohen, T., Weiler, M., Kicanaoglu, B., and Welling, M.  
 503 Gauge equivariant convolutional networks and the icoso-  
 504 hedral cnn. In *International conference on Machine learn-  
 505 ing*, pp. 1321–1330. PMLR, 2019.
- 506 Constantin, P. and Foiaş, C. *Navier-stokes equations*. Uni-  
 507 versity of Chicago press, 1988.
- 508 Crane, K., de Goes, F., Desbrun, M., and Schröder, P. Digital  
 509 geometry processing with discrete exterior calculus. In  
 510 *ACM SIGGRAPH 2013 Courses*, New York, NY, USA,  
 511 2013. ACM.
- 512 Crane, K., Weischedel, C., and Wardetzky, M. The heat  
 513 method for distance computation. *Communications of the  
 514 ACM*, 60(11):90–99, 2017.
- 515 Cranmer, M., Greydanus, S., Hoyer, S., Battaglia, P.,  
 516 Spergel, D., and Ho, S. Lagrangian neural networks.  
 517 In *ICLR 2020 Workshop on Integration of Deep Neural  
 518 Models and Differential Equations*, 2020.
- 519 Cui, Q., Zhang, M., Xiao, M., and Ni, G. Transformer  
 520 based deep learning accelerated numerical simulation for  
 521 incompressible flow. *Physics of Fluids*, 36(12), 2024.
- 522 d'Alembert, J. I. R. Recherches sur la courbe que forme  
 523 une corde tendue mise en vibration. [https://www.exhibit.xavier.edu/oresme\\_2017Sept/4](https://www.exhibit.xavier.edu/oresme_2017Sept/4),  
 524 1747. Presented at the ORESME Reading Group  
 525 Meeting, September 30, 2017.
- 526 Daniels, M. and Rigollet, P. Splat regression models. *arXiv  
 527 preprint arXiv:2511.14042*, 2025.
- 528 Daniels, M., Hodgkinson, L., and Mahoney, M. Uncertainty-  
 529 aware diagnostics for physics-informed machine learning.  
 530 *arXiv preprint arXiv:2510.26121*, 2025.
- 531 de Haan, P., Weiler, M., Cohen, T., and Welling, M. Gauge  
 532 equivariant mesh cnns: Anisotropic convolutions on geo-  
 533 metric graphs. In *International Conference on Learning  
 534 Representations*, 2021. URL <https://openreview.net/forum?id=Jnspzp-oIZE>.
- 535 Defferrard, M., Martin, L., Pena, R., and Perraudin, N.  
 536 Pygsp: Graph signal processing in python, 2017. URL  
 537 <https://github.com/epfl-lts2/pygsp/>.
- 538 Dong, X., Zhang, X., and Wang, S. Rayleigh quotient  
 539 graph neural networks for graph-level anomaly detection.  
 540 In *The Twelfth International Conference on Learning  
 541 Representations*, 2024. URL <https://openreview.net/forum?id=4UIBysXjVq>.
- 542 Esteves, C. Theoretical aspects of group equivariant neural  
 543 networks. *arXiv preprint arXiv:2004.05154*, 2020.
- 544 Ferrandi, G. and Hochstenbach, M. E. A homogeneous  
 545 rayleigh quotient with applications in gradient methods.  
 546 *Journal of Computational and Applied Mathematics*, 437:  
 547 115440, 2024.
- 548 Gao, W., Xu, R., Deng, Y., and Liu, Y. Discretization-  
 549 invariance? on the discretization mismatch errors in neu-  
 550 ral operators. In *The Thirteenth International Conference  
 551 on Learning Representations*, 2025.
- 552 Ghil, M. and Simonnet, E. Geophysical fluid dynamics,  
 553 nonautonomous dynamical systems, and the climate sci-  
 554 ences. In *Mathematical Approach to Climate Change and  
 555 its Impacts: MAC2I*, pp. 3–81. Springer, 2020.
- 556 Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and  
 557 Dahl, G. E. Neural message passing for quantum chem-  
 558 istry. In *International conference on machine learning*,  
 559 pp. 1263–1272. Pmlr, 2017.
- 560 Gruver, N., Finzi, M. A., Goldblum, M., and Wilson, A. G.  
 561 The lie derivative for measuring learned equivariance.  
 562 In *The Eleventh International Conference on Learning  
 563 Representations*, 2023. URL <https://openreview.net/forum?id=JL7Va5Vy15J>.
- 564 Gupta, A. and Lermusiaux, P. F. Generalized neural closure  
 565 models with interpretability. *Scientific Reports*, 13(1):  
 566 10634, 2023.
- 567 Hall, B. C. Lie groups, lie algebras, and representations.  
 568 In *Quantum Theory for Mathematicians*, pp. 333–366.  
 569 Springer, 2013.
- 570 Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi,  
 571 A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu,  
 572 R., Schepers, D., Simmons, A., Soci, C., Abdalla, S.,  
 573 Abellán, X., Balsamo, G., Bechtold, P., Biavati, G.,  
 574 Bidlot, J., Bonavita, M., De Chiara, G., Dahlgren, P.,  
 575 Dee, D., Diamantakis, M., Dragani, R., Flemming, J.,  
 576 Forbes, R., Fuentes, M., Geer, A., Haimberger, L., Healy,  
 577 S., Hogan, R. J., Hólm, E., Janisková, M., Keeley, S.,  
 578 Laloyaux, P., Lopez, P., Lupo, C., Radnoti, G., de Ros-  
 579 nay, P., Rozum, I., Vamborg, F., Villaume, S., and  
 580 Thépaut, J.-N. The era5 global reanalysis. *Quarterly  
 581 Journal of the Royal Meteorological Society*, 146(730):  
 582 1999–2049, 2020. doi: <https://doi.org/10.1002/qj.3803>.  
 583 URL <https://rmets.onlinelibrary.wiley.com/doi/abs/10.1002/qj.3803>.

- 550 Huang, J., Su, H., and Guibas, L. Robust watertight man-  
 551 ifold surface generation method for shapenet models.  
 552 *arXiv preprint arXiv:1802.01698*, 2018.
- 553 Huang, J., Yang, G., Wang, Z., and Park, J. J. Diffusion-  
 554 pde: Generative pde-solving under partial observation.  
 555 *Advances in Neural Information Processing Systems*, 37:  
 556 130291–130323, 2024.
- 557 Janny, S., Bénéteau, A., Nadri, M., Digne, J., Thome, N.,  
 558 and Wolf, C. EAGLE: Large-scale learning of turbulent  
 559 fluid dynamics with mesh transformers. In *The Eleventh  
 560 International Conference on Learning Representations*,  
 561 2023. URL <https://openreview.net/forum?id=mfIX4QpsARJ>.
- 562 Jarvis, M., Bernstein, G., and Jain, B. The skewness of  
 563 the aperture mass statistic. *Monthly Notices of the Royal  
 564 Astronomical Society*, 352(1):338–352, 2004.
- 565 Keriven, N. Not too little, not too much: a theoretical  
 566 analysis of graph (over) smoothing. *Advances in Neural  
 567 Information Processing Systems*, 35:2268–2281, 2022.
- 568 Kiani, B., Fesser, L., and Weber, M. Unitary convolutions  
 569 for learning on graphs and groups. *Advances in Neu-  
 570 ral Information Processing Systems*, 37:136922–136961,  
 571 2024.
- 572 Kipf, T. N. and Welling, M. Semi-supervised classi-  
 573 fication with graph convolutional networks. In *Inter-  
 574 national Conference on Learning Representations*,  
 575 2017. URL <https://openreview.net/forum?id=SJU4ayYgl>.
- 576 Kulick, C., Birnir, B., and Tang, S. Investigating zero-shot  
 577 size transfer of graph neural differential equations for  
 578 learning graph diffusion dynamics. In *Topology, Algebra,  
 579 and Geometry in Data Science*, 2025. URL <https://openreview.net/forum?id=qgbyLknKXY>.
- 580 Lam, R., Sanchez-Gonzalez, A., Willson, M., Wirnsberger,  
 581 P., Fortunato, M., Alet, F., Ravuri, S., Ewalds, T., Eaton-  
 582 Rosen, Z., Hu, W., Merose, A., Hoyer, S., Holland,  
 583 G., Vinyals, O., Stott, J., Pritzel, A., Mohamed, S.,  
 584 and Battaglia, P. Graphcast: Learning skillful medium-  
 585 range global weather forecasting, 2023. URL <https://arxiv.org/abs/2212.12794>.
- 586 Leman, A. and Weisfeiler, B. A reduction of a graph to a  
 587 canonical form and an algebra arising during this reduc-  
 588 tion. *Nauchno-Technicheskaya Informatsiya*, 2(9):12–16,  
 589 1968.
- 590 Li, Q., Han, Z., and Wu, X.-M. Deeper insights into graph  
 591 convolutional networks for semi-supervised learning. In  
 592 *Proceedings of the AAAI conference on artificial intelli-  
 593 gence*, volume 32, 2018.
- 594 Li, W., Fang, R., Jiao, J., Vassilakis, G. N., and Zhu, J. Tu-  
 595 torials: Physics-informed machine learning methods of  
 596 computing 1d phase-field models. *APL Machine Learn-  
 597 ing*, 2(3), 2024.
- 598 Li, Z., Kovachki, N. B., Azizzadenesheli, K., liu, B., Bhat-  
 599 tacharya, K., Stuart, A., and Anandkumar, A. Fourier neu-  
 600 ral operator for parametric partial differential equations.  
 601 In *International Conference on Learning Representations*,  
 602 2021. URL <https://openreview.net/forum?id=c8P9NQVtmnO>.
- 603 Liu, Z., Wang, Y., Vaidya, S., Ruehle, F., Halverson, J., Sol-  
 604 jacic, M., Hou, T. Y., and Tegmark, M. Kan: Kolmogorov-  
 605 arnold networks. In *The Thirteenth International Confer-  
 606 ence on Learning Representations*, 2024.
- 607 Marasca, I., Bamberger, J., Alippi, C., and Bronstein, M. M.  
 608 Over-squashing in spatiotemporal graph neural networks.  
 609 *arXiv preprint arXiv:2506.15507*, 2025.
- 610 Maurizi, M., Gao, C., and Berto, F. Predicting stress, strain  
 611 and deformation fields in materials and structures with  
 612 graph neural networks. *Scientific reports*, 12(1):21834,  
 613 2022.
- 614 Meyer, M., Desbrun, M., Schröder, P., and Barr, A. H. Dis-  
 615 crete differential-geometry operators for triangulated 2-  
 616 manifolds. In *Visualization and mathematics III*, pp. 35–  
 617 57. Springer, 2003.
- 618 Mishayev, Y., Sverdlov, Y., Amir, T., and Dym, N. Short-  
 619 range oversquashing. In *The Fourth Learning on Graphs  
 620 Conference*, 2025.
- 621 Mitchel, T. W., Kim, V. G., and Kazhdan, M. Field convolu-  
 622 tions for surface cnns. In *Proceedings of the IEEE/CVF  
 623 International Conference on Computer Vision*, pp. 10001–  
 624 10011, 2021.
- 625 Mitchel, T. W., Aigerman, N., Kim, V. G., and Kazhdan, M.  
 626 Möbius convolutions for spherical cnns. In *ACM SIG-  
 627 GRAPH 2022 Conference Proceedings*, pp. 1–9, 2022.
- 628 Mitchel, T. W., Esteves, C., and Makadia, A. Single mesh  
 629 diffusion models with field latents for texture generation.  
 630 In *Proceedings of the IEEE/CVF Conference on Com-  
 631 puter Vision and Pattern Recognition*, pp. 7953–7963,  
 632 2024.
- 633 Morel, R., Ramunno, F. P., Shen, J., Bietti, A., Cho, K.,  
 634 Cranmer, M., Golkar, S., GUGNIN, O., Krawezik, G.,  
 635 Marwah, T., et al. Predicting partially observable dy-  
 636 namical systems via diffusion models with a multiscale  
 637 inference scheme. In *The Thirty-ninth Annual Conference  
 638 on Neural Information Processing Systems*, 2025.

- 605 Mustafa, M., Wu, J., Jiang, C., Wang, R., et al. Physics-  
 606 informed machine learning: case studies for weather and  
 607 climate modelling. *Philosophical Transactions of the*  
 608 *Royal Society A*, 379(2194):20200093, 2021.
- 609
- 610 NVIDIA. Nvidia h200 tensor core gpu datasheet,  
 611 2025. URL <https://resources.nvidia.com/en-us-data-center-overview-mc/en-us-data-center-overview/hpc-datasheet-sc23-h200>. Retrieved from  
 612 NVIDIA website.
- 613
- 614 Olver, P. J. *Applications of Lie groups to differential equations*, volume 107. Springer Science & Business Media,  
 615 1993.
- 616
- 617 Pandya, S., Yang, Y., Van Alfen, N., Blazek, J., and Walters,  
 618 R. Iaemu: Learning Galaxy Intrinsic Alignment Corre-  
 619 lations. *The Open Journal of Astrophysics*, 8, dec 2 2025.  
 620 doi: 10.33232/001c.151749.
- 621
- 622 Park, J. Y., Wong, L., and Walters, R. Modeling dynamics  
 623 over meshes with gauge equivariant nonlinear message  
 624 passing. *Advances in Neural Information Processing Systems*, 36:15277–15302, 2023.
- 625
- 626 Pfaff, T., Fortunato, M., Sanchez-Gonzalez, A., and  
 627 Battaglia, P. Learning mesh-based simulation with graph  
 628 networks. In *International Conference on Learning Rep-  
 629 resentations*, 2021. URL [https://openreview.net/forum?id=roNqYL0\\_XP](https://openreview.net/forum?id=roNqYL0_XP).
- 630
- 631 Rasp, S., Hoyer, S., Merose, A., Langmore, I., Battaglia,  
 632 P., Russell, T., Sanchez-Gonzalez, A., Yang, V., Carver,  
 633 R., Agrawal, S., et al. Weatherbench 2: A benchmark for  
 634 the next generation of data-driven global weather models.  
*Journal of Advances in Modeling Earth Systems*, 16(6):  
 e2023MS004019, 2024.
- 635
- 636 Reuter, M., Biasotti, S., Giorgi, D., Patanè, G., and Spagn-  
 637 uolo, M. Discrete laplace–beltrami operators for shape  
 638 analysis and segmentation. *Computers & Graphics*, 33  
 639 (3):381–390, 2009.
- 640
- 641 Rowan, C., Doostan, A., Maute, K., and Evans, J. Solving  
 642 engineering eigenvalue problems with neural networks  
 643 using the rayleigh quotient. *International Journal for Nu-  
 644 matical Methods in Engineering*, 126(24):e70209, 2025.
- 645
- 646 Rusch, T. K., Bronstein, M. M., and Mishra, S. A survey on  
 647 oversmoothing in graph neural networks. *arXiv preprint*  
 648 *arXiv:2303.10993*, 2023.
- 649
- 650 Satorras, V. G., Hoogeboom, E., and Welling, M. E (n)  
 651 equivariant graph neural networks. In *International con-  
 652 ference on machine learning*, pp. 9323–9332. PMLR,  
 653 2021.
- 654
- 655 Schlaginhaufen, A., Wenk, P., Krause, A., and Dörfler, F.  
 656 Learning stable deep dynamics models for partially ob-  
 657 served or delayed dynamical systems. In Beygelzimer,  
 658 A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.),  
 659 *Advances in Neural Information Processing Systems*,  
 660 2021. URL <https://openreview.net/forum?id=u8HmtBBSVJS>.
- 661
- Schneider, P. Weak gravitational lensing. In *Gravitational  
 662 lensing: strong, weak and micro*, pp. 269–451. Springer,  
 663 2006.
- 664
- 665 Shao, Z., Shi, D., Han, A., Guo, Y., Zhao, Q., and Gao, J.  
 666 Unifying over-smoothing and over-squashing in graph  
 667 neural networks: A physics informed approach and  
 668 beyond, 2024. URL <https://openreview.net/forum?id=swPf2hwK18>.
- 669
- 670 Sharp, N. and Crane, K. A laplacian for nonmanifold trian-  
 671 gle meshes. In *Computer Graphics Forum*, volume 39,  
 672 pp. 69–80. Wiley Online Library, 2020.
- 673
- 674 Su, J. and Wu, C. On the interplay between graph struc-  
 675 ture and learning algorithms in graph neural networks.  
 676 In *Forty-second International Conference on Machine  
 677 Learning*, 2025.
- 678
- Suk, J., de Haan, P., Lippe, P., Brune, C., and Wolterink,  
 679 J. M. Mesh neural networks for se (3)-equivariant hem-  
 680 odyamics estimation on the artery wall. *Computers in  
 681 biology and medicine*, 173:108328, 2024.
- 682
- Sullivan, C. and Kaszynski, A. Pyvista: 3d plotting and  
 683 mesh analysis through a streamlined interface for the visu-  
 684 alization toolkit (vtk). *Journal of Open Source Software*,  
 685 4(37):1450, 2019.
- 686
- Tönshoff, J., Ritzert, M., Rosenbluth, E., and Grohe, M.  
 687 Where did the gap go? reassessing the long-range graph  
 688 benchmark. In *The Second Learning on Graphs Confer-  
 689 ence*, 2023.
- 690
- Tönshoff, J., Ritzert, M., Rosenbluth, E., and Grohe, M.  
 691 Where did the gap go? reassessing the long-range graph  
 692 benchmark. *Transactions on Machine Learning Research*,  
 693 2024.
- 694
- 695 Wang, D., Zhu, X., Park, J. Y., Jia, M., Su, G., Platt, R., and  
 696 Walters, R. A general theory of correct, incorrect, and  
 697 extrinsic equivariance. *Advances in Neural Information  
 698 Processing Systems*, 36:40006–40029, 2023.
- 699
- 700 Wang, R., Kashinath, K., Mustafa, M., Albert, A., and  
 701 Yu, R. Towards physics-informed deep learning for  
 702 turbulent flow prediction. In *Proceedings of the 26th  
 703 ACM SIGKDD International Conference on Knowledge  
 704 Discovery & Data Mining*, KDD ’20, pp. 1457–1466,

660  
661 New York, NY, USA, 2020. Association for Comput-  
662 ing Machinery. ISBN 9781450379984. doi: 10.1145/  
663 3394486.3403198. URL <https://doi.org/10.1145/3394486.3403198>.

664  
665 Wang, R., Walters, R., and Yu, R. Incorporating symmetry  
666 into deep dynamics models for improved generalization.  
667 In *International Conference on Learning Representations*,  
668 2021. URL [https://openreview.net/forum?id=wta\\_8Hx2KD](https://openreview.net/forum?id=wta_8Hx2KD).

670  
671 Wang, R., Walters, R., and Yu, R. Approximately equivari-  
672 ant networks for imperfectly symmetric dynamics. In *In-*  
673 *ternational Conference on Machine Learning*, pp. 23078–  
674 23091. PMLR, 2022a.

675  
676 Wang, R., Walters, R., and Yu, R. Data augmentation vs.  
677 equivariant networks: A theory of generalization on dy-  
678 namics forecasting. *International Conference on Machine*  
679 *Learning (ICML) Principles of Distribution Shift Work-*  
680 *shop*, 2022b.

681  
682 Yu, R. and Wang, R. Learning dynamical systems from  
683 data: An introduction to physics-guided deep learn-  
684 ing. *Proceedings of the National Academy of Sci-*  
685 *ences*, 121(27):e2311808121, 2024. doi: 10.1073/pnas.  
686 2311808121. URL <https://www.pnas.org/doi/abs/10.1073/pnas.2311808121>.

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

## 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

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

as well as the special unitary group

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

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

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

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

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

Applying the exponential map to a linear operator is given by

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

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

### A.2. Convolutional oversmoothing

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

**Proposition 3** (Proposition 7 in Kiani et al. (2024)). *Given a simple undirected graph  $\mathcal{G}$  on  $n$  nodes with normalized adjacency matrix  $\tilde{\mathbf{A}} = \mathbf{D}^{-1/2} \mathbf{A} \mathbf{D}^{-1/2}$  and node degree bounded by  $D$ , let  $\mathbf{X} \in \mathbb{R}^{n \times d}$  have rows drawn i.i.d. from the uniform distribution on the hypersphere in dimension  $d$ . Let  $f_{conv}(\mathbf{X}) = \mathbf{A}\mathbf{X}\mathbf{W}$  denote convolution with orthogonal feature transformation matrix  $\mathbf{W} \in O(d)$ . Then, the event below holds with probability  $1 - \exp(-\Omega(\sqrt{n}))$ :*

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

### A.3. 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.

770 Let  $f : \mathcal{X} \rightarrow \mathcal{Y}$  be a map between input and output vector spaces  $\mathcal{X}$  and  $\mathcal{Y}$ . Let  $G$  be a group with  
 771 representations  $\rho^{\mathcal{X}}$  and  $\rho^{\mathcal{Y}}$  which transform vectors in  $\mathcal{X}$  and  $\mathcal{Y}$  respectively. Representations are group homomorphisms  
 772 which map group elements to invertible linear transformations. The map  $f : \mathcal{X} \rightarrow \mathcal{Y}$  is *equivariant* if  
 773

$$774 \quad \rho^{\mathcal{Y}}(g)f(x) = f(\rho^{\mathcal{X}}(g)x), \text{ for all } g \in G, x \in \mathcal{X}.$$

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

778 **Definition 5.** A map  $f : \mathcal{X} \rightarrow \mathcal{Y}$  is *invariant* if  
 779

$$780 \quad f(x) = f(\rho^{\mathcal{X}}(g)x), \text{ for all } g \in G, x \in \mathcal{X}.$$

### 782 A.3.1. EUCLIDEAN EQUIVARIANCE

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

787 **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.  
 788 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

$$789 \quad 790 \quad f(Qx + t) = Qf(x) + t.$$

### 792 A.3.2. GAUGE EQUIVARIANCE

793 We may also choose to embed coordinates locally, using coordinates that are intrinsic to the 2D mesh rather than the extrinsic  
 794 3D coordinates of the embedding space. This approach arises from the desire for a general convolution-like operator  
 795 over arbitrary manifolds discretized as a mesh. To encode data over a mesh it is still necessary to make a choice of local  
 796 coordinate frame at each vertex. In order to guarantee the equivalence of the features resulting from different choices of  
 797 reference frames, the model should be invariant to change of coordinates frame at each vertex, i.e. gauge equivariant.  
 798

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

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

$$807 \quad K_{\text{neigh}}(\theta - g) = \rho_{\text{out}}(-g)K_{\text{neigh}}(\theta)\rho_{\text{in}}(g), \quad K_{\text{self}} = \rho_{\text{out}}(-g)K_{\text{self}}\rho_{\text{in}}(g).$$

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

## 814 A.4. Unitary Learning Framework

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

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

819 **Definition 8** (Fundamental Domain, Definition 4.1 in [Wang et al. \(2023\)](#)). A closed subset  $F$  of  $X$  is called a fundamental  
 820 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  
 821 measure 0 under  $\nu$ .

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 2](#). Denote by  $q(z) = \frac{p(z)}{p(Gx)}$  the density of the orbit  $Gx$  so that  $\int_{Gx} q(z) dz = 1$ . The mean and variance of a function  $f$  on  $Gx$  are given by

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

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

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

*Proof of Theorem 1.* By the reverse triangle inequality,

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

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

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  $D_2 = \{(x, y) : x^2 + y^2 \leq 1\}$ . 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

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

On each orbit, we compute

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

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 1](#)). 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 1](#) 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 strategy in [Kiani et al. \(2024\)](#) still holds.

*Proof.* We first prove invariance for [Eq. 8](#). By the circulant property of the trace,

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

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

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

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 as a function that acts on a vector in  $\mathbb{C}^{nd}$ . By properties of the Kronecker tensor product,

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

Since

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

we have that  $(\mathbf{A} \otimes \mathbf{W}^T)$  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  $\exp(\tilde{\mathbf{A}} \otimes \mathbf{W}^T)$  commutes with  $(\tilde{\mathbf{A}} \otimes \mathbf{I})$ . Thus,

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

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

**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.

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

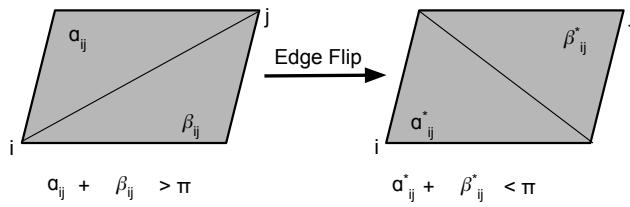
$$937 \quad (\tilde{\mathbf{L}}(s))_i = \frac{1}{2A_i} \sum_{j \in \mathcal{N}(i)} (\cot \alpha_{ij} + \cot \beta_{ij}) (s_j - s_i)$$

940 where  $\mathcal{N}(i)$  denotes the adjacent vertices of  $i$ ,  $\alpha_{ij}$  and  $\beta_{ij}$  are the angles opposite edge  $(i, j)$ , and  $A_i$  is the vertex area of  $i$   
 941 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  
 942 let  $\mathcal{F}(i)$  be the set of faces containing vertex  $i$ . The barycentric cell area is defined

$$943 \quad A_i = \sum_{abc \in \mathcal{F}(i)} \mathcal{A}_{abc} / 3.$$

946 Normalization by the cell area was used for the dataset construction in Park et al. (2023), but it is not used in the definition  
 947 of the Robust Laplacian (Sharp & Crane, 2020). In fact, the cell area introduces asymmetry in the edge weights. This is  
 948 undesirable, as unitary mesh convolution depends on symmetric edge weights in order to preserve the Rayleigh quotient.  
 949

950 **Cotangent Laplacian Edge Weights and Robust Rewiring.** As noted in Sec. 5.3, an arbitrary mesh may have negative  
 951 cotangent weights. These cotangent weights have the following geometric meaning. For vertices  $ij$  connected by an edge,  
 952 we say that the edge is *primal*. For manifold meshes, for each primal edge there exists a *dual edge* that connects the triangle  
 953 circumcenters for the two triangles that share a primal edge. The cotangent weights correspond to the ratio of the primal  
 954 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  
 955 Robust Laplacian applies two edge rewiring algorithms sequentially, the tufted cover algorithm and the Delaunay edge flip  
 956 algorithm. The tufted cover algorithm ensures the mesh is manifold so that the Delaunay edge flip algorithm can be applied.  
 957 The Delaunay edge flip algorithm then ensures that the mesh satisfies the intrinsic Delaunay criterion (Definition 2). A  
 958 sample edge flip is illustrated in Fig. 6.



960 *Figure 6.* Illustration of an intrinsic Delaunay edge flip performed by the Robust Laplacian edge rewiring algorithm. This figure is a  
 961 reproduced version of Figure 7, Sharp & Crane (2020).

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

## 969 A.8. Rayleigh Quotient Sensitivity

970 We include results from Ferrandi & Hochstenbach (2024) and Dong et al. (2024) that illustrate the sensitivity of the Rayleigh  
 971 quotient to small perturbations of the input, such as Taylor series truncation errors. While the hypotheses are stronger than  
 972 what we may actually see in practice, the following proposition provides an intuition for the Rayleigh quotient sensitivity.

973 **Proposition 4** (Proposition 4 in Ferrandi & Hochstenbach (2024)). *Suppose  $\mathbf{u} = \mathbf{x} + \mathbf{e}$  is an approximate eigenvector  
 974 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,  
 975 for the sensitivity of the Rayleigh quotient (as a function of  $\mathbf{u}$ ) it holds that*

$$976 \quad \min_{\lambda_i \neq \lambda} \frac{|\lambda_i - \lambda|}{|\lambda_i|} \varepsilon^2 \lesssim \frac{|R_G(\mathbf{u}) - \lambda|}{|\lambda|} \lesssim \max_{\lambda_i \neq \lambda} \frac{|\lambda_i - \lambda|}{|\lambda_i|} \varepsilon^2.$$

This indicates that the Rayleigh quotient sensitivity is quadratic in perturbations  $\varepsilon$ . 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  $\Delta$  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  $\delta$  on  $\mathbf{x}$ , the change of Rayleigh quotient can be bounded by  $2\mathbf{x}^T \mathbf{L}\delta + o(\delta)$ . If  $\delta$  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. 7.

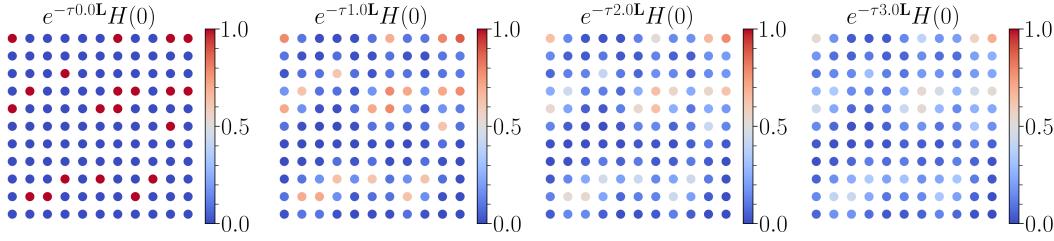


Figure 7. Sample heat diffusion process on a grid discretized as a graph. Node neighbors are the nodes that sit adjacent in the grid.

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

**Results.** We see in Fig. 8 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  $\mathcal{O}\left(\frac{\|\mathbf{A}\mathbf{X}\mathbf{W}\|_{\mathbf{O}}^{t+1} \|\mathbf{X}\|_2}{(t+1)!}\right)$  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.

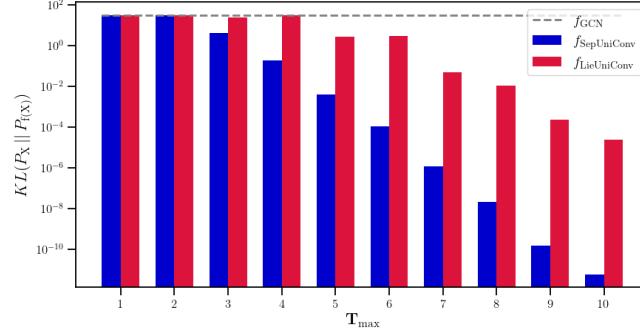


Figure 8. KL divergence between distribution of Rayleigh quotients before and after applying the model. Results are averaged over 10 runs.

In the [supplementary material](#) we include a video that shows the evolving Rayleigh quotient distribution as we increase  $T_{\max}$ . [Fig. 8](#) 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. 9](#), we see that the training runs in [Fig. 3](#) occur frequently, with only a couple runs diverging.

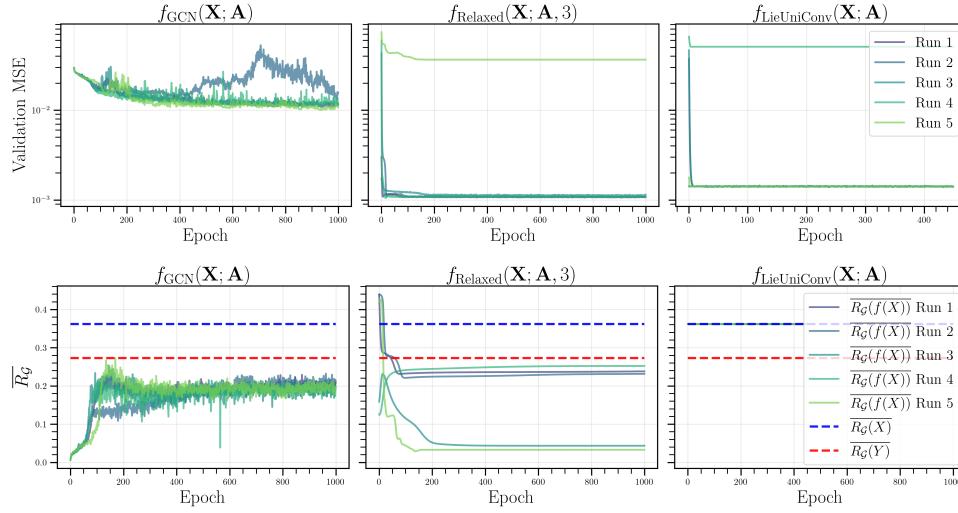


Figure 9. **Top:** Validation MSE for an ensemble of 5 runs for a GCN (left), R-UNIGRAPH (middle), and a Lie unitary convolution network (right) at timestep  $t = 3$ . R-UNIGRAPH significantly outperforms the GCN and also outperforms the Lie unitary network. **Bottom:** The average Rayleigh quotient 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 over constrained and can not model the Rayleigh quotient of the labels because it is forced to preserve the Rayleigh quotient of the input graphs.

## C. 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.

### C.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  $\alpha$  be the thermal diffusivity, and  $c$  a constant. The heat and wave equations on the mesh are then

given by

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

We now define the Cahn-Hilliard equation (Cahn & Hilliard, 1958). Let  $c$  be the fluid concentration,  $M$  the diffusion coefficient,  $\mu$  the chemical potential,  $f$  the double-free energy function, and  $\lambda$  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.

## C.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 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 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.

## C.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:

$$\text{RE}(f) = \frac{1}{\mathbf{T}_{\max}} \sum_t^{\mathbf{T}_{\max}} |R_{\mathcal{M}}(\mathbf{Y}_t) - R_{\mathcal{M}}(f(\mathbf{X}_t))|.$$

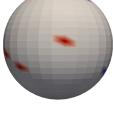
Following Janny et al. (2023) and Pandya et al. (2025), we also consider the scale invariant metrics NRMSE and SMAPE averaged over the entire rollout:

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

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

**Smoothness Errors in Dynamics Models and How to Avoid Them**

---

	Mesh	Heat $T = 0$	Wave $T = 0$
1155			
1156			
1157			
1158			
1159			
1160			
1161			
1162	Armadillo		
1163			
1164			
1165			
1166			
1167			
1168	Bunny		
1169			
1170			
1171			
1172			
1173			
1174			
1175	Lucy		
1176			
1177			
1178			
1179			
1180			
1181			
1182	Sphere		
1183			
1184			
1185			
1186			
1187			
1188			
1189	Spider		
1190			
1191			
1192			
1193			
1194			
1195			
1196	Urn		
1197			
1198			
1199			
1200			
1201			
1202	Woman		
1203			
1204			
1205			
1206			
1207			
1208			
1209			

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

1210	Mesh	Cahn-Hilliard $T = 0$
1211		
1212		
1213		
1214		
1215		
1216	Bunny	
1217		
1218		
1219		
1220		
1221		
1222		
1223	Ellipsoid	
1224		
1225		
1226		
1227		
1228		
1229		
1230	Sphere	
1231		
1232		
1233		
1234		
1235		
1236		
1237	Super Toroid	
1238		
1239		
1240		
1241		

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

#### C.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.

#### C.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  $\delta : \mathbb{R}^3 \rightarrow \mathbb{R}$  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  $\xi$  given in Eq. 10:

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

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. 10 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  $\delta_{ij}$  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

Smoothness Errors in Dynamics Models and How to Avoid Them

Time	Truth	R-UNIMESH (Ours)	EMAN	Hermes	
1265					
1266					
1267					
1268					
1269					
1270					
1271					
1272	10				
1273					
1274					
1275					
1276					
1277					
1278	50				
1279					
1280					
1281					
1282					
1283					
1284					
1285	100				
1286					
1287					
1288					
1289					
1290					
1291					
1292	150				
1293					
1294					
1295					
1296					
1297					
1298					
1299	190				
1300					

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.

our smoothness error by

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

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

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

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.

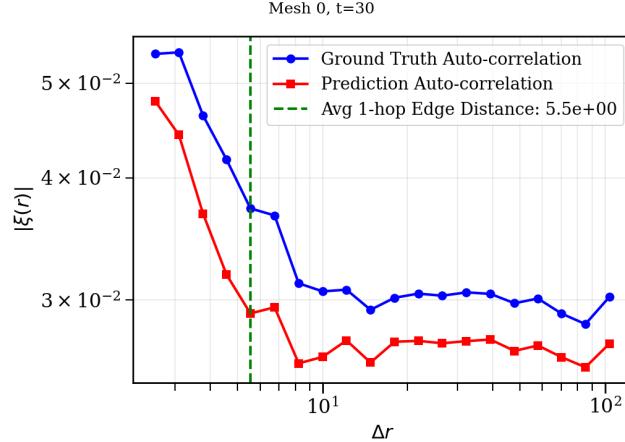


Figure 10. Smoothness for a Hermes model as measured by the 2-point correlation function. The plot indicates undersmoothing in each radial bin.

Heat ( $\alpha = 1$ )	
Model	err <sub>smooth</sub> ( $\downarrow$ )
GemCNN (de Haan et al., 2021)	–
EMAN (Basu et al., 2022)	<b><math>4.04 \times 10^{-3}</math></b>
Hermes (Park et al., 2023)	$9.71 \times 10^{-3}$
Wave ( $c = 1$ )	
Model	err <sub>smooth</sub> ( $\downarrow$ )
GemCNN (de Haan et al., 2021)	–
EMAN (Basu et al., 2022)	<b><math>1.78 \times 10^{-3}</math></b>
Hermes (Park et al., 2023)	$1.38 \times 10^{-2}$
Cahn–Hilliard	
Model	err <sub>smooth</sub> ( $\downarrow$ )
GemCNN (de Haan et al., 2021)	$1.89 \times 10^{-1}$
EMAN (Basu et al., 2022)	$4.59 \times 10^{-1}$
Hermes (Park et al., 2023)	<b><math>9.61 \times 10^{-3}</math></b>

Table 6. err<sub>smooth</sub> for Gauge Equivariant models on the PyVista Mesh datasets. Dashes (–) indicate non-convergence. Best performing model is indicated with **bold text**.

## D. WeatherBench2 Further Details

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

### D.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. C.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.

1375 **D.2. Evaluation Details**

1376 Here we give precise definitions of the evaluation and metrics omitted in the main text. We begin by establishing some  
 1377 notation common to the subsections, and consistent with the notation used in (Rasp et al., 2024).

1379 Let  $f$  denote the forecast,  $o$  the ground-truth observation, and  $c$  the climatology. Let  $t \in \{1, \dots, T\}$  denote the verification  
 1380 time,  $l \in \{1, \dots, L\}$  the lead time,  $i \in \{1, \dots, I\}$  the latitude index, and  $j \in \{1, \dots, J\}$  the longitude index. Forecasts are  
 1381 indexed as  $f_{t,l,i,j}$ , while observations and climatology are indexed by absolute time as  $o_{t,i,j}$  and  $c_{t,i,j}$ .

1382 **D.2.1. LATITUDE WEIGHTING**

1384 In an equiangular latitude-longitude grid, grid cells at the poles have a much smaller area compared to grid cells at the  
 1385 equator. Weighting all cells equally in the computation of RMSE and ACC would result in an inordinate bias towards the  
 1386 polar regions. As a result both metrics are latitude-weighted with weights computed as follows:

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

1391 where  $\theta_i^u$  and  $\theta_i^l$  indicate upper and lower latitude bounds, respectively.

1393 **D.2.2. CLIMATOLOGY**

1395 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  
 1396 1990 to 2019 (inclusive) for each grid point. A sliding window of 61 days is used around each day of year and time of  
 1397 day combination with weights linearly decaying to zero from the center. For notational consistency, we also define the  
 1398 lead-time-indexed climatology  $c_{t,l,i,j} := c_{t+l,i,j}$ , corresponding to the climatology at the forecast valid time.

1399 **D.2.3. ROOT MEAN SQUARED ERROR (RMSE)**

1400 Following the WB2 convention, our work measures error in terms of RMSE. For each variable and level pair, the RMSE at  
 1401 lead time  $l$  is defined as:

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

1402 This choice is important for temperature forecasting, as we are invariant to choice of unit (e.g., temperature in terms of  
 1403 Kelvin and Celsius will have the same RMSE). Moreover, the change in scale over time is less dramatic as it was for the  
 1404 PyVista meshes, where we considered NRMSE.

1405 **D.2.4. ANOMALY CORRELATION COEFFICIENT (ACC)**

1406 The ACC is computed as the Pearson correlation coefficient of the anomalies with respect to the climatology  $c$ . Denote the  
 1407 differences between forecast and climatology and between observation and climatology by

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

1408 The ACC at lead time  $l$  is then defined as

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

1409 ACC ranges from 1, indicating perfect correlation, to  $-1$ , indicating perfect anti-correlation. The ECMWF states that  
 1410 when the ACC value falls below 0.6, it is considered that the positioning of synoptic scale features ceases to have value for  
 1411 forecasting purposes.

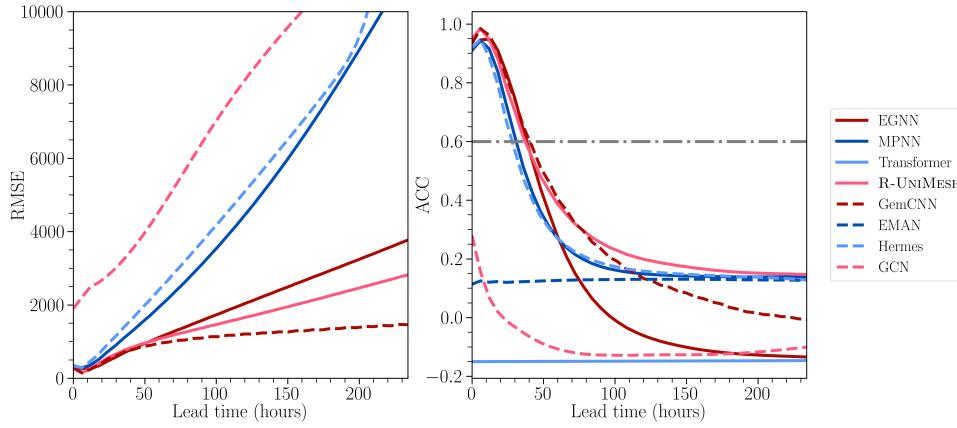
1412 **D.3. Earth Mesh Discretization**

1413 We construct a spherical mesh of the Earth by directly projecting the latitude-longitude grid points onto the unit sphere,  
 1414 and define mesh connectivity according to the original grid neighborhood structure. In order to obtain triangular faces, we

1430 further subdivide each cell into two triangles. The resulting mesh has 29040 nodes, 57600 faces, and 86640 edges. We  
 1431 note that this mesh construction is simpler than those used in other graph-based models, such as GraphCast (Lam et al.,  
 1432 2023), which employs a subdivided icosahedron as the underlying mesh. However, our approach has the advantage that it  
 1433 operates directly on the native latitude-longitude grid and therefore does not require interpolation or regridding of the ERA5  
 1434 data. Although more elaborate mesh constructions is likely to improve performance in real weather forecasting applications,  
 1435 our focus is on methodological experimentation rather than optimized weather prediction, and we therefore leave mesh  
 1436 optimization as an opportunity for future work.  
 1437

#### D.4. Geopotential Extended Results

1438 We report the RMSE and ACC curves for all lead times. We see in Fig. 11 that R-UNIMESH stays valid for the longest lead  
 1439 time in terms of ACC, and is rivaled only by GemCNN in terms of RMSE.  
 1440



1441 *Figure 11.* RMSE and ACC as a function of lead time for all models geopotential prediction. R-UNIMESH has a competitive RMSE,  
 1442 especially at early lead time. R-UNIMESH also maintains viability for lead times of roughly 2 days according to the ECMWF baseline.  
 1443

#### D.5. Smoothness Extended Results

1444 We provide smoothness errors for all models on WB2 temperature and geopotential datasets. As seen in Tab. 7, R-UNIMESH  
 1445 is competitive across both temperature and geopotential, and is within statistical significance of the best performing EGNN  
 1446 model on geopotential.  
 1447

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

1448 *Table 7.* Rayleigh error for all models for all initializations on WB2 temperature and geopotential. Best performing model is indicated  
 1449 with **bold text**. Errors scaled up by  $\times 40$ .  
 1450